arXiv:2304.02596v1 [math.LO] 5 Apr 2023

奠基算子：传递性与树、逻辑性与平衡

Francesco A. Genco

通讯作者。电子邮箱：frgenco@gmail.com;

摘要

我们从推理视角对直接奠基算子和间接奠基算子进行形式化研究。我们讨论若干奠基算子所表现出的行为差异，并考察奠基算子与逻辑算子之间的一般区分。在不预先确定某种特定的奠基概念或奠基关系的情况下，我们提出了一组推理规则；一旦基础奠基演算确定，这些规则便定义出三种奠基算子：直接奠基算子、间接奠基算子（对应于直接奠基算子的传递闭包），以及奠基树算子。后者使我们能够将直接奠基断言链内化，同时不丢失关于这些断言的任何信息。随后，我们对所引入的规则进行了深入的证明论研究，尤其关注以下问题：奠基算子是否可以被视为逻辑算子，以及是否能够为奠基算子定义平衡规则。

关键词：奠基、传递性、规范化、逻辑性、超内涵性。

MSC 分类：03A05，03F05

1 引言

奠基通常被视为一种客观的解释性关系，它连接两个关系项——基础与结果——前者决定或解释后者。在当代哲学文献中，人们投入了大量精力，借助逻辑系统分析奠基的形式方面，参见例如 [1–9], ；这些分析往往依赖于通过推理演算刻画奠基的方式，参见例如

[1, 2, 4–7]。在大多数演算系统中，奠基是通过作用于公式的算子来形式化的。尽管许多研究致力于分析特定的奠基概念，并研究特定的奠基算子，但尚无系统性研究探讨奠基算子共同具有的一般形式特征。在本文中，我们尝试从推理视角首次考察不同奠基算子的形式行为，尤其关注奠基算子的一般性质，以及直接奠基与间接奠基的不同形式化之间的关系。在不固定某种特定奠基概念或奠基关系的前提下，我们研究一种通用直接奠基算子的证明论特征，并从证明论角度考察将其推广为间接奠基算子的两种不同方式。

为此，我们引入三组推理规则；在奠基演算已确定的前提下，这些规则定义三个奠基算子的行为：一个用于直接奠基的算子，一个用于间接奠基的算子——它对应于直接奠基的传递闭包——以及一个奠基树算子，后者使我们能够将直接奠基断言链内化，同时不丢失有关这些断言及其关系的任何信息。直观地说，当一个基础以解释性方式直接关联到一个结果时，直接奠基便将基础与结果联系起来。根据所考虑的具体奠基概念，这种奠基关系的直接性可以有不同的表述方式。例如，它可能取决于基础相对于所采用的复杂度度量而言比结果简单一级这一事实，也可能取决于按照既定层级，基础恰好比结果更基础一个层级这一事实。例如，按照大多数逻辑奠基概念，A 和 B合在一起被认为构成 $A \wedge B$的直接基础。例如，[1, 2, 4, 6, 7]讨论了不同的直接奠基概念。另一方面，当基础与结果由多个直接奠基步骤构成的链相连接时，间接奠基便将它们联系起来。按照某些奠基概念，如果我们说 A、 C 和 D 构成$(A \vee B) \wedge (C \wedge D)$的基础，那么这就是一个简单的逻辑间接奠基实例。事实上，如果 A 是 $A \vee B$的直接基础，而 C 和 D 合在一起构成$C \wedge D$的直接基础，那么我们便可得出结论： A、 C 和 D 合在一起构成$(A \vee B) \wedge (C \wedge D)$的间接基础。直观地说，我们拥有如下树形图所示的奠基关系：

$$\begin{array}{ccc} A & C & D \\ | & | & | \\ A \vee B & C \wedge D & \\ | & | & \\ (A \vee B) \wedge (C \wedge D) & & \end{array}$$

虽然直接奠基说明了公式与紧接在其上方的公式之间的直接联系，但上文讨论的特定间接奠基状态说明了树的根与其叶之间的解释关系。例如，[1, 2, 5, 8, 10]还讨论了不同的间接奠基概念。最后，奠基树旨在将树形图所表达的全部信息编码于一个独特的句子对象中，例如上文所示的树形图；另请参见 [11, § 220]，其中给出了奠基树的类似图示表示。间接奠基使我们能够表达一个陈述与以解释方式同其关联的任意一组陈述之间的间接联系；与此不同，奠基树使我们能够表达从一组句子通向我们希望解释的句子的完整解释步骤链。

严格来说，直接奠基将通过 ▶ 算子形式化，该算子只能在应用直接奠基规则之后立即引入。此规则应用将确保所考察的基础——对应于奠基规则的前提——与所考察的结果——对应于奠基规则的结论——之间存在直接奠基关系。间接奠基将通过 ≫ 算子形式化，它在对象语言中内化了直接奠基算子 ▶的传递闭包。正如我们将展示的，为了刻画间接奠基算子， ≫恰好使我们能够选出奠基推导[1] 中位于某条横线上的所有且仅有的公式，并在针对该奠基推导结论的间接奠基陈述中使用它们。最后，奠基树将通过在一个 ▶ 算子的出现中嵌套若干个 ▷ 算子的出现来形式化。因此，我们将能够构造出与通过嵌套若干个连续的直接奠基断言而建立的奠基推导完全对应的公式。

我们将为所有奠基算子采用的规则完全模块化，且不依赖于对背景奠基演算的特定选择。因此，本文并不依赖于所考察奠基关系的特定特征，适用于文献中已正式提出的若干种奠基关系。具体而言，对于任何能够通过形如 $\frac{A_1 \quad \dots \quad A_n}{B}$ 的奠基规则形式化的奠基概念——其中 A_1，...， A_n 被认为构成 B的基础——我们都可以定义一种奠基演算，并通过加入我们的规则对其加以扩展，从而定义一种奠基算子的行为，使其精确刻画所考察的奠基概念。能够以这种方式形式化的奠基概念包括[1, 2, 4, 5], 中定义的完全奠基和部分奠基，以及[6, 7]中定义的完备奠基。

在介绍并刻画了三种基础运算符的规则之后，我们将深入开展证明论研究，重点关注

[1]正如我们稍后将要解释的，如果将奠基推导视为其结论的渐进分解，那么奠基推导的一条横线可以看作对这一分解某一阶段的完整描述。

关于这些规则是否可被视为相应算子的良好定义，以及通过研究所给出的推理规则我们能从算子本身学到什么。为此，我们将首先考虑奠基算子是否可被视为逻辑算子，然后——在对此问题给出否定回答之后——尝试确定所给出的奠基算子规则是否仍显示出一种平衡形式，使我们能够得出它们适当地定义了这些算子的结论。为此，我们将采用来自结构主义证明论方法对逻辑常项概念刻画的方法——例如见 [12, 13]——这可追溯到 Koslow [14] 和 Popper 的工作，见 [15]。我们将特别考虑两个传统的逻辑性标准，并表明虽然其中一个被奠基算子的规则所满足，但另一个则不满足，除非考虑其较弱版本并且关于底层直接奠基关系的某些假设成立。

我们将考察的第一个判准对应于序列演算规则的“同一者的可推导性” [16] ，并以直接扩展之名出现在 [17]的自然演绎规则中；关于这一判准以及 [19]中讨论的一个类似判准的研究，参见 [18]。特别是，我们将表明，该判准的严格版本——最初用于刻画逻辑算子的版本——并不为所考察的任何奠基算子规则所满足。在讨论这一失败的概念意义，以及它与奠基关系某些重要特征之间的若干有趣联系之后，我们论证说，尽管这一失败意味着我们的奠基算子不符合对逻辑算子的标准理解，但它并不意味着无法对这些算子进行有意义的分析，以理解其规则是否恰当地将它们定义为非逻辑语句算子。事实上，通过找到一种适当的方式来放宽推理语义文献中通常采用的判准，确实可以开展这样的分析。

第二个所考察的判准，我们称之为迂回消除性[2]，它要求：通过演绎地使用由应用该算子构造的句子，不能获得超出得出该句子为真的结论所需的信息。这恰好对应于这样一种可能性：从任何推导中消除任何直接涉及所考察算子的迂回——也就是说，先引入该算子的推理步骤紧接着被消除。迂回消除性不仅是算子逻辑性若干判准的关键，也是规范化结果的一项基本要求。此外，它可以与 [20] 中提出的证明论和谐概念相等同，并且是大多数其他和谐概念的核心组成部分——或者至少是检验这些概念的有效手段；关于这一问题的文献综述，参见例如 [19, 21–23] 以及 [24] 。如果某个算子的规则同时具有迂回消除性和可推导性的

[2]我们使用这个名称，是为了使术语保持明确，并避免使用已经承载过多含义的词语。

此外，如果它们相同，那么就可以将其视为对该算子含义的精确定义，因为演绎地使用该算子的规则与判定通过应用该算子构造的句子何时为真的规则之间达到了完美平衡。正如我们将要展示的，直接奠基算子和奠基树算子的规则 ▶ 和 ▷允许定义普拉维茨式迂回化简，可用来推广 [25]中的规范化结果。另一方面，尽管也可以为间接奠基算子定义绕行约化 ≫——这意味着 ≫ 同样满足局部形式的迂回消除性——但若要论证通过绕行约化扩展的、针对 ≫ 的正规化过程会终止，就会遇到问题。我们将论证，这些问题严重妨碍了证明包含 ≫ 规则的演算系统具有适当的——或称全局的——规范化结果所必需的标准论证。

最后，鉴于 ▶ 和 ▷ 在规范化以及由此产生的迂回消除性方面显然表现良好，我们提出并分析了一个较弱版本的同一者可推导性标准；与传统标准相比，该版本更适合奠基的超强内涵性质。[3] 在此过程中，我们旨在表明，尽管奠基算子无法通过逻辑性测试，但在某些情况下，可以找到能够恰当地定义其含义的平衡规则集。较弱版本的同一者可推导性还构成了一个不会使对非逻辑算子的分析变得琐碎的判准，从而使我们能够研究不同奠基算子之间的差异。事实上，我们将表明，虽然直接奠基算子和奠基树算子 ▶ 和 ▷ 在某些假设下满足这一较弱判准，但间接奠基算子 ≫并不满足。最后，我们从概念视角讨论前两个算子与第三个算子在行为上的差异，强调这些算子的技术特征与其预期解释之间的平行关系。

文章的其余部分结构如下。在第2节中，我们引入将要使用的语言并讨论所引入的奠基算子的含义。在第3节中，我们给出奠基算子的引入与消去规则。特别地，在第3.1节中，我们给出间接奠基算子的那些规则，在第3.2节中给出奠基树算子的那些规则。在第4 节中，我们研究所引入的奠基算子规则的证明论平衡：第4.1 节将致力于迂回消除性，第4.2 节致力于同一者的可推导性。在本节中，我们还讨论奠基算子与所考虑的证明论准则之间关系的概念原因和含义。最后，在第5节中，我们给出一些结论性评论，并讨论进一步发展所呈现分析的可能途径。

[3] 如果使用某个算子构成的句子的真并非必然.....，那么该算子就是超内涵的。 ly pres在将其某些论元替换为逻辑等价的论元后仍然保持不变。

2 语言

我们首先介绍将采用的逻辑语言，其中包括通常的逻辑联结词、用于直接奠基的运算符 ▶ 、用于间接奠基的运算符 $\gg$ ，以及用于构造奠基树的运算符 $\triangleright$ 。

定义 1（语言的公式）

$$\begin{aligned}
\varphi \quad &::= \quad \xi \mid \bot \mid \neg\varphi \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \varphi \to \varphi \mid \\
&\qquad (\Psi \blacktriangleright \varphi) \mid (\Psi[\Psi] \blacktriangleright \varphi) \mid \\
&\qquad (\Phi \gg \varphi) \mid (\Phi[\Phi] \gg \varphi) \mid \\
\psi \quad &::= \quad \varphi \mid (\Psi) \triangleright \varphi \mid (\Psi[\Psi]) \triangleright \varphi \\
\xi \quad &::= \quad p \mid q \mid r \mid \ldots
\end{aligned}$$

其中 Φ 是形如 φ、.....、φ 的列表， Ψ 是形如 ψ、.....、ψ的列表，而 p、q、r、..... 都是该语言中的命题变元。

定义1 中的语法使我们能够通过使用非终结符 φ and ξ 构造经典逻辑语言的任何公式。另一方面，即刻奠基陈述可以通过使用 (Ψ ▶ φ)和 ($\Psi[\Psi]$ ▶ φ) 构造，然后将 Ψ and φ扩展为经典逻辑语言中的公式。例如，如果我们用一个或多个非终结符的扩展来表示，我们可以得到以下实例化序列： $\varphi \rightsquigarrow (\psi, \psi[\psi] \blacktriangleright \varphi)$ $(\varphi, \varphi[\varphi] \blacktriangleright \varphi)$ $(p, q[r] \blacktriangleright \varphi \vee \varphi)$ $(p, q[r] \blacktriangleright \varphi \vee (\varphi \wedge \varphi))$ $(p, q[r] \blacktriangleright r \vee (p \wedge q))$。然而，通过使用非终结符 ψ，我们还可以将 $\triangleright$ 的一次出现嵌套在 ▶ 的一次出现的左侧，从而构造一棵奠基树。一旦我们构造出形如$(\Psi) \triangleright \varphi$ 或$(\Psi[\Psi]) \triangleright \varphi$ 的子公式，就可以选择继续使用非终结符号列表 Ψ ψ ，将另一个算子 $\triangleright$ 嵌套到前一个算子的左侧，以扩展奠基树；也可以停止嵌套 $\triangleright$ ，并将列表 Ψ 中的一个非终结符号 ψ 扩展为 φ。后一种可能性的一个例子是以下实例化序列：$\varphi \rightsquigarrow (\psi \blacktriangleright \varphi)$ $((\psi) \triangleright \varphi \blacktriangleright \varphi)$$((\varphi) \triangleright p \blacktriangleright q)$$((r) \triangleright p \blacktriangleright q)$。请注意，以 $\triangleright$ 为最外层算子的子公式，其括号方式与以 ▶ 或 $\gg$ 为最外层算子的子公式不同。我们将在下面的例子中解释其中的原因。通过$(\Phi \gg \varphi)$ 和$(\Phi[\Phi] \gg \varphi)$构造间接奠基陈述的方式与构造即刻奠基陈述的方式相同，只是非终结符号列表 Φ φ 不允许我们将 $\triangleright$ 嵌套在 $\gg$ 的左侧。

在下文中，我们将使用大写拉丁字母作为公式的元变量，使用大写希腊字母作为公式多重集的元变量，并且在不产生歧义时省略括号。此外，略微滥用记号，当我们写出形如 $\Gamma[\Delta] \blacktriangleright A, \Gamma[\Delta] \gg A$and $(\Gamma[\Delta]) \triangleright A$ 的公式时，我们将承认它们可能表示如下形式的公式

分别形如 $\Gamma \blacktriangleright A$, $\Gamma \gg A$ and $(\Gamma) \triangleright A$ 的公式。我们这样做只是简单地承认 Δ 可能是空列表。

现在让我们讨论如何理解使用我们的语法构造出的奠基公式。形如 $\Gamma[\Delta] \blacktriangleright A$ 的公式（其中 Γ 和 Δ 均不包含形如 $(\Theta[\Sigma]) \triangleright B$ 的公式）对应一个直接奠基断言，表示在条件 Δ 下，Γ 构成结果 A 的直接根据。形如 $\Gamma[\Delta] \gg A$ 的公式对应一个间接奠基断言，表示在条件 Δ 下，Γ 构成结果 A 的间接根据。例如，假设

$$p, q \blacktriangleright p \wedge q \qquad p \wedge q \blacktriangleright \neg\neg(p \wedge q)$$

在我们的系统中是合法的奠基断言，那么我们有：

$$p, q \gg \neg\neg(p \wedge q)$$

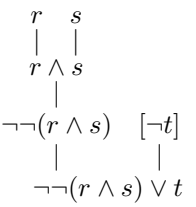
也是一个合法的奠基断言。形如 $\Gamma[\Delta] \blacktriangleright A$ 的公式（其中 Γ 或 Δ 或二者都包含形如 $(\Theta[\Sigma]) \triangleright B$ 的公式）对应一棵奠基树。这种奠基树通过有序展示我们可以从 A 构造出的奠基实例，逐步到达越来越简单的根据，从而对 A 为真的情况作出复杂说明。直观地说，奠基树可以看作一个间接奠基断言，其中还包含了被奠基陈述与其间接根据之间关系的信息。换言之，在奠基树中，我们记录了所有能够证明间接奠基断言的直接奠基步骤。例如，假设

$$r, s \blacktriangleright r \wedge s \qquad r \wedge s \blacktriangleright \neg\neg(r \wedge s) \qquad \neg\neg(r \wedge s)[\neg t] \blacktriangleright \neg\neg(r \wedge s) \vee t$$

是我们系统中合法的直接奠基断言，那么我们得到

$$((r, s) \triangleright r \wedge s) \triangleright \neg\neg(r \wedge s)[\neg t] \blacktriangleright \neg\neg(r \wedge s) \vee t$$

也是一个合理的奠基断言。这个奠基树直观上对应于下面这棵树，其中每条边都表示由一个直接奠基关系实例所建立的联系：



请注意，形如$(\Gamma) \triangleright A$ 的表达式中 Γ 周围的括号旨在强调只有 A——而非整个表达式$(\Gamma) \triangleright A$——是一个部分

的直接根据，其中表达式 $(\Gamma) \triangleright A$ 出现。例如，通过写入 $(r, s) \triangleright r \wedge s$ 在公式 $((r, s) \triangleright r \wedge s) \triangleright \neg\neg(r \wedge s)[\neg t] \blacktriangleright \neg\neg(r \wedge s) \vee t$ 上述，我们强调只有 $r \wedge s$ 是 $\neg\neg(r \wedge s)$ 的直接根据。类似地，通过写入 $((r, s) \triangleright r \wedge s) \triangleright \neg\neg(r \wedge s)$ 我们强调，在此表达式中出现的公式中，只有 $\neg\neg(r \wedge s)$ 是 $\neg\neg(r \wedge s) \vee t$ 的直接根据的一部分。

再举一个奠基树的例子，假设

$$r[\neg s] \blacktriangleright r \vee s \qquad r \vee s \blacktriangleright \neg\neg(r \vee s)$$

在我们的系统中是合法的直接奠基断言，那么我们有

$$(r[\neg s]) \triangleright r \vee s \blacktriangleright \neg\neg(r \vee s)$$

也是一个合法的奠基断言。这棵奠基树对应于以下直接奠基实例树：

$$\begin{array}{c} r \quad [\neg s] \\ | \quad\quad | \\ r \vee s \\ | \\ \neg\neg(r \vee s) \end{array}$$

3 奠基算子的规则

Th该算子的引入规则 ▶ 在表 1中给出。此规则 le

If

- $\dfrac{A_1 \quad \ldots \quad A_n \quad [C_1 \quad \ldots \quad C_m]}{B}$ 是一个奠基规则应用，使得 A_1，...，A_n 构成 B 在可能为空的条件列表 C_1，...，C_m 下的基础
- δ_1，...，δ_n，δ'_1，...，δ'_m 分别是 A_1，...，A_n，C_1，...，C_m 的推导

then

$$\dfrac{\dfrac{\begin{array}{cccccc}\delta_1 & & \delta_n & \delta'_1 & & \delta'_m \\ A_1 & \cdots\cdots & A_n & [C_1 & \ldots & C_m]\end{array}}{B}}{A_1, \ldots, A_n[C_1, \ldots, C_m] \blacktriangleright B}$$ 是一个推导

表 1 直接奠基算子的引入规则 ▶

反映了这样一种观点：以 ▶ 作为最外层算子且不含 ▷的嵌套出现——也就是说，一个直接奠基断言——只能基于合法的基础规则应用而被引入——为了在视觉上立即区分，我们在表示基础规则应用时使用双重推理线。从技术上讲，我们只能引入 ▶

紧接在基础规则应用下方。例如，如果我们考虑奠基演算 [26]，以下是合法的基础规则应用：[4]

$$\frac{p \quad q}{p \wedge q} \qquad \frac{p \quad [\neg q]}{p \vee q}$$

因此，利用我们关于奠基算子的规则，我们可以如下引入 ▶：

$$\frac{\dfrac{p \quad q}{p \wedge q}}{p, q \blacktriangleright p \wedge q} \qquad \frac{\dfrac{p \quad [\neg q]}{p \vee q}}{p[\neg q] \blacktriangleright p \vee q}$$

因此我们得出奠基断言 $p, q \blacktriangleright p \wedge q$，依据的假设是$p$ 和 q 为真，并且我们得出奠基断言 $p[\neg q] \blacktriangleright p \vee q$，依据的假设是 p 和 $\neg q$ 为真。[5]

的消去规则列于表2中。该表中的前三条规则对应于奠基的所谓事实性；也就是说，根据奠基的这一特征，如果连接各元素及其相应结果的奠基断言为真，那么基础的所有元素及相应结果都应当为真。表2 中的最后一条规则使我们能够推导出那些无法通过所选奠基演算中的奠基规则推导出的奠基断言的否定。如果我们排除存在无法在该演算中推导出但却为真的奠基断言这一可能性，这条规则就使我们能够对某些奠基断言的虚假性进行推理。然而，某些奠基演算被认为仅仅提供了一个关于奠基的最小框架，该框架可以通过关于真实奠基断言的进一步假设或进一步的奠基规则加以扩展——例如参见 [2]。如果情况确实如此，那么可以省略表2 中的最后一条规则，从而获得一种更灵活的演算，该演算允许进行此类扩展。

3.1 间接奠基算子

在表3中，我们介绍间接奠基算子的规则≫，其基础是直接奠基算子 ▶。

≫ 的引入规则实现了对 ▶的传递闭包的显然归纳定义。具体而言，表格中的第一条规则对应于基本情形：任何直接奠基断言 $\Gamma[\Delta] \blacktriangleright A$同时也是间接奠基断言 $\Gamma[\Delta] \gg A$。另一方面，表格中的第二条规则使我们能够通过传递性组合两个间接奠基断言。具体而言，如果 $\Gamma[\Delta]$ 构成了 A 的间接根据，而 A 是

[4]我们稍微调整一下记号，用方括号代替析取规则各前提之间的竖线。[5]更一般地说，▶ 引入规则的结论可以采用如下形式：$h(A_1) \ldots, h(A_n)[h(C_1) \ldots, h(C_m)] \blacktriangleright B$ 其中，h 是一个从公式到公式的函数，具体取决于所考虑的奠基规则所定义于的特定系统，以及其应用中各前提的推导。这是因为在某些证明系统中，奠基规则应用的前提并不总能直接解释为其结论的基础。由于这一细节与本文无关，且针对特定证明系统时很容易处理，我们在规则定义中省略函数 h。

$$\frac{\Gamma[\Delta] \blacktriangleright B}{B} \qquad \frac{\Gamma_1, A, \Gamma_2[\Delta] \blacktriangleright B}{A} \qquad \frac{\Gamma[\Delta_1, C, \Delta_2] \blacktriangleright B}{C}$$

其中 A 和 C 的最外层算子不是 $\triangleright$

如果不存在基础规则应用 $\dfrac{A_1 \quad \ldots \quad A_n \quad [C_1 \quad \ldots \quad C_m]}{B}$ then $\dfrac{A_1, \ldots, A_n [C_1, \ldots, C_m] \blacktriangleright B}{\bot}$ 是一个规则应用

表 2 直接奠基算子的消去规则 $\blacktriangleright$

$$\frac{\Gamma[\Delta] \blacktriangleright A}{\Gamma[\Delta] \gg A} \qquad \frac{\Gamma[\Delta] \gg A \qquad \Gamma_1, A, \Gamma_2[\Delta_1] \gg B}{\Gamma_1, \Gamma, \Gamma_2[\Delta, \Delta_1] \gg B}$$

$$\frac{\Gamma[\Delta] \gg A \qquad \Gamma_1[\Delta_1, A, \Delta_2] \gg B}{\Gamma_1[\Delta_1, \Gamma, \Delta, \Delta_2] \gg B}$$

表 3 间接奠基算子的引入规则 $\gg$

包含在 B的间接奠基中，那么我们可以在 B 的基础中插入 Γ 而不是 A，并将 Δ 添加到 B的基础的条件中。

在表 4，我们给出消去规则对于 $\gg$。这些规则用于消 i-

$$\frac{\Gamma[\Delta] \gg B}{B} \qquad \frac{\Gamma_1, A, \Gamma_2[\Delta] \gg B}{A} \qquad \frac{\Gamma[\Delta_1, C, \Delta_2] \gg B}{C}$$

表 4 间接奠基算子的消去规则 $\gg$

nate $\gg$ 简单实现间接奠基的事实性。事实上，给定一个间接奠基断言，它们使我们能够推出结果、基础的任意部分以及条件的任意部分。

3.1.1 间接奠基规则的完备性

我们现在表明，间接奠基规则总是使我们能够内化那些根据所选奠基演算应当成立的间接奠基断言。我们特别表明，如果能够在我们的奠基演算中构造出公式 A 的奠基推导，那么我们就能够推导出一个断言，表达曾被用于构造 A 的奠基推导的某些公式集合是 A的间接根据。我们仅

讨论间接奠基规则的完备性，而不是刻画，因为逆向结果——即，只有当存在适当的奠基推导时，某个间接奠基断言才是可推导的——无法在此证明。事实上，后一个结果本质上取决于所考察的奠基演算的具体特征。[6] 为了区分构成某个公式的适当间接奠基的公式集合 A，我们采用奠基推导横线这一概念。直观地说，如果我们将一个奠基推导视为其结论的渐进分解，那么奠基推导的横线就可以看作对这一分解某个阶段的完整描述。例如，考虑以下推导，并假设根据某一给定的奠基概念，它构成一个合法的奠基推导：

$$\dfrac{\dfrac{p \quad \dfrac{q \quad r}{q \vee r}}{p \wedge (q \vee r)} \quad \dfrac{t \quad s}{s \wedge t}}{(p \wedge (q \vee r)) \vee (s \wedge t)}$$

该推导的其中一条横线恰好包含 p、q、r、t 和 s，也就是说，包含该推导的所有叶。这条横线表示分解的最终阶段：在这一阶段，我们实际上已经分解了通过逐步分解原公式$(p \wedge (q \vee r)) \vee (s \wedge t)$ 所获得的所有复杂子公式。但公式 p、$q \vee r$ 和 $s \wedge t$ 也构成了我们推导中的一条横线。这第二条横线表示分解所处的阶段：在这一阶段，我们已经将 $p \wedge (q \vee r) \vee (s \wedge t)$分解为 $p \wedge (q \vee r)$和 $s \wedge t$，随后又将 $p \wedge (q \vee r)$ 分解为 p 和 $q \vee r$，但尚未分解 $q \vee r$ 或 $s \wedge t$。更一般地说，横线表示这种分解过程中的一个阶段，因为它包含了我们迄今为止在分解过程中获得的所有公式出现，但不包含我们为获得这些公式出现而分解的任何公式出现。

我们先确定奠基推导的显然概念，然后形式化定义其横线是什么。

定义 2（奠基推导）奠基推导是这样一种推导：它仅通过对一组相容假设应用奠基规则构造而成，并且至少包含一次规则应用。

定义 3（推导的分支）给定任意由推理步骤组成的推导 δ，其中每个步骤都有一个或多个公式作为前提，并有一个公式作为结论，δ 的推导树 $t(\delta)$ 是一棵树，使得 $t(\delta)$ 的节点与 δ 中的公式出现之间存在一一对应关系，该对应关系满足以下条件：

[6]为了证明间接奠基断言只有在存在适当的奠基推导时才是可推导的，例如，可以证明所考虑的演算满足规范化结果，并且作为结果，该演算对于 $\gg$享有规范证明性质——也就是说，如果间接奠基断言是可证明的，那么在其规范证明中应用的最后一条规则是 $\gg$ 引入规则。不过，在所考虑的特定演算中，间接奠基断言的并非所有规范证明都是规范的。

- $t(\delta)$的根对应于 δ的结论，并且
- 如果节点 n of $t(\delta)$ 对应于公式出现 o of δ，且 o 是推理步骤 i的结论，那么 n 的每个子节点分别对应于 i 的一个不同前提，且 i 的每个前提分别对应于 n的一个不同子节点。

推导的分支 δ 是一组公式出现，这组公式出现位于 δ 中，不包含 δ 的结论，并且相应的节点构成一个集合，该集合与连接 t（δ）的根和其某个叶的每条路径（即连续节点的集合）恰好共享一个元素。

现在，我们已经可以证明，任意奠基推导中公式 A 的任意横线，都对应于一个可推导的关于 A 的间接奠基断言。

命题 **1** *If* Γ *contains all grounds and* Δ *all conditions occurring in a bar of a grounding derivation of* A *in a fixed calculus* κ*, then the grounding claim* $\Gamma[\Delta] \gg A$ *is derivable in any calculus that contains the rules of* κ*, the rules for* $\blacktriangleright$ *and the rules for* $\gg$*.*

Proof 证明通过对 A的奠基推导中出现的规则应用数量进行归纳法。在基本情形中，仅应用一条奠基规则来推导 A——根据奠基推导的定义，如果推导中未应用任何规则，则它不是奠基推导。因此，$\Gamma[\Delta] \blacktriangleright A$ 是直接可推导的，并且$\Gamma[\Delta] \gg A$ 可从它推导出来。此外，$\Gamma \cup \Delta$ 是该推导的唯一横线。现在假设，如果 Γ' 包含所有基础且 Δ' 包含少于 n 次基础规则应用的 A' 的奠基推导的一条横线节点上出现的所有条件，则 $\Gamma'[\Delta'] \gg A'$ 是可推导的。我们证明这对于包含 n 次基础规则应用的奠基推导也成立。假设存在包含 n次规则应用的 A的奠基推导 δ，并且 Γ 包含所有基础，Δ包含该推导的一条横线上出现的所有条件。然后考虑 δ中最底部的规则应用，我们称之为 r。然后我们可以以这样的方式列出横线 $\Gamma \cup \Delta$ 的元素 $B_1, \ldots, B_n$，使得 $B_1, \ldots, B_m$——对于 $m \leq n$——是 r的前提，并且 $B_{m+1}, \ldots, B_n$ 属于那些未在 $B_1, \ldots, B_m$中列出的 r的前提的不同奠基推导的横线。让我们称 $C_1, \ldots, C_p$为不属于列表 $B_1, \ldots, B_m$ 的 r 的前提。因此 —— 暂时忽略基础和条件之间的差异 —— 我们可以将我们的奠基推导描绘如下：

$$\dfrac{\begin{array}{cccccc} & & & \Theta_1 & & \Theta_p \\ \vdots & & \vdots & & & \\ B_1 & \ldots & B_m & C_1 & \ldots & C_p \end{array}}{A}\, r$$

其中每个 Θ_i 都是包含 B_{m+1}中的某些元素的 C_i 的推导的横线，....., B_n，而 $\Theta_1 \cup \ldots \cup \Theta_p$是多重集 $\{B_{m+1}, \ldots, B_n\}$。根据归纳假设，我们有每个奠基断言 $\Theta_i \gg C_i$——其中 Θ_i的某些元素可能位于方括号之间——都是可推导的。此外，奠基断言 B_1, ..., B_m, C_1, ..., $C_p \gg A$也都是如此——其中一些公式来自

$B_1,\ldots,B_m,C_1,\ldots,C_p$可能位于方括号之间——可通过一次 ▶引入紧接着一次 ≫引入，直接从奠基规则应用 r的结论中推导出来。因此，我们可以如下推导奠基断言 $\Gamma[\Delta] \gg A = B_1,\ldots,B_m,\Theta_1,\ldots,\Theta_p \gg A$：

$$\dfrac{\Theta_2 \gg C_2 \qquad \dfrac{\Theta_1 \gg C_1 \qquad \dfrac{\dfrac{\dfrac{}{B_1 \quad \ldots \quad B_m \quad C_1 \quad \ldots \quad C_p}\, r}{A}}{B_1,\ldots,B_m,C_1,\ldots,C_p \blacktriangleright A}}{B_1,\ldots,B_m,C_1,\ldots,C_p \gg A}}{B_1,\ldots,B_m,\Theta_1,\ldots,C_p \gg A} \qquad \Theta_1 \ldots \quad \Theta_p \ldots$$

$$\dfrac{\Theta_p \gg C_p \qquad B_1,\ldots,B_m,\Theta_1,\ldots,\Theta_{p-1},C_p \gg A}{B_1,\ldots,B_m,\Theta_1,\ldots,\Theta_p \gg A}$$

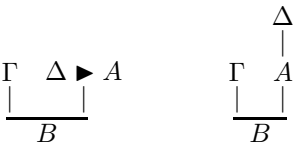
□

3.2 奠基树

现在我们给出构造奠基树的规则。如上文所述，奠基树表示若干直接奠基步骤的串联；我们利用嵌套在算子 ▶中的算子 ▷的出现，将奠基树编码为一个公式。例如，如果奠基主张 Γ、$A \blacktriangleright B$和 $\Delta \blacktriangleright A$成立，那么符号 ▷使我们能够将两个奠基主张组合为一个，具体如下：Γ、$(\Delta) \triangleright A \blacktriangleright B$。这个公式意味着，$B$由 Γ、A奠基，并且作为 B的根据的组成部分 A又由 Δ奠基。

我们采用不同的记法——即 ▷以及相应的括号方式——来表示奠基断言中嵌套的 ▶，以避免公式解释上的歧义。事实上，如果我们在奠基树最外层的奠基断言和嵌套的奠基断言中都使用 ▶，那么在公式 $\Gamma, (\Delta \blacktriangleright A) \blacktriangleright B$中，我们既可以认为公式 $\Delta \blacktriangleright A$本身是 B的根据的一部分，[7] 也可以认为 A是B 的根据的一部分，而 $\Delta \blacktriangleright A$是一个嵌套的奠基断言，用来指定 Δ 是 A的直接根据。为消除这种歧义，我们用 ▷而非 ▶来表示嵌套的奠基断言。因此，公式$\Gamma, (\Delta \blacktriangleright A) \blacktriangleright B$ 意味着 B 的根据包含公式 $\Delta \blacktriangleright A$；而公式 $\Gamma, (\Delta) \triangleright A \blacktriangleright B$ 则意味着 B 的根据包含公式A，而后者又有根据 Δ。直观地说，如果我们将奠基断言表示为树，其中一个节点的子节点代表该节点所对应公式的根据，那么 $\Gamma, (\Delta \blacktriangleright A) \blacktriangleright B$ 对应下方左侧的树，而 $\Gamma, (\Delta) \triangleright A \blacktriangleright B$ 对应下方右侧的树：

[7] F例如，根据某些奠基概念，如果 $\Gamma = \{p\}$ 和 B，这种情况可能会发生 $= p \wedge (\Delta \blacktriangleright A)$。

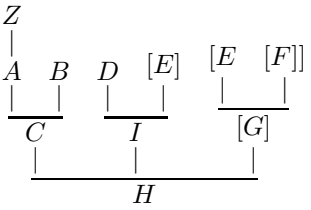


因此，没有公式具有 $(\Delta)\ \triangleright A$ 这种形式：我们仅使用 $\triangleright$ 来区分 $\Sigma \blacktriangleright B$ 中作为根据组成部分的子公式，以及 $\Sigma \blacktriangleright B$ 中作为 B 的直接根据（或直接根据的条件）的子公式。符号 $\triangleright$ 可以无限嵌套，以构造复杂的奠基树。

例如，我们可以写作

$$((Z) \triangleright A, B) \triangleright C\ ,\ \ (D[E]) \triangleright I\ \ [(E[F]) \triangleright G]\ \ \blacktriangleright\ \ H$$

表示 H 在条件 G下以 C 和 I 为根据； G 又在条件 F下以 E 为根据； I 在条件 E下以 D 为根据； C 以 A和 B为根据；最后， C 的根据 A 又以 Z为根据。这样的公式所表达的内容可以用下面的树来表示：



其中，与上面的树一样，一个节点的子节点表示该节点所对应公式的根据中的各个元素，条件则置于方括号内。[8]

因此，将 $\triangleright$ 嵌套在 $\blacktriangleright$中，并不意味着将一个奠基断言用作某个公式之根据的一部分，而是为了构造奠基断言链。正如前面提到的，奠基断言链类似于间接奠基断言，但二者有一个关键区别：间接奠基断言不包含任何关于使其成立的直接奠基断言的信息，而奠基树则包含了其所依据的所有直接奠基断言的信息。

的引入规则列于 $\triangleright$ 表5中。直观地说，这些规则使我们能够将奠基句嵌入其他奠基句中，形成一个

[8]我们不会将其用作正式的记法，而只把它当作一种视觉工具，以便更清楚地理解由 $\blacktriangleright$ 和 $\triangleright$构造的奠基树的结构。请注意，在以$[G]$为根的子树中，基础 E 和条件 $[F]$ 都被置于方括号内。对于对应于某个条件（例如 $[G]$）的节点的子节点，我们采用这一约定，以便在视觉上表明：一个条件的基础不应被计入结果的中介基础之中。

$$\frac{\Delta[\Theta] \blacktriangleright A \qquad \Gamma_1, A, \Gamma_2[\Xi] \blacktriangleright B}{\Gamma_1, (\Delta[\Theta]) \triangleright A, \Gamma_2[\Xi] \blacktriangleright B} \qquad \frac{\Delta[\Theta] \blacktriangleright C \qquad \Gamma[\Xi_1, C, \Xi_2] \blacktriangleright B}{\Gamma[\Xi_1, (\Delta[\Theta]) \triangleright C, \Xi_2] \blacktriangleright B}$$

表 5 奠基树算子的引入规则 $\triangleright$

一致的方式。例如，如果公式 D 可以由复合基础 A, C 奠基，从而我们可以推导出 $A, C \blacktriangleright D$，此外， C 可以由 B 奠基，从而我们可以推导出 $B \blacktriangleright C$；那么我们就可以将第二个奠基断言代入第一个，并得到： $A, (B) \triangleright C \blacktriangleright D$。请注意，正如上文所解释的，并由包围 B 的括号所强调的，根据这一记法，只有 A 和 C 是 D的直接根据； B 在公式中作为 C 的直接根据出现，因此仅仅是 C的间接奠基。相应地，公式 $A, (B) \triangleright C \blacktriangleright D$ 可以如下解读：“D 由 A 和 C直接奠基，而反过来， C 由 B直接奠基”。

$$\frac{\Gamma_1, (\Delta[\Theta]) \triangleright A, \Gamma_2[\Xi] \blacktriangleright B}{\Delta[\Theta] \blacktriangleright A} \qquad \frac{\Gamma[\Xi_1, (\Delta \mid \Theta) \triangleright C, \Xi_2] \blacktriangleright B}{\Delta[\Theta] \blacktriangleright C}$$

$$\frac{\Gamma_1, (\Delta[\Theta]) \triangleright A, \Gamma_2[\Xi] \blacktriangleright B}{\Gamma_1, A, \Gamma_2[\Xi] \blacktriangleright B} \qquad \frac{\Gamma[\Xi_1, (\Delta \mid \Theta) \triangleright C, \Xi_2] \blacktriangleright B}{\Gamma[\Xi_1, C, \Xi_2] \blacktriangleright B}$$

表6 奠基树算子的消去规则 $\triangleright$

$\triangleright$ 6 。

通过归纳定义，我们精确化这样一个想法：一次出现$\blacktriangleright$ 在公式 F 可能形成一棵奠基树，连同一些出现$\triangleright$ 在 F。这将在后面有用。特别是，我们形式地定义在一次出现 $\blacktriangleright$ 或 $\triangleright$ 和所有出现 $\triangleright$对应于同一奠基树的节点之间成立的传递关系。

定义4 我们称出现 $\star$ 是 $\blacktriangleright$ 的出现，或者是 $\triangleright$ *holds* 的出现；出现 $\star'$ 是 $\triangleright$的出现，当且仅当，

- 要么 $\star$ 是如下形式的公式或子公式的最外层算子

$$A_1, \ldots, A_n[C_1, \ldots C_n] \star B$$

并且 $\star'$ 是其中一个子公式 A_1、.. 、 A_n,$C_1, \ldots C_n$的最外层算子；

- 或 $\star$ 包含一个出现项 $\triangleright$，该项满足 $\star'$。

3.2.1 奠基树规则的刻画

现在我们将证明，奠基树算子的规则使我们能够在对象语言中将任何合法的奠基推导内化为一棵奠基树。此外，我们还将证明，如果一棵奠基树可以从一个相容集中推导出来，那么我们就能构造出一个结构与该奠基树完全相同的合法奠基推导。

为此，我们将形式化地说明任意奠基演算中的奠基树与奠基推导之间直观上的对应关系。

定义 5（奠基树对应关系）一种奠基树

$$G_1, \ldots, G_m[C_{m+1} \ldots C_n] \blacktriangleright A,$$

或奠基树的子公式（G_1，...，$G_m[C_{m+1} \ldots C_n]$）$\triangleright A$，当且仅当对应于奠基推导 δ，给主人留下些什么吧

- 的根 δ 是 A,
- 最后一个规则应用 r 的 δ 有 n 个前提（其中，假设前m 个不在方括号内，其余的都在），并且
- 对于每个 G_i，其中 $1 \leq i \leq n$，以下条件之一成立：

  – G_i 的最外层算子不是一个由 $\blacktriangleright$在 $\Gamma[\Delta] \blacktriangleright A$中的最外层出现所持有的 $\triangleright$ 出现——分别由 $\triangleright$在($\Gamma[\Delta]$) $\triangleright A$中的最外层出现所持有——，且 r 的第 i个前提是公式 G_i；– G_i 的最外层算子是一个由 $\blacktriangleright$在 G_1，.....，$G_m[C_{m+1} \ldots C_n] \blacktriangleright A$中的最外层出现所持有的 $\triangleright$ 出现——或者由 $\triangleright$在(G_1，.....，$G_m[C_{m+1} \ldots C_n]$) $\triangleright A$中的最外层出现所持有——，且 r 的第 i个前提的奠基推导对应于 G_i。

容易看出，如果我们假设规则的前提不可交换，那么每个奠基推导恰好对应一棵奠基树，反之亦然。如果我们希望允许规则前提交换，仍然可以通过考虑交换的基础和条件来保持一一对应关系。

现在我们可以证明，存在奠基推导意味着相应的奠基树可以通过奠基规则以及奠基算子 $\blacktriangleright$ 和 $\triangleright$的规则推导出来。

命题 **2** *If the grounding tree* $\Gamma[\Delta] \blacktriangleright A$ *corresponds to a legitimate grounding derivation in a fixed grounding calculus* κ*, then* $\Gamma[\Delta] \blacktriangleright A$ *is derivable from the hypotheses* $\Gamma\ \Delta$ *in any calculus that contains the rules of* κ *the rules for the* , , , *immediate grounding operator and the rules for the grounding tree operator.*

Proof 该证明通过对 A 的奠基推导中规则应用的数量进行归纳法来完成。在基本情形中，只应用了一条奠基规则来推导 A——事实上，如果推导中没有应用任何规则，那么它就不是奠基推导，而是逻辑推导。因此，根据我们为直接奠基算子制定的规则，可以推导出 $\Gamma[\Delta] \blacktriangleright A$。现在假设，若奠基树 $\Gamma'[\Delta'] \blacktriangleright A'$ 对应于 κ 中一个合法的奠基推导，且该推导包含少于 n 次规则应用，那么在扩展了我们为直接奠基和奠基树制定的规则后的 κ 中，可以推导出 $\Gamma'[\Delta'] \blacktriangleright A'$。我们要证明，对于包含 n 次奠基规则应用的奠基推导，这一结论同样成立。假设，包含 n 次规则应用的 A 的奠基推导 δ 对应于 $\Gamma[\Delta] \blacktriangleright A$。那么我们可以考察在 δ 中应用的最后一条规则，并且 δ 具有以下两种形式之一：

$$\dfrac{\Pi_1 \quad \dfrac{\Sigma \quad [\Theta]}{B}\, r \quad \Pi_2 \quad [\Xi]}{A}\, s \qquad\qquad \dfrac{\Pi \quad [\Xi_1 \quad \dfrac{\Sigma \quad [\Theta]}{B}\, r \quad \Xi_2]}{A}\, s$$

如果 δ 在左侧显示的上述形式中，那么根据归纳假设，存在一个由假设 $\Sigma^\star$、$\Theta^\star$ 可导出的奠基树 $\Sigma^\star[\Theta^\star] \blacktriangleright B$，它对应于 B 的奠基推导；并且存在一个由假设 $\Pi_1^\star$、B、$\Pi_2^\star$、$\Xi^\star$ 可导出的奠基树 $\Pi_1^\star$、B、$\Pi_2^\star[\Xi^\star] \blacktriangleright A$，它对应于我们对 A 的奠基推导，在该推导中，我们将 B 作为假设，而不是通过 r 推导出它。因此，根据

$$\dfrac{\Sigma^\star[\Theta^\star] \blacktriangleright B \qquad \Pi_1^\star, B, \Pi_2^\star[\Xi^\star] \blacktriangleright A}{\Pi_1^\star, (\Sigma^\star[\Theta^\star]) \triangleright B, \Pi_2^\star[\Xi^\star] \blacktriangleright A}$$

我们可以从假设$\Sigma^\star, \Theta^\star, \Pi_1^\star, B, \Pi_2^\star, \Xi^\star$推导出 $\Pi_1^\star, (\Sigma^\star[\Theta^\star]) \triangleright B, \Pi_2^\star[\Xi^\star] \blacktriangleright A$。但是由于 B 已经可以从假设$\Sigma^\star, \Theta^\star$中推导出，而且，此外，$\Gamma[\Delta] \blacktriangleright A$ 应当与 δ具有对应关系，因此 $\Gamma[\Delta]$ 的每个元素都适当地与 s 的相应前提或其奠基推导相关联，而后者根据归纳法假设又与 $\Pi_1^\star$的相应元素相对应，$(\Sigma^\star[\Theta^\star] \blacktriangleright B)$ $\Pi_2^\star[\Xi^\star]$。因此，由于奠基树与奠基推导之间的对应关系是一一对应的，我们可以得出 $\Gamma[\Delta] \blacktriangleright A = \Pi_1^\star, (\Sigma^\star[\Theta^\star]) \triangleright B, \Pi_2^\star[\Xi^\star] \blacktriangleright A$，从而我们可以从假设 Γ, Δ推导出 $\Gamma[\Delta] \blacktriangleright A$。

如果 δ 具有右侧上方所示的形式，类似的论证无论如何都会使我们得出结论：$\Gamma[\Delta] \blacktriangleright A$ 可由假设 Γ、Δ 推导得出。□

最后，我们证明，一旦固定了一个奠基演算，我们就可以重构出对应于任何可从一致假设集推导出的奠基树的奠基推导。[9]

命题 **3** *For any consistent calculus κ^+ defined by extending a grounding calculus κ with the rules for immediate grounding and grounding tree operators, if the grounding tree $\Gamma[\Delta] \blacktriangleright A$ is derivable in κ^+ from a consistent set of hypotheses, then*

[9]请注意，我们在此并未证明那个依赖于“奠基树是可证明的”这一假设的显然命题，而是证明了一个更强的命题，该命题仅依赖于奠基树可由一个一致的假设集推出。这是我们想要准确描述奠基树的行为所必需的，因为鉴于奠基的事实性，奠基主张通常被认为依赖于有关其组成部分为真的假设，因而并非可证明的，而只能由一致的假设集推出。

there is a legitimate grounding derivation in κ from a consistent set of hypotheses which exactly corresponds to $\Gamma[\Delta] \blacktriangleright A$.

Proof 让我们假设， δ 是一个 κ^+ 推导， $\Gamma[\Delta] \blacktriangleright A$ 来自一致的假设集 Π。由于 Π 是一致的， κ^+ 是一致的，并且 $\Gamma[\Delta] \blacktriangleright A$ 可由Π 在 κ^+中推导出， $\bot$ 不能由 $\Gamma[\Delta] \blacktriangleright A$ 在 κ^+中推导出。特别是，如果我们考察集合 $\{\Sigma_i[\Theta_i] \blacktriangleright B_i\}_{1\leq i\leq n}$ 中那些可通过 $\triangleright$ 消去规则从 $\Gamma[\Delta] \blacktriangleright A$ 推导出的直接奠基断言元素，我们有（i） $\Sigma_i[\Theta_i] \blacktriangleright B_i$对应于合法的基础规则应用

$$\frac{\Sigma_i[\Theta_i]}{B_i}$$

并且 (ii) 公式的集合 $\bigcup_0^n (\Sigma_i \cup \Theta_i)$ 是一致的。否则我们可以推导$\Gamma[\Delta] \blacktriangleright A$ 从 Π 然后 $\bot$ 从 $\Gamma[\Delta] \blacktriangleright A$ 通过 $\triangleright$ 消去规则和 $\blacktriangleright$ 消去规则，这与关于 Π 的一致性假设相矛盾。

但是，如果 (i) 和 (ii) 成立，我们就可以在 κ 中构造一个奠基推导，其假设为 $\bigcup_0^n(\Sigma_i \cup \Theta_i)$——也可能是这些假设的一个子集——且仅使用奠基规则应用

$$\frac{\Sigma_i[\Theta_i]}{B_i}$$

所得推导将与 $\Gamma[\Delta] \blacktriangleright A$ 完全对应。我们通过对 $\Gamma[\Delta] \blacktriangleright A$ 中 $\triangleright$ 的出现次数进行归纳法来证明这一点。如果 $\Gamma[\Delta] \blacktriangleright A$ 不包含 $\triangleright$的任何一次出现，它就对应于以下形式的奠基规则应用

$$\frac{\Gamma[\Delta]}{A}$$

这正是 κ 中的一个奠基推导，它对应于 $\Gamma[\Delta] \blacktriangleright A$ ，并且只使用 $\Gamma \cup \Delta$中的假设。于是我们假设，对于任何包含少于 $m > 0$ 个 $\triangleright$出现的奠基树 $\Gamma'[\Delta'] \blacktriangleright A'$ ，我们都可以从一个与 $\Gamma'[\Delta'] \blacktriangleright A'$相对应的相容集假设出发，在 κ 中构造一个奠基推导 δ'。我们证明，对于任何包含 m 个 $\triangleright$出现的奠基树 $\Gamma[\Delta] \blacktriangleright A$ ，这一点同样成立。由于 $\Gamma[\Delta] \blacktriangleright A$ 包含 $m > 0$ 个 $\triangleright$出现，因此，对于 $1 \leq k \leq p$，必定存在一些$(\Gamma''_j[\Delta''_j]) \triangleright A''_j$，它们属于 Γ， Δ，并且显然包含少于 m 个 $\triangleright$出现。如果我们考虑所有奠基树 $\Gamma''_j[\Delta''_j] \blacktriangleright A''_j$—— 也就是说，除了最外层算子之外，与 $(\Gamma''_j[\Delta''_j]) \triangleright A''_j$ 相同的奠基树 —— 那么根据归纳假设，存在 κ 中的奠基推导 δ''_j，它们对应于$\Gamma''_j[\Delta''_j] \blacktriangleright A''_j$，并且只依赖于这样的假设：这些假设可以从直接奠基断言推导出来，而这些直接奠基断言又可以通过对每个 $(\Gamma''_j[\Delta''_j]) \triangleright A''_j$ 进行 $\triangleright$ 消去而推导出来。现在，从 $\Gamma[\Delta] \blacktriangleright A$ 我们可以推导出一个形如 $\Gamma^\star[\Delta^\star] \blacktriangleright A$ 的直接奠基断言，其中 $\Gamma^\star$、$\Delta^\star$ 包含 Γ、Δ 的所有元素——这些元素的最外层算子不是 $\triangleright$——以及元素$(\Gamma''_j[\Delta''_j]) \triangleright A''_j$ 中出现的所有公式，其中 Γ、Δ 的最外层算子是 $\triangleright$。这一直接奠基断言对应于如下形式的奠基规则

$$\frac{\Gamma^\star[\Delta^\star]}{A}$$

并且通过将这一规则应用与我们奠基推导 δ''_j 的结论 A''_j 相组合，我们得到奠基推导 γ，它对应于 $\Gamma[\Delta] \blacktriangleright A$，且仅依赖于可由直接奠基断言推导出的假设；而这些直接奠基断言又可通过 $\triangleright$ 消去从 $\Gamma[\Delta] \blacktriangleright A$ 推导得出。事实上，关于 γ 的假设，它们显然具有所要求的性质，因为可通过 $\blacktriangleright$ 和 $\triangleright$ 消去从每个 $(\Gamma''_j[\Delta''_j]) \triangleright A''_j$ 推导出的内容，也可通过 $\blacktriangleright$ 和 $\triangleright$ 消去从 $\Gamma[\Delta] \blacktriangleright A$ 推导得出。至于 γ 与 $\Gamma[\Delta] \blacktriangleright A$ 之间的对应关系，我们有：γ 的根恰好是 A；根据 $\Gamma^\star$ 和 $[\Delta^\star]$ 的定义，γ 中最后应用的规则具有正确数量的方括号外前提和方括号内前提；根据 $\Gamma^\star$ 和 $[\Delta^\star]$ 的定义，γ 中最后应用的规则的所有不以 $\triangleright$ 为最外层算子的前提，都与 Γ 中不以 $\triangleright$ 为最外层算子的元素相同；最后，根据 γ、推导 δ''_j 以及 $\Gamma^\star$ 和 $[\Delta^\star]$ 的定义，γ 中最后应用的规则的所有以 $\triangleright$ 为最外层算子的前提，都由与 Γ 中以 $\triangleright$ 为最外层算子的元素相对应的奠基推导推导得出。□

我们通过强调以下一点来结束本节：在完全刻画奠基树算子的同时，我们也完全刻画了直接奠基算子；事实上，在基本情形中，奠基树就是一个直接奠基断言。

4 逻辑性和平衡

我们现在将研究为我们的三个奠基算子所提出的引入与消去规则的证明论行为，并试图确定这些规则是否以推理术语诱导出算子的良好定义，以及该定义能告诉我们关于算子本身的什么。我们将从语句算子推理定义研究中出现的一个最一般的划界问题开始：逻辑性问题。换言之，我们将探讨我们的奠基算子是否可被视为逻辑算子。为此，我们将采用源自结构主义证明论方法的方法来刻画逻辑常项的概念——例如参见 [12, 13]——该方法可追溯至 Koslow [14] 和Popper的工作，参见 [15]。我们将特别考虑两个传统的逻辑性判准，并表明虽然其中一个未被我们的奠基算子规则满足，但另一个被满足。然后我们将以一种我们认为适合所考虑奠基算子性质的方式弱化第一个判准，并表明弱化版本被其中一些满足，但并非全部。我们将得出关于算子及其所形式化关系的一些结论。

现在我们来考察第一个条件，这一条件常被用作算子的逻辑性判准：同一者的可推导性 [16]——在 [17]中也称为直接扩展。[10]

[10]与算子逻辑性的判准密切相关的另一个条件是 *uniqueness*，详见 [18] 对这两个判准及其之间关系的详细研究。我们决定在此不考虑 *uniqueness*，原因很简单：对于间接奠基算子和奠基树算子而言，它都显然不成立。其不成立的原因很简单，即这些算子的引入规则明确提及了算子的出现

我们先陈述这一判准的传统严格版本。

同一者的可推导性 若算子 $\circ(\ldots,)$ 满足同一者的可推导性，当且仅当对于任意一列公式 $A_1,\ldots,A_n$，我们都可以通过至少应用一次 $\circ$ 的消去规则，并且只使用 $\circ$的引入与消去规则，从 $\circ(A_1,\ldots,A_n)$ 推导出 $\circ(A_1,\ldots,A_n)$。

为了举出一个正面的例子，说明这一条件的运用，我们简要示范如何证明传统自然演绎合取规则满足该条件：

$$\frac{A \quad B}{A \wedge B}\ \wedge i \qquad\qquad \frac{A \wedge B}{A}\ \wedge e \qquad \frac{A \wedge B}{B}\ \wedge e$$

通过以下推导，可以很容易地证明这些规则满足同一者的可推导性：

$$\dfrac{\dfrac{A \wedge A}{A}\ \wedge e \quad \dfrac{A \wedge B}{B}\ \wedge e}{A \wedge B}\ \wedge i$$

从前面的例子可以看出，这一条件要求某算子的消去规则提供重新引入该算子本身所需的全部信息。此外还要注意，关键在于仅凭所考察算子的规则，就足以表明消去该算子所获得的信息足以重新引入它。更一般地说，这一条件要求消去规则所能得到的公式——它们决定了我们使用该算子的方式——与引入规则的前提——它们决定了通过应用该算子构造的句子的真值条件——之间具有直接的图式符合性。

直接奠基算子不满足同一者可推导性标准的这一严格版本，是一个相当明显的事实。事实上，我们无法在不使用并非该算子自身规则的规则的情况下引入这个算子：直接奠基算子的引入需要一次基础规则应用。换言之，消去规则的结论与引入规则的前提之间不存在直接的符合关系。

奠基算子不具备同一者的可推导性，并不是由于我们为该算子所采用的具体规则形式所致。事实上，即使我们考虑采用更直接的规则来引入奠基算子，同一者的可推导性仍然不成立。例如，考虑如下

本身。尽管这一缺失对于一般而言对归纳定义算子的证明论特征的研究可能颇有意义——事实上，归纳定义通常本质上依赖于对所定义算子的其他出现形式的指涉——但它并不能很好地说明我们在此着手研究的奠基算子之间在证明论行为上的差异。

合取的奠基规则：

$$\frac{\underline{\underline{A}} \quad \underline{\underline{B}}}{A \wedge B}$$

在我们的系统中，相对奠基断言 $A, B \blacktriangleright A \wedge B$将按如下方式引入：

$$\frac{\dfrac{\underline{\underline{A}} \quad \underline{\underline{B}}}{A \wedge B}}{A, B \blacktriangleright A \wedge B}$$

但我们也可以定义以下更直接的规则：

$$\frac{\underline{\underline{A}} \quad \underline{\underline{B}} \quad \underline{\underline{A \wedge B}}}{A, B \blacktriangleright A \wedge B}$$

然而，这类用于引入奠基算子的规则也会无法通过与同一者的可推导性相关的测试，因为通过消去奠基算子的一个通用实例所得到的东西，一般而言不足以推断出一个奠基断言。事实上，公式$G_1, \ldots, G_n, C$的句法形式是未知的，并且以下推导只能用于针对公式 C的某些特定选择来推断奠基断言：

$$\frac{G_1, \ldots, G_n \blacktriangleright C}{\underline{G_1} \quad \ldots \quad \underline{G_n} \quad \underline{C}}\ ?$$

在这种情况下，消去规则的结论与引入规则的前提之间没有图式符合性。

同一者可推导性的严格版本也不满足间接奠基算子。事实上，为了通过 $\gg$ 引入规则推导出形式为$\Gamma[\Delta] \gg A$ 的断言，我们需要推导出直接奠基断言 $\Gamma[\Delta] \blacktriangleright A$ ，或者推导出两个能够借助传递性得出 $\Gamma[\Delta] \gg A$的间接奠基断言。而且，我们当然无法仅通过使用 $\gg$ 消去规则——或者就此而言， $\gg$引入规则——从假设 $\Gamma[\Delta] \gg A$ 推导出这些断言中的任何一个。最后，即便是奠基树算子也不具备同一者的可推导性。不过，它之所以如此，唯一的原因在于，在基本情形中，奠基树就是一个直接奠基断言。而且，正如我们上文所论证的，直接奠基算子并不满足判准的严格版本。

同一者的可推导性失败并不特别令人惊讶。事实上，奠基算子并不应当是纯粹的逻辑算子——实际上，甚至逻辑奠基算子也不是——因为，为了引入它们，仅有关于该规则的前提是可推导的逻辑信息是不够的。[11] 这是这些算子的一个基本特征，因为

[11]在此，我们所说的 *logical information* 是指仅涉及某个公式是否能够从一组特定的假设中推导出来的信息；相反，例如，不包括涉及公式的句法形式、公式从一组假设中推导出来的特定方式，或公式的组成部分的语义解释的信息。

它们不仅涉及真和可推导性，还涉及相当多的非逻辑信息。以逻辑奠基为例，公式的句法复-杂度是奠基关系成立条件的一个基本组成部分；而其他复杂度概念或基础性概念，对于其他奠基概念也发挥着类似且同样基本的作用。一般而言，尽管引入奠基算子所需的所有约束，不一定都能在逻辑语言中得到明确表达——因而也就无法编码在消去规则的结论中——但对于不同的奠基概念，我们可以通过关于引入规则前提之推导的证明-论条件，保证这些约束得到满足。奠基算子规则的这一特征，直接反映了奠基关系的超内涵性质。在讨论超内涵性与非逻辑性之间这种联系的普遍性之前，让我们先简要说明我们所谓的超内涵性。若一个关系的项一般不能以逻辑等价者替换而保持真值不变，则该关系是超内涵的。换言之，如果 A 与 B 由一个超内涵关系相关联，那么 A 与 A' 的逻辑等价，并不足以推出 A' 也通过同一关系与 B 相关联。我们在此不能断言超内涵性总是蕴含非逻辑性，因为目前证明论中还不存在关于超内涵性的普遍说明。尽管如此，似乎有充分理由主张，超内涵性确实意味着同一者可推导性标准失效，因为使一个算子具有超内涵性的非逻辑要求，无法由能够通过规则结论传达的纯粹逻辑信息表达。在任何情况下，都无可置疑的是，所考察算子之所以是非逻辑的那些特定理由，正是说明它们何以具有超内涵性的同样理由。

如果我们将奠基算子未能满足逻辑性标准归因于超内涵性，那么自然会想知道：奠基算子规则是否真的不平衡——正如同一者的可推导性失败所表明的那样——还是它们只是体现了一种较弱形式的平衡，因此就超内涵算子的规则而言，仍是良构规则。拥有平衡的引入与消去规则集这一性质，确实并非逻辑联结词所独有。一般而言，语句算子有平衡规则是可取的，因为平衡规则不过意味着这些规则能够准确刻画算子的行为。也就是说，使用算子的规则——即其消去规则——使我们能够按照决定算子何时为真的规则——即其引入规则——所规定的方式，恰当地使用该算子。逻辑算子与其他类型语句算子的区别，并不在于其规则本身是否平衡，而在于这种平衡是否相对于我们认为可以合法地引入这些算子的信息种类成立，以及我们期望在消去这些算子时得到的信息种类成立。换言之，使一个算子成为逻辑算子的，并非仅仅是其引入与消去规则的平衡性，而是我们能够仅通过考察逻辑信息来证明这些规则是平衡的；也就是说，这种信息

关于公式的可推导性[12]。这种特定的平衡正是由直接图式符合性要求所强制的，该要求隐含在严格的同一者可推导性标准中，如前所述。

在接下来的章节中，我们将通过考察所提出的奠基算子规则是否允许能产生规范化结果的绕行约化，以及它们是否符合同一者可导性要求的较弱版本，来精确处理这些规则的平衡问题。

4.1 迂回消除性

我们现在将研究 ▶、 ≫ 和 ▷ 的规则是否满足第二个条件，我们称之为迂回消除性。此处所谓迂回消除性，是指支配算子使用的规则之应用，不会产生比应用那些确定由算子构造的语句何时为真的规则所需更多的信息。因此我们必须表明，奠基算子的消去规则并不能使我们从奠基主张中推断出比引入它们所需更多的内容。在我们的情形中，这归结为证明包含我们规则的演算系统可以证明适当的规范化结果。

尽管这一条件通常被视为算子逻辑性若干判据的一项基本要求——例如参见 [20–23]——但对于某个算子而言，存在一组能够产生规范化结果的规则，并不总被认为足以据此断定该算子是逻辑算子；关于逻辑性的主要现有观点综述，参见 [24] 。此外，正如我们在上一节末尾所论证的，句子算子并非逻辑算子这一事实，并不意味着其规则必定普遍表现不佳，也不意味着探究能否通过推理手段给出其意义的精确定义毫无价值。因此，我们将致力于分析奠基算子规则在迂回消除性方面的表现。

为了表明我们的算子具有迂回消除性，我们将定义与 [27] 中所采用的绕行约化在精神上相似的绕行约化，并将说明，为 ▶ 和 ▷ 所给出的归约可以用来推广 [25] 中所呈现的规范化结果；该结果针对的是一种基于 [6]中所引入的逻辑奠基概念的奠基演算。我们认为，这一特定规范化结果的意义并不局限于 [25] 中所呈现的演算所依据的奠基概念。事实上，用于证明该结果的规范化策略相当普遍且具有一般性，完全可以适用于具有相似证明论特征的各种奠基演算。另一方面，就 ≫ 的归约而言，我们将从技术和概念视角两方面讨论它们在规范化方面所提出的问题。

[12]如上所述，我们所说的 *logical information* ，是指有关 *that* 某些公式可从某些假设集中推导出来的信息。但例如，并非指有关 *way* 某些公式可从某些假设集中推导出来的信息。

归约规则对于 ▶、 ≫ 和 ▷ 分别列于表 7、 8、 9 和 10中。

我们首先证明，如果我们用针对 ▶ 和 ▷的规则扩展在 [25] 中提出的演算，那么它的规范化结果可以推广。随后，我们讨论针对 ≫的规则所产生的问题。

定义6我们将 G称为在 [25] 中定义的奠基演算，并将G+ 称为通过用我们关于直接奠基和奠基树算子的全部规则扩展 G 而定义的演算。

$$\frac{\dfrac{\dfrac{A_1 \quad \dots \quad A_n \quad [C_1 \quad \dots \quad C_m]}{B}}{A_1, \dots, A_n[C_1, \dots, C_m] \blacktriangleright B}}{A_i} \mapsto A_i \qquad \frac{\dfrac{\dfrac{A_1 \quad \dots \quad A_n \quad [C_1 \quad \dots \quad C_m]}{B}}{A_1, \dots, A_n[C_1, \dots, C_m] \blacktriangleright B}}{C_i} \mapsto C_i$$

$$\frac{\dfrac{\dfrac{A_1 \quad \dots \quad A_n \quad [C_1 \quad \dots \quad C_m]}{B}}{A_1, \dots, A_n[C_1, \dots, C_m] \blacktriangleright B}}{B} \mapsto \frac{A_1 \quad \dots \quad A_n \quad [C_1 \quad \dots \quad C_m]}{B}$$

$$\text{If } \frac{A_1, \dots, A_n[C_1, \dots, C_m] \blacktriangleright B}{\bot} \text{ is an elimination of } \blacktriangleright\text{, then } \blacktriangleright \text{ cannot have been introduced since } \frac{A_1 \quad \dots \quad A_n \quad [C_1 \quad \dots \quad C_m]}{B} \text{ is not a rule application}$$

表 7 ▶的绕行约化

我们回顾一下推导的归约定义以及一些相关术语。

定义 7（归约、可约表达式和临界规则）对于任意四个推导s、 s'、 d和 d'，如果 $s \mapsto s'$、 d包含 s作为子推导，并且 d'可以通过在 d中用 s' 替换 s 得到，那么关系 $d \mapsto d'$ 成立，我们称 d 归约到d'。

我们用 $\mapsto^*$ 表示 $\mapsto$的自反传递闭包。

通常，如果某个推导（或子推导）最底部的规则d 与紧接在其上方应用的某条规则构成了 $\mapsto$ 表 7、8、 9 和 10左侧所示的构型，那么我们称 d 是 a *redex*。我们称可约表达式的最后两个规则应用为可约表达式的 *critical rules* 。

我们给出一些简单且通常的定义，这些定义将在正规化证明中使用。

$$\dfrac{\dfrac{\Gamma[\Delta] \gg A \qquad \Gamma_1, A, \Gamma_2[\Delta_1] \gg B}{\Gamma_1, \Gamma, \Gamma_2[\Delta_1, \Delta] \gg B}}{\gamma} \mapsto \dfrac{\Gamma_1, A, \Gamma_2[\Delta_1] \gg B}{\gamma}$$

where $\gamma \in \Gamma_1, A, \Gamma_2, \Delta_1, B$

$$\dfrac{\dfrac{\Gamma[\Delta] \gg A \qquad \Gamma_1, A, \Gamma_2[\Delta_1] \gg B}{\Gamma_1, \Gamma, \Gamma_2[\Delta, \Delta_1] \gg B}}{\gamma} \mapsto \dfrac{\Gamma[\Delta] \gg A}{\gamma}$$

where $\gamma \in \Gamma, \Delta, A$

$$\dfrac{\dfrac{\Gamma[\Delta] \gg A \qquad \Gamma_1[\Delta_1, A, \Delta_2] \gg B}{\Gamma_1[\Delta_1, \Gamma, \Delta, \Delta_2] \gg B}}{\gamma} \mapsto \dfrac{\Gamma_1[\Delta_1, A, \Delta_2] \gg B}{\gamma}$$

where $\gamma \in \Gamma_1, \Delta_1, A, \Delta_2, B$

$$\dfrac{\dfrac{\Gamma[\Delta] \gg A \qquad \Gamma_1[\Delta_1, A, \Delta_2] \gg B}{\Gamma_1[\Delta_1, \Gamma, \Delta, \Delta_2] \gg B}}{\gamma} \mapsto \dfrac{\Gamma[\Delta] \gg A}{\gamma}$$

where $\gamma \in \Gamma, \Delta, A$

$$\dfrac{\dfrac{\Gamma[\Delta] \blacktriangleright A}{\Gamma[\Delta] \gg A}}{\gamma} \mapsto \dfrac{\Gamma[\Delta] \blacktriangleright A}{\gamma}$$

where $\gamma \in \Gamma, \Delta, A$

表 8 针对 $\gg$的绕行约化

定义 8（逻辑复杂度）公式的逻辑复杂度按照惯例通过计算公式中的符号数量来定义。

定义 9（约化式复杂度）约化式 r 的复杂度定义为由 r 的最上层临界规则所引入的公式的逻辑复杂度。

定义10（正规形式）我们称 G+ 推导 d 是正规的，或处于正规形式，如果不存在满足 $d \mapsto d'$ 的推导 d' 。

显然，正规性与不含任何可约式是等价的性质。

$$\dfrac{\dfrac{\Gamma[\Delta] \blacktriangleright A_i \qquad A_1, \ldots, A_i, \ldots, A_n[\Sigma] \blacktriangleright B}{A_1, \ldots, (\Gamma[\Delta] \blacktriangleright A_i), \ldots, A_n[C] \blacktriangleright B}}{\Gamma[\Delta] \blacktriangleright A_i} \mapsto \Gamma[\Delta] \blacktriangleright A_i$$

$$\dfrac{\dfrac{\Gamma[\Delta] \blacktriangleright C_i \qquad \Sigma[C_1, \ldots, C_i, \ldots, C_n] \blacktriangleright B}{\Sigma[C_1, \ldots, (\Gamma[\Delta] \blacktriangleright C_i), \ldots, C_n] \blacktriangleright B}}{\Gamma[\Delta] \blacktriangleright C_i} \mapsto \Gamma[\Delta] \blacktriangleright C_i$$

$$\dfrac{\dfrac{\Gamma[\Delta] \blacktriangleright A_i \qquad A_1, \ldots, A_i, \ldots, A_n[\Sigma] \blacktriangleright B}{A_1, \ldots, (\Gamma[\Delta] \blacktriangleright A_i), \ldots, A_n[\Sigma] \blacktriangleright B}}{A_1, \ldots, A_i, \ldots, A_n[\Sigma] \blacktriangleright B} \mapsto A_1, \ldots, A_i, \ldots, A_n[\Sigma] \blacktriangleright B$$

$$\dfrac{\dfrac{\Gamma[\Delta] \blacktriangleright C_i \qquad \Sigma[C_1, \ldots, C_i, \ldots, C_n] \blacktriangleright B}{\Sigma[C_1, \ldots, (\Gamma[\Delta] \blacktriangleright C_i), \ldots, C_n] \blacktriangleright B}}{\Sigma[C_1, \ldots, C_i, \ldots, C_n] \blacktriangleright B} \mapsto \Sigma[C_1, \ldots, C_i, \ldots, C_n] \blacktriangleright B$$

表 9 ▷的绕行约化，第 1 部分

$$\dfrac{\dfrac{\Gamma[\Delta] \blacktriangleright A_i \qquad A_1, \ldots, A_i, \ldots, A_n[\Sigma] \blacktriangleright B}{A_1, \ldots, (\Gamma[\Delta] \blacktriangleright A_i), \ldots, A_n[C] \blacktriangleright B}}{\Xi[\Theta] \blacktriangleright D} \mapsto \dfrac{\Gamma[\Delta] \blacktriangleright A_i}{\Xi[\Theta] \blacktriangleright D} \quad \text{where } (\Xi[\Theta] \blacktriangleright D) \in \Gamma \cup \Delta$$

$$\dfrac{\dfrac{\Gamma[\Delta] \blacktriangleright A_i \qquad A_1, \ldots, A_i, \ldots, A_n[\Sigma] \blacktriangleright B}{A_1, \ldots, (\Gamma[\Delta] \blacktriangleright A_i), \ldots, A_n[C] \blacktriangleright B}}{\Xi[\Theta] \blacktriangleright D} \mapsto \dfrac{A_1, \ldots, A_i, \ldots, A_n[\Sigma] \blacktriangleright B}{\Xi[\Theta] \blacktriangleright D}$$

$$\text{where } (\Xi[\Theta] \blacktriangleright D) \in \{A_1, \ldots, A_n\} \cup \Sigma$$

$$\dfrac{\dfrac{\Gamma[\Delta] \blacktriangleright C_i \qquad \Sigma[C_1, \ldots, C_i, \ldots, C_n] \blacktriangleright B}{\Sigma[C_1, \ldots, (\Gamma[\Delta] \blacktriangleright C_i), \ldots, C_n] \blacktriangleright B}}{\Xi[\Theta] \blacktriangleright D} \mapsto \dfrac{\Gamma[\Delta] \blacktriangleright C_i}{\Xi[\Theta] \blacktriangleright D} \quad \text{where } (\Xi[\Theta] \blacktriangleright D) \in \Gamma \cup \Delta$$

$$\dfrac{\dfrac{\Gamma[\Delta] \blacktriangleright C_i \qquad \Sigma[C_1, \ldots, C_i, \ldots, C_n] \blacktriangleright B}{\Sigma[C_1, \ldots, (\Gamma[\Delta] \blacktriangleright C_i), \ldots, C_n] \blacktriangleright B}}{\Xi[\Theta] \blacktriangleright D} \mapsto \dfrac{\Sigma[C_1, \ldots, C_i, \ldots, C_n] \blacktriangleright B}{\Xi[\Theta] \blacktriangleright D}$$

$$\text{where } (\Xi[\Theta] \blacktriangleright D) \in \Sigma \cup \{C_1, \ldots, C_n\}$$

表 10 ▷的绕行约化，第 2 部分

规范化将遵循 [28] 中所采用的思路。证明背后的主要直觉是，通常通过应用归约规则，我们会消除一个具有某种复杂度的可约表达式，并可能生成复杂度更低的新可约表达式。对于大多数归约而言，这就是全部要点。如果我们归约推导中的一个适当可约表达式，就会使最大复杂度降低

可约表达式的，或者具有最大复杂度的可约表达式数量的减少。如果我们所有的归约规则都允许这样的论证，我们可以通过对一对值进行归纳来证明正规化过程的终止性，这对值对应于推导中可约表达式的最大复杂度以及推导中出现的具有最大复杂度的可约表达式的数量。然而，并非所有归约规则都表现得如此良好，实际上其中一些实现了规则之间的置换，而置换不会改变可约表达式的复杂度。因此，我们需要一种方法来跟踪置换并在我们的复杂度度量中考虑它们。为此，我们改编了片段的概念，其定义见[28]。片段是推导树内连接两个规则应用的路径，并满足以下两个条件：首先，它连接两个规则应用，如果它们一个紧接另一个出现，将形成一个可约式——并且该可约式必须不同于置换可约式；其次，它可以通过使用置换来缩短，以便最终获得由这两个规则应用形成的可约式。

通过检查 [25], 中的证明，很容易看出，如果我们将 ▶ 和 ▷ 的引入规则视为其他任何引入规则，并将 ▶ 和 ▷ 的消去规则视为其他任何消去规则，那么所有定义和置换都可以推广。直观地说， ▶ 和 ▷ 消去规则及可约表达式的行为与 $\wedge$ 消去规则及可约表达式的行为非常相似，而它们的引入规则之间的差异并不会产生任何特别的问题。

我们回顾在 [25] 中引入的定义。

定义11（片段（改编自 [28]中的定义6.1.1）和片段复杂度）对于任意G+ 推导 d， d中长度为 n 的片段是一个公式出现序列 A_1、 ..... 、 A_n，其中公式出现位于 d 中，并满足以下条件。

1. 对于 $1 < i < n$，以下情况之一成立：
   - A_i 是 d 中一次 $\vee$ 消去应用的次要前提，其结论为 $A_{i+1} = A_i$,
   - A_i 是一个非逻辑规则的前提，其结论 A_{i+1} 与 A_i 具有相同的逻辑复杂度（对于[25], 中的演算，这些规则都是 ε 规则，以及那些不会导致从前提到结论的逻辑复杂度发生变化的逆规则）。
2. A_n 既不是某个 $\vee$ 消去的次要前提，也不是某个非逻辑规则的前提，而该规则的结论与A_n 具有相同的逻辑复杂度（对于 [25], 中的演算，这些规则都是 ε 规则，以及那些不会导致从前提到结论的逻辑复杂度发生变化的逆规则).
3. A_1 既不是某个 $\vee$ 消去规则的结论，也不是某个非逻辑规则的结论；该规则的前提与 A_1 具有相同的逻辑复杂度（对于 [25], 中的演算，这些规则都是 ε 规则，以及那些不会导致前提到结论的逻辑复杂度发生变化的逆规则）。

对于任何片段，如果

- $n > 0$ 或
- A_1 是引入规则的结论且 A_n 是消去规则的主前提

则该片段的复杂度为 A的逻辑复杂度。否则，该片段的复杂度为0。

注意，一个片段中的所有公式都具有相同的逻辑复杂度。这对于 $\vee$ 消去的情况是显而易见的，并且对于其他情况，它根据假设成立。

我们引入一些术语来描述推导中两个片段的相对位置，并证明一个关于推导中片段排列的简单事实，该事实将在规范化证明中用到。

定义12（片段术语）如果一个片段只包含一个公式出现，那么我们所说的 *reducing the segment* 是指：若可能，约化其上方和下方紧邻处应用了临界规则的非置换可约式；否则，约化其结论为该片段最底公式的 $\vee$ 消去的置换可约式。

如果片段 r 的最底公式出现在片段 s的最底公式上方，则称片段 r *occurs above* s 。

如果存在推导 ρ 和 σ，使得 r 中的某个公式出现在 ρ中， s 中的某个公式出现在 σ中， ρ的根和 σ 的根是同一个规则应用的前提，并且相对于该规则应用， ρ 的根出现在 σ 的根右侧，则称片段 r *occurs to the right* 是片段 s 的.....。

命题

4 *For any two distinct segments in a derivation d if neither is to the , , right of the other then one is above the other.*

Proof 参见[25]。 □

我们证明，在 [25] 中给出的演算扩展了 ▶ 和 ▷的规则后是可规范化的。

定理

5 *For any derivation d there is a derivation d' such that d can be reduced , to d' in a finite number of reductions and d' is normal.*

Proof 我们采用以下归约策略。我们归约复杂度最高的片段中最靠右、且不位于任何其他复杂度最高的片段之下的片段。根据命题 4，我们总能找到这样的片段。

我们证明这种归约策略总是产生一系列有限长度的归约，并得到一个正规形式。

我们将推导 d 的复杂度定义为自然数的三元组（m、n、u），其中，m是具有极大片段复杂度的片段的复杂度，n是具有片段复杂度m的片段长度之和，而 u 是 d 中的规则应用次数。然后我们固定一个一般推导 d，并依据自然数三元组上的字典序进行归纳法推理。

如果 d的复杂度为$(0, 0, u)$，则 d是正规的，且该断言成立。

现在假设 d的复杂度为(m, n, u)，$m + n > 0$，并且对于每个比 d更简单的推导，该断言都成立。由于 $m + n > 0$，在 d 中必须至少有一个极大段不出现在任何其他极大段之下。我们归约其中一个这样的片段，并根据归约的形状分情况讨论。我们只考虑一些示例情况，其他情况可在[25]中找到。

- $$\dfrac{\dfrac{\dfrac{A_1 \quad \ldots \quad A_n \quad [C_1 \quad \ldots \quad C_m]}{B}}{A_1, \ldots, A_n[C_1, \ldots, C_m] \blacktriangleright B}}{A_i} \;\mapsto\; A_i$$

  通过归约，我们消除了一个极大段。我们现在证明，没有最大复杂度的片段被复制，没有最大复杂度的片段的长度被增加，并且没有片段变得与被归约的片段一样复杂；因此 d' 的复杂度为$(m', n', u') < (m, n, u)$，因为我们要么降低了片段的最大复杂度，要么降低了具有最大复杂度的片段的长度之和。对于 d 中的每个片段，以下情况恰好有一个成立：(i)该片段不包含 A_i和 $A_1, \ldots, A_n[C_1, \ldots, C_m] \blacktriangleright B$的任何显示出现，$(ii)$该片段包含 A_i 的一些显示出现，(iii)该片段包含 $A_1, \ldots, A_n[C_1, \ldots, C_m] \blacktriangleright B$的显示出现。如果$(i)$该片段既未被归约修改也未被复制。如果$(ii)$归约可能将该片段与另一个满足$(ii)$的片段连接，但所得片段仍然比被归约的片段复杂度更低，因为 A 比 $A_1, \ldots, A_n[C_1, \ldots, C_m] \blacktriangleright B$复杂度更低。我们刚刚消除了唯一满足$(iii)$的片段。

- $$\dfrac{\dfrac{\dfrac{A_1 \quad \ldots \quad A_n \quad [C_1 \quad \ldots \quad C_m]}{B}}{A_1, \ldots, A_n[C_1, \ldots, C_m] \blacktriangleright B}}{C_i} \;\mapsto\; C_i$$

  通过这次归约，我们消除了一个极大段。现在证明：没有任何复杂度最高的片段被复制，没有任何复杂度最高的片段长度增加，也没有任何片段变得与被归约的片段一样复杂；因此，d' 的复杂度为 $(m', n', u') < (m, n, u)$，因为我们要么降低了片段的最大复杂度，要么减少了复杂度最高的片段长度之和。对于 d 中的每个片段，以下情况恰有一种成立：(i) 该片段不包含C_i和 $A_1, \ldots, A_n[C_1, \ldots, C_m] \blacktriangleright B$的任何一个所显示的出现位置；$(ii)$ 该片段包含其中某些所显示的出现位置

C_i的所显示出现位置；(iii) 该片段包含 $A_1, \ldots, A_n[C_1, \ldots, C_m] \blacktriangleright B$的所显示出现位置。如果 (i)，该片段既未被归约修改，也未被复制。如果 (ii)，归约可能会将该片段与另一个满足 (ii) 的片段合并，但所得片段仍然没有被归约的片段复杂，因为 C_i 比 $A_1, \ldots, A_n[C_1, \ldots, C_m] \blacktriangleright B$复杂度低。我们刚刚消除了唯一一个满足 (iii) 的片段。

- $$\dfrac{\dfrac{\dfrac{A_1 \quad \ldots \quad A_n \quad [C_1 \quad \ldots \quad C_m]}{B}}{A_1, \ldots, A_n[C_1, \ldots, C_m] \blacktriangleright B}}{B} \mapsto \dfrac{A_1 \quad \ldots \quad A_n \quad [C_1 \quad \ldots \quad C_m]}{B}$$

通过这次归约，我们消除了一个极大片段。现在我们证明：没有任何复杂度最高的片段被重复，任何复杂度最高的片段的长度都没有增加，也没有任何片段变得与被归约的片段一样复杂；因此，d' 的复杂度是 $(m', n', u') < (m, n, u)$，因为我们要么降低了片段的最大复杂度，要么减少了复杂度最高的片段的长度总和。对于 d 中的每个片段，以下情况恰有一种成立：(i) 该片段不包含所显示的B和 $A_1, \ldots, A_n[C_1, \ldots, C_m] \blacktriangleright B$的任何出现；$(ii)$ 该片段包含所显示的 B的某次出现；(iii) 该片段包含所显示的 $A_1, \ldots, A_n[C_1, \ldots, C_m] \blacktriangleright B$的那次出现。如果满足$(i)$，那么该片段既未被归约修改，也未被归约重复。如果满足(ii)，归约可能会将该片段与另一个满足(ii)的片段合并，但所得片段仍不如被归约的片段复杂，因为 B 不如 $A_1, \ldots, A_n[C_1, \ldots, C_m] \blacktriangleright B$复杂。我们刚刚消除了唯一一个满足$(iii)$的片段。

- $$\dfrac{\dfrac{\Gamma[\Delta] \blacktriangleright A_i \qquad A_1, \ldots, A_i, \ldots, A_n[\Sigma] \blacktriangleright B}{A_1, \ldots, (\Gamma[\Delta] \blacktriangleright A_i), \ldots, A_n[\Sigma] \blacktriangleright B}}{\Gamma[\Delta] \blacktriangleright A_i} \mapsto \Gamma[\Delta] \blacktriangleright A_i$$

通过归约我们消除一个极大段。我们现在证明没有最大复杂度的片段被复制，没有最大复杂度的片段的长度被增加，并且没有片段变得与被归约的那个一样复杂；因此 d' 的复杂度是$(m', n', u') < (m, n, u)$因为我们要么降低了片段的最大复杂度，要么降低了具有最大复杂度的片段的长度之和。对于 d 中的每个片段，以下恰好有一个成立：(i)该片段不包含任何显示的 $\Gamma[\Delta] \blacktriangleright A_i$ 的出现以及显示的 A_1的出现，$\ldots$，$(\Gamma[\Delta] \blacktriangleright A_i) \ldots$，$A_n[\Sigma] \blacktriangleright B$，$(ii)$该片段包含一些显示的 $\Gamma[\Delta] \blacktriangleright A_i$的出现，$(iii)$该片段包含显示的 A_1的出现，$\ldots$，$(\Gamma[\Delta] \blacktriangleright A_i) \ldots$，$A_n[\Sigma] \blacktriangleright B$。如果$(i)$该片段既没有被归约修改也没有被复制。如果$(ii)$归约可能将该片段与另一个满足$(ii)$的片段连接起来，但得到的片段仍然比被归约的那个复杂度更低

由于 $\Gamma[\Delta] \blacktriangleright A_i$ 比 A_1 更不复杂，.....，$(\Gamma[\Delta] \blacktriangleright A_i)$.....，$A_n[\Sigma] \blacktriangleright B$。我们刚刚消除了唯一满足 (iii) 的片段。

- $$\dfrac{\dfrac{\Gamma[\Delta] \blacktriangleright C_i \qquad \Sigma[C_1, \ldots, C_i, \ldots, C_n] \blacktriangleright B}{\Sigma[C_1, \ldots, (\Gamma[\Delta] \blacktriangleright C_i), \ldots, C_n] \blacktriangleright B}}{\Sigma[C_1, \ldots, C_i, \ldots, C_n] \blacktriangleright B} \quad \mapsto \Sigma[C_1, \ldots, C_i, \ldots, C_n] \blacktriangleright B$$

通过归约，我们消去一个极大段。现在证明：没有任何复杂度达到极大的片段被复制，没有任何复杂度达到极大的片段的长度增加，也没有任何片段变得与被归约的片段一样复杂；因此，d' 的复杂度是 $(m', n', u') < (m, n, u)$，因为我们要么降低了片段的极大复杂度，要么减少了复杂度达到极大的片段的总长度。对于 d 中的每个片段，以下情况恰有一种成立：(i) 该片段不包含所显示的 $\Sigma[C_1, \ldots, C_i, \ldots, C_n] \blacktriangleright B$的任何出现，也不包含所显示的 $\Sigma[C_1, \ldots,$的出现 $(\Gamma[\Delta] \blacktriangleright C_i)$ ....., C_n；(ii) 该片段包含所显示的 $\Sigma[C_1, \ldots, C_i, \ldots, C_n] \blacktriangleright B$的某些出现；$(iii)$ 该片段包含所显示的 $\Sigma[C_1, \ldots,$ $(\Gamma[\Delta] \blacktriangleright C_i)$ ....., C_n的出现。如果是 (i)，则该片段既未被归约修改，也未被归约复制。如果是 (ii)，则归约可能会将该片段与另一个满足 (ii) 的片段合并，但所得片段仍然比被归约的片段简单，因为 $\Sigma[C_1, \ldots, C_i, \ldots, C_n] \blacktriangleright B$ 比 $\Sigma[C_1, \ldots,$ $(\Gamma[\Delta] \blacktriangleright C_i)$ ....., C_n简单。我们刚刚消去了唯一满足 (iii) 的片段。

- $$\dfrac{\dfrac{\Gamma[\Delta] \blacktriangleright A_i \qquad A_1, \ldots, A_i, \ldots, A_n[\Sigma] \blacktriangleright B}{A_1, \ldots, (\Gamma[\Delta] \blacktriangleright A_i), \ldots, A_n[C] \blacktriangleright B}}{\Xi[\Theta] \blacktriangleright D} \quad \mapsto \dfrac{\Gamma[\Delta] \blacktriangleright A_i}{\Xi[\Theta] \blacktriangleright D}$$

其中$(\Xi[\Theta] \blacktriangleright D) \in \Gamma \cup \Delta$。通过归约，我们消除了一个最大复杂度的片段。现在我们证明，没有任何最大复杂度的片段被复制，没有任何最大复杂度片段的长度增加，也没有任何片段变得与被归约的片段一样复杂；因此，d' 的复杂度是 $(m', n', u') < (m, n, u)$，因为我们要么降低了片段的最大复杂度，要么降低了具有最大复杂度的片段的长度之和。对于 d 中的每个片段，以下情形恰好有一种成立：(i) 该片段不包含 $\Gamma[\Delta] \blacktriangleright A_i$ 的所显示出现、A_1 的所显示出现，.. . , $(\Gamma[\Delta] \blacktriangleright A_i)$. . ., $A_n[\Sigma] \blacktriangleright B$ 以及 $\Xi[\Theta] \blacktriangleright D$ 的所显示出现；(ii) 该片段包含 $\Gamma[\Delta] \blacktriangleright A_i$ 的所显示出现；(iii) 该片段包含 $\Xi[\Theta] \blacktriangleright D$ 的所显示出现；(iv) 该片段包含 A_1的所显示出现，. . . ,($\Gamma[\Delta] \blacktriangleright A_i$). . ., $A_n[\Sigma] \blacktriangleright B$。如果 (i)，则该片段既未被归约修改，也未被归约复制。如果 (ii) 或 (iii)，归约可能会将该片段与另一个满足(ii)或(iii) 的片段连接起来，但所得片段的复杂度低于被归约的片段，因为 $\Gamma[\Delta] \blacktriangleright A_i$ 和 $\Xi[\Theta] \blacktriangleright D$——后者是 $\Gamma[\Delta] \blacktriangleright A_i$的子公式——都比 A_1 的复杂度低，.. . , $(\Gamma[\Delta] \blacktriangleright A_i)$ . .. , $A_n[\Sigma] \blacktriangleright B$。我们刚刚消除了唯一满足 (iii) 的片段。

□

4.1.1 间接奠基与全局绕行消除性

既然我们已经表明， ▶ 和 ▷ 的绕行约化使我们能够将[25], 中的规范化结果加以推广，我们现在考察由 ≫的规则所生成的绕行的归约。由于由 ▶、▷ 和 ≫ 的规则生成的每一个单独绕行都可以得到适当归约——如果我们考察表7、8、9和10中的归约规则，这一点是显而易见的——我们可以断言，三个算子都具有迂回消除性的局部形式：从消除奠基主张$\Gamma[\Delta] \circ A$中出现的奠基算子 ∘ 所获得的信息，不超过通过引入算子 ∘推导奠基主张 $\Gamma[\Delta] \circ A$ 所需的信息。但是，尽管 ▶ 和 ▷ 的绕行约化以相当通常的方式降低了所考察推导中出现的公式的逻辑复杂度——因此可以使用标准技术为 ▶ 和 ▷ 证明全局绕行消除性结果——≫ 的绕行约化通常并不只生成逻辑复杂度更低的绕行。这是由于一个非常特殊的事实： ≫ 算子的引入规则——与 ▶ 和 ▷的引入规则相反——其前提未必比其结论更简单。如果我们所依据的奠基演算捕捉到一种非逻辑奠基概念，那么可推导的奠基主张中就可能出现这种情况。例如，如果根据我们的奠基概念， $p \gg q$ 和 $q \gg r$不是相互矛盾的奠基主张，那么

$$\frac{p \gg q \quad q \gg r}{p \gg r}$$

是一个完全合理的通过 ≫ 引入而得到的、在假设 $p \gg q$ 和 $q \gg r$ 是正确的奠基断言的前提下，对一个真实的间接奠基断言所作的推导。而在这里，从证明论视角来看，没有任何东西告诉我们， $p \gg r$ 比 $p \gg q$ 和 $q \gg r$更复杂。然而，即使我们的基础奠基演算捕捉了逻辑奠基这一概念，类似的问题仍然可能出现。事实上，我们无法一般性地排除这样的可能性：矛盾的奠基断言会出现在推导中——否则，例如，我们就不可能证明某些奠基断言是错误的或相互矛盾的。因此，例如，以下构型当然可能出现在逻辑奠基推导中：

$$\frac{p \gg q \vee r \vee s \quad q \vee r \vee s, t \gg u}{p, t \gg u}$$

如上所述， ≫ 引入规则可能导致逻辑复杂度降低；反过来，这意味着某些绕行约化会产生逻辑复杂度更高的绕行。如下所示：

$$\dfrac{\dfrac{\dfrac{p \gg v \wedge z \quad v \wedge z \gg q \vee r \vee s}{p \gg q \vee r \vee s} \qquad q \vee r \vee s, t \gg u}{p, t \gg u}}{p} \quad \mapsto \quad \dfrac{\dfrac{p \gg v \wedge z \quad v \wedge z \gg q \vee r \vee s}{p \gg q \vee r \vee s}}{p}$$

在此，我们消除一个其复杂度等于公式 $p,t \gg u$ 的复杂度的迂回，并生成一个其复杂度等于 $p \gg q \vee r \vee s$的复杂度的迂回。显然， $\gg$ 消除可能引发的逻辑复杂度降低，与这样一个事实密切相关：通过传递性结合两个奠基主张往往会导致信息损失。

从技术角度来看，总之，基于其图式形式，对包含 $\gg$ 规则的演算系统进行规范化的一般终止性论证似乎存在很大问题。这反过来意味着，目前没有明确的方法可以通过规范化终止性论证，证明 $\gg$ 相对于一个一般的奠基演算具有全局绕行消除性。

然而，如果我们将采用间接奠基陈述的预期含义视为合法选项，并特别利用每个此类陈述与一棵奠基树之间的对应关系，那么通过对正规化过程进行终止性证明来展示全局绕行消除性结果，似乎仍有不同的方法可用于 $\gg$ 。确实，尽管 $\gg$ 引入规则的结论不一定比其前提更复杂——这正是绕行的归约在标准意义下不会降低推导复杂度的原因——但 $\gg$ 引入规则的一次应用的结论总是对应于一棵比其前提更大的奠基树。之所以如此，是因为通过传递性连接两个奠基主张，恰好对应于用另一棵奠基树替换一棵奠基树的一个叶节点。因此，似乎可以利用基于这种对应关系的复杂度度量，来证明包含 $\gg$ 绕行的推导也能够被规范化。不过，目前尚不清楚，这样的复杂度度量是否能与用于证明其他绕行归约序列终止的逻辑复杂度良好地相互作用。还应注意，这样的复杂度度量不会是句法的，而是语义的；事实上，如果 $\gg$ 引入规则的一个前提是违反奠基的复杂度约束的错误的奠基陈述——例如，就大多数逻辑奠基概念而言， $p \wedge q \wedge r \wedge s \gg p$——那么，基于相应奠基树的复杂度度量就是未定义的，因为不存在相应的奠基树。要成功地将这一技术应用于 $\gg$ 绕行归约的终止性证明，还需要满足另一项要求：能够仅通过简单检查每个间接奠基断言本身，确定与之对应的奠基树。对于大多数逻辑奠基概念而言，这是可行的——确实，重构与任何合法的间接逻辑奠基陈述相对应的奠基树并不困难——但对于更复杂的奠基概念，仅仅检查一个间接奠基陈述可能不足以确定它所指向的树结构，尤其是在底层直接奠基关系的传递闭包不可判定时。不过，这方面的确定性技术结果显然必须留待有关各个演算系统的研究，因为这些演算系统刻画了特定的奠基概念。

关于奠基关系的可判定性或不可判定性与间接奠基这一相对概念的行为之间联系的进一步讨论，推迟到第4.2 节，因为这些联系在该节中也将发挥关键作用。

4.2 同一者可推导性的弱版本

在第 4节开头，我们已经表明，同一者可推导性标准并不适用于我们的奠基算子，并且我们已将这一失败与奠基的超内涵性以及奠基算子严格说来并非逻辑算子这一事实联系起来。随后，我们已经表明，尽管我们的奠基算子规则无法通过逻辑性测试，但只要它们具有迂回消除性，它们仍然适当地定义了直接奠基算子和奠基树算子。那么，人们或许会想知道：尽管我们的奠基算子规则不能被视为定义逻辑算子，它们是否仍然具有某种完全的平衡。因而，我们致力于定义一种较弱版本的同一者可推导性标准，以确定我们的奠基算子引入规则和消去规则在何种意义上是平衡的；而且，相对于一般的超内涵算子规则，这一标准或许会证明是有用的。

为奠基算子定义一个更为细致的平衡判准，还存在另一个理由，这与三个奠基算子的同一者可推导性以不同方式失效这一事实有关。事实上，奠基树算子的同一者可推导性条件之所以失效，基本上取决于这一判准在直接奠基中的失效。另一方面，间接奠基算子的该条件的失效，相对于直接奠基算子的证明论特征而言，则是彻底且独立的。这表明，一个更为细致的判准或许能使我们更好地理解问题所在，并有可能区分同一者可推导性标准的部分失效——即相对于直接奠基和奠基树的失效——与更为严重的失效——即相对于间接奠基的失效。此外，这或许还能进一步阐明我们在研究迂回消除性时所遇到的三个算子在行为上的差异为何产生。

因此，我们通过考虑以下事实来定义同一者可推导性的弱版本判准：我们的直接奠基算子的引入规则实质上指涉其他规则——也就是所选基础演算的奠基规则。这意味着，在某种意义上，在定义该判准时，我们恰当地重视了这样一个事实：严格来说，我们的算子并不是逻辑算子，因为它们是超内涵的。事实上，奠基的超内涵性与以下事实有本质关联：有效的奠基主张取决于一个非逻辑层级——例如，对于逻辑奠基而言，这一层级是由句法复杂度所诱导的层级；对于形而上学奠基而言，则是由形而上学基础性所诱导的层级。通过限制其奠基规则图式的形式，可以将这一层级内化于奠基演算中。这一弱

因此，我们将要引入的同一者可推导性标准，可以被看作是将该标准的严格版本相对于奠基所依据的非逻辑层级进行相对化，而这种相对化是通过将该标准相对于所考察的奠基演算中所包含的奠基规则集合进行相对化来实现的。

我们将要提出的判准告诉我们，通过消去一个算子的某次出现所获得的信息，包含重新引入该算子出现所需的全部信息。这正是同一者严格可推导性判准告诉我们的关于一个算子的内容；二者之间唯一但至关重要的差别在于，该判准的弱版本使我们能够考虑并使用我们背景演算的规则——以便证明所考察算子的引入规则与消去规则之间存在这种平衡——而该算子的引入规则正是参照这些背景演算的规则的。这个观点反映了这样一种思想：正在研究的引入规则本质上依赖于——正如它们应当依赖的那样——编码在基础奠基演算本身的规则中的非逻辑信息。特别是，我们的引入规则的前提是通过哪些具体规则推导出来的，这些规则传达了相关信息。因此，这些信息不仅关乎前提已经被推导出来这一事实，也关乎它们是以何种方式被推导出来的。由于我们在弱化该判准时所考虑的附加信息是非逻辑的，该判准并不适合作为逻辑性判准。然而，弱判准仍然表明，该算子的引入规则与消去规则之间存在一种平衡，尽管严格来说该算子并不是逻辑算子，因而这种平衡并不是建立在这些规则之间直接的图式符合性关系之上的。

同一者可推导性的弱版本一个算子 $\circ(\ldots,)$关于演算 κ 满足同一者可推导性的弱版本，当且仅当，对于任意公式列表 $A_1,\ldots,A_n$，只要 $\circ(A_1,\ldots,A_n)$ 可以作为一个

$\circ$ 引入应用，我们可以从 $\circ(A_1,\ldots,A_n)$ 推导出 $\circ(A_1,\ldots,A_n)$，方法是至少应用一次 $\circ$ 消去规则，并且只使用 $\circ$ 的规则，或使用在 $\circ$ 引入规则的适用条件中明确提及的规则。

换言之，判准要求：如果可以引入 $\circ(A_1,\ldots,A_n)$ 中最外层的 $\circ$，那么通过消去它所提供的逻辑信息，以及 $\circ$ 引入规则的定义中所包含的非逻辑信息，必须足以重新引入这个 $\circ$。要求必须存在一个结论为

$\circ(A_1,\ldots,A_n)$在此处是必要的，因为奠基算子的超内涵性质要求我们对其引入规则施加特别严格的条件，因此可能存在一个公式 A with 一个奠基算子作为最外层联结词，使得没有任何规则应用能够具有

A 作为结论。[13] 这是这样一个事实的副产品：为了定义奠基算子的正确规则，我们还需要对前提 的推导方式施加条件，而不仅仅是对它们的可推导性施加条件。

我们现在展示，我们的直接奠基算子 ▶ 以及我们的奠基树算子满足同一者可推导性的弱版本标准。

命题 **6** *The immediate grounding operator* ▶ *enjoys weak deducibility of identicals.*

Proof 考虑任意一个形如 $A_1, \ldots, A_n[A_{n+1}, \ldots, A_m] \blacktriangleright A_{m+1}$ 的公式，其中所显示的 ▶ 出现位置不包含任何 $\triangleright$ 的出现，并假设存在一个以 , , 为结论的 ▶ $A_1 \ \ldots \ A_n[A_{n+1}, \ldots, A_m] \ \blacktriangleright$ 引入规则应用。我们首先论证，若情况如此，则

$$\frac{A_1 \quad \ldots \quad A_n[A_{n+1} \quad \ldots \quad A_m]}{A_{m+1}} \ r$$

必须是合法的基础规则应用。事实上，如果 A_1、.....、 $A_n[A_{n+1}, \ldots, A_m] \blacktriangleright A_{m+1}$ 可以作为一次 ▶ 引入规则应用的结论，那么 r 必定是在这次规则应用的紧接上一步中应用的。

因此，从 A_1、.....、 $A_n[A_{n+1}, \ldots, A_m] \blacktriangleright A_{m+1}$ 到 A_1、.....、 $A_n[A_{n+1}, \ldots, A_m] \blacktriangleright A_{m+1}$ 的推导，若其中只包含 ▶ 规则的应用，以及在 ▶ 引入规则的适用条件中明确提及的规则的应用，并且至少包含一次 ▶ 消去规则的应用，就是

$$\dfrac{\dfrac{A_1, \ldots, A_n[A_{n+1}, \ldots, A_m] \blacktriangleright A_{m+1}}{A_1} \quad \ldots \quad \dfrac{A_1, \ldots, A_n[A_{n+1}, \ldots, A_m] \blacktriangleright A_{m+1}}{[A_m]}}{\dfrac{A_{m+1}}{A_1, \ldots, A_n[A_{n+1}, \ldots, A_m] \blacktriangleright A_{m+1}}}$$

□

命题 **7** *The grounding tree operator enjoys weak deducibility of identicals.*

Proof 考虑任意一个形如 $\Gamma[\Xi] \blacktriangleright B$ 的公式，使得其中所显示的 ▶ 的出现至少包含一个 $\triangleright$ 的出现，而该出现是形如 $(\Delta[\Theta]) \ \triangleright A$ 的子公式的最外层算子；该子公式或者出现在 Γ 中，或者出现在 Ξ 中。

[13] 如果我们将奠基算子与外延算子（例如经典逻辑和直觉主义逻辑中的合取和析取）或内涵算子（例如直觉主义蕴涵和大多数模态逻辑中的必然算子）进行比较，就会发现：可以为外延联结词定义引入规则，而这些规则对其前提的推导不作任何条件限制；也可以为内涵联结词定义引入规则，而这些规则只对用于推导其前提的假设有所要求。因此，大多数外延算子的引入规则始终可以应用，而不论其前提是如何推导出来的。尽管某个内涵算子的引入规则 $\circ$ 可能无法应用于由某些假设推导出的公式，但通常对于任意公式 $\circ(A_1, \ldots, A_n)$，总能找到一组假设，使我们能够通过 $\circ$ 引入规则推导出 $\circ(A_1, \ldots, A_n)$。但若考察奠基算子的引入规则，情况并非总是如此。事实上，由于这些规则的条件，某些奠基断言永远不会成为合法引入规则应用的结论。

若 $(\Delta[\Theta]) \triangleright A$ 出现在 Γ 中，则 $\Gamma[\Xi] \blacktriangleright B = \Gamma'$，$(\Delta[\Theta]) \triangleright A$，$\Gamma''[\Xi] \blacktriangleright B$，且仅使用 $\triangleright$ 规则从 $\Gamma\ \Xi\ [\blacktriangleright]B$ 推导 $\Gamma\ \Xi\ [\blacktriangleright]B$ 的推导如下：

$$\dfrac{\dfrac{\Gamma', (\Delta[\Theta]) \triangleright A, \Gamma''[\Xi] \blacktriangleright B}{\Delta[\Theta] \blacktriangleright A} \quad \dfrac{\Gamma', (\Delta[\Theta]) \triangleright A, \Gamma''[\Xi] \blacktriangleright B}{\Gamma', A, \Gamma''[\Xi] \blacktriangleright B}}{\Gamma', (\Delta[\Theta]) \triangleright A, \Gamma''[\Xi] \blacktriangleright B}$$

如果 $(\Delta[\Theta]) \triangleright A$ 出现在 Ξ 中，那么 $\Gamma[\Xi] \blacktriangleright B = \Gamma[\Xi', (\Delta[\Theta]) \triangleright A, \Xi''] \blacktriangleright B$ 且仅使用关于 $\triangleright$ 的规则从 $\Gamma\ \Xi \blacktriangleright B$ 推导 $\Gamma\ \Xi \blacktriangleright B$ 的推导如下：

$$\dfrac{\dfrac{\Gamma[\Xi', (\Delta[\Theta]) \triangleright A, \Xi''] \blacktriangleright B}{\Delta[\Theta] \blacktriangleright A} \quad \dfrac{\Gamma[\Xi', (\Delta[\Theta]) \triangleright A, \Xi''] \blacktriangleright B}{\Gamma[\Xi', A, \Xi''] \blacktriangleright B}}{\Gamma[\Xi', (\Delta[\Theta]) \triangleright A, \Xi''] \blacktriangleright B}$$

□

关于命题 7的证明，有三点需要说明。首先，我们不应被证明中所使用的推导仅包含关于 $\triangleright$的规则这一事实所误导：该证明并未表明同一者可推导性的严格版本对于奠基树算子成立，因而也未表明$\triangleright$是一个逻辑算子。事实上，在基本情形中，奠基树是一个直接奠基断言。因此，严格来说，为了表明奠基树算子具有同一者可推导性的弱版本，我们既需要命题 7 的证明，也需要命题6的证明。

第二点评论涉及用于证明命题7的推导的实际可构造性。事实上，对于 Γ 的任意元素，以及 Ξ 以 $\triangleright$ 作为最外层算子时，都可以构造出这一推导。这一点很重要，因为它告诉我们无需作出选择，并且我们可以从 $\triangleright$ 的任意一个最外层出现处开始，以消除 $\triangleright$ 的所有相关出现，而这些出现由所考虑的 $\blacktriangleright$所持有。

第三条说明涉及“消去一棵奠基树”这一表达可能存在的歧义。事实上，严格来说，奠基树并非由算子的一次出现表示，而是由 $\blacktriangleright$ 的一次出现，以及该次出现所包含的 $\triangleright$ 的所有出现共同表示。相应地，形式为 $(\Delta[\Theta]) \triangleright A$ 的表达式可以作为公式的子公式出现——例如，在 Γ、$(\Delta[\Theta]) \triangleright A[\Xi] \blacktriangleright B$中——但它本身从来不是公式。因此，当我们有一个形式为 Γ、$(\Delta[\Theta]) \triangleright A[\Xi] \blacktriangleright B$ 的公式时，不能将单独的 $\blacktriangleright$ 出现或单独的 $\triangleright$ 出现视为奠基树算子的一次应用。从实质上说，奠基树算子的一次应用应包含这两者，以及其间连续嵌套的 $\triangleright$ 的所有其他出现。因此，可以主张，为了消去一棵奠基树的一次应用，必须消去由最外层的 $\blacktriangleright$出现所包含的 $\triangleright$ 的所有出现。比如，为了消去奠基树 $(p, q) \triangleright p \wedge q$、$(r, \neg s) \triangleright r \vee s \blacktriangleright (p \wedge q) \wedge (r \vee s)$，并得到构成它的所有直接奠基主张，我们必须构造以下三个推导：

$$\delta_1 \quad = \quad \dfrac{(p, q) \triangleright p \wedge q, (r, \neg s) \triangleright r \vee s \blacktriangleright (p \wedge q) \wedge (r \vee s)}{p, q \blacktriangleright p \wedge q}$$

$$\delta_2 \quad = \quad \frac{(p,q) \triangleright p \wedge q, (r, \neg s) \triangleright r \vee s \blacktriangleright (p \wedge q) \wedge (r \vee s)}{r, \neg s \blacktriangleright r \vee s}$$

and

$$\delta_3 \quad = \quad \frac{\dfrac{(p,q) \triangleright p \wedge q, (r, \neg s) \triangleright r \vee s \blacktriangleright (p \wedge q) \wedge (r \vee s)}{p \wedge q, (r, \neg s) \triangleright r \vee s \blacktriangleright (p \wedge q) \wedge (r \vee s)}}{p \wedge q, r \vee s \blacktriangleright (p \wedge q) \wedge (r \vee s)}$$

那么，即使我们采用这种对奠基树算子出现的消除概念，同一者可推导性的弱版本也适用于奠基树算子。事实上，显然，对于奠基树算子的任何出现，都可以进行类似于前一个例子中所展示的分解，并且可以利用所导出的直接奠基断言来完全重新引入原来的奠基树。例如，对于前一个例子中所考虑的公式，结果如下：

$$\frac{\dfrac{}{r, \neg s \blacktriangleright r \vee s}\,\delta_2 \qquad \dfrac{\dfrac{}{p, q \blacktriangleright p \wedge q}\,\delta_1 \qquad \dfrac{}{p \wedge q, r \vee s \blacktriangleright (p \wedge q) \wedge (r \vee s)}\,\delta_3}{(p,q) \triangleright p \wedge q, r \vee s \blacktriangleright (p \wedge q) \wedge (r \vee s)}}{(p,q) \triangleright p \wedge q, (r, \neg s) \triangleright r \vee s \blacktriangleright (p \wedge q) \wedge (r \vee s)}$$

我们正式证明，始终可以通过 $\triangleright$ 消去规则完全分解任何非平凡的奠基树——对于平凡的奠基树，即直接奠基断言，命题6 的证明已经足够——并通过 $\triangleright$ 引入规则将其重新组合。为此，我们首先定义奠基树的形式概念，即大小。直观地说，奠基树的大小 $\Gamma[\Delta] \blacktriangleright A$ 是 $\triangleright$ 的最外层出现所包含的 $\blacktriangleright$ 在 $\Gamma[\Delta] \blacktriangleright A$ 中的出现次数加 1。

定义 13（奠基树的大小 $|\ |$）直接奠基断言 $\Gamma[\Delta] \blacktriangleright A$ 的大小 $|\Gamma[\Delta] \blacktriangleright A|$ 为 1。形式为 $\Gamma[\Delta] \blacktriangleright A$ 的奠基树的大小 $|\Gamma[\Delta] \blacktriangleright A|$ 为 $|(\Gamma_1[\Delta_1] \blacktriangleright A_1)| + \ldots + |(\Gamma_n[\Delta_n] \blacktriangleright A_n)| + 1$，其中 Γ 和 Δ 的所有元素都以 $\triangleright$ 为最外层算子，分别为 $(\Gamma_1[\Delta_1]) \triangleright A_1$，……，$(\Gamma_n[\Delta_n]) \triangleright A_n$。

命题 8 *For any grounding tree $\Gamma[\Delta] \blacktriangleright A$ of size greater than 1, it is possible to derive from it all immediate grounding claims that compose it by $\triangleright$ elimination rules and derive it from these immediate grounding claims by $\triangleright$ introduction rules.*，

Proof 证明通过对 $\Gamma[\Delta] \blacktriangleright A$ 的大小进行归纳法完成。若 $|\Gamma[\Delta] \blacktriangleright A| = 1$，则该陈述显然成立。现在假设该陈述对所有大小小于 n 的奠基树都成立，我们证明它对所有大小为 n 的奠基树也成立。考虑任意一个大小为 $n > 1$ 的奠基树 $\Gamma[\Delta] \blacktriangleright A$。由于 $|\Gamma[\Delta] \blacktriangleright A| > 1$，必定至少有一个 $\Gamma[\Delta] \blacktriangleright A$ 的子公式 $(\Sigma[\Theta]) \triangleright B$，满足以下任一条件：$\Gamma[\Delta] \blacktriangleright A = \Gamma'$、$(\Sigma[\Theta]) \triangleright B$、$\Gamma''[\Delta] \blacktriangleright A =$ 或 $\Gamma[\Delta] \blacktriangleright A = \Gamma[\Delta']$，$(\Sigma[\Theta]) \triangleright B$、$\Delta''] \blacktriangleright A$。

如果 $\Gamma[\Delta] \blacktriangleright A = \Gamma'$ $\Sigma[\Theta] \triangleright B$ $\Gamma''[\Delta] \blacktriangleright A$ $|\Sigma[\Theta] \blacktriangleright B| < n >|$，$(\ )$，，那么，显然，，，$\Gamma''[\Delta] \blacktriangleright A|$。因此，根据归纳假设，存在推导 γ 关于 $\Sigma[\Theta] \blacktriangleright B$ 以及 δ 关于 Γ'，B，$\Gamma''[\Delta] \blacktriangleright A$，它们来自组成它们的直接奠基断言

仅包含 $\triangleright$的引入与消去规则。因此我们可以构造如下推导：

$$\frac{\overset{\gamma}{\Sigma[\Theta] \blacktriangleright B} \qquad \overset{\delta}{\Gamma', B, \Gamma''[\Delta] \blacktriangleright A}}{\Gamma', (\Sigma[\Theta]) \triangleright B, \Gamma''[\Delta] \blacktriangleright A}$$

这也验证了对于 $\Gamma[\Delta] \blacktriangleright A$ 的陈述。

另一方面，如果

$\Gamma[\Delta] \blacktriangleright A = \Gamma[\Delta',\ \Sigma[\Theta]\ \triangleright B\ \Delta''] \blacktriangleright A \qquad | \Sigma[\Theta] \blacktriangleright 0$，，那么。因 $B \mid< n >\mid \Gamma[\Delta', B, \Delta''] \blacktriangleright A \mid$

此，根据归纳假设，存在推导γ，其从构成它们的直接奠基断言出发，仅包含针对 $\triangleright$的引入与消去规则，并且分别推导出 $\Sigma[\Theta] \blacktriangleright B$ 和 δ 的 $\Gamma[\Delta', B, \Delta''] \blacktriangleright A$。因此，我们可以构造出以下推导：

$$\frac{\overset{\gamma}{\Sigma[\Theta] \blacktriangleright B} \qquad \overset{\delta}{\Gamma[\Delta', B, \Delta''] \blacktriangleright A}}{\Gamma[\Delta', (\Sigma[\Theta]) \triangleright B, \Delta''] \blacktriangleright A}$$

这也验证了当前命题对 $\Gamma[\Delta] \blacktriangleright A$ 同样成立。 □

现在让我们考察间接奠基。首先，显然，对于一个固定的间接奠基断言 $\Gamma[\Delta] \gg A$，不存在通过使用 $\gg$ 规则、$\blacktriangleright$ 规则以及可能的奠基规则来构造从 $\Gamma[\Delta] \gg A$ 到 $\Gamma[\Delta] \gg A$ 的非平凡推导的一般策略。事实上，例如，我们没有一般方法知道哪些奠基断言可以用来引入 $\Gamma[\Delta] \gg A$中 $\gg$ 的最外层出现。即使我们假定直接奠基的基础关系是可判定的，并且我们的奠基断言 $\Gamma[\Delta] \gg A$ 是可推导的，也可能不存在任何找到其推导的机械方法。例如，如果我们考察任意一个可有限公理化但不可判定的形式理论，那么我们会发现，直接奠基算子满足同一者可导性要求，因为从某些前提到某个结论的一步可导性是可判定的；但另一方面，间接奠基算子不满足同一者可导性要求，因为可导关系并不是可判定的。这显然且本质上与传递性所意味着的信息损失有关。事实上，间接奠基算子内在地包含了一个结果与其某个间接根据之间的关系；这一关系可以用直接奠基明确表述，也可以与刻画直接奠基关系的演算中的可推导性概念隐含地联系起来——可能是通过推导的分支这一概念。能够由有限演算刻画的直接奠基算子表达了某个规则应用的存在，而间接奠基算子表达了具有某种结构的复杂推导的存在。因此，为了说明这类直接奠基断言的可推导性，只需检查有限规则集中是否存在一条能够应用于奠基断言内部出现的公式的规则。另一方面，为了说明间接奠基断言的可推导性，必须重构一个大小未知的特定复杂推导，而间接奠基断言并未提供关于该推导结构的任何信息。奠基树的情形与间接奠基的情形相似。

奠基，但存在一个本质区别：奠基树表达了具有某种结构的复杂推导的存在——该结构由断言规定——并且包含某些公式——这些公式同样由断言规定。由于奠基树明确提供了重构证明该断言本身的可推导性所需的全部信息，因此很容易重构出这样的推导，并将奠基树的可推导性归约为直接奠基断言的可推导性。这一方面的间接奠基与另一方面的直接奠基和奠基树之间的区别，显然与以下事实有关：依据传递性，以一组直接奠基断言为基础获得一个间接奠基断言，会相对于原始的直接奠基断言集合造成相当大的信息损失。为了定义间接奠基而取直接奠基的传递闭包所导致的信息损失问题，也具有哲学意义，因为关于这一问题的讨论可见于[29–31] 见证者。

5 结论

我们介绍了三组推理规则，它们可用于在一般奠基演算的基础上，定义三种不同类型的奠基算子的行为：直接奠基算子、间接奠基算子——对应于直接奠基关系的传递闭包——以及奠基树算子，即一种能够让我们将直接奠基断言构成的链内化，同时不丢失任何相关信息的算子。我们刻画了这些算子的行为，并研究了它们的证明论性质。

特别是，我们证明了这三种算子都具有局部绕行消除性，因为可以为它们分别定义绕行约化。不过，我们也证明，尽管直接奠基算子 $\blacktriangleright$ 和奠基树算子 $\triangleright$的规则具有示意性行为，使我们能够推广现有奠基演算的规范化结果——[25] 中的规范化结果便展示了这种推广——从而也证明相对于奠基演算的全局绕行消除性；但间接奠基算子 $\gg$ 的规则在全局绕行消除性结果方面带来了严重的技术难题。正如我们所论证的，这些难题与间接奠基的概念特征有关，而 $\gg$ 算子旨在将这些特征形式化。

我们还考察了同一者可推导性标准，它与迂回消除性标准一道，被提出作为检验逻辑性的标准。我们已经证明，这三个算子都未能通过这一检验，因此主张存在有力的技术证据反对奠基算子是逻辑算子的说法。我们讨论了这一失败背后的哲学原因，以及奠基的超内涵性与奠基算子的非逻辑性之间的关联。

为了区分所考虑算子的逻辑性与其引入与消去规则之间的平衡相互作用，我们随后定义了同一者可推导性标准的一个较弱版本，该版本考虑了奠基的超内涵性质。根据较弱判准，我们表明，虽然直接奠基和奠基树算子的规则展现了满足同一者可推导性的弱版本判准所需的引入与消去之间的平衡，但间接奠基算子的规则则不然。我们讨论了间接奠基算子在全局绕行消除性和同一者可推导性的弱版本方面的不良行为，这是鉴于通过取直接奠基的传递闭包来定义间接奠基意味着可能相当大的信息损失。我们提出了与奠基传递性所带来的哲学问题的可能平行。

本文提出了两个一般性问题。第一个问题涉及间接奠基作为一种奠基概念是否合适。本文所呈现的结果并非旨在构成关于特定奠基关系的特定特征的确凿证据，而只是阐明一类极为一般的形式奠基算子所共有的特征；然而，此处研究的间接奠基算子在技术上的不足，似乎指向一些非常具体的哲学问题，而这些问题同样关系到非形式的间接奠基概念。不过，本文呈现的技术结果也非常清楚地表明，这些不足本质上与底层直接奠基关系的特定特征息息相关，因此未必适用于所有奠基概念。因此，进一步具体研究奠基关系的传递性与可判定性之间的关系，将会十分有意义。第二个问题涉及能否提出一个普遍有效的论证，以确立超内涵性与逻辑性标准之间的确切联系。目前尚无法给出这样的论证，因为它需要对超内涵性作出一般的证明论刻画；不过，近来哲学界对超内涵概念的关注，无疑使发展合适的形式方法成为一项极具意义的工作。

致谢。

声明

参考文献

[1] Schnieder，B.：《因为》的逻辑。《符号逻辑评论》4(3)，445–465（2011）[2]Fine，K.：《基础指南》，第37–80页。剑桥大学出版社，剑桥（2012）

ance[3] Fine，K.：基础的纯逻辑。《符号逻辑评论》5(1)，1–25（2012）[4] Correia，F.，Schnieder，B.：《形而上学奠基：理解实在的结构》。剑桥大学出版社，剑桥（2012）[5] Correia，F.：逻辑基础。《符号逻辑评论》7(1)，31–59（2014）[6] Poggiolesi，F.：论“完全且直接的形式奠基”概念的定义。《综合》193，3147–3167（2016）[7] Poggiolesi，F.：论为“完全且直接的形式奠基”概念构建一种逻辑。《综合》195，1231–1254（2018）[8]Korbmacher，J.：部分基础的公理理论II。《哲学逻辑杂志》47(2)，193–226（2018）[9]Korbmacher，J.：部分基础的公理理论I。《哲学逻辑杂志》47(2)，161–191（2018）[10] Poggiolesi，F.：多种奠基类型的基于证明的框架。《逻辑与分析》252，387–414（2020）[11] Bolzano，B.：《科学理论》。牛津大学出版社，牛津（2014）。Rolf George和PaulRusnok译[12]Doˇsen，K.：逻辑常项：证明论论稿。博士论文，牛津大学（1980）[13] Doˇsen，K.：作为标点符号的逻辑常项。《圣母大学形式逻辑杂志》30(3)，362–381（1989）[14] Koslow，A.：《逻辑的结构主义理论》。剑桥大学出版社，剑桥（2005）[15]Schroeder-Heister，P.：Popper 的逻辑结构主义理论。阿什盖特，伦敦（2005）[16] Hacking，I.：什么是逻辑？《哲学杂志》76(6)，285–319（1979）[17] Prawitz，D.：证明论中的思想与成果。载于：《第二届斯堪的纳维亚逻辑研讨会论文集》（1971）[18] Naibo，A.，Petrolo，M.：同一者的唯一性与可推导性是同一回事吗？《理论》81(2)，143–181（2015）

奠基算子：传递性与树、逻辑性与平衡 43[19] Belnap, N.D.：《Tonk、plonk 与 plink》。Analysis 22(6), 130–134（1962）[20] Prawitz, D.：《意义与证明：论经典逻辑与直觉主义逻辑之间的冲突》。Theoria 43(1), 2–40（1977）[21]Dummett, M.：《形而上学的逻辑基础》。哈佛大学出版社，剑桥（1991）[22] Read, S.：《经典逻辑中的和谐与自主性》。Jo urnal of Philosophical Logic 29(2), 123–154（2000）[23]Tennant, N.：《推理主义、逻辑主义、和谐与一个对位》。载于《Crispin Wright 纪念文集：逻辑、语言与数学》2，105–132（2007）[24] Poggiolesi, F.：《模态命题逻辑的根岑演算》。Springer，多德雷赫特（2010）[25]Genco, F.A.：《作为逻辑推导的形式解释》。Journal of Applied Non-Classical Logics 31(3–4), 279–342（2021）[26] Genco, F.A., Poggiolesi, F., Rossi, L.：《奠基、量词与悖论》。Journal of Philosophical Logic 50, 1417–1448（2021）[27] Prawitz, D.：《自然演绎：一项证明论研究》。Courier Dover Publications，纽约（2006）[28]Troelstra, A.S., Schwichtenberg, H.：《基础证明论》。剑桥大学出版社，剑桥（1996）[29]Schaffer, J.：《奠基、传递性与对比性》，第122–138页。剑桥大学出版社，剑桥（2012）[30] Litland, J.E.：《论奠基传递性的若干反例》。Essays in Philosophy 14(1), 19–32（2013）[31] Raven, M.J.：《基础是否是严格偏序？》。美国哲学季刊 50(2), 193–201（2013）