# 多态依赖类型论中的不可推导性结果

Herman Geuvers
拉德堡德大学 埃因霍温理工大学 奈梅亨，荷兰 荷兰

计算与信息科学研究所 理学院
herman@cs.ru.nl

在纯构造演算（CC）中，可以定义数据类型及其上的函数，并且有强大的高阶逻辑可用于推理这些函数和数据类型 [10, 5]。这得益于非直谓性与依赖类型的结合，而这些特性大多已可在多态（二阶）依赖类型论 λP2 中观察到。数据类型的非直谓编码（在 λP2 或 CC 中）虽然强大，却并非完全令人满意：例如，归纳原理无法证明。事实上，可以证明，无论采用何种可能的数据类型表示，归纳都无法证明[12]。在 [1], Awodey、Frey 和 Speight 的研究中，他们表明，在 λP2 扩展加入 Σ-类型、具有恒等证明唯一性的恒等类型以及函数外延性后，可以定义满足可证明归纳原理的数据类型。在 [8] 中已有研究表明，在 λP2 的这一扩展中，也可以定义具有恰当归纳原理的商类型，并利用商类型定义具有恰当余归纳原理的余归纳类型。

这留下了若干尚待解答的问题：带有归纳原理的商类型是否无法在原始的 λP2 中定义？余归纳类型又如何？在 λP2 中是否不可能得到强余归纳原理？反过来看： [1] 中使用的扩展，究竟哪些是实现归纳和余归纳所必需的?

本文对这些问题给出了部分答案：参数化商类型无法在 λP2 中定义，而众所周知可定义的流类型并不具备余归纳原理。对于后一问题，我们证明，只要将 λP2 扩展为加入 Σ-类型和满足恒等证明唯一性的恒等类型，仍然无法证明自然数的归纳原理。因此，函数外延性对于使归纳原理可证至关重要。

我们通过研究 λP2 的模型来证明这些结果：在这些模型中，表示上述原理的类型是空的，因此这些模型构成了这些原理不可导出的反模型。关于多态类型论，许多经过深入研究的模型都是参数化的，或者无论如何都具有相当强的外延性[2, 21, 14], ，因此在这些模型中，归纳和余归纳都成立。一个有趣的例外是 Berardi 及其合作者的研究 [6, 3, 7], ，他们描述了（非依赖）多态类型论的非参数模型。我们在此研究的模型具有更强的句法性质，且已在 [12, 11] 中得到描述和研究。它们可以看作一组项模型，通过特定的选择可以构造出反模型。[1]

## 1 引言

本文谨献给 Stefano Berardi，以庆祝他的第 $40^2$ 个生日，或许更是为了庆祝他在逻辑与类型论方面做出的众多科学贡献。我遇见 Stefano

[1]致谢：感谢审稿人提出有益且建设性的意见。[2]在适当的数制中

T. Altenkirch、F. Barbanera、F. Damiani 和 U. de'Liguoro（编）：《逻辑与类型论：献给 Stefano Berardi 的论文集》，纪念他 64 岁生日，EPTCS441，2026 年，第 148–165 页，
doi:10.4204/EPTCS.441.9

© H. Geuvers 本作品采用知识共享署名许可协议授权。

是在 1988 年秋天第一次，在比萨，他从都灵前来访问。我当时正在比萨度过秋季学期，Stefano 在蒂雷尼亚与我们同住了好几个晚上。我立刻被他广博的学识、深刻的洞见和独到的想法所打动，此后一直如此。除此之外，Stefano 是一位非常友善的人，慷慨地分享自己的知识，是一位真正的科学家。非常感谢，也祝贺你，Stefano！

本文的内容源自 Stefano 在攻读博士学位期间及其后不久（例如，参见 [5]）所做的研究：探讨如何在构造演算中表示数据类型、函数以及关于它们的推理原理。当时人们已经注意到（这在当时也算是众所周知的说法），由于归纳原理不可推导，因此必须将其加入。后来，我正式证明了，无论采用何种巧妙的数据类型编码，归纳都不可推导 [12]。本文沿着这一研究脉络继续展开，证明其他各种推理原理也不可推导。为简化工作，我们将研究范围限定在二阶依赖类型论，而非完整的构造演算；不过，我们相信这些结果可以直接推广。

在二阶依赖类型论中，λP2，我们可以编码各种归纳数据类型，例如自然数、列表、树等类型（一般性介绍见 [13]），由此可将众所周知的多态丘奇数作为自然数的解释。例如，这种编码在非依赖二阶类型论（众所周知的多态 λ-演算λ2）中已经可行；但依赖类型还有一项额外优势：我们也可以陈述归纳数据类型的归纳原理。比如，若 nat 是众所周知的多态丘奇数类型，其零为 O，后继函数为 succ[3], ，那么归纳原理就由如下定义的类型 ind 表示

$$\mathsf{ind} := \Pi P : \mathsf{nat} \to \star.\ P\ \mathsf{O} \to (\Pi y : \mathsf{nat}.\ P\ y \to P\ (\mathsf{succ}\ y)) \to \Pi x : \mathsf{nat}.\ P\ x.$$

这里，$\star$ 表示所有类型的“种类”（宇宙），它既涵盖集合（nat : $\star$），也涵盖命题（ind : $\star$）。如果存在一个类型为 ind 的闭项，就称 nat 的归纳原理可在 λP2 中导出。

众所周知，nat 的归纳原理无法在 λP2 中导出。针对 nat 的证明见 [24], ，该证明使用可实现性语义；另一个使用 PER模型 的证明见 [22] 。不过，正如我们在 [12] 中所证明的，情况更为一般：若 N 是一个类型，且 0 : N、s : N → N，则不存在类型为 ΠP : N → ⋆. P O → (Πy : N. P y → P (succ y)) → Πx : N. P x 的（闭）项。因此，在 λP2 中不存在将自然数编码为初始代数的巧妙方法。一种显而易见的巧妙编码方式，是通过相对化到“归纳自然数”来实现，将归纳原理纳入自然数的概念之中。接着，我们将如下定义 N。

$$\begin{aligned} \mathsf{Ind}\ x &:= \Pi P : \mathsf{nat} \to \star.\ P\ \mathsf{O} \to (\Pi y : \mathsf{nat}.\ P\ y \to P\ (\mathsf{succ}\ y)) \to P\ x, \\ N &:= \exists x : \mathsf{nat}.\ \mathsf{Ind}\ x. \end{aligned}$$

这里，∃采用众所周知的二阶方式定义：∃x : σ.τ:= Πα : ⋆.(Πx : σ. τ → α) → α。利用可定义的 ∃-消去和 ∃-引入规则，现在很容易为这种编码定义 O 和后继函数：

_ O := λα: ⋆. λh : Πx : 自然数. 归纳 x → α. h O $q_O$, 后继

函数 := λn : N. n N (λx : 自然数. λp : 归纳 x.λα: ⋆. λh : Π y : 自然数. 归纳 y → α). h (后继函数 x) ($q_{\mathsf{succ}}$ x p),

[3]提醒一下：自然数 := Πα : ⋆. α → (α → α) → α, 后继函数 := λy : 自然数. λα : ⋆. λx : α. λf : α → α. f (y α x f) and O := λα: ⋆. λx : α. λf : α → α. x

其中 $q_O$ 和 $q_{succ}$ 是这样的项：$q_O$ : 归纳 O，且 $q_{succ}$ : Πx : 自然数. 归纳 x → 归纳 (后继函数 x)。但同样，对于这些自然数，在λP2中也无法证明归纳，这一点由 [12] 证明。

在 [1], 中，Awodey、Frey 和 Speight 展示了：在 λP2 的一个适度扩展中，借鉴了同伦类型论中的概念，并采用上述“相对化”定义时，可以证明自然数的归纳原理，其中使用的是强 Σ-类型，而非可定义的（较弱的） ∃-类型。事实上， [1] 通过一个称为 Lim 的不同谓词实现相对化，它直接翻译了初始代数这一概念：Lim 筛选出类型为 N、且能使 N 成为满足初始代数性质的类型的那些项。在 [8] 中，我们展示了，通过翻译范畴初始性这一性质，可以用多种方式进行相对化。能够证明归纳原理的 [1] 类型论，是在 λP2 的基础上扩展以下内容所得：

1. 恒等类型，2. 恒等证明唯一性（UIP），3. 函数外延性（FunExt），4. Σ-类型。

在 [8] 中，我们已经证明，在 λP2 的这一扩展中，也可以将商类型（作为初始代数）连同适当的归纳原理一并定义；利用这些商类型，还可以定义带有适当余归纳原理的余归纳类型。这也引出了一些问题：商类型难道不能直接在 λP2 本身中定义吗？那么余归纳类型呢？可定义的余归纳类型（例如流）难道不已经是终结余代数了吗？如果不是，是否能在 λP2 中以某种“巧妙”的方式编码终结余代数？另一方面，我们也不禁要问，为了获得归纳， λP2 的哪些扩展才是真正必需的：函数外延性真的必不可少吗？ Σ-类型真的必不可少吗？

### 1.1 主要结果概述

本文针对这些问题给出了一些初步解答，主要通过句法分析和构造反模型来实现：我们构造了一个 λP2 的模型，其中不存在参数化商类型，也就是说，不存在商运算为多态的商类型。我们还展示了一个模型，在其中，众所周知可多态定义的流数据类型并非终结余代数。最后，我们研究了为 λP2 加入 Σ-类型和恒等类型（如 [1]）所得的扩展，并给出了该扩展的一个模型：在这个模型中，UIP 成立，但自然数归纳并不成立。

我们构造的模型具有句法性质，是我们在[12] 中所考虑的那些模型的变体，用以表明归纳在“普通的 λP2”中不可证。就此而言，这项工作可以看作沿袭 Stefano Berardi 的风格，他也研究了更具句法性的模型，例如并非参数化的多态 λ演算模型 [6, 3, 7], ，这些模型基于无类型 λ 演算的模型。我们的模型在精神上与之相似，但基于任意的弱外延组合代数 WECA。WECA 的典型例子是（开）无类型 λ-项在模 β或 βη意义下的集合，因此我们会大量使用它。

## 2 二阶依赖类型论

二阶依赖类型论 λP2 扩展了多态 λ-演算（λ2 或系统F），加入了依赖类型；作为这样的系统，它最早由 [17] 引入。它是构造演算（[10], [9]）的一个子系统，其中形成类型构造子的操作被限制为二阶类型构造子。（因此，存在种类为 $\sigma \to \star$ 的类型构造子，但不存在种类为（$\sigma \to \star$）$\to \star$ 的类型构造子。）它也可以看作一阶系统 λP 的扩展，即加入了对类型构造子的量化。关于这些系统及其相互关系的详细讨论，请参阅 [4] 或 [19]。这里我们只定义系统 λP2，并对其作一些初步说明。

定义 2.1。类型系统 λP2 的伪项集合 T 定义如下:

$$\mathsf{T} ::= \star \mid \mathsf{Kind} \mid \mathsf{Var} \mid (\Pi \mathsf{Var} : \mathsf{T}.\ \mathsf{T}) \mid (\lambda \mathsf{Var} : \mathsf{T}.\ \mathsf{T}) \mid \mathsf{T}\ \mathsf{T},$$

其中 Var 是一个可数的变量集合。在 T 上，我们有通常的 β-归约、$\longrightarrow_\beta$ 概念。$\longrightarrow_\beta$ 的传递自反对称闭包记为$=_\beta$。

λP2 的推导规则导出形如 $\Gamma \vdash \mathrm{M} : \mathrm{T}$ 的判断式，其中 M 与 T 为伪项，Γ为上下文，即声明的有限序列 $\mathrm{v}_1 : \mathrm{T}1, \ldots, \mathrm{v}_n : \mathrm{T}_n$。此处，v 取遍 Var，s、$\mathrm{s}_1$与 $\mathrm{s}_2$取遍 {⋆,种类}，而 M、N、T 与 U 取遍 T。

$$\frac{}{\star : \mathsf{Kind}}\ \text{axiom} \qquad \frac{\Gamma \vdash T : \star/\mathsf{Kind} \quad \Gamma \vdash M : U}{\Gamma, v : T \vdash M : U}\ \text{var}$$

$$\frac{\Gamma, v : T \vdash M : U \quad \Gamma \vdash T : \star/\mathsf{Kind}}{\Gamma, v : T \vdash M : U}\ \text{weak} \qquad \frac{\Gamma \vdash M : T \quad \Gamma \vdash U : \star/\mathsf{Kind}}{\Gamma \vdash M : U}\ \text{conv}_\beta, \text{ if } T =_\beta U$$

$$\frac{\Gamma \vdash U : s_1 \quad \Gamma, x : U \vdash T : s_2}{\Gamma \vdash \Pi x : U.T : s_2}\ \Pi \text{ if } (s_1, s_2) \neq (\mathsf{Kind}, \mathsf{Kind})$$

$$\frac{\Gamma \vdash \Pi x : U.T : s \quad \Gamma, x : U \vdash b : T}{\Gamma \vdash \lambda x : U.b : \Pi x : U.T}\ \lambda \qquad \frac{\Gamma \vdash f : \Pi x : U.T \quad \Gamma \vdash a : U}{\Gamma \vdash f\ a : T[x := a]}\ \text{app}$$

在规则 (var) 与 (weak) 中，总是假定新声明的变量是新鲜的，即它尚未在Γ中被声明。

为记号方便，我们将集合 Var 划分为集合 $\mathrm{Var}^\star$（项变量）与 $\mathrm{Var}^{\mathsf{Kind}}$（构造子变量）。项变量记为 x,y,z . . .，构造子变量记为 α,β, . . .。在规则 (var) 与 (weak) 中，若 $\mathrm{s} = \star$，则从 $\mathrm{Var}^\star$ 中取变量 v；若 s =种类，则从 $\mathrm{Var}^{\mathsf{Kind}}$ 中取变量 v。良类型项可划分为以下互不相交的子集:

- {种类},
- 种类的集合：满足对某个 Γ 有 $\Gamma \vdash \mathrm{A} : $种类的表达式 A；其中包括 ⋆。在 λP2 中，所有种类都具有以下形式： $\Pi \mathrm{x}1 : \sigma_1 \ldots \Pi \mathrm{x}_n : \sigma_n . \star$，其中 $\sigma_1$、.....、$\sigma_n$个类型，以及 x1、.....、$\mathrm{x}_n \in \mathrm{Var}^\star$。
- 构造子的集合：类型为某个“种类”的表达式，即满足对某个种类 A 有 $\Gamma \vdash \mathrm{P} : \mathrm{A}$ 的表达式 P；其中包括类型，以及类型为 ⋆ 的表达式。在 λP2 中，所有构造子都具有以下形式之一：– $\alpha \in \mathrm{Var}^{\text{种类}}$，– P t，其中 P 是构造子，t 是一个项，– $\lambda \mathrm{x} : \sigma . \mathrm{P}$，其中 σ 是一个类型，P 是构造子，$\mathrm{x} \in \mathrm{Var}^\star$，– $\Pi \mathrm{x} : \sigma . \tau$，其中 σ 和 τ 是类型，$\mathrm{x} \in \mathrm{Var}^\star$，– $\Pi \alpha : \mathrm{A}. \tau$，其中 A 是一种类，τ 是一个类型，$\alpha \in \mathrm{Var}^{\text{种类}}$。

- 项：类型为某个“类型”的表达式，即满足 $\Gamma \vdash \mathrm{M} : \sigma$某个类型 σ 的表达式。在λP2 中，所有项都具有以下形式之一——$\mathrm{x} \in \mathrm{Var}^\star$,

——q t，其中 q 和 t 是项；——q P，其中 P 是构造子且 q 是项；—— λx : σ. t，其中 σ某个类型、t 是项、x ∈ Var⋆；——λα : A. t，其中 A 是一种类、t 是项、 α ∈ Var^Kind。

约定用 A、B、C.....表示种类，用 σ、τ、.....表示类型，用 P、Q.....表示构造子，用 t、q、.....表示项。

## 2.1 扩展

在 [1] 和 [8], 中，系统 λP2扩展了若干特性，以定义带有归纳原理的自然数数据类型，并定义带有余归纳原理的商类型和余归纳类型。我们在此列出这些扩展，并将在 λP2的模型中考察其中一些，以了解哪些扩展是实现 [1, 8]的数据类型所必需的。

定义 2.2。我们将以下规则视为 λP2的可能扩展。恒等类型

$$\frac{\Gamma\vdash\sigma:\star\quad\Gamma\vdash a:\sigma\quad\Gamma\vdash b:\sigma}{\Gamma\vdash a=^{\sigma}b:\star}\ \text{identity}\qquad\frac{\Gamma\vdash\sigma:\star\quad\Gamma\vdash a:\sigma}{\Gamma\vdash \mathsf{refl}:a=^{\sigma}a}\ \text{refl}$$

$$\frac{\Gamma\vdash a:\sigma\qquad\Gamma\vdash b:\sigma\qquad\Gamma\vdash q:a=^{\sigma}b\qquad\Gamma,x:\sigma,y:\sigma,p:x=^{\sigma}y\vdash\tau:\star\qquad\Gamma\vdash c:\Pi z:\sigma.\tau[x,y:=z,p:=\mathsf{refl}]}{\Gamma\vdash \mathsf{J}(c,a,b,q):\tau[x:=a,y:=b,p:=q]}\ \mathsf{J}$$

(恒等关系)的规则附带一条等式规则：

$$\mathsf{J}(c,a,a,\mathsf{refl})\longrightarrow_{\mathsf{refl}} c\ a$$

对于恒等类型，可以考虑以下两个原则：函数外延性（FunExt）和恒等证明唯一性（UIP）。（在恒等类型中，我们省略大写标签，因此将 x = y 写作 x =^α y，等等。）

$$\begin{aligned}\mathsf{FunExt}\quad&:\quad\Pi f,g:(\Pi x:\sigma.\,\tau).\,(\Pi x:\sigma.\,f\,x=g\,x)\implies f=g\\ \mathsf{UIP}\quad&:\quad\Pi\alpha:\star.\,\Pi x,y:\alpha.\,\Pi p,q:x=y.\,p=q\end{aligned}$$

Σ-类型

$$\frac{\Gamma\vdash\sigma:\star\quad\Gamma,x:\sigma\vdash\tau:\star}{\Gamma\vdash\Sigma x:\sigma.\tau:\star}\ \Sigma\qquad\frac{\Gamma\vdash a:\sigma\quad\Gamma\vdash b:\tau[x:=a]}{\Gamma\vdash\langle a,b\rangle:\Sigma x:\sigma.\tau}\ \langle-,-\rangle$$

$$\frac{\Gamma\vdash p:\Sigma x:\sigma.\tau}{\Gamma\vdash\pi_1\ p:\sigma}\ \pi_1\qquad\frac{\Gamma\vdash p:\Sigma x:\sigma.\tau}{\Gamma\vdash\pi_2\ p:\tau[x:=\pi_1\ p]}\ \pi_2$$

Σ的规则也包含等式规则：

$$\begin{aligned}\pi_1\ \langle a,b\rangle&=_{\beta}a\\ \pi_2\ \langle a,b\rangle&=_{\beta}b\end{aligned}$$

多态 λ-演算中归纳数据类型的著名编码可以直接扩展到λP2。

例 2.3. 1. 自然数可以用自然数 := Πα: ⋆.α → (α → α) → α 来编码，并带有零和后继：

$$\begin{aligned} \mathsf{O} &:= \lambda\alpha:\star.\,\lambda x:\alpha.\,\lambda f:\alpha\to\alpha.\,x,\\ \mathsf{succ} &:= \lambda n:\mathsf{nat}.\,\lambda\alpha:\star.\,\lambda x:\alpha.\,\lambda f:\alpha\to\alpha.\,f\ (n\ \alpha\ x\ f). \end{aligned}$$

归纳原理如下

$$\mathsf{ind}_{\mathsf{nat}} := \Pi P:\mathsf{nat}\to\star.\ P\ \mathsf{O}\to(\Pi y:\mathsf{nat}.\ P\ y\to P\ (\mathsf{succ}\ y))\to\Pi x:\mathsf{nat}.\ P\ x.$$

2. 该　布尔值的类型可以用布尔类型 := Πα: ⋆. α → α → α 来编码，并带有真和假：

$$\begin{aligned} \mathsf{true} &:= \lambda\alpha:\star.\,\lambda x,y:\alpha.\,x\\ \mathsf{false} &:= \lambda\alpha:\star.\,\lambda x,y:\alpha.\,y. \end{aligned}$$

归纳原理如下

$$\mathsf{ind}_{\mathsf{bool}} := \Pi P:\mathsf{bool}\to\star.\ P\ \mathsf{true}\to P\ \mathsf{false}\to\Pi x:\mathsf{bool}.\ P\ x.$$

3. 在 λP2 中，相等性是用莱布尼茨相等定义的：对于 σ : ⋆, t, q : σ

$$t =_\sigma q\ :=\ \Pi P:\sigma\to\star.\ P\ t\to P\ q.$$

我们希望归纳原理能够在 λP2 中得到证明，但事实并非如此；自然数和布尔类型都不行，实际上，其他自然数或布尔值的定义也不行[12]。此外，还有一些可以在 λP2 中定义的共归纳数据类型，我们或许可以为它们设想一个共归纳原理。我们将展示流的示例，但在此之前，先回顾可定义的存在类型，并介绍一些相关记法。

示例 2.4. 对于可能包含类型变量 α 的 σ 类型，存在类型 ∃α: ⋆. σ 定义为 Πβ : ⋆. (Πα: ⋆. σ → β) → β。

- 对于 τ : ⋆ 和 f : σ[α := τ]，我们有项 pack τ f := λβ : ⋆. λk : Πα : ⋆. σ → β . k τ f，因此 pack : Πβ : ⋆. σ[α:= β] → ∃α: ⋆. σ 是编码 ∃-引入的项。
- 对于消去规则，给定τ 类型 α ∉ FV(τ)、g : Πα : ⋆. σ → τ和t : ∃α: ⋆. σ，我们有项 rec∃ τ g t := t τ g : τ。因此，rec∃: Πβ : ⋆. (Πα : ⋆. σ → β) → (∃α: ⋆. σ) → β 是 ∃- 消去项，定义为 rec∃:= λβ : ⋆. λ g : Πα : ⋆. σ → β . λt : ∃α: ⋆. σ . t β g。

在 λP2中，我们还有众所周知的积数据类型 σ × τ，这里不再详述。配对和投影都是可定义的：对于 a : σ 和 b : τ，我们有 ⟨a, b⟩ : σ × τ，并且有 π1 : σ × τ → σ 和π2 : σ × τ → τ，满足 π1 ⟨a, b⟩ ⟶β a 和 π2 ⟨a, b ⟩ ⟶β b。对于积类型上的函数，我们允许使用一些语法糖，通过“模式抽象”写作λ⟨x, y⟩ : σ × τ . . . .，而不是 λ z : σ × τ . . . .；第二种情况下必须引用 π1 z和 π2 z，第一种情况下则可以直接引用 x 和 y。

例 2.5. 类型 σ 上的流类型 Stream 定义为 ∃α : ⋆.α × (α → σ) ×(α → α)[4]。析构器 hd(s : Stream) 定义为 rec∃ σ (λα.λ⟨x,h,t⟩. h x) s，而析构器 tl(s : Stream) 是

$$\mathsf{rec}_\exists\ \mathsf{Stream}\ (\lambda\alpha.\lambda\langle x,h,t\rangle.\ \mathsf{pack}\ \alpha\ \langle t\ x,h,t\rangle)\ s\ :\ \mathsf{Stream}.$$

[4]如果将其完整写出，并展开示例 2.4 中给出的存在类型定义，我们得到 Stream = Πβ : ⋆. (Πα : ⋆.α → (α → σ) → ( α → α) → β) → β

最后，我们将余递归子 $\mathsf{corec}_\mathsf{s}$ 定义如下。

$$\mathsf{corec}_\mathsf{s} := \lambda\alpha : \star.\ \lambda h : \alpha \to \sigma.\ \lambda t : \alpha \to \alpha.\ \lambda x : \alpha.\ \mathsf{pack}\ \alpha\ \langle x, h, t\rangle.$$

余递归子和析构函数（头函数和尾函数）的计算符合预期：对于 $\mathrm{h} : \tau \to \sigma$、$\mathrm{t} : \tau \to \tau$ 和 $\mathrm{x} : \tau$，我们有

$$\mathsf{hd}\ (\mathsf{corec}_\mathsf{s}\ \tau\ h\ t\ x) =_\beta h\ x \qquad\qquad \mathsf{tl}\ (\mathsf{corec}_\mathsf{s}\ \tau\ h\ t\ x) =_\beta \mathsf{corec}_\mathsf{s}\ \tau\ h\ t\ (t\ x)$$

有多种（等价的）方法可以使该流类型成为终余代数，详见 [8]。下面介绍其中两种。

1. 如果流的双模拟等价蕴含它们相等，那么流数据类型满足余归纳原理；更确切地说，我们有一个类型为

$$\Pi s, t : \mathsf{Stream}.\ s \sim t \to s = t,$$

其中 $\mathrm{s} \sim \mathrm{t}$ 定义为 $\exists \mathrm{R} : \mathrm{Stream} \to \mathrm{Stream} \to \star.$ (R 是一个双模拟关系，且 R s t)，其中 R 是一个双模拟关系，当且仅当它是对称的，并且对于所有 x,y : Stream，若 R x y，则 hd x = hd y 且 R (tl x) (tl y)。

2. 余　　递归子是唯一的：若对于 $\mathrm{h} : \tau \to \sigma$ 和 $\mathrm{t} : \tau \to \tau$，有项 $\mathrm{g} : \tau \to \mathrm{Stream}$，满足

$$\mathsf{hd}\ (g\ x) = h\ x \qquad\qquad \mathsf{tl}\ (g\ x) = g\ (t\ x),$$

则 $\mathrm{g} = \mathsf{corec}_\mathsf{s}\ \tau \mathrm{h}\ \mathrm{t}$

在定理 4.3 中，我们将证明这些可定义流并非终余代数，因此这两个原理并不成立。

## 2.2 商

为了将例 2.5 中的流类型变为流的终余代数，正如我们在 [8]，中所做的那样，我们需要对它按照双模拟关系取商。因此，我们现在介绍商。

定义 2.6。给定 $\sigma$、$\tau : \star$ 以及 $\mathrm{R} : \sigma \to \sigma \to \star$，若对于所有满足 R x y 的 $\mathrm{x}$、$\mathrm{y} : \sigma$，都有 $\mathrm{f\,x} = \mathrm{f\,y}$，则称函数 $\mathrm{f} : \sigma \to \tau$ 保持 R；我们将类型 $\Pi \mathrm{x}, \mathrm{y} : \sigma . \mathrm{R\,x\,y} \to \mathrm{f\,x} = \mathrm{f\,y}$ 记作 Resp f R。我们将商类型 quot σR 定义为

$$\mathsf{quot}\ \sigma\ R := \Pi C : \star.\ \Pi f : \sigma \to C.\ \mathsf{Resp}\ f\ R \to C.$$

类函数 $\mathsf{cls} : \sigma \to \mathsf{quot}\ \sigma\ \mathrm{R}$ 定义为

$$\mathsf{cls} := \lambda d : \sigma.\ \lambda C : \star.\ \lambda f : \sigma \to C.\ \lambda H : \mathsf{Resp}\ f\ R.\ f\ d.$$

请注意，假设 H 并未出现在主体 f d 中；定义 cls 并不需要它。我们记作 $\sigma/\mathrm{R} := \mathsf{quot}\ \sigma\ \mathrm{R}$。我们可以将保持 R 的函数 $\mathrm{f} : \sigma \to \mathrm{E}$ 提升为函数 $\mathrm{f} : \sigma/\mathrm{R} \to \mathrm{E}$。此提升通过商类型的递归子 $\mathsf{rec}_\mathsf{q}$ 完成，其定义如下。

$$\mathsf{rec}_\mathsf{q} := \lambda C : \star.\ \lambda f : \sigma \to C.\ \lambda H : \mathsf{Resp}\ f\ R.\ \lambda q : \sigma/R.\ q\ C\ f\ H$$

$\widehat{\ }$如果上下文中 C 和 H 已明确，我们记作 $\mathrm{f} := \lambda \mathrm{q}.\ \mathsf{rec}_\mathsf{q}\ \mathrm{C\,f\,H\,q}$。提升后的函数满足 $\widehat{\mathrm{f}} \circ \mathsf{cls} = \mathrm{f}$。

文献 [8], 证明了，如果 R x y，那么 cls x $=_{\text{cls}}$ y。证明使用了函数外延性。文中还展示了如何通过使用 Σ-类型来定义 $\sigma/R$ ^中满足 LimQuot 谓词的项所构成的子类型，从而得到一种更强的商形式：基本上，我们想定义 $\sigma/R$ 的子集，其中 f ^是唯一满足 f $\circ$cls $=$f 的函数。这也引出了一个问题：在 λP2 中，可定义商类型 $\sigma/R$ 本身是否已经是强商类型？或者在 λP2 中，是否存在另一种更“聪明”的商类型表示，能够成为强商类型？我们将先明确商类型应具备哪些性质，再证明这两个问题的答案是否定的。

定义 2.7。参数化商类型由以下内容给出：

1. 对于 $\sigma : \star$ 和 $R : \sigma \to \sigma \to \star$，一个类型 $\sigma/R$，称为 σ 模 R 的商类型；2. 一个项 $\text{cls} : \Pi\alpha : \star.\, \Pi R : \alpha \to \alpha \to \star.\, \alpha \to \alpha/R$；3. 满足下述类型有居留项（对每个 $\sigma : \star$）： $\Pi x, y : \sigma.\, R\, x\, y \to \text{cls}\ \sigma\ R\, x =_{\sigma/R} \text{cls}\ \sigma R\, y$；4. 一个满足下述条件的项构造 $\text{rec}_{\mathsf{q}}$

$$\frac{\tau : \star \quad f : \sigma \to \tau \quad p : \Pi x, y : \sigma.\, R\, x\, y \to f\, x =_\tau f\, y}{\text{rec}_{\mathsf{q}}\ p\ f : \sigma/R \to \tau}$$

^其中，我们将 $\text{rec}_{\mathsf{q}}$ p f 缩写为 f

^5. 且对每个 x : σ，f(cls x) $=_\beta$ f x。

如果除上述条件外还满足以下条件，我们称 $\sigma/R$ 为强参数化商类型

6. 满足 IndQuot 的术语构造

$$\frac{P : \sigma/R \to \star \quad t : \Pi x : \sigma.\, P\,(\text{cls}\, x)}{\text{IndQuot}\ t : \Pi y : \sigma.\, R.P\, y}$$

在商类型的定义中，我们并未要求 R 是等价关系。要定义 f，我们要求 Resp f R，即 f 应当尊重 R；但显然，$=_{\sigma/R}$ 是等价关系，因此实际上，定义 2.7 中的类型 $\sigma/R$ 是 σ 对包含 R [5] 的最小等价关系取商所得的商。

我们可以期待强商具备比定义 2.7 所要求的更多性质，其中一些性质可以推导出来，有时只需弱附加假设。例如，cls y$=_{\sigma/R}$ y^（其中 $y : \sigma/R$）可由 IndQuot 推导得出；如果还满足函数外延性，则可推导出 f 的唯一性（商的 η-规则）。最后，我们或许希望通过要求 Πx, y : σ. cls σ R x $=_{\sigma/R}$ cls σ R y $\to$ R x y，使商具有有效性。在 [8], 中，我们展示了如何在 λP2 + 函数外延性的扩展中构造参数化商。本文不再进一步利用这一结果，因为我们已经可以在 λP2 中证明定义 2.7 的参数化商不存在，见命题 4.4。

需要注意的是，我们并未证明诸如“商在 λP2 中不可定义”这样的普遍命题，因为一般而言这是错误的：对于特定的 σ 和 R，商 $\sigma/R$ 确实存在。我们给出平凡商的例子，其中取 R 为莱布尼茨相等。

例 2.8. 取 σ: 自然数，且 R x y := x $=_{\text{nat}}$ y，我们定义 nat/R := nat，并可见它是定义 2.7 意义下的非参数化商，其中我们省略了 cls 是一个多态函数这一要求。

要成为前 因此，我们有 cls : 自然数 $\to$自然数/R，由 cls := λx : 自然数. x 定义，显然满足 s

[5] Hofmann [15] 也曾采用类似的思路来定义商。

- Πx, y : 自然数. R x y → cls x =自然数 cls y;

•对于 f : σ → τ ，满足 Resp f R，取 f :≡ f，则 f ∘cls = f;
• 若 Πx : nat. P (cls x)，则 Πy : nat/R. P y;

- Πx, y : 自然数. cls x =自然数 cls y → R x y,

因此，这个平凡商的例子也是一个有效商。

## 3 λP2 的模型构造

我们对 λP2 采用的模型概念与 [12, 11] 中的相同。它可以扩展为构造演算的一类模型，相关工作见 [23]。我们的模型属于可实现性语义的一种，其含义是：类型将被解释为实现子的集合，而实现子取自组合代数。项将被解释为实现子。为了适应依赖类型，我们的模型基于弱外延组合代数（WECA）构造。组合代数（简称 CA）是一个元组 $\mathscr{A} = \langle A, \cdot, k, s\rangle$，其中 A 是一个集合，$\cdot$ 是从 $A \times A$ 到 A 的二元函数（通常以中缀形式表示），$k、s \in A$，并满足 $(k \cdot a) \cdot b = a$ 以及 $((s \cdot a) \cdot b) \cdot c = (a \cdot c) \cdot (b \cdot c)$。对于 $\mathscr{A}$ 一个组合代数，$\mathscr{A}$ 上的项集合 $\mathscr{T}(\mathscr{A})$ 的定义如下：令 $\mathscr{T}(\mathscr{A})$ 包含无限多个变量 $v_1, v2、\ldots\ldots$，以及对每个 $a \in A$ 都有一个不同的元素 $c_a$；再令 $\mathscr{T}(\mathscr{A})$ 对应用（运算 $\cdot$）封闭。给定一个项 t 和一个赋值 ρ（将变量映射到 A 中的元素），t 在 A 中关于 ρ 的解释（记作 $[\![t]\!]_\rho^{\mathscr{A}}$）按通常方式定义（$[\![c_a]\!]_\rho^{\mathscr{A}} = a$、$[\![MN]\!]_\rho^{\mathscr{A}} = [\![M]\!]_\rho^{\mathscr{A}} \cdot [\![N]\!]_\rho^{\mathscr{A}}$，等等）。CA 的一个关键性质是组合完备：若 $t[v] \in \mathscr{T}(\mathscr{A})$ 是一个自由变量为 v 的项，则 A 中存在一个元素，通常记作 $\lambda^* v.\, t[v],$，使得对于所有 ρ 以及 $a \in A$，都有 $[\![(\lambda^* v.\, t[v])]\!]_\rho^{\mathscr{A}} \cdot a = [\![t[x]]\!]_{\rho(x:=a)}^{\mathscr{A}}$。若对所有 $a \in A$，$[\![t_1]\!]_{\rho(x:=a)}^{\mathscr{A}} = [\![t_2]\!]_{\rho(x:=a)}^{\mathscr{A}}$ 都蕴含 $[\![\lambda^* x.t_1]\!]_\rho^{\mathscr{A}} = [\![\lambda^* x.t_2]\!]_\rho^{\mathscr{A}}$，则称 CA 弱外延。

对弱外延 CAs 的需求源于这样一个事实：为了确保转换规则是可靠的，我们希望以下结论在我们的 λP2-模型中成立。

$$M =_\beta N \implies (\!|M|\!)_\rho = (\!|N|\!)_\rho \text{ for all } \rho,$$

其中，$(\!|-|\!)_\rho$ 使用自由变量的赋值 ρ将伪项解释为 A 的元素。解释 $(\!|-|\!)_\rho$ 与 $[\![-]\!]_\rho^{\mathscr{A}}$相近，但现在我们也解释抽象：在 $(\!|-|\!)_\rho$下，λ被解释为 $\lambda^*$。

例 3.1。1. WECA 的一个标准例子是 Λ，它由开 λ-项按 β-相等性取商后得到的等价类构成。因此，A 就是 $\Lambda/\beta$，并且 $[M] = [N]$ 当且仅当 $M =_\beta N$。容易验证，这构成一个 WECA。

2. 给定一个常量集合 C，我们将 WECA $\Lambda(C)$ 定义为开 $\lambda_C$-项（即常量集合 C 上的 lambda 项）按 βc-相等性取商后得到的等价类，其中 c-相等性规则规定

$$cN =_c c.$$

对于所有 $c \in C$ 和 $N \in \Lambda_C$。

3. WECA 的另一个例子是 1，即退化 WECA，其中 $A = 1$ 是单元素集合。在这种情况下，$k = s$，这在组合代数中通常是不允许的，但请注意，我们在此处确实允许这种情况。

λP2 的类型将被解释为 A 的子集。

定义 3.2. 弱外延组合代数 $\mathscr{A}$ 上的多集合结构是一个集合$\mathscr{P} \subseteq \wp$(A)，满足

1. A $\in \mathscr{P}$, 2. $\emptyset \in \mathscr{P}$[6], 3. $\mathscr{P}$ 在任意交集下
封闭 $\bigcap$,

4. $\mathscr{P}$ 在依赖积下封闭，也就是说，若 X $\in \mathscr{P}$ 且 F : X $\to \mathscr{P}$，则 $\Pi_{t\in X}$F(t) $\in \mathscr{P}$，其中$\Pi_{t\in X}$F(t) 定义为

$$\{a \in \mathbf{A} \mid \forall t \in X(a \cdot t \in F(t))\}.$$

多集合结构的元素称为多集合。若 F 是取值为 Y 的常值函数，我们用 X $\to$ Y 表示，而不写作 $\Pi_{t\in X}$ Y。

例 3.3. 1. 若取 $\mathscr{P} = \wp$(A)，则可得到 WECA $\mathscr{A}$ 上的完全多集结构。

2. 取 $\mathscr{P} = \{/0, \mathrm{A}\}$，即可得到 WECA $\mathscr{A}$ 上的简单多集结构。容易验证，这确实是一个多集合结构。

3. 给定例 3.1 中定义的 WECA $\Lambda$(C)（因此 C 是一个常量集合），我们定义由 C 生成的多集合结构 $\mathscr{P}^{\Lambda}$(C) 如下：

$$\mathscr{P}^{\Lambda}(C) := \{X \subseteq \Lambda(C) \mid X = \emptyset \vee C \subseteq X\}.$$

要证明 $\mathscr{P}^{\Lambda}$(C) 是一个多集合结构，我们需要验证它在依赖积下封闭，这很容易：若 X $\neq$/0，且对所有 t $\in$X 都有 F(t) $\neq$/0，那么 C $\subseteq \Pi_{t\in X}$F(t)，因为对于 c $\in$C 和 t $\in$X，有 ct $=_c$ c $\in$C $\subseteq$F(t)。

4. 给定 WECA $\mathscr{A}$，以及集合 C $\subseteq$A，使得 $\forall$a, b $\in$A(a$\cdot$b $\in$C $\Longrightarrow$a $\in$C，我们定义 C 的幂多集结构如下：

$$\mathscr{P} := \{X \subseteq \mathbf{A} \mid X \subseteq C \vee X = \mathbf{A}\}.$$

要验证这是一个多集合结构，只需证明对于 X $\in \mathscr{P}$，以及 F : X $\to \mathscr{P}$，$\Pi_{t\in X}$F(t) $\in \mathscr{P}$。这可通过简单的分类讨论得出：$\forall$t $\in$X (F(t) $=$A)，或者 $\exists$t $\in$X (F(t) $\subseteq$C)。幂多集结构的一个有趣实例是由 C $=$HNF（即具有头部范式的 $\lambda$-项集合）在 WECA $\Lambda/\beta$中导出的结构。

要解释种类，我们需要一个谓词结构。

定义 3.4. 对于 $\mathscr{P}$ 一个多集合结构，其上的谓词结构是由归纳定义的集合族 $\mathscr{N}$：

1. $\mathscr{P} \in \mathscr{N}$，2. 如果 X $\in \mathscr{P}$ 且 $\forall$t $\in$X(F(t) $\in \mathscr{N}$，
那么 $\prod_{t\in X}$ F(t) $\in \mathscr{N}$。

如果 F 是一个取值为 $\mathscr{P}$ 的常值函数，我们记作 X $\to \mathscr{P}$，而不是 $\prod_{t\in X} \mathscr{P}$。

定义 3.5. 若 $\mathscr{A}$是一个组合代数，$\mathscr{P}$ 是 $\mathscr{A}$上的一个多集合结构，则称 $\langle \mathscr{A}, \mathscr{P} \rangle$ 为一个多集模型。

[6]在 [12, 11, 23] 中我们不要求这一点，并将 $\emptyset \in \mathscr{P}$ 的模型称为“一致的”，但无论如何，这里我们只想讨论一致模型。

若将定义 3.4 视为 $\mathscr{N}$的闭包条件，则多集合结构 $\mathscr{P}$ 上的谓词结构 $\mathscr{N}$也可以作为多集模型的附加参数。不过，我们在这里不需要如此一般的设定。它旨在为种类提供一个解释域。例如，如果类型 $\sigma$ 被解释为多集合 X，那么种类 $\sigma \to \sigma \to \star$ 就被解释为 $\prod_{t\in X}\prod_{q\in X}\mathscr{P}$，我们通常将其记作 $X \to X \to \mathscr{P}$。

现在我们定义三个解释函数：一个用于种类的解释函数 $\mathscr{V}(-)$，将种类映射到 $\mathscr{N}$ 的元素；一个用于构造子（以及类型）的解释函数 $[\![-]\!]$，将构造子映射到 $\bigcup\mathscr{N}$ 的元素（并将类型映射到 $\mathscr{P}$ 的元素，而 $\mathscr{P}$ 是 $\bigcup\mathscr{N}$ 的子集）；以及一个用于项的解释函数 $(\!|-|\!)$，将项映射到组合代数 $\mathscr{A}$ 的元素。所有这些解释都以赋值为参数，将自由变量（在上下文中声明）赋予相应的值。

以下设 $\mathscr{M} = \langle \mathscr{A}、\mathscr{P}\rangle$ 为一个多集模型：$\mathscr{A} = \langle A、k、s\rangle$ 是一个组合代数，$\mathscr{P}$ 是 $\mathscr{A}$ 上的多集合结构，而 $\mathscr{N}$ 是多集合结构 $\mathscr{P}$ 上的谓词结构。

定义 3.6。构造子变量赋值是一个从 $\mathrm{Var}^{\mathsf{Kind}}$ 到 $\bigcup\mathscr{N}$ 的映射 $\xi$。项变量赋值是一个从 $\mathrm{Var}^{\star}$ 到 A 的映射 $\rho$。

定义 3.7。对于 $\rho$一个项变量赋值，我们定义从项集到 A 的映射 $(\!|-|\!)^{\mathscr{M}}_{\rho}$如下。（模型 $\mathscr{M}$ 保持隐含。）

$$
\begin{aligned}
(\!|x|\!)_\rho &:= \rho(x),\\
(\!|tq|\!)_\rho &:= (\!|t|\!)_\rho \cdot (\!|q|\!)_\rho, \text{ if } q \text{ is an term},\\
(\!|tQ|\!)_\rho &:= (\!|t|\!)_\rho, \text{ if } Q \text{ is a constructor},\\
(\!|\lambda x:\sigma.t|\!)_\rho &:= \lambda^* v.(\!|t|\!)_{\rho(x:=v)}, \text{ if } \sigma \text{ is a type},\\
(\!|\lambda \alpha:A.t|\!)_\rho &:= (\!|t|\!)_\rho, \text{ if } A \text{ is a kind}.
\end{aligned}
$$

定义 3.8。对于 $\rho$一个项变量赋值和 $\xi$ 一个构造子变量赋值，我们分别定义从种类到 $\mathscr{N}$、从构造子到 $\bigcup\mathscr{N}$的映射 $\mathscr{V}(-)^{\mathscr{M}}_{\xi\rho}$ 和 $[\![-]\!]^{\mathscr{M}}_{\xi\rho}$ 如下。（模型 $\mathscr{M}$ 保持隐含。）

$$
\begin{aligned}
\mathscr{V}(\star)_{\xi\rho} &:= \mathscr{P},\\
\mathscr{V}(\Pi x:\sigma.B)_{\xi\rho} &:= \prod_{t\in[\![\sigma]\!]_{\xi\rho}} \mathscr{V}(B)_{\xi\rho(x:=t)},\\
[\![\alpha]\!]_{\xi\rho} &:= \xi(\alpha),\\
[\![\Pi\alpha:A.\tau]\!]_{\xi\rho} &:= \bigcap_{a\in\mathscr{V}(A)_{\xi\rho}} [\![\tau]\!]_{\xi(\alpha:=a)\rho}, \text{ if } A \text{ is a kind},\\
[\![\Pi x:\sigma.\tau]\!]_{\xi\rho} &:= \Pi_{t\in[\![\sigma]\!]_{\xi\rho}}[\![\tau]\!]_{\xi\rho(x:=t)}, \text{ if } \sigma \text{ is a type},\\
[\![Pt]\!]_{\xi\rho} &:= [\![P]\!]_{\xi\rho}((\!|t|\!)_\rho),\\
[\![\lambda x:\sigma.P]\!]_{\xi\rho} &:= \lambda t\in[\![\sigma]\!]_{\xi\rho}.[\![P]\!]_{\xi\rho(x:=t)}.
\end{aligned}
$$

定义 3.9。对于 $\Gamma$ 一个 $\lambda$P2-上下文、$\rho$一个项变量赋值和 $\xi$ 一个构造子变量赋值，若对所有 $x \in \mathrm{Var}^{\star}$和 $\alpha\in\mathrm{Var}^{\mathsf{Kind}}$，均有 $x:\sigma\in\Gamma \Rightarrow \rho(x) \in [\![\sigma]\!]_{\xi\rho}$且 $\alpha: A \in \Gamma \Rightarrow \xi(\alpha) \in \mathscr{V}(A)_{\xi\rho}$，则称 $\xi,\rho$ 满足 $\Gamma$，记作 $\xi,\rho \models \Gamma$。

定义中（隐含地）规定：仅当对所有声明 $x:\sigma\in\Gamma$，$[\![\sigma]\!]_{\xi\rho}$ 均有定义时，$\xi\rho \models \Gamma$ 才成立（对 $\alpha: A \in \Gamma$ 亦然）。

定义 3.10。多集模型中的真这一概念（记作 $\models^{\mathscr{M}}$ ）以及真这一概念（记作 $\models$）定义如下。对于上下文 Γ、项 t、类型 σ、构造子 P，以及 λP2 的种类 A,

$$\Gamma \models^{\mathscr{M}} t : \sigma \quad \text{if} \quad \forall \xi, \rho\, [\xi, \rho \models \Gamma \Rightarrow (\!|t|\!)\rho \in [\![\sigma]\!]_{\xi\rho}],$$
$$\Gamma \models^{\mathscr{M}} P : A \quad \text{if} \quad \forall \xi, \rho\, [\xi, \rho \models \Gamma \Rightarrow [\![P]\!]_{\xi\rho} \in \mathscr{V}(A)_{\xi\rho}].$$

对所有多集模型这一类进行量化，对于 λP2 中的项或构造子 M，我们定义：

$$\Gamma \models M : T \text{ if } \Gamma \models^{\mathscr{M}} M : T \text{ for all } \lambda P2\text{-models } \mathscr{M}.$$

可靠性表明，若判断 Γ ⊢ M : T 可导出，则它在所有模型中都为真。该结论通过对 λP2 中的推导进行归纳来证明。事实上，可靠性还蕴含 $\mathscr{V}$(A)ξρ and[![P]!]ξρ 是良定义的。

定理 3.11（ 可靠性）。设 Γa 为上下文，M 为项或构造子，T 为 λP 中的类型或种类。 2,

$$\Gamma \vdash M : T \Rightarrow \Gamma \models M : T.$$

例 3.12。设 $\mathscr{A}$ 为一个 WECA。

1. 在 $\mathscr{A}$ 上的完全多集模型是 $\mathscr{M} = \langle \mathscr{A}, \mathscr{P} \rangle$，其中 $\mathscr{P}$ 是 $\mathscr{A}$ 上的完全多集合结构（定义见例 3.3 (1)）。2. 在 $\mathscr{A}$ 上的简单多集模型是 $\mathscr{M} = \langle \mathscr{A}, \mathscr{P} \rangle$，其中 $\mathscr{P}$ 是 $\mathscr{A}$ 上的简单多集合结构。（定义见例 3.3 (2)，因此 $\mathscr{P}$ = {/0，A}。）3. 在退化的 $\mathscr{A}$ 上的简单多集模型也称为 λP2 的证明无关模型或 PI模型。4. 对于常量集合 C，由 C 生成的多集模型定义为 $\mathscr{M} = \langle \Lambda(C)\ \mathscr{P}^{\Lambda}(C) \rangle$，其中 $\mathscr{P}^{\Lambda}$(C) 是由 C 生成的多集合结构。参见例 3.3 (3) 和例 3.1 (2)。

## 4 λP2 中的不可推导性结果

我们不讨论多集模型的一般性质（相关内容见 [12] ），而是通过构造一个具体的反模型，说明某些类型构造无法在 λλP2 中定义。

在逻辑中，公式 φ 的有效性意味着 φ的解释在模型中为真。在类型论中，如果一个类型的解释非空，即该类型有居留项，我们就称该类型有效。

定义 4.1。对于 $\mathscr{M}$一个多集模型、 Γ一个上下文、 σ 中及 Γ中的一个类型，以及满足 ξ、ρ ⊨Γ的 ξ、ρ 赋值，我们称 σ 在 $\mathscr{M}$中关于 ξ、ρ有效，记作 $\mathscr{M}$、ξ、ρ $\Vdash^{\lambda P2}$ σ，如果

$$[\![\sigma]\!]^{\mathscr{M}}_{\xi\rho} \neq \emptyset.$$

我们通常只写作 $\Vdash^{\lambda P2}$ σ，尤其是在模型 $\mathscr{M}$ 可由上下文明确确定且类型 σ是闭类型时。

因此，当我们所说的意思是类型 σ在类型论中有居留项时，有时会简略地说“σ成立”；类似地，当我们所说的意思是 σ的解释在模型中非空时，也会说“σ在模型中成立”。

我们回顾一个关于多集模型的重要引理，后续将会用到它。

引理 4.2。（基本上是 [12] 的引理 4。）设 σ 为一个类型，带有项 t, q : σ，可能处于某个上下文中。

- 对于每个多集模型 $\mathscr{M}$，我们有

$$\mathscr{M}, \xi, \rho \models t =_\sigma q \iff (\!|t|\!)_\rho = (\!|q|\!)_\rho,$$

其中右侧的相等性是该模型的 WECA 中的相等性。

- $\Gamma \vdash M : t =_\sigma q$ 蕴含：对于每个多集模型 $\mathscr{M}$ 以及每个 ξ、ρ，如果 ξ、ρ ⊨ Γ，那么 $(\!|t|\!)_\rho = (\!|q|\!)_\rho$。

证明。第一条是 [12] 的引理 4。第二条立即由此得到。 □

定理 4.3。λP2 中的可定义流（如例 2.5 所给出）不具有余归纳原理。

证明。我们考虑布尔类型上的流，并定义两个流 $s_1, s_2 : \mathsf{Stream}_{\mathsf{bool}}$，使得 $s_1 \sim s_2$，但不满足 $s_1 =_{\mathsf{Stream}_{\mathsf{bool}}} s_2$。定义

$$\begin{aligned} s_1 &:= \mathsf{pack\ bool}\ \langle \mathsf{true}, \neg, \neg \rangle \\ s_2 &:= \mathsf{pack\ bool}\ \langle \mathsf{false}, \mathsf{I}, \neg \rangle, \end{aligned}$$

其中 I 是布尔类型上的恒等关系，真和假如例 2.3 中所定义，且 ¬ := λb : 布尔类型。b 布尔类型 假 真。注意，hd $s_1$ = ¬ 真 = 假 = hd $s_2$，类似地，hd(tl $s_1$) = ¬ (¬ 真) = 真，且 hd(tl $s_2$) =I (¬ 假) =真：$s_1$ 和 $s_2$ 的所有有限观测都相同。

我们证明 (1) $s_1 \sim s_2$，但 (2) 不满足 $s_1 =_{\mathsf{Stream}_{\mathsf{bool}}} s_2$。

对于 (1)：对于 x, y : $\mathsf{Stream}_{\mathsf{bool}}$，定义 R x y := (x = $s_1$ ∧ y = $s_2$) ∨ (y = $s_1$ ∧ x = $s_2$) ∨ (x = tl $s_1$ ∧ y =tl $s_2$) ∨ (y = tl $s_1$ ∧ x = tl $s_2$)。那么 R 是对称的；若 R x y，则 hd x = hd y，正如我们已经观察到的。此外，若 R x y，则 R (tl x) (tl y)，因为 tl (tl, $s_1$) = $s_1$ 且 tl (tl, $s_2$) = $s_2$。

对于（2）：假设 $s_1 =_{\mathsf{Stream}_{\mathsf{bool}}} s_2$。那么，在任何模型中，$(\!|s_1|\!) = (\!|s_1|\!)$（引理 4.2），因此，若我们考虑模 βη 的无类型 λ-项的 WECA，就会发现 $(\!|s_1|\!) =_{\beta\eta} (\!|s_1|\!)$。现在，若将 pack 完整展开，则有 pack σ ⟨ a, h,t⟩ = λY : ⋆.λk : ΠX : ⋆X →(X →布尔类型) →(X →X ) →Y.k σ a h t，因此我们发现，在λ-演算中，$(\!|s_1|\!) = \lambda k.\ k\ K\ N\ N = \lambda k.\ k\ K_*\ I\ N = (\!|s_2|\!)$[7]，其中 N = $(\!|\neg|\!) = \lambda b.\ b\ K_*\ K$。但 λk.k K N N 和 λk.k $K_*$ I N 是两个不同的范式，因此它们不等价于 βη。矛盾，因此我们得出结论：非 $s_1 =_{\mathsf{Stream}_{\mathsf{bool}}} s_2$[8]。□

定理 4.4. 参数化商（定义 2.7 中的）在 λP2 中不可定义。

证明。我们可以证明，定义 2.7 中前五项的组合无法在 λP2 中实现。假设有一个项 cls : Πα: ⋆.ΠR : α → α → ⋆.α → α/R，使得对于所有 σ : ⋆，类型 Πx,y : σ.R x y →cls σ R x =σ/R cls σ R y 有居留项。取 α a 为类型变量，并令 λx, y : α. ⊤ for R（其中 ⊤ := Πβ: ⋆.β → β，类型为“真”），我们可以看到存在一个项 t，使得

$$\alpha : \star, x, y : \alpha \vdash t : \mathsf{cls}\ \alpha\ R\ x =_{\alpha/R} \mathsf{cls}\ \alpha\ R\ y.$$

在由 WECA Λ 构建的模型中，取 $[\![\alpha]\!]_{\xi\rho} = \Lambda$，我们看到对于每个 M、N ∈ Λ，$(\!|\mathsf{cls}|\!)\ M =_\beta (\!|\mathsf{cls}|\!)\ N$。（这使用了引理 4.2。）因此 $(\!|\mathsf{cls}|\!)$ 在 Λ 中是常函数。但事实并非如此：如果我们^ ^取，例如，σ: = 布尔类型，R x y : = x =bool y 以及 f : = $I_{\mathsf{bool}}$，则 f : 布尔类型 → 布尔类型，且 f (cls x) = x 在模型 Λ 中得到 K =$K_*$，而这是假的。（如果 $(\!|\mathsf{cls}|\!)$ 在模型 Λ 中是常值，则我们在模型中有：^ ^真 = I 真 = I (cls 真) =I (cls 假) = I 假 = 假。□

[7] $(\!|真|\!) = \lambda x\ y.\ x = K$ 和 $(\!|假|\!) = \lambda x\ y.\ y = K_*$

[8] 对于 (2)，我们实际上并不需要模型，因为我们知道在 λP2 中，若 ⊢ M : $s_1 =_{\mathsf{Stream}_{\mathsf{bool}}} s_2$（对某个 M），则 $s_1 =_\beta s_2$,；但事实并非如此，因为 $s_1$ 和 $s_2$ 是不同的范式。

# 5 λP2的扩展模型

我们现在扩展定义 3.5 的多集模型，以纳入恒等类型和 Σ-类型。我们将证明定义 2.2 中恒等类型与 Σ-类型的规则在该模型中是可靠的，并且 UIP 成立。我们还将给出一个模型，其中函数外延性不成立，自然数归纳也不成立，这表明函数外延性对于 [1] 中的构造至关重要。

定义 5.1. 我们通过在无类型的 λ演算中加入两个常量 J 和 refl，并采用以下归约规则，来定义 Λid。

$$\begin{array}{llll} \mathsf{J}\ c\ a\ b\ \mathsf{refl} & \longrightarrow_J & c\ a & \qquad \mathsf{refl}\ q \longrightarrow_{\mathsf{refl}} \mathsf{refl}, \\ \pi_1\ \langle a,b\rangle & \longrightarrow_\Sigma & a & \qquad \pi_1\ \mathsf{refl} \longrightarrow_{\mathsf{refl}} \mathsf{refl}, \\ \pi_2\ \langle a,b\rangle & \longrightarrow_\Sigma & b & \qquad \pi_2\ \mathsf{refl} \longrightarrow_{\mathsf{refl}} \mathsf{refl}. \end{array}$$

$\longrightarrow_J$ 的规则表示恒等类型的归约规则，它结合了 J消去子和 refl构造子； $\longrightarrow_\Sigma$ 的规则表示 Σ类型的β规则。在有类型的情形下不会出现 $\longrightarrow_{\mathsf{refl}}$ 的规则，因为 refl M 和 $\pi_1$ refl 永远不是良类型项。不过，这些规则无害，也就是说，它们不会破坏 Church-Rosser性质（见 [18, 16]），并且使我们能够定义一个有趣的模型。

将解释扩展到 Σ类型，可以通过扩展多集结构（定义 3.2）的概念，以一种直接的方式纳入依赖和。然而，我们尤其关注构造一个反模型，而这将基于 Λid和具体的多集合结构 $\mathscr{P}^{\Lambda\mathrm{id}}(\{\mathrm{refl}\})$。因此，我们将定义从 Σ类型到 $\mathscr{P}^{\Lambda\mathrm{id}}(\{\mathrm{refl}\})$的具体解释。

定义 5.2。我们取 $\mathscr{P}$ 为 $\mathscr{P}^{\Lambda\mathrm{id}}(\{\mathrm{refl}\})$。对于 $X \in \mathscr{P}$，若 $F : X \to \mathscr{P}$，则称 F 是 proper，当且仅当 $F(\mathrm{refl}) = \emptyset \to \forall t \in X\ (F(t) \neq /0)$。

对于 $X \in \mathscr{P}$ 以及 proper 的 $F : X \to \mathscr{P}$，我们定义依赖和 $\Sigma_{t\in X}.F(t)$ 如下。

$$\Sigma_{t\in X}.F(t) := \{q \mid \pi_1\ q \in X \wedge \pi_2\ q \in F(\pi_1\ q)\}.$$

由于我们为 Λid 选择的归约规则，以及我们将范围限制为 proper 的 $F : X \to \mathscr{P}$，可以看出 $\mathscr{P}^{\Lambda\mathrm{id}}(\{\mathrm{refl}\})$ 在依赖和下封闭。

引理 5.3。我们取 $\mathscr{P}$ 为 $\mathscr{P}^{\Lambda\mathrm{id}}(\{\mathrm{refl}\})$。对于 $X \in \mathscr{P}$ 以及 proper 的 $F : X \to \mathscr{P}$，依赖和$\Sigma_{t\in X}.F(t)$ 也属于 $\mathscr{P}$。

证明。设 $X \in \mathscr{P}$，且 $F : X \to \mathscr{P}$ proper。我们需要证明 $\Sigma_{t\in X}.F(t) \neq /0$ 或 $\mathrm{refl} \in \Sigma_{t\in X}.F(t)$。分情况讨论 $X \neq /0$。于是 $\Sigma_{t\in X}.F(t) \neq /0$。再讨论 $X \neq /0$ 的情况。子情况 $F(\mathrm{refl}) \neq /0$。于是 $\mathrm{refl} \in X$ 且 $\mathrm{refl} \in F(\mathrm{refl})$，所以 $\mathrm{refl} \in \Sigma_{t\in X}.F(t)$。子情况 $F\ (\mathrm{refl}) = /0$，则 $\forall t \in X\ (F(t) = /0)$，所以 $\Sigma_{t\in X}.F(t) = /0$。□

多集合结构中 Σ-类型的解释如下。

$$[\![\Sigma x : \sigma.\tau]\!]_{\xi\rho} := \Sigma_{t\in[\![\sigma]\!]_{\xi\rho}} [\![\tau]\!]_{\xi\rho(x:=t)}.$$

涉及配对和投影的项的解释很直接，因此 Σ-类型推导规则的可靠性也是如此，我们不加证明地陈述如下。

引理5.4。多集模型 $\mathscr{P}^{\Lambda\mathrm{id}}(\{\mathrm{refl}\})$ 对 $\lambda P2^\Sigma$ 是可靠的，也就是说，它是扩展了 Σ-类型规则的 λP2。

现在我们将解释扩展到恒等类型。请注意，恒等类型在逻辑上等价于莱布尼茨相等（我们有类型为 $\forall x, y : \sigma.\, x =_\sigma y \to x =^\sigma y$ 的项，以及类型为 $\forall x, y : \sigma.\, x =^\sigma y \to x =_\sigma y$ 的项），但它们之间不存在同构，因此我们不能直接将多集合模型中莱布尼茨相等的解释用作恒等类型的解释。

定义 5.5。设 $\mathscr{P}$ 是一个多集合模型。设 σ 是一个类型，且 a, b : σ。给定赋值 ξ,ρ，我们将类型 a $=^{\sigma}$ b 解释如下。为使记号清晰易读，我们将 a′ 写作 $(\!|a|\!)_\rho$，将 b′ 写作 $(\!|b|\!)_\rho$，并将 σ′ 写作 $[\![\sigma]\!]_{\xi\rho}$。

$$[\![a =^{\sigma} b]\!]_{\xi\rho} := \bigcap_{C\in\sigma'\to\sigma'\to\mathbf{A}\to\mathscr{P}} \{q \mid \forall t \in \Pi_{x\in\sigma'}.C(x,x,\mathrm{refl})\ [J\ t\ a'\ b'\ q \in C(a',b',q)]\}.$$

注意，$[\![a =^{\sigma} b]\!]_{\xi\rho}$ 不一定存在。我们知道，$\mathscr{P}$ 对任意交封闭，但集合 $\{q \mid \forall t \in \Pi_{x\in\sigma'}.$ C(x, x, refl) [J t a′b′q ∈C(a′, b′, q)]} 不一定属于 $\mathscr{P}$。

引理 5.6。定义 5.5 中 a $=^{\sigma}$ b 的解释是良定义的，适用于

1. 全多集模型 $\mathscr{P} := \wp(\Lambda\mathrm{id})$；2. 由 C := {refl}, $\mathscr{P}^{\Lambda\mathrm{id}}$({refl}) 生成的多集模型。

证明。对于第一种情形，结论显然；对于第二种情形，分情况讨论：(1) 若 a′ =b′，则 refl $\in [\![a =^{\sigma} b]\!]_{\xi\rho}$。(2) 若 a′ ≠ b′，我们可以选取一个 C，使得对于所有 q ∈ A，都有 C(a′, b′, q) = /0；而对于所有 t ∈ σ′ 以及所有 q ∈ A，都有 C(t,t, q) = A。于是 $[\![a =^{\sigma} b]\!]_{\xi\rho}$ = /0。□

引理 5.7。恒等类型的规则在一个多集合模型中是可靠的，其中 $=^{\sigma}$b 的解释是良定义的。

证明。我们需要检查 refl 的规则和 J 的规则。

refl 我们必须检查，对于任意类型 σ 和 a : σ，refl $\in [\![a =^{\sigma} a]\!]_{\xi\rho}$。这是成立的，因为再次记 a′ 为 $(\!|a|\!)_\rho$，我们有 t a′ ∈ C(a′, a′, refl)，且 J t a′ a′ refl = t a′，因此 J t a′ a′ refl ∈ C(a′, a′, refl)。

J 我们来看 J 规则的假设：设 Γ 是一个上下文，并且在 Γ 中有 σ : ⋆, a, b : σ, q : a $=^{\sigma}$ b 和 c : Πz : σ. τ[x, y := z, p := refl]。此外，我们有 τ，其中 Γ, x : σ, y : σ, p : x $=^{\sigma}$y ⊢ τ : ⋆。设 ξ, ρ ⊨ Γ。我们需要证明 $(\!|J(c, a, b, q)|\!)_\rho \in [\![\tau[x := a, y := b, p := q]]\!]_{\xi\rho}$。和之前一样，我们将 $(\!|a|\!)_\rho$ 记作 a′，将 $[\![\sigma]\!]_{\xi\rho}$ 记作 σ′ 等等。在模型中解释 λx, y : σ. λp : x $=^{\sigma}$ y. τ，我们得到一个元素 $C \in \Pi_{x,y\in\sigma'}.[\![x =^{\sigma} y]\!]_{\xi\rho} \to \mathscr{P}$，需要证明 $(\!|J(c, a, b, q)|\!)_\rho \in C(a', b', q')$。我们将 C 扩展为 $C_0 \in \sigma' \to \sigma' \to A \to \mathscr{P}$，定义 $C_0(x, y, r) := /0$，其中 $r \notin [\![x =^{\sigma} y]\!]_{\xi\rho}$。由 $c' \in \Pi_{z\in\sigma'}.C(z, z, \mathrm{refl})$ 可得 $c' \in \Pi_{z\in\sigma'}.C_0(z, z, \mathrm{refl})$；由 $q' \in [\![a =^{\sigma} b]\!]_{\xi\rho}$，我们得出 $J\ c'a'b'q' \in C_0(a', b', q') = C(a', b', q')$，证毕。□

引理 5.8。在一个多集模型中，若 $=^{\sigma}$b 的解释是良定义的，则 UIP 成立。

证明。设 $p \in [\![a =^{\sigma} b]\!]_{\xi\rho}$，其中 a, b : σ。取 $C \in \sigma' \to \sigma' \to A \to \mathscr{P}$，使得 C(–, –, refl) = A，且当 r ≠ refl 时，C(–, –, r) = /0。于是存在 $t \in \Pi_{x\in\sigma'}.C(x, x, \mathrm{refl})$，并且对这样的 t，有 J t a′b′p ∈ C(a′, b′, p)。这意味着 C(a′, b′, p) ≠ /0，因此 p = refl 在 A 中成立。□

以下我们将范围限定为 WECA A := Λid。

引理 5.9。由 C := {refl}, $\mathscr{P}^{\Lambda\mathrm{id}}$({refl}) 生成的多集模型中，函数外延性不成立。

证明。考虑由以下方式定义的 f ,g : 布尔类型 → 布尔类型

$$\begin{aligned} f &:= \lambda b : \mathrm{bool}.\ b\ \mathrm{bool\ true\ false} \\ g &:= \lambda b : \mathrm{bool}.\ b\ \mathrm{bool\ true\ false\ bool\ true\ false}. \end{aligned}$$

我们证明 (1) $[\![\Pi b : \text{布尔类型}。 f\ x =^{bool} g\ x]\!]_{\xi\rho} \neq /0$，以及 (2) $[\![ f =^{bool\to bool} g]\!]_{\xi\rho} = /0$。

(1) bool 的解释是 $bool' := \cap_{x\in\mathscr{P}} X \to X \to X$，即 $\{K, K_*, \text{refl}\ \lambda x.\ \text{refl}\ \lambda x, y.\ \text{refl}\}$，其中 $K = \lambda x\ y.\ x = \text{《真》}$ 且 $K_* = \lambda x\ y.\ y = \text{《假》}$。模型中 f 和 g 的解释分别是 $f' := \lambda b.b\ K\ K_*$ 和 $g' := \lambda b.b\ K\ K_*\ K\ K_*$

我们声称，对于所有 $b \in bool'$，都有：

$$\text{refl} \in \bigcap_{C\in bool'\to bool'\to \mathbf{A}\to\mathscr{P}} \{q \mid \forall t \in \Pi_{x\in bool'}.C(x,x,\text{refl})\ [J\ t\ (f'\ b)\ (g'\ b)\ q \in C((f'\ b),(g'\ b),q)]\}.$$

这是因为对于 $b = K, b = K_*, b = \text{refl}, b = \lambda x.\ \text{refl}$ 和 $b = \lambda x,y.\ \text{refl}$，我们有 $f'b = g'b\ in \Lambda id$，因此对于任意 C，$J\ t\ (f'b)\ (g'b)\ \text{refl} = t\ (f'b) \in C((f'b),(g'b),\text{refl})$。所以，我们有一个项 $in\Pi_{b\in bool'}.[\![f\ x =^{bool} g\ x]\!]$。

(2) 若 $q \in [\![f =^{bool\to bool} g]\!]_{\xi\rho}$，则取一个 $D \in (\text{布尔类型}' \to \text{布尔类型}') \to (\text{布尔类型}' \to \text{布尔类型}') \to A \to \mathscr{P}$，使得对任意 $h \in \text{布尔类型}' \to \text{布尔类型}'$，都有 $D(h, h, \text{refl}) = A$，且 $D(f', g', _) = /0$。于是我们有一个项 $t \in \Pi_{h\in bool'\to \text{布尔类型}'} D(h,h,\text{refl})$，因此 $J\ t\ f'\ g'\ q \in D(f', g', q)$。矛盾，因为 $D(f', g', q) = /0$。所以 $[\![f =^{bool\to bool} g]\!]_{\xi\rho} = /0$。

(注：$f' \neq g'$ 在 $\Lambda id$ 中成立，这由 Church-Rosser性质 得出。由 $f' = g'$ 可知，$\Lambda id$ 中的所有项都相等：只需考虑 $f'\ I\ K$ 和 $g'\ I\ K$。) □

引理 5.10. 从 $C := \{\text{refl}\}$, $\mathscr{P}^{\Lambda id}(\{\text{refl}\})$ 生成的多集模型中，自然数的归纳不成立。

证明。证明基本上与 [12], 中相同，其中证明了对于多集合模型 $\mathscr{M}$，若 $\mathscr{M} \Vdash \text{ind}_{N,0,s}$，则 $[\![N]\!] = \{S^n 0 \mid n \in \mathbb{N}\}$。但在当前模型中，$[\![N]\!]$ 也包含 refl。□

最后，我们给出以下定理，它汇总了引理 5.7、5.8 和 5.10 的结果。

定理 5.11. 在加入恒等类型和 UIP 的 λP2 中，我们无法定义一种自然数数据类型，并证明其归纳原理。

## 6 结论与讨论

我们已经证明了多态依赖类型论 λP2 中的一些不可推导性结果。我们相信，将 λP2 替换为纯粹的构造演算 CC 后，这些证明仍然成立，无需任何改动。

在第 4 节中，我们考察了商类型和余归纳类型，尤其是流这一数据类型。将这些结果推广到以下方向会很有意思。

- 证明在 λP2 中不存在具有余归纳原理的流类型。我们的引理 4.3 仅表明众所周知的可定义流不具有余归纳原理，但或许可以为流设计一种更巧妙的编码。

- 定理 4.4 证明参数化商并不存在。另一方面，例 2.8 给出了一个非参数化商的具体例子，但那里的关系其实是平凡的：仅仅是恒等关系本身。是否还存在其他非参数化商（除了这个平凡的之外）？ 取 $R\ x\ y := \top$ 我们可以定义一个 cls 函数和 f，但 IndQuot 已经不清楚了。在 [20] 中已经研究了哪些商类型是可定义的、哪些不是。但这是在带有归纳类型的直谓类型论中，它具有更强的表达能力。

- 借助函数外延性，可定义商具有参数性，如 [8]所示。但要构造强商类型，我们还要加入 Σ-类型、恒等类型和 UIP。能否证明在 λP2 + FunExt 中不存在强商?

在第 5 节中，我们考察了 λP2 的扩展，其中可以定义归纳类型（包括商类型）和余归纳类型，并赋予它们相应的归纳原理和余归纳原理，如 [1, 8] 中所研究的那样。我们已经看到，如果加入 Σ-类型和带有 UIP 的恒等类型，仍然无法编码带有归纳原理的自然数。我们还需要加入函数外延性。这个定理并不局限于自然数，而是适用于任何代数数据类型，例如布尔值、列表、树等。结合 [1], 的结果，这意味着函数外延性对于定义归纳至关重要。

## 参考文献

[1] Steve Awodey、Jonas Frey & Sam Speight（2018）：（高阶）归纳类型的非谓词编码。载于：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，ACM，第76–85页，doi:10.1145/3209108.3209130。[2] Edwin Stewart Bainbridge、Peter J. Freyd、Andre Scedrov & Philip J. Scott（1990）：函子多态。Theor. Comput. Sci. 70(1)，第35–64页，doi:10.1016/0304-3975(90)90151-7。[3] Franco Barbanera & Stefano Berardi（2003）：多态性的全连续模型。Theor. Comput. Sci. 290(1)，第407–428页，doi:10.1016/S0304-3975(01)00379-6。[4] Henk Barendregt（1992）：带类型的 λ 演算。载于 S. Abramsky、D. M. Gabbay & T. S. E. Maibaum 编：《计算机科学逻辑手册》，牛津大学出版社，第117–309页。[5] Stefano Berardi（1993）：纯构造演算中数据类型的编码：一种语义论证。载于：第二届逻辑环境年度研讨会论文集，剑桥大学出版社（美国），第30–60页。[6] Stefano Berardi & Chantal Berline（2002）：System F 的 BetaEta 完备模型。Math. Struct. Comput. Sci. 12(6)，第823–874页，doi:10.1017/S0960129502003778。[7] Stefano Berardi & Chantal Berline（2004）：构造 System F 的连续网状模型。Theor. Comput. Sci. 315(1)，第3–34页，doi:10.1016/J.TCS.2003.11.011。[8] Steven Bronsveld、Herman Geuvers & Niels van der Weide（2025）：归纳类型与余归纳类型的非谓词编码。载于 Maribel Fern´andez 编：《第10届国际计算与演绎形式结构会议（FSCD 2025）》，莱布尼茨国际信息学论文集（LIPIcs）337，Schloss Dagstuhl – Leibniz-Zentrum f¨ur Informatik，德国达格施图尔，第11:1–11:22页，doi:10.4230/LIPIcs.FSCD.2025.11。[9] T. Coquand（1989）：构造演算的元数学研究。技术报告 RR-1088，法国国家信息与自动化研究所。可访问：https://inria.hal.science/inria-00075471。[10] Thierry Coquand & G´erard P. Huet（1988）：构造演算。Inf. Comput. 76(2/3)，第95–120页，doi:10.1016/0890-5401(88)90005-3。[11] Herman Geuvers（1996）：将二阶谓词逻辑的模型扩展为二阶依赖类型论的模型。载于 Dirk van Dalen & Marc Bezem 编：《计算机科学逻辑》，第10届国际研讨会（CSL ’96），EACSL 年会，乌得勒支，荷兰，1996年9月21日至27日，论文选集，计算机科学讲义1258，施普林格，第167–181页，doi:10.1007/ 3-540-63172-0_38。[12] Herman Geuvers（2001）：归纳原理无法在二阶依赖类型理论中导出。载于 Samson Abramsky 编：《带类型的 λ 演算及其应用》，第5届国际会议（TLCA 2001），波兰克拉科夫，计算机科学讲义2044，施普林格，第166–181页，doi:10.1007/ 3-540-45413-6_16。

[13] Jean-Yves Girard、Paul Taylor & Yves Lafont（1993）：证明与类型。剑桥大学出版社。[14] Ryu Hasegawa（1994）：参数多态中的范畴数据类型。Math. Struct. Comput. Sci. 4(1)，第71–109页，doi:10.1017/S0960129500000372。[15] Martin Hofmann（1995）：商类型的一个简单模型。载于 Mariangiola Dezani-Ciancaglini & Gordon D. Plotkin 编：《带类型的 λ 演算及其应用》，第二届带类型的 λ 演算及其应用国际会议（TLCA ’95），英国爱丁堡，1995年4月10日至12日，论文集，计算机科学讲义902，施普林格，第216–234页，doi:10.1007/BFB0014055。[16] Jan Willem Klop（1980）：组合子归约系统。博士论文，乌得勒支大学。[17] Giuseppe Longo & Eugenio Moggi（1991）：构造性自然演绎及其“Omega-Set”解释。Math. Struct. Comput. Sci. 1(2)，第215–254页，doi:10.1017/S0960129500001298。[18] G. Mitschke（1977）：组合逻辑与 λ-演算中的判别子。技术报告，预印本第336号，达姆施塔特工业大学。[19] Rob Nederpelt & Herman Geuvers（2014）：类型论与形式化证明：导论。剑桥大学出版社。[20] Li Nuo（2015）：类型论中的商类型。博士论文，诺丁汉大学。[21] Gordon D. Plotkin & Mart´ín Abadi（1993）：参数多态逻辑。载于 Marc Bezem & Jan Friso Groote 编：《带类型的 λ 演算及其应用》，带类型的 λ 演算及其应用国际会议（TLCA ’93），荷兰乌得勒支，1993年3月16日至18日，论文集，计算机科学讲义664，施普林格，第361–375页，doi:10.1007/BFB0037118。[22] Ivar Rummelhoff（2004）：PER 模型中的 Polynat。Theor. Comput. Sci. 316(1)，第215–224页，doi:10.1016/ J.TCS.2004.01.031。[23] Milena Stefanova & Herman Geuvers（1995）：构造演算的一个简单模型构造。载于 Stefano Berardi & Mario Coppo 编：《证明与程序的类型》，TYPES’95 国际研讨会，意大利都灵，1995年6月5日至8日，论文选集，计算机科学讲义1158，施普林格，第249–264页，doi:10.1007/3-540-61780-9_74。[24] Thomas Streicher（1992）：归纳原理与选择公理在纯构造演算中的独立性。Theor. Comput. Sci. 103(2)，第395–408页，doi:10.1016/0304-3975(92)90021-7。