arXiv:2602.08846v1 [cs.LO] 9 Feb 2026

线性依赖类型理论中的非谓词性

Sam L. Speight 伯明翰大学计算机科学学院，英国

Niels van der Weide iCIS，奈梅亨拉德堡德大学，荷兰

摘要

我们基于线性组合代数构造了一个线性依赖类型论的可实现性模型。该模型促使我们对类型论作出若干补充。具体而言，我们加入了一个宇宙及其两种解码操作：一种将代码解码为笛卡尔类型，另一种将代码解码为线性类型。该宇宙具有非谓词性，即它同时对大型笛卡尔依赖积和大型线性依赖积封闭。我们还加入了一条规则，以保证将线性项转为笛卡尔项的模态具有单射性。作出这些补充后，我们能够编码（线性）归纳类型。作为案例研究，我们考察了线性类型上的列表类型，并证明我们的编码满足相关的唯一性原理。该可实现性模型的构造已在 Rocq 证明助手中得到完整形式化。

2012 年 ACM 学科分类 计算理论 → 范畴语义；计算理论 → 类型论；计算理论 → 线性逻辑

关键词与短语 可实现性、范畴语义、线性类型论、非谓词性

补充材料 软件： https://doi.org/10.5281/zenodo.18431109 [60]

资助 *Sam L. Speight*：本研究获得工程与自然科学研究委员会资助，项目编号为 EP/W034514/1

Niels van der Weide：本研究由 NWO 项目“平等的力量”OCENW.M20.380 资助，该项目由荷兰研究委员会（NWO）提供经费。

致谢 第一位作者感谢 Samson Abramsky 最初向其介绍线性组合代数。第二位作者感谢 Dan Frumin 提供的有益讨论。

1 引言

各种证明助手，例如 Rocq [62] 和 Lean [21], , 都基于**高阶类型论**。高阶类型论（例如构造演算） [18], 是强大的系统，为数学和计算机科学提供了合适的基础，也就是说，既适用于证明，也适用于编程。它们的关键特性之一是非谓词性。

非谓词性有多种形式，我们关注依赖类型论中的一种非直谓类型宇宙。这意味着该宇宙对于从任意类型到一个小类型的函数是封闭的。简而言之，小类型对于“大”积是封闭的。

非谓词性为逻辑系统增添了很大的力量。例如，这样的系统允许从 System F [24, 52], 进行翻译，使人们能够使用非谓词编码（或多态编码）将归纳类型归约为（依赖）函数类型。这类编码保证所得类型满足必要的递归和 β-原理。然而，它们通常不满足相关的归纳原理 [23, 55]。为弥补这些缺陷，人们已经开发了多种方法 [7, 9, 43, 71, 72]。

高阶类型理论面临的一项挑战是为其寻找语义。由于不存在高阶类型理论的（经典）集合论模型 [53], , 因此需要采用其他技术来构造其模型。**可实现性**可以追溯到 Kleene [34], , 如今已成为实现这一目标的主要方法之一 [61]。它的应用

范围从程序提取 [49] 到构造模型，以证明诸如形式化丘奇论题等公理的一致性 [31, 69] 以及参数多态性 [36]。

本文旨在将可实现性方法扩展到在类型系统中纳入**线性性** 的高阶类型理论。直观而言，线性性意味着函数不能删除或复制其参数：必须恰好使用每个参数一次。这种约束适用于多个应用领域，例如量子编程语言 [22] 和会话类型 [63]。

具体而言，我们着手构造一个具有非谓词性的线性依赖类型论（LDTT） [12, 13, 37, 68, 67, 40, 41] 可实现性模型。该模型证明非谓词性与 LDTT 相容。作为非谓词性在线性依赖类型论中的一个应用，我们使用非谓词编码构造一个列表的初始代数，其元素取自某个线性类型。

形式化 第 3 节和第 4 节的结果已在 Rocq 证明助手中形式化 [62]，使用 UniMath 库 [70]。我们大量运用了 UniMath 中关于范畴论的众多结果 [4, 6, 73]。我们的形式化成果已在网上公开 [60]，，文件 `Paper.v`说明了本文与形式化结果之间的对应关系。

本文始终采用集合论记法，而我们的形式化则基于类型论。尽管大部分开发内容不受这种基础差异的影响，但有一个方面值得一提：在集合论中，人们使用函子来定义 Grothendieck 纤维化；而在类型论中，则使用显示范畴 [5]。

贡献与概要 本文的主要贡献如下。

- 我们构造了一个具有非谓词类型宇宙的线性依赖类型论的可实现性模型（第 4节）。我们在 Rocq 证明助手中对该模型进行了形式化。我们展示了如何利用线性依赖类型论中
- 的非谓词性来构造初始列表代数（第 5节）。
-

第 2 节讨论线性依赖类型论。第 3节回顾（线性）组合子代数的基本理论。相关工作将在第 6 节讨论。

2 线性依赖类型论

本节回顾**线性依赖类型论**（LDTT）的语法，并介绍本文中贯穿使用的一些扩展。我们采用的 LDTT 版本最接近 Lundfall 的版本 [40, 41]，，而后者又基于 Krishnaswami、Pradic 和 Benton 的双上下文线性依赖类型理论 [37]。线性依赖类型论的不同形式的判断列于表1。上下文、类型和项的判断等式也有相应的判断，但我们将其省略。注意，我们同时具有笛卡尔上下文和混合上下文；后者包含一个笛卡尔部分和一个线性部分（其中任一部分都可以为空）。类型（无论是笛卡尔类型还是线性类型）只有在笛卡尔上下文中才是良构的，因此不能依赖于线性变量。笛卡尔项在笛卡尔上下文中进行类型判断；而线性项则在混合上下文中进行类型判断。

我们的类型论包含各种类型构造子。在笛卡尔侧，我们有单位元类型、$\prod$ 类型和 $\sum$ 类型，以及笛卡尔类型的恒等类型。我们对这些恒等类型采用等式反射，从而将命题等式与定义等式合并。这些构造子的规则都是常见的规则。

判断	预期含义
$\vdash \Gamma$上下文	Γ 是一个良构笛卡尔上下文
$\vdash \Gamma \mid \Xi$上下文	$\Gamma \mid \Xi$ 是一个良构的混合上下文，其笛卡尔部分为 Γ，线性部分为 Ξ
$\Gamma \vdash A$类型	A是笛卡尔上下文 Γ中的良构笛卡尔类型
$\Gamma \vdash A$线性	A 是笛卡尔上下文 Γ中的良构线性类型
$\Gamma \vdash a\colon A$	a 是上下文 Γ中的笛卡尔类型 A的笛卡尔项
$\Gamma \mid \Xi \vdash a\colon A$	a 是混合上下文 $\Gamma \mid \Xi$中的线性类型 A 的线性项

表1 线性依赖类型论中的不同形式的判断

$$\frac{\Gamma \vdash A\,\mathsf{type} \qquad \Gamma, x : A \vdash B\,\mathsf{lin}}{\Gamma \vdash \prod_{x:A} B\,\mathsf{lin}}\ \prod\text{-F} \qquad \frac{\Gamma, x : A \mid \Xi \vdash b : B}{\Gamma \mid \Xi \vdash \lambda x.\, b : \prod_{x:A} B}\ \prod\text{-I} \qquad \frac{\Gamma \mid \Xi \vdash t : \prod_{x:A} B \qquad \Gamma \vdash a : A}{\Gamma \mid \Xi \vdash t\ a : B[a/x]}\ \prod\text{-E}$$

$$\frac{\Gamma, x : A \mid \Xi \vdash b : B \qquad \Gamma \vdash a : A}{\Gamma \mid \Xi \vdash (\lambda x.\, b)\ a \equiv b[a/x] : B[a/x]}\ \prod\text{-}\beta \qquad \frac{\Gamma \mid \Xi \vdash b : \prod_{x:A} B}{\Gamma \mid \Xi \vdash \lambda x.\, b\ x \equiv b : \prod_{x:A} B}\ \prod\text{-}\eta$$

图1 线性依赖函数的规则

在线性侧，我们有非依赖乘法联结词（例如 $\multimap$、$\otimes$及其单位元），还有加性联结词（例如 $\oplus$、$\&$ 及其单位元）。这些规则都是线性类型论中熟知的常见规则，并以环境笛卡尔上下文的形式表述。此外，我们分别有 $\prod$ 和 $\sum$的线性类比项，即 $\prod$ 和 $\sum$. 。为展示其形式，$\prod$ 的规则见图1。请注意，索引类型必须是笛卡尔类型；$\sum$.也同样如此。

连接笛卡尔侧与线性侧，我们有两种 **模态**。模态 M将线性类型和项转换为笛卡尔类型和项。我们只能将M 应用于空线性上下文中的线性项。模态 L将线性类型和项转换为笛卡尔类型和项。我们在M 的图 2中给出了规则。

我们所假设的其余规则由第 4 节中的可实现性模型证明其合理性；据我们所知，此前在线性依赖类型论的框架中尚未考虑过这些规则。它们全都对第5节中的非谓词编码发挥作用。

等化子第一组规则对应于线性侧存在等化子。形成规则如下：

$$\frac{\Gamma \mid \cdot \vdash f : A \multimap B \qquad \Gamma \mid \cdot \vdash g : A \multimap B}{\Gamma \vdash \mathsf{Eq}(f, g)\,\mathsf{lin}}\ \text{EQ-F}$$

接下来是引入规则，它对应于等化子的泛性质中的存在性部分。

$$\frac{\Gamma \mid \Xi \vdash h : C \multimap A \qquad \Gamma \mid \Xi \vdash f \circ h \equiv g \circ h : C \multimap B}{\Gamma \mid \Xi \vdash \overline{h} : C \multimap \mathsf{Eq}(f, g)}\ \text{EQ-I}$$

消去规则给出了等化子映射。

$$\frac{}{\Gamma \mid \Xi \vdash \iota_{f,g} : \mathsf{Eq}(f, g) \multimap A}\ \text{EQ-E}$$

我们有两条 β-规则。第一条 β-规则表示，映射 f 和 g 在映射 $\iota_{f,\ g}$下被等化。注意，它们是在判断等式意义下被等化的。第二条 β-规则

$$\frac{\Gamma \vdash A\ \mathsf{lin}}{\Gamma \vdash \mathsf{M}(A)\ \mathsf{type}}\ \mathsf{M}\text{-F} \qquad \frac{\Gamma \mid \cdot \vdash a : A}{\Gamma \vdash \mathsf{M}(a) : \mathsf{M}(A)}\ \mathsf{M}\text{-I} \qquad \frac{\Gamma \vdash a : \mathsf{M}(A)}{\Gamma \mid \cdot \vdash \mu(a) : A}\ \mathsf{M}\text{-E}$$

$$\frac{\Gamma \mid \cdot \vdash a : A}{\Gamma \mid \cdot \vdash \mu(\mathsf{M}(a)) \equiv a : A}\ \mathsf{M}\text{-}\beta \qquad \frac{\Gamma \vdash a : \mathsf{M}(A)}{\Gamma \vdash \mathsf{M}(\mu(a)) \equiv a : \mathsf{M}(A)}\ \mathsf{M}\text{-}\eta$$

图 2 将线性类型和项映射为笛卡尔类型和项的模态 M 的规则

对应于泛态射上的交换条件。

$$\Gamma \mid \Xi \vdash f \circ \iota_{f,g} \equiv g \circ \iota_{f,g} : \mathsf{Eq}(f,g) \multimap B \qquad (\text{EQ-}\beta_1)$$
$$\Gamma \mid \Xi \vdash \iota_{f,g} \circ \overline{h} \equiv h : C \multimap A \qquad (\text{EQ-}\beta_2)$$

最后，我们有一条 η-规则，对应于泛态射的唯一性。

$$\frac{\Gamma \mid \Xi \vdash h, h' : C \multimap \mathsf{Eq}(f,g) \qquad \Gamma \mid \Xi \vdash \iota_{f,g} \circ h \equiv \iota_{f,g} \circ h' : C \multimap A}{\Gamma \mid \Xi \vdash h \equiv h' : C \multimap \mathsf{Eq}(f,g)}\ \text{EQ-}\eta$$

单射性 我们添加一条规则，用于表达 M 模态的单射性。

$$\frac{\Gamma \mid \cdot \vdash \mathsf{M}(a) \equiv \mathsf{M}(a') : \mathsf{M}(A)}{\Gamma \mid \cdot \vdash a \equiv a' : A}\ \mathsf{M}\text{-INJ}$$

从语义上说，这对应于函子 M 的忠实性（参见 例 4.11）。请注意，线性上下文为空，因为我们只允许在空线性上下文中应用 M 。

笛卡尔类型的恒等类型与模态 M 使得可以导出 线性 类型的恒等类型（参见 [40, 第 6 节。2]）。我们用 $a =_{\mathsf{L}} a'$ 表示 $\mathsf{M}(a) = \mathsf{M}(a')$，其中我们用 $=$ 表示笛卡尔类型的恒等类型。因此，我们只需考虑空线性上下文中两个线性项的恒等类型。

将 M-inj 与等式反射结合，可得到派生线性恒等类型的等式反射：假设我们有一个证明 $a =_{\mathsf{L}} a'$。根据定义，这就是一个证明，表明 $\mathsf{M}(a) = \mathsf{M}(a')$。通过等式反射，我们可以推出 $\mathsf{M}(a) \equiv \mathsf{M}(a')$。最后，利用 M-inj，我们得出 $\mathsf{M}(a) \equiv \mathsf{M}(a')$。我们将在第5节中用到线性等式反射。

我们为 $\prod$. 推导出以下函数外延性原理。为此，我们先使用 $\prod$-η 推导出一个判断性版本，然后利用等式反射得到命题性版本：

$$\frac{\Gamma \mid \cdot \vdash f, g : \prod_{x:A} B \qquad \Gamma, x : A \vdash p : f\ x =_{\mathsf{L}} g\ x}{\Gamma \vdash \mathsf{FunExt}(p) : f =_{\mathsf{L}} g}\ \prod\text{-FunExt}$$

类似地，非依赖线性函数类型也满足函数外延性。

非谓词宇宙 最后，我们给出一个非谓词塔斯基风格宇宙$\mathcal{U}$的规则。请注意，$\mathcal{U}$ 本身是一个笛卡尔类型，因为我们希望能够不受限制地使用类型为$\mathcal{U}$的变量（即类型的代码）。

$$\frac{\vdash \Gamma\ \text{context}}{\Gamma \vdash \mathcal{U}\ \mathsf{type}}\ \mathcal{U}\text{-F}$$

该宇宙配备了两种解码操作：第一种将代码解码为笛卡尔类型，第二种将代码解码为线性类型。这与分别拥有

笛卡尔宇宙和线性宇宙形成对比。

$$\frac{\Gamma \vdash A : \mathcal{U}}{\Gamma \vdash \mathsf{El_C}\,(A)\,\mathsf{type}}\ \mathsf{El_C} \qquad \frac{\Gamma \vdash A : \mathcal{U}}{\Gamma \vdash \mathsf{El_L}\,(A)\,\mathsf{lin}}\ \mathsf{El_L}$$

该宇宙是非谓词的，因为它对“大的”笛卡尔依赖积以及“大的”线性依赖积封闭。

$$\frac{\Gamma \vdash A\,\mathsf{type} \qquad \Gamma, x : A \vdash B : \mathcal{U}}{\Gamma \vdash \widehat{\prod_{x:A}} B : \mathcal{U}}\ \widehat{\prod}\text{-F} \qquad \frac{\Gamma \vdash A\,\mathsf{type} \qquad \Gamma, x : A \vdash B : \mathcal{U}}{\Gamma \vdash \widehat{\bigsqcap_{x:A}} B : \mathcal{U}}\ \widehat{\bigsqcap}\text{-F}$$

为了将这些“$\mathcal{U}$-内部”类型构造子与“外部”类型构造子区分开来，我们使用了 $(-)$ 装饰。我们规定，相关的内部与外部类型构造子之间存在一个同构：

$$\mathsf{El_C}\left(\widehat{\prod_{x:A}} B\right) \cong \prod_{x:A} \mathsf{El_C}\,(B) \qquad \mathsf{El_L}\left(\widehat{\bigsqcap_{x:A}} B\right) \cong \bigsqcap_{x:A} \mathsf{El_L}\,(B) \tag{1}$$

上述同构可用于导出抽象算子和应用算子。当不存在歧义时，例如当我们完全在$\mathcal{U}$中工作时，我们将对这些算子使用通常的记法（不带装饰）。对于第5节中的非谓述编码，我们还要求宇宙对线性函数和等化子封闭。

3 线性可实现性

本节中，我们回顾**线性组合代数** [2]的概念。在此之前，我们先考察**应用结构**和**组合代数** [56]的概念。

▶**定义3.1**。 应用结构是一个集合 A，连同一个二元运算 $(-) \bullet (=) : A \times A \to A$，我们称之为应用。

我们通常省略 $\bullet$，并且总是向左结合，因此，例如 $abc = (a \bullet b) \bullet c$。

以下定义的对象通常称为**组合子代数**。这里我们在前面加上“笛卡尔”一词，以强调它们与稍后将介绍的线性组合子代数之间的区别。

▶**定义3.2** （Schönfinkel [56]）。 一个笛卡尔组合代数（*CCA*）是一个应用结构 $(A, \bullet)$，其中存在组合子$\mathsf{S}', \mathsf{K}' \in A$，使得$\mathsf{S}'abc = ac\,(bc)$ 且$\mathsf{K}'ab = a$。

我们在CCA中的组合子上加撇号，以便将其与线性组合子代数中的组合子区分开来。

以下线性组合子代数的定义 [2]对应于线性逻辑中！$\multimap$片段的希尔伯特式公理化。

▶**定义 3.3**。 线性组合代数 （*LCA*）是一种应用结构$(A, \bullet)$，并配有一元运算！：$A \to A$，其中存在“组合子”$\mathsf{B}, \mathsf{I}, \mathsf{C}, \mathsf{W}, \mathsf{K}, \mathsf{D}\ \delta, \mathsf{F} \in A$，它们满足表 2中的恒等式。

线性组合代数的例子可见于文献 [2, 30]。我们遵循的另一项约定是，！的绑定优先级高于 $\bullet$。请注意，某些组合子只有在特定参数由!保护时才会进行计算。

组合子恒等式	主类型	逻辑原理
$\mathsf{B}abc = a(bc)$	$(\alpha \multimap \beta) \multimap (\gamma \multimap \alpha) \multimap \gamma \multimap \beta$	切割／复合
$\mathsf{I}a = a$	$\alpha \multimap \alpha$	恒等
$\mathsf{C}abc = acb$	$(\alpha \multimap \beta \multimap \gamma) \multimap \beta \multimap \alpha \multimap \gamma$	交换
$\mathsf{W}a!b = a!b!b$	$(!\alpha \multimap !\alpha \multimap \beta) \multimap !\alpha \multimap \beta$	收缩
$\mathsf{K}a!b = a$	$\alpha \multimap !\beta \multimap \alpha$	弱化
$\mathsf{D}!a = a$	$!\alpha \multimap \alpha$	降级
$\delta!a = !!a$	$!\alpha \multimap !!\alpha$	余乘法
$\mathsf{F}!a!b = !(ab)$	$!(\alpha \multimap \beta) \multimap !\alpha \multimap !\beta$	函子性

表 2 组合子恒等式及其对应的主类型和逻辑原理

对于应用结构，存在 S′ 和 K′ 组合子，当且仅当存在 B′、C′、K′ 和 W′ 组合子，其中 B′、C′ 和 W′ 满足表 2 中的恒等式，但省略其中所有出现的 !。注意，我们可以定义 I′ :=W′K′。

正如 Abramsky、Haghverdi 和 Scott 所示 [2, 27],，从任意线性组合代数 $A = (A, \bullet, !)$，我们都可以构造一个笛卡尔组合代数 $A_! := (A, \bullet_!)$，其底层集合与之相同，且 $a \bullet_! b := a \bullet !b$。

组合完备性 笛卡尔组合代数的关键性质是能够模拟 λ-抽象。这一性质称为 **组合完备性**。线性组合代数也具有自身形式的组合完备性，它与线性 λ-演算有关。在为我们模型的实现元提供实现子时，这些性质极为有用，详见 第 4 节。

设 A 为一个集合， V 为一个可数无限的变量集合。 A 上的多项式由以下文法的前三条规则给出；!-多项式则由整个文法给出。

$$t, t' ::= x \in V \mid a \in A \mid t \bullet t' \mid !t$$

对于带 ! 的应用系统，我们采用相同的约定来处理（!-）多项式。

闭多项式可以视为 A 的元素。如果 A 带有一个 ! : $A \to A$，那么闭 !-多项式可以视为 A 的元素。若对于所有 $a_1, ..., a_n \in A$, $t[a_1, ..., a_n/x_1, ..., x_n] = t'[a_1, ..., a_n/x_1, ..., x_n]$ 都成立，则两个（!-）多项式 t, t' （其变量取自 $\{x_1, ..., x_n\}$）视为相等。若变量在 !-多项式 t 中恰好出现一次，且不处于 ! 的作用域内，则称其为线性的 。我们用 vars(t) 表示（!-）多项式 t 中出现的变量集合，用 linvars(t') 表示 !- 多项式 t' 中出现的线性变量集合。

▶ 命题 *3.4*（*Curry* [20]）。对于应用系统 $(A, \bullet)$，以下条件等价：$(A, \bullet)$ 是笛卡尔组合代数。$(A, \bullet)$ 是笛卡尔组合完备的。也就是说，对于每个在 A 上的多项式 t，都存在一个在 A 上的多项式 $\lambda^* x.t$，满足 $vars(\lambda^* x.t) = vars(t) - \{x\}$，使得 $(\lambda^* x.t)x = t$。

▶ **命题 3.5** (Simpson [59])。设 $(A, \bullet)$ 是一个应用系统，并且 ! : $A \to A$。以下条件等价：

- $(A, \bullet, !)$ 是线性组合代数。$(A, \bullet, !)$ 是线性组合完备的。也就是说，对于每个关于 A 的 !- 多项式 t : 如果 $x \in linvars(t)$，那么存在一个关于 A 的多项式 $\lambda^* x.t$，满足 $vars(\lambda^* x.t) = vars(t) - \{x\}$ 且 $linvars(\lambda^* x.t) = linvars(t) - \{x\}$，使得 $(\lambda^* x.t)x = t$；还存在一个关于 A 的多项式 $\lambda!^* x.t$，满足 $vars(\lambda!^* x.t) = vars(t) - \{x\}$ 且 $linvars(\lambda!^* x.t) = linvars(t) - \{x\}$，使得 $(\lambda!^* x.t)!x = t$。

4 线性可实现性模型

本节中，我们构造线性依赖类型论的可实现性模型。传统的可实现性方法从一个（笛卡尔）组合代数出发，然后利用该笛卡尔组合代数上的装配构造 Martin-Löf 类型论的模型 [42, 51]。这类模型通过某种范畴结构来表示，通常会用到理解范畴 [33]。我们旨在采用相同的方法，但使用线性组合代数来处理线性依赖类型论。为了对线性依赖类型论建模，我们采用一种更精细的范畴结构，即**线性理解范畴**，其基础是 Lundfall 的相关工作 [40, 41]。

本节首先回顾理解范畴（定义 4.2）以及 Martin-Löf 类型论的可实现性模型（例 4.6）。随后，我们介绍线性理解范畴（定义 4.9）。在本节中，我们始终说明每个线性组合代数都会产生一个线性理解范畴，称为 **线性可实现性模型**。我们将逐步讨论其相关特性。具体而言，我们首先讨论笛卡尔类型（例 4.6）和线性类型（例 4.8），以及模态（例 4.11）。最后，我们解释各种类型构造子（例 4.13），包括等化子类型（例 4.15）和非谓词宇宙（例 4.18）。

4.1 类型论的可实现性模型

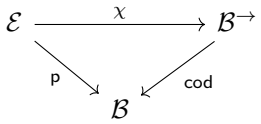
我们首先简要概述理解范畴以及通常的依赖类型论可实现性模型。理解范畴的关键要素是 *Grothendieck* 纤维化 [26]，其定义如下。

▶ 定义 *4.1.* 设 $p : E \to B$ 是一个函子，且 $f : x \to y$ 是 E 中的一个态射。如果对于所有对象 $z \in E$ 以及态射 $h : z \to y$ 和 $g : p(z) \to p(x)$，只要满足 $p(h) = p(f) \circ g$，就存在唯一的 $k : z \to y$ 使得 $p(k) = g$，并使所得三角形交换，则称 f 为笛卡尔态射。如果对于 B 中所有 $f : x \to p(y)$，都存在 E 中的对象 $x \in$，满足 $p(x) = x$，并存在一个笛卡尔态射 $f : x \to y$，满足 $p(f) = f$，则称 p 是一个（格罗滕迪克）纤维化。

若有一个纤维化 $\mathrm{p} : \mathcal{E} \to \mathcal{B}$，则可以将 $\mathcal{B}$ 和 $\mathcal{E}$ 分别看作上下文范畴和类型范畴。给定一个对象 $x \in \mathcal{B}$，我们用 $\mathrm{p}[x]$表示 p 沿 x 的纤维范畴。每个态射 $s : x \to y$ 都会导出一个函子 $s^* : \mathrm{p}[y] \to \mathrm{p}[x]$，从而为类型赋予代换运算。

理解范畴 [33] 将纤维化与用于解释上下文扩展的理解函子结合起来。我们将其定义如下。

▶**定义 4.2**。 理解范畴是如下所示的严格交换函子图表。



我们假设p 是一个纤维化，并且 χ 保留笛卡尔态射。我们将这样的理解范畴记为 $\chi : E \to B^{\to}$。若 $\chi : E \to B^{\to}$ 是满的，则称其为满理解范畴；若 χ 是全忠实的，则称其为满。

若我们有 $\overline{x} \in \mathcal{E}$ 且 $\mathrm{p}(x) = \overline{x}$，则将 $\chi(x)$ 记为 $\pi_x : x.\overline{x} \to x$。–$\chi$保留笛卡尔态射这一事实意味着，它将每个笛卡尔提升映射为一个拉回方块。

▶注 4.3. 本文始终只考虑满理解范畴。若一个理解范畴不是满的，那么它还带有附加结构，即态射

类型之间的态射。这种结构可以解释子类型 [19, 47], ，而这通常不会出现在 Martin-Löf 类型论中。由于我们只关注不含子类型的模型，因此只考虑满理解范畴。

此外，我们只考虑这样的理解范畴：p 具有纤维终对象，并且 χ 保持终对象。在这类理解范畴中，上下文 x 中类型为 x 的项，与 $\mathsf{p}[x]$ 中从终对象到 x的态射相同。

依赖积和依赖和的解释基于 Lawvere 的观点：量词是伴随 [33, 39]。

▶ **定义 4.4**。 如果一个满理解范畴 $\chi\colon E \to B^{\to}$ 支持 $\sum$-类型，那么对于所有 $s : x \to y$，函子 s^* 都有一个满足通常 *Beck-Chevalley* 条件的左伴随。我们用 $\sum$表示这个左伴随。如果 $\chi : E \to B^{\to}$ 支持 $\sum$-类型，那么如果对于所有对象 $x \in B$， $x \in \mathsf{p}[x]$ 和 $y \in \mathsf{p}[x.x]$，从 $x.x.y$ 到 $x.\sum_x y$ 的典范态射是同构，我们就称这些 $\sum$-类型是强的 。

理解范畴 $\chi : E \to B^{\to}$ 支持 $\prod$-类型，如果对所有 $s : x \to y$，函子 s^* 都有一个满足通常 *Beck-Chevalley* 条件的右伴随。

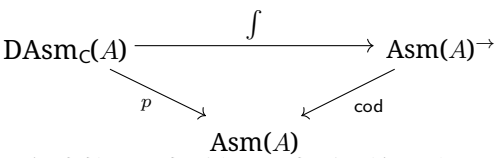
▶ 注 4.5。 关于定义 4.4，有几点重要观察。虽然通常只考虑弱化函子 π_x^*的伴随，但我们考虑所有态射的伴随。因此，定义 4.4 所刻画的不仅有 $\sum$-类型，还有外延恒等类型，后者由收缩函子的左伴随给出。此外，只需检验 $\sum$-类型的 Beck-Chevalley 条件，因为 $\prod$-类型的条件可由此推出 [57]。最后，为了诠释依赖和的 $\sum$-规则，我们要求 η-类型具有强性。

现在我们构造依赖类型论 [42, 51] 的可实现性模型。

▶ **例 4.6**。 我们证明，每个笛卡尔组合代数 A 都能导出一个具有 $\prod$ 和 $\sum$-类型的外延类型论模型。首先，我们定义上下文范畴。回忆一下， A 上的一个组装是一个集合 X 与一个关系 $\Vdash \subseteq A \times X$ 组成的有序对，满足对每个 $x \in X$，都存在 $a \in A$ 使得 $a \Vdash x$。我们称 $\Vdash$ 为X的“可实现性关系”。给定组装 X 和 Y，一个组件态射由函数 $f\colon X \to Y$ 给出，且存在 $e \in A$，使得对每个满足 $a \Vdash x$ 的 $a \in A$，都有 $ea \Vdash f(x)$。我们用 $\mathsf{Asm}(A)$ 表示这样的范畴，其对象是 A 上的组装，态射是组件态射。事实上，对于任何具有 B 与 I 组合子的应用结构，我们都可以构造组装范畴。

接下来，我们定义类型范畴 $\mathsf{DAsm_C}(A)$。该范畴的对象是**组件族**，即由一个装配 X 以及对每个 $x \in X$ 都有一个装配 Y_x 组成的序对。从 $(X,\ Y)$ 到 $(X',\ Y')$ 的态射由一个组件态射 $f\colon X \to X'$ 以及对每个 x 都有一个映射 $g_x\colon Y_x \to Y_{f(x)}$ 组成，并且存在 $e \in A$，使得当 $a \Vdash x$ 且 $b \Vdash y$ 时， $eab \Vdash g_x(y)$。第一投影 $p\colon \mathsf{DAsm_C}(A) \to \mathsf{Asm}(A)$ 是一个纤维化。

最后，我们定义一个函子 $\int : \mathsf{DAsm_C}(A) \to \mathsf{Asm}(A)^{\to}$。若给定一个装配 X以及对 $x \in X$ 的装配 Y_x，则定义 $\int_{x:\ X} Y_x$ 为其底层集合是 $\sum_{x:\ X} Y_x$ 的装配。总而言之，我们得到如下理解范畴。



这种理解支持 $\sum$-类型和 $\prod$-类型，并且函子 $\int$ 是全忠实的，且保留终对象。此外，每个纤维范畴 $p[x]$ 都有一个终对象。

4.2 线性理解范畴

接下来，我们将理解范畴的概念扩展到涵盖线性类型论的特征，并定义线性理解范畴。这类理解范畴还有两个额外特征。第一个特征是，它们包含两种类型概念，即笛卡尔类型和线性类型。我们通过同一范畴上的两个纤维化来表示这两类类型。为体现线性性，我们要求其中一个纤维化是**对称幺半闭**的。

▶ **定义 4.7.** 设 $\mathsf{q} : E \to B$ 是一个纤维化。若对于每个 $x \in B$，纤维 $\mathsf{q}[x]$ 都是对称幺半闭范畴，且对于每个 $s : x \to y$，函子 $s^* : \mathsf{q}[y] \to \mathsf{q}[x]$ 都是对称幺半闭函子，则称q 是一个对称幺半闭纤维化 。

请注意，文献中存在多种关于幺半纤维化的概念，[46, 58]具体取决于幺半结构是定义在总范畴上还是纤维范畴上。对于我们的目的，需要求纤维范畴具有幺半结构。

从类型论的角度来看，我们可以如下理解对称幺半闭纤维化。每个纤维范畴 $\mathsf{q}[x]$ 都可视为某个上下文x中的类型范畴。幺半结构为这些范畴赋予了三个类型构造子：单位元、幺半积和线性函数类型，因此每个 $\mathsf{q}[x]$ 都构成线性逻辑的一个模型。由于函子 $s^* : \mathsf{q}[y] \to \mathsf{q}[x]$ 是对称幺半闭函子，上述类型构造子在代入下保持稳定，直到同构。

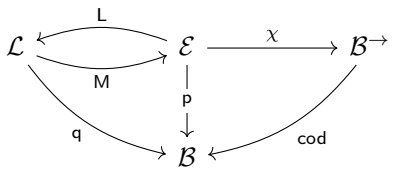
利用线性组合代数，我们构造一个幺半纤维化的例子。

▶ **例 4.8.** 令 A 为一个线性组合代数（LCA）。我们定义范畴 $\mathsf{DAsm}_{\mathsf{L}}(A)$，其对象是 A 上的线性组件族，态射则是它们之间的态射。一个**线性组件族**over A由一对对象组成：一个是装配 $X \in \mathsf{Asm}(A_!)$，它定义在与 LCA A相关联的笛卡尔组合代数 $A_!$ 上；另一个是对每个 $x \in X$，一个定义在 LCA A 上的装配 $Y_x : \mathsf{Asm}(A)$。从 (X, Y) 到 (X', Y') 的态射由一个态射 $f : X \to X'$（属于 $\mathsf{Asm}(A_!)$）以及对每个 x的一个映射 $g_x : Y_x \to Y_{f(x)}$ 组成，并且存在 $e \in A$，使得当 $a \Vdash x$ 且 $b \Vdash y$时，$e!ab \Vdash g_x(y)$。函子 $q : \mathsf{DAsm}_{\mathsf{L}}(A) \to \mathsf{Asm}(A_!)$ 是第一投影。

函子 q 是一个对称幺半闭纤维化。$q[X]$ 上的幺半结构的单位元由一个以 I 为唯一实现元的单位集合给出。线性组件 Y与 Y'的幺半积以 $Y(x) \times Y'(x)$（$x \in X$）为载体；若存在 $b, b' \in A$，使得 $b \Vdash y$、$b' \Vdash y'$ 和 $a = \lambda^* x.\, xbb'$，则称 $a \Vdash (y, y')$。

线性理解范畴的第二个关键特征在于对线性指数化的诠释。这个联结词有多种范畴论表示方式，而我们的方法基于线性-非线性模型 [10]。具体而言，我们要求每个线性理解范畴都带有线性类型范畴与笛卡尔类型范畴之间的一个伴随。

▶**定义 4.9**。 一个线性理解范畴由如下严格交换的函子图表给出。



我们假设p是一个纤维化，并且 E 具有逐纤维终对象和二元积。我们要求 χ 是全忠实的，并保持笛卡尔态射和终对象。

语法	线性理解范畴	可实现性模型
上下文	$\mathcal{B}$中的对象	$A_!$ 上的组装
笛卡尔类型	$\mathcal{E}$ 中的对象	$A_!$ 上的装配族
类型为 X 的项	π_X 的截面	π_X 的截面
线性类型	$\mathcal{L}$ 中的对象	A 上的装配族
线性项	$\mathcal{L}$ 中的态射	线性组装的态射

表 3 线性依赖类型论在线性理解范畴中的解释，特别是在我们的可实现性模型中的解释

我们假设q 是一个对称幺半闭纤维化，并且L ⊣ M 是一个逐纤维对称幺半伴随。最后，L 和M 需要保持笛卡尔态射。

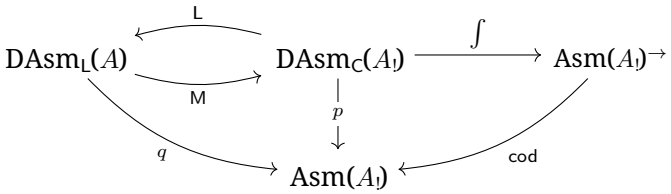
线性依赖类型论在线性理解范畴中的解释见表 3。最重要的一点是线性项的解释：它们被解释为 $\mathcal{L}$ 中的态射，而笛卡尔类型的项则被解释为投影的截面。

函子 L 和 M 解释了线性逻辑中的模态，而它们的复合给出线性指数化。这里需要注意的是，M 也作用于线性项，但仅限于不使用任何线性变量的线性项。从语义角度来看，这一限制也合乎情理。假设我们有某个类型 $x \in \mathrm{p}[\underline{x}]$。回想一下，类型为 x 的项也可以等价地表示为从终对象到 x 的态射。由于 M 是强幺半的，空线性上下文中的线性项对应于类型为 $\mathrm{M}(x)$ 的项。

构造线性理解范畴的例子时，使用以下对称幺半伴随的刻画会很有帮助。

▶ **命题 4.10** ([45, 引理 13])。伴随 $L \dashv R$ 对称幺半范畴 $\mathcal{C}$ 和 $\mathcal{D}$ 当且仅当 L 是一个强幺半函子。

▶ 例 4.11。函子 $\mathrm{M} : \mathrm{DAsm}_{\mathsf{L}}(A) \to \mathrm{DAsm}_{\mathsf{C}}(A_!)$ 将每个线性组装 (X, Y) 发送到组装族 Y_x，其中每个 $x \in X$ 对应一个组装。注意，这一定义是有意义的，因为 A 和 $A_!$ 具有相同的元素底层集合。接下来我们定义函子 $\mathrm{L} : \mathrm{DAsm}_{\mathsf{C}}(A_!) \to \mathrm{DAsm}_{\mathsf{L}}(A)$。假设对于每个 $x \in X$，我们都有一个组装族 Y_x。对于每个 $x \in X$，我们定义一个位于 A 上的组装 Y'_x，其底层集合为 Y_x，并且$a \Vdash y$ 当且仅当存在 $b \in A$ 使得 $b \Vdash y$ 且 $a = !b$。函子 L 将组装族 Y_x 映射为线性组装 Y'_x。由此得到如下线性理解范畴，其中笛卡尔侧是将笛卡尔组合代数 $A_!$ 代入例 4.6 后得到的。



M 和 L 都 是 忠实的。该示例将 Hoshino 的 [30] 线性-非线性模型推广到了依赖类型。

在考察线性可实现性模型中的类型构造子之前，我们先明确描述该模型中的（线性）类型和项。$\mathrm{Asm}(A_!)$ 中的对象和态射分别表示上下文

和代入。请注意，我们使用组合代数 $A_!$，因为线性类型以直觉主义方式依赖于其变量。$\mathsf{DAsm}_C(A_!)$ 中的对象为我们提供笛卡尔类型，而 $\mathsf{DAsm}_L(A)$ 中的对象为我们提供线性类型。虽然 $\mathsf{DAsm}_L(A)$ 和 $\mathsf{DAsm}_C(A_!)$ 都是纤维逐点幺半群范畴，但只有 $\mathsf{DAsm}_C(A_!)$ 能保证是笛卡尔幺半群。因此，通常而言，$\mathsf{DAsm}_L(A)$ 只支持线性逻辑。

4.3 线性理解范畴中的类型构造子

最后，我们考察线性理解范畴中的类型构造子，特别是线性可实现性模型中的类型构造子。我们从线性量词开始，并将其解释为代入的伴随。

▶**定义 4.12**。 如果对于每个 $s\colon x \to y$ *in* B，函子 $s^*\colon \mathsf{q}[y] \to \mathsf{q}[x]$ 都有一个满足通常 *Beck-Chevalley* 条件的右伴随，则称线性理解范畴支持 线性依赖积 。

如果对于每个 $s\colon x \to y$*in base*，函子 $s^*\colon \mathsf{q}[y] \to \mathsf{q}[x]$ 都有一个满足通常 *Beck-Chevalley* 条件的左伴随，则称线性理解范畴支持 线性依赖和 。 s^* 的左伴随记作 $\underline{\sqcap}_s$。我们还要求满足 *Frobenius* 互反律：若有一个态射 $s\colon x \to y$，以及对象 $x\colon \mathsf{q}[x]$ 和 $y\colon \mathsf{q}[y]$，则从 $\underline{\sqcap}_s(s^*(y) \otimes x)$ 到 $y \otimes \underline{\sqcap}_s x$ 的典范态射是一个同构。

▶ 例 4.13。例 4.11 中定义的线性可实现性模型支持线性依赖积和线性依赖和。我们只讨论线性依赖积的构造。假设有一个态射 $s\colon X \to X'$ in $\mathsf{Asm}(A_!)$，并且对每个 $x \in X$，都有一个位于 A上的组装 Y_x。令 $x' \in X'$，并令 f 为一个函数，它将每个 $x \in s^{-1}(x')$ 映射到一个元素 $f(x) \in Y_x$。如果 $a!b \Vdash f(x)$ whenever $b \Vdash x$，我们就称 f由 a跟踪。$\prod_s Y$ 的载体定义为所有此类 f 的集合，其中每个 f 都由某个 a跟踪；若 a 跟踪 f，我们就称 $a \Vdash f$。

接下来，我们来看等化子类型。我们使用纤维逐点极限来解释它们。

▶ **定义 4.14**。 设 $\mathsf{p}\colon E \to B$ 是一个纤维化。若每个纤维 $\mathsf{p}[x]$ 都有等化子，并且函子 $s^*\colon \mathsf{p}[y] \to \mathsf{p}[x]$ 保持等化子，则称 p 具有纤维逐点等化子。若纤维化 $\mathsf{q}\colon L \to B$ 具有纤维逐点等化子，则称线性理解范畴支持等化子类型 。

如果 s^* 有左伴随和右伴随，那么它保持极限和余极限，因此只需验证每个 $\mathsf{p}[x]$都有等化子。

▶ **例 4.15**。 例 4.11 中定义的线性理解范畴 $q[x]$ 支持等化子类型。纤维中的等化子是按通常方式构造的。

线性逻辑的加性联结词也以相同的方式处理。例如，加性合取通过纤维式二元积来解释。由于我们在后文中不会用到这些联结词，因此不作详细讨论。

我们通过证明该模型支持一个非直谓类型宇宙来完成我们的构造。在可实现性理论中，这类宇宙建立在温和集合概念之上，而关键定理是：温和集合等价于部分等价关系。注意，该定理对 BI-代数成立；BI-代数是带有 B组合子和 I组合子的应用结构。这种一般性对我们的构造十分重要，因为它使我们能够将其应用于可实现性模型中的线性类型和笛卡尔类型。

▶**定义 4.16**。设 X 是 *BI*-代数 A上的一个组装。若对于某个 a，当 $a \Vdash x$ 且 $a \Vdash y$ 时都有 $x = y$，则称 X 为温和的。部分等价关系（*PER*）是 A上的一种关系，它具有对称性和传递性。

▶ **定理 4.17** (Hyland [32])。设 A 是一个 *BI*-代数。关于 A的温和集合范畴与关于 A 的部分等价关系范畴等价。

请注意，定理4.17 此前已针对部分组合代数形式化 [14]。定理4.17的重要性在于，我们可以用较小的部分等价关系集来表示庞大的温和集合类。具体而言，每个温和集合都同构于由某个部分等价关系所诱导的温和集合。这一等价关系构成了我们可实现性模型中非谓词宇宙的基础。

▶ **例 4.18**。给定一个组装 $X \in \mathsf{Asm}(A_!)$，我们将其上的族 $\mathcal{U}$ 定义为关于 X 的部分等价关系集合。若 R 是一个偏等价关系，则 $a \Vdash R$ 总是成立。设 t是在上下文 X中类型为 $\mathcal{U}$ 的项。由于每个偏等价关系都会产生一个温和集合，t 在上下文 X中诱导出一个笛卡尔类型 $\mathsf{El}_{\mathsf{C}}(t)$，而线性类型 $\mathsf{El}_{\mathsf{L}}(t)$也以同样方式定义。

事实是 $\mathcal{U}$ 是一个非谓词宇宙，这一点可由两个观察得出。首先，注意到根据定理 4.17 ，每个温和集合都同构于一个由部分等价关系诱导的温和集合。其次，温和集合在依赖积和线性依赖积下都封闭。具体而言，若对于每个 $x : X$ 和 $y : \int Y$，我们都有装配 Y_x 和 Z_y，那么只要每个 Z_y 都是温和集合，它们的依赖积就是温和集合。线性依赖积也有类似的结论。

5 非谓词编码

最后，我们展示如何在线性依赖类型论中使用非谓述编码来构造初始代数。回顾一下，在带有非谓词宇宙$\mathcal{U}$的二阶依赖类型论中，可以通过归纳数据类型的递归原理 [25]对其进行编码。例如，若给定某个类型 A，我们将其上的列表类型 A 定义为

$$\prod_{X:\mathcal{U}} X \to (A \to X \to X) \to X$$

由于我们假设 $\mathcal{U}$ 是一个非谓词宇宙，因此这个类型属于 $\mathcal{U}$。可以证明，这个类型在定义上满足列表的递归原理和计算规则。

然而，这种非谓述编码未必会产生初始代数 [23, 55], ，这意味着无法为此类型证明合适的归纳原理。解决这一问题有多种方法，其中一种由 Awodey、Frey 和 Speight 提出 [9]。他们的想法是，对这种非谓述编码进行适当的精化，使其满足归纳原理。具体而言，可以将所需的初始代数构造为给定非谓述编码的一个子类型。这一想法可应用于各种情形。例如，它适用于高阶归纳类型，且对消去子没有截断限制 [54]。此外，还可以构造余归纳类型 [11]。

在本节余下部分，我们将 Awodey、Frey 和 Speight 的思路加以调整，以在线性依赖类型论中构造初始代数。我们关注线性类型范畴中的初始代数，因为 Awodey、Frey 和 Speight 的方法可以直接用于构造笛卡尔类型中的初始代数。在本节中，我们始终关注元素类型为某个闭线性类型 A 的列表类型。由于本节余下部分我们只在宇宙 $\mathcal{U}$ 内工作，因此使用通常的记法，而不使用 $\mathcal{U}$ 内部⌢记法，例如，我们写作 $\sqcap$ 而不是 $\sqcap$.

遵循 Girard 的思路 [25], 我们从以下编码开始。

$$\mathsf{List}^*(A) :\equiv \prod_{X:\mathcal{U}} \mathsf{El}_{\mathsf{L}}(X) \multimap !(A \multimap \mathsf{El}_{\mathsf{L}}(X) \multimap \mathsf{El}_{\mathsf{L}}(X)) \multimap \mathsf{El}_{\mathsf{L}}(X)$$

我们按如下方式定义构造子 $\mathsf{nil}^* : \mathsf{List}^*(A)$。

$$\mathsf{nil}^* :\equiv \lambda(X:\mathcal{U})(n:\mathsf{El}_{\mathsf{L}}(X))(c:!(A \multimap \mathsf{El}_{\mathsf{L}}(X) \multimap \mathsf{El}_{\mathsf{L}}(X))).\, n$$

如果我们有 $a : A$ 和 $l : \mathsf{List}^*(A)$，那么 $a ::^* l : \mathsf{List}^*(A)$ 定义如下。

$$a ::^* l :\equiv \lambda(X:\mathcal{U})(n:\mathsf{El}_{\mathsf{L}}(X))(c:!(A \multimap \mathsf{El}_{\mathsf{L}}(X) \multimap \mathsf{El}_{\mathsf{L}}(X))).\, c\, a\, (l\, X\, n\, c)$$

为了定义递归子，我们假设有一个类型 $X : \mathcal{U}$、一个元素 $n : \mathsf{El}_{\mathsf{L}}(X)$，以及一个运算 $c : !(A \multimap \mathsf{El}_{\mathsf{L}}(X) \multimap \mathsf{El}_{\mathsf{L}}(X))$，并将 $\mathsf{rec}_X(n, c, l) : \mathsf{El}_{\mathsf{L}}(X)$ 定义为

$$\mathsf{rec}_X(n, c, l) :\equiv l\, X\, n\, c$$

一般而言，类型 $\mathsf{List}^*(A)$ 只是一个弱初始代数，不一定是初始的 [23,55]。我们通过取一个适当的子类型来获得所需的初始代数，因此，我们首先定义代数及其态射。

$$\mathsf{Alg} :\equiv \sum_{X:\mathcal{U}} \mathsf{M}(\mathsf{El}_{\mathsf{L}}(X)) \times \mathsf{M}(!(A \multimap \mathsf{El}_{\mathsf{L}}(X) \multimap \mathsf{El}_{\mathsf{L}}(X)))$$

一个代数由一个类型 $X : \mathcal{U}$ 以及类型为 $\mathsf{M}(\mathsf{El}_{\mathsf{L}}(X))$ 和 $\mathsf{M}(!(A \multimap \mathsf{El}_{\mathsf{L}}(X) \multimap \mathsf{El}_{\mathsf{L}}(X)))$ 的笛卡尔项构成。这些笛卡尔项分别对应于空线性上下文中类型为 $\mathsf{El}_{\mathsf{L}}(X)$ 和 $!(A \multimap \mathsf{El}_{\mathsf{L}}(X) \multimap \mathsf{El}_{\mathsf{L}}(X))$ 的线性项。我们引入以下记号：如果我们有一个代数 $X : \mathsf{Alg}$，则将 $n_X : \mathsf{El}_{\mathsf{L}}(X)$ 和 $c_X : !(A \multimap \mathsf{El}_{\mathsf{L}}(X) \multimap \mathsf{El}_{\mathsf{L}}(X))$ 记作结构映射。

接下来我们定义代数之间的态射。给定 $f : \mathsf{El}_{\mathsf{L}}(X) \multimap \mathsf{El}_{\mathsf{L}}(Y)$，我们定义

$$\mathsf{IsMor}(X, Y, f) :\equiv (f\, n_X =_{\mathsf{L}} n_Y) \times \prod (a : A)(x : X).f(c_X\, a\, x) =_{\mathsf{L}} c_Y\, x\, (f\, y)$$

$$\mathsf{Mor}(X, Y) :\equiv \sum (f : \mathsf{M}(\mathsf{El}_{\mathsf{L}}(X) \multimap \mathsf{El}_{\mathsf{L}}(Y))).\mathsf{IsMor}((X, n_X, c_X), (Y, n_Y, c_Y), f)$$

每个笛卡尔项 $f : \mathsf{M}(\mathsf{El}_{\mathsf{L}}(X) \multimap \mathsf{El}_{\mathsf{L}}(Y))$ 都会在空上下文中导出一个类型为 $\mu(f) : \mathsf{El}_{\mathsf{L}}(X) \multimap \mathsf{El}_{\mathsf{L}}(Y)$ 的线性项（参见 M-e 规则）。为简化记号，我们省略 μ。因此，每个代数态射都会在空上下文中导出一个线性项 $f : \mathsf{El}_{\mathsf{L}}(X) \multimap \mathsf{El}_{\mathsf{L}}(Y)$。根据单射性和反射，预期的图表是交换的。

现在我们可以定义精化编码了。Awodey、Frey 和 Speight 使用等化子构造初始代数，由此得到 $\mathsf{List}^*(A)$的一个子类型。尽管他们使用$\sum$-类型来构造等化子，但我们不采用其线性类比，即 $\sqsubset$-types.类型 A 在 $\sqsubset_{x:A} B\, x$ 中必须是笛卡尔的，因此我们无法使用 $\sqsubset$-types 来构造线性类型的子类型。这就是我们在第 2 节中引入等化子的原因。

必要的态射定义如下。

$$\varphi, \psi : \mathsf{List}^*(A) \multimap \prod_{(X,n_X,c_X):\mathsf{Alg}} \prod_{(Y,n_Y,c_Y):\mathsf{Alg}} \prod_{(f,p,q):\mathsf{Mor}(X,Y)} \mathsf{El}_{\mathsf{L}}(Y)$$

$$\varphi :\equiv \lambda l\, X\, n_X\, c_X\, Y\, n_Y\, c_Y\, f\, p\, q.f(\mathsf{rec}_X(n_X, c_X, l))$$

$$\psi :\equiv \lambda l\, X\, n_X\, c_X\, Y\, n_Y\, c_Y\, f\, p\, q.\mathsf{rec}_Y(n_Y, c_Y, l)$$

请注意，在 φ 的定义中，态射 f 指的是空线性上下文中类型为 $\mathsf{El}_{\mathsf{L}}(X) \multimap \mathsf{El}_{\mathsf{L}}(Y)$ 的对应线性项。

我们将类型 $\mathsf{List}(A)$ 定义为等式 $\mathsf{Eq}(\varphi, \psi)$。现在我们证明 $\mathsf{List}(A)$ 是一个初始代数，并首先证明 $\mathsf{List}(A)$ 是一个代数。为此，我们证明 nil^* 和 $::^*$可在 $\mathsf{List}(A)$ 上定义运算，如下述引理所述。

▶ **引理 5.1**。 **</style}我们有 (**$\mathsf{nil}^*) = \psi(\mathsf{nil}^*)$。此外，对于所有$a : X$ 以及 l : 列表(A)，我们都有 $\varphi(a ::^* l) = \psi(a ::^* l)$。

证明。 为证明 $\varphi(\mathsf{nil}^*) = \psi(\mathsf{nil}^*)$，我们先注意到

$$\varphi(\mathsf{nil}^*) = f(\mathsf{rec}_X(n_X, c_X, \mathsf{nil}^*)) = f\ n_X$$
$$\psi(\mathsf{nil}^*) = \mathsf{rec}_Y(n_Y, c_Y, \mathsf{nil}^*) = n_Y$$

由于 f 是一个态射，根据反射和单射性，确实有 $f\ n_X = n_Y$。对于另一个陈述，我们假设

$$\varphi\ l = f(\mathsf{rec}_X(n_X, c_X, l)) = \mathsf{rec}_Y(n_Y, c_Y, l) = \psi\ l.$$

现在我们有

$$\varphi(a ::^* l) = f(\mathsf{rec}_X(n_X, c_X, a ::^* l)) = f(c_X\ a\ (\mathsf{rec}_X(n_X, c_X, l)))$$
$$\psi(a ::^* l) = \mathsf{rec}_Y(n_Y, c_Y, a ::^* l) = c_Y\ a\ (\mathsf{rec}_Y(n_Y, c_Y, l)) = c_Y\ a\ (f(\mathsf{rec}_X(n_X, c_X, l)))$$

根据单射性，可得 $\varphi(a ::^* l) =_{\mathsf{L}} \psi(a ::^* l)$，因为 f 是一个态射。 ◀

根据引理 5.1，我们有一个元素 nil : List(A)，以及一个函数，它将 a : A 和 l : List(A) 映射为 $a :: l$: List(A)。由此我们得到一个代数 List(A)，其载体为 List(A)，其代数映射由 nil 和 :: 给出。我们通过证明 List(A) 是初始的来结束本节。

▶**定理 5.2**。 代数 列表(A) 是初始的。

证明。 给定一个代数 X，我们定义 f 为 $\lambda(l :$ 列表$(A))$。$\mathsf{rec}_X(n_X, c_X, l)$。注意到

$$\mathsf{rec}_X(n_X, c_X, \pi_1\ \mathsf{nil}) = \mathsf{rec}_X(n_X, c_X, \mathsf{nil}^*) = n_X$$
$$\mathsf{rec}_X(n_X, c_X, \pi_1\ (a :: l)) = \mathsf{rec}_X(n_X, c_X, a ::^* l) = c_X\ a\ (\mathsf{rec}_X(n_X, c_X, l))$$

要证明映射 f是唯一的，只需证明对于每个l : 列表(A)，$\mathsf{rec}^{\mathsf{List}(A)}(\mathsf{nil}, ::, l) = l$。根据函数外延性，我们需要证明对于所有 X、 n_X 和 c_X，

$$\mathsf{rec}_{\mathsf{List}(A)}(\mathsf{nil}, ::, l)\ X\ n_X\ c_X = l\ X\ n_X\ c_X.$$

根据定义，我们有 $l\ X\ n_X\ c_X = \mathsf{rec}_X(n_X, c_X, l)$，并且

$$\mathsf{rec}_{\mathsf{List}(A)}(\mathsf{nil}, ::, l)\ X\ n_X\ c_X = \mathsf{rec}_X(n_X, c_X, \mathsf{rec}_{\mathsf{List}(A)}(\mathsf{nil}, ::, l)).$$

只需证明

$$\mathsf{rec}_X(n_X, c_X, \mathsf{rec}_{\mathsf{List}(A)}(\mathsf{nil}, ::, l)) = \mathsf{rec}_X(n_X, c_X, l).$$

由于 $\lambda l.\mathsf{rec}_X(n_X, c_X, l)$ 是一个态射，根据 l : 列表(A) 这一事实，定理得证。◀

6 相关工作

线性组合代数最初由 Abramsky 在一系列讲座中提出[1]；首篇正式发表的论述见 [2]，，其中列举了许多实例。首个线性可实现性模型由 Abramsky 和 Lenisa 提出 [3]，，他们考察了基于线性组合代数的部分等价关系范畴，以获得各种类型 λ演算的完全完备性结果。Hoshino [30] 首次深入研究了基于（关系型）线性组合代数的装配和谦逊集。此后，人们对

“次结构”组合代数及其相关范畴结构的研究日益受到关注，例如：带状组合代数 [28, 29] 和平面组合代数 [64, 65, 66]。Kuzmin、Nester、Reimaa 和 Speight [38]利用结构化多范畴提出了一套相当通用的组合完备性理论。近年来，单子组合代数也被用于效应可实现性研究 [15, 16, 17]。

文献中对线性依赖类型论有多种处理方式。最早由 Cervesato 和 Pfenning 提出了线性依赖类型论的框架 [12, 13], ，但他们只考虑了所需类型构造子的一部分 [13]。Krishnaswami、Pradic 和 Benton 提出了一种线性依赖类型论，其中包含线性逻辑的所有联结词 [37]。Vákár [68]和 Lundfall [40], 也提出了类似的处理方式，但存在一些差异。与 Krishnaswami、Pradic 和 Benton 的处理方式相比，Lundfall 为恒等类型和模态给出了不同的规则，而 Vákár 对 $\sum$类型的处理略有不同。我们采用了与 Krishnaswami、Pradic 和 Benton 类似的语法。主要差异见第2节：我们的语法包含等化子类型、模态M的单射公理，以及一个非直谓类型宇宙。

Lundfall 和 [40] Vákár [68] 都为线性依赖类型论提出了范畴模型的概念。Lundfall 的处理基于理解范畴，而 Vákár 则使用指标范畴。此外，Lundfall 通过伴随来表示线性逻辑的模态，而 Vákár 使用余单子。我们的语义基于 Lundfall 的线性理解范畴概念。Lundfall 和 Vákár 都没有考虑可实现性模型。Krishnaswami、Pradic 和 Benton [37] 使用部分等价关系为他们的演算提供语义，但没有对可实现性模型进行一般性的发展。

定量类型理论 [8, 44, 48] 是线性依赖类型论的一种推广，它通过允许变量以某个半环所指示的数量出现，来细化线性条件。Atkey [8] 使用定量组合代数构造了一个定量类型理论模型。由此得到的模型与我们的模型颇为相似，关键区别在于我们的模型是非谓词的。由于 Atkey 将类型解释为由集合指标化的组装，而不是组装，因此该模型中不可能存在非谓词宇宙 [50]。因此，第5 节中的非谓述编码只能在我们的模型中得到解释。

我们在第5 节采用的非谓词编码细化方法，基于 Awodey、Frey 和 Speight 的工作 [9], ，该工作研究（高阶）归纳类型。消去子的截断限制在 [54]中被取消。该方法也已扩展到余归纳类型 [11]。修正非谓词编码的另一种方法是借助参数多态性。通过假设内部参数性，可以证明非谓述编码的初始性 [7, 71, 72]。类型论的 PER 模型验证了这类公理 [36]。我们可以在高阶分离逻辑中使用非谓词编码来定义归纳谓词 [35]。由于这些编码使用证明无关谓词，因此无需对其进行细化。

7 结论

本文定义了一个线性依赖类型论的可实现性模型，其中笛卡尔类型和线性类型都具有一个非谓词宇宙，且均由线性组合代数参数化。我们还展示了 Awodey、Frey 和 Speight 的方法 [9] 如何应用于线性类型的非谓词编码，从而构造出线性列表的初始代数。

有多种方式可以进一步发展这项工作。一种方式是通过考虑任意归纳类型，而不只是列表，进一步推广第5 节中的论述。我们推测可以使用相同的思路。此外，本文始终使用外延性，我们的单射公理也是为这种情形量身定制的。一个问题是如何

16 线性依赖类型理论中的非谓词性

将我们的工作推广到线性依赖类型论的内涵版本。

参考文献

1 Samson Abramsky。交互、组合子与复杂性。讲义，意大利锡耶纳，1997年。未发表手稿。2 Samson Abramsky、Esfandiar Haghverdi 和 Philip J. Scott。交互几何与线性组合代数。《计算机科学中的数学结构》，12(5):625–665，2002年。 doi:10.1017/S0960129502003730。 3 Samson Abramsky 和 Marina Lenisa。带类型λ演算的线性可实现性与完全完备性。《纯粹与应用逻辑年刊》，134(2-3):122–168，2005年。 doi:10.1016/j.apal.2004.08.003。 4 Benedikt Ahrens、Krzysztof Kapulkin 和 Michael Shulman。单价范畴与Rezk完备化。《计算机科学中的数学结构》，25:1010–1039，2015年。doi:10.1017/S0960129514000486。 5 Benedikt Ahrens 和 Peter LeFanu Lumsdaine。显示范畴。《计算机科学中的逻辑方法》，15(1)，2019 年。 doi:10.23638/LMCS-15(1:20)2019。 6 Benedikt Ahrens、Ralph Matthes、Niels van der Weide 和 Kobe Wullaert。线性逻辑语义的显示幺半范畴。载于 Dmitriy Traytel、Brigitte Pientka 和 Sandrine Blazy（编），第13届ACM SIGPLAN经认证的程序与证明国际会议论文集（CPP 2024），英国伦敦，2024年1月15日至16日，第260–273页。ACM，2024年。 doi:10.1145/3636501.3636956。 7 Thorsten Altenkirch、Yorgo Chamoun、Ambrus Kaposi 和 Michael Shulman。无区间的内部参数性。《ACM程序设计语言汇刊》，8(POPL):2340–2369，2024年。 doi:10.1145/3632920。 8 Robert Atkey。定量类型理论的语法与语义。载于 Anuj Dawar 和 Erich Grädel（编），第33届ACM/IEEE计算机科学逻辑研讨会论文集（LICS 2018），英国牛津，2018年7月9日至12日，第56–65页。ACM，2018年。doi:10.1145/3209108.3209189。 9 Steve Awodey、Jonas Frey 和 Sam Speight。（高阶）归纳类型的非谓词编码。载于 Anuj Dawar 和 Erich Grädel（编），第33届ACM/IEEE计算机科学逻辑研讨会论文集（LICS 2018），英国牛津，2018年7月9日至12日，第76–85页。ACM，2018年。 doi:10.1145/3209108.3209130。 10 Nick Benton。一种混合线性与非线性逻辑：证明、项与模型。载于 Leszek Pachol-ski 和 J erzy Tiuryn（编），第8届计算机科学逻辑国际研讨会论文集（CSL'94），《计算机科学讲义》第933卷，第121–135页。施普林格，1995年。 doi:10.1007/3-540-58277-0_6。 11 Steven Bronsveld、Herman Geuvers 和 Niels van der Weide。归纳类型与余归纳类型的非谓词编码。载于 Maribel Fernández（编），第10届计算与演绎形式结构国际会议（FSCD 2025），《莱布尼茨国际信息学会议录》（Lipics）第337卷，第11:1–11:22页，德国达格施图尔，2025年。Dagstuhl城堡—莱布尼茨信息学中心。 doi:10.4230/LIPIcs.FSCD.2025.11。 12 Iliano Cervesato 和 Frank Pfenning。一种线性逻辑框架。载于第11届IEEE计算机科学逻辑年度研讨会论文集（LICS '96），第264–275页。IEEE计算机学会出版社，1996年。扩展摘要；修订版以期刊论文形式发表。网址：https://doi.org/10.1006/inco.2001.2951。 13 Iliano Cervesato 和 Frank Pfenning。一种线性逻辑框架。《信息与计算》，179:19–75，2002年。 doi:10.1006/inco.2001.2951。 14 Rahul Chhabra。温和集合等价于偏等价关系。arXiv预印本 arXiv:2411.08421，2024年。 arXiv:2411.08421。 15 Liron Cohen。计算优先：以效应重构构造主义。载于 Maribel Fernández（编），第10届计算与演绎形式结构国际会议（FSCD 2025），《莱布尼茨国际信息学会议录》（Lipics）第337卷，第

1:1–1:20页，德国达格施图尔，2025年。Dagstuhl城堡—莱布尼茨信息学中心。doi:10.4230/LIPIcs.FSCD.2025.1。 16 Liron Cohen、Sofia Abreu Faro 和 Ross Tate。效应对构造主义的影响。载于 Barbara König（编），第三十五届程序语义数学基础会议论文集（MFPS 2019），英国伦敦，2019年6月4日至7日，《理论计算机科学电子札记》第347卷，第87–120页。Elsevier，2019年。 doi10.1016/J.ENTCS.2019.09.006。 17 Liron Cohen、Ariel Grunfeld、Dominik Kirst 和 Étienne Miquey。从部分到单子式：带效应的组合代数。载于 Maribel Fernández（编），第10届计算与演绎形式结构国际会议（FSCD 2025），《莱布尼茨国际信息学会议录》（Lipics）第337卷，第14:1–14:22页，德国达格施图尔，2025年。Dagstuhl城堡—莱布尼茨信息学中心。 doi:10.4230/LIPIcs.FSCD.2025.14。 18 Thierry Coquand 和 Gérard Huet。构造演算。《信息与计算》，76(2-3):95–120，1988年。 doi:10.1016/0890-5401(88)90005-3。 19 Greta Coraglia 和 Jacopo Emmenegger。子类型的范畴模型。载于 Delia Kesner、Eduardo Hermo Reyes 和 Benno van den Berg（编），第29届证明与程序类型国际会议（TYPES 2023），《莱布尼茨国际信息学会议录》（Lipics）第303卷，第3:1–3:19页，德国达格施图尔，2024年。Dagstuhl城堡—莱布尼茨信息学中心。 doi:10.4230/LIPIcs.TYPES.2023.3。 20 Haskell B. Curry。组合逻辑基础。《美国数学杂志》，52(3):509–536，1930年。 doi:10.2307/2370619。 21 Leonardo Mendonça de Moura、Soonho Kong、Jeremy Avigad、Floris van Doorn 和 Jakob von Raumer。Lean定理证明器（系统描述）。载于 Amy P. Felty 和 Aart Middeldorp（编），自动演绎——CADE-25——第25届自动演绎国际会议，德国柏林，2015年8月1日至7日，论文集，《计算机科学讲义》第9195卷，第378–388页。施普林格，2015年。 doi:10.1007/978-3-319-21401-6_26。 22 Peng Fu、Kohei Kishida 和 Peter Selinger。量子编程语言的线性依赖类型论。《计算机科学中的逻辑方法》，18(3):第28号论文，44页，2022年。doi:10.46298/lmcs-18(3:28)2022。 23 Herman Geuvers。二阶依赖类型论中不可导出归纳。载于 Samson Abramsky（编），《类型化Lambda演算及其应用》，第5届类型化Lambda演算及其应用国际会议（TLCA 2001），波兰克拉科夫，2001年5月2日至5日，论文集，《计算机科学讲义》第2044卷，第166–181页。施普林格，2001年。 doi:10.1007/3-540-45413-6_16。 24 Jean-Yves Girard。高阶算术的函数解释与切消去。博士论文，巴黎第七大学，法国巴黎，1972年。 25 Jean-Yves Girard 、 Yves Lafont 和 Paul Taylor。《证明与类型》。《剑桥理论计算机科学丛书》第7卷。剑桥大学出版社，剑桥，1989年。 26 Alexander Grothendieck。代数几何中的下降技巧与存在定理。I. 概论。通过忠实平坦态射下降。载于《Bourbaki研讨班》第5卷，第190号报告，第299–327页。法国数学学会，巴黎，1960年。 27 Esfandiar Haghverdi。线性逻辑、证明几何与完全完备性的范畴方法。博士论文，渥太华大学，2000年。 28 Masahito Hasegawa。一种编织Lambda演算。《理论计算机科学电子论文集》，353:94–108，2021年。 doi:10.4204/EPTCS.353.5。 29 Masahito Hasegawa 和 Serge Lechenne。组合代数中的编织、扭结、迹与对偶性。载于第39届ACM/IEEE计算机科学逻辑研讨会论文集（LICS '24），第42号文章，第1–14页，2024年。 doi:10.1145/3661814.3662098。 30 Naohiko Hoshino。线性可实现性。载于 Jacques Duparc 和 Thomas A. Henzinger（编），《计算机科学逻辑》，第21届国际研讨会、EACSL第16届年度会议（CSL 2007），瑞士洛桑，2007年9月11日至15日，论文集，《计算机科学讲义》第4646卷，第420–434页。施普林格，2007年。 doi:10.1007/978-3-540-74915-8_32。

31 J. M. E. Hyland. 有效拓扑斯。载于 A. S. Troelstra 和 D. van Dalen（编），《L. E. J. Brouwer 百年纪念研讨会》，《逻辑与数学基础研究》第110卷，第165–216页。North-Holland，阿姆斯特丹，1982年。 doi:10.1016/S0049-237X(09)70129-6。32 Martin Hyland. 小完备范畴。《纯粹与应用逻辑年刊》，40(2):135–165，1988年。 doi:10.1016/0168-0072(88)90018-8。33 Bart Jacobs.《范畴逻辑与类型论》，《逻辑与数学基础研究》第141卷。North-Holland，2001年。34 Stephen Cole Kleene. 论直觉主义数论的解释。《符号逻辑杂志》，10:109–124，1945年。 doi:10.2307/2269016。35 Robbert Krebbers、Luko van der Maas 和 Enrico Tassi. 通过高阶分离逻辑中的最小不动点定义归纳谓词。载于 Yannick Forster 和 Chantal Keller（编），第16届交互式定理证明国际会议（ITP 2025），冰岛雷克雅未克，2025年9月28日至10月1日，《LIPIcs》第352卷，第27:1–27:21页。Dagstuhl城堡—莱布尼茨信息学中心，2025年。 doi:10.4230/LIPICS.ITP.2025.27。36 Neelakantan R. Krishnaswami 和 Derek Dreyer. 在外延构造演算中内化关系参数多态性。载于 Simona Ronchi Della Rocca（编），《计算机科学逻辑2013》（CSL 2013），CSL 2013，意大利都灵，2013年9月2日至5日，《LIPIcs》第23卷，第432–451页。Dagstuhl城堡—莱布尼茨信息学中心，2013年。 doi:10.4230/LIPICS.CSL.2013.432。37 Neelakantan R. Krishnaswami、Cécilia Pradic 和 Nick Benton. 整合线性类型与依赖类型。载于 Sriram K. Rajamani 和 David Walker（编），第42届 ACM SIGPLAN-SIGACT 编程语言原理年会论文集（POPL 2015），印度孟买，2015年1月15日至17日，第17–30页。ACM，2015年。 doi:10.1145/2676726.2676969。38 Ivan Kuzmin、Chad Nester、Ülo Reimaa 和 Sam Speight. 结构化多范畴中的组合完备性。arXiv预印本 arXiv:2511.17152，2025年。 arXiv:2511.17152。39 F. William Lawvere. 基础中的伴随性。《范畴理论与应用重印集》，第1–16页，2006年。40 Martin Lundfall. 线性依赖类型论模型。硕士论文，斯德哥尔摩大学，2017年。41 Martin Lundfall. 线性依赖类型论的图表模型。arXiv 电子预印本，arXiv:1806.09593，2018年。URL: https://arxiv.org/abs/1806.09593, doi:10.48550/arXiv.1806.09593。42 Zhaohui Luo.《扩展构造演算》。博士论文，爱丁堡大学，1990年。43 Andrew Marmaduke、Larry Diehl 和 Aaron Stump. Cedille 中归纳—归纳数据的非谓词编码。载于 Stephen Chang（编），《函数式编程趋势——第24届国际专题讨论会》（TFP 2023），美国马萨诸塞州波士顿，2023年1月13日至15日，修订精选论文，《计算机科学讲义》第13868卷，第1–15页。施普林格，2023年。 doi:10.1007/978-3-031-38938-2_1。44 Conor McBride. 我啥也不缺。载于 Sam Lindley、Conor McBride、Philip W. Trinder 和 Donald Sannella（编），《改变世界的成功事例集——献给 Philip Wadler 60岁生日的论文集》，《计算机科学讲义》第9600卷，第207–233页。施普林格，2016年。 doi:10.1007/978-3-319-30936-1_12。45 Paul-André Melliès. 重新审视线性逻辑的范畴模型。未发表，2003年10月。URL： https://hal.science/hal-00154229。46 Joe Moeller 和 Christina Vasilakopoulou. 幺半格罗滕迪克构造。《范畴理论与应用》，35：论文第31号，1159–1207页，2020年。47 Niyousha Najmaei、Niels van der Weide、Benedikt Ahrens 和 Paige Randall North. 从语义到语法：理解范畴的类型论。《ACM 程序设计语言论文集》，10(POPL)，2026年1月。 doi:10.1145/3776725。

48 Georgi Nakov 和 Fredrik Nordvall Forsberg. 定量多项式函子。载于 Henning Basold、Jesper Cockx 和 Silvia Ghilezan（编），第27届证明与程序类型国际会议（TYPES 2021），荷兰莱顿（线上会议），2021年6月14日至18日，《LIPIcs》第239卷，第10:1–10:22页。Dagstuhl城堡—莱布尼茨信息学中心，2021年。 doi:10.4230/LIPICS.TYPES.2021.10。49 Christine Paulin-Mohring. 从构造演算中的证明提取 F(omega) 程序。载于第十六届 ACM 编程语言原理年会论文集，美国得克萨斯州奥斯汀，1989年1月11日至13日，第89–104页。ACM出版社，1989年。 doi:10.1145/75277.75285。50 Wesley Phoa.《纤维化、托扑斯理论、有效拓扑斯与温和集合导论》。LFCS，爱丁堡大学计算机科学系，1992年。51 Bernhard Reus. 类型论的可实现性模型。载于 Lars Birkedal、Jaap van Oosten、Giuseppe Rosolini 和 Dana S. Scott（编），与 FLoC'99（1999年逻辑联合会议）相关的实现性语义与应用教程研讨会，意大利特伦托，1999年6月30日至7月1日，《理论计算机科学电子札记》第23卷，第128–158页。Elsevier，1999年。 doi:10.1016/S1571-0661(04)00108-2。52 John C. Reynolds. 迈向类型结构理论。载于 Bernard Robinet（编），《编程专题讨论会论文集》，《编程研讨会》，《计算机科学讲义》第19卷，第408–425页，柏林／海德堡，1974年。施普林格出版社。 doi:10.1007/3-540-06859-7_148。53 John C. Reynolds. 多态性不是集合论的。载于 Gilles Kahn、David B. MacQueen 和 Gordon D. Plotkin（编），《数据类型语义》，国际专题讨论会论文集，法国索菲亚安提波利斯，1984年6月27日至29日，《计算机科学讲义》第173卷，第145–156页。施普林格，1984年。 doi:10.1007/3-540-13346-1_7。54 Xavier Ripoll Echeveste.《归纳类型的替代性非谓述编码》。硕士论文，阿姆斯特丹大学，2023年。55 Ivar Rummelhoff. PER模型中的 Polynat。《理论计算机科学》，316(1-3):215–224，2004年。 doi:10.1016/j.tcs.2004.01.031。56 M. Schönfinkel. 论数理逻辑的构件。《数学年刊》，92:305–316，1924年。 doi:10.1007/BF01448013。57 Robert A. G. Seely. 超教义、自然演绎与贝克条件。《数理逻辑与数学基础杂志》，29(6):505–542，1983年。 doi:10.1002/malq.19830291005。58 Michael Shulman. 框架双范畴与幺半纤维化。《范畴理论与应用》，20：第18号，650–738页，2008年。59 Alex Simpson. 线性λ演算中的归约及其在操作语义中的应用。载于第16届项重写与应用国际会议论文集（RTA 2005），《计算机科学讲义》第3467卷，第219–234页，日本奈良，2005年4月。施普林格。 doi:10.1007/978-3-540-32033-3_17。60 Sam Speight 和 Niels van der Weide. 线性可实现性，2026年1月。 doi:10.5281/zenodo.18431109。61 Thomas Streicher.《类型论语义——正确性、完备性与独立性结果》。理论计算机科学进展。比克豪泽，1991年。62 Rocq开发团队。《Rocq证明器》。Zenodo，2025年4月。 doi:10.5281/zenodo.15149629。63 Peter Thiemann 和 Vasco T. Vasconcelos. 标签依赖会话类型。《ACM 程序设计语言论文集》，4(POPL):67:1–67:29，2020年。 doi:10.1145/3371135。64 Hiroshi Tomita. 无对称性的可实现性。载于 Christel Baier 和 Jean Goubault-Larrecq（编），第29届EACSL计算机科学逻辑年会（CSL 2021），《莱布尼茨国际信息学会议录（LIPIcs）》第183卷，第38:1–38:16页，德国达格施图尔，2021年。Dagstuhl城堡—莱布尼茨信息学中心。 doi:10.4230/LIPIcs.CSL.2021.38。

65 Hiroshi Tomita. 通过左右应用实现平面可实现性。载于 Florin Manea 和 Alex Simpson（编），第30届EACSL计算机科学逻辑年会（CSL 2022），莱布尼茨国际信息学会议录（LIPIcs）第216卷，第35:1– 35:17页，德国达格施图尔，2022年。Dagstuhl城堡—莱布尼茨信息学中心。 doi:10.4230/LIPIcs.CSL.2022.35。66 Hiroshi Tomita. 非对称闭结构的范畴可实现性。arXiv预印本 arXiv:2307.04119，2023年。 arXiv:2307.04119。67 Mathijs Vákár. 探寻效应依赖类型。哲学博士论文，牛津大学，英国牛津，2017年。网址： https://ora.ox.ac.uk/objects/uuid:e91e19b3-7e10-4fda-9433-f23b469e4049。68 Matthijs Vákár. 线性逻辑框架的范畴语义。载于 Andrew M. Pitts（编），《软件科学与计算结构基础——第18届国际会议 FoSSaCS 2015》，作为欧洲软件理论与实践联合会议 ETAPS 2015 的一部分举行，英国伦敦，2015年4月11日至18日。会议录，计算机科学讲义第9034卷，第102–116页。施普林格，2015年。 doi:10.1007/978-3-662-46678-0 7.69 Jaap van Oosten. 可实现性：范畴方面导论，逻辑与数学基础研究第152卷。爱思唯尔有限公司，阿姆斯特丹，2008年。70 Vladimir Voevodsky、Benedikt Ahrens、Daniel Grayson 等。UniMath——单价数学的计算机验证库。 doi:10.5281/zenodo.10849216。71 Philip Wadler. Girard-Reynolds同构。信息与计算，186(2)：260– 284，2003年。 doi:10.1016/S0890-5401(03)00141-X。72 Philip Wadler. Girard-Reynolds同构（第二版）。理论计算机科学，375(1-3)：201–226，2007年。 doi:10.1016/J.TCS.2006.12.042。73 Kobe Wullaert、Ralph Matthes 和 Benedikt Ahrens. 单价幺半范畴。载于 Delia Kesner 和 Pierre-Marie Pédrot（编），第28届证明与程序类型国际会议（TYPES 2022），2022年6月20日至25日，法国南特 LS2N 南特大学，第269卷 LIPIcs，第15:1–15:21页。Dagstuhl城堡—莱布尼茨信息学中心，2022年。 doi:10.4230/LIPICS.TYPES.2022.15。