# 资源有界 Martin-Löf 类型论：依赖类型的组合式成本分析

Mirco A. Mannucci[1] 和 Corey Thuro[2]

[1]HoloMathics, LLC mirco@holomathics.com[2]马里兰大学巴尔的摩县分校数学与统计系 cthuro1@umbc.edu

arXiv:2601.10772v1 [cs.LO] 15 Jan 2026

2025年12月

**关键词：**依赖类型、Martin-Löf类型理论、资源分析、分级模态、按大小索引的界、归纳族、预层语义

**2020年数学主题分类：**03B38（类型论）、03G30（范畴逻辑、托扑斯）、68Q25（算法分析）、18C50（范畴语义）

**ACM计算机分类系统:** 计算理论 →类型论；依赖类型；程序分析

**摘要**

我们将资源有界类型理论扩展到具有依赖类型的 Martin-Löf 类型论（MLTT），从而能够为归纳族上的程序建立按大小索引的成本界。我们的演算包含依赖函数类型 $\Pi_{x}^{b(x)} :_{A} B(x)$，其界 $b(x)$ 依赖于参数；依赖对类型 $\Sigma_{x:A} B(x)$，具有加法成本组合；以及归纳族（尤其是 Vec($A$, $n$) 和 Fin($n$)），其结构递归消去子带有依赖于规模的成本界。我们引入资源索引的宇宙层级 $\mathcal{U}_r$，其中 $r \in L$跟踪类型构造成本；并引入分级模态 $\Box_r$ 以进行可行性认证。我们的主要成果包括：(1) 成本可靠性定理，证明合成上界能够过近似操作成本，且界以大小索引的函数表示；(2) 在预层托扑斯 $\mathbf{Set}^{L}$ 中构造的语义模型，并将其扩展为依赖预层和理解结构；(3) 内涵片段的正则性；以及 (4) 句法模型的初始性。我们通过若干案例研究展示该框架，包括具有 $O(n)$ 界的按长度索引的向量操作，以及将 $O(\log n)$ 界表达在类型中的二分查找。这项工作弥合了依赖类型论与定量资源分析之间的鸿沟，使依赖于规模的算法能够获得经认证的成本界。

## 目录

**1 引言 3**1.1 与先前工作的关系.4

**2 设计原则** **4**

**3 资源有界 MLTT：演算** **4**

3.1 判断.4 3.2 上下文.5 3.3 宇宙.5 3.4 依赖函数类型（Π-类型）.5

3.5 依赖对类型（Σ-类型）.6 3.6 恒等类型.6 3.7 自然数.6 3.8 归纳族：向量.7 3.9 有限类型.7 3.10 安全索引.7 3.11 分级模态.8

**4 依赖于大小的界限综合** **8**

4.1 界表达式.8 4.2 带大小相关界的类型化.9 4.3 有界推理规则.9

**5 操作语义** **9**

5.1 值.10 5.2 求值规则.10

**6 元理论 10**

6.1 替换.10 6.2 类型保持.11 6.3 代价可靠性.11 6.4 正则性.12

**7 语义模型** **12**

7.1 群胚值预层.13 7.2 群胚情形下的内部格.13 7.3 作为群胚值预层的上下文.14 7.4 作为纤维化的依赖类型.14 7.5 代价自然变换.14 7.6 恒等类型的解释.14 7.7 Π-类型的解释.15 7.8 Σ-类型的解释.15 7.9 归纳类型的解释.15 7.10 宇宙的解释.16 7.11 语义可靠性.16 7.12 群胚上的资源有界 CwF.16 7.13 初始性.17

**8 案例研究** **18**

8.1 向量求和： $O(n)$.18 8.2 向量反转： $O(n)$.18 8.3 二分查找： $O(\log n)$.18 8.4 归并排序： $O(n \log\ n)$.19 8.5 矩阵乘法： $O(n^3)$.19

**9 实现 说明** **19**

9.1 Lean 4 集成.19 9.2 Agda 集成.20

**10 相关工作** **20**

**11 局限性与未来工作** **21**

# 1 引言

在先前的工作中， [1], 我们为简单类型λ演算开发了资源有界类型理论，并通过抽象资源格上的分级模态建立了组合式成本分析。尽管该框架能有效处理一阶程序，但许多算法的代价取决于其输入的大小： 排序需要 $O(n\log\ n)$ 时间，矩阵乘法需要 $O(n^3)$，而树遍历需要 $O(n)$，其中 $n$是节点数。

要在类型系统中表达这类依赖于大小的界限，我们需要依赖类型。本文将资源有界类型理论扩展到 Martin-Löf 类型论（MLTT），从而实现：

- **大小索引界限：** $\mathsf{Vec}(A, n) \to^{O(n)} B$ 表示一个作用于长度为 $n$ 的向量上的函数，其代价与 $n$ 呈线性关系。
- **依赖成本函数：** 依赖于项级值而不只是类型的界限 $b(x)$。
- **归纳族：** $\mathsf{Vec}(A, n)$、$\mathsf{Fin}(n)$ 及其他带索引类型，其消去子携带精确界限。
- **带资源跟踪的宇宙：** $\mathcal{U}_r$ 对形成代价至多为$r$的类型进行分类。

**动机示例**。考虑一个对长度索引向量求和的函数：

$$\mathsf{sum} : \Pi^{\perp}_{n:\mathbb{N}} \mathsf{Vec}(\mathbb{N}, n) \to^{c\cdot n+d} \mathbb{N}$$

该类型表达的是：给定任意一个 $n$ （接收代价为 $\perp$ ），以及一个长度为 $n$的向量，该函数会以 $c \cdot n + d$ 的代价返回一个自然数，其中常数为 $c$， $d$。这个界限与 $n$呈线性关系，而这种线性关系在类型中清晰可见。

**贡献**。

1. 一种带资源界的依赖类型论： Π-类型、 Σ-类型、恒等类型、宇宙 ，以及归纳族（§3）。2. 依赖于大小的界限综合：将界限表示为大小索引的函数（§4）。3. 内涵片段的代价可靠性（定理 6.3）。4. 具有依赖结构的预层语义模型（§7）。5. 正则性（定理 6.5）和初始性（定理 7.25）。6. 案例研究：向量操作、二分查找，以及具有经验证界限的排序（§8）。

**范围**。 我们研究不含单值性或高阶归纳类型的内涵 MLTT。恒等类型 $\mathsf{Id}_A(x,\ y)$ 具有标准的 J 消去子；我们不讨论同伦论问题。扩展至同伦类型论属于未来工作。

### 1.1 与前期工作的关系

本文是系列论文中的第二篇：

**1.RB-TT** [1]: 资源有界简单类型 λ 演算。建立了基本框架：抽象资源格、分级模态 $\Box_r$、代价可靠性、在$\mathbf{Set}^L$中的预层语义以及初始性。**2. RB-MLTT** （本文）：扩展到依赖类型。增加了带有依赖于大小的界限的 Π-类型、 Σ-类型、恒等类型、宇宙 $\mathcal{U}_r$以及带消去子的归纳族。**3. RB-HoTT** （未来工作）：扩展到同伦类型论，包括资源有界路径和高阶结构。

我们假设读者熟悉 [1], 中的基本框架，尤其是资源格（$L$， $\preceq$， $\oplus$， $\sqcup$， $\bot$）以及类型判断 $\Gamma \vdash_{r;\,b} t :\ A$。

## 2 设计原则

我们将 [1] 中的原则扩展到依赖类型环境：

**P1. 抽象而非具体化**。资源仍然是抽象格 $L$中的元素。**P2. 将预算作为一等公民**。资源界出现在类型和判断中，并且现在依赖于项级值。**P3. 组合规则**。界限合成反映项的结构，其中依赖类型使用依赖界。**P4. 大小索引界限**。界可以是输入的函数 $b :\ A \to L$ ，从而支持 $O(n)$、$O(n^2)$ 和 $O(\log n)$ 表达式。**P5. 保持单调性**。如果 $r_1 \preceq r_2$，则$\Box_{r_1} A \to \Box_{r_2} A$。**P6. 宇宙跟踪代价**。类型形成本身可能产生代价； $\mathcal{U}_r$ 将可在预算 $r$内形成的类型分类。

**记号**。 我们用 $L$ 表示资源格；用 $r, s \in L$ 表示资源元素；用 $\preceq$ 表示序；$\oplus$ 表示顺序组合； $\sqcup$ 表示分支（连接）； $\bot$表示零。对于依赖于大小的界限，当界限依赖于某个项时，我们记作 $b(x)$ 或 $b(n)$。

## 3 资源受限 MLTT：演算

### 3.1 判断

我们有四种判断形式：

1. $\Gamma \vdash_r \mathsf{ok}$ — 上下文 $\Gamma$ 在预算 $r$下是良构的2. $\Gamma \vdash_{r;\,b} A\ \mathsf{type}$ — $A$ 是一个类型，其形成代价为 $b$3. $\Gamma \vdash_{r;\,b} t :\ A$ — 项 $t$ 具有类型 $A$ ，代价为 $b$4. $\Gamma \vdash_{r;\,b} A \equiv B$ — 类型 $A$ 和 $B$ 定义相等（检查代价为 $b$ ）

下标 $r$ 表示环境预算； $b$ 表示满足 $b \preceq r$ 的合成上界 .

**备注 3.1**（环境预算的作用）。 本文中，关于资源使用的所有实质性信息都由合成上界 $b$ 承载；环境预算 $r$ 则作为参数保留，以匹配 *[1]* 的记号和元理论。我们不 使用 $r$ 来截断推导或强制施加全局约束：每条规则的表述都确保，只要一个判断可导出，其上界 $b$ 就已经满足 $b \preceq r$。未来工作或可使用 $r$ 来表达全局预算策略（例如，禁止合成上界超过 $r$ 的推导），但在本文中， $r$ 在操作层面不起作用。

## 3.2 上下文

$$\frac{}{\cdot \vdash_r \mathsf{ok}} \quad (\text{Ctx-Empty}) \qquad \frac{\Gamma \vdash_r \mathsf{ok} \qquad \Gamma \vdash_{r;\,b} A\ \mathsf{type}}{\Gamma, x : A \vdash_r \mathsf{ok}} \quad (\text{Ctx-Ext})$$

## 3.3 宇宙

我们引入一个资源索引宇宙层级 $\mathcal{U}_r$ for $r \in L$。

**定义 3.2 (**资源索引宇宙**)**。 $\mathcal{U}_r$ 是类型的宇宙，其编码可在代价不超过 $r$ 的情况下构造。我们有：

- $\mathcal{U}_r$： $\mathcal{U}_{r \oplus \delta_{\mathcal{U}}}$ *(*宇宙构成一个代价为 $\delta_{\mathcal{U}}$ 的层级*)*
- 如果 $r_1 \preceq r_2$，则 $\mathcal{U}_{r_1} \subseteq \mathcal{U}_{r_2}$ （累积性）（即，更大的索引至少允许同样多的代码 *s)*

$$\frac{\Gamma \vdash_r \mathsf{ok} \qquad s \preceq r}{\Gamma \vdash_{r;\,\delta_{\mathcal{U}}} \mathcal{U}_s\ \mathsf{type}} \quad (\mathcal{U}\text{-Form}) \qquad \frac{\Gamma \vdash_{r;\,b} A : \mathcal{U}_s}{\Gamma \vdash_{r;\,b \oplus \delta_{\mathsf{El}}} \mathsf{El}(A)\ \mathsf{type}} \quad (\mathsf{El}\text{-Form})$$

$$\frac{\Gamma \vdash_{r;\,b} A : \mathcal{U}_{s_1} \qquad s_1 \preceq s_2}{\Gamma \vdash_{r;\,b} A : \mathcal{U}_{s_2}} \quad (\mathcal{U}\text{-Cumul})$$

El 算子将代码 $A : \mathcal{U}_s$ 解码为实际类型。实践中，在无歧义时，我们通常省略 El。

## 3.4 依赖函数类型（Π-类型）

关键创新在于，界可以依赖于参数。

**定义 3.3**（带界依赖函数类型）。 对于类型 $A$ 以及 $B(x)$ 在 $A$ 上的族，以及上界函数 $b : A \to L$：

$$\Pi^{b(x)}_{x:A} B(x)$$

是函数类型：给定 $a$： $A$，产生 $B(a)$，其代价为 $b(a)$。

$$\frac{\Gamma \vdash_{r;\,b_A} A\ \mathsf{type} \qquad \Gamma, x : A \vdash_{r;\,b_B} B\ \mathsf{type} \qquad \Gamma, x : A \vdash_{r;\,b(x)} b(x) : L}{\Gamma \vdash_{r;\,b_A \oplus \delta_\Pi} \Pi^{b(x)}_{x:A} B\ \mathsf{type}} \quad (\Pi\text{-Form})$$

$$\frac{\Gamma, x : A \vdash_{r;\,b(x)} t : B}{\Gamma \vdash_{r;\,\bot} \lambda x.t : \Pi^{b(x)}_{x:A} B} \quad (\Pi\text{-Intro}) \qquad \frac{\Gamma \vdash_{r;\,b_f} f : \Pi^{b(x)}_{x:A} B \qquad \Gamma \vdash_{r;\,b_a} a : A}{\Gamma \vdash_{r;\,b_f \oplus b_a \oplus b(a) \oplus \delta_{\mathsf{app}}} f\,a : B[x := a]} \quad (\Pi\text{-Elim})$$

$$\frac{\Gamma, x : A \vdash_{r;\,b(x)} t : B \qquad \Gamma \vdash_{r;\,b_a} a : A}{\Gamma \vdash_{r;\,b_a \oplus b(a) \oplus \delta_\beta} (\lambda x.t)\,a \equiv t[x := a] : B[x := a]} \quad (\Pi\text{-}\beta)$$

**关键洞见**。 应用规则（Π-消去）包含 $b(a)$——即在特定参数 $a$ 处求值的上界。这使得依赖于大小的成本跟踪成为可能：如果 $b(n) = c{\cdot}n$，那么将函数应用于参数 $n$ 的代价为 $c \cdot n$。

**非依赖情形**。当 $B$ 不依赖于 $x$，且 $b(x) = b$ 为常量时，我们就得到简单的函数类型 $A \to^b B$。

### 3.5 依赖对类型（Σ-类型）

$$\frac{\Gamma \vdash_{r;\, b_A} A\ \mathsf{type} \qquad \Gamma, x : A \vdash_{r;\, b_B} B\ \mathsf{type}}{\Gamma \vdash_{r;\, b_A \oplus \delta_\Sigma} \Sigma_{x:A} B\ \mathsf{type}} \quad (\Sigma\text{-Form})$$

$$\frac{\Gamma \vdash_{r;\, b_a} a : A \qquad \Gamma \vdash_{r;\, b_b} b : B[x := a]}{\Gamma \vdash_{r;\, b_a \oplus b_b} (a, b) : \Sigma_{x:A} B} \quad (\Sigma\text{-Intro}) \qquad \frac{\Gamma \vdash_{r;\, b_p} p : \Sigma_{x:A} B}{\Gamma \vdash_{r;\, b_p \oplus \delta_{\pi_1}} \pi_1(p) : A} \quad (\Sigma\text{-Elim}_1)$$

$$\frac{\Gamma \vdash_{r;\, b_p} p : \Sigma_{x:A} B}{\Gamma \vdash_{r;\, b_p \oplus \delta_{\pi_2}} \pi_2(p) : B[x := \pi_1(p)]} \quad (\Sigma\text{-Elim}_2)$$

配对构造的代价以加法方式组合；投影会增加少量开销 $\delta_{\pi_i}$ .

### 3.6 恒等类型

$$\frac{\Gamma \vdash_{r;\, b_A} A\ \mathsf{type} \qquad \Gamma \vdash_{r;\, b_x} x : A \qquad \Gamma \vdash_{r;\, b_y} y : A}{\Gamma \vdash_{r;\, b_A \oplus b_x \oplus b_y \oplus \delta_{\mathsf{Id}}} \mathsf{Id}_A(x, y)\ \mathsf{type}} \quad (\mathsf{Id}\text{-Form})$$

$$\frac{\Gamma \vdash_{r;\, b_a} a : A}{\Gamma \vdash_{r;\, b_a \oplus \delta_{\mathsf{refl}}} \mathsf{refl}_a : \mathsf{Id}_A(a, a)} \quad (\mathsf{Id}\text{-Intro})$$

J 消去子的代价取决于动机和方法：

$$\frac{\Gamma \vdash_{r;\, b_p} p : \mathsf{Id}_A(x, y) \qquad \Gamma, z : A, w : \mathsf{Id}_A(x, z) \vdash_{r;\, b_C} C\ \mathsf{type} \qquad \Gamma \vdash_{r;\, b_d} d : C[z := x, w := \mathsf{refl}_x]}{\Gamma \vdash_{r;\, b_p \oplus b_d \oplus \delta_{\mathsf{J}}} \mathsf{J}(p, d) : C[z := y, w := p]} \quad (\mathsf{J})$$

$$\frac{}{\Gamma \vdash_{r;\, \delta_{\mathsf{J}\beta}} \mathsf{J}(\mathsf{refl}_a, d) \equiv d : C[z := a, w := \mathsf{refl}_a]} \quad (\mathsf{J}\text{-}\beta)$$

### 3.7 自然数

$$\frac{\Gamma \vdash_r \mathsf{ok}}{\Gamma \vdash_{r;\, \delta_{\mathbb{N}}} \mathbb{N}\ \mathsf{type}} \quad (\mathbb{N}\text{-Form}) \qquad \frac{\Gamma \vdash_r \mathsf{ok}}{\Gamma \vdash_{r;\, \delta_{\mathsf{Z}}} \mathsf{Z} : \mathbb{N}} \quad (\mathbb{N}\text{-Intro}_{\mathsf{Z}}) \qquad \frac{\Gamma \vdash_{r;\, b} n : \mathbb{N}}{\Gamma \vdash_{r;\, b \oplus \delta_{\mathsf{S}}} \mathsf{S}(n) : \mathbb{N}} \quad (\mathbb{N}\text{-Intro}_{\mathsf{S}})$$

消去子的代价取决于数值的高度：

$$\frac{\begin{array}{c}\Gamma \vdash_{r;\, b_n} n : \mathbb{N} \\ \Gamma, m : \mathbb{N} \vdash_{r;\, b_C} C\ \mathsf{type} \qquad \Gamma \vdash_{r;\, b_z} c_z : C[\mathsf{Z}] \qquad \Gamma, m : \mathbb{N}, ih : C[m] \vdash_{r;\, b_s(m)} c_s : C[\mathsf{S}(m)]\end{array}}{\Gamma \vdash_{r;\, b_n \oplus b_z \oplus \delta_{\mathsf{natrec}} \oplus \mathsf{sum}_{i<n}(b_s(i) \oplus \delta_{\mathsf{natrec}})} \mathsf{natrec}(n, c_z, c_s) : C[n]} \quad (\mathbb{N}\text{-Elim})$$

这里，$\mathsf{sum}_{i<n}(b_s(i) \oplus \delta_{\mathsf{natrec}})$ 是逐步成本的总和（包括消去子开销）。当 $b_s$ 为常数时，这就变成 $n \cdot (b_s \oplus \delta_{\mathsf{natrec}})$，而完整界还包括额外的基础开销 $\delta_{\mathsf{natrec}}$。

$$\frac{}{\mathsf{natrec}(\mathsf{Z}, c_z, c_s) \equiv c_z} \quad (\mathbb{N}\text{-}\beta_{\mathsf{Z}})$$

$$\frac{}{\mathsf{natrec}(\mathsf{S}(n), c_z, c_s) \equiv c_s[m := n, ih := \mathsf{natrec}(n, c_z, c_s)]} \quad (\mathbb{N}\text{-}\beta_{\mathsf{S}})$$

### 3.8 归纳族：向量

长度索引向量是具有依赖于大小的界限的归纳族的典型示例。

**定义 3.4** ($\mathsf{Vec}(A, n)$)。 对于类型 $A$ $a$ 和 $n\colon \mathbb{N}$，$\mathsf{Vec}(A, n)$ 是长度恰好为 $n$ 的向量类型。

$$\frac{\Gamma \vdash_{r;\, b_A} A\ \mathsf{type} \qquad \Gamma \vdash_{r;\, b_n} n : \mathbb{N}}{\Gamma \vdash_{r;\, b_A \oplus b_n \oplus \delta_{\mathsf{Vec}}} \mathsf{Vec}(A, n)\ \mathsf{type}} \quad (\textsc{Vec-Form}) \qquad \frac{\Gamma \vdash_{r;\, b_A} A\ \mathsf{type}}{\Gamma \vdash_{r;\, \delta_{\mathsf{nil}}} \mathsf{nil} : \mathsf{Vec}(A, \mathsf{Z})} \quad (\textsc{Vec-Intro}_{\mathsf{nil}})$$

$$\frac{\Gamma \vdash_{r;\, b_a} a : A \qquad \Gamma \vdash_{r;\, b_v} v : \mathsf{Vec}(A, n)}{\Gamma \vdash_{r;\, b_a \oplus b_v \oplus \delta_{\mathsf{cons}}} \mathsf{cons}(a, v) : \mathsf{Vec}(A, \mathsf{S}(n))} \quad (\textsc{Vec-Intro}_{\mathsf{cons}})$$

消去子的代价与长度呈线性关系：

$$\frac{\Gamma \vdash_{r;\, b_v} v\colon \mathsf{Vec}(A, n)\ \Gamma, m\colon \mathbb{N}, w\colon \mathsf{Vec}(A, m) \vdash_{r;\, b_C} C\text{类型}\ \Gamma \vdash_{r;\, b_{\mathsf{nil}}} c_{\mathsf{nil}}\colon C[\mathsf{Z}, \text{空列表}]\Gamma,\ m\colon \mathbb{N}, a\colon A, w\colon \mathsf{Vec}(A, m)\ ih\colon C[m, w] \vdash_{r;\, b_{\mathsf{cons}}(m)} c_{\mathsf{cons}}\colon C[\mathsf{S}(m), \mathsf{cons}(a, w)]}{\Gamma \vdash_{r;\, b_v \oplus b_{\mathsf{nil}} \oplus \delta_{\mathsf{vecrec}} \oplus \mathsf{sum}_{i<n}(b_{\mathsf{cons}}(i) \oplus \delta_{\mathsf{vecrec}})}\ \mathsf{vecrec}(v, c_{\mathsf{nil}}, c_{\mathsf{cons}}) : C[n, v]} \quad \text{(Vec-消去)}$$

对于常量 $b_{\mathsf{cons}}$，合成上界关于 $n$ 呈线性关系（特别是包含每步开销 $\delta_{\mathsf{vecrec}}$），因此 $O(n)$。

### 3.9 有限类型

$\mathsf{Fin}(n)$ 是恰好包含 $n$ 个元素的类型，适用于有界索引。

$$\frac{\Gamma \vdash_{r;\, b_n} n : \mathbb{N}}{\Gamma \vdash_{r;\, b_n \oplus \delta_{\mathsf{Fin}}} \mathsf{Fin}(n)\ \mathsf{type}} \quad (\mathsf{Fin\text{-}Form}) \qquad \frac{\Gamma \vdash_{r;\, b_n} n : \mathbb{N}}{\Gamma \vdash_{r;\, \delta_{\mathsf{fz}}} \mathsf{fzero} : \mathsf{Fin}(\mathsf{S}(n))} \quad (\mathsf{Fin\text{-}Intro}_{\mathsf{fz}})$$

$$\frac{\Gamma \vdash_{r;\, b_i} i : \mathsf{Fin}(n)}{\Gamma \vdash_{r;\, b_i \oplus \delta_{\mathsf{fs}}} \mathsf{fsucc}(i) : \mathsf{Fin}(\mathsf{S}(n))} \quad (\mathsf{Fin\text{-}Intro}_{\mathsf{fs}})$$

### 3.10 安全索引

借助 Vec 和 Fin，我们可以定义具有精确边界的安全索引：

**示例 3.5** （安全向量查找）。

$$\mathsf{lookup} : \Pi^{\perp}_{n:\mathbb{N}} \Pi^{\perp}_{A:\mathcal{U}} \mathsf{Vec}(A, n) \to^{c_1} \mathsf{Fin}(n) \to^{c_2 \cdot k} A$$

其中 $k$ 是索引值。代价取决于我们在向量中的查找位置。

更确切地说，使用直接索引时：

$$\mathsf{lookup} : \Pi^{\perp}_{n:\mathbb{N}} \Pi^{\perp}_{A:\mathcal{U}} \mathsf{Vec}(A, n) \to \mathsf{Fin}(n) \to^{c} A$$

若我们有 $c$ $O$（1）数组访问，其代价为常数。

### 3.11 分级模态

我们保留了来自 $\Box_r$ [1]的分级模态:

$$\frac{\Gamma \vdash_{r;\, b} A\ \mathsf{type} \qquad s \in L}{\Gamma \vdash_{r;\, b \oplus \delta_{\Box}} \Box_s A\ \mathsf{type}}\ (\Box\text{-Form}) \qquad \frac{\Gamma \vdash_{r;\, b} t : A \qquad b \preceq s}{\Gamma \vdash_{r;\, b} \mathsf{box}_s(t) : \Box_s A}\ (\Box\text{-Intro})$$

$$\frac{\Gamma \vdash_{r;\, b} t : \Box_s A}{\Gamma \vdash_{r;\, b \oplus \delta_{\mathsf{unbox}}} \mathsf{unbox}(t) : A}\ (\Box\text{-Elim}) \qquad \frac{\Gamma \vdash_{r;\, b} t : \Box_{s_1} A \qquad s_1 \preceq s_2}{\Gamma \vdash_{r;\, b} t : \Box_{s_2} A}\ (\Box\text{-Mono})$$

### RB-TT 作为 RB-MLTT 的片段

[1] 原始的资源有界类型理论 RB–TT 可被视为当前系统的简单类型片段。

**命题 3.6** （RB–TT 片段）。 如果我们将 *RB–MLTT* 限制为:

- 非依赖函数类型和积类型（不包含依赖于项的 Π-类型或 Σ-类型），
- 没有宇宙和归纳族（仅包含 *[1]* 的基础类型），
- 与 *[1],* 中相同的分级模态 $\Box_r$ 和格 $(L, \preceq, \oplus, \bot)$

并将判断 $\Gamma \vdash_{r;\, b} t : A$ 读作等级为 $b$ 的 *RB–TT* 类型推导判断，那么 *RB–MLTT* 的类型规则就归约为 *RB–TT* 的类型规则。此外，操作成本语义和可靠性定理也特化为 *[1]* 中的相应结果。

证明概要。 RB–MLTT 规则由 RB–TT 规则通过以下方式得到:

1. 添加依赖上下文扩展和依赖类型构造子（Π、 Σ、恒等类型、宇宙、归纳族）； 2. 将相同的格（$L, \preceq, \oplus, \bot$）和分级模态 $\Box_r$ 贯穿扩展后的语法； 3. 细化成本界，使得在简单类型片段中，每条规则都与其对应的 RB–TT 规则一致。

消去依赖关系和额外的类型构造子后，恰好得到 RB–TT 系统。在语义方面，当我们只考虑离散群胚和非依赖类型时，第 7 节中构建的群胚值预层模型会限制为 [1] 的集合值预层模型，因此，成本可靠性和正则性也特化为先前的结果。 □

## 4 大小依赖的界限合成

RB-MLTT 的关键特性是，界限可以是大小索引的函数 。

### 4.1 界表达式

**定义 4.1 (**界语言)。 界限 $b$ 是 $L$ 的元素，或是映射到 $L$ 的函数 :

$$b ::= r \mid \bot \mid b_1 \oplus b_2 \mid b_1 \sqcup b_2 \mid c \cdot n \mid b(t) \mid \mathsf{sum}_{i<n} b(i)$$

其中 $r \in L$, $c$ 是一个常数， $n$ 是一个自然数项，而 $b(t)$ 是界应用。我们还使用导出记法 $n \otimes b$ 表示将 $\oplus$ 对界 $b$ 应用 $n$ 次: $0 \otimes b \triangleq \bot$ *and* $(n{+}1) \otimes b \triangleq b \oplus (n \otimes b)$。我们还使用导出记法 $n \otimes b$ 表示将 $\oplus$ 对界 $b$ 应用 $n$ 次: $0 \otimes b \triangleq \bot$ *and* $(n{+}1) \otimes b \triangleq b \oplus (n \otimes b)$。

**例4.2**（常见界模式）。• $O(1)$：常数界 $c \in L$

- $O(n)$：线性界 $c \cdot n$，大小为 $n$
- $O(n^2)$：二次界 $c \cdot n \cdot n$
- $O(\log n)$：对数界 $c \cdot \lceil \log_2 n \rceil$
- $O(n\log\ n)$：线性对数界 $c \cdot n \cdot \lceil \log_2 n \rceil$

### 4.2 依赖于大小的界限的类型推导

**例4.3**（具有线性界的向量求和）。

$$\begin{aligned}&\mathsf{sum} : \Pi^{\perp}_{n:\mathbb{N}} \mathsf{Vec}(\mathbb{N}, n) \to^{c\cdot n+d} \mathbb{N}\\ &\mathsf{sum} = \lambda n.\, \lambda v.\, \mathsf{vecrec}(v, \mathsf{Z}, \lambda m.\lambda a.\lambda w.\lambda ih.\, a + ih)\end{aligned}$$

界 $c \cdot n + d$ 由以下部分合成：

- 基例：返回 Z 的代价为 $d_0$
- 递归情形：每个元素的代价为 $c$（加法）
- 总计：$d_0 + n \cdot c = c \cdot n + d$，其中 $d = d_0$

**例4.4**（具有线性界的向量映射）。

$$\begin{aligned}&\mathsf{map} : \Pi^{\perp}_{n:\mathbb{N}} \Pi^{\perp}_{A,B:\mathcal{U}} (A \to^{c_f} B) \to \mathsf{Vec}(A, n) \to^{c_f \cdot n+d} \mathsf{Vec}(B, n)\\ &\mathsf{map} = \lambda n.\lambda A.\lambda B.\lambda f.\lambda v.\, \mathsf{vecrec}(v, \mathsf{nil}, \lambda m.\lambda a.\lambda w.\lambda ih.\, \mathsf{cons}(f\ a, ih))\end{aligned}$$

界为 $c_f \cdot n + d$：我们对每个 $n$ 元素应用 $f$（代价为 $c_f$）。

### 4.3 界推断规则

对于结构化递归，可以系统地计算界：

**命题 4.5**（线性递归界）。若 natrec$(n, c_z, c_s)$ 具有：

- 基例代价 $b_z$
- 步进情形代价 $b_s$（每步为常数）

那么 总界为 $b_z \oplus (n \otimes (b_s \oplus \delta_{\mathsf{natrec}})) \oplus \delta_{\mathsf{natrec}}$，其中 $n \otimes b_s$ 表示 $n$ 重复应用 $\oplus$ 次于 $b_s$。

**命题 4.6 (**分治界**)**。若一个输入规模为 $n$ 的函数具有：

- 划分代价 $d$
- 规模为 $\lfloor n/2 \rfloor$的递归调用
- 合并代价 $c$

则总界满足 $T(n) = d \oplus T(\lfloor n/2 \rfloor) \oplus c$，从而得到 $T(n) = O(\log\ n) \cdot (d \oplus c)$。

## 5 操作语义

我们将大步语义从 [1] 扩展到依赖类型。

### 5.1 值

**定义 5.1 (值)。**

$$\begin{aligned} v \in \mathsf{Val} ::= {} & \lambda x.t \mid (v_1, v_2) \mid \mathsf{refl}_v \mid \mathsf{Z} \mid \mathsf{S}(v) \\ & \mid \mathsf{nil} \mid \mathsf{cons}(v_1, v_2) \mid \mathsf{fzero} \mid \mathsf{fsucc}(v) \\ & \mid \mathsf{box}_s(v) \mid \mathcal{U}_r \mid \Pi^b_{x:A} B \mid \dots \end{aligned}$$

### 5.2 求值规则

我们用 $t \Downarrow v \triangleright k$ 表示“$t$ 求值为 $v$，代价为 $k$”。

$$\frac{}{v \Downarrow v \triangleright \bot} \ (\mathrm{Val}) \qquad \frac{f \Downarrow \lambda x.t \triangleright k_f \qquad a \Downarrow v_a \triangleright k_a \qquad t[x := v_a] \Downarrow v \triangleright k_t}{f\ a \Downarrow v \triangleright k_f \oplus k_a \oplus k_t \oplus \delta_{\mathsf{app}}} \ (\mathrm{App})$$

$$\frac{t \Downarrow v_t \triangleright k_t \qquad u \Downarrow v_u \triangleright k_u}{(t, u) \Downarrow (v_t, v_u) \triangleright k_t \oplus k_u} \ (\mathrm{Pair}) \qquad \frac{p \Downarrow (v_1, v_2) \triangleright k}{\pi_1(p) \Downarrow v_1 \triangleright k \oplus \delta_{\pi_1}} \ (\mathrm{Fst})$$

$$\frac{p \Downarrow (v_1, v_2) \triangleright k}{\pi_2(p) \Downarrow v_2 \triangleright k \oplus \delta_{\pi_2}} \ (\mathrm{Snd})$$

**自然数递归。**

$$\frac{n \Downarrow \mathsf{Z} \triangleright k_n \qquad c_z \Downarrow v \triangleright k_z}{\mathsf{natrec}(n, c_z, c_s) \Downarrow v \triangleright k_n \oplus k_z \oplus \delta_{\mathsf{natrec}}} \ (\mathsf{natrec\text{-}Z})$$

$$\frac{n \Downarrow \mathsf{S}(v_n) \triangleright k_n \qquad \mathsf{natrec}(v_n, c_z, c_s) \Downarrow v_{ih} \triangleright k_{ih} \qquad c_s[m := v_n, ih := v_{ih}] \Downarrow v \triangleright k_s}{\mathsf{natrec}(n, c_z, c_s) \Downarrow v \triangleright k_n \oplus k_{ih} \oplus k_s \oplus \delta_{\mathsf{natrec}}} \ (\mathsf{natrec\text{-}S})$$

**向量递归。**

$$\frac{v \Downarrow \mathsf{nil} \triangleright k_v \qquad c_{\mathsf{nil}} \Downarrow u \triangleright k_{\mathsf{nil}}}{\mathsf{vecrec}(v, c_{\mathsf{nil}}, c_{\mathsf{cons}}) \Downarrow u \triangleright k_v \oplus k_{\mathsf{nil}} \oplus \delta_{\mathsf{vecrec}}} \ (\mathsf{vecrec\text{-}nil})$$

$$\frac{\begin{array}{c} v \Downarrow \mathsf{cons}(v_a, v_w) \triangleright k_v \\ \mathsf{vecrec}(v_w, c_{\mathsf{nil}}, c_{\mathsf{cons}}) \Downarrow v_{ih} \triangleright k_{ih} \qquad c_{\mathsf{cons}}[a := v_a, w := v_w, ih := v_{ih}] \Downarrow u \triangleright k_c \end{array}}{\mathsf{vecrec}(v, c_{\mathsf{nil}}, c_{\mathsf{cons}}) \Downarrow u \triangleright k_v \oplus k_{ih} \oplus k_c \oplus \delta_{\mathsf{vecrec}}} \ (\mathsf{vecrec\text{-}cons})$$

**恒等消去。**

$$\frac{p \Downarrow \mathsf{refl}_v \triangleright k_p \qquad d \Downarrow v_d \triangleright k_d}{\mathsf{J}(p, d) \Downarrow v_d \triangleright k_p \oplus k_d \oplus \delta_{\mathsf{J}}} \ (\mathsf{J\text{-}refl})$$

## 6 元理论

### 6.1 替换

**引理 6.1 (替换)。** 若 $\Gamma$、$x :\ A \vdash_{r;b} t :\ B$，且 $\Gamma \vdash_{r;b_a} a :\ A$，则 $\Gamma \vdash_{r;b[x:=a]} t[x :\ = a] : B[x :\ = a]$。

证明。 对类型推导进行归纳，并针对每个类型构造子分别讨论。界限 $b$可能依赖于 $x$；替换之后，$b[x{:=}\,a]$ 就是在 $a$处求值得到的界限。 □

## 6.2 类型保持

**定理 6.2** (保持性)。 若 $\cdot \vdash_{r;b} t : A$ 且 $t \Downarrow v \triangleright k$，则 $\cdot \vdash_{r;b'} v : A$ 对于某个 $b' \preceq b$。证明。对求值推导进行归纳。

**情形（应用）**。$t = f\,a$ with $f \Downarrow \lambda x.s \triangleright k_f$, $a \Downarrow v_a \triangleright k_a$, $s[x := v_a] \Downarrow v \triangleright k_s$. 由反演可得：$\cdot \vdash_{r;b_f} f : \Pi_x^{b(x)} {}_{:A}\, B$ 以及 $\cdot \vdash_{r;b_a} a : A$. 对 $f$ 应用归纳假设，得 $\cdot \vdash_{r;b'_f} \lambda x.s : \Pi_x^{b(x)} {}_{:A}\, B$ with $b'_f \preceq b_f$. 对 $a$ 应用归纳假设，得 $\cdot \vdash_{r;b'_a} v_a : A$ with $b'_a \preceq b_a$. 由引理 6.1 可得：$\cdot \vdash_{r;b(v_a)} s[x := v_a] : B[x := v_a]$. 由归纳假设可得：$\cdot \vdash_{r;b'} v : B[x := v_a]$ with $b' \preceq b(v_a)$.

**情形**（natrec-S）。根据递归调用和步骤的归纳假设，代价会正确累积。

**其他情形**。类似地，使用归纳假设和求值规则。 □

## 6.3 代价可靠性

**定理 6.3** (代价可靠性)。 若 $\cdot \vdash_{r;b} t : A$，则存在 $v \in$ 值集合 和 $k \in L$，使得$t \Downarrow v \triangleright k$ 且 $k \preceq b$。

证明。对类型推导进行归纳。

**基例**。

- **变量：** 与空上下文矛盾。
- **λ 抽象：** $\lambda x.t$ 是一个值；$\lambda x.t \Downarrow \lambda x.t \triangleright \bot$ 和 $\bot \preceq b$。
- **零、空列表、自反性：**带代价的值 $\bot$。

**归纳情形（应用）**。$t = f\,a$ 其中：

$$\begin{aligned} &\cdot \vdash_{r;b_f} f : \Pi_{x:A}^{b(x)} B \\ &\cdot \vdash_{r;b_a} a : A \\ &b = b_f \oplus b_a \oplus b(a) \oplus \delta_{\mathsf{app}} \end{aligned}$$

由归纳假设：$f \Downarrow \lambda x.s \triangleright k_f$ 其中 $k_f \preceq b_f$。
由归纳假设：$a \Downarrow v_a \triangleright k_a$ 其中 $k_a \preceq b_a$。
由类型推导 $\lambda x.s$：$x : A \vdash_{r;b(x)} s : B$，因此 $\cdot \vdash_{r;b(v_a)} s[x := v_a] : B[x := v_a]$。由归纳假设：$s[x := v_a] \Downarrow v \triangleright k_s$ 其中 $k_s \preceq b(v_a)$。由（应用）：$f\,a \Downarrow v \triangleright k_f \oplus k_a \oplus k_s \oplus \delta_{\mathsf{app}}$。由单调性：$k_f \oplus k_a \oplus k_s \oplus \delta_{\mathsf{app}} \preceq b_f \oplus b_a \oplus b(v_a) \oplus \delta_{\mathsf{app}} = b$。

**归纳情形**（natrec）。令 $t = \mathsf{natrec}(n, c_z, c_s)$ 满足：

$$\begin{aligned} &\cdot \vdash_{r;b_n} n : \mathbb{N} \\ &\cdot \vdash_{r;b_z} c_z : C[\mathsf{Z}] \\ &m : \mathbb{N}, ih : C[m] \vdash_{r;b_s(m)} c_s : C[\mathsf{S}(m)] \end{aligned}$$

根据归纳假设：$n \Downarrow v_n \triangleright k_n$ 和 $k_n \preceq b_n$ 与 $v_n$ 都是数值。
**子情形** $v_n = \mathsf{Z}$**：** 根据归纳假设，$c_z \Downarrow v \triangleright k_z$ 与 $k_z \preceq b_z$。
总计：$k_n \oplus k_z \oplus \delta_{\mathsf{natrec}} \preceq b_n \oplus b_z \oplus \delta_{\mathsf{natrec}} \preceq b$。
**子情形** $v_n = \mathsf{S}^m(\mathsf{Z})$**：** 对 $m$ 进行归纳：

- 递归调用：代价有界于 $b_z \oplus \sum_{i=0}^{m-1} b_s(i) \oplus (m) \cdot \delta_{\mathsf{natrec}}$
- 步骤：代价 $b_s(m-1)$

总计：$k_n \oplus b_z \oplus \sum_{i=0}^{m-1} b_s(i) \oplus m \cdot \delta_{\mathsf{natrec}} \preceq b_\circ$

**归纳情形**（vecrec）。与 natrec 类似，代价与向量长度呈线性关系。

**其他情形**。有序对、投影、恒等类型和 box 遵循 [1] 中的模式。 □

**推论 6.4** (依赖于大小的界限是可靠的)。如果 $\cdot \vdash_{r;\, c\cdot n+d} t : A$，其中 $n$ 是一个大小参数，那么 $t$ 的求值代价至多为 $c \cdot n + d$。

### 6.4 正则性

**定理 6.5**（正则性）。如果 $\cdot \vdash_{r;\, b} t : A$，其中 $A$ 是闭类型，那么 $t$ 求值为规范形式：

- 如果 $A = \mathbb{N}$，那么对于某个 $n \in \mathbb{N}$，$t \Downarrow \mathsf{S}^n(\mathsf{Z})$。
- 如果 $A = \mathsf{Vec}(B,\ n)$，那么 $t \Downarrow \mathsf{cons}(v_1,\ \ldots\ldots,\ \mathsf{cons}(v_n,\ \mathsf{nil})\ \ldots\ldots)$。
- 如果 $A = \mathsf{Fin}(n)$，那么对于某个 $k < n$，$t \Downarrow \mathsf{fsucc}^k(\mathsf{fzero})$。
- 如果 $A = \mathsf{Id}_B(x,\ y)$ 且 $x \equiv y$，那么 $t \Downarrow \mathsf{refl}_v$。
- 如果 $A = \Pi^{b\ (x)\ \ x:\ B} C$，那么 $t \Downarrow \lambda x.s$。
- 如果 $A = \Sigma_{x:\ B} C$，则 $t \Downarrow (v_1,\ v_2)$。

证明。对类型推导进行归纳，利用进展性（每个闭合且类型正确的项要么是值，要么可以执行步骤）以及消去子的结构。 □

## 7 语义模型

我们将预层模型从 [1] 扩展到依赖类型。主要挑战包括：

1. 依赖类型要求在元素范畴上使用预层
2. Martin-Löf 恒等类型具有非平凡的结构（J 消去子）
3. 内部格 $\mathbb{L}$ 必须与群胚结构兼容

在第一篇论文 [1] 中，我们是在取值于集合的函子 $\mathbf{Set}^L$ $L \to$**Set**构成的预层托扑斯中开展工作的。对于 RB–MLTT，我们将其提升为取值于群胚的预层范畴 $\mathbf{Gpd}^L$，以解释依赖类型和内涵恒等类型。将 $\mathbf{Set}^L$ 作为离散群胚的满子范畴嵌入 $\mathbf{Gpd}^L$ 是全忠实的，因此，[1] 中的 RB–TT 模型表现为本节所构建的群胚模型中 0-截断（取值于集合）的部分。

为了恰当地建模内涵 MLTT，我们使用群胚值预层，而不是集合值预层。这体现了恒等证明可能并非平凡，并且可以复合这一事实。

### 7.1 群胚值预层

**定义 7.1 (**范畴 $\mathbf{Gpd}$)。 $\mathbf{Gpd}$ 是由小群胚及其间的函子构成的范畴。群胚是一种每个态射都是同构的范畴。

**定义 7.2 (**群胚值预层，定义在 $L$)。 我们在函子范畴 $\mathbf{Gpd}^L$中工作，其中：

- 对象：函子 $F\colon L \to \mathbf{Gpd}$
- 态射：自然变换

对于每个 $r \in L$， $F(r)$ 都是一个群胚（不只是一个集合）。对于 $r_1 \preceq r_2$，我们有一个函子 $F(r_1 \preceq r_2) : F(r_1) \to F(r_2)$。

**备注 7.3 (**为什么是群胚？**)**。 在内涵 *MLTT* 中，恒等类型 $\mathsf{Id}_A(x, y)$ 可能有多个元素（恒等证明），而这些证明本身也具有恒等结构。群胚模型刻画了：

- $[\![A]\!](r)$的对象：类型为 $A$ 、代价为 $\preceq r$ 的元素

  $[\![A]\!](r)$中的态射：元素之间的恒等证明
- 复合：恒等的传递性（$p\colon\ x = y$ 和 $q\colon\ y = z$ 给出 $q \circ p\colon\ x = z$）
- 逆：对称性（$p\colon\ x = y$ 给出 $p^{-1}\colon\ y = x$）

这是类型论的标准群胚模型 *[5]*。

### 7.2 群胚设置中的内部格

**定义 7.4** （内部格 $\mathbb{L}$）。 定义 $\mathbb{L}\colon L \to \mathbf{Gpd}$ 如下：

$$\mathbb{L}(r) = \mathsf{disc}(\{a \in L \mid a \preceq r\})$$

其中 $\mathsf{disc}(S)$ 是集合 $S$ 上的离散群胚（仅含恒等态射）。

过渡函子是包含映射：$\mathbb{L}(r_1 \preceq r_2)\colon\ a \mapsto a$。

内部格是离散的，因为成本界没有非平凡的恒等态射：如果 $a$， $b \in L$作为成本相等，那么它们就是同一个成本。这是一个关键的简化。

**引理 7.5**。 格运算在$\mathbf{Gpd}^L$中诱导出自然变换：

- $\oplus\colon\ \mathbb{L} \times \mathbb{L} \Rightarrow \mathbb{L}$
- $\widetilde{\sqcup}\colon\ \mathbb{L} \times \mathbb{L} \Rightarrow \mathbb{L}_{\sim}$
- $\bot\colon\ \mathbf{1} \Rightarrow \mathbb{L}$

证明。 由于对所有 $r$， $\mathbb{L}(r)$都是离散的，因此离散群胚之间的函子就是对象上的函数。这些运算由 $\oplus$ 和 $\sqcup$的单调性保证定义良好。 □

## 7.3 上下文作为群胚值预层

**定义 7.6 (**上下文解释**)。** 通过归纳将上下文解释为预层 $[\![\Gamma]\!] : L \to \mathbf{Gpd}$:

$$[\![\cdot]\!](r) = \mathbf{1} \quad \textit{(terminal groupoid: one object, one morphism)}$$
$$[\![\Gamma, x : A]\!](r) = \int_{\gamma \in [\![\Gamma]\!](r)} [\![A]\!]_\gamma(r)$$

其中右侧是格罗滕迪克构造（总群胚）。

**定义 7.7 (**群胚的格罗滕迪克构造**)。** 对于 $G$ 一个群胚和 $F : G \to \mathbf{Gpd}$函子，总群胚 $\int_G F$ 具有:

- 对象：有序对 $(\gamma, a)$，其中 $\gamma \in \mathrm{ob}(G)$ 且 $a \in \mathrm{ob}(F(\gamma))$
- 态射 $(\gamma_1, a_1) \to (\gamma_2, a_2)$：有序对 $(p, q)$，其中 $p : \gamma_1 \to \gamma_2$ 属于 $G$，且 $q : F(p)(a_1) \to a_2$ 属于 $F(\gamma_2)$

## 7.4 依赖类型作为纤维化

**定义 7.8** （类型解释）。 对于 $\Gamma \vdash_{r;b} A$ 类型，定义 $[\![A]\!] : \int [\![\Gamma]\!] \to \mathbf{Gpd}$，其中 $\int [\![\Gamma]\!]$是 $L$上的格罗滕迪克构造:

$$[\![A]\!](r, \gamma) = \textit{groupoid of} \left\{ \begin{array}{l} \textit{Objects: } (v, b) \textit{ with } \cdot \vdash_{r;b} v : A[\gamma], b \preceq r \\ \textit{Morphisms } (v_1, b_1) \to (v_2, b_2) : \textit{ proofs } p : \mathsf{Id}_{A[\gamma]}(v_1, v_2) \end{array} \right\}$$

**注 7.9** （态射追踪恒等证明）。 $[\![A]\!](r, \gamma)$中的一个态射 $(v_1, b_1) \to (v_2, b_2)$ 是一个恒等证明 $p : v_1 =_A v_2$。证明 $p$ 的代价不超过 $\max(b_1, b_2) \oplus \delta_{\mathsf{Id}}$。态射的复合对应于传递性；逆对应于对称性。

## 7.5 代价自然变换

**定义 7.10** （代价作为自然变换）。 对于上下文 $\Gamma$ 中的每个类型 $A$ :

$$\mathsf{cost}_A : [\![A]\!] \Rightarrow \mathbb{L} \circ \pi$$

其中 $\pi : \int [\![\Gamma]\!] \to L$ 是投影。对对象而言：代价$_A(r, \gamma)(v, b) = b$。对态射而言：由于 $\mathbb{L}(r)$是离散的，所有恒等证明都映射为代价上的恒等态射。

**引理 7.11.** 代价$_A$ 是群胚值预层之间的自然变换。

证明。 我们必须证明每个阶段上的函子性。对于对象，这是显然的。对于态射：如果 $p : (v_1, b_1) \to (v_2, b_2)$ 是一个恒等证明，那么 $\mathsf{cost}_A(p)$ 必须是 $\mathbb{L}(r)$ 中的一个态射 $b_1 \to b_2$。由于 $\mathbb{L}(r)$ 是离散的，这就要求 $b_1 = b_2$。

这可以通过定义群胚 $[\![A]\!](r, \gamma)$ 来实现，其中只有成本界相同的有序对之间才存在态射；或者采用松弛版本，使代价可沿恒等证明递减。我们采用第一种方法：恒等证明保持认证界。□

## 7.6 恒等类型的解释

群胚结构直接刻画了恒等类型。

**定义 7.12** （恒等类型解释）。 对于 $\Gamma \vdash a, b : A$:

$$[\![\mathsf{Id}_A(a, b)]\!](r, \gamma) = \mathsf{Hom}_{[\![A]\!](r,\gamma)}([\![a]\!]_\gamma, [\![b]\!]_\gamma)$$

这是群胚 $[\![A]\!](r, \gamma)$ 中从 $a$ 的解释到 $b$ 的解释的一个离散群胚（集合），其元素为态射。

对象：带有成本界的恒等证明 $p$: $a[\gamma] =_A b[\gamma]$ 。

**引理 7.13 (**自反性**)**。自反性$_a$ 被解释为群胚$[\![A]\!](r,\gamma)$中的恒等态射 $\mathrm{id}_{[\![a]\!]_\gamma}$ 。

**定理 7.14** （J 消去子是可靠的）。*J* 消去子由群胚中恒等态射的泛性质建模：给定路径空间上的族 $C$ 以及 $d: C(a,$自反性$_a)$，对于任意 $p: a=b$，我们都可以通过沿 $p$传输得到 $\mathsf{J}(p,d): C(b,p)$。

证明。 在群胚 $G$中，对于任意函子 $F: G \to \mathbf{Gpd}$ 以及任意态射 $p: x \to y$ in $G$，都存在一个诱导函子 $F(p): F(x) \to F(y)$。J 消去子对应于：

$$\mathsf{J}(p,d) = F(p)(d)$$

其中 $F$解释动机 $C$。计算规则 $\mathsf{J}(\mathsf{refl}_a, d) = d$成立，因为 $F(\mathsf{id}_x) = \mathsf{id}_{F(x)}$。

□

## 7.7 Π 类型的解释

**定义 7.15** （Π-类型在群胚中的解释）。

$$\mathrm{ob}([\![\Pi^{b(x)}_{x:A} B]\!](r,\gamma)) = \left\{ (\lambda x.t, b_\lambda) \;\middle|\; \begin{array}{l} \cdot \vdash_{r;\, b_\lambda} \lambda x.t : (\Pi^{b(x)}_{x:A} B)[\gamma] \\ \forall (v, b_v) \in \mathrm{ob}([\![A]\!](r,\gamma)). \\ \quad \exists (w, b_w) \in \mathrm{ob}([\![B]\!](r,(\gamma,v))). \\ \quad t[\gamma][x := v] \Downarrow w \triangleright k, k \preceq b(v) \end{array} \right\}$$

态射 $(\lambda x.t_1, b_1) \to (\lambda x.t_2, b_2)$：证明 $\lambda x.t_1 =_\Pi \lambda x.t_2$，即逐点恒等证明 $\forall x.\, t_1[x] = t_2[x]$。

**注记 7.16** （函数外延性）。 $[\![\Pi^{b(x)}_{x:A} B]\!]$ 中的态射对应于函数外延性：两个函数相等，当且仅当它们逐点相等。在内涵 *MLTT* 中，函数外延性无法推导得出，但在群胚模型中成立。

## 7.8 Σ-类型的解释

**定义 7.17** （Σ-类型解释）。

$$\mathrm{ob}([\![\Sigma_{x:A} B]\!](r,\gamma)) = \left\{ ((v_1,v_2), b_1 \oplus b_2) \;\middle|\; \begin{array}{l} (v_1,b_1) \in \mathrm{ob}([\![A]\!](r,\gamma)) \\ (v_2,b_2) \in \mathrm{ob}([\![B]\!](r,(\gamma,v_1))) \end{array} \right\}$$

态射 $((v_1,v_2)\ b) \to ((w_1,w_2)\ b')$：有序对 $(p_1,p_2)$，其中 $p_1: v_1 = w_1$ 且 $p_2 : v_2 =_{B[p_1]} w_2$ （依赖路径）。

## 7.9 归纳类型的解释

**自然数**。

$$[\![\mathbb{N}]\!](r) = \mathsf{disc}(\{(\mathsf{S}^n(\mathsf{Z}), \bot) \mid n \in \mathbb{N}\})$$

自然数构成一个离散群胚：两个数值相等，当且仅当它们完全相同。

**向量**。

$$\mathrm{ob}([\![\mathsf{Vec}(A,n)]\!](r,\gamma)) = \left\{ (v,b) \;\middle|\; \begin{array}{l} v = \mathsf{cons}(v_1, \ldots, \mathsf{nil}) \\ \mathrm{length} = [\![n]\!]_\gamma \\ \mathrm{each}\ v_i \in \mathrm{ob}([\![A]\!]) \end{array} \right\}$$

态射：逐分量的恒等证明 $(p_1, \ldots, p_n)$，其中 $p_i: v_i = w_i$ 属于 $[\![A]\!]$。

## 7.10 宇宙的解释

**定义 7.18**（宇宙解释）。

$$\mathrm{ob}([\![\mathcal{U}_s]\!](r)) = \{(A, b) \mid A \ is\ a\ type\ code, \cdot \vdash_{r;b} A : \mathcal{U}_s, b \preceq s \preceq r\}$$

态射 $(A_1, b_1) \to (A_2, b_2)$：保留成本界的类型等价 $A_1 \simeq A_2$。

**备注 7.19 (**不具单值性**)**。 在群胚模型中，宇宙中的态射是类型同构（等价），但单值性公理（$\mathsf{ua} : (A \simeq B) \to (A = B)$）并未得到验证。我们使用内涵 *MLTT*，其中 $A = B$ 在 $\mathcal{U}$ 中要求定义相等，而不仅仅是等价。群胚结构捕捉了同伦 *1*-类型结构，但不包含更高阶单值性。

## 7.11 语义可靠性

**定理 7.20**（语义可靠性）。 如果 $\Gamma \vdash_{r;b} t : A$ 且 $\gamma \in \mathsf{ob}([\![\Gamma]\!](r))$，则：

*1.* 对某个 $v \in$ 值集合，$t[\gamma] \Downarrow v \triangleright k$
*2.* $k \preceq b[\gamma]$ *3.* $(v, b[\gamma]) \in \mathrm{ob}([\![A]\!](r, \gamma))$
*4.* 代价$_A(r, \gamma)(v, b[\gamma]) = b[\gamma]$

证明。 与定理 6.3 中一样，对类型推导进行归纳，但现在要追踪群胚结构。关键情形如下：

**情形（J 消去子）**。 给定 $p : \mathsf{Id}_A(a, b)$，将其解释为 $[\![a]\!] \to [\![b]\!]$ in $[\![A]\!](r, \gamma)$ 中的一个态射；又给定 $d : C[a, \text{自反性}_a]$，则 $\mathsf{J}(p, d) : C[b, p]$，它是沿 $p$ 传输得到的。代价为 $b_p \oplus b_d \oplus \delta_{\mathsf{J}}$。

**情形（Π-消去）**。 与前面一样，但现在函数之间的态射（恒等证明）由应用保留。

□

## 7.12 群胚上的资源有界上下文范畴

**定义 7.21**（群胚上的资源有界 CwF）。 一个 群胚上的资源有界族范畴 *(RB-CwF-Gpd)*格上的 $L$由以下部分组成：

*1.* 一个带有终对象的范畴，其中每个同态集 $\mathcal{C}(\Gamma, \Delta)$ 都具有群胚结构（从而使 $\mathcal{C}$ 成为一个 *(2,1)*-范畴） *2.* 一个类型的伪函子 $Ty : \mathcal{C}^{op} \to Gpd$ *3.* 对每个 $\Gamma \in \mathcal{C}$ 和 $A \in Ty(\Gamma)$，都有一个项的群胚 $Tm(\Gamma, A)$ *4.* 具有同构意义下泛性质的概括 *5.* 一个内部离散格对象 $\mathbb{L}_{\mathcal{C}}$*6.* 代价函子 $cost_A : Tm(\Gamma, A) \to disc(\mathbb{L}_{\mathcal{C}})$ *7.* 保持群胚结构和代价的类型构造子（$\Pi$, $\Sigma$, *Id*, $\mathbb{N}$, *Vec* 等）

**注记 7.22** (与普通 CwF 的关系)。 我们所说的 *RB-CwF-Gpd* 是对用于建模 *Martin–Löf* 类型论的标准带族范畴（*CwF*）的一种资源分级细化，例如 *Hofmann [4]* 所提出的范畴。当内部格对象取为终对象，且所有代价函子 代价$_A$ 都恒等于 $L$ 的最小元素时，*RB-CwF-Gpd* 就退化为群胚上的普通 *CwF*。从这个意义上说，*RB-CwF-Gpd* 并非一种全新的语义框架，而是在通常的 *CwF* 结构中增添成本数据。

**例 7.23** (句法 RB-CwF-Gpd)。 *RB-MLTT* 的语法构成一个 *RB-CwF-Gpd*，其中：

- $\mathcal{C}$（$\Gamma, \Delta$） $=$ 模定义相等的代换
- $\mathsf{Tm}$（$\Gamma, A$）上的群胚结构：将模定义相等的项视为相同，并以恒等证明作为态射
- $\mathbb{L}_{\mathcal{C}} = L$上的离散群胚

**例 7.24** (群胚预层 RB-CwF-Gpd)。 群胚预层模型 $\mathbf{Gpd}^L$ 构成一个 *RB-CwF-Gpd*，其中：

- $\mathcal{C} =$ 在 $L$上的群胚值预层
- $\mathsf{Ty}$（$[\![\Gamma]\!]$） $=$ 群胚值预层（在 $\int [\![\Gamma]\!]$ 上）
- $\mathsf{Tm}$ $=$ 纤维化的截面
- $\mathbb{L}_{\mathcal{C}} = \mathbb{L}$ （离散）

## 7.13 初始性

**定理 7.25** （初始性）。 句法 *RB-CwF-Gpd* 是初始的：对于任意 *RB-CwF-Gpd* $\mathcal{M}$，存在唯一的（在同构意义下）*RB-CwF-Gpd* 态射 $[\![-]\!]_{\mathcal{M}} : \mathbf{Syn} \to \mathcal{M}$。

证明。 对语法进行归纳，同时遵循群胚结构：

**上下文**。$[\![\Gamma]\!]_{\mathcal{M}}$ 由概括定义。

**类型**。$[\![A]\!]_{\mathcal{M}}$ 使用 $\mathcal{M}$ 中的类型构造子定义。

**项**。$[\![t]\!]_{\mathcal{M}}$ 使用项构造子定义。对于恒等证明，使用群胚结构。

**恒等证明**。 对于 $p$ ： 上下文 $\Gamma$ 中的 $a = b$ ，是群胚 $\mathsf{Tm}([\![\Gamma]\!]_{\mathcal{M}}, [\![A]\!]_{\mathcal{M}})$ 中的态射 $[\![a]\!]_{\mathcal{M}} \to [\![b]\!]_{\mathcal{M}}$ $[\![p]\!]_{\mathcal{M}}$ 。

**代价保持**。 代价函子与解释可交换：

$$\mathsf{cost}_{[\![A]\!]_{\mathcal{M}}}([\![t]\!]_{\mathcal{M}}) = [\![b]\!]_{\mathcal{M}}$$

其中 $\Gamma \vdash_{r;b} t :\ A$。

**唯一性**。 在自然同构意义下，由类型构造子的泛性质和群胚结构所强制确定。

□

**推论 7.26 (**可靠嵌入**)**。 对于任意闭项 $\cdot \vdash_{r;b} t : A$ 以及任意 *RB-CwF-Gpd*$\mathcal{M}$:

*1.* $[\![t]\!]_{\mathcal{M}} \in \mathrm{ob}(\mathsf{Tm}(\mathbf{1}, [\![A]\!]_{\mathcal{M}}))$*2.* 代价$([\![t]\!]_{\mathcal{M}}) \preceq b$*3.*
恒等证明得以保留： $[\![\text{自反性}_a]\!]_{\mathcal{M}} = \text{恒等类型}_{[\![a]\!]_{\mathcal{M}}}$

**备注 7.27 (**与高阶模型的关系**)**。 群胚模型捕捉了类型的 *1*-截断结构（带有非平凡恒等类型的集合）。要构建具有单值性和高阶归纳类型的完整同伦类型论，需要 $\infty$-群胚值预层（单纯集合或类似结构）。这是 *RB-HoTT* 的研究主题（未来工作）。

# 8 案例研究

## 8.1 VectorSum： $O(n)$

$$\begin{aligned}&\mathsf{sum} : \Pi^{\perp}_{n:\mathbb{N}} \mathsf{Vec}(\mathbb{N}, n) \to^{3n+2} \mathbb{N}\\ &\mathsf{sum} = \lambda n.\, \lambda v.\, \mathsf{vecrec}(v, \mathsf{Z}, \lambda m.\lambda a.\lambda w.\lambda ih.\, a + ih)\end{aligned}$$

**界推导**。

- 基例：返回 Z 成本 $\delta_{\mathsf{Z}} = 1$
- 步骤：加法的成本为 $\delta_{+} = 2$，cons 遍历的成本为 1
- 总计： $1 + n \cdot 3 = 3n + 1$，再加上开销 $\delta_{\mathsf{vecrec}} = 1$
- 最终界： $3n + 2$根据定理 6.3，$\mathsf{sum}\, n\, v$的求值代价为 $\leq 3n + 2$。

## 8.2 VectorReverse： $O(n)$

$$\begin{aligned}&\mathsf{reverse} : \Pi^{\perp}_{n:\mathbb{N}} \Pi^{\perp}_{A:\mathcal{U}} \mathsf{Vec}(A, n) \to^{4n+2} \mathsf{Vec}(A, n)\\ &\mathsf{reverse} = \lambda n.\lambda A.\lambda v.\, \mathsf{rev\text{-}acc}(v, \mathsf{nil})\end{aligned}$$

其中，rev-acc 是带有累加器的尾递归辅助函数。

## 8.3 二分查找： $O\left(\log n\right)$

对于有序向量，二分查找

**数据结构假设**。 此界限假设如 §3.10 所示支持 $O(1)$ 随机访问（直接索引）。如果向量仅以归纳方式表示，且查找从头部开始遍历，那么每次探查的代价为 $O(n)$，整体界限则变为 $O(n\log\ n)$。其代价为对数级：

$$\mathsf{bsearch} : \Pi^{\perp}_{n:\mathbb{N}} \mathsf{Vec}(\mathbb{N}, n) \to \mathbb{N} \to^{5\cdot\lceil \log_2(n+1)\rceil + 3} \mathsf{Option}(\mathsf{Fin}(n))$$

**界推导**。

- 每次迭代：比较（2）、索引计算（2）、切片（1） $= 5$ 总计
- 迭代次数： $\lceil \log_2(n+1) \rceil$
- 基例和封装：3
- 总计： $5 \cdot \lceil \log_2(n+1) \rceil + 3$

### 8.4 归并排序： $O(n \log\ n)$

$$\text{归并排序}: \Pi^{\perp}_{n:\mathbb{N}} \Pi^{\perp}_{A:\mathcal{U}} (A \to A \to^{c_{\text{cmp}}} \text{布尔值}) \to \text{Vec}(A,n) \to^{c \cdot n \cdot \lceil \log_2(n+1) \rceil + d} \text{Vec}(A,n)$$

**界推导**。

- 分割：将向量分成两半，代价 $O(n)$
- 求解：对大小为 $n/2$ 的部分进行两次递归调用
- 合并：合并两个已排序的两半，代价 $O(n)$
- 递推关系： $T(n) = 2T(n/2) + O(n)$
- 解： $T(n) = O(n \log\ n)$

### 8.5 矩阵乘法： $O(n^3)$

对于表示为 $\text{Vec}(\text{Vec}(R,n)\ n)$ 的 $n \times n$ 矩阵：

$$\text{matmul} : \Pi^{\perp}_{n:\mathbb{N}} \text{Mat}(R,n) \to \text{Mat}(R,n) \to^{c \cdot n^3 + d} \text{Mat}(R,n)$$

其中 $\text{Mat}(R,n) = \text{Vec}(\text{Vec}(R,n)\ n)$。

## 9 实现说明

### 9.1 Lean 4 集成

该框架可使用 Lean 4 实现，方法如下：

```
-- 按大小索引的向量归纳定义的向量 Vec(A :
Type) : Nat -> Type where| 空列表 : Vec A 0|
cons : A -> Vec A n -> Vec A (n + 1)

    -- 界注解（自定义属性）@[bound
(funn => 3 * n + 2)]def sum : (n : Nat) -
> Vec Nat n -> Nat| 0, .空列表 => 0| n + 1,
.cons x xs => x + sum n xs

    -- 界定理定理 sum_bound (n : Nat) (v : Vec
Nat n) : cost (sumn v) <= 3 * n + 2 := 通过对
n 归纳，其中| zero =>简化 [sum, cost]| succ n
ih => 简化 [sum, cost]; omega
```

### 9.2 Agda 集成

在 Agda 中使用大小类型:

```
数据 Vec (A: 集合) : 自然数 -> 集合，其中[] : Vec A 零_::_ :
对所有{n} -> A -> Vec A n -> Vec A (后继 n)

sum : forall{n} -> Vec Nat n -> Nat sum
  [] = 0sum (x :: xs) = x + sum xs

-- 在类型中表达的界限 sum-bounded :forall n -> (v : Vec Nat n) -> Cost (sum v) <= 3
* n + 2
```

本文实现所基于的底层（不受资源界限约束的）外延式 MLTT 语法及类型判定基础设施的 Lean 4 机械化形式化，可在 Corey Thuro 的 RB-TT 仓库中找到。具体而言，文件 src/RBTT/Core/ExtrinsicMLTT.lean 提供了外延式判断呈现（语法、上下文、替换和类型推导谓词），我们计划按照此处所述为其扩展资源注释和界限合成:

https://github.com/CoreyThuro/RB-TT/blob/main/src/RBTT/Core/ExtrinsicMLTT.lean.

## 10 相关工作

**依赖类型与代价。** Danielsson [9] 在 Agda 中使用依赖类型开展轻量级的半形式化复杂度分析。我们的工作将资源界作为类型系统中的一等公民，并提供形式化的可靠性保证。

**大小类型。** Abel [7] 和 Hughes 等人 [8]使用大小注释来保证终止性。我们将这一思想扩展到资源界，而不仅限于终止性。

**类型论基础。** Martin-Löf [2]奠定了基础；Nordström 等人 [3]提供了入门介绍。我们为 MLTT 增添了资源跟踪。

**范畴语义。** Hofmann [4] 使用上下文范畴（CwFs）发展了依赖类型的语法与语义。Jacobs [6] 提供了全面的论述。我们的 RB-CwF 在此基础上扩展了成本结构。

**余效应与分级类型。**Petricek 等人 [10], Brunel 等人 [11], 和 Orchard 等人 [12]发展了分级类型系统。我们将分级扩展到依赖类型，并为其提供大小索引界限。

**定量类型论。**Atkey [13] 和 McBride [14] 追踪使用量。我们追踪的是计算成本，而非使用量。

**RAML 与摊还分析。**Hoffmann 等人 [15] 自动推断多项式界。我们的方法通过类型推导而非推断来证明界限。

**先前工作。** 本文将 [1] 从简单类型扩展到依赖类型。

# 11 局限性与未来工作

**当前局限性**。

- 界限必须能用我们的界语言表示（多项式、对数）
- 没有自动界推断；界会被检查，但不会被推导
- 仅支持内涵恒等类型；不支持单值性
- 不支持高阶归纳类型

**未来方向**。

- **RB-HoTT：** 扩展到带资源有界路径的同伦类型论
- **自动推断：** 与 RAML 风格推断集成
- **立方类型论：** 资源有界立方结构
- **概率界：** 期望成本分析
- **机械化形式化：** 在 Agda 或 Lean 中进行完整形式化
- **商类型：** 资源有界商

# 12 结论

我们已将资源有界类型理论扩展至 Martin-Löf类型理论，从而能够为依赖类型给出依赖于规模的成本界。主要创新如下：

1. **依赖界：** $\Pi_x^{b(x)} : A^B$ 其中 $b(x)$ 取决于参数 2. **归纳族：** Vec($A$, $n$) 和 Fin($n$), 具有依赖于规模的消去子界 3. **资源宇宙：** $\mathcal{U}_r$ 跟踪类型形成代价

4. **代价可靠性：** 定理 6.3 证明这些界是对代价的上界
5. **语义模型：** 具有依赖结构和初始性的预层

这弥合了依赖类型论与定量资源分析之间的鸿沟，使得可以直接在类型中表达依赖于规模的算法，并为其提供经认证的成本界。该框架涵盖常见的复杂度类（$O(1)$、$O(n)$、$O(\log\ n)$、$O(n\log\ n)$、$O(n^2)$、$O(n^3)$），并验证了向量运算、二分查找和排序等基础算法的界。

未来工作将把该框架扩展至同伦类型论（RB-HoTT），探索资源有界路径和高阶结构。

# 参考文献

[1] M. A. Mannucci 和 C. Thuro。《资源有界类型理论：通过分级模态进行组合式成本分析》。*arXiv* 预印本，2025。

[2] P. Martin-Löf。《直觉主义类型论》。Bibliopolis，那不勒斯，1984年。

[3] B. Nordström、K. Petersson 和 J. M. Smith。《Martin-Löf 类型论编程》。牛津大学出版社，1990年。[4] M. Hofmann。依赖类型的语法与语义。载于《计算的语义与逻辑》，第79–130页。剑桥大学出版社，1997年。[5] M. Hofmann 和 T. Streicher。类型论的群胚解释。载于《构造型类型论二十五年》，第83–111页。牛津大学出版社，1998年。[6] B. Jacobs。《范畴逻辑与类型论》。爱思唯尔，1999年。[7] A. Abel。MiniAgda：整合大小类型与依赖类型。载于《PAR会议录》，2010年。[8] J. Hughes、L. Pareto 和 A. Sabry。使用大小类型证明反应式系统的正确性。载于《POPL会议录》，第410–423页。美国计算机协会，1996年。[9] N. A. Danielsson。纯函数式数据结构的轻量级半形式化时间复杂度分析。载于《POPL会议录》，第133–144页。美国计算机协会，2008年。[10] T. Petricek、D. Orchard 和 A. Mycroft。余效应：上下文依赖计算的演算。载于《ICFP会议录》，第123–135页。美国计算机协会，2014年。[11] A. Brunel、M. Gaboardi、D. Mazza 和 S. Zdancewic。核心定量余效应演算。载于《ESOP会议录》，《计算机科学讲义》第8410卷，第351–370页。施普林格，2014年。[12] D. Orchard、V.-B. Liepelt 和 H. Eades III。利用分级模态类型进行定量程序推理。《美国计算机协会程序设计语言会议录》，3(ICFP):110:1–110:30，2019年。[13] R. Atkey。定量类型论的语法与语义。载于《LICS会议录》，第56–65页。美国计算机协会，2018年。[14] C. McBride。我有的是啥也没有。载于《改变世界的成功集锦》，《计算机科学讲义》第9600卷，第207–233页。施普林格，2016年。[15] J. Hoffmann、K. Aehlig 和 M. Hofmann。多变量摊销资源分析。《美国计算机协会程序设计语言与系统汇刊》，39(1):3:1–3:58，2017年。[16] S. Mac Lane 和 I. Moerdijk。《几何与逻辑中的层》。施普林格出版社，1992年。[17] S. Awodey。同伦类型论的自然模型。《计算机科学中的数学结构》，28(2):241–286，2018年。[18] P. Dybjer。内在类型论。载于《证明与程序的类型》，《计算机科学讲义》第1158卷，第120–134页。施普林格，1996年。