arXiv:2602.16602v1 [math.CT] 18 Feb 2026

弱 ω-范畴中可逆性的类型论

Thibaut Benjamin
thibaut.benjamin@universite-paris-saclay.fr 巴黎-萨克雷大学，法国国家科学研究中心，巴黎-萨克雷高等师范学院，LMF 吉夫叙尔伊韦特，法国

Camil Champin
camil.champin@ens-lyon.fr
里昂高等师范学院 里昂，法国

Ioannis Markakis
ioannis.markakis@cl.cam.ac.uk
剑桥大学 剑桥，英国

摘要

我们提出依赖类型论 CaTT 的一个保守扩张 ICaTT，适用于弱 ω-范畴，并加入一种类型，用以见证胞腔的余归纳可逆性。该扩张能够将“游走等价”简洁地描述为一个上下文，并将一组刻画 ω-等纤维化的映射描述为代换。我们还提供了该类型论的实现，并用它形式化了可逆胞的基本性质。这些性质使我们能够给出 ICaTT 在带标记弱 ω-范畴中的语义，从而为 ICaTT 的每个模型构造一个纤维化带标记 ω-范畴。

1 引入

同伦类型论赋予每个类型一个弱高阶群胚结构，这种结构源自其迭代恒等类型塔 [1, 30, 39]。基于这一观察，Brunerie 开发了弱 ω-群胚的类型论表述 [10], ，Finster 和 Mimram [15] 则引入了有向变体 CaTT，这是一种用于弱 ω-范畴的类型论。本文中，我们引入 ICaTT，这是 CaTT 的扩展，将可逆性内化其中。

高阶范畴中的可逆性概念通常十分微妙。在普通范畴中，若一个态射 f存在一个逆 f^{-1}，使得

$$f \circ f^{-1} = \mathrm{id} \qquad\qquad f^{-1} \circ f = \mathrm{id}$$

在双范畴中，这些等式被可逆2-胞腔的存在所取代。用高维可逆胞腔逐步替代等式，可用于描述有限维范畴中的可逆性。对于 ω-范畴而言，由于它们是无限维的，这一过程应当以余归纳的方式理解，因此证明一个胞腔可逆，就意味着要构造无限量的数据。理论 ICaTT 是一种用于推理此类无限结构的句法方法。

可逆性在高阶范畴的同伦论中发挥着关键作用，尤其是在定义它们之间的弱等价时：这类函子在可逆性意义下是满射的，并且在同态高阶范畴上是弱等价。对于严格ω-范畴，其复合运算严格满足结合律和单位律，这些弱等价构成模型结构 [26] 的一部分。弱 ω-范畴，乃至弱 ω-群胚的类似结论仍未解决，但已在其他模型 [12, 29] 中得到证明。Henry 和 Lanari [21] 将群胚的情形解决到了 3 维，他们构造出一个等价于 3- 截断同伦类型的模型结构。在有向情形下，Fujii 等人 [18], 已开展了大量工作，证明弱等价满足 2/6 性质。同一批作者 [17] 还证明，等纤维化这一严格 ω-范畴中纤维化的类比，其特征由右提升性质刻画。Benjamin 等人 [6] 独立地在类型论 CaTT 中部分构造了有向圆柱 。将他们的自然性构造改编到理论 ICaTT 中，或可用于构造这些圆柱的无向类比，这是模型结构所要求的。

贡献。 我们提出了弱 ω-范畴类型论 CaTT 的扩展 ICaTT，其中包含定义在项 t上的可逆性结构类型 Inv(t)。该类型配备了析构子，用于为 t提供可逆性数据；同时还包含将可逆胞的闭包性质内化的构造子。我们还公设了通常的β/η规则，并证明所得理论对于一类重要的项是可规范化的。

我们为 ICaTT 提供了一个实现[1] ，并借此以最少的工作形式化了关于可逆性结构的既有结果；这些结果无法在理论 CaTT 中表达。我们将此视为证据，表明该理论非常适合处理可逆性结构，也可能有助于进一步研究如何在弱 ω-范畴上构造模型结构。

我们证明 ICaTT 是 CaTT 的保守扩张，并且二者相应句法范畴之间的包含是全忠实的。借助这一包含，我们可以从 ICaTT 的一个模型诱导出一个 ω-范畴，从而为 ICaTT 提供取值于 ω-范畴的语义。利用这些语义，我们得以重构 Ozornova 和 Rovelli [34], 提出的自由等价，该对象用于 Fujii 等人 [17]对等纤维化的刻画。最后，我们将 Henry 和 Loubaton [22]关于带标记 ω-范畴的定义推广到弱情形，并扩展语义，使其取值于纤维的带标记 ω-范畴。

相关工作。多年来，人们提出了若干种代数球状弱 ω-范畴的定义。Batanin [3] 和 Leinster [28]最先对其进行了研究。Maltsiniotis[31] 则独立提出了一种弱 ω-范畴模型，其灵感源自 Grothendieck [19]提出的弱 ω-群胚定义。Ara [2] 和 Bourke [9] 证明了这些定义彼此等价。

CaTT 理论由 Finster 和 Mimram [15], 提出，其灵感来自 Maltsiniotis 和 Brunerie [10]。Ben- jamin 等人 [8]随后证明，CaTT 的模型与 Grothendieck-Maltsiniotis 方法完全对应。最后，Dean 等人[14]提出了另一种基于计算图式的描述，并证明它与 Batanin-Leinster 定义等价。最终，Benjamin 等人 [7] 完成了整个对应关系的证明，表明 CaTT 上下文与有限 computad 相对应。

弱 ω-范畴中的可逆性已在各种模型中得到研究。Cheng [13] 和 Rice 最初以一种与模型无关的方式进行了研究

[1]可在 https://zenodo.org/records/18343317 获取

[35], , 最近 Hoshino 等人 [23] 以及 Benjamin和Markakis [4] 针对弱 ω-范畴开展了研究。Ozornova 和 Rovelli [34] 引入了ω-范畴的游走等价，而 Hadzihasanovic 等人 [20]证明了它在严格情形下是可缩的。在弱情形下，Fujii 等人 [17]利用它刻画了等纤维化。

论文结构。第 2 节介绍依赖类型论 CaTT。第 3 节介绍理论的模型这一概念，并用它来定义弱 ω-范畴。我们在第 4 节引入理论 ICaTT，随后在第 5 节将可逆性的一些构造形式化。第 6 节证明 ICaTT 对 CaTT 是保守的，并证明可以将游走等价恢复为 ICaTT 中的一个上下文。最后，第 7 节引入带标记的弱 ω-范畴，并构造 ICaTT 在纤维化带标记 ω-范畴中的语义。

2 类型论 CaTT

我们先简要介绍 Finster 和 Mimram 的依赖类型论 CaTT [15], , 其模型是弱 ω-范畴，然后回顾其上的悬挂元操作。CaTT 的原始语法由下列文法定义，其中 V 是一个可数的变量名集合：

Variable	x	::=	$x \in \mathcal{V}$		
Type	A, B	::=	$\star$	|	$t \to_A u$
Term	t, u	::=	x	|	$\mathrm{coh}_{\Gamma,A}[\gamma]$
Context	Γ, Δ	::=	$\varnothing$	|	$\Gamma \triangleright (x : A)$
Substitution	γ, δ	::=	$\diamond$	|	$\gamma \triangleright (x \mapsto t)$

代换对项和类型的作用，以及代换的复合，由以下公式互递归地定义：

$$\star[\gamma] = \star \qquad (u \to_A v)[\gamma] = u[\gamma] \to_{A[\gamma]} v[\gamma]$$

$$x[\diamond] = x \qquad x[\gamma \triangleright (y \mapsto t)] = \begin{cases} t & \text{if } y = x \\ x[\gamma] & \text{otherwise} \end{cases}$$

$$\mathrm{coh}_{\Gamma,A}[\delta][\gamma] = \mathrm{coh}_{\Gamma,A}[\delta \circ \gamma]$$

$$\diamond \circ \gamma = \diamond \qquad (\delta \triangleright (x \mapsto t)) \circ \gamma = (\delta \circ \gamma) \triangleright (x \mapsto t[\gamma])$$

类型和上下文的维数 递归定义如下：

$$\dim(\star) = -1 \qquad \dim(u \to_A v) = \dim A + 1$$
$$\dim(\varnothing) = -1 \qquad \dim(\Gamma \triangleright (x : A)) = \max(\dim \Gamma, \dim A + 1)$$

与依赖类型论中的常见做法一样，CaTT 以以下判断的形式表述；这些判断依据一组推导规则归纳定义。

$\Gamma \vdash$	Γ 是有效上下文
$\Gamma \vdash A$	A 是 Γ中的有效类型
$\Gamma \vdash t\colon A$	t 是上下文 Γ中类型为 A 的项
$\Gamma \vdash \gamma\colon \Delta$	γ 是将 Δ 代入 Γ的有效代换。

首先，我们要求变量、上下文和代换满足通常的结构规则，其中 $x \in \mathrm{Var}\,\Gamma$ 和 $(x\colon A) \in \Gamma$ 是关系以显然的方式递归定义。

$$\frac{}{\varnothing \vdash} \qquad \frac{\Gamma \vdash A}{\Gamma \triangleright (x\colon A) \vdash}\ (x \notin \mathrm{Var}\,\Gamma) \qquad \frac{\Gamma \vdash}{\Gamma \vdash x\colon A}\ (x\colon A) \in \Gamma$$

$$\frac{\Gamma \vdash}{\Gamma \vdash \diamond\colon \varnothing} \qquad \frac{\Delta \vdash \gamma\colon \Gamma \qquad \Gamma \vdash A \qquad \Delta \vdash t\colon A[\gamma]}{\Delta \vdash \gamma \triangleright (x \mapsto t)\colon \Gamma \triangleright (x\colon A)}$$

这些规则的一个推论是，存在恒等代换 $\Gamma \vdash \mathrm{id}_\Gamma\colon \Gamma$，它将每个变量映射到自身。

CaTT 的类型形成规则如下，其中最后两条规则在早期的表述中合并为一条：

$$\star\text{-INTRO}\ \frac{\Gamma \vdash}{\Gamma \vdash \star} \qquad \to\text{-INTRO}_0\ \frac{\Gamma \vdash u\colon \star \qquad \Gamma \vdash v\colon \star}{\Gamma \vdash u \to_\star v}$$

$$\to\text{-INTRO}_+\ \frac{\Gamma \vdash u\colon v \to_A w \qquad \Gamma \vdash u'\colon v \to_A w}{\Gamma \vdash u \to_{v \to_A w} u'}$$

为便于阅读，我们有时会在表达式 $u \to_A v$ 中省略下标 A，因为它可以推断出来。请注意，为便于阅读，我们使用了命名变量，但将相差 α-重命名的上下文视为相同。特别是，我们也可以改用 De Bruijn 指数来等价地表述该理论。我们将项 $\Gamma \vdash t\colon A$ 的维数 定义为 $\dim A + 1$。当$A = u \to v$时，我们分别称 u 和 v为 t 的源 和目标 。

为了陈述 CaTT 的项形成规则，我们引入两个辅助判断 $\Gamma \vdash_{\mathrm{ps}}$ 和 $\Gamma \vdash_{\mathrm{ps}} x\colon A$ 它们以参数化方式定义一组上下文，这些上下文对应于粘贴图，亦即 globular ω-范畴中运算的元数。下面给出这些判断的规则：

$$\mathrm{PSS}\ \frac{}{(x\colon \star) \vdash_{\mathrm{ps}} x\colon \star} \qquad \mathrm{PSD}\ \frac{\Gamma \vdash_{\mathrm{ps}} f\colon x \to_A y}{\Gamma \vdash_{\mathrm{ps}} y\colon A}$$

$$\mathrm{PSE}\ \frac{\Gamma \vdash_{\mathrm{ps}} x\colon A}{\Gamma \triangleright (y\colon A) \triangleright (f\colon x \to_A y) \vdash_{\mathrm{ps}} f\colon x \to_A y} \qquad \mathrm{PS}\ \frac{\Gamma \vdash_{\mathrm{ps}} x\colon \star}{\Gamma \vdash_{\mathrm{ps}}}$$

源（分别为目标）变量 是指不是任何其他变量的目标（分别为源）的变量。我们记作 $\Gamma \vdash_{\mathrm{full}} u \to_A v$，并称 $u \to_A v$ 是维数为n 的粘贴图 Γ 的满类型 ，当且仅当满足以下两个条件之一：

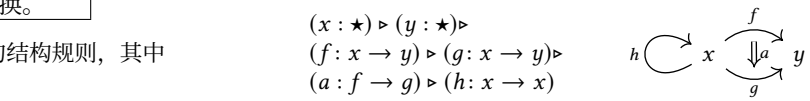
- u和 v都使用了 Γ的所有变量，
- $\dim u = \dim v = n-1$，且 u和 v分别使用了维数至多为 $n-1$的 Γ的所有源变量和目标变量。

根据 Dean 等人 [14, Proposition2.10]的工作，这个满类型的定义等价于原始定义。因此，类型论 CaTT 的最后一条规则为：

$$\text{coh-INTRO}\ \frac{\Gamma \vdash_{\mathrm{ps}} \qquad \Gamma \vdash_{\mathrm{full}} A \qquad \Delta \vdash \gamma\colon \Gamma}{\Delta \vdash \mathrm{coh}_{\Gamma,A}[\gamma]\colon A[\gamma]}$$

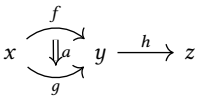
例 *2.1*。 以下是 CaTT 的一个有效上下文。我们用点表示类型为 $\star$的变量，用从 u到 v的箭头表示类型为 $u \to v$的变量。

$$(x : \star) \triangleright (y : \star) \triangleright$$
$$(f\colon x \to y) \triangleright (g\colon x \to y) \triangleright$$
$$(a : f \to g) \triangleright (h\colon x \to x)$$



例 *2.2*。 以下是与粘贴图对应的一个有效上下文

$$(x\colon\star)\triangleright(y\colon\star)\triangleright(f\colon x\to y)\triangleright (g\colon x\to y)\triangleright(a\colon f\to g)\triangleright (z\colon\star)\triangleright(h\colon y\to z)$$



该上下文的源变量是 f、 a和 h，而目标变量是 g、 a和 h。

例 *2.3.* 我们将推导出与恒等运算和复合运算对应的项。对于前者，我们考虑以下项:

$$(x\colon\star)\vdash \mathsf{coh}_{(x\colon\star),x\to x}[\mathsf{id}]\colon x\to x$$

我们将其记为 $\mathsf{id}\,x$。对于后者，考虑上下文 Γ:

$$(x\colon\star)\triangleright(y\colon\star)\triangleright(f\colon x\to y)\triangleright (z\colon\star)\triangleright(g\colon y\to z)\qquad x\xrightarrow{f} y\xrightarrow{g} z$$

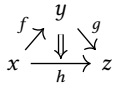
类型 $x\to_\star z$ 是满的，因此我们可以推导出以下项的有效性:

$$\Gamma\vdash \mathsf{coh}_{\Gamma,x\to z}[\mathsf{id}]\colon x\to z$$

我们将其记为 $f*g$。

例*2.4.* 以下是一个有效上下文，但无法将其视为粘贴图，因为 α 的源是一个非变量项:

$$(x\colon\star)\triangleright(y\colon\star)\triangleright(f\colon x\to y)\triangleright (z\colon\star)\triangleright(g\colon y\to z)\triangleright(h\colon x\to z)\triangleright (\alpha\colon f*g\to h)$$



悬挂 在语义上对应于 Benjamin 和 Markakis [5] 的 hom ω-范畴的构造，是一种元操作。同伦类型论学者或许会注意到，任何关于恒等类型的构造，都可以通过改变基类型来解释为高阶恒等类型上的构造。悬挂通过对有效上下文、类型、项和代换进行相互递归定义:

$$\Sigma(\varnothing)=(v^-\colon\star)\triangleright(v^+\colon\star)\qquad \Sigma(\Gamma\triangleright(x\colon A))=(\Sigma\Gamma)\triangleright(\Sigma x\colon\Sigma A)$$

$$\Sigma(\star)=v^-\to_A v^+\qquad \Sigma(u\to_A v)=\Sigma u\to_{\Sigma A}\Sigma v\qquad \Sigma(x)=x$$

$$\Sigma(\mathsf{coh}_{\Gamma,A}[\gamma])=\mathsf{coh}_{\Sigma\Gamma,\Sigma A}[\Sigma\gamma]\qquad \Sigma(\diamond)=(v^-\mapsto v^-)\triangleright(v^+\mapsto v^+)$$

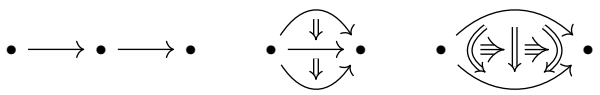
$$\Sigma(\gamma\triangleright(x\mapsto t))=(\Sigma\gamma)\triangleright(\Sigma x\mapsto\Sigma t)$$

这里假设变量 v^- 和 v^+ 在 Γ 中是新鲜的；使用 De Bruijn 指数可以轻松实现这一点。此运算保留了 CaTT 的所有判断，包括关于粘贴图的辅助判断。

例 *2.5*。 对例 2.3 中的恒等项和复合项反复应用悬挂，我们也可以得到对应于恒等和高维项复合的项。更确切地说，我们可以推导出以下项:

$$\Sigma^n(x\colon\star)\vdash\Sigma^n(\mathsf{id}\,x)\colon x\to x\qquad \Sigma^n\Gamma\vdash\Sigma^n(f*g)\colon x\to z$$

我们仍将其简单地记作 $\mathsf{id}\,x$ 和 $f*g$。我们将在下文说明上下文 Γ、$\Sigma\Gamma$ 和 $\Sigma^2\Gamma$。恒等项和复合项都会使用其定义所在上下文中的所有变量。



圆盘和球是两族上下文，分别用于归类项和类型。它们递归定义如下:

$$\mathbb{D}^0=(\mathsf{d}_0\colon\star)\qquad \mathbb{D}^{n+1}=\Sigma\mathbb{D}^n\qquad \mathbb{S}^{-1}=\varnothing\qquad \mathbb{S}^{n+1}=\Sigma\mathbb{S}^n$$

我们将 $\mathbb{D}^n$ 的最高维变量记为 d_n。对于每个 n，通过悬挂空代换 $\mathbb{D}^0\vdash\diamond\colon\mathbb{S}^{-1}$，可得到一个代换 $\mathbb{D}^n\vdash\iota_n\colon\mathbb{S}^{n-1}$。这些句法对象的泛性质由 Benjamin 等人的以下引理给出: [8, 引理 2.3.8]:

引理 2.6。 *CaTT* 中维数为 n 的类型 $\Gamma\vdash A$与代换 $\Gamma\vdash\chi_A\colon\mathbb{S}^n$ 之间存在自然双射；维数为 n 的项 $\Gamma\vdash t\colon A$ 与代换 $\Gamma\vdash\chi_t\colon\mathbb{D}^n$之间也存在自然双射。在这些双射下，与 ι_n 复合会将一个项映射为其类型。

3 弱 ω-范畴

本节中，我们回顾族这一描述类型论模型的方式。然后，我们将弱 ω-范畴定义为 CaTT 的模型，并讨论弱ω-范畴之间的等价与等纤维化。

3.1 族的模型

我们将 CaTT 视为 Cartmell [11] 意义上的广义代数理论，并采用 Joyal [25] 的族框架——即 Taylor [37] 带有展示映射的范畴的一种变体——来描述其模型。得益于 Benjamin 等人 [8, 的引理 1.13], 这一 CaTT 模型概念与先前使用的概念一致。

定义 *3.1.* 一个族是带有终对象的范畴 C，并配备一类箭头，该类包含每一个同构以及每一个到终对象的态射，且在复合与基变换下稳定。我们将此类中的箭头称为展示映射，并用 $\twoheadrightarrow$ 表示它们。

所谓在基变换下稳定，是指对每一个展示映射 $p\colon A\twoheadrightarrow B$ 以及每一个态射 $f\colon C\to B$，如下拉回方块存在:

$$\begin{array}{ccc} D & \xrightarrow{p^*f} & A \\ {\scriptstyle f^*p}\downarrow & \lrcorner & \downarrow{\scriptstyle p} \\ C & \xrightarrow[f]{} & B \end{array}\qquad (1)$$

并且 f^*p 仍是展示映射。

定义 *3.2*。 族态射 $C\to D$ 是保持终对象、展示映射类以及沿展示映射的拉回的函子。族态射之间的变换就是函子的自然变换。我们将部族的严格 2-范畴记作 Clan。

例 *3.3.* 依赖类型论的句法范畴 $\mathsf{S}_\mathbb{T}$ 的对象是模判断相等的上下文，态射 $\Gamma\to\Delta$ 是模判断相等的 $\Delta\vdash\gamma$代换 Γ。它具有族的结构，其展示映射是形如 $\Gamma\triangleright(x\colon A)\vdash\mathsf{p}_{\Gamma,A}\colon\Gamma$ 的弱化代换在复合和同构下的闭包，其中每个 Γ 变量都映到自身。此类映射在基变换下稳定，因为

每个 $\Delta \vdash \gamma\colon \Gamma$，下面的方块在句法范畴中是一个拉回：

$$\begin{array}{ccc} \Delta \triangleright (x\colon A[\gamma]) & \xrightarrow{(\gamma \circ \mathsf{p}_{\Delta,A[\gamma]}) \triangleright (x \mapsto x)} & \Gamma \triangleright (x\colon A) \\ {\scriptstyle \mathsf{p}_{\Delta,A[\gamma]}}\downarrow & \lrcorner & \downarrow{\scriptstyle \mathsf{p}_{\Gamma,A}} \\ \Delta & \xrightarrow[\gamma]{} & \Gamma \end{array} \tag{2}$$

例如，Benjamin 等人 [8, 引理 1.1.1] 已证明这一点。

定义 *3.4.* 有限完备范畴 C 是一个族，其中每个态射都是展示映射。我们将类型论 $\mathbb{T}$ 的模型范畴定义为 $\mathrm{Mod}(\mathbb{T}) = \mathrm{Clan}(\mathsf{S}_{\mathbb{T}}$，集合$)$。

由于集合中的每个态射都是展示映射，且拉回方块可以彼此粘合，因此类型论 $\mathbb{T}$ 的一个模型恰好是一个函子 $F\colon \mathsf{S}_{\mathbb{T}} \to$ 集合，它保持终对象以及对每个类型 $\Gamma \vdash A$ 和代换 $\Delta \vdash \gamma\colon \Gamma$对应的形如 (2) 的拉回方块。由于协变可表示函子是连续的，因此存在一个忠实满嵌入：

$$[\![-]\!]\colon \mathcal{S}_{\mathbb{T}}^{\mathrm{op}} \to \mathrm{Mod}(\mathbb{T})$$
$$[\![\Gamma]\!](\Delta) = \mathcal{S}_{\mathbb{T}}(\Gamma, \Delta)$$

我们注意到，模型范畴是从 $\mathsf{S}_{\mathbb{T}}$ 到集合的函子范畴的一个反射子范畴，并且在滤过余极限下封闭，因此特别是局部有限呈示的，正如 Frey [16, 注释 2.9]所说明的。

3.2 CaTT 的模型

Benjamin 等人 [7] 和 Benjamin 等人 [8] 分别精确描述了句法范畴和 CaTT 的模型，将前者识别为 Dean 等人 [14] 所述的有限 computad 范畴，将后者识别为 Leinster [28] 所述的弱 ω- 范畴与严格 ω-函子范畴。这些弱 ω-范畴是 Batanin [3] 提出的弱范畴的一种变体，并且如 Ara [2] 和 Bourke [9]所示，它们是 Grothendieck [19] 和 Maltsiniotis [31], 意义下的相干器模型。

定义 *3.5.* 弱 -范畴与严格 ω-函子构成的范畴Mod(CaTT) 是CaTT的模型范畴。

我们将 n-胞腔定义为 ω-范畴 X的 $X_n = X(\mathbb{D}^n)$ 中的元素。根据 Yoneda 引理，它们与态射 $[\![\mathbb{D}^n]\!] \to X$自然双射。源函数和目标函数 src,tgt: $X_{n+1} \to X_n$ 是通过与代换 $\mathbb{D}^{n+1} \to \mathbb{D}^n$ 复合得到的；这些代换根据引理 2.6 分别对应于 $\mathbb{D}^{n+1}$ 的最高维变量 d_{n+1} 的源和目标。我们用 $x\colon \star$ 表示 $x \in X_0$，用 $x\colon y \to z$ 表示 x是一个正维胞腔，其源为 y，目标为 z。

将（2）应用于引理 2.6 给出的代换，可得每个弱化代换都是代换$\mathbb{D}^n \vdash \iota_n\colon \mathbb{S}^{n-1}$的拉回；根据构造，这些代换本身也是弱化代换：

$$\begin{array}{ccc} \Gamma \triangleright (x\colon A) & \xrightarrow{\chi_x} & \mathbb{D}^n \\ {\scriptstyle \mathsf{p}_{\Gamma,A}}\downarrow & \lrcorner & \downarrow{\scriptstyle \iota_n} \\ \Gamma & \xrightarrow[\chi_A]{} & \mathbb{S}^{n-1} \end{array} \tag{3}$$

由拉回的粘贴引理可知，函子$X\colon \mathsf{S}_{\mathrm{CaTT}} \to$ 集合是模型，当且仅当 $X(\varnothing)$是终对象，并且它保持形如(3)的拉回方块。此外，对上下文的长度作归纳法可知，模型态射$f\colon X \to Y$完全由其在 X的胞腔上的取值决定，并且它可逆当且仅当它在胞腔上是双射。

定义 *3.6*。 每个维数为 n 的项 $\Gamma \vdash t\colon A$ 都会导出一个运算：

$$[\![t]\!]\colon \omega\mathbf{Cat}([\![\Gamma]\!], X) \to X_n$$
$$[\![t]\!](f) = f \circ [\![\chi_t]\!]$$

In 特别地，例 2.5 中的项会导出运算

$$\mathsf{id}\colon X_n \to X_{n+1} \qquad -*-\colon X_n \times_{X_{n-1}} X_n \to X_n$$

因为从它们的定义上下文出发的态射分别对应于一个n-胞腔，以及一对可复合的 n-胞腔。这些运算的源和目标如下：

$$\mathrm{src}(\mathsf{id}\, x) = \mathrm{tgt}(\mathsf{id}\, x) = x \qquad \mathrm{src}(f * g) = \mathrm{src}\, f \qquad \mathrm{tgt}(f * g) = \mathrm{tgt}\, g$$

3.3 可逆胞腔

我们以研究 ω-范畴中的可逆胞 来结束本节。这些胞腔使我们能够描述 ω-范畴之间的弱等价 和等纤维化。这两类映射在严格 ω-范畴的同伦论中发挥着重要作用，正如 Lafont 等人 [26], 所展示的那样；我们预计它们在弱情形中也会发挥类似作用。

定义 *3.7.* 胞腔 $f\colon x \to y$ 在ω-范畴 X 中的可逆性结构 余归纳地定义为由以下胞腔组成：

$$f^{\mathsf{linv}}\colon y \to x \qquad f^{\mathsf{rinv}}\colon y \to x$$
$$f^{\mathsf{lunit}}\colon f^{\mathsf{linv}} * f \to \mathsf{id}\, y \qquad f^{\mathsf{runit}}\colon f * f^{\mathsf{rinv}} \to \mathsf{id}\, x$$

以及 f^{lunit} 和 f^{runit}上的可逆性结构。若 f具有可逆性结构，我们就称其为可逆 的。

我们注意到，这种可逆胞的概念先验上弱于通常的概念；在通常的概念中，左逆 f_{linv} 与右逆 f_{rinv}严格相等。尽管如此，Rice [35] 以及 Fujii 等人 [17, 命题 3.3.1] 已分别证明，这两种概念实际上等同。我们采用这种可逆性结构定义，受到 Hadzihasanovic 等人 [20] 关于游走等价 的工作，以及 Fujii 等人 [18] 关于等纤维化 的工作的启发。类比同伦类型论中的 [38, 定理 4.3.2], , 我们预计，要求分别给出左逆和右逆，应当会使胞腔 f上的可逆性结构唯一。

Benjamin和Markakis [4] 以及 Hoshino 等人 [23] 独立证明了，可逆胞在弱 ω-范畴的运算下封闭。更确切地说，由可逆的 n-胞构造出的任何 n 维胞腔本身也是可逆的。这涵盖了可逆胞的复合，也涵盖了所有恒等胞腔和结合子、单位子等高阶相干性。

定理 3.8。 设 $\gamma\colon [\![\Gamma]\!] \to X$ 是一个从粘贴图出发的 ω-范畴态射，且 $\Gamma \vdash_{\mathsf{full}} u \to v$ 是一个维数为 n 的满类型。当对每个 $(n+1)$ 维变量 $x \in \mathrm{Var}(\Gamma)$， $[\![x]\!](\gamma)$ 均可逆时，胞腔 $[\![\mathrm{coh}_{\Gamma,u\to v}]\!](\gamma)$ 可逆。

该定理的证明通过余归纳地为这种形式的胞腔构造可逆结构，其左逆等于

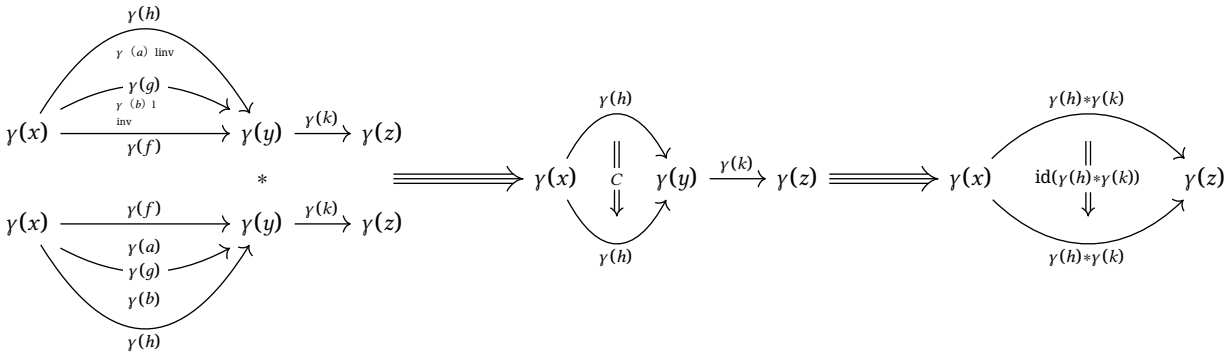


图 1：可逆胞复合的左消去元 $C = \gamma(b)^{\mathsf{linv}} * (\gamma(a)^{\mathsf{linv}} * \gamma(a)) * \gamma(b$).

右逆。当 $\dim \Gamma \le n$ 时，对 Γ 的条件是空泛的，逆元为 $[\![\mathrm{coh}_{\Gamma,\ v\to u}]\!](\gamma)$，消去元也仍为 $[\![\mathrm{coh}_{\Gamma,\ A}]\!](\gamma)$ 这种形式。$\dim \Gamma = n+1$ 的情形对应于可逆胞的复合，图 1 对此作了部分说明，其中第一个 3-胞腔是一个广义结合子，第二个则是消去元和单位子的复合。关于这一构造的更详细说明见附录 B。此外，Benjamin 和 Markakis [4, 命题 37] 证明了定理 3.8 的逆命题，其中 X 是一个 CaTT 上下文：

命题 3.9. </style}项：A 在维数 n 上可逆于$[\![\Delta]\!]$，当且仅当它具有如下形式：$t = \mathrm{coh}_{\Gamma,A}[\gamma]$，其中每个 n维的 $x \in \mathrm{Var}(\Gamma)$ 都是可逆的。

可逆性在弱等价的定义中起着关键作用，弱等价即 ω-范畴之间本质满射于对象和迭代同态集的 ω-函子。人们猜想，这些等价属于某种模型结构，该模型结构的余纤维化由球嵌入圆盘的包含映射生成。Fujii 等人定义等纤维化时也用到了可逆性。 [18], 等纤维化是范畴等价纤维化的类比，预计会像严格情形 [26]那样，在猜想的模型结构中发挥重要作用。

定义 3.10. </style}函子 ω-函子 $f\colon X \to Y$ 是 ω-范畴之间的 ω-等纤维化，当且仅当对于 X中的每个 n-胞腔 x，以及 Y中的一个可逆胞腔 $e\colon y \to f(x)$，都存在 X 中的一个可逆 n- 胞腔 $\bar{e}\colon \bar{y} \to x$，使得 $f\bar{e} = e$。

4 ICaTT 类型论

本节将依赖类型论 ICaTT 定义为 CaTT 的扩展，它允许为项配备可逆性结构。

4.1 原始语法

我们首先在 CaTT 的原始语法中扩展一个类型构造子 $\mathrm{Inv}_A(u)$，其中 A 是一个类型，u 是一个项；并加入九个项构造子：

$$\mathsf{can}(t, \{e_x\})$$
$$\mathsf{inv}(t, t_l, t_r, t_{lu}, t_{ru}, t_{ilu}, t_{iru})\ \mathsf{rec}(t, t_l, t_r, t_{lu}, t_{ru}, t_{ilu}, t_{iru}, \gamma)$$
$$\mathsf{linv}(t)\ \mathsf{rinv}(t)\ \mathsf{lunit}(t)\ \mathsf{runit}(t)\ \mathsf{ilunit}(t)\ \mathsf{irunit}(t)$$

其中，γ 是一个代换，$\{e_x\}$ 是一个由变量集合索引的项族，其余参数均为项。我们通过以下方式将原始代换的作用扩展到这些新类型和项：

$$\mathsf{can}(t, \{e_x\})[\sigma] = \mathsf{can}(t[\sigma], \{e_x[\sigma]\})$$
$$\mathsf{rec}(t, t_l, t_r, t_{lu}, t_{ru}, t_{ilu}, t_{iru}, \gamma)[\sigma] = \mathsf{rec}(t, t_l, t_r, t_{lu}, t_{ru}, t_{ilu}, t_{iru}, \gamma \circ \sigma)$$

并使代换在每一种类型和项构造子下都能计算，如 C 节所述。我们还通过以下方式扩展悬垂运算在原始语法上的作用：

$$\Sigma\mathsf{inv}(t, t_l, t_r, t_{lu}, t_{ru}, t_{ilu}, t_{iru}) = \mathsf{inv}(\Sigma t, \Sigma t_l, \Sigma t_r, \Sigma t_{lu}, \Sigma t_{ru}, \Sigma t_{ilu}, \Sigma t_{iru})$$
$$\Sigma\mathsf{rec}(t, t_l, t_r, t_{lu}, t_{ru}, t_{ilu}, t_{iru}, \gamma) = \mathsf{rec}(\Sigma t, \Sigma t_l, \Sigma t_r, \Sigma t_{lu}, \Sigma t_{ru}, \Sigma t_{ilu}, \Sigma t_{iru}, \Sigma\gamma)$$
$$\Sigma(\mathsf{Inv}_A(t)) = \mathsf{Inv}_{\Sigma A}(\Sigma t) \qquad \Sigma(\mathsf{can}(t, \{e_x\})) = \mathsf{can}(\Sigma t, \{\Sigma e_x\})$$
$$\Sigma(\mathsf{linv}(e)) = \mathsf{linv}(\Sigma e) \qquad \Sigma(\mathsf{rinv}(e)) = \mathsf{rinv}(\Sigma e)$$
$$\Sigma(\mathsf{lunit}(e)) = \mathsf{lunit}(\Sigma e) \qquad \Sigma(\mathsf{runit}(e)) = \mathsf{runit}(\Sigma e)$$
$$\Sigma(\mathsf{ilunit}(e)) = \mathsf{ilunit}(\Sigma e) \qquad \Sigma(\mathsf{irunit}(e)) = \mathsf{irunit}(\Sigma e)$$

4.2 可逆性结构的类型

描述完原始语法后，我们接下来给出 ICaTT 的规则。首先假设 CaTT 的每条规则都由 ICaTT 继承。类型 Inv（可逆性结构）的引入规则如下：

$$\textsc{Inv-intro}\ \frac{\Gamma \vdash t : u \to_A v}{\Gamma \vdash \mathsf{Inv}_{u\to_A v}(t)}$$

为便于阅读，有时在可以推断出索引 A 时会将其省略。为了使项 $\Gamma \vdash e\colon \mathsf{Inv}(t)$具备 t 可逆性结构，我们提供六个析构子和一个对偶构造子：

$$\frac{\Gamma \vdash e\colon\ \mathsf{Inv}_{u\to v}(t)}{\Gamma \vdash \mathsf{linv}(e) : v \to u} \qquad \frac{\Gamma \vdash e\colon\ \mathsf{Inv}_{u\to v}(t)}{\Gamma \vdash \mathsf{rinv}(e) : v \to u}$$

$$\frac{\Gamma \vdash e\colon\ \mathsf{Inv}_{u\to v}(t)}{\Gamma \vdash \mathsf{lunit}(e) : \mathsf{linv}(e) * t \to \mathsf{id}\, v} \qquad \frac{\Gamma \vdash e\colon\ \mathsf{Inv}_{u\to v}(t)}{\Gamma \vdash \mathsf{runit}(e) : t * \mathsf{rinv}(e) \to \mathsf{id}\, u}$$

$$\frac{\Gamma \vdash e : \mathrm{Inv}_{u\to v}(t)}{\Gamma \vdash \mathrm{ilunit}(e) : \mathrm{Inv}(\mathrm{lunit}(e))} \qquad \frac{\Gamma \vdash e : \mathrm{Inv}_{u\to v}(t)}{\Gamma \vdash \mathrm{irunit}(e) : \mathrm{Inv}(\mathrm{runit}(e))}$$

$$\text{inv-intro}\ \frac{\begin{array}{c}\Gamma \vdash t : x \to y \qquad \Gamma \vdash t_l : y \to x \qquad \Gamma \vdash t_r : y \to x \\ \Gamma \vdash t_{lu} : t_l * t \to \mathrm{id}\, y \qquad \Gamma \vdash t_{ru} : t * t_r \to \mathrm{id}\, x \\ \Gamma \vdash t_{ilu} : \mathrm{Inv}(t_{lu}) \qquad \Gamma \vdash t_{iru} : \mathrm{Inv}(t_{ru})\end{array}}{\Gamma \vdash \mathrm{inv}(t,\ t_l,\ t_r,\ t_{lu},\ t_{ru},\ t_{ilu},\ t_{iru}) : \mathrm{Inv}(t)}$$

此外，我们要求这些构造子满足通常的 β-归约规则和 η-扩展规则，完整呈现在图 4d 中。例如，我们要求以下 η-规则和 β-规则：

$$e \equiv \mathrm{inv}(t, \mathrm{linv}(e), \mathrm{rinv}(e), \mathrm{lunit}(e), \mathrm{runit}(e), \mathrm{ilunit}(e), \mathrm{irunit}(e))$$
$$\mathrm{linv}(\mathrm{inv}(t, t_l, t_r, t_{lu}, t_{ru}, t_{ilu}, t_{iru})) \equiv t_l$$

4.3 相干项的可逆性

定理 3.8 提供了一种为相干胞腔构造可逆性结构的方法。项构造子 can 使我们能够在 ICaTT 中将此定理内化。其引入规则如下：

$$\text{can-intro}\ \frac{\Delta \vdash \mathrm{coh}_{\Gamma,A}[\gamma] : A[\gamma] \quad \{\Delta \vdash e_x : \mathrm{Inv}(x[\gamma])\}_{x \in \mathrm{Var}_{\dim A+1}(\Gamma)}}{\Delta \vdash \mathrm{can}(\mathrm{coh}_{\Gamma,A}[\gamma], \{e_x\}) : \mathrm{Inv}(\mathrm{coh}_{\Gamma,A}[\gamma])}$$

其中 $\mathrm{Var}_n(\Gamma)$ 表示维数为 n 的 Γ 的变量集合。此外，我们要求以下计算规则，用以确定通过该规则获得的可逆性结构与通过该定理获得的结构相同：

$$\mathrm{linv}(\mathrm{can}(t, \{e_x\})) \equiv t^{\mathrm{linv}} \qquad \mathrm{rinv}(\mathrm{can}(t, \{e_x\})) \equiv t^{\mathrm{rinv}}$$
$$\mathrm{lunit}(\mathrm{can}(t, \{e_x\})) \equiv t^{\mathrm{lunit}} \qquad \mathrm{runit}(\mathrm{can}(t, \{e_x\})) \equiv t^{\mathrm{runit}}$$
$$\mathrm{ilunit}(\mathrm{can}(t, \{e_x\})) \equiv \mathrm{can}(t^{\mathrm{lunit}}, \{-\})$$
$$\mathrm{irunit}(\mathrm{can}(t, \{e_x\})) \equiv \mathrm{can}(t^{\mathrm{runit}}, \{-\})$$

最后两条规则对应于这样一个事实：左、右单位通过余归纳地方式配备了该规则前提所要求的结构。更详细的说明见附录 B。

4.4 游走等价

在陈述 ICaTT 的最后一条规则之前，我们引入游走等价，即一个对可逆性结构进行分类的上下文，以及一个项和一个类型的分类代换。

定义 *4.1.* 该游走等价对于 $n \in \mathbb{N}$ 是如下上下文：

$$\mathcal{E}^{n+1} = \mathbb{D}^{n+1} \triangleright (\mathsf{e}_{n+1} : \mathrm{Inv}(\mathsf{d}_{n+1}))$$

我们将用 $\mathrm{E}^{n+1} \vdash \mu_{n+1} : \mathbb{D}^{n+1}$ 表示其弱化代换。

引理 2.6 的一个类似结果对 ICaTT 成立，即类型由球和圆盘分类，而项由圆盘和游走等价分类：我们可以通过对语法的归纳法，为每个类型 A 指定一个代换 χ_A，方式如下：

$$\chi_\star = \diamond \qquad \chi_{\mathrm{Inv}_A(t)} = \chi_A \triangleright (\mathsf{d}_{n+1} \mapsto t)$$
$$\chi_{u \to_A v} = \chi_A \triangleright (\mathsf{d}_n^- \mapsto u) \triangleright (\mathsf{d}_n^+ \mapsto v)$$

其中，$n = \dim A + 1$ 和 $\mathsf{d}_n^\pm$ 是 $\mathbb{S}^n$ 的最后两个变量。我们还可以为每一对项 t 和类型 A 指定一个代换 $\chi_{t,A}$，令

$$\chi_{t,A} = \begin{cases} \chi_A \triangleright (\mathsf{d}_{n+1} \mapsto t) & \text{when } A = \star \text{ or } A = u \to v \\ \chi_A \triangleright (\mathsf{e}_{n+1} \mapsto t) & \text{when } A = \mathrm{Inv}(u) \end{cases} \tag{4}$$

其中同样有 $n = \dim A + 1$。我们将看到，当 $\Gamma \vdash A$ 是一个类型时，要么 $\Gamma \vdash \chi_A : \mathbb{S}^n$，要么 $\Gamma \vdash \chi_A : \mathbb{D}^{n+1}$，这取决于它是否为 Inv 类型。类似地，对于 $\Gamma \vdash t : A$，我们将有 $\Gamma \vdash \chi_{t,A} : \mathbb{D}^n$ 或 $\Gamma \vdash \chi_{t,A} : \mathrm{E}^{n+1}$，在这种情况下我们将记为 $\chi_t = \chi_{t,A}$。这些代换的重要性来自引理 4.4

4.5 可逆性结构的递归子

至此，我们完成了 ICaTT 规则的介绍，最后介绍项构造子 rec 的规则，它允许递归地构造类型为 Inv 的项。为了描述依赖类型论中的递归，传统上需要函数类型的存在。然而，为 CaTT 配备函数类型并非易事，因为其模型范畴并非笛卡尔闭——有向同伦类型论中也存在同样的障碍，并且人们已提出若干种规避方法 [27, 32, 33, 36]。因此，在我们的理论中，我们通过扩展上下文并加入归纳假设来建模递归调用。

更准确地说，给定一个项 $\mathrm{E}^{n+1} \vdash t : u \to v$，我们定义扩展后的上下文

$$\mathrm{IH}_t^{n+1} = \mathcal{E}^{n+1} \triangleright (h^- : \mathrm{Inv}(\Sigma t)[\chi_{\mathrm{ilunit}(\mathsf{e}_{n+1})}]) \triangleright (h^+ : \mathrm{Inv}(\Sigma t)[\chi_{\mathrm{irunit}(\mathsf{e}_{n+1})}])$$

其中包含两个新变量，用于提供归纳假设。rec 的引入规则与 inv 的规则类似，区别在于最后两个前提中的上下文扩展了归纳假设：

$$\text{rec-intro}\ \frac{\begin{array}{c}\mathcal{E}^{n+1} \vdash t : x \to y \qquad \Gamma \vdash \gamma : \mathcal{E}^{n+1} \\ \mathcal{E}^{n+1} \vdash t_l : y \to x \qquad \mathcal{E}^{n+1} \vdash t_r : y \to x \\ \mathcal{E}^{n+1} \vdash t_{lu} : t_l * t \to \mathrm{id}\, y \qquad \mathcal{E}^{n+1} \vdash t_{ru} : t * t_r \to \mathrm{id}\, x \\ \mathrm{IH}_t^{n+1} \vdash t_{ilu} : \mathrm{Inv}(t_{lu}) \qquad \mathrm{IH}_t^{n+1} \vdash t_{iru} : \mathrm{Inv}(t_{ru})\end{array}}{\Gamma \vdash \mathrm{rec}(t, t_l, t_r, t_{lu}, t_{ru}, t_{ilu}, t_{iru}, \gamma) : \mathrm{Inv}(t[\gamma])}$$

此外，我们还要求为这个构造子规定计算规则，其中将上下文 IH_t^{n+1} 中额外的变量代换为实际的递归调用。为此，给定一个项 $\mathrm{E}^{n+1} \vdash r : \mathrm{Inv}(t)$，我们定义一个代换 $\mathrm{E}^{n+1} \vdash \mathrm{inst}_r : \mathrm{IH}_t^{n+1}$，它将 IH_t^{n+1} 中的变量实例化为如下所示的递归调用：

$$\mathrm{inst}_r = \mathrm{id}_{\mathcal{E}^{n+1}} \triangleright (h^- \mapsto \Sigma(r)[\chi_{\mathrm{ilunit}(\mathsf{e}_{n+1})}]) \triangleright (h^+ \mapsto \Sigma(r)[\chi_{\mathrm{irunit}(\mathsf{e}_{n+1})}])$$

完整的 β-规则见图 4h。产生箭头的析构子情形与 inv 的 β-规则类似。其余针对 $e = \mathrm{rec}(t, t_l, t_r, t_{lu}, t_{ru}, t_{ilu}, t_{iru}, \gamma)$ 的规则由下式给出

$$\mathrm{ilunit}\, e = t_{ilu}[\mathrm{inst}_e \circ \gamma] \qquad \mathrm{irunit}\, e = t_{iru}[\mathrm{inst}_e \circ \gamma]$$

至此，理论 ICaTT 的定义告一段落。该理论的完整介绍可见 C 节。

4.6 元理论考量

我们以记录该定义的一些简单推论，并讨论理论 ICaTT 的规范化，作为本节的收尾。

引理 4.2。 代换保持 *ICaTT* 中的类型判断，即以下规则是可容许的：

$$\frac{\Delta \vdash \gamma : \Gamma \quad \Gamma \vdash A}{\Delta \vdash A[\gamma]} \qquad \frac{\Delta \vdash \gamma : \Gamma \quad \Gamma \vdash t : A}{\Delta \vdash t[\gamma] : A[\gamma]} \qquad \frac{\Delta \vdash \gamma : \Gamma \quad \Gamma \vdash \gamma' : \Gamma'}{\Delta \vdash \gamma' \circ \gamma : \Gamma'}$$

引理 4.3。 悬挂保持 *ICaTT* 中的类型判断，即以下规则是可容许的：

$$\frac{\Gamma \vdash}{\Sigma\Gamma \vdash} \qquad \frac{\Gamma \vdash A}{\Sigma\Gamma \vdash \Sigma A} \qquad \frac{\Gamma \vdash t : A}{\Sigma\Gamma \vdash \Sigma t : \Sigma A} \qquad \frac{\Gamma \vdash \gamma : \Delta}{\Sigma\Gamma \vdash \Sigma\gamma : \Sigma\Delta}$$

证明。两者均通过对语法作归纳法证明，因为所有给定规则（包括计算规则）在代换和悬挂下都保持稳定。□

引理 4.4。 *ICaTT* 的类型 $\Gamma \vdash A$ 与代换 $\Gamma \vdash \chi_A : \mathbb{S}^n$ 或 $\Gamma \vdash \chi_A : \mathbb{D}^{n+1}$ 之间存在自然双射（模掉判断相等）。项 $\Gamma \vdash t : A$ 与代换 $\Gamma \vdash \chi_t : \mathbb{D}^n$ 或 $\Gamma \vdash \chi_t : \mathcal{E}^{n+1}$之间也存在自然双射。在这些双射下，分别与 ι_n 或 μ_{n+1} 复合会将项映射到其类型。

证明。该引理的证明类似于引理 2.6 的证明，只需 ICaTT 的每条规则在代换下保持稳定。注意，形如 $\Gamma \vdash \chi_A : \mathbb{S}^n$ 的代换对应于 A 形如 $\star$ 或 $u \to v$ 的情形，而形如 $\Gamma \vdash \chi_A : \mathbb{D}^{n+1}$ 的代换对应于 $\mathsf{Inv}(t)$ 的情形。□

与 CaTT 不同，ICaTT 理论具有非平凡的计算规则。为了使 ICaTT 的重写系统具有局部合流性，我们将方向定为 β-约化和 η-扩展。这是由以下示例所迫：考虑一个项 t：

$$\Delta \vdash \mathsf{coh}_{\Gamma,A}[\gamma] : A[\gamma]$$

并假设它的维数严格大于上下文Γ的维数，因此它具有以下可逆性结构：

$$\Delta \vdash \mathsf{can}(t, \{\}) : \mathsf{Inv}(t)$$

将此可逆性结构记作 e，我们注意到，从该项出发可以进行以下转换：

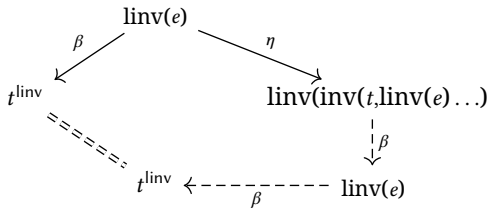
$$\begin{array}{c} e \\ \Vert \eta \\ \mathsf{inv}(t,\ \mathsf{linv}(e),\ \mathsf{rinv}(e),\ \mathsf{lunit}(e),\ \mathsf{runit}(e),\ \mathsf{ilunit}(e),\ \mathsf{irunit}(e)) \\ \Vert \beta \\ \mathsf{inv}(t, t^{\mathsf{linv}}, t^{\mathsf{rinv}}, t^{\mathsf{lunit}}, t^{\mathsf{runit}}, \mathsf{can}(t^{\mathsf{lunit}}, \{\}), \mathsf{can}(t^{\mathsf{runit}}, \{\})) \end{array}$$

其中 inv 的各个分量如第 4.3 节所述。若将 η 定向为相反方向，则会构成局部合流性的反例。

命题 4.5。 β-约化 η-展开重写系统是局部合流的，但不是终止的

证明。首先，我们通过考察与上文相同的例子，证明该系统不是终止的。上文 inv 项的最后两个分量本身也具有相同的形式，因此可以再次展开。这个过程不会终止。

通过考察所有临界对，可以证明局部合流性。我们在此给出一个临界对的示例：考虑一个可逆结构 $e = \mathsf{can}(t,\ \{e_x\})$，并省略项中与论证无关的部分。



其他临界对与此类似。 □

非终止性并不令人意外。事实上，构造子 rec 和 can 的存在理由，就是在语法上捕捉 inv 构造子的无限次应用。我们认为这对我们的理论并无妨碍，因为 ICaTT 的规则不要求检查类型为 Inv 的项之间的相等性，而且该理论对其他项是可规范化的。我们将形式为$\star$ 或 $u \to v$ 范畴类型 的类型及其 inhabitants 称为范畴项。

命题 4.6. 由 β-约化和η-展开定义的重写系统对范畴类型和项是可规范化的，也就是说，存在一个过程 nf，能够从每个 $\beta\eta$-等价类中选出一个指定代表。

证明。我们通过定义一种正规化策略来证明这一点。为此，我们引入一个中间重写系统，在其中按维数限制 η-展开。我们将 n维 η-展开称为这样一种重写系统：我们只对维数不超过 n 的可逆性结构进行 η-展开。我们还限制 η-展开，使其不在任何析构子之下应用，正如 Jay 和 Ghani [24] 所做的那样。受限系统是终止的，并且按照与命题 4.5 相同的证明，它仍然是局部合流的。

两个 n维范畴项等价于 $\beta\eta$，当且仅当它们具有相同的 n维范式。事实上，在 β-约化形式中不会出现维数高于 n 的可逆性结构，因此，对维数大于 n 的可逆性结构进行 η-展开无关紧要。因此，我们的策略是计算 n维范畴项在 n维受限系统下的范式。□

在本文余下部分中，需要注意的是，范式中的范畴类型和项可能不会使用构造子 can、rec 或 inv，因为它们会被规范化过程消去。因此，此类项中可能出现的可逆性结构是变量，或对这些变量反复应用析构子 ilunit 和 irunit。

5 使用可逆性的内部证明

我们以证明助手的形式开发了类型理论 ICaTT 的原型实现，使人们能够对

ω-范畴及其可逆性结构进行推理。本节中，我们在证明助手中形式化了文献中的若干近期结果。我们的主要创新并不在于所证明的结果，而在于 ICaTT 理论中的形式化方法，它免去了复杂的元理论推理。

5.1 实现与附加功能

我们的实现建立在现有 CaTT 实现之上，使我们能够利用并扩展该实现已有的若干功能。特别是，我们为 ICaTT 实现了合一算法，以解析隐式参数。该合一算法遵循尽力而为原则，并不完备，但足以显著减少实际编写项时所需的长度。此外，如果合一失败，也始终可以手动编写满项。实际使用中，实现会自动推断哪些参数可以显式给出；对于通常需要显式给出的参数，若在某个具体情形下可以推断出来，也可以使用通配符_。现有的 CaTT 实现有两个用于声明名称的关键字

```
coh [name] [ctx] : [ty] let [name] [ctx] : [ty] = [tm]
```

关键字 coh 用于声明“相干”，即一个粘贴图与其中一个满类型构成的一对。在我们的具体语法中，这些并不是项，因为它们需要经过代换才能成为项；它们实际上是与某种特定切割形式相对应的定义模式。关键字 let 用于声明任意项，这些项可以是变量、相干，或经代换作用的项。在我们的 ICaTT 实现中，我们加入了类型构造子 Inv，以及两个新关键字

```
inv [name] [ctx]={[tm],[tm],[tm],[tm],[tm],[tm],[tm]}
rec [name] [ctx]={[tm],[tm],[tm],[tm],[tm],[tm],[tm]}
```

关键字 inv 用于声明由构造子 inv 构造的项，而关键字 rec 用于声明递归定义：对于后者，上下文必须具有 E^{n+1} 的形式；我们提供关键字 IHleft 和 IHright，以便在递归定义的最后两个参数中访问归纳假设。关键字 rec 类似于 coh，因为它并不直接对应于使用同名构造子构造的项，而是对应于切割 cut 的一次应用；要生成一个项，还需要进行代换。最后，我们还提供项构造子 can，它接受一个项和一个可逆性结构列表作为参数；该列表由逗号分隔的项组成，并以花括号括起。

CaTT 和 ICaTT 的悬垂运算是隐式的，这意味着我们编写的每个定义都被视为由悬挂生成的一族构造。我们还使用内置关键字 id，它表示恒等相干；以及 comp，它计算沿余维1面的复合。关键字 comp 的特殊之处在于它是多元的：它可以接受任意数量的参数，并生成这些参数的无偏复合。

5.2 可逆性结构的形式化结果

本节旨在介绍我们在 ICaTT 中形式化证明的一些结果。所有这些结果都见于补充材料中的文件 proofs/invertibility.catt[2]，，为求完整，我们也在 D 节中再次列出这些结果。为了便于展示，我们将命题陈述置于命题环境中，并将构造过程的部分内容作为证明给出。我们证明的第一个结果是定理 3.8 的一个特例，用以展示如何运用我们的理论。

命题 5.1. 两个可逆胞腔的复合仍然可逆。

证明。直接应用 can-intro 规则即可。注意，我们在维数 1 中陈述此结果，但可以通过悬挂将其推导到任意维数。

```
let compinv (x : *) (y : *) (z : *)
            (f : x ->y) (g : y ->z)
            (e : Inv (f)) (e' : Inv (g))
: Inv (comp f g)   =can ( comp f g { e , e' })
```

□

命题 5.2。 任何可逆性结构的左逆和右逆都是可逆的。

证明。我们只给出左逆的构造，右逆的构造是对称的，可参见 D 节或补充材料中的文件 proofs/invertibility.catt。我们首先定义一个联系左逆和右逆的胞腔。

```
let lri (x : *) (y : *) (f : x -> y) (e : Inv(f))
: linv (e) -> rinv (e)
```

形式化定义请参见 D 节，这里我们在图 2 中对其进行说明。然后我们构造见证，证明只要 ilunit(e) 的左逆是可逆的，上述胞腔也是可逆的：

```
let lriU-aux (x : *) (y : *) (f : x -> y)
           (e : Inv(f)) (e' : Inv (linv (irunit (e))))
: Inv (lri e)
```

左逆上的可逆性结构随后通过如下递归定义得到：

```
rec linv-inv (x : *) (y : *) (f : x ->y) (e : Inv(f))= { linv(e), f ,
f , comp (whiskl f (lri e)) (runit (e)), lunit (e) , can (_
{can (_ { lriU-aux IHright }) , (irunit (e))}), ilunit (e) }
```

□

命题 5.3. 若 t 可逆，则其左逆与右逆之间存在一个可逆胞。

命题 5.4. 给定一个可逆胞 $f\colon u \to v$，若 u 可逆，则 v也可逆。

命题 5.5. 可逆胞满足 *2/6* 性质：给定三个可复合的胞腔 f、g和 h，若 $f*g$和 $g*h$可逆，则 f、 g和 h也可逆。

各命题对应的明确构造，请参阅补充材料中的文件 proofs/invertibility.catt 或 D 节。

[2]可在 https://zenodo.org/records/18343317 获取

$$\mathrm{linv}(e) \to \mathrm{linv}(e) * \mathrm{id}\ \mathrm{linv}(e) * (f * \mathrm{rinv}(e))\ (\mathrm{linv}(e) * f) * \mathrm{rinv}(e)\ \mathrm{id} \to \mathrm{rinv}(e)\ \mathrm{rinv}(e)$$

图 2：将左逆与右逆联系起来

6 在 ω-范畴中的应用

由于 ICaTT 是类型论 CaTT 的扩展，因此存在一个族态射 $\mathsf{J}\colon \mathsf{S}_{\mathrm{CaTT}} \to \mathsf{S}_{\mathrm{lCaTT}}$，从而导出一个伴随

$$\mathsf{J}^*\colon \mathrm{Mod}(\mathrm{lCaTT}) \rightleftarrows \omega\mathbf{Cat}\colon \mathsf{J}_!$$

其中 J^* 由与 J 的预复合给出，如 Frey [16, 第 4] 节所示。为避免记号冲突，我们将改用以下方式表示 ICaTT 的可表模型：

$$(\!|-|\!)\colon \mathcal{S}^{\mathrm{op}}_{\mathrm{lCaTT}} \to \mathrm{Mod}(\mathrm{lCaTT})$$
$$(\!|\Gamma|\!)(\Delta) = \mathcal{S}_{\mathrm{lCaTT}}(\Gamma, \Delta)$$

我们将证明 J 是全忠实的，特别地，ICaTT 是 CaTT 的保守扩张。此外，我们将证明 $\mathsf{J}^*(\!|\mathbb{E}^1|\!)$ 是 Ozornova 与 Rovelli [34], 的游走等价，Fujii 等人 [17] 用它来分类等纤维化。

6.1 保守性

为了证明 ICaTT 对 CaTT 是保守的，我们将使用第 4.6 节所述的正规化策略。

引理 6.1。 给定 *CaTT* 的上下文 Γ，以及 *ICaTT* 中的范畴类型 $\Gamma \vdash A$，则 $\Gamma \vdash \mathrm{nf}(A)$ 属于 *CaTT*。此外，对于 *ICaTT* 中的任意范畴项 $\Gamma \vdash t\colon A$，则 $\Gamma \vdash \mathrm{nf}(t)\colon \mathrm{nf}(A)$ 属于 *CaTT*。

证明。我们回顾一下，$\mathrm{nf}(A)$ 和 $\mathrm{nf}(t)$ 中可能出现的可逆性结构只有变量，或对这些变量反复应用析构子 ilunit 和 irunit 得到的项。由于 Γ 是 CaTT 的上下文，其中不包含作为可逆性结构的变量。因此，$\mathrm{nf}(A)$ 和 $\mathrm{nf}(t)$ 中都不会包含可逆性结构，故它们是 CaTT 中有效的类型和项。□

定理 6.2。 理论 *ICaTT* 对 *CaTT* 是保守的。

证明。考虑 CaTT 中的一个上下文 Γ、CaTT 中的一个类型 $\Gamma \vdash A$，以及 ICaTT 中的一个项 $\Gamma \vdash u\colon A$。由引理 6.1，这给出了 CaTT 中 $\Gamma \vdash \mathrm{nf}(u) : \mathrm{nf}(A)$ 的一个推导。由于 A 是一个 CaTT 类型，它处于范式，$\mathrm{nf}(A) = A$，从而表明 A 在 CaTT 中已经有居留元。□

Proposition 6.3. *The functor* $\mathsf{J}\colon \mathcal{S}_{CaTT} \to \mathcal{S}_{lCaTT}$ *is fully faithful.*

证明。我们考虑 CaTT 中的两个上下文 Γ 和 Δ，并通过对 Δ 的归纳法证明函数 $\mathsf{J}\colon \mathsf{S}_{\mathrm{CaTT}}(\Gamma, \Delta) \to \mathsf{S}_{\mathrm{lCaTT}}(\mathsf{J}\,\Gamma, \mathsf{J}\,\Delta)$ 是一个双射。基础情形由 J 在两种理论中都保持终对象这一事实给出。对于归纳情形，我们考虑 CaTT 中的一个类型 $\Delta \vdash A$。我们首先证明单射性：考虑两个代换，它们在 J 下的像相等：

$$\Gamma \vdash \gamma \triangleright (x \mapsto u)\colon \Delta \triangleright (X\colon A)$$
$$\Gamma \vdash \gamma' \triangleright (x \mapsto u')\colon \Delta \triangleright (X\colon A)$$

于是由归纳法 $\gamma = \gamma'$，且 $u = \mathrm{nf}(u) = \mathrm{nf}(u') = u'$，从而证明了单射性。接下来，我们假设存在一个代换 $\mathsf{J}\ \Gamma \vdash \gamma \triangleright (x \mapsto u) : \mathsf{J}(\Delta \triangleright (x\colon A))$，以证明满射性。由归纳法可得一个代换 $\Gamma \vdash \bar{\gamma}\colon \Delta$，使得 $\mathsf{J}\ \bar{\gamma} = \gamma$；根据引理 6.1，我们得到一个项 $\Gamma \vdash \mathrm{nf}(u) : \mathrm{nf}((\mathsf{J}\ A)[\gamma])$。接着注意到，$\mathsf{J}(A[\bar{\gamma}])$ 是一个处于范式且等于 $(\mathsf{J}\ A)[\gamma]$ 的类型，因此 $\mathrm{nf}(u)$ 的类型为 $A[\bar{\gamma}]$，由此可以构造代换

$$\Gamma \vdash \bar{\gamma} \triangleright (x \mapsto \mathrm{nf}(u))\colon \Delta \triangleright (x\colon A)$$

从而给出 $\gamma \triangleright (x \mapsto u)$的一个原像。 □

推论 6.4. 给定一个 *CaTT* 上下文 Γ，我们有 $\mathsf{J}^*(\!|\Gamma|\!) = [\![\Gamma]\!]$

6.2 游走等价

我们证明，$\mathsf{J}^*(\!|\mathbb{E}^1|\!)$ 是 Ozornova 与 Rovelli [34]的游走等价 $\mathbb{E}^1_{\mathrm{OR}}$。我们首先借助 Benjamin 等人 [7]所建立的 CaTT 上下文与有限 computad 之间的等价，回顾后者的定义。为此，我们通过归纳法定义一列上下文 $\mathbb{E}^{1,n}$，并连同针对 $n \geq 1$的三个代换

$$\mathcal{E}^{1,n} \vdash i^n\colon \mathcal{E}^{1,n-1} \qquad \mathcal{E}^{1,n} \vdash f^n\colon \Sigma\mathcal{E}^{1,n-1} \qquad \mathcal{E}^{1,n} \vdash g^n\colon \Sigma\mathcal{E}^{1,n-1}$$

其中 i^n 是一个展示映射。在基础情形 $n = 0$中，我们定义：

$$\mathcal{E}^{1,0} = \mathbb{S}^0 = (x\colon \star) \triangleright (y\colon \star)$$

然后，对于 $n = 1$，我们定义

$$\begin{aligned}
\mathcal{E}^{1,1} &= \mathcal{E}^{1,0} \triangleright (u\colon x \to y) \triangleright (v\colon y \to x) \triangleright (w\colon y \to x)\\
i^1 &= (x \mapsto x) \triangleright (y \mapsto y)\\
f^1 &= (v^- \mapsto x) \triangleright (v^+ \mapsto x) \triangleright (x \mapsto v * u) \triangleright (y \mapsto \mathrm{id}\, y)\\
g^1 &= (v^- \mapsto y) \triangleright (v^+ \mapsto y) \triangleright (x \mapsto u * w) \triangleright (y \mapsto \mathrm{id}\, x)
\end{aligned}$$

其中 $v^{\pm}$ 是悬挂所引入的新变量。代换 i^1 是一个展示映射，因为它是三个弱化代换的复合。然后我们将上下文 $\mathbb{E}^{1,n+1}$ 连同这三个代换定义为 $\mathsf{S}_{\mathrm{lCaTT}}$ 中的如下极限，由图中的实线部分标出：

$$\begin{array}{ccccc}
\Sigma\mathcal{E}^{1,n} & \xleftarrow{f^{n+1}} & \mathcal{E}^{1,n+1} & \xrightarrow{g^{n+1}} & \Sigma\mathcal{E}^{1,n}\\
{\scriptstyle \Sigma i^n}\downarrow & & \downarrow{\scriptstyle i^{n+1}} & & \downarrow{\scriptstyle \Sigma i^n}\\
\Sigma\mathcal{E}^{1,n-1} & \xleftarrow[f^n]{} & \mathcal{E}^{1,n} & \xrightarrow[g^n]{} & \Sigma\mathcal{E}^{1,n-1}
\end{array} \tag{5}$$

由于 Σ 保持展示映射，Σi^n 是一个展示映射，并且该极限 can 计算为沿 Σi^n的两个拉回组成的序列，从而表明存在性。这也将 i^{n+1} 实现为两个展示映射的复合，表明它同样是一个展示映射。直观而言，上下文 $\mathbb{E}^{1,n}$ 是游走等价的 n-截断。上下文 $\mathbb{E}^{1,2}$ can 被显式计算为：

$$\begin{aligned}
\mathcal{E}^{1,2} = \mathcal{E}^{1,1} \triangleright (u_v\colon v * u \to \mathrm{id}\, y) \triangleright\\
(v_v\colon \mathrm{id}\, y \to v * u) \triangleright (w_v\colon \mathrm{id}\, y \to v * u) \triangleright\\
(u_w\colon u * w \to \mathrm{id}\, x) \triangleright\\
(v_w\colon \mathrm{id}\, x \to u * w) \triangleright (w_w\colon \mathrm{id}\, x \to u * w)
\end{aligned}$$

我们注意到这些上下文的大小呈指数增长。于是游走等价 $\mathsf{E}^1_{\mathrm{OR}}$ 是由这些上下文所呈现的 ω-范畴的余极限：

$$\mathcal{E}^1_{\mathrm{OR}} = \operatorname{colim}\left([\![\mathcal{E}^{1,0}]\!] \xrightarrow{[\![i^1]\!]} [\![\mathcal{E}^{1,1}]\!] \xrightarrow{[\![i^2]\!]} [\![\mathcal{E}^{1,2}]\!] \xrightarrow{[\![i^3]\!]} \dots \right)$$

这个余极限是一个计算图，但由于它有无穷多个生成元，因此无法将其描述为 CaTT 的上下文。尽管如此，它仍可由 ICaTT 的一个上下文表示。

定理 6.5. 存在一个 ω-范畴的同构：

$$\mathsf{J}^*(\!|\mathcal{E}^1|\!) \cong \mathcal{E}^1_{\mathrm{OR}}$$

证明。回顾一下，E^1 是 ICaTT 的如下上下文：

$$\mathcal{E}^1 = (\mathsf{d}_0^- : \star) \triangleright (\mathsf{d}_0^+ : \star) \triangleright (\mathsf{d}_1 : \mathsf{d}_0^- \to \mathsf{d}_0^+) \triangleright (\mathsf{e}_1 : \mathsf{Inv}(\mathsf{d}_1))$$

并且我们在 $\mathsf{S}_{\mathsf{ICaTT}}$ 中定义一个由以下代换组成的锥

$$\mathcal{E}^1 \vdash \gamma^n : \mathcal{E}^{1,n}$$

这些代换满足以下相容条件

$$i^n \circ \gamma^n = \gamma^{n-1}$$
$$f^n \circ \gamma^n = \Sigma\gamma^{n-1} \circ \chi_{\mathsf{ilunit}(\mathsf{e}_1)}$$
$$g^n \circ \gamma^n = \Sigma\gamma^{n-1} \circ \chi_{\mathsf{irunit}(\mathsf{e}_1)}$$

其中，$\mathsf{E}^1 \vdash \chi_{\mathsf{ilunit}(\mathsf{e}_1)} : \Sigma\mathsf{E}^1$ 和 $\mathsf{E}^1 \vdash \chi_{\mathsf{irunit}(\mathsf{e}_1)} : \Sigma\mathsf{E}^1$ 是利用 $\Sigma\mathsf{E}^1 = \mathsf{E}^2$应用引理 4.4得到的。在基例中，我们令：

$$\gamma^0 = (x \mapsto \mathsf{d}_0^-) \triangleright (y \mapsto \mathsf{d}_0^+)$$
$$\gamma^1 = \gamma^0 \triangleright (u \mapsto \mathsf{d}_1) \triangleright (v \mapsto \mathsf{linv}(\mathsf{e}_1)) \triangleright (w \mapsto \mathsf{rinv}(\mathsf{e}_1))$$

根据所涉及代换的定义，显然 γ^1 满足所要求的三个方程。对于归纳情形，我们利用 J 保持沿展示映射的拉回，因此特别保持图表 (5) 的极限。由该极限的泛性质，我们得到唯一的代换 γ^{n+1}，使其适合以下图表：

$$\begin{array}{ccccc}
\Sigma\mathcal{E}^1 & \xleftarrow{\chi_{\mathsf{ilunit}(\mathsf{e}_1)}} & \mathcal{E}^1 & \xrightarrow{\chi_{\mathsf{irunit}(\mathsf{e}_1)}} & \Sigma\mathcal{E}^1 \\
\downarrow{\scriptstyle \Sigma\gamma^n} & & \downarrow{\scriptstyle \gamma^{n+1}} & & \downarrow{\scriptstyle \Sigma\gamma^n} \\
\Sigma\mathcal{E}^{1,n} & \xleftarrow{f^{n+1}} & \mathcal{E}^{1,n+1} & \xrightarrow{g^{n+1}} & \Sigma\mathcal{E}^{1,n} \\
\downarrow{\scriptstyle \Sigma i^n} & & \downarrow{\scriptstyle i^{n+1}} & & \downarrow{\scriptstyle \Sigma i^n} \\
\Sigma\mathcal{E}^{1,n-1} & \xleftarrow[f^n]{} & \mathcal{E}^{1,n} & \xrightarrow[g^n]{} & \Sigma\mathcal{E}^{1,n-1}
\end{array}$$

断言。 代换 γ^n 在 $\mathcal{E}^{1,n}$ 的变量与维数至多为 n 的 $\mathcal{E}^1$ 的中性范畴项之间诱导出一个双射。

断言的证明。我们对 n进行归纳。维数为 $n = 0$ 的唯一中性项是变量 $\mathsf{d}_0^{\pm}$，由此可得基情形。维数为 $n = 1$ 的范畴中性项恰好是变量 d_1 及其逆元 $\mathsf{linv}(\mathsf{e}_1)$ 和 $\mathsf{rinv}(\mathsf{e}_1)$，从而证明了 $n = 1$的断言。于是设 $n \geq 2$。

通过归纳法，我们可以证明维数 n中恰好存在 2^{n-1} 个 Inv 类型的中性项，因为它们的形式为 $\mathsf{ilunit}(e)$ 和 $\mathsf{irunit}(e)$，其中 e 是低一维的中性可逆性结构。因此，维数 n 中有 $3 \cdot 2^{n-1}$ 个范畴中性项：它们要么由维数 n 的中性项经 $\mathsf{linv}(e)$ 和 $\mathsf{rinv}(e)$ 得到，要么由维数 $(n-1)$ 的中性项经 $\mathsf{lunit}(e')$ 和 $\mathsf{runit}(e')$ 得到。

$\mathsf{E}^{1,n}$ 的最高维变量是这些 E Σ^{n-1} 变量的两个副本的不交并。根据归纳法可知，这样的变量恰好有 $3 \cdot 2^{n-1}$ 个。由归纳假设以及悬挂对中性项的保留性质，通过代换 $\chi_{\mathsf{ilunit}(\mathsf{e}_1)} \circ \Sigma\gamma^{n-1}$ 和 $\chi_{\mathsf{irunit}(\mathsf{e}_1)} \circ \Sigma\gamma^{n-1}$，它们分别单射地映射到互不相交的中性项集合中。因此，γ^n 将变量单射地映射到中性项中，因而也是双射。□

将函子 $\mathsf{J}^*(\!|-|\!)$ 应用于代换族 $\{\gamma^n\}$，我们得到一个余锥，并由此从余极限得到一个泛态射：

$$\gamma : \mathcal{E}^1_{\mathrm{OR}} \to \mathsf{J}^*(\!|\mathcal{E}^1|\!)$$

还需证明 γ 是模型的自然同构。由于模型的滤余极限是逐点计算的 [16]，只需证明对于每个上下文 Γ，下面的映射是双射：

$$\gamma_\Gamma : \operatorname*{colim}_n([\![\mathcal{E}^{1,n}]\!](\Gamma)) \to \mathsf{J}^*(\!|\mathcal{E}^1|\!)(\Gamma)$$

由于 i^n 是一个仅添加维数为 n的变量的展示映射，因此它在维数不超过$(n-1)$的项上是双射。由此，通过结构归纳法，它在映至该维数上下文的代换上也为双射。于是问题归结为证明：

$$(\!|\gamma^n|\!)(\Gamma) : \mathcal{S}_{\mathsf{ICaTT}}(\mathcal{E}^{1,n}, \Gamma) \to \mathcal{S}_{\mathsf{ICaTT}}(\mathcal{E}^1, \Gamma)$$

对于每个维数不超过 n的 CaTT 上下文 Γ，该映射是双射。我们将对 $(n, \dim \Gamma)$作强归纳。固定 n后，$\dim \Gamma < n$ 的情形由下式得出：

$$(\!|i^n|\!)(\Gamma) : \mathcal{S}_{\mathsf{ICaTT}}(\mathcal{E}^{1,n}, \Gamma) \to \mathcal{S}_{\mathsf{ICaTT}}(\mathcal{E}^{1,n-1}, \Gamma)$$

该映射是双射，并且对 $n-1$使用归纳假设。现在假设 $\dim \Gamma = n$，并进行结构归纳。

首先假设 $\Gamma = \mathbb{D}^n$，并令 $\mathsf{E}^1 \vdash \chi_t : \Gamma$ 为一个代换。如果 t 的范式是一个中性项，那么根据断言，$\mathsf{E}^{1,n}$ 中存在唯一一个映至它的变量。如果 t 的范式是 $\mathsf{coh}_{\Delta, A}[\delta]$，那么根据结构归纳法，存在唯一的映射 δ' 使得 $\delta = \delta' \circ \gamma^n$。因此，$\mathsf{coh}_{\Delta, A}[\delta']$ 是 γ^n下t的唯一原像，因为没有变量会被 γ^n映至相干。

那么，假设 $\Gamma = \Gamma' \triangleright (x : A)$ 是一个任意的维数为 $n \geq 0$ 的上下文，并令 $\mathsf{E}^1 \vdash \sigma \triangleright (x \mapsto t) : \Gamma$ 为一个代换。由结构归纳法，我们可以假设存在唯一的 σ' 和t'，使得

$$\sigma = \sigma' \circ \gamma^n \qquad t = t'[\gamma^n]$$

根据从 $\mathbb{S}^k$ $k < n$ $\mathsf{E}^{1,n} \vdash$ 出发的态射的唯一性，我们有 $\mathsf{E}, \vdash \sigma' \triangleright x \mapsto t' : A[\sigma']$。因此，$\mathsf{E}.(\,)[\Gamma]$ 是 $\sigma \triangleright (x \mapsto t)$ 在 $(\!|\gamma^n|\!)(\Gamma)$ 下的唯一原像。□

推论 6.6. 等纤维化恰好是对以下态射族具有右提升性质的态射：

$$\left\{ \mathsf{J}^*(\!|\chi_{\mathsf{src}\,\mathsf{d}_n}|\!) : [\![\mathbb{D}^n]\!] \to \mathsf{J}^*(\!|\mathcal{E}^{n+1}|\!) \mid n \in \mathbb{N} \right\}$$

证明。由定理 6.5 和 Fujii 等人 [17, 定理 4.3.1、5.4.9]的工作可得。□

7 ICaTT 模型与标记 ω-范畴

在第 6 节中，我们研究了可以用 ICaTT 描述的 ω-范畴。值得注意的是，本节的结果并不依赖项构造子 inv、can 或 rec，而只依赖于六个析构子的存在。本节中，我们将证明，上述项构造子正是要在纤维化标记 ω-范畴中构造语义所需的。

7.1 标记 ω-范畴

近期关于高阶范畴的几项研究都考察了标记 ω-范畴。正如 Henry 和 Louba- ton [22]所解释的，标记特定胞腔有助于定义更精细的可逆性概念。这些作者在标记严格 ω-范畴上引入了一族模型结构，其纤维对象是标记ω-范畴；在这些范畴中，带标记的胞腔恰好是那些具有标记逆元的胞腔 [22, 引理 3.30]。在弱情形下，Fujii 等人 [17] 最近使用标记 ω-范畴来刻画弱 ω-范畴的等纤维化。

定义 *7.1*。一个标记 ω-范畴 是一个 ω-范畴（X， tX），带有一组满足以下饱和条件的指定正维胞腔：对于每个来自粘贴图的态射 γ： $[\![\Gamma]\!] \to X$，以及每个维数为n 的满类型 $\Gamma \vdash_{\mathsf{full}} A$，若对每个变量 $x \in \mathrm{Var}_{n+1}(\Gamma)$， $[\![x]\!](\gamma)$都是带标记的，则胞腔 $[\![\mathrm{coh}_{\Gamma,\ A}]\!](\gamma)$也是带标记的。我们将标记 ω-范畴及保持标记的严格 ω-函子的范畴记为 $\omega\mathbf{Cat}^{+}$。

我们注意到，Fujii 等人 [17] 不要求我们的饱和条件，该条件类似于 Henry 和 Loubaton [22]所施加的条件。当然，任意一组胞腔都可以在此饱和条件下闭合：给定一个 ω-范畴 X和一组胞腔 S，我们定义一个递增的集合序列 $S_0 = S$，并且

$$S_{n+1} = S_n \cup \left\{ [\![\mathrm{coh}_{\Gamma,A}]\!](\gamma) \;\middle|\; \begin{array}{l} \Gamma \vdash_{\mathsf{ps}}, \\ \Gamma \vdash_{\mathsf{full}} A \text{ of dimension } n, \\ \gamma : [\![\Gamma]\!] \to X, \\ \forall x \in \mathrm{Var}_{n+1}\,\Gamma,\ f(x) \in S_n \end{array} \right\}$$

我们将 $\mathrm{Sat}(S)$ 定义为该序列的并。由于 ω-范畴的运算是有限元的，这个集合给出了包含 S 的 X 上的最小标记。

命题 7.2。遗忘标记的函子 $\mathsf{U}\colon \omega\mathbf{Cat}^{+} \to \omega\mathbf{Cat}$ 有一个左伴随 $(-)^{\flat}$，使得 $\mathsf{U} \circ (-)^{\flat} = \mathrm{id}$。

证明。对于每个 ω-范畴 X，我们定义 $X^{\flat} = (X, \mathrm{Sat}(\emptyset))$；对于每个 ω-范畴之间的态射 $f\colon X \to Y$，我们定义 $f_{\flat} = f\colon X^{\flat} \to Y^{\flat}$。我们可以通过对饱和的定义进行归纳，并利用 f 的函子性，证明 $f_{\flat}$保标记。此外，根据定义，它满足 $\mathsf{U} \circ (-)^{\flat} = \mathrm{id}$。我们取此伴随的单位为恒等严格 ω-函子 $X \to \mathsf{U}\, X^{\flat}$，余单位 $(\mathsf{U}\, X)^{\flat} \to X$则取为 $\mathsf{U}\, X$ 的恒等态射。由于饱和的最小性，余单位保标记。三角恒等式均成立，因为单位和余单位都是由恒等函子给出的。□

命题 7.3. 标记 ω-范畴构成一个完备且余完备的范畴。

证明。标记 ω-范畴的极限和滤过余极限，可通过分别取其底层 ω-范畴以及标记胞腔集合的极限和滤过余极限来计算。利用 ω-范畴的极限和滤过余极限在函子范畴 $\mathsf{S}_{\mathsf{CaTT}} \to \mathbf{Set}$ 中按点计算这一事实，可以证明这些标记胞腔类是饱和的。一般的余极限通过取底层 ω-范畴的余极限，以及标记胞腔余极限的饱和化来计算。□

Cheng [13] 证明，每个胞腔都有对偶的 ω-范畴必定是一个 ω-群胚。因此，这会迫使任何ω-配边范畴的定义都产生一个 ω-群胚。通过研究带标记的 ω-范畴，可以弥补这一缺陷；它们提供了另一种可逆性概念，Henry 和 Loubaton [22]对此进行了讨论。

定义 *7.4*。胞腔 $f\colon x \to y$ 称为在标记意义下可逆，如果存在逆元：

$$f^{\mathsf{linv}} : y \to x \qquad\qquad f^{\mathsf{rinv}} : y \to x$$

以及一对带标记的消去元：

$$f^{\mathsf{lunit}} : f^{\mathsf{linv}} * f \to \mathrm{id}\, y \qquad f^{\mathsf{runit}} : f * f^{\mathsf{rinv}} \to \mathrm{id}\, x$$

一般而言，带标记的胞腔与在标记意义下可逆的胞腔之间未必存在任何关系。因此，我们将注意力限制在纤维的带标记 ω-范畴上，其定义类比于 Henry 和 Loubaton [22, 定理 3. 31]中的饱和归纳模型结构的纤维对象。要判断这些对象是否是某个模型结构的纤维对象，可能也需要在普通 ω-范畴上构造一个模型结构。

定义 *7.5*。一个标记 ω-范畴在满足以下条件时称为纤维化的：

- 每个标记胞腔都在标记意义下可逆，
- 标记胞腔的所有逆元都是标记的，
- 给定一个标记胞腔 $e\colon x \to y$，若 x是标记的，则 y也如此，
- 对于每个由胞腔 f, g, h组成的可复合三元组，若 $f * g$和 $g * h$是标记的，则 f、 g和 h也都是标记的。

在纤维化的标记 ω-范畴中，每个在标记意义下可逆的胞腔也都是余归纳可逆的。对于 CaTT 上下文，反之亦成立：

命题 7.6. $[\![\Gamma]\!]^{\flat}$ 中的胞腔被标记，当且仅当它在标记意义下可逆，当且仅当它是余归纳可逆的。因此， $[\![\Gamma]\!]^{\flat}$ 是纤维化的。

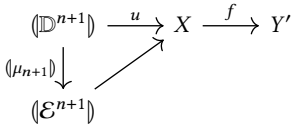
证明。这是命题 3.9 的直接推论。 □

7.2 ICaTT 与带标记 ω-范畴的模型

现在我们给出 ICaTT 在标记 ω-范畴中的语义。它由一个函子 $\mathsf{L}\colon \mathrm{Mod}(\mathsf{ICaTT}) \to \omega\mathbf{Cat}^{+}$ 给出，其值落在纤维化标记 ω-范畴中。我们定义 $\mathsf{L}\ X = (\mathsf{J}^{*} X, t\mathsf{L}\ X)$ 为 X 的底范畴，其中胞腔 u恰当且仅当它属于 $X(\mu_{n+1})\colon X(\mathbb{E}^{n+1}) \to X(\mathbb{D}^{n+1})$ 的像时才是标记的。根据 Yoneda 引理， u 在可分解为以下形式时是标记的：

$$\begin{array}{ccc} (\!|\mathbb{D}^{n+1}|\!) & \xrightarrow{\ u\ } & X \\ {\scriptstyle (\!|\mu_{n+1}|\!)}\downarrow & \nearrow & \\ (\!|\mathcal{E}^{n+1}|\!) & & \end{array}$$

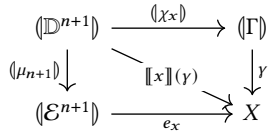
对于模型态射 $f\colon X \to Y$，我们定义 $\mathrm{L}\, f = \mathrm{J}^* f\colon \mathrm{L}\, X \to \mathrm{L}\, Y$。给定 $\mathrm{L}\, X$ 的一个标记胞腔 u，以下交换图表明，u 在 $\mathrm{L}\, f$ 下的像是标记的：



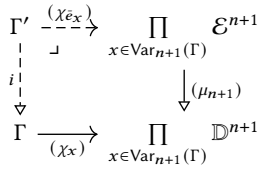
下面的引理表明，这个构造定义了一个函子 $\mathrm{L}\colon \mathsf{Mod}(\mathsf{ICaTT}) \to \omega\mathbf{Cat}^+$。

引理 7.7. 带标记的 ω-范畴 $\mathrm{L}\, X$ 定义良好。

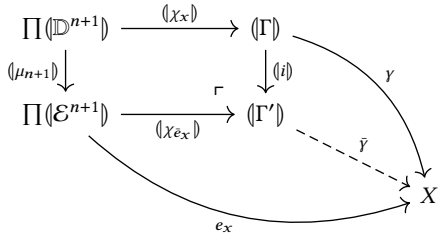
证明。我们需要证明其标记胞腔的集合是饱和的。这由构造子 can 给出：给定一个胞腔 $u = [\![\mathrm{coh}^{\Gamma,A}]\!](\gamma)$，属于 X，且维数为 $n = \dim A$，并且对于每个 $x \in \mathrm{Var}_{n+1}(\Gamma)$，胞腔 $[\![x]\!](\gamma)$ 都是标记的。根据定义，对于每个这样的 x，这就给出了一个交换图



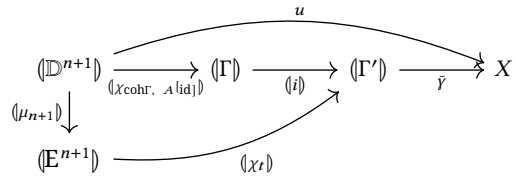
根据定义，μ_{n+1} 是一个展示映射，而展示映射在每个族中都对积稳定，因此我们可以定义如下拉回。它可以具体描述为从 Γ 出发，对每个维数为 $n+1$ 的变量 x，添加一个类型为 $\mathrm{Inv}(x)$ 的变量 $\bar{e}_x$ 后得到的上下文：



注意到 $(\!|-|\!)$ 将沿展示映射的拉回映为推出，上述交换图族恰好表明 γ 可以通过该推出分解：



我们将项 t 定义为 $\Gamma' \vdash \mathrm{can}(\mathrm{coh}^{\Gamma,\ A[i]}\{\bar{e}_x\}) : \mathrm{Inv}(\mathrm{coh}^{\Gamma,\ A[i]})$，借此可以构造以下交换图，表明 u 是带标记的：

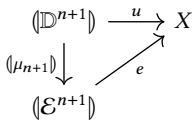


□

现在，我们利用第 5 节的内部证明，说明 L 落在纤维化带标记的 ω-范畴中。由于在纤维化带标记的 ω-范畴中，带标记的胞腔是可逆的，这就证实了 Inv 类型的项可作为可逆性的见证这一断言。

命题 7.8. 标记的 ω-范畴 $\mathrm{L}\, X$ 是纤维化的。

证明。考虑 $\mathrm{L}\, X$ 中的一个标记胞腔 u。存在一个交换三角形



将下列由析构子得到的六个态射与 e 复合，可知胞腔 u 在标记意义下可逆：

$$(\!|\chi_{\mathrm{linv}(\mathrm{e}_{n+1})}|\!) : (\!|\mathbb{D}^{n+1}|\!) \to (\!|\mathcal{E}^{n+1}|\!) \qquad (\!|\chi_{\mathrm{rinv}(\mathrm{e}_{n+1})}|\!) : (\!|\mathbb{D}^{n+1}|\!) \to (\!|\mathcal{E}^{n+1}|\!)$$
$$(\!|\chi_{\mathrm{lunit}(\mathrm{e}_{n+1})}|\!) : (\!|\mathbb{D}^{n+2}|\!) \to (\!|\mathcal{E}^{n+1}|\!) \qquad (\!|\chi_{\mathrm{runit}(\mathrm{e}_{n+1})}|\!) : (\!|\mathbb{D}^{n+2}|\!) \to (\!|\mathcal{E}^{n+1}|\!)$$
$$(\!|\chi_{\mathrm{ilunit}(\mathrm{e}_{n+1})}|\!) : (\!|\mathcal{E}^{n+2}|\!) \to (\!|\mathcal{E}^{n+1}|\!) \qquad (\!|\chi_{\mathrm{irunit}(\mathrm{e}_{n+1})}|\!) : (\!|\mathcal{E}^{n+2}|\!) \to (\!|\mathcal{E}^{n+1}|\!)$$

若将命题 5.2 中构造的项记为：

$$\mathcal{E}^1 \vdash l\colon \mathrm{Inv}(\mathrm{linv}(\mathrm{e}_1)) \qquad \mathcal{E}^1 \vdash r\colon \mathrm{Inv}(\mathrm{rinv}(\mathrm{e}_1))$$

于是，对以下两个态射进行预复合可知，u 的左右逆均为标记的：

$$(\!|\chi_{\Sigma^n l}|\!) : (\!|\mathcal{E}^{n+1}|\!) \to (\!|\mathcal{E}^{n+1}|\!) \qquad (\!|\chi_{\Sigma^n r}|\!) : (\!|\mathcal{E}^{n+1}|\!) \to (\!|\mathcal{E}^{n+1}|\!)$$

其余两个纤维性条件也可类似地通过预复合得到：使用将命题 5.4 和 5.5 中构造的项的悬挂分类的态射。□

Corollary 7.9. *For a context Γ of CaTT, we have $\mathrm{L}(\!|\Gamma|\!) = [\![\Gamma]\!]^\flat$*

证明。推论 6.4 表明，$\mathrm{L}(\!|\Gamma|\!)$ 和 $[\![\Gamma]\!]^\flat$ 的底层 ω-范畴相同。由纤维性可知，$\mathrm{L}(\!|\Gamma|\!)$ 中每个标记胞腔都是余归纳可逆的，因此根据命题 7.6，它在 $[\![\Gamma]\!]^\flat$ 中也是标记的。由于 $(-)^\flat$ 是最小标记的，两种标记必定相同。□

在其对偶定理中，Frey [16, 定理 3.3(ii)] 将到余完备范畴 C 的余连续函子 $\mathsf{Mod}(\mathbb{T}) \to C$ 与保持始对象以及沿展示映射作出的推出的态射 $\mathbb{S}^{\mathrm{op}}_{\mathbb{T}} \to C$ 相对应。这使我们猜想：

猜想 7.10. 函子 L 存在一个右伴随，其定义如下：

$$\mathrm{R}\colon \omega\mathbf{Cat}^+ \to \mathsf{Mod}(\mathit{ICaTT})$$
$$(\mathrm{R}\, X)(\Gamma) = \omega\mathbf{Cat}^+(\mathrm{L}(\!|\Gamma|\!), X)$$

证明这一猜想需要说明 $\mathrm{L} \circ (\!|-|\!)$ 保持沿展示映射的推出，并将 L 识别为其限制 $\mathrm{L} \circ (\!|-|\!)$ 的 Yoneda 延拓。等价地，这意味着 L 是余连续的。

参考文献

[1] Thorsten Altenkirch 和 Ondrej Rypacek。2012。弱ω-群胚的句法方法。载于《计算机科学逻辑》（CSL'12）——第26届国际研讨会/EACSL第21届年度会议（CSL 2012）。施洛斯·达格施图尔—莱布尼茨信息学中心有限公司，瓦德恩/萨尔布吕肯，德国，15页。doi: 10 4230 2012 16 。/LIPICS.CSL.。Dimitri Ara。博士学位论文。巴黎第七大学。2010。https://www. normalesup.org/~ara/files/these.pdf[3] Michael A. Batanin。1998。幺半球形范畴：弱 *n*-范畴理论的自然环境。《数学进展》136，1（1998 39 103 10 1006 1998 1724），–。doi:。/aima.。Thibaut Benjamin 和 Ioannis Markakis。2024。 *ω*-范畴中的可逆胞。arXiv:2406.12127[5] Thibaut Benjamin 和 Ioannis Markakis。2025。计算图的 Hom *ω*-范畴是自由的。《伦敦数学会杂志》112，6（2025）。doi:10.1112/ jlms.70367[6] Thibaut Benjamin、Ioannis Markakis、Wilfred Offord、Chiara Sarti 和 Jamie Vicary。2025。高维路径类型的自然性。载于第40届ACM/IEEE计算机科学逻辑研讨会（LICS 2025），新加坡，6月 *23 26, 2025* 275 288 10 1109 65433 2025 00028 -。IEEE，–。doi:。/LICS。。[7] Thibaut Benjamin、Ioannis Markakis 和 Chiara Sarti。2024。。CaTT 上下文是有限 computad。载于第40届程序语义数学基础会议论文集。doi: 10 48550 2405 00398 。/arxiv.。Thibaut Benjamin、Samuel Mimram 和 Eric Finster。2024。。作为类型论模型的球形弱*ω*-范畴。《高阶结构》8，2（2024），1–69。doi:10.21136/HS.2024.07[9] John Bourke。2020。迭代代数单射性与忠实性猜想。《高阶结构》 4, 2 2020 183 210 10 21136 2020 13 ( )，–。doi:。/HS.。[10] Guillaume Brunerie。2016。。同伦类型论中的球面同伦群。博士学位论文。尼斯-索菲亚·安蒂波利斯大学。doi:10.48550/ arXiv.1606.05916[11] John Cartmell。1986。广义代数理论与上下文范畴。《纯粹与应用逻辑年刊》32（1986），209–243。doi:10.1016/0168-0072(86) 90053-9[12] Clémence Chanavat 和 Amar Hadzihasanovic。2024。图解（ ∞ *n* 2410 19053 , ）-范畴的模型结构。arXiv:。[13] Eugenia Cheng。2007。。具有所有对偶的一个-范畴是一个 *ω*-群胚。《应用范畴结构》 *ω* 15, 4 2007 439 453 10 1007 10485 007 9081 8 ( )，–。doi:。/s - - [14] - Christopher J. Dean 、 Eric Finster 、 Ioannis Markakis 、 David Reutter 和 Jamie Vicary。2024。弱 *ω*-范畴的计算图作为归纳类型。《数学进展》 450 2024 10 1016 2024 109739 ( )。doi:。/j.aim.。Eric Finster 和 Samuel Mimram。2017。。弱*ω*-范畴的类型论定义。载于第32届ACM/IEEE计算机科学逻辑研讨会论文集。ACM， 1 12 10 5555 3329995 3330059 –。doi:。/。[16] Jonas Frey。2025。。族的对偶性：Gabriel-Ulmer对偶的扩展。《符号逻辑杂志》（ 2025 1 38 10 1017 2024 79 ），–。doi:。/jsl.。Soichiro Fujii、Keisuke Hoshino 和 Yuki Maehara。2025。。严格与弱 *ω* 2511 09849 -范畴之间的 *ω*-等纤维化。[18] arXiv:。Soichiro Fujii、Keisuke Hoshino 和 Yuki Maehara。2025。。弱 *ω*-范畴之间的 *ω*-弱等价。《数学进展》480（2025）。doi:10.1016/j. aim.2025.110490[19] Alexander Grothendieck。1983。《追寻叠》。[20] Amar Hadzihasanovic、Félix Loubaton、Viktoriya Ozornova 和 Martina Rovelli。2025。《相干的行走 *ω*-等价的一个模型》。美国数学会会刊 153, 7 2025 2813 2827 10 1090 17197 ( )，–。doi: 。/proc/ Simon Henry 和 Edoardo Lanari。2023。。《关于 3-群胚的同伦假设》。范畴理论与应用 39，26（2023），735–768。[22] Simon Henry 和 Félix Loubaton。2025。《严格 ∞-范畴的归纳模型结构》。纯粹与应用代数杂志 229，1（2025）。doi:10.1016/j.jpaa. 2024 107859 。Keisuke Hoshino、Soichiro Fujii 和 Yuki Maehara。2024。《弱 *ω*-范畴中的弱可逆胞腔》。《高阶结构》8，2（2024），386–415。doi:10.21136/hs.2024.14[24] C. Barry Jay 和 Neil Ghani。1995。《η-展开的优点》。函数式编程杂志 5, 2 1995 135 154 10 1017 0956796800001301 ( )，–。doi: 。/ S Andre Joyal。2017。。《关于族与部落的札记》。arXiv: 。1710 10238 Yves Lafont、François Métayer 和 Krzysztof Worytkiewicz。2010。《 *ω*-cat 上的民间模型结构》。《数学进展》224，3（2010），1183–1231。arXiv: 0712 0617 10 1016 2010 01 007 。doi: 。/j.aim. . 。Andrea Laretto、Fosco Loregian 和 Niccolò Veltri。2026。《有向即 Di-is：通过二重自然性的一阶有向类型论》。ACM 编程语言会议论文集 10, 1759 1789 10 1145 3776703 –。POPL 专刊。[28] doi: 。/ Tom Leinster。2004。。《高阶歌剧，高阶范畴》。剑桥大学出版社。doi:10.1017/cbo9780511525896 [29] 2024 ∞ *ω* 2406 05425 Félix Loubaton。2009。。《(,)-范畴的范畴论》。arXiv: 。Peter LeFanu Lumsdaine。。《以内涵类型论构造弱 -范畴》。载于《带类型 λ 演算及其应用》，Pierre-Louis Curien（编）。第5608 卷。施普林格柏林海德堡，柏林，海德堡，172–187。doi:10.1007/978-3-642-02273-9 14[31] Georges Maltsiniotis。2010。《 Grothendieck ∞-群胚，以及另一种 ∞ 1009 2331 -范畴定义》。arXiv: 。Jacob Neumann 和 Thorsten Altenkirch。2025。《有向类型论中的合成 1-范畴》。载于第 30 届国际证明与程序类型会议（TYPES 2024）（《莱布尼茨国际信息学论文集》（LIPIcs），第 336 卷），Rasmus Ejlers Møgelberg 和 Benno van den Berg（编）。达格施图尔城堡—莱布尼茨信息学中心，7:1–7:23。doi:10.4230/LIPIcs.TYPES.2024.7[33] Paige Randall North。2019。《迈向有向同伦类型论》。《理论计算机科学电子札记》347，223–239。doi:10.1016/j.entcs.2019.09.012 编程语义数学基础第三十五届会议论文集。[34] Viktoriya Ozornova 和 Martina Rovelli。2024。《高阶范畴中的等价是什么？》《伦敦数学会会刊》56，1（2024），1–58。doi:10.1112/blms.12947 [35] 2020 2008 10307 Alex Rice。。[36] 2017 ∞ 。《高阶范畴中的余归纳可逆性》。arXiv: 。Emily Riehl 和 Michael Shulman。。《合成 -范畴的类型论》。《高阶结构》 1, 1 2017 147 224 10 21136 2017 06 ( )，–。doi: 。/hs. 。Paul Taylor。1987。。《递归域、指标化范畴论与多态性》。博士学位论文。剑桥大学。[38] 高等研究院单值基础计划。2013。《同伦类型论》。《[39] Benno van den Berg 和 Richard Garner。2011。《类型是弱 *ω*-群胚》。《伦敦数学会会刊》102，2（2011），370–394。doi:10.1112/plms/pdq026

A CaTT规则

在本附录中，我们完整介绍 CaTT 的规则。为便于阅读，我们用命名变量给出所有规则；也可以采用无名称变量的表示方式，通常通过 De Bruijn 指数实现。

$$\frac{}{\varnothing \vdash} \qquad \frac{\Gamma \vdash A}{\Gamma \triangleright (x : A) \vdash}\ (x \notin \operatorname{Var}\Gamma) \qquad \frac{\Gamma \vdash}{\Gamma \vdash x : A}\ (x : A) \in \Gamma$$

$$\frac{\Gamma \vdash}{\Gamma \vdash \diamond : \varnothing} \qquad \frac{\Delta \vdash \gamma : \Gamma \qquad \Gamma \vdash A \qquad \Delta \vdash t : A[\gamma]}{\Delta \vdash \gamma \triangleright (x \mapsto t) : \Gamma \triangleright (x : A)}$$

$$\star\text{-INTRO}\ \frac{\Gamma \vdash}{\Gamma \vdash \star} \qquad \to\text{-INTRO}_0\ \frac{\Gamma \vdash u : \star \qquad \Gamma \vdash v : \star}{\Gamma \vdash u \to_\star v}$$

$$\to\text{-INTRO}_+\ \frac{\Gamma \vdash u : v \to_A w \qquad \Gamma \vdash u' : v \to_A w}{\Gamma \vdash u \to_{v \to_A w} u'}$$

$$\text{PSS}\ \frac{}{(x : \star) \vdash_{\text{ps}} x : \star} \qquad \text{PSD}\ \frac{\Gamma \vdash_{\text{ps}} f : x \to_A y}{\Gamma \vdash_{\text{ps}} y : A}$$

$$\text{PSE}\ \frac{\Gamma \vdash_{\text{ps}} x : A}{\Gamma \triangleright (y : A) \triangleright (f : x \to_A y) \vdash_{\text{ps}} f : x \to_A y} \qquad \text{PS}\ \frac{\Gamma \vdash_{\text{ps}} x : \star}{\Gamma \vdash_{\text{ps}}}$$

$$\text{coh-INTRO}\ \frac{\Gamma \vdash_{\text{ps}} \qquad \Gamma \vdash_{\text{full}} A \qquad \Delta \vdash \gamma : \Gamma}{\Delta \vdash \operatorname{coh}_{\Gamma,A}[\gamma] : A[\gamma]}$$

B CaTT中的可逆性

本附录将更详细地介绍 Benjamin 和 Markakis [4], 的定理 3.8 的构造；ICaTT 中 can 构造子的 β-归约规则正是基于该构造。除了悬挂之外，逆元的构造还依赖于 CaTT 的另一项元操作，该操作由 Benjamin 和 Markakis [5, 第 5], 节定义，即形成对偶。给定一个正自然数 $n \in \mathbb{N}_{>0}$，运算 op_n 会交换 n维项的源和目标。它在有效上下文、类型、项和代换上递归定义如下：

$$\operatorname{op}_n \varnothing = \varnothing \qquad \operatorname{op}_n \Gamma \triangleright (x : A) = (\operatorname{op}_n \Gamma) \triangleright (x : \operatorname{op}_n A)$$

$$\operatorname{op}_n \star = \star \qquad \operatorname{op}_n (u \to_A v) = \begin{cases} \operatorname{op}_n u \to \operatorname{op}_n v & (\dim A + 2 \neq n) \\ \operatorname{op}_n v \to \operatorname{op}_n u & (\dim A + 2 = n) \end{cases}$$

$$\operatorname{op}_n x = x \qquad \operatorname{op}_n \operatorname{coh}_{\Gamma,A}[\gamma] = \operatorname{coh}_{\overline{\operatorname{op}_n \Gamma},(\operatorname{op}_n A)[\operatorname{op}_n^{\Gamma,-1}]}[\operatorname{op}_n^\Gamma \circ \operatorname{op}_n \gamma]$$

$$\operatorname{op}_n \diamond = \diamond \qquad \operatorname{op}_n \gamma \triangleright (x \mapsto t) = (\operatorname{op}_n \gamma) \triangleright (x \mapsto \operatorname{op}_n t)$$

其中 $\operatorname{op}_n \Gamma \vdash \operatorname{op}_n^\Gamma : \overline{\operatorname{op}_n \Gamma}$是 $\operatorname{op}_n \Gamma$ 与一个粘贴图之间唯一的同构。该元操作保持有效上下文、项、类型和代换，并且在唯一同构意义下保持粘贴图和满性。

这使我们能够更详细地阐述定理 3.8 的构造。我们从 Benjamin 和 Markakis 的以下中间结果开始： [4, 引理 31] , 它允许我们为复合构造逆元。

命题 B.1. 考虑一个维数至多为 n 的粘贴图 Γ，以及一个 ω-函子 $\gamma : [\![\Gamma]\!] \to X$，并且对于每个 n 维变量 $x \in \operatorname{Var}\Gamma$，都有一个定义在 $[\![x]\!](\gamma)$ 上的可逆性结构($x^{\text{linv}}, x^{\text{rinv}}$, $x^{\text{lunit}}, x^{\text{runit}}, x^{\text{ilunit}}, x^{\text{irunit}}$) 。则 $\operatorname{op}_n \Gamma = \Gamma$，且存在两个 ω-函子

$$\gamma^{\text{linv}} : [\![\Gamma]\!] \to X \qquad \gamma^{\text{rinv}} : [\![\Gamma]\!] \to X$$

它们在维数小于 n的胞腔上与 γ 一致，并将n维变量 $x \in \operatorname{Var}_n(\Gamma)$ 映射到

$$[\![x]\!](\gamma^{\text{linv}}) = (x[\operatorname{op}_n^\Gamma])^{\text{linv}} \qquad [\![x]\!](\gamma^{\text{rinv}}) = (x[\operatorname{op}_n^\Gamma])^{\text{rinv}}$$

我们通过几个例子来说明这一构造。首先，我们注意到，若 Γ 的维数严格小于 n，则该构造是平凡的，因为同构 $\operatorname{op}_n^\Gamma$是恒等映射，并且有 $\gamma^{\text{linv}} = \gamma^{\text{rinv}} = \gamma$。考虑以下一维粘贴图

$$\Gamma = \quad x \xrightarrow{f} y \xrightarrow{g} z$$

以及一个态射 $\gamma : [\![\Gamma]\!] \to X$，还有定义在 f和 g的像上的可逆性结构。代换 γ^{linv} 和 γ^{rinv} 可以图示如下：

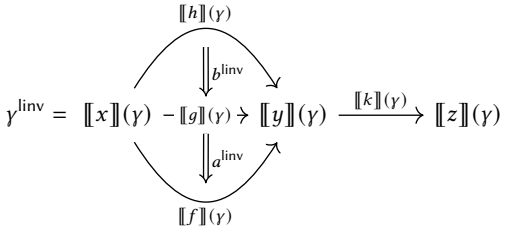
$$\gamma^{\text{linv}} = [\![z]\!](\gamma) \xrightarrow{g^{\text{linv}}} [\![y]\!](\gamma) \xrightarrow{f^{\text{linv}}} [\![x]\!](\gamma)$$

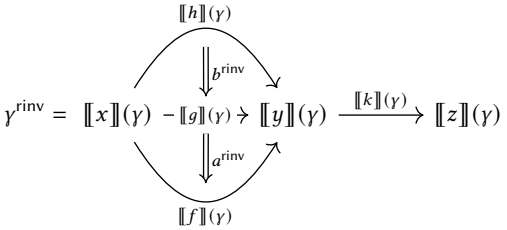
$$\gamma^{\text{rinv}} = [\![z]\!](\gamma) \xrightarrow{g^{\text{rinv}}} [\![y]\!](\gamma) \xrightarrow{f^{\text{rinv}}} [\![x]\!](\gamma)$$

现在考虑二维粘贴图

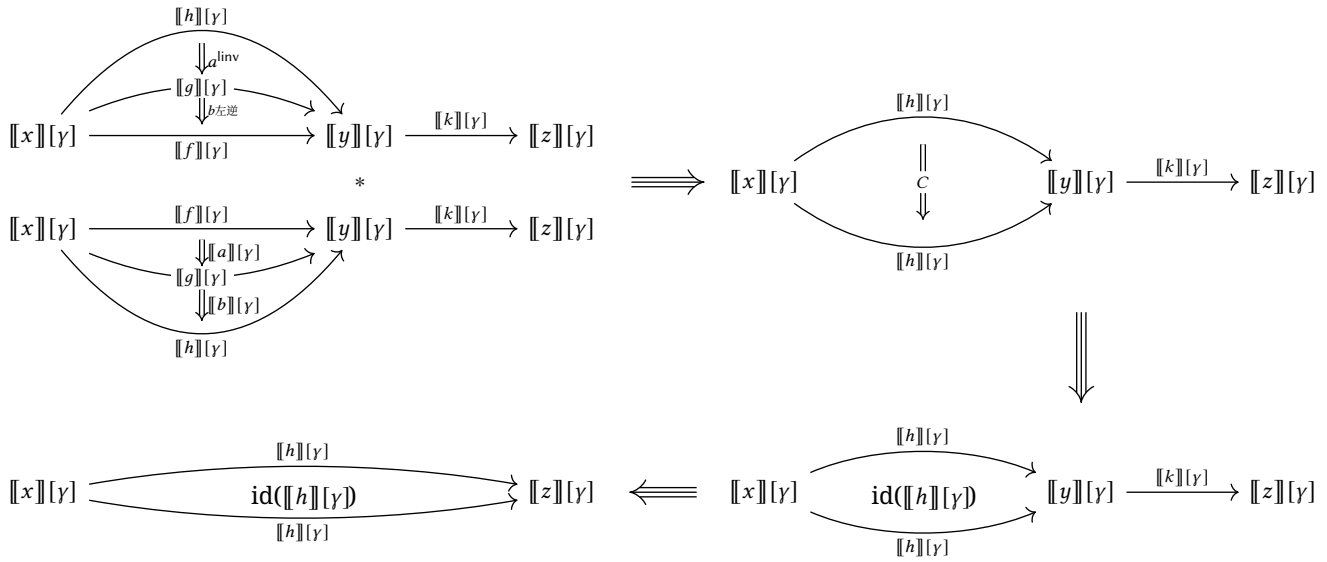
$$\Gamma = \quad x \underset{\Downarrow b}{\overset{f}{\underset{g}{\Rightarrow}}} y \xrightarrow{k} z$$

以及一个态射 $\gamma : [\![\Gamma]\!] \to X$，还有定义在 a和 b的像上的可逆性结构。代换 γ^{linv} 和 γ^{rinv} 可以图示如下：





一般来说，这一构造将 n维参数替换为其左逆或右逆，并颠倒复合的顺序。这使我们能够给出证明 ICaTT 所依赖的定理的构造：



Figu图 3：可逆胞腔复合的左消去元。此处 $C = (b^{\mathsf{linv}} * a^{\mathsf{linv}}) * (\llbracket a \rrbracket[\gamma] * \llbracket b \rrbracket[\gamma]$).

定理 3.8. *let* $\gamma\colon \llbracket \Gamma \rrbracket \to X$ 为从粘贴图出发的 ω-范畴态射，并 *let* $\Gamma \vdash_{\mathsf{full}} u \to v$ 为维数为n的满类型。当对每个$(n+1)$维变量 $x \in \mathsf{Var}(\Gamma)$， $\llbracket x \rrbracket(\gamma)$ 均可逆时，胞腔 $\llbracket \mathsf{coh}_{\Gamma, u \to v} \rrbracket(\gamma)$ 可逆。

证明概要。该定理通过余归纳证明，并给出显式的左右逆及左右单位。根据构造，左右单位满足该定理的假设，因此我们可以为它们构造可逆性结构。记 $t = \llbracket \mathsf{coh}_{\Gamma,\ u \to v} \rrbracket(\gamma)$为我们必须为其构造可逆性结构的胞腔。我们首先如下定义左右逆：

$$t^{\mathsf{linv}} = \llbracket \mathsf{op}_{n+1}\, \mathsf{coh}_{\Gamma, A} \rrbracket(\gamma^{\mathsf{linv}}) \qquad t^{\mathsf{rinv}} = \llbracket \mathsf{op}_{n+1}\, \mathsf{coh}_{\Gamma, A} \rrbracket(\gamma^{\mathsf{rinv}})$$

随后分三步定义左消去见证。第一步是一个结合子，它重新结合 $t^{\mathsf{linv}} * t$，以便将 n-胞腔链与其对应的逆元归在一起。第二步是将这些胞腔链消去为恒等，最后一步是一个广义单位子，它将恒等的复合与一个恒等联系起来。我们在图 3 中展示左消去见证的这三个步骤。右消去见证的情形是对称的。□

C ICaTT 的规则

本节旨在汇集并集中呈现 ICaTT 的所有规则。首先，我们汇集原始语法中代换对类型和项的作用规则，以及 action of the suspension on the raw syntax.

$$\mathsf{Inv}_A(u)[\sigma] = \mathsf{Inv}_{A[\sigma]}(u[\sigma])$$

$$\mathsf{can}(t, \{e_x\})[\sigma] = \mathsf{can}(t[\sigma], \{e_x[\sigma]\})$$

$$\mathsf{inv}(t, t_l, t_r, t_{lu}, t_{ru}, t_{ilu}, t_{iru})[\sigma] = \mathsf{inv}(t[\sigma], t_l[\sigma], t_r[\sigma], t_{lu}[\sigma], t_{ru}[\sigma], t_{ilu}[\sigma], t_i ru[\sigma])$$

$$\mathsf{rec}(t, t_l, t_r, t_{lu}, t_{ru}, t_{ilu}, t_{iru}, \gamma)[\sigma] = \mathsf{rec}(t, t_l, t_r, t_{lu}, t_{ru}, t_{ilu}, t_{iru}, \gamma \circ \sigma)$$

$$\mathsf{linv}(t)[\sigma] = \mathsf{linv}(t[\sigma]) \qquad \mathsf{rinv}(t)[\sigma] = \mathsf{rinv}(t[\sigma])$$

$$\mathsf{lunit}(t)[\sigma] = \mathsf{lunit}(t[\sigma]) \qquad \mathsf{runit}(t)[\sigma] = \mathsf{runit}(t[\sigma])$$

$$\mathsf{ilunit}(t)[\sigma] = \mathsf{ilunit}(t[\sigma]) \qquad \mathsf{irunit}(t)[\sigma] = \mathsf{irunit}(t[\sigma])$$

$$\Sigma(\mathsf{Inv}_A(t)) = \mathsf{Inv}_{\Sigma A}(\Sigma t) \qquad \Sigma(\mathsf{can}(t, \{e_x\})) = \mathsf{can}(\Sigma t, \{\Sigma e_x\})$$

$$\Sigma\, \mathsf{inv}(t, t_l, t_r, t_{lu}, t_{ru}, t_{ilu}, t_{iru}) = \mathsf{inv}(\Sigma t, \Sigma t_l, \Sigma t_r, \Sigma t_{lu}, \Sigma t_{ru}, \Sigma t_{ilu}, \Sigma t_{iru})$$

$$\Sigma\, \mathsf{rec}(t, t_l, t_r, t_{lu}, t_{ru}, t_{ilu}, t_{iru}, \gamma) = \mathsf{rec}(\Sigma t, \Sigma t_l, \Sigma t_r, \Sigma t_{lu}, \Sigma t_{ru}, \Sigma t_{ilu}, \Sigma t_{iru}, \Sigma\gamma)$$

$$\Sigma(\mathsf{linv}(e)) = \mathsf{linv}(\Sigma e) \qquad \Sigma(\mathsf{rinv}(e)) = \mathsf{rinv}(\Sigma e)$$

$$\Sigma(\mathsf{lunit}(e)) = \mathsf{lunit}(\Sigma e) \qquad \Sigma(\mathsf{runit}(e)) = \mathsf{runit}(\Sigma e)$$

$$\Sigma(\mathsf{ilunit}(e)) = \mathsf{ilunit}(\Sigma e) \qquad \Sigma(\mathsf{irunit}(e)) = \mathsf{irunit}(\Sigma e)$$

最后，图 4 汇总了 ICaTT 关于新类型和项构造子的规则。这些规则应与附录 A 中的规则结合起来理解。

$$\textsc{Inv-intro}\ \frac{\Gamma \vdash t : u \to_A v}{\Gamma \vdash \mathsf{Inv}_{u \to_A v}(t)}$$

(a) 可逆性结构类型的引入

$$\frac{\Gamma \vdash t :\ u \to_A v \qquad \Gamma \vdash e :\ \mathsf{Inv}(t)}{\Gamma \vdash \mathsf{linv}(e) :\ v \to_A u} \qquad \frac{\Gamma \vdash t :\ u \to_A v \qquad \Gamma \vdash e :\ \mathsf{Inv}(t)}{\Gamma \vdash \mathsf{rinv}(e) :\ v \to_A u} \qquad \frac{\Gamma \vdash t :\ u \to_A v \qquad \Gamma \vdash e :\ \mathsf{Inv}(t)}{\Gamma \vdash \text{左单位}(e) : \text{左逆}(e) \ast t \to \text{恒等} v}$$

$$\frac{\Gamma \vdash t :\ u \to_A v \qquad \Gamma \vdash e :\ \mathsf{Inv}(t)}{\Gamma \vdash \mathsf{runit}(e) :\ t \ast \mathsf{rinv}(e) \ \to \mathsf{id}\, u} \qquad \frac{\Gamma \vdash t :\ u \to_A v \qquad \Gamma \vdash e :\ \mathsf{Inv}(t)}{\Gamma \vdash \mathsf{ilunit}(e) :\ \mathsf{Inv}(\mathsf{lunit}(e))} \qquad \frac{\Gamma \vdash t :\ u \to_A v \qquad \Gamma \vdash e :\ \mathsf{Inv}(t)}{\Gamma \vdash \mathsf{irunit}(e) :\ \mathsf{Inv}(\mathsf{runit}(e))}$$

(b) 可逆性结构的析构子

$$\text{inv-intro}\ \frac{\Gamma \vdash t :\ u \to_A v \qquad \Gamma \vdash t_l :\ v \to_A u \qquad \Gamma \vdash t_r :\ v \to_A u \qquad \Gamma \vdash t_{lu} :\ t_l \ast t \to \text{恒等} v \qquad \Gamma \vdash t_{ru} :\ t \ast t_r \to \text{恒等} u \qquad \Gamma \vdash t_{ilu} :\ \mathsf{Inv}(t_{lu}) \qquad \Gamma \vdash t_{iru} :\ \mathsf{Inv}(t_{ru})}{\Gamma \vdash \mathsf{inv}(t,\ t_l,\ t_r,\ t_{lu},\ t_{ru},\ t_{ilu},\ t_{iru}) : \mathsf{Inv}(t)}$$

(c) 余归纳可逆结构的引入

$$\frac{\beta\text{-左逆}\ \Gamma \vdash \mathsf{inv}(t, t_l, t_r, t_{lu}, t_{ru}, t_{ilu}, t_{iru}) : \mathsf{Inv}^{u \to_A v}(t)\ \Gamma \vdash \text{左逆}(\mathsf{inv}(t, t_l, t_r, t_{lu}, t_{ru}, t_{ilu}, t_{iru}))\ \equiv t_l :\ v \to_A u \qquad \beta\text{-右逆}}{\Gamma \vdash \mathsf{inv}(t, t_l, t_r, t_{lu}, t_{ru}, t_{ilu}, t_{iru}) : \mathsf{Inv}^{u \to_A v}(t)\ \Gamma \vdash \text{右逆}(\mathsf{inv}(t, t_l, t_r, t_{lu}, t_{ru}, t_{ilu}, t_{iru}))\ \equiv t_r :\ v \to_A u}$$

$$\beta\text{-lunit}\ \frac{\Gamma \vdash \mathsf{inv}(t,\ t_l,\ t_r,\ t_{lu},\ t_{ru},\ t_{ilu},\ t_{iru}) :\ \mathsf{Inv}^{u \to}{}_{A v}(t)}{\Gamma \vdash \mathsf{lunit}(\mathsf{inv}(t,\ t_l,\ t_r,\ t_{lu},\ t_{ru},\ t_{ilu},\ t_{iru}))\ \equiv t_{lu} :\ t_l \ast t \ \to \mathsf{id}\, v} \qquad \beta\text{-runit}\ \frac{\Gamma \vdash \mathsf{inv}(t,\ t_l,\ t_r,\ t_{lu},\ t_{ru},\ t_{ilu},\ t_{iru}) :\ \mathsf{Inv}^{u \to}{}_{A v}(t)}{\Gamma \vdash \mathsf{runit}(\mathsf{inv}(t,\ t_l,\ t_r,\ t_{lu},\ t_{ru},\ t_{ilu},\ t_{iru}))\ \equiv t_{ru} :\ t \ast t_r \to \mathsf{id}\, u}$$

$$\beta\text{-ilunit}\ \frac{\Gamma \vdash \mathsf{inv}(t, t_l, t_r, t_{lu}, t_{ru}, t_{ilu}, t_{iru}) : \mathsf{Inv}^{u \to}{}_{A v}(t)}{\Gamma \vdash \mathsf{ilunit}(\mathsf{inv}(t,\ t_l,\ t_r,\ t_{lu},\ t_{ru},\ t_{ilu},\ t_{iru}))\ \equiv t_{ilu} : \mathsf{Inv}(t_{lu})} \qquad \beta\text{-irunit}\ \frac{\Gamma \vdash \mathsf{inv}(t,\ t_l,\ t_r,\ t_{lu},\ t_{ru},\ t_{ilu},\ t_{iru}) : \mathsf{Inv}^{u \to}{}_{A v}(t)}{\Gamma \vdash \mathsf{irunit}(\mathsf{inv}(t,\ t_l,\ t_r,\ t_{lu},\ t_{ru},\ t_{ilu},\ t_{iru}))\ \equiv t_{iru} : \mathsf{Inv}(t_{ru})}$$

$$\eta\text{-Inv}\ \frac{\Gamma \vdash e :\ \mathsf{Inv}(t)}{\Gamma \vdash e \equiv \mathsf{inv}(t,\ \mathsf{linv}(e),\ \mathsf{rinv}(e),\ \mathsf{lunit}(e),\ \mathsf{runit}(e),\ \mathsf{ilunit}(e),\ \mathsf{irunit}(e)) : \mathsf{Inv}(t)}$$

(d) β 和 η 余归纳可逆结构的规则

$$\textsc{can-intro}\ \frac{\Delta \vdash \mathsf{coh}_{\Gamma, A}[\gamma] : A[\gamma] \qquad \{\Delta \vdash e_x :\ \mathsf{Inv}(x[\gamma])\}_{x \in \mathsf{Var}_{\dim A + 1}(\Gamma)}}{\Delta \vdash \mathsf{can}(\mathsf{coh}_{\Gamma, A}[\gamma], \{e_x\}) :\ \mathsf{Inv}(\mathsf{coh}_{\Gamma, A}[\gamma])}$$

(e) 典范可逆性结构的引入规则

$$\beta_{\mathsf{can}}\text{-linv}\ \frac{\Delta \vdash \mathsf{can}(t, \{e_x\}_{x \in X}) : \mathsf{Inv}^{u \to v}(t)}{\Delta \vdash \mathsf{linv}(\mathsf{can}(t, \{e_x\}))\ \equiv t^{\mathsf{linv}} :\ v \to u} \qquad \beta_{\mathsf{can}}\text{-rinv}\ \frac{\Delta \vdash \mathsf{can}(t,\ \{e_x\}_{x \in X}) :\ \mathsf{Inv}^{u \to v}(t)}{\Delta \vdash \mathsf{rinv}(\mathsf{can}(t,\ \{e_x\}))\ \equiv t^{\mathsf{rinv}} :\ v \to u}$$

$$\beta_{\mathsf{can}}\text{-lunit}\ \frac{\Delta \vdash \mathsf{can}(t,\ \{e_x\}_{x \in X}) :\ \mathsf{Inv}^{u \to v}(t)}{\Delta \vdash \mathsf{lunit}(\mathsf{can}(t,\ \{e_x\}))\ \equiv t^{\mathsf{lunit}} :\ t^{\mathsf{linv}} \ast t \ \to \mathsf{id}\, v} \qquad \beta_{\mathsf{can}}\text{-runit}\ \frac{\Delta \vdash \mathsf{can}(t,\ \{e_x\}_{x \in X}) :\ \mathsf{Inv}^{u \to v}(t)}{\Delta \vdash \mathsf{runit}(\mathsf{can}(t,\ \{e_x\}))\ \equiv t^{\mathsf{runit}} :\ t \ast t^{\mathsf{rinv}} \to \mathsf{id}\, u}$$

$$\beta_{\mathsf{can}}\text{-ilunit}\ \frac{\Delta \vdash \mathsf{can}(t,\ \{e_x\}_{x \in X}) :\ \mathsf{Inv}^{u \to v}(t)}{\Delta \vdash \mathsf{ilunit}(\mathsf{can}(t,\ \{e_x\}))\ \equiv \mathsf{can}(t^{\mathsf{lunit}},\ \{-\}) :\ \mathsf{Inv}(t^{\mathsf{lunit}})} \qquad \beta_{\mathsf{can}}\text{-irunit}\ \frac{\Delta \vdash \mathsf{can}(t,\ \{e_x\}_{x \in X}) :\ \mathsf{Inv}^{u \to v}(t)}{\Delta \vdash \mathsf{irunit}(\mathsf{can}(t,\ \{e_x\}))\ \equiv \mathsf{can}(t^{\mathsf{runit}},\ \{-\}) :\ \mathsf{Inv}(t^{\mathsf{runit}})}$$

(f) β 典范可逆性结构的规则

$$\text{rec-INTRO}\ \frac{\begin{array}{c}\mathcal{E}^{n+1} \vdash t : x \to_A y \qquad \mathcal{E}^{n+1} \vdash t_l : y \to_A x \qquad \mathcal{E}^{n+1} \vdash t_r : y \to_A x \\ \mathcal{E}^{n+1} \vdash t_{lu} : t_l * t \to \mathsf{id}\, y \qquad \mathcal{E}^{n+1} \vdash t_{ru} : t * t_r \to \mathsf{id}\, x \qquad \mathsf{IH}_t^{n+1} \vdash t_{ilu} : \mathsf{Inv}(t_{lu}) \qquad \mathsf{IH}_t^{n+1} \vdash t_{iru} : \mathsf{Inv}(t_{ru}) \qquad \Gamma \vdash \gamma : \mathcal{E}^{n+1}\end{array}}{\Gamma \vdash \mathsf{rec}(t, t_l, t_r, t_{lu}, t_{ru}, t_{ilu}, t_{iru}, \gamma) : \mathsf{Inv}(t[\gamma])}$$

(g) Introduction rule for recursive definitions

$$\beta_{\mathsf{rec}}\text{-linv}\ \frac{\Gamma \vdash \mathsf{rec}(t, t_l, t_r, t_{lu}, t_{ru}, t_{ilu}, t_{iru}, \gamma) : \mathsf{Inv}_{u \to_A v}(t[\gamma])}{\Gamma \vdash \mathsf{linv}(\mathsf{rec}(t, t_l, t_r, t_{lu}, t_{ru}, t_{ilu}, t_{iru}, \gamma)) \equiv t_l[\gamma] : (v \to_A u)}$$

$$\beta_{\mathsf{rec}}\text{-rinv}\ \frac{\Gamma \vdash \mathsf{rec}(t, t_l, t_r, t_{lu}, t_{ru}, t_{ilu}, t_{iru}, \gamma) : \mathsf{Inv}_{u \to_A v}(t[\gamma])}{\Gamma \vdash \mathsf{rinv}(\mathsf{rec}(t, t_l, t_r, t_{lu}, t_{ru}, t_{ilu}, t_{iru}, \gamma)) \equiv t_r[\gamma] : (v \to_A u)}$$

$$\beta_{\mathsf{rec}}\text{-lunit}\ \frac{\Gamma \vdash \mathsf{rec}(t, t_l, t_r, t_{lu}, t_{ru}, t_{ilu}, t_{iru}, \gamma) : \mathsf{Inv}_{u \to_A v}(t[\gamma])}{\Gamma \vdash \mathsf{lunit}(\mathsf{rec}(t, t_l, t_r, t_{lu}, t_{ru}, t_{ilu}, t_{iru}, \gamma)) \equiv t_{lu}[\gamma] : t_l[\gamma] * t[\gamma] \to \mathsf{id}\, v}$$

$$\beta_{\mathsf{rec}}\text{-runit}\ \frac{\Gamma \vdash \mathsf{rec}(t, t_l, t_r, t_{lu}, t_{ru}, t_{ilu}, t_{iru}, \gamma) : \mathsf{Inv}_{u \to_A v}(t[\gamma])}{\Gamma \vdash \mathsf{runit}(\mathsf{rec}(t, t_l, t_r, t_{lu}, t_{ru}, t_{ilu}, t_{iru}, \gamma)) \equiv t_{ru}[\gamma] : t[\gamma] * t_r[\gamma] \to \mathsf{id}\, u}$$

$$\beta_{\mathsf{rec}}\text{-ilunit}\ \frac{\Gamma \vdash \mathsf{rec}(t, t_l, t_r, t_{lu}, t_{ru}, t_{ilu}, t_{iru}, \gamma) : \mathsf{Inv}_{u \to_A v}(t[\gamma])}{\Gamma \vdash \mathsf{ilunit}(\mathsf{rec}(t, t_l, t_r, t_{lu}, t_{ru}, t_{ilu}, t_{iru}, \gamma)) \equiv t_{ilu}[\mathsf{inst}_{\mathsf{rec}(t,t_l,t_r,t_{lu},t_{ru},t_{ilu},t_{iru},\gamma)} \circ \gamma] : \mathsf{Inv}(t_{lu}[\gamma])}$$

$$\beta_{\mathsf{rec}}\text{-irunit}\ \frac{\Gamma \vdash \mathsf{rec}(t, t_l, t_r, t_{lu}, t_{ru}, t_{ilu}, t_{iru}, \gamma) : \mathsf{Inv}_{u \to_A v}(t[\gamma])}{\Gamma \vdash \mathsf{irunit}(\mathsf{rec}(t, t_l, t_r, t_{lu}, t_{ru}, t_{ilu}, t_{iru}, \gamma)) \equiv t_{iru}[\mathsf{inst}_{\mathsf{rec}(t,t_l,t_r,t_{lu},t_{ru},t_{ilu},t_{iru},\gamma)} \circ \gamma] : \mathsf{Inv}(t_{ru}[\gamma])}$$

(h) β rules for recursive definition of coinductive invertibility structures

Figure 4: Rules of the theory ICaTT

D ICaTT 中的内部证明

本节旨在介绍我们在 ICaTT 中验证的结果的形式化证明。下方是我们在 ICaTT 类型论的原型实现中验证过的文件副本；该文件也作为 proofs/invertibility.catt 提供于补充材料中（可通过 https://zenodo.org/records/ 18343317 获取）。

有用的相干 coh unitr- (x(f)y) : f ->comp f (id _) coh unitl (x(f)y) : comp (id_) f -> f coh unitl- (x(f)y) : f ->comp (id _) f coh assoc (x(f)y(g)z(h)w) : comp f (comp g h) ->comp (comp f g) h coh assoc-(x(f)y(g)z(h)w) : comp (comp f g) h ->comp f (comp g h) coh unit3 (x(f)y(g)z): comp f (id _) g -> comp f g coh whiskl (x(f)y(g(a)h)z) : comp f g -> comp f h coh whiskr (x(f(a)g)y(h)z) : comp f h -> comp g h coh whisk3 (x(f)y(g(a)h)z(k)w) : comp f g k -> comp f h k coh assoc-le (x(f)y(g)z(h)w(k)v(l)u) : comp (comp (comp f g) h) (comp k l) -> comp f (comp g(comp hk)) l coh assoc-re (x(f)y(g)z(h)w(k)v(l)u) : comp (comp f g) (comp h (comp k l)) -> comp f (comp (compg h) k) l

命题 5.1

let compinv (x : *) (y : *) (z : *) (f : x -> y) (g : y ->z) (e : Inv (f)) (e' : Inv (g)) : Inv (comp f g)= can (comp f g { e , e'})

命题 5.2

let lri（x : *）（y : *）（f : x -> y）（e : Inv(f)）: linv（e）-> rinv（e）= comp（unitr-（linv（e））） （whiskl（linv（e））（linv（irunit（e））））（assoc _ _ _）（whiskr（lunit（e））（rinv（e））） （unitl（rinv(e)））

let lriU-aux (x : *) (y : *) (f : x -> y) (e : Inv(f)) (e' : Inv (linv (irunit (e)))) : Inv (lri e) = can (lri e {
can (_ {}) ,
can (_ { e' }) ,
can (_ {}) ,
can (_ { ilunit (e) }) ,
can (_ {})
})

rec linv-inv (x : *) (y : *) (f : x -> y) (e : Inv(f))= { linv(e) , f , f , comp (whiskl f (lri e)) (runit (e)) , lunit (e) , can (_ {can (_ { lriU-aux IHright }) , (irunit (e))}), ilunit (e) }

命题 5.3
let lriU (x : *) (y : *) (f : x -> y) (e : Inv(f)) : Inv (lri e) = lriU-aux (linv-inv irunit (e))

命题 5.4
inv rinv-inv (x : *) (y : *) (f : x -> y) (e : Inv(f))= { rinv(e) , f , f , runit (e) , comp (whiskr (linv (lriU e)) f) (lunit (e)) , irunit (e) , can (_ {can (_ { linv-inv (lriU e) }) , (ilunit (e))}) }

命题 5.5
逆传输 (x : *) (y : *) (f : x -> y) (e : Inv (f)) (g : x ->y) (a : f -> g) (e' : Inv (a))= {g, linv (e), rinv (e), comp (whiskl (linv (e)) (linv (e'))) (lunit (e)), comp (whiskr (linv (e')) (rinv (e))) (runit (e)), can (_ {can (_{ linv-inv e' }), ilunit (e)}), can (_ {can (_{ linv-inv e' }), irunit (e)}) }

命题 5.6 ### 逆 2of6-g (x : *) (y : *) (z : *) (w : *) (f : x ->y) (g : y ->z) (h : z -> w) (e : Inv(comp f g)) (e' : Inv(comp g h)) ={ g, comp (linv (e)) f, comp h (rinv (e')), comp (assoc- (linv (e)) f g) (lunit (e)), comp (assoc g h (rinv (e'))) (runit (e')), can (_ { can (_ {}), ilunit (e) }), can (_ { can (_ {}), irunit (e') }) }

let 2of6-f-runit
(x : *) (y : *) (z : *) (w : *) (f : x ->y) (g : y ->z) (h : z -> w) (e : Inv(comp f g)) (e' : Inv(comp g h))

```
    : comp f (comp g (rinv (e))) -> id x
    = comp (assoc f g rinv(e))
           (runit (e))
```

let 2of6-f-rwit

```
    (x : *) (y : *) (z : *) (w : *) (f : x -> y) (g : y ->
    z) (h : z -> w) (e : Inv(comp f g)) (e' :
    Inv(comp g h)) : Inv(2of6-f-runit e e')=
    can (2of6-f-runit e e' { can (_{}) , irunit
    (e)})
```

let 2of6-f-lunit

```
    (x : *) (y : *) (z: *) (w: *) (f : x -> y) (g : y ->z) (h : z -
    > w) (e : Inv(comp f g)) (e' :Inv(comp g h)) :
    comp (comp g (linv (e))) f -> id y= comp (unitr-
    (comp(comp g (linv (e))) f)) (whiskl_(linv (irunit
    (2of6-g e e')))) (assoc-le_ _ _ _ _) (whisk3 _
    (lunit (e))_) (unit3 _ _) (runit (2of6-g ee'))
```

let 2of6-f-lwit

```
    (x : *) (y : *) (z: *) (w: *) (f : x -> y) (g : y ->z) (h :
    z -> w) (e : Inv(comp f g)) (e' :Inv(comp g h)) :
    Inv (2of6-f-lunit e e')= can (2of6-f-lunit e e'{
    can(_ {}) , can(_ {linv-inv (irunit (2of6-g e
    e'))}) , can(_ {}) , can(_ { ilunit (e) }) , can(_ {}) ,
    (irunit (2of6-g e e')) })
```

inv 2of6-f

```
    (x : *) (y : *) (z: *) (w: *) (f : x -> y) (g : y ->z)
(h : z -> w) (e : Inv(comp f g)) (e' :Inv(comp g h))
= { f , comp g (linv (e)) , comp g (rinv (e)) ,
2of6-f-lunit e e', 2of6-f-runit e e' , 2of6-f-lwit e e' ,
2of6-f-rwit e e' }
```

let 2of6-h-lunit

```
    (x : *) (y : *) (z : *) (w : *)
    (f : x -> y) (g : y -> z) (h : z -> w)
    (e : Inv(comp f g)) (e' : Inv(comp g h))
    : comp (comp (linv (e')) g) h -> id w
    = comp (assoc- (linv (e')) g h)
           (lunit (e'))
```

let 2of6-h-lwit

```
    (x : *) (y : *) (z : *) (w : *)
    (f : x -> y) (g : y -> z) (h : z -> w)
    (e : Inv(comp f g)) (e' : Inv(comp g h))
    : Inv (2of6-h-lunit e e')
    = can (2of6-h-lunit e e' {
                can(_ {}) ,
                ilunit (e')})
```

let 2of6-h-runit

```
    (x : *) (y : *) (z : *) (w : *) (f : x ->y) (g : y ->z) (h : z -
    > w) (e : Inv(comp f g)) (e' : Inv(comp g h)) :
    comp h (comp (rinv (e')) g) -> id z= comp (unitl-
    (comp h (comp (rinv (e')) g))) (whiskr (rinv
    (ilunit (2of6-g e e'))) _) (assoc-re (linv (2of6-g e
    e')) g _ _ _) (whisk3 _ (runit (e'))_) (单位3 _ _)
    (lunit (2of6-g e e'))
```

let 2of6-h-rwit

```
    (x : *) (y : *) (z : *) (w : *) (f : x ->y) (g : y ->z) (h : z ->
    w) (e : Inv(comp f g)) (e' : Inv(comp g h)) :
    Inv(2of6-h-runit e e')= can (2of6-h-runit e e' { can(_
    {}) , can(_ {rinv-inv (ilunit (2of6-g e e'))}) , can(_ {}) ,
    can(_ { irunit (e') }) , can(_ {}) , (ilunit (2of6-g e e')) })
```

inv 2of6-h

```
    (x : *) (y : *) (z : *) (w : *) (f : x ->y) (g : y ->z) (h :
z -> w) (e : Inv(comp f g)) (e' : Inv(comp g h))= { h ,
comp (linv (e')) g , comp (rinv (e')) g , 2of6-h-lunit
e e' , 2of6-h-runit e e' , 2of6-h-lwit e e' ,
2of6-h-rwit e e' }
```