一个温和的提议：明确支持基础多元主义

Martin Berger ! Montanarius Ltd，伦敦，英国 Turing Core，Huawei 2012 Labs，Huawei R&D Ltd，伦敦，英国 萨塞克斯大学，布莱顿，英国

Dominic P. Mulligan ! 自动推理组，亚马逊云科技，剑桥，英国[1]

arXiv:2302.10137v1 [cs.LO] 20 Feb 2023

摘要

数学家通常默认采用经典推理原则，但在工作时却经常进行上下文切换，将自己限制在各种形式的次经典推理之中。这种模式在逻辑学家和集合论学家中尤其常见，但日常从事数学工作的数学家也经常如此，证明附带的叙述性注释便是明证——例如“以下证明是构造性的”，或“以下证明不使用选择公理”。然而，当前的证明助手在正式捕捉这些叙述性注释方面支持不足；对于基于 Gordon 的高阶逻辑这一经典高阶逻辑的系统而言，这一点尤其明显。因此，高阶逻辑及其众多实现似乎反而比数学家主流自身更加坚定地遵循经典推理，从而限制了人们能够轻易形式化的数学内容。

为便于进行这些上下文切换，我们建议数学家在暂时采用次经典方式工作时，在脑中使用一种简单的污点系统——这一想法目前尚未在证明助手中得到探索。因此，我们对 HOL 引入了一系列看似 modest 但影响深远的改动，将标准的二元自然演绎关系扩展为纳入一个污点标签；该标签取自某个特定的格，用于描述或限制证明中所使用的经典推理“量”。污点既可以被视为 HOL 证明上的一种简单类型系统，也可以被视为对证明树进行的一种静态分析；它将我们的逻辑划分为表达力各异、并列存在的多个片段。结果可以无需修改地从逻辑中“较不经典”的片段进入“更经典”的片段，但不能反向进行；不同世界之间的结果流动由一条类似于子类型规则或包含规则的推理规则控制。所有世界中的成员都基于同一组定义进行推理；因此，为了最大限度地复用结果，用户应将结果表述在能够满足需求的最弱世界中，避免不必要地采用经典推理。

2012 ACM 学科分类 计算理论 → 证明论；计算理论 → 自动推理

关键词和短语 高阶逻辑、污点、构造主义、基础多元主义

数字对象标识符 10.4230/LIPIcs...

致谢 作者谨向 Jim Grundy、John Harrison、Konrad Slind 和 Christian Urban 致谢，感谢他们对本文早期草稿提出的建设性意见。

1 引言

数学家默认采用经典推理原则，但会在不同的基础（通常是次经典）推理风格之间自由地“切换上下文”。

逻辑学家、集合论学家以及其他从事基础性问题研究的人，是这一模式最明显的代表；例如，他们可能会避免使用选择公理，转而采用更受限的选择原则——如依赖公理和

[1] 所有工作均在受雇于 Arm Research 期间完成



©Martin Berger 和 Dominic P. Mulligan；依据知识共享署名 4.0 国际许可协议授权莱布尼茨国际信息学论文集 达格施图尔城堡——莱布尼茨信息学中心，Dagstuhl 出版社，德国

XX:2 一个温和的提议：明确支持基础多元主义

可数选择公理——以研究其推论 [24]。在构造数学内部，对于“构造主义”的含义也存在几种相互竞争的观点，因此人们还在争论哪些推理原则是可接受的——例如，植根于递归论的俄罗斯构造主义学派，就与直觉主义截然不同 [9]。这种多元主义反映了这样一个事实：数学并不存在唯一且普遍认可的基础体系，而是存在许多不同的、具有合理性的基础体系，每一个都具有独立的研究价值。

这种上下文切换在更主流的非基础数学中也很常见。证明往往会在自然语言叙述中附带相关的补充说明，例如：“以下证明是构造性的”，或“以下证明没有使用选择公理”。[2] 这些评论邀请读者暂时切换上下文，离开默认的经典推理模式，在证明过程中转而采用某种受限的次经典模式。作为读者，我们实际上是受作者邀请，暂时改变所假定的基础体系。

这些补充说明本身也颇有意思，因为它们的数学地位并不明确。如果一个证明被声称是构造性的，但实际上并非如此，那么这个证明就是错误的吗？同样，如果一个证明被声称不依赖选择公理，却援引了佐恩引理或其他已知与选择公理等价的结果，那么这个证明就是错误的吗？在这两种情况下，我们都认为证明显然确实是错误的。因此，这些说明具有某种数学内容，会改变定理陈述：不仅仅是ϕ，而是 ϕ 构造性地，或 ϕ 不使用选择公理。在这些会改变陈述的说明所处的上下文中，读者似乎会通过简单的心智分析来检查证明：证明中出现的经典推理原则，或间接出现在引理中的经典推理原则，都会不可磨灭地污染结果。在主流数学的默认基础体系中采用经典推理时，这种污点机制是默认的，只有在采用次经典推理时才变得必要且显而易见。

遗憾的是，现有证明助手在支持这种基础层面的上下文切换方面表现不佳。这一观察同样适用于基于类型论和经典高阶逻辑的证明助手——它们代表了多种已实现的系统。

以 Agda [31]、Coq [7]，或 Matita [1]，为例，基于类型论的系统从直觉主义基础逻辑出发，并采用各种技术来获得更经典的基础。最直接的技术是直接添加经典公理。然而，由于类型论中定义等式与命题等式的区别，经典推理器处于不利地位：缺乏计算内容的公理可能会“搅乱”转换。嵌入经典逻辑的其他方案包括哥德尔－根岑式双重否定翻译 [3, 16, 19] 和稳定嵌入 [32]——这些方法会使目标中充斥多余的否定或附加条件，降低其在扩展工作中的易用性；或者，也可以在专用单子[3]，中进行推理，此外还有许多其他方法。尽管经典数学在技术上可以 纳入直觉主义类型论，但在实践中，其拥护者始终只能成为二等公民。

实现 Gordon 的高阶逻辑这一经典高阶逻辑的系统——例如 HOL4 [35]，Isabelle/HOL [36]，或 HOL Light [21]，——也存在问题。尤其是，HOL 系统无法轻松地将推理限制在逻辑的某个次经典片段中。通过希尔伯特式不定描述算子体现的选择公理的使用，如theϵ-演算 [4]，一样，在这种逻辑中无处不在。因此，根据 Diaconescu 定理 [14, 20]

[2] 例如，参见此处随定理 7.2 附带的注释 [12]，或此处引理 4.3 前面的注释 [25]。[3] 例如，Agda 标准库中就是如此：https://agda.github.io/agda-stdlib/v1.1/Relation.Nullary.Negation.html#1727。

HOL 布尔类型——既充当命题的类型，又充当二元素数据类型——满足排中律。[4] 讽刺的是，HOL 及其众多实现，比数学家自身更执着于经典推理。

无法在 HOL 内限制主流推理方式，会产生实际后果。光滑无穷小分析 （SIA）是一种无穷小分析方法，它要求采用构造性推理，以避免理论不可避免地崩溃 [6]。固定一个名称颇具启发性的载体集合 $\mathbb{R}$、常量 $\{0,1\} \subseteq \mathbb{R}$，并假设其上封闭地定义了具有通常性质的域运算 $\mathbb{R}$，再定义集合 Δ，其中包含平方为零的幂零元，其中$\Delta = \{\epsilon \in \mathbb{R} \mid \epsilon \cdot \epsilon = 0\}$。假设排中律， Δ 可以被证明会崩溃为单元素集合 $\{0\}$。然而，如果我们避开排中律，也就避开了这种崩溃；我们最多只能说 $0 \in \Delta$，并且对于任意 $\epsilon \in \Delta$，都有 $\epsilon \cdot \epsilon = 0$，从而留下了 Δ 中存在非零元素的可能性。将 Δ 视为无穷小量的集合 —— 即平方后为零、因而小到如此程度的元素——我们还可以进一步假设公理，断言所有函数 $f \colon \mathbb{R} \to \mathbb{R}$ 都具有连续性。这样，我们便可以利用无穷小量，以综合方式发展分析学。无论如何，只要我们仍维持构造性的假设。

然而，由于 HOL 坚持经典数学，在实现 HOL 的系统中，要以符合人体工学的方式形式化 SIA 并不容易。以作者熟悉的 Isabelle/HOL 为例，要处理 SIA，我们不得不采用一些技巧，例如在直觉主义逻辑的内部深嵌入中进行工作。显然，在定理证明器中的定理证明器 里工作并不可取，也偏离了 SIA 在实践中的引入方式。例如，Bell 在 SIA 的入门概述中 [6], 仅仅指出，本书将始终不采用排中律及其他经典推理原则——这是适用于整本书的叙述性说明，而非仅适用于单个证明。

总而言之，我们的论点是，当前的证明助手在基础立场上过于 武断。不同推理风格之间，甚至不同主要数学思想流派之间的转换，都很不顺畅；要进行转换，就必须通过采用古怪或不切实际的编码与嵌入作出妥协。这些教条式的立场与数学实践格格不入，也在一定程度上促成了形式化数学社区分裂为彼此隔绝的子社区；每个子社区都依附于自己效忠的基础体系：构造主义者聚集在一类工具周围，经典数学家则聚集在另一类工具周围。我们的贡献如下：

1. 我们观察到一种“上下文切换”现象，而证明检查软件无法很好地对其建模，因而难以自然地形式化某些数学内容。**2.** 在 §2 中，我们对 Gordon 的高阶逻辑做了一些幅度不大、影响深远的修改，以便更准确地对这些上下文切换建模。我们扩展了自然演绎关系，使其还包含一个污点标签，用以记录证明中使用经典推理的程度。起初，该逻辑以一个通用的污点标签格（有下界）作为参数；随后，我们将其专门化为一个与某些我们感兴趣的数学系统相对应的格。**3.** 在 §3 中，我们为逻辑加入数据类型，并讨论定义原则。我们探讨如何改进现有的高阶逻辑定义原则，并推测性地描述另一种潜在原则：它将我们从固定的污点标签集合转向一种能够动态添加这些标签的逻辑，从而引入一个污点的“开放世界”。**4.** 在 §4 中，我们讨论对高阶逻辑所做的修改可能会对系统中编码的形式化数学产生哪些实际影响。**5.** 在 §5 中，我们讨论以 Scala 编写的逻辑原型实现；该实现采用 LCF 风格证明检查系统。

[4] 至少在 HOL Light 中，排中律实际上就是这样推导出来的。

2 更具多元主义色彩的高阶逻辑

我们提出一种逻辑，使持有不同基础立场的用户能够并肩工作，理想情况下尽可能处于平等地位。我们以高阶逻辑作为基础逻辑，并对其做了一些虽小却重要的改动。高阶逻辑的一大优势是，它只包含一种相等概念，且没有内在的计算概念，因此不同公理不会彼此干扰。原则上，证明污染的思想可以应用于任何其他逻辑系统。

在 §2.2 中，我们将介绍对标准二元自然演绎关系的扩展，引入一个三元关系 $\Gamma \vdash \phi : \ell$，其联系上下文、公式和污点标签，并进一步发展这一关系的元理论。不过，在此之前，我们会在 §2.1中介绍基本的标准内容。我们省略证明，因为它们都很直接，而且任何熟悉证明论或类型论的人都不会感到意外。行家可以直接跳到后文。

2.1 语言：种类、类型和项

我们首先通过以下递归文法定义**种类**：

$$\kappa, \kappa', \kappa'' ::= \star \mid \star \Rightarrow \kappa$$

种类箭头是右结合的，我们用 $\star \Rightarrow \star \Rightarrow \star$ 作为 $\star \Rightarrow (\star \Rightarrow \star)$ 的简写。种类的形式受到限制，**类型的种类** $\star$ 总是出现在种类箭头 $-\Rightarrow-$ 的左侧，这使得种类与自然数同构。

我们还固定一个可数无限的**类型变量**集合，并使用 α、 β、 γ等任意表示类型变量。对于类型变量，我们采用**置换约定**：假定名称不同的类型变量彼此不同，因此 $\alpha \neq \beta$ 始终如此；这一技巧将在后文简化某些定理陈述。

对于每一种类 κ，我们都关联一个可数无限的**类型构造子**集合，并使用 $\mathtt{F}{:}\kappa$、$\mathtt{G}{:}\kappa$、 $\mathtt{H}{:}\kappa$ 等任意表示与种类 κ相关联的类型构造子。之后，我们有时可能省略类型构造子的种类标注，倾向于写作 $\mathtt{F}$而非 $\mathtt{F}{:}\kappa$，不过省略时，其种类可以从上下文推断出来。如果 $\kappa \neq \kappa'$，那么 $\mathtt{F}{:}\kappa$ 与 $\mathtt{F}{:}\kappa'$之间并无关联，不过我们会避免这种名称冲突。

今后，对于种类 $\star$，我们假定其关联的无限类型构造子集合至少包含指定的类型构造子 $\mathtt{Prop}$；同样，我们假定与种类 $\star \Rightarrow \star \Rightarrow \star$ 关联的无限类型构造子集合至少包含指定的类型构造子 $-\!\!\rightarrow\!-$。之后还会假定存在更多类型构造子。这里，我们指出我们的逻辑与 HOL 之间的一项早期区别：我们将 HOL 的布尔类型拆分为两种类型，各自专用于不同目的。我们用 $\mathtt{Prop}$ 作为命题类型。之后，我们将引入布尔类型 $\mathtt{Bool}$，将其定义为一种二元素数据类型，并像对待其他类型一样，为其赋予通常的递归和归纳原理。HOL 将命题类型视为数据，因此又引入了一条后门，使排中律得以进入，因为该原理可以直接从布尔类型的结构归纳规则推导出来。

接下来，我们通过以下递归文法定义**预类型**：

$$\tau, \tau', \tau'' ::= \alpha \mid \mathtt{F}{:}\kappa \mid \tau\tau'$$

我们用 $ftv(\tau)$ 表示出现在（预）类型 τ中的**自由类型变量** 集合，并用$\tau[\beta{:=}\tau']$表示将预类型 τ'替换 τ中 β的**类型替换**，满足：

▶ **引理 1.** $\tau[\alpha := \alpha] = \tau$

▶**引理 2**。如果 $\beta \notin ftv(\tau)$ 那么 $\tau[\beta := \tau'] = \tau$。

▶**引理 3**。如果 $\beta \notin ftv(\tau'')$ 那么 $\tau[\beta := \tau'][\gamma := \tau''] = \tau[\gamma := \tau''][\beta := \tau'[\gamma := \tau'']]$。

▶**引理 4.** $ftv(\tau[\beta := \tau']) \subseteq (ftv(\tau) - \{\beta\}) \cup ftv(\tau')$

我们使用以下规则在预类型上定义一种**种类关系**，并用$\vdash \tau : \kappa$表示存在一棵按照下列规则构造、且以$\vdash \tau : \kappa$为根的推导树：

$$\frac{}{\vdash \alpha : \star}\,(\mathbf{Kv}) \qquad \frac{}{\vdash \mathtt{F}{:}\kappa : \kappa}\,(\mathbf{Kf}) \qquad \frac{\vdash \tau : \star \Rightarrow \kappa \quad \vdash \tau' : \star}{\vdash \tau\tau' : \kappa}\,(\mathbf{Ka})$$

如果$\vdash \tau : \star$，那么我们称 τ 为一种**类型**；在本文余下部分，我们通常只讨论类型。种类关系具有一些“显然的”正确性性质：

▶**引理 5. </style}如果 ‘ :** κ 且 ‘ $\tau' : \star$，那么 ‘ $\tau[\beta := \tau'] : \kappa$。

▶ **引理 6.** 如果 ‘ $\tau : \kappa$ 且 ‘ $\tau : \kappa'$，那么 $\kappa = \kappa'$。

我们在语法上稍作变通，用 `Prop` 表示**命题的类型**，也表示其底层的类型构造器。此外，若 τ 和 τ' 是类型，那么 $\tau \to \tau'$ 也同样是类型；我们称之为**函数类型**。函数箭头按右结合，因此我们可以用 $\tau \to \tau' \to \tau''$ 简写 $\tau \to (\tau' \to \tau'')$。

现在，对于每个前类型 τ，我们都关联一个可数无限的**项变量**集合——若不易与类型变量混淆，也简称**变量** ——以及一个可数无限的**常量**集合。我们用 $x{:}\tau$、 $y{:}\tau$、 $z{:}\tau$ 等，任意表示与前类型 τ 关联的变量；用 $\mathtt{C}{:}\tau$、 $\mathtt{D}{:}\tau$、 $\mathtt{E}{:}\tau$ 等，任意表示与之关联的常量。为方便起见，我们也可以省略类型注释，不过在这种情况下，前类型 τ 始终可以推断出来。此外，若 $\tau \neq \tau'$，那么变量 $x{:}\tau$ 与 $x{:}\tau'$ 之间并无特殊关联，常量 $C{:}\tau$ 与 $C{:}\tau'$ 之间也一样。对于变量，我们再次采用**置换约定**，并假设存在以下常量，它们分别对应逻辑联结词和量词，且具有相应的类型：

$\top, \bot$		`Prop`
$\wedge, \vee, \longrightarrow, \longleftrightarrow$		$\mathtt{Prop} \to \mathtt{Prop} \to \mathtt{Prop}$
$\neg$	with type	$\mathtt{Prop} \to \mathtt{Prop}$
$=$		$\alpha \to \alpha \to \mathtt{Prop}$
$\exists, \forall$		$(\alpha \to \mathtt{Prop}) \to \mathtt{Prop}$

眼尖的读者会注意到，我们的逻辑与高阶逻辑之间还有另一处差异：缺少希尔伯特 ϵ-算子的任何类似机制。借助这一机制，人们可以选择 一个满足某个谓词的元素，其中项 $\epsilon x.P\ x$ 要么表示一个满足 P的元素，要么在不存在这样的元素时保持未定义。为了赋予这一构造以意义，高阶逻辑实现通常会使用一条额外的自然演绎规则：

$$\frac{\Gamma \vdash \exists x.\ P\ x}{\Gamma \vdash P\ (\epsilon x.P\ x)}$$

直观地说：如果我们能先证明这样的元素存在，那么性质 P 就适用于 ϵ-算子选出的元素。借助这一点，我们可以定义 函数的逆并对其进行推理，还可以证明 与选择公理等价的高阶逻辑陈述。

就我们的目的而言，这存在问题。从某种意义上说，希尔伯特的 ϵ算子表达力过于强大，因为正如刚才所指出的，可以用它推导出选择公理，这会使我们无法将推理限制在各种较弱的选择原则中的任何一种。此外，如果我们将 ϵ算子作为常量纳入其中，那么当我们受到

次经典推理限制时，就必须决定如何处理使用这一构造的定义。一种做法是将 ϵ算子视为未解释的符号，并在进行次经典推理时限制使用上文给出的规则。从审美角度看，这种做法相当不美观。因此，我们不采用描述算子，而是将选择公理作为一条明确的公理引入。这一决定对如何运用我们的逻辑有显著影响；我们将在本文第§4节稍后再讨论这一问题。

我们通过以下语法递归定义显式类型的 λ**-项**：

$$r, s, t ::= x{:}\tau \mid \mathsf{C}{:}\tau \mid rs \mid \lambda x{:}\tau.\ r$$

与通常情况一样，变量 $x{:}\tau$ 在项 $\lambda x{:}\tau.\ r$中称为**约束** 变量。此后，我们将项视为在 α-等价意义下相同，并确保从此处开始的所有定义均关于 α-等价良定义。为了体谅读者，我们采用常见的数学惯例，例如写作 $\forall x{:}\tau.\ \phi$而非 $\forall[\alpha := \tau](\lambda x{:}\tau.\ \phi)$，写作$\phi \longrightarrow \psi$ 而非 $(\longrightarrow \phi)\psi$。其他逻辑联结词和量词也同理。

我们用 $fv(r)$ 表示出现在 r 中的**自由变量** 集合，因此 $fv(\lambda x{:}\tau.\ y{:}\tau') = \{y{:}\tau'\}$；用 $ftv(r)$ 表示出现在 r 中的**类型变量** 。我们用 $r[\alpha := \tau]$表示将类型替换扩展为作用于项的**类型替换作用**，并用$r[x{:}\tau := s]$ 表示**避免捕获的替换作用**，它将 $x{:}\tau$ 的所有出现替换为 s，同时酌情重命名约束变量。类型替换作用满足一些显而易见的性质：

▶ **引理 7.** $r[\beta := \beta] = r$▶ **引理 8. 若** $\beta \notin ftv(r)$**，则** $r[\beta := \tau'] = r$.▶ **引理 9. 若** $\beta \notin ftv(\tau'')$**，则** $r[\beta := \tau'][\gamma := \tau''] = r[\gamma := \tau''][\beta := \tau'[\gamma := \tau'']]$.▶ **引理 10.** $ftv(r[\beta := \tau']) \subseteq (ftv(r) - \{\beta\}) \cup ftv(\tau')$

同样，避免捕获的替换作用：

▶ **引理 11.** $r[x{:}\tau := x{:}\tau] = r$▶ **引理 12. 若** $x{:}\tau \notin fv(r)$**，则** $r[x{:}\tau := s] = r$.▶ **引理 13.** $fv(r[x{:}\tau := s]) \subseteq (fv(r) - \{x{:}\tau\}) \cup fv(s)$▶ **引理 14. 若** $x{:}\tau \notin fv(t)$**，则** $r[x{:}\tau := s][y{:}\tau' := t] = r[y{:}\tau' := t][x{:}\tau := s[y{:}\tau' := t]]$.

我们根据以下规则定义项上的**类型关系**，并用$\vdash r : \tau$表示存在一棵依照以下规则构造、且以$\vdash r : \tau$为根的推导树：

$$\frac{\vdash \tau : \star}{\vdash x{:}\tau : \tau}\ (\mathbf{Tv}) \qquad \frac{\vdash \tau : \star}{\vdash \mathsf{C}{:}\tau : \tau}\ (\mathbf{Tc}) \qquad \frac{\vdash r : \tau \to \tau' \quad \vdash s : \tau}{\vdash rs : \tau'}\ (\mathbf{Ta}) \qquad \frac{\vdash r : \tau' \quad \vdash \tau : \star}{\vdash \lambda x{:}\tau.\ r : \tau \to \tau'}\ (\mathbf{Tl})$$

若对于某个 τ，有$\vdash r : \tau$，则称 r **类型良好**——同样，通常我们始终处理类型良好的项。此外，若$\vdash r : \mathtt{Prop}$，则称 r 为一个**公式**；我们用 ϕ、ψ、ξ 等任意表示希望理解为公式的项。同样，这个类型关系满足一些显然的正确性性质：

▶ 引理 *15.* 若 $\lq r : \tau$，则 $\lq \tau : \star$.▶ 引理 *16.* 若 $\lq r : \tau$ 且 $\lq r : \tau'$，则 $\tau = \tau'$.▶ 引理 *17.* 若 $\lq r : \tau$ 且 $\lq \tau' : \star$，则 $\lq r[\beta := \tau'] : \tau[\beta := \tau']$.▶ 引理 *18.* 若 $\lq r : \tau$ 且 $\lq s : \tau'$，则 $\lq r[y{:}\tau' := s] : \tau$.

最后请注意，类型为 $\mathtt{Prop}$ 的相等与双条件相一致。我们假设另有一个类型为 $\mathtt{Prop} \to \mathtt{Prop} \to \mathtt{Prop}$ 的常量 $\longleftrightarrow$，并用 $\phi \longleftrightarrow \psi$ 代替 $\phi = \psi$。

2.2 自然演绎

我们对高阶逻辑最显著的偏离，是引入了污点跟踪，将 § 中描述的非正式过程形式化1。高阶逻辑通常以自然演绎的形式呈现，其中自然演绎关系是上下文与公式之间的二元关系， $\Gamma \vdash \phi$。我们将这一关系扩展为三元关系，关联上下文、公式和污点标签， $\Gamma \vdash \phi : \ell$。

我们将通过在任意格上构造的核心逻辑，来介绍自然演绎的扩展。用于“划分”逻辑的具有基础理论意义的公理[5]应如何选择，取决于个人偏好，也反映了用户感兴趣的数学推理风格。只要引入污点的公理及其生成的格满足某些性质，这些公理就可以有所变化。尽管如此，我们稍后仍会通过一些公理扩展该逻辑，以刻画若干有趣的数学体系，并讨论一种新的定义性原则，用于动态添加公理及相关污点标签。

为此，我们首先固定一组污点标签 L，并用 ℓ、 ℓ'、 ℓ'' 等任意表示污点标签。我们还在污点标签上固定一个具有封闭性的二元运算 $-\mathbin{\mathsf{t}}-$，使得只要 $\ell \in \mathrm{L}$ 且 $\ell' \in \mathrm{L}$，就有 $\ell \mathbin{\mathsf{t}} \ell' \in \mathrm{L}$。我们假设污点标签集合 L 至少包含指定标签 I，不过之后还会假设存在更多污点标签。我们还引入污点标签之间可导等价的概念，其规则见图 1。我们写作 $\ell \equiv \ell'$，表示存在一棵按照图 1 中的规则构造、且根节点为 $\ell \equiv \ell'$ 的推导树。

$$\frac{}{\vdash \ell \equiv \ell}\,(\mathbf{Lrefl}) \qquad \frac{\vdash \ell \equiv \ell'}{\vdash \ell' \equiv \ell}\,(\mathbf{Lsym}) \qquad \frac{\vdash \ell \equiv \ell' \quad \vdash \ell' \equiv \ell''}{\vdash \ell \equiv \ell''}\,(\mathbf{Ltrns}) \qquad \frac{}{\vdash \ell \sqcup \ell \equiv \ell}\,(\mathbf{Lidm})$$

$$\frac{}{\vdash \ell \sqcup (\ell' \sqcup \ell'') \equiv (\ell \sqcup \ell') \sqcup \ell''}\,(\mathbf{Lassc}) \qquad \frac{\vdash \ell \equiv \ell' \quad \vdash \ell'' \equiv \ell'''}{\vdash \ell \sqcup \ell'' \equiv \ell' \sqcup \ell'''}\,(\mathbf{Lcng}) \qquad \frac{}{\vdash I \sqcup \ell \equiv \ell}\,(\mathbf{Lbase})$$

图 1 污点标签之间的等式理论

上述规则赋予污点标签一种**格结构**，并以代数形式呈现于此。与所有格一样，我们也可以用序理论方式来呈现，并写作 $\vdash \ell \leq \ell'$，表示 $\vdash \ell \sqcup \ell' \equiv \ell'$。我们称此关系为污点标签上的**导出序**。代数形式和序理论形式的呈现都很有用。此格**有下界**，其中 I 是相对于此导出序的最小元，也就是说，对于所有污点标签 ℓ，都有 $\vdash I \leq \ell$。标签 I 表示纯构造性推理。此序满足以下若干性质：

▶ **引理 19.** 若 $\ell \equiv \ell'$，则 $\ell \leq \ell'$。▶ **引理 20.** 若 $\ell \leq \ell'$ 且 $\ell' \leq \ell''$，则 $\ell \leq \ell''$。▶ **引理 21.** 若 $\ell \leq \ell'$ 且 $\ell' \leq \ell$，则 $\ell \equiv \ell'$。▶ **引理 22.** 若 $\ell \leq \ell'$ 且 $\ell'' \leq \ell'''$，则 $\ell \mathbin{\mathsf{t}} \ell'' \leq \ell' \mathbin{\mathsf{t}} \ell'''$。

我们将有限的项集合称为**上下文**，并用 Γ、 Γ'、 Γ''等任意表示上下文。若 $\vdash \phi : \mathtt{Prop}$ 对每个 $\phi \in \Gamma$都成立，我们就称上下文 Γ**有效**，并写作 $\vdash \Gamma$ valid，以断言上下文 Γ 有效。

[5] 我们仍不确定究竟什么才算是一个在基础层面上有趣的公理，但我们认为，这类公理应以某种有趣的方式改变数学对象的全集，或改变关于这些对象能够推导出的内容。

XX：8 一个温和的提议：明确支持基础多元主义

▶**引理 23**。我们有：
1. $\vdash\{\}$ 有效，2. 若 $\vdash \Gamma$ 有效且 $\vdash \Gamma'$ 有效，则 $\vdash \Gamma \cup \Gamma'$ 有效。

我们用 $fv(\Gamma)$ 表示上下文 Γ 的**自由变量**，即集合 $\bigcup\{fv(\phi) \mid \phi \in \Gamma\}$。我们还定义了上下文上的**逐点类型替换作用**，$\Gamma[\beta := \tau]$ 通过$\Gamma[\beta := \tau] = \{\phi[\beta := \tau] \mid \phi \in \Gamma\}$。

▶ **引理 24**。如果‘ Γ 有效且 ‘ $\tau :\ \star$，那么 ‘ $\Gamma[\beta :\ = \tau]$ 有效。

现在我们定义一个**自然演绎**关系，关联上下文、项和污点标签，$\Gamma \vdash r :\ \ell$，使用图2中的规则。我们写作 $\Gamma \vdash r :\ \ell$，以断言存在一棵按照图2中的规则构造、且根位于 $\Gamma \vdash r :\ \ell$的推导树。

▶ **引理 25**。如果 $\Gamma \vdash \phi :\ \ell$，那么$\vdash \phi :\ \texttt{Prop}$。

▶ **引理 26**。 如果 Γ ‘ $\phi :\ \ell$，那么 ‘Γ 有效。

▶ **引理 27**。 如果 $\Gamma \vdash \phi$: ℓ 那么 $\ell \in \mathcal{L}$。

▶ **引理 28**。 如果 Γ ‘ $\phi :\ \ell$ 且 $\Gamma \subseteq \Gamma'$，并且 ‘ Γ' 有效，那么 Γ' ‘ $\phi :\ \ell$。

直观而言，污点表示对推导树进行的一种简单静态分析，其中具有基础理论意义的公理实例会将污点标签“注入”树中。到目前为止，这些公理仍未出现在上述规则中，但最终会包括排中律、选择公理等类似公理，并引入相应的污点标签。逻辑的推理规则经过修改，允许这些污点标签沿推导树逐级“向下传递”，最终出现在根节点处。合取引入规则就是一个很有说明性的例子：

$$\frac{\Gamma \vdash \phi : \ell \quad \Gamma \vdash \psi : \ell}{\Gamma \vdash \phi \wedge \psi : \ell}\ (\mathbf{NconjI})$$

两个前提中出现的污点标签完全相同，只是经由该规则传递而来。所有推理规则都以这种方式修改，仅仅向前传播污点标签。因此，任何具有基础理论意义的公理最终都会反映在推导树的根节点处，这也解释了“污点”这一名称。为了在前提中弱化污点标签——并最终例如从构造性证明转向经典证明——我们还引入了一条**嵌入规则**，它与子类型规则十分相似：

$$\frac{\Gamma \vdash \phi : \ell \quad \vdash \ell \leq \ell'}{\Gamma \vdash \phi : \ell'}\ (\mathbf{Nlift})$$

请注意，在设计我们的自然演绎关系时，我们面临一个设计选择：与其强制显式使用上文的嵌入规则，以便在前提中获得共同的污点标签，我们也可以将合取引入等规则表述如下：

$$\frac{\Gamma \vdash \phi : \ell \quad \Gamma \vdash \psi : \ell'}{\Gamma \vdash \phi \wedge \psi : \ell \sqcup \ell'}$$

这里，$\ell \sqcup \ell'$是污点标签 ℓ 和 ℓ' 的最小上界，它本身也是取自我们格中的一个污点标签，能够汇集污点并在证明过程中恰当地传播污点。无需额外诉诸嵌入规则，该系统便能计算出精确的 污点标签，描述某个结果成立所依赖的逻辑的最弱子系统，

$$\frac{\vdash \Gamma \text{ valid} \quad \phi \in \Gamma}{\Gamma \vdash \phi : I}\ (\textbf{Ninit}) \qquad \frac{\vdash \Gamma \text{ valid}}{\Gamma \vdash \top : I}\ (\textbf{NtrueI}) \qquad \frac{\Gamma \vdash \bot : \ell \quad \vdash \phi : \texttt{Prop}}{\Gamma \vdash \phi : \ell}\ (\textbf{NfalseE})$$

$$\frac{\Gamma \vdash \phi : \ell \quad \vdash \ell \leq \ell'}{\Gamma \vdash \phi : \ell'}\ (\textbf{Nlift}) \qquad \frac{\vdash \Gamma \text{ valid} \quad \vdash r : \tau}{\Gamma \vdash r = r : I}\ (\textbf{Nrefl}) \qquad \frac{\Gamma \vdash r = s : \ell}{\Gamma \vdash s = r : \ell}\ (\textbf{Nsym})$$

$$\frac{\Gamma \vdash r = s : \ell \quad \Gamma \vdash s = t : \ell}{\Gamma \vdash r = t : \ell}\ (\textbf{Ntrans}) \qquad \frac{\Gamma \vdash r = s : \ell \quad x{:}\tau \notin fv(\Gamma)}{\Gamma \vdash \lambda x{:}\tau.\ r = \lambda x{:}\tau.\ s : \ell}\ (\textbf{Nlcong})$$

$$\frac{\Gamma \vdash f = g : \ell \quad \Gamma \vdash r = s : \ell \quad \vdash fr : \tau'}{\Gamma \vdash fr = gs : \ell}\ (\textbf{Nacong}) \qquad \frac{\Gamma \vdash \phi : \ell \quad \vdash r : \tau}{\Gamma[x{:}\tau := r] \vdash \phi[x{:}\tau := r] : \ell}\ (\textbf{Nsubst})$$

$$\frac{\vdash \Gamma \text{ valid} \quad \vdash \lambda x{:}\tau.\ r : \tau \to \tau' \quad \vdash s : \tau}{\Gamma \vdash (\lambda x{:}\tau.\ r)s = r[x{:}\tau := s] : I}\ (\textbf{Nbeta}) \qquad \frac{\Gamma \vdash \phi : \ell \quad \vdash \tau : \star}{\Gamma[\beta := \tau] \vdash \phi[\beta := \tau] : \ell}\ (\textbf{Ninst})$$

$$\frac{\vdash \Gamma \text{ valid} \quad \vdash \lambda x{:}\tau.\ (fx) : \tau \to \tau' \quad x{:}\tau \notin fv(f)}{\Gamma \vdash \lambda x{:}\tau.\ (fx) = f : I}\ (\textbf{Neta}) \qquad \frac{\Gamma \cup \{\phi\} \vdash \bot : \ell}{\Gamma \vdash \neg\phi : \ell}\ (\textbf{NnegI})$$

$$\frac{\Gamma \vdash \phi : \ell \quad \Gamma \vdash \neg\phi : \ell}{\Gamma \vdash \bot : \ell}\ (\textbf{NnegE}) \qquad \frac{\Gamma \vdash \phi \longleftrightarrow \psi : \ell \quad \Gamma \vdash \phi : \ell}{\Gamma \vdash \psi : \ell}\ (\textbf{NiffE1}) \qquad \frac{\Gamma \vdash \phi : \ell \quad \vdash \psi : \texttt{Prop}}{\Gamma \cup \{\psi\} \vdash \phi : \ell}\ (\textbf{Nwk})$$

$$\frac{\Gamma \vdash \phi \longleftrightarrow \psi : \ell \quad \Gamma \vdash \psi : \ell}{\Gamma \vdash \phi : \ell}\ (\textbf{NiffE2}) \qquad \frac{\Gamma \cup \{\psi\} \vdash \phi : \ell \quad \Gamma \cup \{\phi\} \vdash \psi : \ell}{\Gamma \vdash \phi \longleftrightarrow \psi : \ell}\ (\textbf{NiffI})$$

$$\frac{\Gamma \vdash \phi : \ell \quad \Gamma \vdash \psi : \ell}{\Gamma \vdash \phi \wedge \psi : \ell}\ (\textbf{NconjI}) \qquad \frac{\Gamma \vdash \phi \wedge \psi : \ell}{\Gamma \vdash \phi : \ell}\ (\textbf{NconjE1}) \qquad \frac{\Gamma \vdash \phi \wedge \psi : \ell}{\Gamma \vdash \psi : \ell}\ (\textbf{NconjE2})$$

$$\frac{\Gamma \vdash \phi : \ell \quad \vdash \psi : \texttt{Prop}}{\Gamma \vdash \phi \vee \psi : \ell}\ (\textbf{NdisjI1}) \qquad \frac{\Gamma \vdash \phi \vee \psi : \ell \quad \Gamma \cup \{\phi\} \vdash \xi : \ell \quad \Gamma \cup \{\psi\} \vdash \xi : \ell}{\Gamma \vdash \xi : \ell}\ (\textbf{NdisjE})$$

$$\frac{\Gamma \vdash \psi : \ell \quad \vdash \phi : \texttt{Prop}}{\Gamma \vdash \phi \vee \psi : \ell}\ (\textbf{NdisjI2}) \qquad \frac{\Gamma \cup \{\phi\} \vdash \psi : \ell}{\Gamma \vdash \phi \longrightarrow \psi : \ell}\ (\textbf{NimpI}) \qquad \frac{\Gamma \vdash \phi \longrightarrow \psi : \ell \quad \Gamma \vdash \phi : \ell}{\Gamma \vdash \psi : \ell}\ (\textbf{NimpE})$$

$$\frac{\Gamma \vdash \forall x{:}\tau.\ \phi : \ell \quad \vdash r : \tau}{\Gamma \vdash \phi[x{:}\tau := r] : \ell}\ (\textbf{NallE}) \qquad \frac{\Gamma \vdash \phi : \ell \quad x{:}\tau \notin fv(\Gamma)}{\Gamma \vdash \forall x{:}\tau.\ \phi : \ell}\ (\textbf{NallI}) \qquad \frac{\Gamma \vdash \phi[x{:}\tau := r] : \ell}{\Gamma \vdash \exists x{:}\tau.\ \phi : \ell}\ (\textbf{NexI})$$

$$\frac{\Gamma \vdash \exists x{:}\tau.\ \phi : \ell \quad \Gamma \cup \{\phi[x{:}\tau := y{:}\tau]\} \vdash \psi : \ell \quad y{:}\tau \notin fv(\Gamma) \cup fv(\phi)}{\Gamma \vdash \psi : \ell}\ (\textbf{NexE})$$

图 2 自然演绎关系的核心规则

代价是要求每两个污点标签都有一个最小上界。[6] 在这个系统中，嵌入规则是冗余的：规则隐含地“内置”了弱化，而且嵌入规则与 其他每条规则都可交换；推导树可以通过降低高度的方式重写，使得嵌入规则（若存在）只出现在树根处。

然而，这种方法会使计算机上的策略驱动的反向证明变得复杂，因为一条规则的前提所带的污点标签集合可能与其结论不同。相比之下，我们倾向采用的表述有利于简化计算机实现，因为在大多数情况下，除了（**Nlift**）之外，策略生成的新子目标的前提中出现的污点标签，与被简化的原始子目标中的污点标签相同。在这种方案中，由于污点标签存在最小上界，一条规则的前提总能嵌入逻辑的一个共同片段中。尽管如此，经过修改的规则（例如上面那条）以及其他类似规则，包括以下备选的析取消去规则

$$\frac{\Gamma \vdash \phi \vee \psi : \ell \quad \Gamma \cup \{\phi\} \vdash \xi : \ell' \quad \Gamma \cup \{\psi\} \vdash \xi : \ell''}{\Gamma \vdash \xi : (\ell \sqcup \ell') \sqcup \ell'}$$

都是我们系统中的**可导规则**，这要归功于嵌入规则以及以下事实：对于所有污点标签 ℓ 和 ℓ'，$\ell \leq \ell \sqcup \ell'$。请注意，由于 $-\sqcup-$ 满足结合律，且以下结果表明自然演绎规则尊重关于污点标签的等式理论，因此上述规则是定义良好的：

▶ **引理 29**。若 $\Gamma \vdash \phi : \ell$ 且 $\vdash \ell \equiv \ell'$，则 $\Gamma \vdash \phi : \ell'$。

还可以找到其他导出规则。例如，下面的规则类似于切割 规则，可用于反向证明，并且必须考虑“污点”：

▶ 引理 *30*。*1.* 若 $\Gamma \vdash \phi : \ell$ 且 $\Gamma \cup \{\phi\} \vdash \psi : \ell$，则 $\Gamma \vdash \psi : \ell$。
2. 若 $\Gamma \vdash \phi : \ell$ 且 $\Gamma \cup \{\phi\} \vdash \psi : \ell'$，则 $\Gamma \vdash \psi : \ell \sqcap \ell'$。

2.3 添加基础公理

在我们的逻辑中猜想某个结果时，污点标签会对证明过程中可以使用的推理类型设定一个上界。然而，要观察到这一点，我们首先需要更多污点标签。假设 $\mathcal{L}$ 包含一个额外的特殊标签 C，并分别更新图 1 中的等式理论和图2中的自然演绎关系，通过添加排中律来引入一个新的“经典世界”：

$$\frac{\vdash \Gamma \text{ valid} \quad \vdash \phi : \mathtt{Prop}}{\Gamma \vdash \phi \vee \neg\phi : C}\,(\mathbf{Nlem}) \qquad \frac{}{\vdash C \sqcup I \equiv C}$$

请注意，$\vdash I \leq C$ 由于上述结果，结合嵌入规则，带有 I 标记、属于“构造主义世界”的结果便可提升到由 C 标记的“经典世界”中。此外，经典世界中的成员立即可以使用新的推理原则，而构造主义者则无法使用，例如命题分类讨论和 归谬法，它们表示为导出规则：

$$\frac{\Gamma \cup \{\phi\} \vdash \xi : \ell \quad \Gamma \cup \{\neg\phi\} \vdash \xi : \ell \quad \vdash \phi : \mathtt{Prop}}{\Gamma \vdash \xi : C} \qquad \frac{\Gamma \cup \{\neg\phi\} \vdash \bot : \ell}{\Gamma \vdash \phi : C}\,(\mathbf{Nraa})$$

[6] 请注意，这在某种程度上让人想起逆向数学 [15]。

我们引入更多基础公理，以表示若干常见的公理化基础体系。与上文相同，我们假定还有更多标签， W 和 Ch，并通过修改图1中的规则，使它们在句法上与其他标签建立联系，具体如下：

$$\frac{}{\vdash W \sqcup I \equiv W} \qquad \frac{}{\vdash C \sqcup W \equiv C} \qquad \frac{}{\vdash Ch \sqcup \ell \equiv Ch}$$

P除此之外，$\vdash I \leq W \leq C \leq Ch$。这些标签与更多基础公理相关联 ：

$$\frac{\vdash \Gamma \text{ valid} \quad \vdash \phi : \texttt{Prop}}{\Gamma \vdash \neg\phi \vee \neg\neg\phi : W}\,(\star) \qquad \frac{\vdash \Gamma \text{ valid} \quad \vdash P : \alpha \to \beta \to \texttt{Prop}}{\Gamma \vdash \forall x.\ \exists y.\ P\ x\ y \longrightarrow \exists f.\ \forall x.\ P\ x\ (f\ x) : Ch}\,(\star\star)$$

这里，标注为（$\star$）的规则导出 Jankov 逻辑，也就是弱排中律 [23], 的逻辑；它是一种弱于经典逻辑的超直觉主义逻辑，而标注为（$\star\star$）的规则则引入选择公理。每条新公理都满足与其他所有基础公理之间所需的蕴涵关系——请记住，根据 Diaconescu 定理，选择公理蕴涵其他每一条公理，并且是污点标签序的上界。

添加新标签时必须谨慎，确保句法污点标签格确实仍为一个格，并且这个句法格中的标签必须始终反映底层语义格中不同基础公理之间的蕴涵关系。我们将这种反映称为**相干**性质。两个污点标签，$\{\ell, \ell'\} \subseteq \mathcal{L}$，满足关系$\vdash\ \ell \leq \ell'$，当且仅当与标签相关联的公理 ℓ 可由与 ℓ'相关联的公理推出。具体来说，这里，排中律蕴涵弱排中律，因此$\vdash W \leq C$，但反之不成立。我们将在§3 讨论定义原则时再回到这个话题。

最后请注意，逻辑对于向其中添加哪一种逻辑等价的基础公理聚集，态度是模棱两可的。已知许多与选择公理等价的结果，包括佐恩引理、塔斯基定理，以及每个满射都有一个单射逆这一事实。显然，这些结果中的每一个都可以作为我们逻辑的 Ch片段中的导出规则给出。不过，也可以将它们分别作为独立的基础公理引入——尽管对某些结果而言，这样做或许不太方便——并为每个公理配上专属的污点标签。由于两个与选择公理等价的命题之间存在双条件，且$\vdash\ \ell \leq \ell'$ 与$\vdash\ \ell' \leq \ell$ 蕴含$\vdash\ \ell \equiv \ell'$，这个新标签可导出地等于标签 Ch，因此任一世界中的结果都能立即用于另一个世界。

3 数据与定义原则

为便于展示，我们在系统中直接断言**严格正数据**和**归纳定义关系**的存在，以及相应的推理原则，而不通过保守扩张来构造它们。数据类型 `Bool` 将证明很有用；它由**构造子**引入，并配有**原始递归子**，可概括如下：

`true`, `false`	with type	`Bool`
`ite`		$\texttt{Bool} \to \alpha \to \alpha \to \alpha$

我们再次体谅读者，使用 `if` r `then` t `else` f `end` 代替 $((\texttt{ite}\ r)t)f$，并继续省略类型代换，以使项类型良好。我们还假设存在公理，断言 `Bool` 的构造子是自由的，所有构造子彼此不同且具有单射性；同时也假设存在描述递归子等式性质的公理。最后，我们还假设一条**结构归纳规则**，用于推导类型为 `Bool`的元素所具有的性质。所有公理都存在于 I（纯构造主义世界）中，并且可以自由地提升到任何其他世界。

集合是一种有用的数据结构，在现代数学中无处不在。与高阶逻辑相比，我们的逻辑在定义集合的方式上提供了更细粒度的选择，因为我们可以

XX:12 一个温和的提议：明确支持基础多元主义

将集合定义为指向 Prop 或 Bool 的谓词。二者都将发挥作用——具体取决于我们所处的世界——因为前者更适用于规格说明，后者更适用于定义，至少在除最经典的世界之外的所有世界中都是如此。对于由基于 Prop 的谓词构造的集合，我们保留**集合**这一术语；对于基于 Bool 的谓词，则使用**聚集**这一术语。

为引入集合，我们固定一个额外的类型构造器 Set，其种类为 $\star \Rightarrow \star$，使得对于所有类型 τ，Set τ 都是一个类型，并将其设为 $\tau \to$ Prop 的类型同义词。假设另有一个多态常量 $-\in-$，其类型为 $\alpha \to$ Set $\alpha \to$ Prop；分别用 $x \in S$ 和 $\{x : \tau \mid \phi\}$ 作为 $S\ x$ 和 $\lambda x{:}\tau.\ \phi$ 的缩写。借此，我们可以用“显而易见”的方式定义更多具有预期类型的集合运算，例如：

$\emptyset$		Set α		$\{x : \alpha \mid \bot\}$
UNIV		Set α		$\{x : \alpha \mid \top\}$
$- \cup -$	with type	Set $\alpha \to$ Set $\alpha \to$ Set α	defined by	$\lambda S.\ \lambda T.\ \{x : \alpha \mid x \in S \vee x \in T\}$
cmpl		Set $\alpha \to$ Set α		$\lambda S.\ \{x : \alpha \mid \neg(x \in S)\}$

显然，根据所处的世界不同，从这些定义得到的集合论也会大相径庭：

$$\frac{\vdash \Gamma \text{ valid} \quad \vdash S, T : \texttt{Set}\ \tau}{\Gamma \vdash S \cup T = T \cup S : I} \qquad \frac{\vdash \Gamma \text{ valid}}{\Gamma \vdash \texttt{cmpl}\ \emptyset = \texttt{UNIV} : I} \qquad \frac{\vdash \Gamma \text{ valid}}{\Gamma \vdash \texttt{cmpl} \circ \texttt{cmpl} = \texttt{id} : C}\ (\dagger)$$

这里，在标注为 (†) 的导出规则中，我们隐式地使用了一个类型为 $(\alpha \to \beta) \to (\beta \to \gamma) \to \alpha \to \gamma$ 的**函数复合**常量 $-\circ-$，以及一个类型为 $\alpha \to \alpha$ 的**恒等**函数 id，二者均采用标准定义。请注意，这条导出规则在构造主义世界 I 中并不成立；其表述方式是为了最方便地运用逻辑的自然外延性，具体如下述导出规则所示：[7]

$$\frac{\Gamma \vdash f\ x = g\ x : \ell \quad x \notin fv(\Gamma)}{\Gamma \vdash f = g : \ell}$$

这种外延性也体现在我们的内部集合论中：在所有可能世界里，如果两个集合包含相同的元素，就认为它们相等；这与纸笔数学和其他 HOL 变体中的标准做法一致：

$$\frac{\Gamma \vdash x{:}\tau \in S \longleftrightarrow x{:}\tau \in T : \ell \quad x{:}\tau \notin fv(\Gamma)}{\Gamma \vdash S = T : \ell}$$

回到 Prop 和 Bool 之间的区别，我们有一个类型为 Bool $\to$ Prop 的嵌入函数 lift，它可以用 ite 常量定义，并按分段形式描述如下：

$$\frac{\vdash \Gamma \text{ valid}}{\Gamma \vdash \texttt{lift true} = \top : I} \qquad \frac{\vdash \Gamma \text{ valid}}{\Gamma \vdash \texttt{lift false} = \bot : I}$$

在较不经典的世界中，我们无法得到类型为Prop $\to$ Bool的逆函数，因为我们不能对 Prop 的元素按情况进行推理。不过，在可以使用选择公理的 Ch 世界中，情况便有所不同。引入关系 R：Prop $\to$ Bool $\to$ Prop，使得 $R \top$ true 和 $R \bot$ false 均成立，我们便有：

$$\dfrac{\dfrac{\dfrac{\vdash \Gamma \text{ 有效}}{\vdots}}{\Gamma \text{ `` } \forall x.\ \exists y.\ R\ x\ y : C} \qquad \dfrac{\vdash \Gamma \text{ 有效}}{\Gamma \vdash\ \forall x.\ \exists y.\ R\ x\ y \longrightarrow \exists f.\ \forall x.\ R\ x\ (f\ x) : Ch}}{\Gamma \vdash\ \exists f.\ \forall x.\ R\ x\ (f\ x) : Ch}$$

[7] 如果需要，我们还可以通过调整 η-公理，引入一个表示构造性外延推理的新标签，并重新诠释 I，以进一步区分内涵推理和外延推理。

缺失的证明可作为 C中排中律的推论得出。由此我们得到 drop： $\texttt{Prop} \to \texttt{Bool}$ ，且可证明 lift 和 drop 互为逆函数。这样一来， Prop 与 Bool 之间的区别便不复存在，集合与聚集之间的区别亦然。

换个角度来看，Gordon 在高阶逻辑中相较于 Church 的简单类型理论所做的一项创新，是引入了一条定义规则：利用已有宿主类型中元素的非空子集，从旧类型“刻画出”新类型。在 HOL 系列定理证明系统中引入这一定义性原则，较早的 LCF 系列是一个进步；后者引入新类型主要依赖公理化。事实上，这种机制是从头构造任意数据类型的一种方式——我们在上文只是直接假定了这些数据类型，而许多 HOL 实现确实采用了这种做法。

有趣的是，我们可以引入类似机制来定义类似的**子集类型**，不过似乎每个世界本质上都有自己的版本，对应于各个世界中不同的存在概念。例如，在经典世界 C 中，我们将存在理解为与不存在的不可能性相一致：

$$\frac{\Gamma \cup \{\forall x.\ \neg(\phi\ x)\} \vdash \bot : \ell}{\Gamma \vdash \exists x.\ \phi\ x : C}$$

这当然不同于构造主义世界 I所采用的构造性存在概念；后者要求明确构造一个见证。鉴于此，考虑通过归纳将集合的**有限性**定义为一种关系， finite 其类型为 $\texttt{Set}\ \alpha \to \texttt{Prop}$:

$$\frac{\vdash \Gamma \text{ valid}}{\Gamma \vdash \texttt{finite}\ \emptyset : I} \qquad \frac{\Gamma \vdash \texttt{finite}\ S : \ell \quad \vdash S : \texttt{Set}\ \alpha}{\Gamma \vdash \texttt{finite}\ (S \cup \{x{:}\alpha\}) : \ell}$$

然后，我们可以使用 finite从 $\texttt{Set}\ \alpha$ 类型中“划出”一个显式的有限集类型 $\texttt{Fset}\ \alpha$。为此，我们要确立满足 finite的集合 S 的存在；这可以通过构造性方式在 I中完成，并给出一个明确的见证——例如 $\emptyset$——也可以通过经典方式在 C 或 Ch中完成。由此，我们得到单射函数 **单射** 和投影函数 **投影** ，分别将元素映入和映出这个新类型，二者由以下一对定律联系起来:

$$\frac{\vdash \Gamma \text{ valid}}{\Gamma \vdash \texttt{proj} \circ \texttt{inj} = \texttt{id} : \ell} \qquad \frac{\vdash \Gamma \text{ valid} \quad \vdash S : \texttt{Set}\ \alpha}{\Gamma \vdash \texttt{finite}\ S \longrightarrow \texttt{proj}\ (\texttt{inj}\ S) = S : \ell}$$

借助这些函数，我们可以将定义在集合上的函数“提升”到新类型中，并通过“降回”用于划出新类型的底层宿主类型，来确立这些提升后函数的性质。这里，我们推测，上述规则结论中出现的标签 ℓ 应与划出过程中用于确立某个见证存在性的标签一致。

最后，我们推测性地考虑一种用于污点的新定义性原则的可能性。到目前为止，我们一直使用任意的有界格，并在上一节对其进行了专门化。如前所述，这种专门化的具体形式取决于个人喜好。另一种做法是，增加一条用于动态添加污点的新原则，并在系统内部通过证明将新污点与旧污点关联起来，从而引入一个基础公理的“开放世界”。

具体而言，对我们的污点，一种相当朴素的理解是将其视为一种省力工具：污点消除了在证明中手动贯穿公理的需要，从而消除了妨碍高效工作的障碍。如前所述，由此便产生了污点标签的格结构，反映出由标签表示的公理之间的蕴涵关系——这就是我们的一致性性质。因此，我们有如下**展开性质**，它通过对（广义）陈述进行相对化，使我们能够将一个经典结果“放回”构造主义世界：

▶ **引理 31**。若 $\Gamma \vdash\ \psi :\quad C$ 则 $\Gamma \vdash (\forall \phi.\ \phi \vee \neg\phi) \longrightarrow \psi :\quad I$。

我们将其视为添加标签时的正确性概念：添加新标签时，我们保持一致性，保留污点标签格结构，并通过找到一个现有标签，使其成为新标签与其他每个标签的最小上界，从而使与每个污点标签相关联的基础公理相对于与其他所有标签相关联的公理**得到定位**。[8]

然而，我们必须谨慎，避免被过于简单、纯粹蕴涵式的消解处理所迷惑，如上文引理 31 所示。如果公理只提及基础类型（如排中律），这种做法确实可行；但当涉及包含多态类型的公理时——例如选择公理——它就行不通了。在这里，高阶逻辑的类型系统迫使我们区分两种公式：一种出现在包含多态类型的定理陈述中，另一种则作为推理规则的结论出现。出现在公式中的类型会被固定，在证明过程中无法变化；而出现在推理规则中的类型则可以任意变化，这通过 (**Ninst**) 规则实现。

因此，我们进一步猜想，可以在放置新污点标签的过程中暂时“关闭”污点系统，从而将新污点标签与格中已有的污点标签建立关系。例如，假设目前我们的系统仅包含污点标签 I 和 C，那么我们可以添加与 Jankov 逻辑相关的新污点标签 W：若有 $\Gamma \vdash \neg\phi \vee \neg\neg\phi : I$ 的证明，则证明 $\Gamma \vdash \top : I$；同理，若有 $\Gamma \vdash \neg\phi \vee \phi : I$的证明，则证明 $\Gamma \vdash \neg\phi \vee \neg\neg\phi : I$。这些证明必须在纯粹的构造性逻辑片段中完成——不得援引任何其他基础公理——并且所有公理都必须暂时处于“构造主义世界” I中。由此可得$\vdash I \sqcup W = W$ 和$\vdash W \sqcup C = C$，进而得到$\vdash I \leq W \leq C$，符合预期。这个过程很容易推广到考虑与某个污点标签相关的多个公理，而非单个公理。

这种基于证明的方法只能确立某个标签是另外两个标签的上界，而不是最小上界。要确立这一点，我们还需要考虑反证，检查与其他任何标签相关联的公理是否也能作为上界。不过，要做到这一点，可能需要更复杂的方法，例如模型论技术，而这些方法无法在逻辑内部实现。若要容纳污点的“开放世界”，那么任何用于添加新污点的定义性原则都必须纳入逻辑的可信基础，并且每次使用都要经过审计，以确保放置过程中使用的上界确实是最小上界。我们将这一新定义性原则的完整形式化留待后续工作。

4 数学编码的语用学

此前，我们谈到了排除任何类似希尔伯特式描述算子的东西，ϵ。因此，我们无法定义那些原本需要 ϵ 的概念，例如函数的逆，或有序集的极小元素；而在高阶逻辑中则可以这样做。相反，我们必须使用相对化：先定义一个类型为 $(\alpha \to \beta) \to (\beta \to \alpha) \to \texttt{Prop}$ 的谓词 `isinv`，其中 `isinv` $f\ g$ 断言函数 g 是 f 的函数的逆。定义和定理陈述——原本会用函数的逆来表述，f——都以函数 g 为参数，并带有显式假设 `isinv` $f\ g$，该假设可通过证明消除。这显然不那么方便，但也体现出令人赞赏的公平进步——无论信仰或哲学宗教立场如何，每个人都同样处于劣势。至于探索通过自动化或类似方式减轻这种不便，则留待未来工作。

此外，我们系统一个有趣之处在于，它强烈鼓励人们尝试用足以成立的最弱系统来表述结果，因为这样做能最大限度地

[8] 在此，我们将 $\top$ 视为与构造主义污染标签 I相关联的基础公理。

复用不同世界中的结果。这是因为所有人都基于同一组定义进行推理，也使用相同的类型：例如，构造主义者有自己的自然数类型，而经典推理器使用的则是另一种类型，这种情况并不存在——(Nlift)使得结果能够轻松地在不同世界之间转换。因此，身处较少经典的世界中的人，可能需要引入定义、类型和新的定理陈述，以区分那些倾向于更经典的人无法观察到的概念；随着人们采用越来越经典的推理方式，这些概念也会随之消失。这种现象的一个例子是我们已经讨论过的概念，即集合的有限性；在构造主义框架下，它可以细分为若干彼此相关的不同概念。同样的观察也适用于无限集、域的定义、分析学中的各种结果，等等。

请注意，我们在 §3 中选择将严格正数据类型直接嵌入逻辑，这才使我们承诺了无限集的存在，例如自然数集。大多数 HOL 实现都明确包含一条断言无限集存在的公理，并通过保守扩张的过程用它来“构造”其他数据类型。

最后，我们能够合理捕捉的特定构造主义流派，受限于这样一个事实：与 HOL 一样，我们的逻辑是非谓述的。这使我们的构造主义流派在精神上类似于 Coq 和 Matita 的非谓述宇宙中实现的构造主义，但不同于 Agda、Coq 和 Matita 的谓述宇宙中实现的谓述式构造主义。我们逻辑的内部集合论也别具特色：可以自由构造幂集，因此与某些其他流派的构造性集合论相悖；在那些理论中，幂集的构造受到严格限制 [11]。尽管如此，我们认为，我们的构造主义在精神上类似于主流数学中的非形式构造主义。这里，“构造性”指不使用排中律、选择公理或类似原则，并在存在性证明中描述明确的见证。

5 实现

我们编写了一个原型 LCF 风格的 [30] 证明检查内核，用于实现我们的逻辑[9]，它以 *Scala 3* 实现。按照惯例，推导树由一个抽象类型表示——在我们的案例中，它是一个类 ，用于表示 `Thm` 对象。我们还支持由策略驱动的基本反向证明；我们的自然演绎中的每条推理规则和公理，以及若干有用的导出规则，都与一个基本策略相关联。我们的原型系统还提供了策略组合器——也就是高阶策略值泛函——用于从简单构件构建更大、更复杂的策略。在我们的系统中，它们以深嵌入 的策略领域特定语言 [27,28,34] 实现，并由系统解释，从而对证明状态执行操作。

借此，我们可以举例说明，本系统能够正确地认证 皮尔士定律属于“经典世界”，方法是证明猜想 $\{\}\vdash (\phi \to \psi) \to \phi \to \phi : C$。这源于经典的 归谬法原则，该原则由（**Nraa**）导出规则所体现，而本系统则提供了与此原则相关联的显式策略。尽管总体结果是经典的，子证明却只需要纯构造性推理。因此，可以使用一种反转（**Nlift**）规则作用的策略，将这些纯构造性子证明显式提升到“经典世界” C；该策略会依据导出序，将子目标降至用户指定的、在污点格中低于当前子目标所在位置的世界；若用户指定的世界不满足这一条件，则操作失败。

[9] 我们的代码采用开源形式，可在此处获取 https://github.com/martinberger/hol-c。

6 结论

数学家在检查证明时，常会临时在不同的基础系统之间进行上下文切换；证明或其他数学文本中附带的非正式叙述性说明，会明确提示何时进行这些上下文切换。然而，现有的证明检查软件对不同推理模式之间的切换支持相对薄弱，导致某些数学材料的形式化工作十分繁琐。

在某些 HOL 实现中，例如 HOL4，定理对象会被加上标签，以追踪公理使用情况——例如，参见 HOL4 内核实现中对 `Tag.merge` function 的使用。这种污点不会作为逻辑的一部分暴露出来，仅仅是一项实现细节。我们的创新本质上是将这一想法贯彻到底，把污点纳入逻辑本身的构造中，从而将逻辑“划分”为若干彼此并列、表达力各异的子逻辑。

虽然从某种意义上说，我们的污染系统是一种省力工具，无需在证明中传递公理，但有趣的是，它看起来也很像一种针对 HOL 证明的静态分析或简化版类型系统。实际上，我们逻辑的各个子系统之间的结果流转，是通过一条新的推理规则来实现的；这条规则令人强烈联想到子类型规则或包含规则，参见 [33, 第 15.1 节“包含”]。从表面上看，我们的污点也多少让人联想到某种模态性 [8], ，不过我们提醒读者不要这样理解。我们逻辑的公式语言——如 §2所述——与标准 HOL 的公式语言完全相同。因此， $\phi:\ \ell$ 在不是我们系统中任何 $\ell \in \mathcal{L}$的公式。

我们的系统类似于用于高阶逻辑的标记演绎系统，仿照Gabbay [17]。据我们所知，尚不存在用于高阶逻辑的类似标记方法，尽管已有大量工作将标记应用于相对常见的逻辑，例如经典逻辑和直觉主义一阶逻辑 [18]，，以及更为特殊的逻辑，例如次结构逻辑 [10,13], 、模态逻辑 [5,26], 和时态逻辑 [22]。这一资源可以为未来的工作提供灵感：具体聚焦于次结构逻辑，标记此前已被用于为一系列次结构逻辑构造类似的参数化系统，其标签取自某个特定代数，并且可以像我们的格一样对该代数加以改变，从而获得不同的次结构逻辑。[10]

我们推测，我们的污点思想也可以以类似方式进一步“切分”高阶逻辑，以研究高阶逻辑的次结构片段。这里我们聚焦于添加或移除公理，然而，次结构逻辑限制了自然演绎关系中不同结构规则 的使用。在我们的系统中，由于我们使用集合表示上下文，这些规则大多是隐含的，只有（**Nwk**）除外。不过，也可以将上下文建模为假设列表，并引入显式规则来管理和操作假设，从而使这些规则显现出来。结构规则随后也可以引入污点，例如：

$$\frac{\Gamma \vdash \psi : \ell \quad \vdash \phi : \mathtt{Prop}}{\phi :: \Gamma \vdash \psi : \ell \sqcup Wk}$$

（这里， $-::-$ 是列表构造运算符，用于将一个假设添加到上下文 Γ的开头； Wk 是与弱化结构规则相关联的专用污点标签。）

这可能是比我们在论文正文中所引入的变化影响更为深远的一项改变。限制结构规则的使用，可能会使标准逻辑常量和联结词分化为不同的联结词子族——例如，线性逻辑就将标准联结词分解为乘法族和加法族。

[10]在某种程度上相关的研究中，定量类型理论 [2] 及其前身系统 [29] 也使用一种代数——具体来说，是一种资源半环，以便将线性逻辑与依赖类型理论平滑地整合起来。

参考文献 1 Andrea Asperti、Wilmer Ricciotti、Claudio Sacerdoti Coen 和 Enrico Tassi。Matita 交互式定理证明器。载于 Nikolaj S. Bjørner 和 Viorica Sofronie-Stokkermans 编，《自动演绎——第23届国际自动演绎会议（CADE-23）》，波兰弗罗茨瓦夫，2011年7月31日—8月5日。论文集，《计算机科学讲义》第6803卷，第64–69页。Springer，2011年。doi:10.1007/978-3-642-22438-6_7。2 Robert Atkey。《定量类型论的语法与语义》。载于第33届 ACM/IEEE计算机科学逻辑研讨会论文集，LICS '18，第56–65页，美国纽约州纽约，2018年。美国计算机协会。 doi:10.1145/3209108.3209189。3 Jeremy Avigad 和 Saul Feferman。哥德尔函数式（"Dialectica"）解释。载于 Samuel Buss 编，《证明论手册》，第337–405页。爱思唯尔科学出版公司，BV，1998年。4 Jeremy Avigad 和 Richard Zach。《ε演算》。载于 Edward N. Zalta 编，《斯坦福哲学百科全书》。斯坦福大学形而上学研究实验室，2020年秋季版，2020年。5 David A. Basin、Seán Matthews 和 Luca Viganò。带标签命题模态逻辑：理论与实践。《逻辑与计算杂志》，7(6):685–717，1997年。 doi:10.1093/logcom/7.6.685。6 John L. Bell。《无穷小分析入门》。剑桥大学出版社，第2版，2008年。doi:10.1017/CBO9780511619625。7 Yves Bertot 和 Pierre Castéran。交互式定理证明与程序开发：Coq'Art：归纳构造演算。施普林格科学与商业媒体，2013年。8 Patrick Blackburn、J. F. A. K. van Benthem 和 Frank Wolter 编。《模态逻辑手册》，《逻辑与实践推理研究》第3卷。北荷兰出版公司，2007年。网址： https://www.sciencedirect.com/bookseries/studies-in-logic-and-practical-reasoning/vol/3/suppl/C。9 Douglas Bridges、Erik Palmgren 和 Hajime Ishihara。《构造数学》。载于 Edward N. Zalta 和 Uri Nodelman 编，《斯坦福哲学百科全书》。斯坦福大学形而上学研究实验室，2022年秋季版，2022年。10 Krysia Broda、Marcelo Finger 和 Alessandra Russo。次结构逻辑的带标签自然演绎。《IGPL逻辑杂志》，7(3):283–318，1999年。 doi:10.1093/jigpal/7.3.283。11 Laura Crosilla。《集合论：构造性与直觉主义ZF》。载于 Edward N. Zalta 编，《斯坦福哲学百科全书》。斯坦福大学形而上学研究实验室，2020年夏季版，2020年。12 Juan Cuadra 和 Ehud Meir。群代数扭曲中的整Hopf序的存在性，2022年。网址： https://arxiv.org/abs/2211.00097, doi:10.48550/ARXIV.2211.00097。13 Marcello D'Agostino 和 Dov M. Gabbay。通过带标签演绎系统推广分析演绎。第一部分：基本次结构逻辑。《自动推理杂志》，13(2):243–281，1994年。doi:10.1007/BF00881958。14 Radu Diaconescu。选择公理与补集运算。《美国数学会会刊》，51:176–178，1975年。15 Damir D. Dzhafarov 和 Carl Mummert。《逆向数学：问题、归约与证明》。施普林格·自然，2022年。16 Gilda Ferreira 和 Paulo Oliva。论各种负翻译。《理论计算机科学电子论文集》，47:21–33，2011年1月。网址： https://arxiv.org/abs/1101.5442, doi:10.4204/eptcs.47.4。17 Dov M. Gabbay。《带标签演绎系统》，第一卷。克拉伦登出版社，1996年。18 Dov M. Gabbay 和 Uwe Reyle。经典与非经典逻辑的带标签归结。Stud Logica，59(1):179–216，1997年。 doi:10.1023/A:1004952317276。19 库尔特·哥德尔。关于有限观点的一种迄今未曾使用的扩展。《Dialectica》，第280–287页，1958年。20 N. Goodman 和 J. Myhill。选择蕴含排中律。《数学逻辑季刊》，24(25-30):461–461，1978年。网址： https://onlinelibrary.wiley.com/doi/abs/10.1002/malq.19780242514, arXiv:https://onlinelibrary.wiley.com/

doi/pdf/10.1002/malq.19780242514, doi:https://doi.org/10.1002/malq.19780242514. 21 John Harrison. HOL Light：概述。载于 Stefan Berghofer、Tobias Nipkow、Christian Urban 和 Makarius Wenzel（编），《高阶逻辑定理证明》，第22届国际会议，TPHOLs 2009，2009年8月17日至20日，德国慕尼黑。论文集，《计算机科学讲义》第5674卷，第60–66页。Springer，2009年。 doi:10.1007/978-3-642-03359-9_4。22 Andrzej Indrzejczak。线性时态逻辑的带标签自然演绎系统。Stud Logica，75(3)：345–376，2003年。 doi:10.1023/B:STUD.0000009565.98020.9b。23 V. A. Jankov。《弱“排中律”的演算》。《苏联数学—消息报》，2(5)：997，1968年。URL： https://dx.doi.org/10.1070/IM1968v002n05ABEH000690, doi:10.1070/IM1968v002n05ABEH000690。24 Thomas J. Jech。《选择公理》。Dover Publications, Inc.，2008年。25 Asaf Karagila。佐恩式泛函分析，或：我如何学会不再担心并爱上选择公理，2020年。URL: https://arxiv.org/abs/2010.15632, doi:10.48550/ARXIV.2010.15632。26 Sonia Marin、Marianela Morales 和 Lutz Straßburger。直觉主义模态逻辑的完全带标签证明系统。《逻辑与计算杂志》，31(3)：998–1022，2021年。 doi:10.1093/logcom/exab020。27 A. P. Martin、P. H. B. Gardiner 和 J. C. P. Woodcock。《策略演算——删节版》。《形式化方面的计算》，8(4)：479–489，1996年7月。 doi:10.1007/BF01213535。28 Andrew Martin。《软件工程中的机器辅助定理证明》。博士论文，牛津大学计算实验室，1996年。技术专著 PRG-121，可从 https://www.cs.ox.ac.uk/files/3449/PRG121.pdf获取。29 Conor McBride。《我有的是好东西》，第207–233页。Springer International Publishing，2016年。 doi:10.1007/978-3-319-30936-1_12。30 Robin Milner。机器在辅助严格证明中的应用。伦敦皇家学会哲学汇刊 A辑，312：411–422，1984年。31 Ulf Norell。Agda中的依赖类型编程。载于 Andrew Kennedy 和 Amal Ahmed（编），TLDI'09论文集：2009年 ACM SIGPLAN 类型在语言设计与实现中的应用国际研讨会，2009年1月24日，美国佐治亚州萨凡纳，第1–2页。ACM，2009年。 doi:10.1145/1481861.1481862。32 Russell O'Connor。构造性世界中的经典数学。计算机科学中的数学结构，21(4)：861–882，2011年7月。URL: https://arxiv.org/abs/1008.1213,doi:10.1017/s0960129511000132。33 Benjamin C. Pierce。《类型与程序设计语言》。MIT出版社，2002年。34 David A. Schmidt。《策略推理的程序设计表示法》。载于 R. E. Shostak（编），第7届国际自动演绎会议，第445–459页，美国纽约州纽约市，1984年。Springer New York。35 Konrad Slind 和 Michael Norrish。HOL4概述。载于 Otmane Aït Mohamed、César A. Muñoz 和 Sofiène Tahar（编），《高阶逻辑定理证明》，第21届国际会议，TPHOLs 2008，2008年8月18日至21日，加拿大蒙特利尔。论文集，《计算机科学讲义》第5170卷，第28–32页。Springer，2008年。 doi:10.1007/978-3-540-71067-7_6。36 Makarius Wenzel、Lawrence C. Paulson 和 Tobias Nipkow。Isabelle框架。载于 Otmane Aït Mohamed、César A. Muñoz 和 Sofiène Tahar（编），《高阶逻辑定理证明》，第21届国际会议，TPHOLs 2008，2008年8月18日至21日，加拿大蒙特利尔。论文集，《计算机科学讲义》第5170卷，第33–38页。Springer，2008年。 doi:10.1007/978-3-540-71067-7_7。