直觉主义 S4 是可判定的

Marianna Girlando Roman Kuznets Sonia Marin Marianela Morales Lutz Straßburger 阿姆斯特丹大学 维也纳工业大学 伯明翰大学 Inria Saclay 阿姆斯特丹，荷兰 维也纳，奥地利 伯明翰，英国 帕莱索，法国 帕莱索，法国

arXiv:2304.12094v1 [cs.LO] 24 Apr 2023

摘要——本文证明了由 Fischer Servi 首次提出的直觉主义模态逻辑 S4 的可判定性。这解决了一个悬而未决近三十年的问题，该问题最早由 Simpson 在 1994年 的博士论文中提出。我们通过在一个带标签演绎系统中执行证明搜索得出这一结果。该系统不只在标签上使用一个二元关系，而是使用两个：一个对应于模态逻辑的可达关系，另一个对应于直觉主义 Kripke 框架的序关系。我们的搜索算法要么输出一个证明，要么输出一个有限反模型，因此还确立了直觉主义 S4 的有限模型性质，这也是该领域另一个长期悬而未决的问题。

1. 引言

要使一种逻辑具有可判定性，就必须存在一个递归过程，对每个公式判定它是否为该逻辑的定理，理想情况下，该过程最终应以演绎系统中的证明或适当的反模型结束。

直觉主义命题逻辑的可判定性。Gentzen 于 1935年 首先证明了直觉主义命题逻辑的可判定性，他使用的是自己设计的序列演算 [11]。他的方法是限制由给定初始序列推出的后果数量。后来人们独立地注意到（Ono [23], 、Ketonen [14], 和 Kleene [15]），如果将结构规则内置于记法中，就可以采用自根向上的证明搜索方法。也就是说，持续搜索序列演算证明，直到它以证明终止，或者到达一个沿着通向该序列的分支已经出现过的序列；在后一种情形下，可以从该分支重构出一个 Kripke 反模型 [28]。有关详细的历史论述，请参见 Dyckhoff 的综述 [5]。

经典模态逻辑的可判定性。关于经典模态逻辑（基于经典命题基础）的可判定性，早期就已有研究。Ladner 仅基于 Kripke 语义，为包括 S4 在内的一些常见模态逻辑提供了判定过程，其中没有提及序列演算 [17]。此外，上述过程同样可以应用于模态逻辑 S4，因为它也是自反传递 Kripke 框架的逻辑。一般而言，对于具有无割序列系统的模态逻辑，可以采用通过序列演算进行证明搜索的方法 [25] （或者在某些情形下，采用使用类似方法的嵌套序列系统 [3]）。

使用带标签序列的判定程序。带标签序列将克里普克语义的某些要素内化到序列语法中，这对直觉主义命题逻辑 [6], 以及模态逻辑 [22]的判定程序产生了若干有趣的影响。

这是 LICS'23 论文的作者版本，附有详细附录。

首先，在直觉主义命题逻辑的语境下，没有必要将自身限制在单结论序列上（即后件中至多包含一个公式的序列）。这意味着所有规则都可以设为可逆，从而无需回溯，并使证明搜索过程具有确定性。其次，在两种语境中，用于确保终止的循环检查都局限于证明搜索树某个分支的最上层序列，而不是应用于该分支上的两个序列之间：只需检查最上层序列中的两个标签是否携带相同的公式即可。最后，当发现循环时，可以直接根据最上层序列构造反模型。事实上，带标签序列的语法使得从序列中读取反模型变得十分容易，只需替换标签或添加回边即可；这两种构造都可以直接在带标签语法中完成。

直觉主义模态逻辑。直觉主义模态逻辑 IS4 是将直觉主义命题逻辑与模态逻辑 S4 相结合的一种方式。它是直觉主义模态逻辑 IK 的扩展，IK 首次于 [8], [24] 中得到研究，并在 [26], 中得到详细考察，其中证明了直觉主义模态 S5 立方体中的大多数逻辑具有可判定性。IS5（又称 MIPQ ）的可判定性此前已由 Mints 证明 [21, §11]，但 IK4 和 IS4 的可判定性仍未解决。事实上，IS4 的可判定性问题隐藏着远比其任一前身复杂得多的内容。由于 IS4 可以在将直觉主义 Kripke 框架的预序关系与模态 Kripke 框架的可达关系结合起来的 Kripke 框架上进行解释，基于相应序列演算的判定过程中众所周知的循环问题，会因模态算子与直觉主义蕴涵之间的相互作用而加剧。

贡献。本文证明了 IS4 是可判定的，并猜想同样的方法也可用于证明 IK4 的可判定性。我们给出一个构造性判定程序：给定一个公式，该程序要么生成一个证明，表明该公式有效；要么生成一个使该公式为假的有限反模型。该程序基于 [20] 中提出的 IS4 全标记证明系统。该系统继承了直觉主义命题逻辑和经典模态逻辑中带标签系统的优点，尤其是所有推理规则都是可逆的，并且序列与模型之间存在直接对应关系。这一选择对于解决方案的形成至关重要；同时，还需要对证明搜索和循环检查进行复杂的组织，才能实现完整的判定程序。

相关工作。人们已经探索了多种策略来

IS4 和相关逻辑的可判定性。 [10] 中通过基于证明搜索的程序，证明了多种直觉主义多模态逻辑的仅含2的片段是可判定的。

此外，对于直觉主义模态逻辑的仅含2的片段，研究者通过将其嵌入经典双模态逻辑（定义为正规单模态逻辑的融合） [29]，得到了可判定性结果。逻辑 IS4 与一种经典双模态逻辑密切相关，后者定义为乘积 S4 × S4 [8]；二者都相对于具有两个构成“网格”的预序关系的 Kripke 模型可靠且完备。有趣的是，Gabelaia等人 [9]证明，逻辑 S4 × S4 不可判定。这一结果表明，模态逻辑中很容易出现不可判定性；然而， [9] 中的证明策略无法扩展到 IS4，因为其模型必须满足关于其中一个预序关系的单调性。类似地，研究者曾利用将直觉主义模态逻辑嵌入可判定的一元二阶逻辑片段的方法来证明其可判定性 [1]，；但正如作者所解释的，这种方法无法应用于 IS4。 [12] 中用于证明某些直觉主义模态逻辑可判定性的另一种常见方法是语义过滤，但直觉主义 S5 立方体中这些逻辑的特性迄今仍使该方法无法应用于它们。另一方面，构造性 S4（模态逻辑 S4 的另一种直觉主义变体）已在 [13], 中被证明是可判定的，其有限模型性质也已在 [2]中得到证明。

组织。第 2 节回顾 IS4 的句法和语义。第3节解释为其设计判定程序的困难。第 4 节介绍我们解决方案的第一个要素：基于完全标记序列的 IS4 证明系统 labIS4$_{\leq}$ 。第 5 节描述如何从这样的序列式中读取模型。第 6 节确立证明搜索过程中在 labIS4$_{\leq}$ 中出现的序列式的有用性质。第 7 节发展出介绍基于修改后的证明搜索算法的判定程序所必需的概念。第 8 节演示如何从失败的证明搜索中提取反模型。第 9 节展示算法成功时如何重构真实证明。第 10 节证明该算法总会终止。最后，第 11 节给出我们的结论。所有证明以及一些详细示例都可以在附录中找到。

2. P预备知识：逻辑IS4

本文用大写字母表示公式 A， B， C，.....这些公式由一个可数集合 $\mathcal{A}$ 中的原子命题（用小写字母表示 a， b， c，.....）构成，形式如下：

$$A ::= \bot \mid a \mid (A \wedge A) \mid (A \vee A) \mid (A \supset A) \mid \Box A \mid \Diamond A$$

直觉主义模态逻辑 K，简称 IK，是通过在直觉主义命题逻辑的句法中加入两个模态算子 2 和 3（通常分别表示必然性和可能性），并添加 k-公理而得到的

$$\begin{array}{ll} \mathsf{k}_1\colon \Box(A \supset B) \supset (\Box A \supset \Box B) & \mathsf{k}_3\colon \Diamond(A \vee B) \supset (\Diamond A \vee \Diamond B) \\ \mathsf{k}_2\colon \Box(A \supset B) \supset (\Diamond A \supset \Diamond B) & \mathsf{k}_4\colon (\Diamond A \supset \Box B) \supset \Box(A \supset B) \\ & \mathsf{k}_5\colon \quad \Diamond\bot \supset \bot \end{array} \tag{1}$$

一个公式是 IK 的定理，当且仅当它可以通过必然化规则和肯定前件式从这个集合中推导出来：

$$\mathsf{nec}\ \frac{A}{\Box A} \qquad \text{and} \qquad \mathsf{mp}\ \frac{A \qquad A \supset B}{B}$$

直觉主义模态逻辑 S4，简称 IS4，是通过在 IK 中添加公理得到的

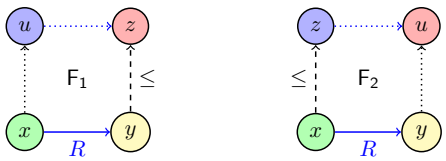
$$\begin{array}{l} \mathsf{4}\colon (\Diamond\Diamond A \supset \Diamond A) \wedge (\Box A \supset \Box\Box A) \\ \mathsf{t}\colon (A \supset \Diamond A) \wedge (\Box A \supset A) \end{array} \tag{2}$$

请注意，在经典情形下，(1) 中的公理 k_2–k_5 可由 k_1, 推出；但由于缺乏德摩根对偶性，在直觉主义逻辑中情况并非如此。同样，在 (2) 中两个合取项都是必需的，因为与经典情形不同，它们不能相互推出。

现在让我们回顾直觉主义模态逻辑的双关系模型 [7], [24] ，它结合了直觉主义命题逻辑的 Kripke 语义与经典模态逻辑。

定义 2.1. 双关系框架 F 是一个三元组 $\langle W, R, \leq\rangle$ ，其中 W 是一个非空世界集合，配备可达关系 R 和预序 $\leq$（即自反且传递的关系），并满足：

(F_1) 对所有 x、y、$z \in W$，若 xRy 且 $y \leq z$，则存在 $u \in W$ 使得 $x \leq u$ 且 uRz （见左下图）；(F_2) 对所有 x、y、$z \in W$，若 $x \leq z$ 且 xRy，则存在 $u \in W$ 使得 zRu 且 $y \leq u$ （见右下图）。



定义 2.2。双关系模型 M 是一个四元组 $\langle W, R, \leq, V\rangle$ ，其中 $\langle W, R, \leq\rangle$ 是一个双关系框架，且V: $W \to 2^{\mathcal{A}}$ 是一个赋值函数，即一个将每个世界 w 映射到在 w为真的命题原子子集的函数，此外还满足单调性条件：如果 $w \leq w'$，则 $V(w) \subseteq V(w')$。

记 $m\ w \Vdash a$ 当且仅当 $a \in V(w)$ ，并按照直觉主义和模态克里普克模型的规则递归地将关系 $\Vdash$ 扩展到所有公式： $m\ w \nVdash \bot$;

M $w \Vdash A \wedge B$ 当且仅当 M $w \Vdash A$ 且 M $w \Vdash B$;
M $w \Vdash A \vee B$ 当且仅当 M $w \Vdash A$ 或 M $w \Vdash B$;
M $w \Vdash A \supset B$ 当且仅当对于所有满足 $w \leq w'$ 的 w' ，若 M $w' \Vdash A$，则 M $w' \Vdash B$; M $w \Vdash \Box A$ 当且仅当对于所有满足 $w \leq w'$和 $w'Ru$的 w' 和 u ，我们有 M $u \Vdash A$; M $w \Vdash \Diamond A$ 当且仅当存在满足以下条件的 u : wRu 且 M $u \Vdash A$.

由赋值函数 V 的单调性，我们得到关系 $\Vdash$ 的单调性：

命题 2.3。 （单调性）对于任意公式 A 以及任意 w, $w' \in W$，如果 $w \leq w'$ 且 $m\ w \Vdash A$，那么 $m\ w' \Vdash A$。

定义 2.4（有效性）。公式 A 在模型 $\mathrm{M} = \langle W、R、\leq、V\rangle$ 中有效，当且仅当 M $w \Vdash A$ 对所有 $w \in W$ 均成立。公式 A 在框架 $\mathrm{F} = \langle W、R、\leq\rangle$ 中有效，当且仅当它在 $\langle W、R、\leq、V\rangle$ 中对所有赋值 V 均有效。

IS4 的句法与语义之间的对应关系可以表述如下：

定理 2.5（完备性 [8], [24]）。公式 A 是 IS4 的定理，当且仅当 A 在每个双关系框架 $\langle W、R、\leq\rangle$ 中均有效，其中 R 是自反且传递的。

3. 为什么IS4的可判定性如此困难?

本节将重点介绍我们在处理 IS4 的可判定性问题时遇到的主要困难，并在此过程中简要说明我们方法中的关键要素。

证明一个逻辑具有可判定性的一种方法，是在可靠且完备的演绎系统中进行证明搜索，旨在通过证明搜索找到证明，或从失败的证明搜索中构造一个反模型。

对于 IS4，存在若干这样的演绎系统，首先是 Simpson 的带标签系统 [26]。此外，还有各种类型的嵌套序列系统：单结论 [27],多结论 [16]，以及聚焦变体 [4]。一个自然的问题是，为什么这些系统中没有任何一个被用于证明 IS4 的可判定性。

需要更多标记。上述系统采用了一种我们可以称为混合的方法：它们将模态可达关系 R 内化到序列语法中，通过标签和关系原子或嵌套来实现；但对于逻辑的直觉主义方面，它们依赖传统的结构化方法，例如使用单结论序列（至少在某些规则中）。有人可能会认为，将直觉主义部分的传统循环检查与模态部分基于标签的循环检查结合起来，就能得到一种判定程序。但情况更为复杂，因为

1）经典 S4 循环测试用于查找沿 R-关系的重复，因此无法应用于序列的右侧，因为结论公式有时会被替换为新的公式；2）直觉主义系统采用结构化方法，也意味着并非所有规则都可逆，而且该程序需要回溯，因此模态的循环检查也需要与必要的回溯相结合。

这两个问题都可以通过使用一种同时纳入关系 R 和 $\leq$ [19], [20] 的全标记证明系统来解决。这与引言中提到的从结构化方法转向直觉主义命题逻辑的标记方法具有相同的优势。该系统不仅重新建立了序列与模型之间的密切联系，这种联系在经典模态逻辑中已广为人知，还使我们能够将系统中的所有规则都变为可逆的。此外，序列语法中的显式关系原子使循环检查易于实现，也使我们能够在构造反模型时显式表示回边。

回溯/终止性权衡。朴素证明搜索不是终止的，其不终止可能有两个来源：

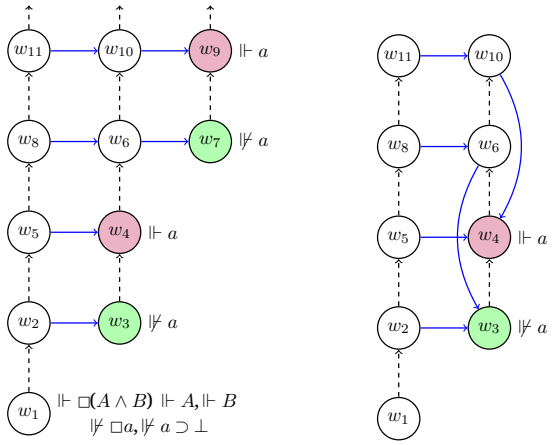


图 1。左：潜在的非终止问题示意图。右：不受限制地识别节点时违反条件 F_1 的示意图。

第一个来源继承自经典模态逻辑 S4，第二个来源则来自直觉主义命题逻辑。在这两种情形中，问题都可以分别通过简单的循环检查得到解决。然而，将二者结合起来并不直接，如下面的例子所示：

例 3.1. 考虑以下公式，它在 IS4 中不可证明：

$$\Box\Big((\Box a \supset \bot) \wedge \big((a \supset \bot) \supset \bot\big)\Big) \supset \bot. \tag{3}$$

设 $A = \Box a \supset \bot$ and $B = (a \supset \bot) \supset \bot$。为了构造一个反模型 M，我们需要一个满足 $2(A \wedge B)$ 的世界 w_1，因此它也满足 A 和 B。由此，所有满足 $w_1 \leq w$ 且对某个 w 有 wRw' 的世界 w' 都应满足这些公式。于是，所有这些世界都不能满足 $\Box a$，也不能满足 $a \supset \bot$。后者意味着，对每个这样的世界 w'，都必须存在一个世界 v，使得 $w' \leq v$ 且 M $v \Vdash a$。当然，反过来，v 也不能满足 $\Box a$，因此必须存在世界 u 和 u'，使得 $v \leq u'$、$u' R u$，且 M $u \nVdash a$。但这个 u 不能满足 $a \supset \bot$，所以必须存在一个世界 v_1，使得 $u \leq v_1$ 且 M $v_1 \Vdash a$，如此等等。因此，通过证明搜索来构造反模型的朴素实现会不断添加世界，直至无穷，因为两种循环检查都无法检测到重复（见图 1 左侧）。换句话说，

3）为了解释模态算子与直觉主义蕴含之间的相互作用，两个循环检查必须彼此协调。

证明搜索与循环的交错。为了解决第三个问题，我们必须实现一个涉及两种关系的更复杂的循环检查。这直接引出了第四个问题。

例 3.2。 假设我们设计了一个合适的循环检查，使得可以在图 1（左）所示结构的阶段停止证明搜索，并且我们确定了 w_7 到 w_3 以及 w_9 到 w_4。这将在 w_{10} 和 w_4 之间创建“回链”，以及

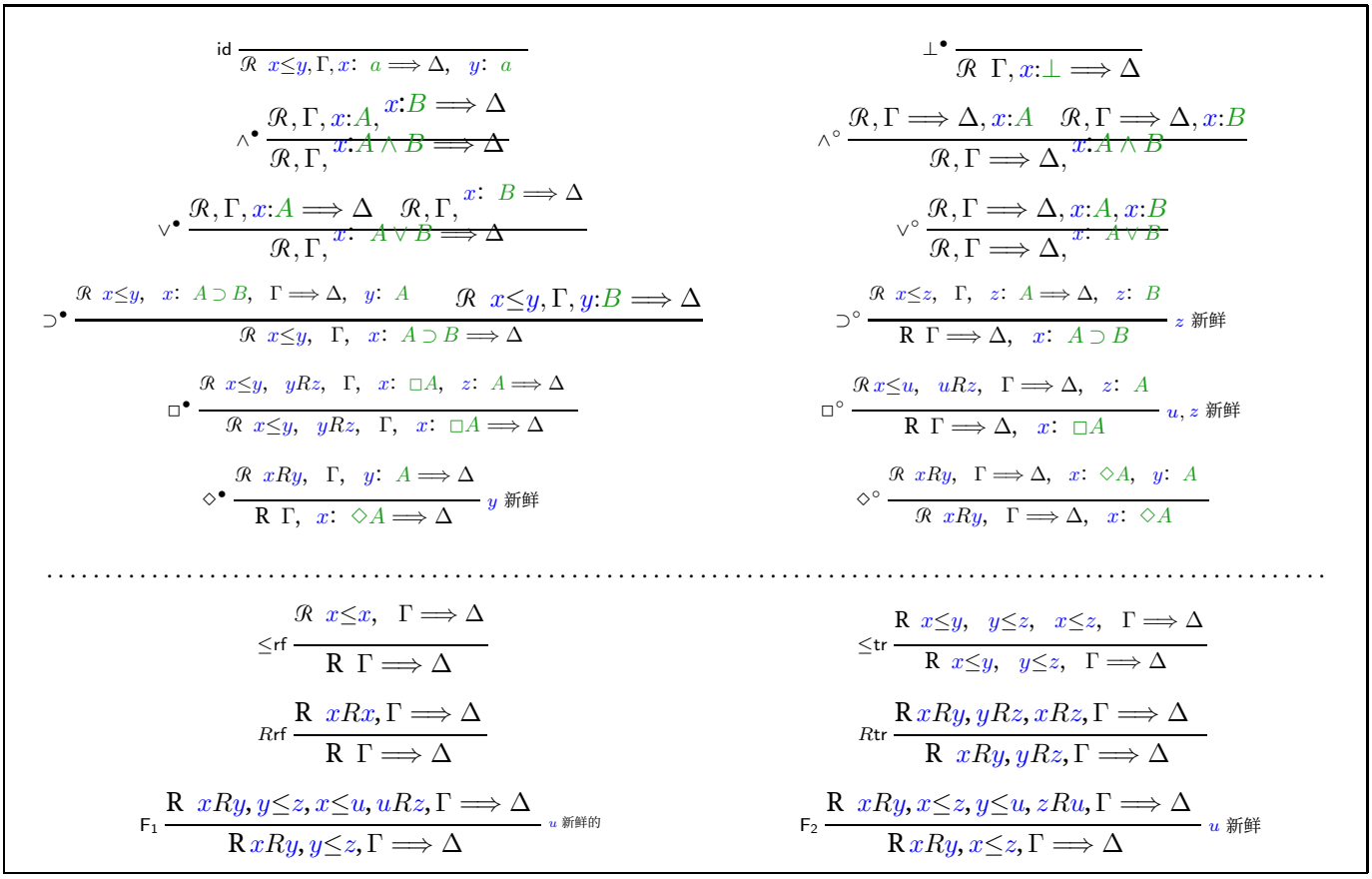


图2. 系统 labIS4$_{\leq}$

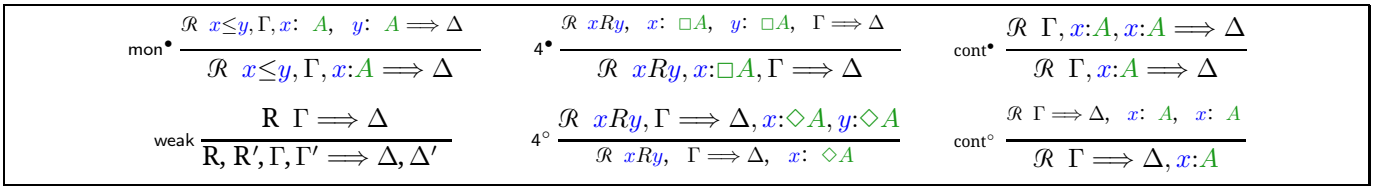


图3。可容许规则

在 w_6 和 w_3之间。然而，这会导致违反 F_1 ，因为现在 $w_{10}Rw_4{\leq}w_6$ ，但不存在 w' 使得$w_{10}{\leq}w'Rw_6$。（见图 1，右）。这意味着

4）在双关系模型的语境下，通过识别形成循环的标签/世界，从失败的证明搜索中构造（有限）反模型的标准方法失效。我们破坏了 F_1/F_2 性质，进而不得不添加新的世界，这意味着我们必须继续进行证明搜索。

我们通过在证明搜索完成之后以及进行过程中识别（替换）标签来解决这一问题。这保留了不可证性，但先验地说，可能并不可靠。这意味着，当在一个非公理性的序列上终止一个分支时，仍然可以从中提取出一个反模型。然而，当最终只到达公理叶节点时，仍需证明能够从该证明尝试（可能包含标签识别）中获得一个可靠的证明。因此，我们不再先进行朴素证明搜索，然后通过“折叠”失败的序列，从失败的证明搜索中构造反模型，而是在证明搜索过程中就执行折叠；此时该过程已成为反模型搜索。随后，再通过“展开”搜索树，从失败的反模型搜索中构造一个适当的证明。为确保终止而进行的循环检查必须经过精细校准，以完成最后一步，即将证明尝试展开为真实证明。

4. 预备知识：完全标记序列

本节介绍我们所使用的直觉主义模态逻辑labIS4$_{\leq}$ [20]的完全标记序列证明系统，以证明IS4的可判定性。出发点是带标签公式的概念，它是一个二元组$x\colon A$ 由一个标签 x和一个公式 A组成。一个关系原子要么是表达式xRy ，要么是 $x{\leq}y$ ，其中 x 和 y 都是标签。一个（带标签）序列 $\mathfrak{G}$是一个三元组$\mathcal{R}\ \Gamma \Longrightarrow \Delta$ ，其中$\mathcal{R}$ 是关系原子的集合，而 Γ 和 Δ 是带标签公式的多重集，均写成

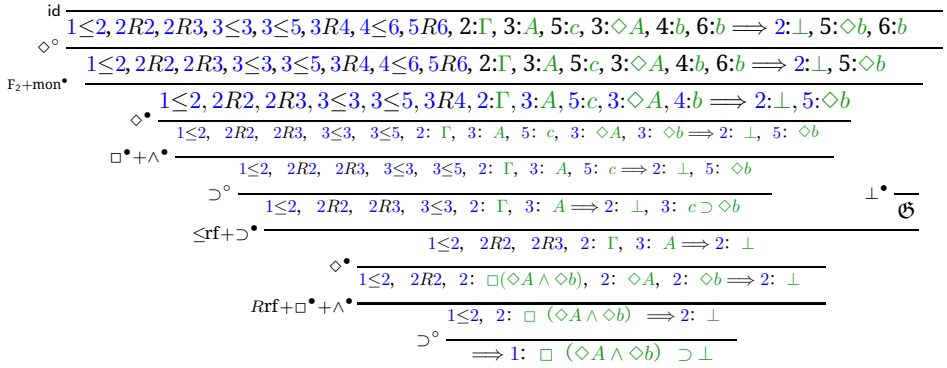


图 4。左：在 labIS4$_{\leq}$ 中，$\Box(\Diamond A \wedge \Diamond b) \supset \bot$ 其中 $A = (c \supset \Diamond b) \supset \bot$，$\Gamma = \{\Box(\Diamond A \wedge \Diamond b)\ \Diamond b\}$ 以及 $\mathfrak{G} = 1{\leq}2,\ 2R2,\ 2R3,\ 3{\leq}3,\ 2{:}\ \Gamma,\ 3{:}\ \bot \Longrightarrow 2{:}\ \bot$。右：推导中左上方序列的图示表示。

逗号分隔的列表。图 2 展示了 labIS 的推理规则4$_{\leq}$。[1] 图 3 展示了一些附加规则，我们将 labIS4$'_{\leq}$ 定义为：=labIS4$_{\leq}$ ∪{mon•、弱化规则 4•、 4°、cont•、cont°}。

推导是一棵有限树，其中每条边都是一个序列，每个内部节点都是一条规则。当且仅当每个叶节点都是无前提的规则时，该推导才是根处序列的证明，即 $\bot^{\bullet}$ or id。如 [20], labIS4$_{\leq}$ 所示，它对 IS4 是可靠且完备的。

定理 4.1 ([20])。 公式 A 是 IS4 中的定理，当且仅当对于每个 x，序列 $\Longrightarrow x{:}\ A$ 可在 labIS4$_{\leq}$中导出。

例 4.2。 在图 4 的左侧，我们给出了一个在labIS4$_{\leq}$ 中关于有效公式 $F = \Box(\Diamond A \wedge \Diamond b) \supset \bot$的证明，其中$A = (c \supset \Diamond b) \supset \bot$；由于篇幅所限，有时会将若干规则压缩为一个步骤。

推论 4.3。规则 mon•、弱化规则、 4•、 4°、cont• 和 cont° 对于 labIS4$_{\leq}$ 是可容许的。因此，公式 A 在 labIS4$'_{\leq}$ 中可导出，当且仅当它在 labIS4$_{\leq}$ 中可导出。

5. 从序列到模型

本节确立序列与模型之间所承诺的对应关系。

记法 5.1。设 $\mathfrak{G}$ 是一个序列 R $\Gamma \Longrightarrow \Delta$。我们记作

- $x{\leq_{\mathfrak{G}}}y$ 当且仅当 $x{\leq}y$ 出现在 R 中；
- $xR_{\mathfrak{G}}y$ 当且仅当 xRy 出现在 R 中；
- $\mathfrak{G}$，$x{:}\ A^{\bullet}$ 当且仅当 $x{:}\ A$ 出现在 Γ(此时我们也说 $A^{\bullet}$ 出现在 x 中 $\mathfrak{G}$)；
- $\mathfrak{G}$，$x{:}\ A^{\circ}$ 当且仅当 $x{:}\ A$ 出现在 Δ(此时我们也说 A° 出现在 x 中 $\mathfrak{G}$)；
- $\ell(\mathfrak{G})$表示出现在 $\mathfrak{G}$中的标签集合。

本文通篇广泛使用这一记号。[2]

例 5.2。这种记法还使我们能够以图形方式表示序列式，如图 4 右侧所示，其中描绘了证明中展示的左上角序列式；该证明展示在图的左侧。该序列式用一个有向图表示，其节点是序列式的标签（为方便起见，我们将使用自然数），虚线边表示 $\leq$- 关系，实线边表示 R- 关系。现在，我们只需在标签 $A^{\bullet}$ (A°旁写上公式 w，以表示带标签公式 $w{:}\ A$ 出现在 $\Longrightarrow$的左侧（右侧）。

定义 5.3 （快乐带标签公式）。 公式 $A^{\bullet}$ （A°）在序列 $\mathfrak{G}$中的标签 x 处，或 $\mathfrak{G}$，$x{:}\ A^{\bullet}$ （$\mathfrak{G}$，$x{:}\ A^{\circ}$）简写为快乐，当且仅当满足以下条件：

- $\mathfrak{G}$，$x{:}\ a^{\bullet}$ 始终是快乐的；
- $\mathfrak{G}$，$x{:}\ a^{\circ}$ 是快乐的，当且仅当不存在 $\mathfrak{G}$，$x{:}\ a^{\bullet}$；
- $\mathfrak{G}$，$x{:}\ \bot^{\bullet}$ 从来不是快乐的；
- $\mathfrak{G}$，$x{:}\ \bot^{\circ}$ 始终开心；
- $\mathfrak{G}$，$x{:}\ A \wedge B^{\bullet}$ 开心当且仅当 $\mathfrak{G}$，$x{:}\ A^{\bullet}$ 且 $\mathfrak{G}$，$x{:}\ B^{\bullet}$；
- $\mathfrak{G}$，$x{:}\ A \wedge B^{\circ}$ 开心当且仅当 $\mathfrak{G}$，$x{:}\ A^{\circ}$ 或 $\mathfrak{G}$，$x{:}\ B^{\circ}$；
- $\mathfrak{G}$，$x{:}\ A \vee B^{\bullet}$ 开心当且仅当 $\mathfrak{G}$，$x{:}\ A^{\bullet}$ 或 $\mathfrak{G}$，$x{:}\ B^{\bullet}$；
- $\mathfrak{G}$，$x{:}\ A \vee B^{\circ}$ 开心当且仅当 $\mathfrak{G}$，$x{:}\ A^{\circ}$ 且 $\mathfrak{G}$，$x{:}\ B^{\circ}$；
- $\mathfrak{G}$，$x{:}\ A \supset B^{\bullet}$ 开心当且仅当 $\mathfrak{G}$，$x{:}\ A^{\circ}$ 或 $\mathfrak{G}$，$x{:}\ B^{\bullet}$；
- $\mathfrak{G}$，$x{:}\ A \supset B^{\circ}$ 满足当且仅当$\mathfrak{G}$，$y{:}\ A^{\bullet}$ 并且 $\mathfrak{G}$，$y{:}\ B^{\circ}$ 对于某个 y 使得 $x{\leq_{\mathfrak{G}}}y$；
- $\mathfrak{G}$，$x{:}\ \Box A^{\bullet}$ 满足当且仅当$\mathfrak{G}$，$z{:}\ A^{\bullet}$ 并且 $\mathfrak{G}$，$z{:}\ \Box A^{\bullet}$ 对于所有 z 满足 $xR_{\mathfrak{G}}z$；
- $\mathfrak{G}$，$x{:}\ \Box A^{\circ}$ 满足当且仅当$\mathfrak{G}$，$z{:}\ A^{\circ}$ 对于某个 y，使得 z 满足 $x{\leq_{\mathfrak{G}}}y$ 并且 $yR_{\mathfrak{G}}z$；
- $\mathfrak{G}$，$x{:}\ \Diamond A^{\bullet}$ 满足当且仅当 $\mathfrak{G}$，$y{:}\ A^{\bullet}$ 对于某个 y 使得 $xR_{\mathfrak{G}}y$；
- $\mathfrak{G}$，$x{:}\ \Diamond A^{\circ}$ 满足当且仅当$\mathfrak{G}$，$y{:}\ A^{\circ}$ 并且 $\mathfrak{G}$，$y{:}\ \Diamond A^{\circ}$ 对于所有 y 满足 $xR_{\mathfrak{G}}y$。

否则，该公式不满足。

定义 5.4（快乐标签）。在序列 $\mathfrak{G}$ 中出现的标签 x 是快乐的，当且仅当在 $\mathfrak{G}$ 中 x 处出现的所有公式都是快乐的。标签 x 几乎快乐，当且仅当在 x 处出现的所有公式都是快乐的，但形状为 $\bot^{\bullet}$、 a°、 $A \supset B^{\circ}$ 和 $\Box A^{\circ}$ 的公式可能例外。标签 x 朴素满足，当且仅当在 x 处出现的所有公式都是快乐的，但形状为 $\bot^{\bullet}$、 $\Diamond A^{\bullet}$、a°、 $A \supset B^{\circ}$ 和 $\Box A^{\circ}$ 的公式可能例外。

定义 5.5（结构饱和序列）。序列 $\mathfrak{G}$是结构饱和的，当且仅当满足以下条件：

[1]请注意，labIS4$_{\leq}$ 没有切规则，这意味着切是可容许的。关于切消去的句法证明，参见 [20]。[2]以这种方式使用 • 和 ◦ 可以追溯到 Lamarche [18]。

（mon•）如果 $x \leq_{\mathfrak{G}} y$ 且 $\mathfrak{G}$，x: $C^{\bullet}$，则 $\mathfrak{G}$，y: $C^{\bullet}$；（F_1）如果 $xR_{\mathfrak{G}}y$ 且 $y \leq_{\mathfrak{G}} z$，则存在 u 使得 $x \leq_{\mathfrak{G}} u$且 $uR_{\mathfrak{G}}z$；（F_2）如果 $xR_{\mathfrak{G}}y$ 且 $x \leq_{\mathfrak{G}} z$，则存在 u 使得 $y \leq_{\mathfrak{G}} u$且 $zR_{\mathfrak{G}}u$；（≤tr）如果$x \leq_{\mathfrak{G}} y$ 且 $y \leq_{\mathfrak{G}} z$，则 $x \leq_{\mathfrak{G}} z$；（≤rf）对于出现在 $\mathfrak{G}$中的所有 x，$x \leq_{\mathfrak{G}} x$；（Rtr）如果 $xR_{\mathfrak{G}}y$ 且 $yR_{\mathfrak{G}}z$，则 $xR_{\mathfrak{G}}z$；（Rrf）对于出现在 $\mathfrak{G}$中的所有 x，$xR_{\mathfrak{G}}x$。

定义 5.6（快乐序列）。当且仅当一个序列 $\mathfrak{G}$ 是结构饱和的，且该序列中的所有标签都是快乐的，它才是快乐的。

定义 5.7（序列模型）。设 $\mathfrak{G}$ 为一个序列。我们将模型 $\mathbf{M}_{\mathfrak{G}}$ 称为 $\mathfrak{G}$ 的四元组 $\mathbf{M}_{\mathfrak{G}} = \langle \ell(\mathfrak{G}) \leq_{\mathfrak{G}}, R_{\mathfrak{G}}, V \rangle$，其中 $\ell(\mathfrak{G}) \leq_{\mathfrak{G}}, R_{\mathfrak{G}}$ 如记法 5.1 中所定义，而 V 是函数 V: $\ell(\mathfrak{G}) \to 2^{\mathcal{A}}$，定义为对所有原子命题 $a \in \mathsf{A}$，都有 $a \in V(w)$ 当且仅当 $\mathfrak{G}, w{:}a^{\bullet}$。

定理 5.8（完备性）。对于一个快乐序列 $\mathfrak{G}$，其模型 $\mathbf{M}_{\mathfrak{G}} = \langle \ell(\mathfrak{G}) \leq_{\mathfrak{G}}, R_{\mathfrak{G}}, V \rangle$ 是一个传递且自反的双关系模型，并具有以下两个性质：

- 若 $\mathfrak{G}$，x: $A^{\bullet}$，则 $m_{\mathfrak{G}}$，$x \Vdash A$；
- 若 $\mathfrak{G}$，x: A°，则 $m_{\mathfrak{G}}$，$x \nVdash A$。

定义 5.9（公理序列）。序列 $\mathfrak{G}$ 称为公理性的，当且仅当存在一个标签x 满足以下条件之一：$\mathfrak{G}$，x: $a^{\bullet}$ 且 $\mathfrak{G}$，x: a° 对于某个 a；或者 $\mathfrak{G}$，x: $\bot^{\bullet}$。否则，$\mathfrak{G}$ 称为非公理。

注记 5.10。公理序列 $\mathfrak{G}$ 永远不会满足，因为 $\mathfrak{G}$，x: a° 或 $\mathfrak{G}$，x: $\bot^{\bullet}$ 不满足。

6. 层、簇与饱和序列

通过在labIS4$_{\leq}$ 中进行证明搜索而得到的所有 $\Longrightarrow x$: A 推导都具有一种特殊结构：它们的标签被划分为若干层，每一层都具有树结构。这在我们的证明搜索算法中发挥着重要作用。

定义 6.1（层）。对于序列 $\mathfrak{G}$，我们将关系 $R_{\mathfrak{G}}^{\leftrightarrows}$定义为 $R_{\mathfrak{G}} \cup R_{\mathfrak{G}}^{-1}$ 的传递自反闭包。由于这是一个等价关系，我们可以将 $\mathfrak{G}$ 中的一层 L 定义为 $R_{\mathfrak{G}}^{\leftrightarrows}$的一个等价类。

定义 6.2 (分层序列)。我们称序列 $\mathfrak{G}$ 是分层的，当且仅当对于其中出现的任意标签 x、x'、y和 y'：$\mathfrak{G}$:

1) 若 $xR_{\mathfrak{G}}^{\leftrightarrows}y$ 对于 $x \neq y$成立，则 $x \not\leq_{\mathfrak{G}} y$ 和 $y \not\leq_{\mathfrak{G}} x$成立；并且

2) 若 $xR_{\mathfrak{G}}^{\leftrightarrows}y$、$x'R_{\mathfrak{G}}^{\leftrightarrows}y'$和 $x \leq_{\mathfrak{G}} x'$ 对于 $x \neq x'$成立，则 $y' \not\leq_{\mathfrak{G}} y$ 成立。对于层 L_1 和 L_2，当存在标签 $x \in L_1$ 和 $y \in L_2$ 使得 $x \leq_{\mathfrak{G}} y$ 成立时，我们定义 $L_1 \leq L_2$。我们用 $L_1 < L_2$表示 $L_1 \leq L_2$ 和 $L_1 \neq L_2$均成立。

命题 6.3. 对于分层且结构饱和的序列 $\mathfrak{G}$，关系 $\leq$ 是其各层上的序关系。

定义 6.2 中的第二个条件仅用于在命题 6.3 中确立反对称性。如果标签上的 $\leq_{\mathfrak{G}}$ 已经具有反对称性，那么第二个条件可由第一个条件推出。在我们这里讨论的所有序列中，层上的序关系 $\leq$ 定义了一个树结构。

定义 6.4（树分层序列）。当且仅当满足以下条件时，分层序列是树分层的：(i) 存在一个层 L_0，使得对于所有层 $L_0 \leq L$ L，以及 (ii) 对于所有层 L、L'和 L''，只要 $L' \leq L$ 且 $L'' \leq L$，那么 $L' \leq L''$ 或 $L'' \leq L'$。

定义 6.5（顶层与内层）。分层序列 $\mathfrak{G}$ 中的层 L 称为顶层，当且仅当它关于 $\leq$是极大的；也就是说，对于某个层 L'，若 $L \leq L'$ 成立，则 $L = L'$成立。否则，层 L 称为内部层。

注意，在相对于 $\leq$ 的树分层结构饱和序列中，最顶层的层是树结构的叶节点。

在由 labIS4$_{\leq}$ 构造的序列 $\mathfrak{G}$ 中，每一层相对于 $R_{\mathfrak{G}}$ 也具有树结构。但是，如前所述，为了同时搜索证明和反模型，我们需要弱化层上的这一树结构。

定义 6.6（簇）。如果 $\mathfrak{G}$ 是结构饱和的，那么 $R_{\mathfrak{G}} \cap R_{\mathfrak{G}}^{-1}$ 是一个等价关系，并且我们可以将 $\mathfrak{G}$ 中的一个簇 C 定义为 $R_{\mathfrak{G}} \cap R_{\mathfrak{G}}^{-1}$ 的一个等价类 $C = \{x_1, \ldots, x_n\}$。只包含一个标签的簇 $C = \{x_1\}$ 称为单例。在簇上，我们定义两个二元关系：

- $C_1 \leq_{\mathfrak{G}} C_2$ 当且仅当对于所有 $y \in C_2$，存在满足 $x \leq_{\mathfrak{G}} y$ 的 $x \in C_1$。
- $C_1 R_{\mathfrak{G}} C_2$ 当且仅当存在满足 $xR_{\mathfrak{G}}y$ 的 $x \in C_1$ 和 $y \in C_2$。

有时我们会滥用记号：即使 $\{x\}$ 不是一个簇，也将其中一个簇替换为标签 x。不过，此时定义仍按原样应用于 $\{x\}$。

命题 6.7。对于结构饱和序列 $\mathfrak{G}$，关系 $R_{\mathfrak{G}}$ 是一个序，而 $\leq_{\mathfrak{G}}$ 是其簇上的预序。若 $\mathfrak{G}$ 是分层的，则 $\leq_{\mathfrak{G}}$ 也是一个序。

定义 6.8（树簇序列）。若一个结构饱和序列 $\mathfrak{G}$ 满足以下条件，则称其为树簇序列：(i) 对任意簇 C' 和 C''，都存在一个簇 C，使得 $CR_{\mathfrak{G}}C'$ 和 $CR_{\mathfrak{G}}C''$；(ii) 对所有簇 C、C' 和 C''，只要 $C'R_{\mathfrak{G}}C$ 和 $C''R_{\mathfrak{G}}C$，则 $C'R_{\mathfrak{G}}C''$ 或 $C''R_{\mathfrak{G}}C'$。

定义 6.9（快乐层）。若层 L 的所有标签分别都是快乐、几乎快乐或朴素满足的，则称层 L 分别为快乐层、几乎快乐层或朴素满足层。

注 6.10。注意，定义 5.3 中 $A \supset B^{\bullet}$ 和 $\Box A^{\bullet}$ 的快乐标记公式定义，与相应联结词的强制条件不同。这种不匹配是有意为之：缺失的条件改由 $\leq$ 外包给 (mon•)-结构饱和（定义 5.5）来处理。因此，幸福性对层而言几乎是一种局部性质：如果在某个快乐层之外修改序列（添加带标签公式或关系原子），那么只需重新检查 $A \supset B^{\circ}$ 和 $\Box A^{\circ}$ 的幸福性。我们在下文稳定序列式的概念中利用这一观察；其中内部层绝不会被修改，并且始终保持快乐。

定义 6.11（稳定且饱和的序列）。序列 $\mathfrak{G}$是稳定的，当且仅当它是树分层的、树簇化的，并且其所有内部层都几乎快乐。序列 $\mathfrak{G}$ 是饱和的，当且仅当它是稳定的，并且其所有最顶层都几乎快乐。当它是稳定的且其所有最顶层

都朴素满足时，称其为半饱和。若一个序列 $\mathcal{G}$ 的集合被称为稳定的/饱和的/半饱和的，当且仅当它是有限的，并且 $\mathcal{G}$ 中的所有元素分别都是稳定的/饱和的/半饱和的。

由于树分层序列是分层的，而树簇序列在结构上是饱和的（事实上，簇的定义只有在这种情况下才有意义），因此所有稳定序列都是分层的，并且在结构上饱和。

7. 搜索算法

我们将在此介绍的算法以稳定序列为处理对象。粗略地说，我们会处理此类序列的最顶层，直到这些层几乎快乐为止。实现方法是先进行半饱和（本质上就是对某一层应用所有不会给序列添加新标签的推理规则），然后选取该层中一个不满足的$x\colon \Diamond A^\bullet$，并使其变为满足（方法是应用 $\Diamond^\bullet$-规则，或者在循环检测时创建一个簇）。之后再次进行半饱和，并继续处理，直到该层中的所有 $x\colon \Diamond A^\bullet$ 都得到满足，因此该层几乎快乐为止。

所有层都几乎快乐后，所得饱和序列中剩下的不满足公式只出现在最顶层，其形式为$x\colon \bot^\bullet$、$x\colon a^\circ$、$x\colon A\supset B^\circ$ 或 $x\colon \Box A^\circ$。对于前两种形式，该序列是公理性的，因此我们可以停止处理。

如果所有不满足公式都位于最顶层，且形如 $x\colon A\supset B^\circ$ 或 $x\colon \Box A^\circ$，我们就通过创建新层来使它们满足。这些新层成为待饱和的最顶层，而原先的最顶层则转变为良好内部层。重复这一过程，直到出现层的重复。为了确保这一过程终止，我们实现了另外两种循环检测机制（不同于用于$x\colon \Diamond A^\bullet$ 公式的循环检测机制）。

在本节余下部分，我们将正式介绍理解该算法细节所需的概念，并在图 7 中展示该算法。

A. 半饱和

我们首先使所有不属于以下形式的公式满足：$x\colon \bot^\bullet$、$x\colon a^\circ$、$x\colon \Diamond A^\bullet$、$x\colon A\supset B^\circ$，或 $x\colon \Box A^\circ$。

定义 7.1 (半饱和)。我们在序列式集合 $\mathcal{G}$ 上定义重写关系 $\mathcal{G}\rightsquigarrow\!\!\!\!\!\cdot\;\mathcal{G}'$，当且仅当存在一个序列 $\mathfrak{G}\in\mathcal{G}$，使得以下某种形式的带标签公式 C 出现在 x 中的 $\mathfrak{G}$，且该公式不满足；并且 $\mathcal{G}'$ 由下述方式得到：根据下面的情形 1)–4)，将集合 $\mathfrak{G}$ 中的序列 $\mathcal{G}$ 替换为序列 $\mathfrak{G}'$；或者根据下面的情形 5)–7)，将其替换为序列 $\mathfrak{G}'$ 和 $\mathfrak{G}''$，如下所示：

1) 对于 $C=A\wedge B^\bullet$，通过将缺失的所有 $x\colon A^\bullet$ 和 $x\colon B^\bullet$ 添加到 $\mathfrak{G}$ 中，得到 $\mathfrak{G}'$；2) 对于 $C=A\vee B^\circ$，通过将缺失的所有 $x\colon A^\circ$ 和 $x\colon B^\circ$ 添加到 $\mathfrak{G}$ 中，得到 $\mathfrak{G}'$；3) 对于 $C=\Box A^\bullet$，每当 $xR_{\mathfrak{G}}z$ 时，通过将缺失的所有 $z\colon A^\bullet$ 和 $z\colon \Box A^\bullet$ 添加到 $\mathfrak{G}$ 中，得到 $\mathfrak{G}'$；4) 对于 $C=\Diamond A^\circ$，每当 $xR_{\mathfrak{G}}y$ 时，通过将缺失的所有 $y\colon A^\circ$ 和 $y\colon \Diamond A^\circ$ 添加到 $\mathfrak{G}$ 中，得到 $\mathfrak{G}'$；

5) 对于 $C=A\vee B^\bullet$，通过将带标签公式 $x\colon A^\bullet$（分别为 $x\colon B^\bullet$）添加到 $\mathfrak{G}$ 中，得到 $\mathfrak{G}'$（分别为 $\mathfrak{G}''$）；6) 对于 $C=A\wedge B^\circ$，通过将带标签公式 $x\colon A^\circ$（分别为 $x\colon B^\circ$）添加到 $\mathfrak{G}$ 中，得到 $\mathfrak{G}'$（分别为 $\mathfrak{G}''$）；7) 对于 $C=A\supset B^\bullet$，通过将带标签公式 $x\colon A^\circ$（分别为 $x\colon B^\bullet$）添加到 $\mathfrak{G}$ 中，得到 $\mathfrak{G}'$（分别为 $\mathfrak{G}''$）。

我们用 $*_{\mathsf{s}}$ 表示 s 的传递自反闭包。如果 $\mathcal{G}\rightsquigarrow\!\!\!\!\!\cdot\,^{*_{\mathsf{s}}}\mathcal{G}'$ 且 $\mathcal{G}'$ 关于 s 处于范式中，即如果 $\mathcal{G}'$ 中所有序列式里出现的所有标签都已朴素满足，那么 $\mathcal{G}'$ 就称为 $\mathcal{G}$ 的半饱和。

之所以使用这一术语，是因为稳定序列式集合的半饱和是半饱和的。

引理 7.2。以下所有陈述均成立：

a) 如果集合 $\mathcal{G}$ 是稳定的且 $\mathcal{G}\rightsquigarrow\!\!\!\!\!\cdot_{\mathsf{s}}\mathcal{G}'$，那么 $\mathcal{G}'$ 是稳定的。b) 在有限集合 $\mathcal{G}$ 上，重写关系 s 是终止的。c) 集合 $\mathcal{G}$ 是半饱和的，当且仅当它是稳定的，并且关于 s 处于范式中。

B. 饱和

在下一步中，我们展示如何处理不满足的$x\colon \Diamond A^\bullet$。这是终止性的第一个来源；为此，我们采用经典 S4 中通常使用的相同方法。然而，在经典 S4 的情形下，循环检测会在当前分支中完成证明搜索，而在这里我们必须继续进行证明搜索。因此，我们通过簇来实现循环。

定义 7.3（等价标签/簇）。分别出现在序列 $\mathfrak{G}$ 和 $\mathfrak{H}$ 中的标签 x 和 y 是等价的，记作 $_{\mathfrak{G}}x\sim{}_{\mathfrak{H}}y$，当且仅当 $\mathfrak{G}, x\colon A^\bullet\iff\mathfrak{H}, y\colon A^\bullet$，并且 $\mathfrak{G}, x\colon A^\circ\iff\mathfrak{H}, y\colon A^\circ$ 对所有公式 A 均成立。如果上下文中 $\mathfrak{G}$ 和 $\mathfrak{H}$ 很明确，我们就简单地写作 $x\sim y$。对于结构饱和序列 $\mathfrak{G}$ 和 $\mathfrak{H}$，我们可以将其推广到簇：$_{\mathfrak{G}}C_1\sim{}_{\mathfrak{H}}C_2$ 当且仅当存在一个双射 $f\colon C_1\to C_2$，使得对所有 $x\in C_1$ 都有 $fx\sim x$。

定义 7.4（无过去）。序列 $\mathfrak{G}$ 中的标签 x 无过去，当且仅当 $y\leq_{\mathfrak{G}}x$ 蕴含对于所有 y 都有 $y=x$。

定义 7.5（饱和）。我们定义序列上的一个二元关系$\rightsquigarrow_\Diamond$如下：$\mathfrak{G}\rightsquigarrow_\Diamond\mathfrak{G}'$ 当且仅当存在一个标签 y 和一个公式A，使得 $\mathfrak{G}, y\colon \Diamond A^\bullet$ 是不满足的，并且所有满足$u\neq y$ 但不满足 $uR_{\mathfrak{G}}y$ 的标签 $yR_{\mathfrak{G}}u$ 几乎快乐，并且以下两者之一成立

1) 选项 1：存在一个标签 $x\neq y$，使得 $xR_{\mathfrak{G}}y$ 和 $x\sim y$，且 $\mathfrak{G}, x\colon \Diamond A^\bullet$ 是满足的；每个带有 $xR_{\mathfrak{G}}u$ 的标签 u 在 $\mathfrak{G}$ 中都没有过去。然后，先将 x 替换 y 的每次出现，再将所得的 $R_{\mathfrak{G}}$ 在传递性下闭包，即可从 $\mathfrak{G}$ 得到 $\mathfrak{G}'$；或者 2) 选项 2：不存在这样的标签 x。然后，先为某个新标签 z 添加 $z\leq z$、yRz、$z\colon A^\bullet$，再将所得的 $R_{\mathfrak{G}}$ 在传递性和自反性下闭包，即可从 $\mathfrak{G}$ 得到 $\mathfrak{G}'$。

对于序列式的集合 $\mathcal{G}$ 和 $\mathcal{G}'$，我们记作 $\mathcal{G}\rightsquigarrow\!\!\!\!\!\cdot_\Diamond\mathcal{G}'$，当且仅当 $\mathcal{G}$ 是半饱和的，$\mathfrak{G}\rightsquigarrow_\Diamond\mathfrak{G}'$ 对某个 $\mathfrak{G}\in\mathcal{G}$ 而言，且 $\mathcal{G}'$ 是一个半饱和

化的 $(\mathfrak{G}\backslash\{\mathfrak{G}\})\cup\{\mathfrak{G}'\}$。[3] 我们将 $\diamond$ 的传递自反闭包记作 $\twoheadrightarrow^*_\diamond$。如果 $\mathfrak{G}\twoheadrightarrow^*_\diamond\mathfrak{G}'$ 且 $\mathfrak{G}'$ 关于 $\diamond$ 处于范式，那么称 $\mathfrak{G}'$ 是 $\mathfrak{G}$ 的一个饱和。

与之前一样，这个术语之所以合理，是因为对半饱和序列集合进行饱和化后，该集合就是饱和的。

引理 7.6. 以下陈述均成立：

a) 若集合 $\mathfrak{G}$ 是半饱和的，且 $\mathfrak{G}\twoheadrightarrow_\diamond\mathfrak{G}'$，则 $\mathfrak{G}'$ 是半饱和的。b) 重写关系 $\diamond$ 是终止的。c) 集合 $\mathfrak{G}$ 是饱和的，当且仅当它是半饱和的，并且关于 $\diamond$ 处于范式。

C. 提升饱和

在讨论了如何对序列集合进行饱和化之后，我们现在将说明如何处理不满足的 $A\supset B^\circ$和 $\Box A^\circ$。为此，我们必须创建新层。最直接的方法是直接复制包含不满足 R 的层的 x结构：$A\supset B^\circ$ 或 $x\colon \Box A^\circ$。然而，由于簇的存在，有时我们需要创建包含不满足公式的簇的两个副本。

定义 7.7（序列式和）。设 $\mathfrak{G}$ 和 $\mathfrak{G}'$ 分别为序列 $\mathrm{R}\ \Gamma\Longrightarrow\Delta$ 和 $\mathrm{R}',\Gamma'\Longrightarrow\Delta'$。我们将它们的和 $\mathfrak{G}+\mathfrak{G}'$ 定义为序列 $\mathrm{R},\mathrm{R}',\Gamma,\Gamma'\Longrightarrow\Delta,\Delta'$。

构造 7.8（层提升 I）。设 $\mathfrak{G}$ 为一个饱和序列，L 为其最顶层之一，并设 $x\in L$。那么 x 属于一个簇 $C_x=\{x_1,\dots,x_h\}$，其中 $h\geq 1$ 且 $x=x_m$，对于某个 $1\leq m\leq h$；我们通常将其简写为 $m=1..h$。特别地，对于任意 $i,i'=1..h$，有 $x_iR_{\mathfrak{G}}x_{i'}$。设 $\{y_1,\dots,y_l\}=L\setminus C_x$，其中 $l\geq 0$。那么，若 $h=1$，令 $\hat{}L$ 为新标签集合 $\{\hat{}y_1,\dots,\hat{}y_l,\hat{}x\}$。否则，若 $h>1$，令 $\hat{}L:=\{\hat{}y_1,\dots,\hat{}y_l,\hat{}x,\hat{}x'_1,\dots,\hat{}x'_h,\hat{}x''_1,\dots,\hat{}x''_h\}$，其中所有标签同样都是新的。我们将 $\mathfrak{G}{\uparrow^x}$ 定义为由以下部分组成：

$\hat{}$1）对于所有 $v\in L$，关系原子 $v{\leq}v$ 和 vRv；2）对于每个 $i=1..l$、每个 $j=1..h$以及每个 $w\in\ell(\mathfrak{G})$：a）只要 $w{\leq_{\mathfrak{G}}}y_i$，关系原子 $w{\leq}\hat{}y_i$；b）只要 $w{\leq_{\mathfrak{G}}}x$，关系原子 $w{\leq}\hat{}x$；c）仅对于 $h>1$：只要 $w{\leq_{\mathfrak{G}}}x_j$，关系原子 $w{\leq}\hat{}x'_j$ 和 $w{\leq}\hat{}x''_j$；

3）对于所有 i、$i'=1..l$ 和 j、$j'=1..h$，

a）只要 $y_iR_{\mathfrak{G}}y_{i'}$，关系原子 $\hat{}y_iR\hat{}y_{i'}$；b）仅对于 $h>1$：只要 $y_iR_{\mathfrak{G}}x_j$，关系原子 $\hat{}y_iR\hat{}x'_j$ 和 $\hat{}y_iR\hat{}x''_j$；c）只要 $y_iR_{\mathfrak{G}}x$，关系原子 $\hat{}y_iR\hat{}x$；d）仅对于 $h>1$：只要 $x_jR_{\mathfrak{G}}y_i$，关系原子 $\hat{}x'_jR\hat{}y_i$ 和 $\hat{}x''_jR\hat{}y_i$；e）只要 $xR_{\mathfrak{G}}y_i$，关系原子 $\hat{}xR\hat{}y_i$；f）仅对于 $h>1$：关系原子 $\hat{}x'_jR\hat{}x'_{j'}$、$\hat{}x'_jR\hat{}x$、$\hat{}xR\hat{}x''_j$和 $\hat{}x''_jR\hat{}x''_{j'}$。

4) 对于每个 $i=1..l$、每个 $j=1..h$以及每个 C，添加：

a) 带标签公式 $\hat{}y_i\colon C^\bullet$ 当 $\mathfrak{G}$时，$y_i\colon C^\bullet$，

[3]严格来说，在第一种情形中，$\mathfrak{G}'$ 已经是半饱和的，但为了简化表述，我们在两种情形中都进行半饱和处理。

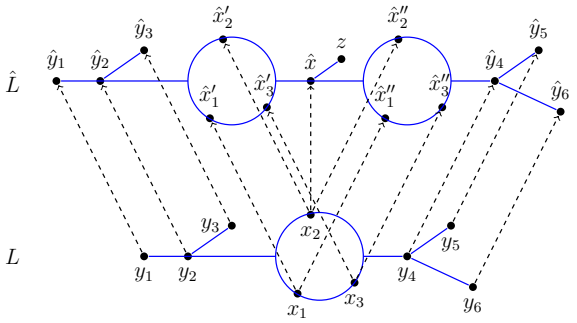


图 5. 构造 7.8 和构造 7.9 中定义的层提升

b) 仅适用于 $h>1$：带标签公式 $\hat{}x'_j\colon C^\bullet$ 以及 $\hat{}x''_j\colon C^\bullet$每当 $\mathfrak{G}$，$x_j\colon C^\bullet$，c) 带标签公式 $\hat{}x\colon C^\bullet$ 每当 $\mathfrak{G}$，$x\colon C^\bullet$。

此构造依据标签 $x\in L$，提升 L 在 $\mathfrak{G}$ 中的层。如果 x 是单元素簇，这便是符合预期的简单提升。否则，如果 x 位于一个非单元素簇中，该簇会被复制一份，而 x 的提升则置于这两个副本之间，如图 5 所示（暂且忽略标签 z）。

在构造 7.8 中，步骤 1)–2) 确保在$\mathfrak{G}+\mathfrak{G}{\uparrow^x}$中，新层确实位于 L 之上，符合预期，并且满足（tr$\leq$）、（rf$\leq$）和（rf$R$）；步骤 3) 确保满足（tr$R$）、（F_1）和（F_2）；步骤 4) 确保满足（mon$^\bullet$）。因此，$\mathfrak{G}+\mathfrak{G}{\uparrow^x}$ 是树簇化的、树分层的，且结构饱和。然而，它可能不稳定，因为 L 现在是一个内部层。由于 $\mathfrak{G}$ 的饱和，它几乎快乐，但可能包含不满足的 $A\supset B^\circ$和 $\Box B^\circ$ 公式。

构造 7.9（层提升 II）。设 $\mathfrak{G}$ 是一个饱和序列，其中 $\mathfrak{G}$, $x\colon F^\circ$ 对于某个标签 x 和某个形如 F的公式 $A\supset B$或 $\Box B$不满足。我们沿用构造 7.8 中的记号（注意，由于 L 是饱和的，$\mathfrak{G}$ 必须是 最顶层。我们定义 $\mathfrak{G}{\uparrow^{x:F}}$ 如下：

- 如果 $F=A\supset B$，则 $\mathfrak{G}{\uparrow^{x\colon A\supset B}}$ 是 $\mathfrak{G}{\uparrow^x}$ 向其中添加带标签公式 $\hat{}x\colon A^\bullet$ 和 $\hat{}x\colon B^\circ$，并称 $\hat{}x$ 为 的 suricata 标签 x

- 如果 $A=\Box B$，则 $\mathfrak{G}{\uparrow^{x\colon \Box B}}$ 是 $\mathfrak{G}{\uparrow^x}$ 向其中添加 zRz，$z{\leq}z$、以及 $z\colon B^\circ$ 对于新标签 z，此外还添加$\hat{}$关系原子 $vR\hat{}z$，只要 $v\in L$ 和 $vR\hat{}x$ 属于 $\mathfrak{G}{\uparrow^x}$。这里z 被称为 的 suricata 标签 x。

一般来说，序列 $\mathfrak{G}+\mathfrak{G}{\uparrow^{x:F}}$恰好包含需要添加到 $\mathfrak{G}+\mathfrak{G}{\uparrow^x}$中的关系原子和带标签公式，从而使 (i) 不满足的 $\mathfrak{G}$，$x\colon F^\circ$ 变得满足（suricata 标签包含导致这种幸福性的白公式），并且 (ii) 结果仍然是结构饱和的（且是树分层的和簇分层的）。图 5 展示了 $x\colon \Box B^\circ$ 的情形，其中添加了新标签 z（该标签没有过去）。为了同时保持稳定这一性质，我们需要为同一层中的所有不满足的 $\mathfrak{G}$，$x\colon F^\circ$ 添加这样的一层 L。

构造 7.10（层提升 III）。设 L 是饱和的非公理序列 $\mathfrak{G}$ 中的最顶层，其中 $x_1\colon F_1^\circ$，……，$x_m\colon F_m^\circ$（形式为 $A\supset B$或 $\Box B$）的公式全部是不满足的

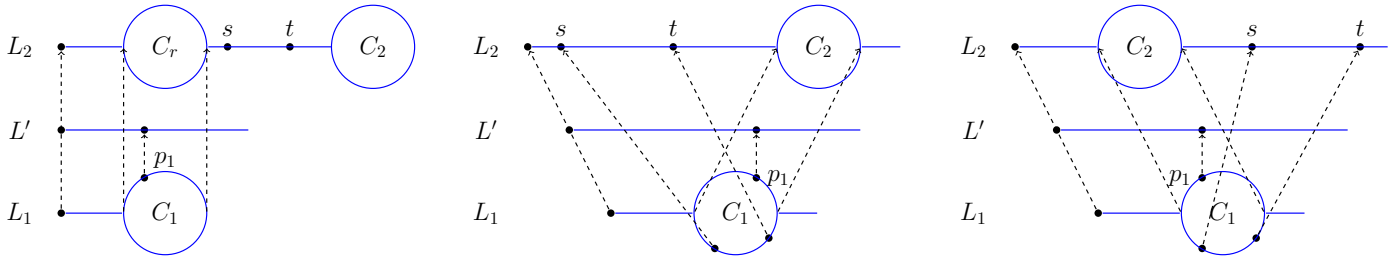


图 6。左：定义 7.16 中定义的不满足的 R-三角循环的结构；中和右：定义7.17 中定义的不满足的 U-三角循环的结构

公式位于 L中。那么 $\mathfrak{G}{\uparrow^{L}} := \mathfrak{G}{\uparrow^{x_1:F_1}} + \ldots + \mathfrak{G}{\uparrow^{x_m:F_m}}$ 其中我们假设，在以下各项中引入的新标签集合$\mathfrak{G}{\uparrow^{x_1:F_1}}$..... $\mathfrak{G}{\uparrow^{x_m:F_m}}$ 两两不相交。

如果我们天真地进行这种层的添加，就不会终止。因此，我们需要进行循环检查，这可以看作是直觉主义逻辑序列演算中循环检查的推广版本。

定义 7.11 （层的模拟）。 设 L' 和 L 是分层序列中的层 $\mathfrak{G}$。 层模拟是 L' 与 L之间的非空二元关系$\mathsf{S} \subseteq (L' \times L) \cap \sim$ ，满足对所有 $x' \in L'$， x， $y \in L$，

（S1）每当 $x'\mathsf{S}x$ 且 $xR_{\mathfrak{G}}y$ 时，都存在 $y' \in L'$ ，使得$x'R_{\mathfrak{G}}y'$ 且 $y'\mathsf{S}y$，并且

（S2）每当 $x'\mathsf{S}x$ 且 $yR_{\mathfrak{G}}x$ 时，都存在 $y' \in L'$ ，使得$y'R_{\mathfrak{G}}x'$ 且 $y'\mathsf{S}y$。

我们称 L' 模拟 L ，当且仅当 L' 与 L之间存在层模拟。

命题 7.12。设 L' 和 L 是分层序列 $\mathfrak{G}$中的层。若 S 是 L' 与 L 之间的层模拟，则对于所有 $x \in L$，存在一个 $x' \in L'$ 使得 $x'\mathsf{S}x$。

定义 7.13。设 L 是分层序列 $\mathfrak{G}$中的最顶层。我们称 L 已被模拟，当且仅当在 $\mathfrak{G}$ 中存在一个层 L'，使得 $L' < L$ 且 L' 模拟 L。

定义 7.14（提升饱和）。设 $\mathfrak{G}$ 是一个饱和的非公理序列，且 L_1、.....、 L_n 是 $\mathfrak{G}$中所有未模拟的最顶层。我们将 $\mathfrak{G}$ 的提升饱和定义为$\mathfrak{G}{\uparrow} := \mathfrak{G} + \mathfrak{G}{\uparrow^{L_1}} + \cdots + \mathfrak{G}{\uparrow^{L_n}}$，其中在 $\mathfrak{G}{\uparrow^{L_1}}$、.....、 $\mathfrak{G}{\uparrow^{L_n}}$ 中的每一个中引入的新标签集合两两不相交。

引理 7.15. 如果 $\mathfrak{G}$ 是饱和的，那么 $\mathfrak{G}{\uparrow}$ 是稳定的。

D. 循环饱和

我们已经介绍了通过添加新的带标签公式或关系原子来使序列增长的所有步骤。因此，不满足公式变得满足，或者序列变得结构饱和。定义 7.14 给出了停止应用增长步骤的条件：即，当最顶层的层被下方的层模拟时。

不过请注意，一般来说，新层比前一层更大，增长有两个来源。首先，在不满足的情形 x: $\Box B^{\circ}$中，会添加一个新标签 z ；其次，如果标签 x 位于一个非单元素簇中，该簇就会被复制。这两种情况都可以在图 5 中看到。

因此，我们需要找到一种收缩层的方法。这将通过创建簇来实现，方式类似于定义 7.5 对 $\Diamond^{\bullet}$ 所做的处理。不同之处在于，这次潜在重复会涉及多个层，而不是局限于单个层。困难在于，层 L_1 的某一部分会在其上方出现的层 L_2 中重复，但为了保持序列树分层，我们不能跨多个层创建簇。我们在此采用的解决方案是：在层 L_2 内为未来某一层中将要重复的部分创建一个簇，前提是我们能够在层 L_2 中重复层 L_1 中发生的情况。我们将这类循环称为三角循环。这类循环有两种。第一种情况是，要在 L_2 中创建的簇位于 L_2 中一块在 L_1 中没有过去的区域。这可能是由例如 $\Box^{\circ}$ 的重复造成的。我们将这类循环称为 R-三角循环。第二种情况是，重复是由层提升过程中簇的反复复制造成的（见图 5）。我们将这类循环称为 U-三角循环。

在给出形式化定义之前，请注意，我们的算法所创建的所有新层都具有 $\mathfrak{G}{\uparrow^{x:F}}$的形式，如构造 7.9 中所定义的那样，并且每个这样的层恰好包含一个 suricata 标签。因此，这可以称为 该层的 suricata 标签，并且（紧接提升之后）它是唯一包含白公式的标签。[4]

定义 7.16 （R-三角循环）。 设 $\mathfrak{G}$ 是一个具有两层的饱和序列 $L_1 < L_2$。我们称簇$C_1 \subseteq L_1$ 和 $C_2 \subseteq L_2$ 构成一个 R-三角循环，当且仅当

1) $C_1 \sim C_2$; 2) 存在一个标签 $p_1 \in C_1$ ，使得存在一层 L'，满足 $L_1 < L' \leq L_2$ ，其中包含一个属于 p_1 的 suricata 标签；3) 存在一个簇 C_r ，使得 $C_1 \trianglelefteq_{\mathfrak{G}} C_r$，且 $C_r R_{\mathfrak{G}} C_2$，并且不存在满足 $C_r\ R_{\mathfrak{G}}\ v\ R_{\mathfrak{G}}\ C_2$ 且其过去在 L_1 中的标签 $v \in L_2 \setminus C_r$ ，即对于任意 $u \in L_1$，都有 $u \not\trianglelefteq_{\mathfrak{G}} v$ 。

当且仅当另外满足以下条件时，这个 R-三角循环是不满足的：

4) L_2 是最顶层；5) 存在标签 s、 $t \in L_2 \setminus C_2$ ，使得 $s \sim t$，且 $s \neq t$，且 $C_r\ R_{\mathfrak{G}}\ s\ R_{\mathfrak{G}}\ t\ R_{\mathfrak{G}}\ C_2$；6) 不存在满足 $C_r\ R_{\mathfrak{G}}\ u\ R_{\mathfrak{G}}\ t$ 的 suricata 标签 u ； 7) 我们不满足 $C_2\ R_{\mathfrak{G}}\ t$。

（注意，s 和 t 可能位于同一个簇中。）

图6（左）展示了该定义。

[4]另一方面，请注意，如果 x 中有多个 $\supset^{\circ}$-或 $\Box^{\circ}$-公式，那么一个标签 x 在其他层中可以有多个 suricata 标签。

0) 给定一个公式 F，定义 $\mathfrak{G}_0(F)$ 为序列 $r{\leq}r, rRr \Longrightarrow r{:}F$，并令 $\mathfrak{S}'_0 := \{\mathfrak{G}_0(F)\}$。 1) 对集合 $\mathfrak{S}'_i$，计算一个完全饱和 $\mathfrak{S}_i$。 2) 如果 $\mathfrak{S}_i$ 中的所有序列都是公理性的，则终止。公式 F 可证，并且我们可以在 labIS4$_{\leq}$ 中给出 $\Longrightarrow r{:}F$ 的证明（见第 9 节）。 3) 否则，选取一个非公理序列 $\mathfrak{G}_i \in \mathfrak{S}_i$，并计算其提升饱和 $\mathfrak{G}_i{\uparrow}$（见定义 7.14）。 4) 如果 $\mathfrak{G}_i{\uparrow} = \mathfrak{G}_i$，则终止。公式 F 不可证，而序列 $\mathfrak{G}_i$ 定义了一个反模型（见第 8 节）。 5) 否则，令 $\mathfrak{S}'_{i+1} := (\mathfrak{S}_i \setminus \{\mathfrak{G}_i\}) \cup \{\mathfrak{G}_i{\uparrow}\}$。 6) 转到步骤 1)。

图 7. 证明搜索算法

定义 7.17 (U-三角循环)。 设 $\mathfrak{G}$ 是一个有两层的饱和序列 $L_1 < L_2$。 我们称簇 $C_1 \subseteq L_1$ 和$C_2 \subseteq L_2$ 构成一个 U-三角循环当且仅当

1) $C_1 \sim C_2$; 2) 存在一个标签 $p_1 \in C_1$，使得存在一层 L'，满足$L_1 < L' \leq L_2$，且该层包含 p_1 的一个 suricata 标签; 3) $C_1 {\leq_{\mathfrak{G}}} C_2$。

当且仅当另外满足以下条件时，U-三角循环是不满足的:

4) L_2 是最顶层; 5) 存在标签 s、$t \in L_2 \setminus C_2$，使得 $C_1 {\leq_{\mathfrak{G}}} s$，且$C_1 {\leq_{\mathfrak{G}}} t$，且 $s \sim t$，且 $s \neq t$，并且 $C_2\ R_{\mathfrak{G}}\ s\ R_{\mathfrak{G}}\ t$或 $s\ R_{\mathfrak{G}}\ t\ R_{\mathfrak{G}}\ C_2$; 6) 不存在满足 $s\ R_{\mathfrak{G}}\ u\ R_{\mathfrak{G}}\ t$ 的 suricata 标签 u; 7) 我们不满足 $s\ R_{\mathfrak{G}}\ C_2\ R_{\mathfrak{G}}\ t$。

（同样，s 和 t 可能属于同一个簇。）

该定义所设想的两种可能性取决于 $C_2\ R_{\mathfrak{G}}\ s\ R_{\mathfrak{G}}\ t$ 还是 $s\ R_{\mathfrak{G}}\ t\ R_{\mathfrak{G}}\ C_2$，如图 6（中部和右部）所示。

非正式地说，当我们能够重现从创建 L' 开始并导致 L_2的步骤时，就称存在一个三角形循环。如果我们能在 L_2的新部分中观察到某种重复，则该循环是不满足的。在这种情形下，我们可以通过创建一个簇或缩小现有簇来消除这种重复，如下所示:

定义 7.18（循环饱和）。设 $\mathfrak{G}$ 和 $\mathfrak{G}'$ 是饱和序列。若在 $\mathfrak{G}$ 中存在一个不满足的 R-三角形或 U-三角循环，其中标签 s 和 t 分别如定义 7.16 或定义7.17 所示，并且 $\mathfrak{G}'$ 是通过将 s 替换 $\mathfrak{G}$ 中所有出现的 t，再对所得 $R_{\mathfrak{G}}$ 取传递闭包而得到的，则记为 $\mathfrak{G} \rightsquigarrow_{\circlearrowleft} \mathfrak{G}'$。对于序列的集合 $\mathfrak{S}$ 和 $\mathfrak{S}'$，若 $\mathfrak{S}$ 是饱和的，$\mathfrak{G} \rightsquigarrow_{\circlearrowleft} \mathfrak{G}'$ 对于某个 $\mathfrak{G} \in \mathfrak{S}$成立，并且 $\mathfrak{S}' = (\mathfrak{S} \setminus \{\mathfrak{G}\}) \cup \{\mathfrak{G}'\}$ 成立，则记为 $\mathfrak{S} \rightsquigarrow_{\circlearrowleft} \mathfrak{S}'$。我们将 $\circlearrowleft$的传递自反闭包记为 $\overset{*}{\circlearrowleft}$。如果 $\mathfrak{S} \rightsquigarrow^{*}_{\circlearrowleft} \mathfrak{S}'$ 并且 $\mathfrak{S}'$ 相对于 $\circlearrowleft$，处于范式，则 $\mathfrak{S}'$ 是 $\mathfrak{S}$的循环饱和。

和之前一样，这个术语是合理的，因为序列式饱和集的循环饱和是饱和的。

引理 7.19. 有:

a) 如果 $\mathfrak{S}$ 是饱和的且 $\mathfrak{S} \rightsquigarrow_{\circlearrowleft} \mathfrak{S}'$，，那么 $\mathfrak{S}'$ 是饱和的。b) 重写关系 $\circlearrowleft$ 是终止的。

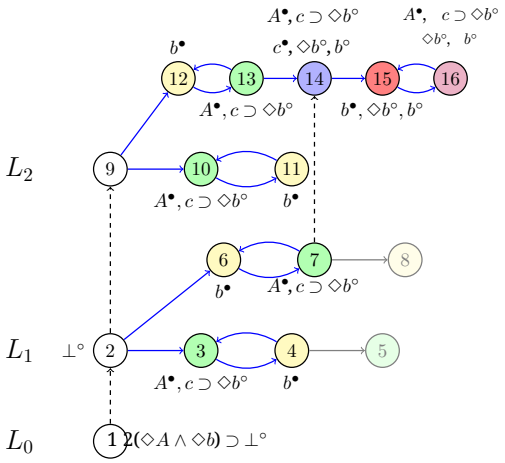


图 8 $\Box \Diamond A \wedge \Diamond b \supset \perp$ $A = c \supset \Diamond b \supset \perp$ $\Gamma^\bullet =$。公式 ()，其中 ()。 $\{\Box \Diamond A \wedge \Diamond b^\bullet\ \Diamond A \wedge \Diamond b^\bullet\ \Diamond A^\bullet\ \Diamond b^\bullet\}$ 1 集合 ()、、、存在于除 之外的所有节点中。

定义 7.20（完全饱和）。设 $\mathfrak{S}$ 是稳定序列式的集合。设 $\mathfrak{S}'$ 是 $\mathfrak{S}$的半饱和，$\mathfrak{S}''$ 是 $\mathfrak{S}'$的饱和，且 $\mathfrak{S}'''$ 是 $\mathfrak{S}''$的循环饱和。那么$\mathfrak{S}'''$ 是 $\mathfrak{S}$的完全饱和。

推论 7.21。如果序列式集合 $\mathfrak{S}$ 是稳定的，且 $\mathfrak{S}'''$ 是其完全饱和，则 $\mathfrak{S}'''$ 是饱和的。

E. 搜索算法

现在我们已经具备了证明/反模型搜索算法所需的全部要素，该算法见图 7。它使用本节讨论的变换，生成一个由饱和序列式集合组成的序列 $\mathfrak{S}_0$、$\mathfrak{S}_1$、$\mathfrak{S}_2$.....如果 $\mathfrak{S}_i$ 中的所有序列式都是公理性的（在这种情形下，我们可以在 labIS 4$_{\leq}$中生成证明，参见第 9 节），或者存在一个不再增长的序列式 $\mathfrak{G}_i \in \mathfrak{S}_i$（在这种情形下，我们可以构造一个反模型，参见第 8 节），则我们在步骤 i 终止。

例 7.22。 图 8 展示了算法运行在例 4.2 中的有效 IS4 公式上时生成的一个序列的（部分）图示表示。在层 L_1 簇 $\{3, 4\}$ 和 $\{6, 7\}$ 由饱和（定义 7.5 的选项 1）生成。层 L_2 通过提升饱和（定义 7.14）创建，其中14 是一个 suricata 标签，两个簇

$\{12,13\}$ 和 $\{15,16\}$ 来自 $\{6,7\}$。该序列是公理性的，因为15。

8. 反模型构造

假设我们以公式 F 启动该算法。如果算法在步骤 4 终止，我们就得到了一个饱和序列 $\mathfrak{G}_i$，使得 $\mathfrak{G}_i r{:}F^\circ$ 且 $\mathfrak{G}{\uparrow_i} = \mathfrak{G}_i$。这意味着，$\mathfrak{G}_i$ 的每个最顶层层要么是快乐的，要么可被模拟。因此，对于每个不满足的层 L，都存在一个良好内部层 L'，通过模拟 S_L模拟 L。现在，我们定义 $\mathfrak{G}_i^\star$：在 $\mathfrak{G}_i$ 中，对于某个不满足的层 L，只要满足$x'\mathsf{S}_L x$，就添加关系原子 $x{\leq}x'$；然后对所得结果取传递闭包。这样一来，最顶层层中的所有不满足的$A \supset B^\circ$ 和$\Box A^\circ$ 都变得快乐，同时保持结构饱和。因此，$\mathfrak{G}_i^\star$ 是一个快乐序列，从而可以应用定理 5.8，为 F得到一个有限反模型。

定理 8.1. 如果图 7 所示的算法在步骤 4 终止，那么公式 F 不是 IS4 的定理。

例 8.2。图 9 表示一个序列（的一个 $\leq$-分支），该序列由算法在处理第3节中的公式 (3) 时生成，也就是处理 $2(A \wedge B) \supset \bot$ with $A = \Box a \supset \bot$ and $B = (a \supset \bot) \supset \bot$ 时生成。如果我们纳入标签 12（但移除原子命题 $9 \leq 14$），该图描绘了算法发现一个不满足的 R-三角循环时的情形：采用定义 7.16 中的术语，我们取 $C_1 = \{3\}$、$p_1 = 3$、$C_2 = \{13\}$、$s = 14$、$C_r = \{s\}$ 和 $t = 12$。然后将 12 替换为 14，循环饱和便不再存在于序列中。随后，对图示 $\leq$-分支的搜索停止，因为最顶层 L_6 层可以由层 L_4 模拟。通过向序列中添加关系原子 $15{\leq}8$、$14{\leq}6$ 和 $13{\leq}7$（定理 8.1），我们得到公式的反模型（的一部分）。

9. 序列式证明展开

现在我们转向算法在步骤 2 终止的情形。此时，$\boldsymbol{\mathfrak{G}}_i$ 中的所有序列都是公理性的，我们要构造$\Longrightarrow r{:}\ F$ 在 labIS4$_\leq$中的一个证明。为此，我们将通过应用labIS4$'_\leq$的推理规则来模拟算法的步骤，首先应用规则 Rrf 和 $\leq$rf，得到序列 $r{\leq}r$，$rRr \Longrightarrow r{:}\ F$ 它就是算法的输入。一方面，这种展开看起来很容易，因为我们的搜索算法是作为labIS4$'_\leq$ 中的有组织证明搜索而设计的，而且大多数步骤确实可以通过应用labIS4$'_\leq$的规则来执行。然而，困难在于labIS4$'_\leq$ 生成的序列没有非单元素簇。我们称这类序列为适当的。

定义 9.1 (垂直序列式)。若分层序列 $\mathfrak{G}$ 满足以下条件，则称其为垂直的：对所有标签 u，u'，v和 v'，在 $\mathfrak{G}$中，(a) 若 $u{\leq_\mathfrak{G}}v$，且 $u{\leq_\mathfrak{G}}v'$，且 $vR^{\leftrightarrow}_\mathfrak{G}v'$，则 $v = v'$；(b) 若 $u{\leq_\mathfrak{G}}v$，且$u'{\leq_\mathfrak{G}}v$，且 $uR^{\leftrightarrow}_\mathfrak{G}u'$，则 $u = u'$；也就是说，对每个标签，每层至多有一个未来和一个过去。

定义 9.2（适当层/序列式）。树簇序列中的层 L 称为适当的，当且仅当所有簇 $C \subseteq L$都是单例。树簇序列 $\mathfrak{G}$ 称为适当的，当且仅当它是树分层的、垂直的，并且其所有层都是适当的。

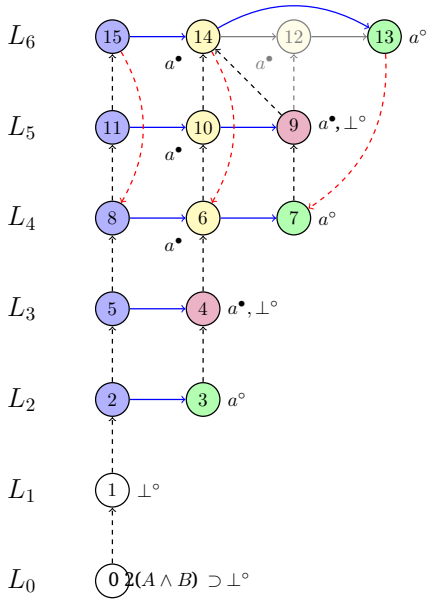


图 9. 公式 $\Box(A \wedge B) \supset \bot$ 与 $A = \Box a \supset \bot$ 和 $B = (a \supset \bot) \supset \bot$。$\Gamma^\bullet = \{\Box(A \wedge B)^\bullet,\ A \wedge B^\bullet,\ A^\bullet,\ B^\bullet,\ \Box a^\circ,\ a \supset \bot^\circ\}$ 存在于所有节点中，但 0。

构造证明的基本思想是通过一个“逐层”运作的推导来模拟该算法。关键观察是，每当我们在算法中创建一个簇时，都可以将其展开，也就是说，在一个仅包含适当序列式的实际证明中任意多次重复这个簇。而且，当我们将该簇提升到下一层时，这一性质仍会保留。

ˆ 定义 9.3（展开）。对于 $n \in \mathbb{N}$，适当序列 $\mathfrak{G}$ 称为稳定序列式 $\mathfrak{G}$ 的 n-展开，当且仅当存在一个二元 ˆn-展开关系 $\mathsf{U} \subseteq \ell(\mathfrak{G}) \times \ell(\mathfrak{G})$，其具有以下性质；其中我们假设 x Uˆx 且 $y\mathsf{U}$ˆy，并用 L_u 表示 u 的层：

（U1）公式不变性：$x \sim$ ˆx；（U2）R不变性：如果x 和 y 在 $\mathfrak{G}$中不属于同一簇，则 $xR_\mathfrak{G}y$当且仅当ˆ$xR_{\hat{\mathfrak{G}}}$ˆy；（U3）层不变性：$L_x \leq_\mathfrak{G} L_y$ 当且仅当 $L_{\hat{x}} \leq_{\hat{\mathfrak{G}}} L_{\hat{y}}$；（U4）“无过去”不变性：$x$ 在 $\mathfrak{G}$ 中没有过去，当且仅当ˆx 没有ˆ过去于 $\mathfrak{G}$中；（U5）单例簇不变性：如果 $C = \{z\}$ 是 $\mathfrak{G}$ 中的单例簇，则存在唯一的ˆ$z \in \ell(\hat{\mathfrak{G}})$，使得 $z\mathsf{U}$ˆz；（U6）非单例簇展开：如果C 是 $\mathfrak{G}$中的簇，且具有 $|C| = k \geq 2$，则对于所有 $i = 1..k$ 和 $j = 1..n$ ˆ，存在 $z_i \in \ell(\mathfrak{G})$以及ˆ$z_i$、$_j \in \ell(\mathfrak{G})$，使得 $z_i\mathsf{U}$ˆz_i、$_j$和 $C = \{z_1, \ldots, z_k\}$，并且还有ˆz_1、$_1R \cdots R$ˆz_k、$_1R$ˆz_1、$_2R \cdots R$ˆz_k、$_2R \cdots R$ˆz_1、$_nR \cdots R$ˆz_k、$_n$ˆ在 $\mathfrak{G}$中；（U7）单射性：如果ˆ$x =$ ˆy，则 $x = y$。

注意，每个 $(n+1)$ 展开 U 也是一个 n 展开；并且对于任意 n，任何适当稳定序列 $\mathfrak{G}$ 都是其自身的一个 n 展开，这是通过对角关系 $\{(x,x) \mid x \in \ell(\mathfrak{G})\}$ 实现的。

为确保展开仅作用于适当序列式，我们现在将引入 labIS4$^\star_\leq$，这是一个证明系统，其规则

对应于若干 labIS4′≤ 步骤的应用，从而强制 labIS4′≤ 的规则以特定方式应用。

令 $\mathfrak{G}$ 为序列 $\mathcal{R}\ \Gamma \Longrightarrow \Delta$，其中 $x{:}\ F$ 出现在 Δ中。于是记 $(\mathcal{R}\ \Gamma \Longrightarrow \Delta)\Uparrow^{x:F}$为 $\mathfrak{G}+\mathfrak{G}\uparrow^{x:F}$。由此可以定义以下推理规则：

$$\supset^{\circ}_{\star}\ \frac{(\mathcal{R}, \Gamma \Longrightarrow \Delta, x{:}A \supset B)\Uparrow^{x:A\supset B}}{\mathcal{R}, \Gamma \Longrightarrow \Delta, x{:}A \supset B} \qquad \Box^{\circ}_{\star}\ \frac{(\mathcal{R}, \Gamma \Longrightarrow \Delta, x{:}\Box A)\Uparrow^{x:\Box A}}{\mathcal{R}, \Gamma \Longrightarrow \Delta, x{:}\Box A}$$

对于标签 x 在序列 $\mathfrak{G}$中，定义 $\overleftarrow{x}$ 为标签集合 x'，其中 $x' R_{\mathfrak{G}} x$。对于标签集合 X，记 XRy 为关系原子集合 $\{xRy \mid x \in X\}$。使用此记号，可以定义规则：

$$\Diamond^{\bullet}_{\star}\ \frac{\mathcal{R}, \overleftarrow{x} R y, yRy, y{\leq}y, \Gamma, x{:}\Diamond A, y{:}A \Longrightarrow \Delta}{\mathcal{R}, \Gamma, x{:}\Diamond A \Longrightarrow \Delta}\ y \text{ fresh}$$

现在定义系统 labIS4$^{\star}_{\leq}$ 为

- 图2中的推理规则 id、$\bot^{\bullet}$ 和 $\Diamond^{\circ}$；
- 规则 $\Box^{\bullet}_{\star}$ 通过设置 $x = y$ 在 $\Box^{\bullet}$ 图2中得到；
- 规则 $\wedge^{\bullet}_{\star}$、$\wedge^{\circ}_{\star}$、$\vee^{\bullet}_{\star}$、$\vee^{\circ}_{\star}$和 $\supset^{\bullet}_{\star}$ 分别是 $\wedge^{\bullet}$、$\wedge^{\circ}$、$\vee^{\bullet}$、$\vee^{\circ}$和 $\supset^{\bullet}$ 在图2中的变体，其中主公式不会在前提中被删除，前提在单调性下封闭，并且另外，对于 $\supset^{\bullet}_{\star}$，$x = y$；
- 规则 $4^{\bullet}$ 和 4° 见图3；
- 规则 $\supset^{\circ}_{\star}$、$\Box^{\circ}_{\star}$和 $\Diamond^{\bullet}_{\star}$ 如上所示。

注意，labIS4$^{\star}_{\leq}$中没有结构规则。

命题 9.4。labIS4$^{\star}_{\leq}$ 中的每条规则 r 都可在 labIS4′≤中推导出来；如果 $\mathfrak{G}$ 是 r 的某个实例的结论，那么每个前提都具有 $\mathfrak{G}+\mathfrak{H}$ 的形式，其中某个 $\mathfrak{H}$成立。

在书写前提 $\mathfrak{G}+\mathfrak{H}$时，我们假定 $\mathfrak{G}$ 与 $\mathfrak{H}$ 不相交，即 $\mathfrak{H}$ 的所有带标签公式和关系原子都不出现在结论中。

定义 9.5（触及）。设 r 是 labIS4$^{\star}_{\leq}$ 中一条规则的实例，且其结论中有一个序列 $\mathfrak{G}$。如果对于 r 的某个前提 $\mathfrak{G}+\mathfrak{H}$，我们有 $u{\leq_{\mathfrak{H}}}x$，或者对于某个标签 u 有 $uR_{\mathfrak{H}}x$，或者对于某个公式 B 有$\mathfrak{H}$、$x{:}\ B^{\bullet}$，或者 $\mathfrak{H}$、$x{:}\ B^{\circ}$，则称 r 触及标签 $x \in \ell(\mathfrak{G})$。如果 r 触及 X 中的某个 $x \in X$，则称 r 触及标签集合 X。

定义 9.6（受限序列式/禁用标签）。如果序列式 $\mathfrak{G}$ 配备了一个禁用标签的集合 $\mathrm{f}(\mathfrak{G}) \subseteq \ell(\mathfrak{G})$，并满足以下两个条件，则称其为受限序列式：

（a）如果 $y \in \mathrm{f}(\mathfrak{G})$ 且 $xR_{\mathfrak{G}}y$，则 $x \in \mathrm{f}(\mathfrak{G})$，即禁用标签的所有父节点均为禁用标签；（b）如果对于某个 $y \in \ell(\mathfrak{G})$，$L_x <_{\mathfrak{G}} L_y$，则 $x \in \mathrm{f}(\mathfrak{G})$，即来自内部层的所有标签均为禁用标签。

$\ell(\mathfrak{G})$ 中所有未被禁用的标签均为允许标签。

在下文中，我们假设所有序列式均为受限序列式，特别是推导中出现的所有序列式。

定义 9.7（整洁规则/推导）。设 $\mathfrak{G}$ 是 labIS4$^{\star}_{\leq}$ 中某个规则 r 的一个实例的结论。如果（i）r 不触及 $\mathfrak{G}$ 中的任何禁用标签，并且（ii）r 的每个前提 $\mathfrak{K}$ 的禁用标签为

$$f(\mathfrak{K}) = \begin{cases} \ell(\mathfrak{G}) & \text{if r} \in \{\supset^{\circ}_{\star}, \Box^{\circ}_{\star}, \Diamond^{\bullet}_{\star}\}, \\ f(\mathfrak{G}) & \text{otherwise.} \end{cases} \tag{4}$$

当且仅当推导 D 由整洁规则实例组成时，称其为整洁推导。

换言之，整洁推导中的规则不会触及禁用标签，并且要么保持所有标签和禁用标签不变，要么引入新的允许标签，同时将所有旧标签变为禁用标签。这尤其足以确保整洁推导中的所有序列式都是适当的。

命题 9.8. 设 D 是一个整洁推导。若 D 的结论适当，则 D 中的每个序列也都适当。

定义 9.9（可展开）。若推导 D_n 是稳定集 $\mathfrak{S}$ 的一个 n-展开，当且仅当 D_n 的每个前提都是来自 $\mathfrak{S}$ 的某个序列的一个n-展开。对于公式 F，稳定集 $\mathfrak{S}$ 是 F-可展开的，当且仅当对每个 $n \in \mathbb{N}$，都存在一个整洁推导 D_n，其结论为 $\mathfrak{G}_0(F)$（如图 7 中算法的步骤 0 所定义），且 D_n 是 $\mathfrak{S}$ 的一个 n-展开。

显然，$\mathfrak{S}'_0 = \{\mathfrak{G}_0(F)\}$ 是 F-可展开的。现在需要证明，算法中执行的所有操作，尤其是重写关系 s、$\Diamond$ 和 $\circlearrowleft$，都保持这一性质。对于 s，这很直接，因为我们只需应用 labIS4$^{\star}_{\leq}$ 的推理规则来匹配半饱和步骤。对于 $\Diamond$ 和 $\circlearrowleft$，我们需要展开新创建的簇。基本思想是，将导致该簇首次出现的所有证明步骤重复 n 次。

引理 9.10（展开引理）。算法根据图 7 生成的所有序列集合 $\mathfrak{S}_i$ 都是 F- 可展开的。

因此，当我们在步骤 2 中终止时，可以构造出 $\mathfrak{S}_i$ 的 1-展开，这构成了 labIS4$^{\star}_{\leq}$ 中的证明，因为公理序列的展开始终是公理性的。

定理 9.11。如果图 7 所示的算法在步骤 2 中终止，那么公式 F 是 IS4 的定理。

10. 终止

我们已经在第 7 节中证明，算法中的每一步（如图 7 所示）都会终止。接下来需要证明，我们不可能永远运行主循环，即序列 $\mathfrak{S}_0$、$\mathfrak{S}_1$、……、$\mathfrak{S}_i$、…… 最终会在步骤 2（找到证明）或步骤 4（找到反模型）中终止。基本思想是限制 $\mathfrak{S}_i$ 的序列中一层的大小。这样一来，不同可能层的数量是有限的，因此我们最终会找到一个模拟（定义 7.11、7.13）。

在本节余下部分，我们假设步骤 0 中的一个公式 F 包含 n 个子公式出现。

我们将序列 $|x|$ 中标签 x 的 $\mathfrak{G}$ 大小定义为不同公式出现的数量 $x{:}\ A^{\bullet}$ 或者 $x{:}\ A^{\circ}$ 在 $\mathfrak{G}$中。

引理 10.1。某个 $\mathfrak{S}_i$ 的序列中出现的标签大小至多为 n。并且，关于 $\sim$的标签等价类有 2^n 个。

簇的大小 $|C|$ 是 C 中标签的数量 C。

引理 10.2。某个 $\mathfrak{S}_i$ 的序列式中出现的簇的大小至多为 2^n。相对于 $\sim$，非空簇的等价类共有 $2^{2^n}-1$ 个。

算法只访问稳定序列式。此类序列式中的层 L 是由簇构成的树。令 M 为 L 中的一条分支，即从根到叶节点依次经过的簇序列 C_1、C_2、……、C_l。M 的长度 $|M|$ 是总和 $|C_1|+|C_2|+\cdots+|C_l|$。

引理 10.3。集合 $\mathfrak{S}_i$ 中序列式的一层中的分支长度有界，且该界由 F 决定。

定理 10.4（终止性）。图 7 中给出的证明搜索算法是终止的。

11. 结论

在这篇论文中，我们解决了一个悬而未决近 30 年的问题。我们的解决方案包含两个关键要素。

首先，对两个二元关系都使用带有关系原子的完全标记系统，使我们能够给出一个仅包含可逆规则的证明系统，并且还实现了序列与模型之间更紧密的对应。

其次，尽管在证明搜索过程中通过识别标签来实现循环在先验上并不可靠，但在适当的情况下，如果我们以某种系统化的方式组织证明搜索，就可以保持可靠性。

我们推测，同样的方法也可以应用于 IK4。IK4 是不包含 t-公理的 IS4，也是直觉主义版本的 S5 立方体中另一个可判定性尚未解决的逻辑。事实上，整体论证是相同的，但由于缺少自反性，许多定义和证明论证会存在细微差异。因此，对 IK4 的完整处理将超出本文的范围。

算法的实现是当前正在进行的工作。未来，我们希望研究 IS4 中可证明性的复杂度。

致谢。Roman Kuznets 受到奥地利科学基金（FWF）项目 ByzDEL（P 33600）的资助。Marianna Girlando 受到 Horizon 2021 计划资助，资助项目为 Marie Skłodowska-Curie 项目 CYDER（101064105）。

参考文献

[1] N. Alechina 和 D. Shkatov。《直觉主义模态逻辑可判定性的通用证明方法》。《应用逻辑杂志》，4 230，2006 3 219 10 1016 2005 06 007 ( ): – . doi: . /j.jal. . . . [2] P. Balbiani、M. Di´eguez 和 D. Fern´andez-Duque。《具有有限模型性质的 S4 的若干构造性变体》。载于 LICS 2021。IEEE，2021。doi:10.1109/LICS52264.2021.9470643。[3]K. Br¨unnler。《模态逻辑的深层序列演算系统》。《数理逻辑档案》，48(6):551–577，2009。doi:10.1007/s00153-009-0137-3。[4] K. Chaudhuri、S. Marin 和 L. Straßburger。《直觉主义模态逻辑的模块化聚焦证明系统》。载于 FSCD 2016。LZI，2016。doi:10.4230/LIPIcs.FSCD.2016.16。[5]R. Dyckhoff。《自 Gentzen 以来的直觉主义判定程序》。载于《证明论进展》，第 245–267 页。施普林格，2016。doi:10.1007/978-3-319-29198-7_6。

[6]R. Dyckhoff 和 S. Negri. 中间逻辑中的证明分析。《数理逻辑档案》，51(1–2):71–92, 2012. doi:10.1007/s00153-011-0254-7.[7] W. B. E wald. 直觉主义时态逻辑与模态逻辑。《符号逻辑杂志》, 51(1):166–179, 1986. doi:10.2307/2273953. [8] G. Fischer Servi. 若干直觉主义模态逻辑的公理化。《都灵理工大学数学研讨会报告》, 42(3):179–194, 1984. [9] D. Gabelaia, A. Kurucz, F. Wolter 和 M. Zakharyaschev。“传递的”模态逻辑的乘积。《符号逻辑杂志》, 70 1021, 2005 3 993 10 2178 1122038925 ( ): – . doi: . /jsl/ . [10] D. Garg, V. Genovese 和 S. Negri. 多模态逻辑中由序列演算得到的反模型。载于 LICS 2012, 第 315–324 页。IEEE, 2012. doi:10.1109/LICS.2012.42. [11] G. Gentzen. 关于逻辑推理的研究 I.《数学杂志》, 39:176–210, 1935. doi:10.1007/BF01201353. [12] Y. Hasimoto. 若干直觉主义模态逻辑的有限模型性质。《逻辑分部通报》, 30(2):87–97, 2001. [13] S. Heilala 和 B. Pientka. 直觉主义命题模态逻辑 IS4 的双向判定程序。载于 CADE 2007, 第 116–131 页。施普林格, 2007. doi:10.1007/978-3-540-73595-3_9. [14] O. Ketonen. 关于谓词演算的研究。《芬兰科学院年刊》, Series A, I, 23, 1944. [15] S. C. Kleene. 《元数学导论》。北荷兰出版社, 1952. [16] R. Kuznets 和 L. Straßburger. Maehara 风格的模态嵌套演算。《数理逻辑档案》, 58(3–4):359–385, 2019. doi:10.1007/s00153-018-0636-1. [17] R. E. Ladner. 模态命题逻辑系统中可证明性的计算复杂性。《SIAM计算杂志》, 6 480, 1977 3 467 10 1137 0206033 ( ): – . doi: . / .[18] F. Lamarche. 关于结构上下文代数。已被《计算机科学中的数学结构》接受, 2001. URL: https://hal.inria.fr/inria-00099461. [19] P. Maffezioli, A. Naibo 和 S. Negri. 从结构证明论的角度看 Church–Fitch可知性悖论。《综合》, 190(14):2677–2716, 2013. doi:10.1007/s11229-012-0061-7. [20] S. Marin, M. Morales 和 L. Straßburger. 直觉主义模态逻辑的全标记证明系统。《逻辑与计算杂志》, 31(3):998–1022, 2021. doi:10.1093/logcom/exab020. [21] G. E. Minc. 关于若干模态逻辑演算。《斯捷克洛夫数学研究所报告》, 98(1968):97–124, 1971. [22] S. Negri. 模态逻辑中的证明分析。《哲学逻辑杂志》, 34(5–6):507–544, 2005. doi:10.1007/s10992-005-2267-3. [23] K. Ono. 关于数学基础的逻辑研究。《东京帝国大学理学部纪要》, section I, III(7):329–389, 1938. [24] G. Plotkin 和 C. Stirling. 直觉主义模态逻辑的框架。载于 TARK 1986, 第 399–406 页, 1986. URL: http://www.tark.org/proceedings/tark mar19 86/p399-plotkin.pdf. [25] M. Sato. 以 Gentzen序列方法研究若干模态逻辑的克里普克型模型。《京都大学数理解析研究所讲究录》, 13 468, 1977 2 381 10 2977 1195189814 ( ): – . doi: . /PRIMS/ .[26] A. K. Simpson. 《直觉主义模态逻辑的证明论与语义学》。博士论文，爱丁堡大学, 1994. URL: http://hdl.handle.net/1842/407. [27] L. Straßburger. 直觉主义模态逻辑嵌套序列中的切消去。载于 FoSSaCS 2013, 第 209–224 页。施普林格, 2013. doi:10.1007/978-3-642-37075-5_14. [28] A. S. Troelstra 和 H. Schwichtenberg. 《基础证明论》。剑桥大学出版社, 2000. doi:10.1017/CBO9781139168717. [29] F. Wolter 和 M. Zakharyaschev. 作为经典双模态逻辑片段的直觉主义模态逻辑。载于《工作中的逻辑》, 第 168–186 页。Physica, 1999.

附录

A. 示例

本节列举了一些示例，用以说明搜索算法（图 7）的工作方式。有关图示符号的说明，请参阅例 5.2。

由于篇幅所限，我们只展示算法生成的一个序列，也就是公理序列，或能够据此为根处的公式生成反模型的序列。请注意，即使找到公理序列，算法也可能继续运行：要使算法停止（在步骤 2），集合中的所有序列都必须是公理性的。反之，如果算法找到一个包含足够信息以构造反模型的序列，就会停止（在步骤 4）。

示例 A.1（有效公式）。考虑公式 $2(\Diamond A \wedge \Diamond b) \supset \bot$，其中 $A = (c \supset \Diamond b) \supset \bot$，该公式在 IS4 中有效。图 10 展示了算法生成的一个序列。令 $\Gamma^\bullet$ 为集合$\{\Box(\Diamond A \wedge \Diamond b)^\bullet, \Diamond A \wedge \Diamond b^\bullet, \Diamond A^\bullet, \Diamond b^\bullet\}$。图中除 1 外的每个标签都对应于 $\Gamma^\bullet$，以及图中节点旁明确显示的公式。下列 $\leq$-关系未显示出来，但包含在序列中： $3{\leq}10$、 $4{\leq}11$、 $6{\leq}12$、 $6{\leq}15$、 $7{\leq}13$ 和 $7{\leq}16$。

在层 L_1，由于定义 7.5 的选项 1（饱和），对两个 R-分支的搜索均停止：其中，5 被替换为 3，8 被替换为 6。这生成了两个非单例簇：$C_1 = \{3, 4\}$ 和 $C_2 = \{6, 7\}$。随后，由于 L_1中的所有标签几乎快乐，应用提升饱和：对于序列中存在的每个 $\supset^\circ$-公式，重写都会“提升”该层的一个副本。我们只表示其中一层，即由公式 7 生成的 L_2: $c \supset \Diamond b^\circ$。由于 7 属于簇 C_2，根据定义 7.8–7.9，该簇在 L_2中被复制。然后，经过一个半饱和步骤，我们得到 15:$b^\bullet$和 15:b°均成立。因此，我们停止沿此分支的搜索。如果算法处理的所有序列最终都得到公理序列，那么算法就会停止；根据定理 9.11，根处的公式是 IS4 的一个定理。事实上，其在 labIS4$_\leq$ 中的推导可以在图 4（左）中找到。

示例 A.2（无效公式，R-三角循环）。现在考虑第3节中的公式3，即，$2(A \wedge B) \supset \bot$ with$A = \Box a \supset \bot$ 和 $B = (a \supset \bot) \supset \bot$。在图11的左侧，我们表示了由算法生成的一个序列。令$\Gamma^\bullet = \{\Box(A \wedge B)^\bullet, A \wedge B^\bullet, A^\bullet, B^\bullet, \Box a^\circ, a \supset \bot^\circ\}$。对于图中的每个标签（0除外），我们关联 $\Gamma^\bullet$以及节点旁明确显示的公式。准确地说，该图仅显示了序列的一个 $\leq$-分支：公式$\Box a^\circ$ 和 $a \supset \bot^\circ$ 位于所有标签中，并且每当它们不满足时，也会将其提升，从而生成一个新的层（图中未显示）。

在层 L_6 检测到一个不满足的 R-三角循环：采用定义 7.16 中的术语，我们取 $C_1 = \{3\}$、$p_1 = 3$、$C_2 = \{13\}$、 $s = 14$、 $C_r = \{s\}$ 和 $t = 12$。图11右侧表示了循环饱和后得到的序列，其中 12 被替换为 14。注意，簇 $\{14\}$ 保持不变。循环饱和后，

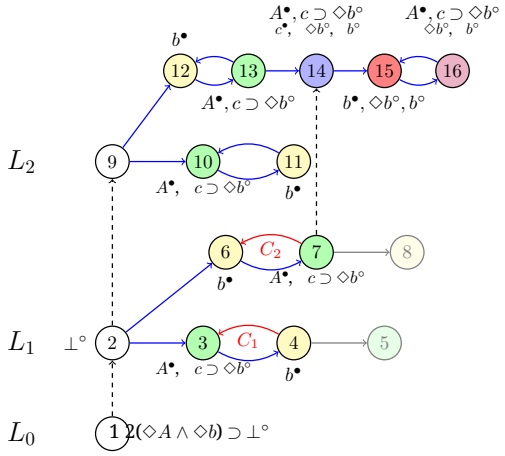


图 10。公式 $\Box(\Diamond A \wedge \Diamond b) \supset \bot$，其中 $A = (c \supset \Diamond b) \supset \bot$。

沿着 $\leq$-分支进行的证明搜索停止：这是因为层 $L_{6'}$ 可以由层 L_4模拟。根据定理 8.1，将关系原子 $15{\leq}8$、 $14{\leq}6$ 和 $13{\leq}7$ 添加到序列中后，它们对应于图中向下指向的虚线箭头，我们便得到公式的反模型的（部分）模型。要完成反模型，我们需要考虑序列中存在的不满足 $\Box^\circ$ 和 $\supset^\circ$ 公式所生成的层。具体而言，$8{:}\ a \supset \bot^\circ$ 和 $8{:}\ a \supset \bot^\circ$ 是不满足的。这些公式会在序列中产生更多的 $\leq$-分支（但所有分支都是有限的，并且分支的数量也是有限的）。

例 A.3 (无效公式，U-三角循环)。 考虑公式 $\Box(\Diamond D \wedge \Diamond c) \supset \bot$，其中 $D = (a \supset b) \supset \bot$。图 12 展示了由搜索算法生成的一个序列。设 $\Gamma^\bullet = \{\Box(\Diamond D \wedge \Diamond c)^\bullet,\ \Diamond D \wedge \Diamond c^\bullet,\ \Diamond D^\bullet, \Diamond c^\bullet\}$。我们为每个标签关联以下集合，这些集合在图中以不同颜色标示：

1	$\Gamma^\bullet \cup \{\Box(\Diamond D \wedge \Diamond c) \supset \bot^\circ\}$
2	$\Gamma^\bullet \cup \{\bot^\circ\}$
$10 \sim 19 \sim 29$	$\Gamma^\bullet$
$3 \sim 8 \sim 6 \sim 18 \sim 15 \sim 12 \sim 28 \sim 23 \sim 21$	$\Gamma^\bullet \cup \{D^\bullet, a \supset b^\circ\}$
$4 \sim 9 \sim 7 \sim 17 \sim 14 \sim 13 \sim 27 \sim 24 \sim 22 \sim$	$\Gamma^\bullet \cup \{c^\bullet\}$
$16 \sim 26 \sim 25$	$\Gamma^\bullet \cup \{D^\bullet, a^\bullet, a \supset b^\circ\}$
$5 \sim 11 \sim 20$	$\Gamma^\bullet \cup \{D^\bullet, a^\bullet, b^\circ, a \supset b^\circ$

$\}$与前一个示例一样，这里只表示了一个 $\leq$-分支。此外，出于篇幅考虑，我们没有画出由R通过饱和产生的第二个2 -分支，以使公式2：$\Diamond c^\bullet$ happy。通过提升该分支得到的标签存在于所有层中。

如示例 A.1 所示，在层 L_1中，饱和后会创建一个非单例循环， $\{3, 4\}$。随后，通过提升饱和生成层 L_2，该过程会复制簇。重复此过程以生成 L_3。在 L_3处，出现一个不满足的 U-三角形

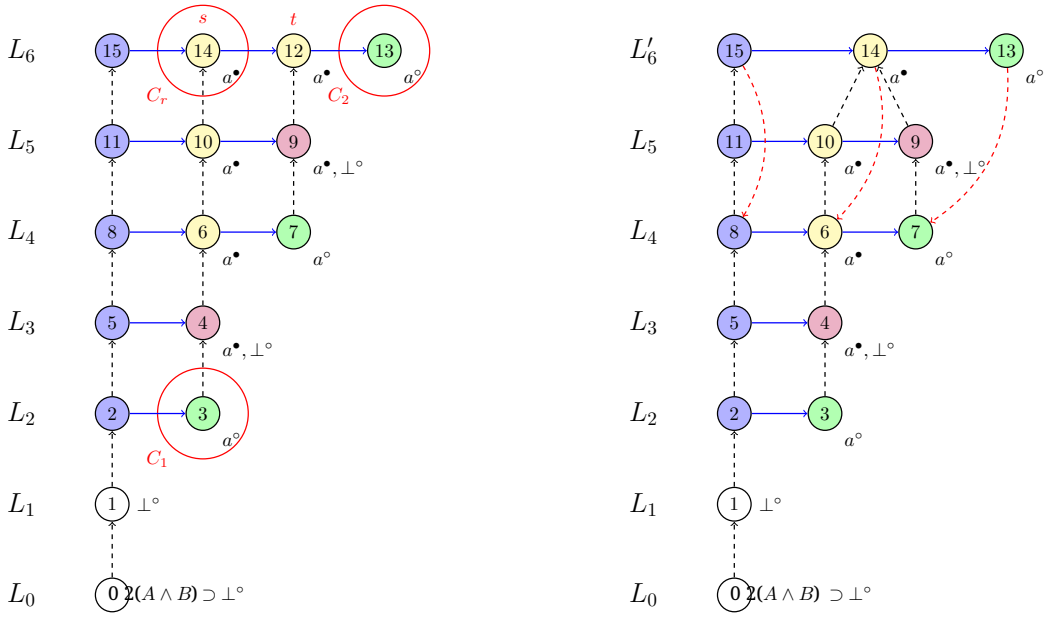


图11。公式$\square(A \wedge B) \supset \perp$其中 $A = \square a \supset \perp$ 和 $B = (a \supset \perp) \supset \perp$。对最左侧序列应用循环饱和步骤，饱和结果即为最右侧序列。

发现了循环。使用定义7.17中的术语，取$C_1 = \{3, 4\}$、$C_2 = \{12, 13\}$、$s = 18$和$t = 15$。因此，我们将 15 替换为 18。此外，通过取 $s' = 17$ 和 $t' = 14$，可以找到第二个不满足的 U-三角形循环（参见图12的最上层序列）。层 $L_{3'}$ 是循环饱和的结果：其中包含一个簇 $\{18, 17, 16\}$。算法生成一个由层 L_0 - L_2 组成的序列，$L_{3'}$ 并继续执行（图 12 中的中间序列）。

随后，经过提升饱和后，生成层 L_4。这里存在三个不满足的U三角循环：取 $C_1 = \{3, 4\}$，然后是$C_2 = \{21, 22\}$，再接着是 $s = 28$ 和 $t = 23$、$s' = 27$ 和$t' = 24$、$s'' = 26$ 和 $t'' = 25$。经过循环饱和后，我们得到层 $L_{4'}$，它可以由层 $L_{3'}$（图12中的最下层序列）模拟。添加关系原子 $29 \leq 19$、$28 \leq 18$、$27 \leq 17$、$26 \leq 16$、$20 \leq 11$、$21 \leq 12$和 $22 \leq 13$后，我们得到一个针对根处公式的（部分）反模型。要使反模型完备，我们需要纳入通过提升序列中与每个其余不满足公式相对应的层所生成的层（$a \supset b$）°。

B. 第 5 节中的证明

定理 5.8（完备性）。对于一个满足条件的序列 $\mathfrak{G}$，其模型 $\mathrm{M}_{\mathfrak{G}} = \langle \ell(\mathfrak{G}) \leq_{\mathfrak{G}}, R_{\mathfrak{G}}, V \rangle$ 是一个传递且自反的双关系模型，并具有以下两个性质：

- 若 $\mathfrak{G}, x\colon A^\bullet$，则 $m_{\mathfrak{G}}, x \Vdash A$;
- 若 $\mathfrak{G}, x\colon A^\circ$，则 $m_{\mathfrak{G}}, x \nVdash A$。

证明。$\mathrm{M}_{\mathfrak{G}}$ 的世界 W 是序列的标签。条件F_1 和F_2, 所表达的 $\leq_{\mathfrak{G}}$ 和$R_{\mathfrak{G}}$的传递性与自反性，以及 V的单调性，都因结构饱和而通过构造得到。因此，$\mathrm{M}_{\mathfrak{G}}$ 是一个传递且自反的双关系模型。剩下的只是证明这两个关于强制的性质；我们通过对A的大小进行互归纳，并根据 A的主要联结词进行分类讨论来证明：

- $\mathfrak{G}, x\colon \perp^\bullet$：对于满足条件的序列，这是不可能的。
- $\mathfrak{G}, x\colon \perp^\circ$：根据定义，我们有 $m_{\mathfrak{G}}, x \nVdash \perp$。
- $\mathfrak{G}, x\colon a^\bullet$：根据定义 5.7，$m_{\mathfrak{G}}, x \Vdash a$。
- $\mathfrak{G}, x\colon a^\circ$：并非 $\mathfrak{G}, x\colon a^\bullet$ 根据 x的幸福性，因此，$m_{\mathfrak{G}}, x \nVdash a$ 根据定义 5.7。
- $\mathfrak{G}, x\colon B \wedge C^\bullet$：根据$x$的幸福性，$\mathfrak{G}, x\colon B^\bullet$ 且$\mathfrak{G}, x\colon C^\bullet$。于是，$m_{\mathfrak{G}}, x \Vdash B$且$m_{\mathfrak{G}}, x \Vdash C$根据归纳假设。因此，$m_{\mathfrak{G}}, x \Vdash B \wedge C$。
- $\mathfrak{G}, x\colon B \wedge C^\circ$：根据 x的幸福性，$\mathfrak{G}, x\colon B^\circ$ 或$\mathfrak{G}, x\colon C^\circ$。于是，要么$m_{\mathfrak{G}}, x \nVdash B$要么$m_{\mathfrak{G}}, x \nVdash C$根据归纳假设。因此，$m_{\mathfrak{G}}, x \nVdash B \wedge C$。
- 关于 $\mathfrak{G}, x$的情形：$B \vee C^\bullet$ 和 $\mathfrak{G}, x\colon B \vee C^\circ$ 是类似的。
- $\mathfrak{G}, x$：考虑任意满足 $x \leq_{\mathfrak{G}} y$ 的 y。根据（mon$^\bullet$）-结构饱和性，有 $\mathfrak{G}, y\colon B \supset C^\bullet$。根据 y 的幸福性，要么 $\mathfrak{G}, y\colon B^\circ$，要么 $\mathfrak{G}, y\colon C^\bullet$。根据归纳假设，要么 $\mathrm{M}_{\mathfrak{G}}, y \nVdash B$，要么 $\mathrm{M}_{\mathfrak{G}}, y \Vdash C$。因此，对于所有满足 $x \leq_{\mathfrak{G}} y$ 的 y，$\mathrm{M}_{\mathfrak{G}}, y \Vdash B$ 蕴含 $\mathrm{M}_{\mathfrak{G}}, y \Vdash C$。所以，$\mathrm{M}_{\mathfrak{G}}, x \Vdash B \supset C$。
- $\mathfrak{G}, x\colon B \supset C^\circ$：根据 x 的幸福性，存在一个世界 y 使得 $x \leq_{\mathfrak{G}} y$，$\mathfrak{G}, y\colon B^\bullet$，并且 $\mathfrak{G}, y\colon C^\circ$。根据归纳假设，$m_{\mathfrak{G}}, y \Vdash B$以及 $m_{\mathfrak{G}}, y \nVdash C$。因此，$m_{\mathfrak{G}}, x \nVdash B \supset C$。
- $\mathfrak{G}, x\colon \Diamond B^\bullet$：根据$x$的幸福性，存在一个世界 y 使得 $xR_{\mathfrak{G}}y$ 且 $\mathfrak{G}, y\colon B^\bullet$。根据归纳假设，我们有$m_{\mathfrak{G}}, y \Vdash B$因此，$m_{\mathfrak{G}}, x \Vdash \Diamond B$。
- $\mathfrak{G}, x\colon \Diamond B^\circ$：根据 x的幸福性，我们有 $\mathfrak{G}, y\colon B^\circ$ 对于所有世界 y 使得 $xR_{\mathfrak{G}}y$。因此，根据归纳假设，$m_{\mathfrak{G}}, y \nVdash B$只要 $xR_{\mathfrak{G}}y$。因此，$m_{\mathfrak{G}}, x \nVdash \Diamond B$。
- $\mathfrak{G}, x\colon \square B^\bullet$: 任意考虑 y 和 z，其中有 $x \leq_{\mathfrak{G}} y$ 和$yR_{\mathfrak{G}}z$。根据（mon$^\bullet$）-结构饱和性，$\mathfrak{G}, y\colon \square B^\bullet$。根据 y 的幸福性，我们有 $\mathfrak{G}, z\colon B^\bullet$。因此，根据归纳假设，$\mathrm{M}_{\mathfrak{G}}, z \Vdash B$ 只要 $x \leq_{\mathfrak{G}} y$ 和 $yR_{\mathfrak{G}}z$。所以，$\mathrm{M}_{\mathfrak{G}}, x \Vdash \square B$。

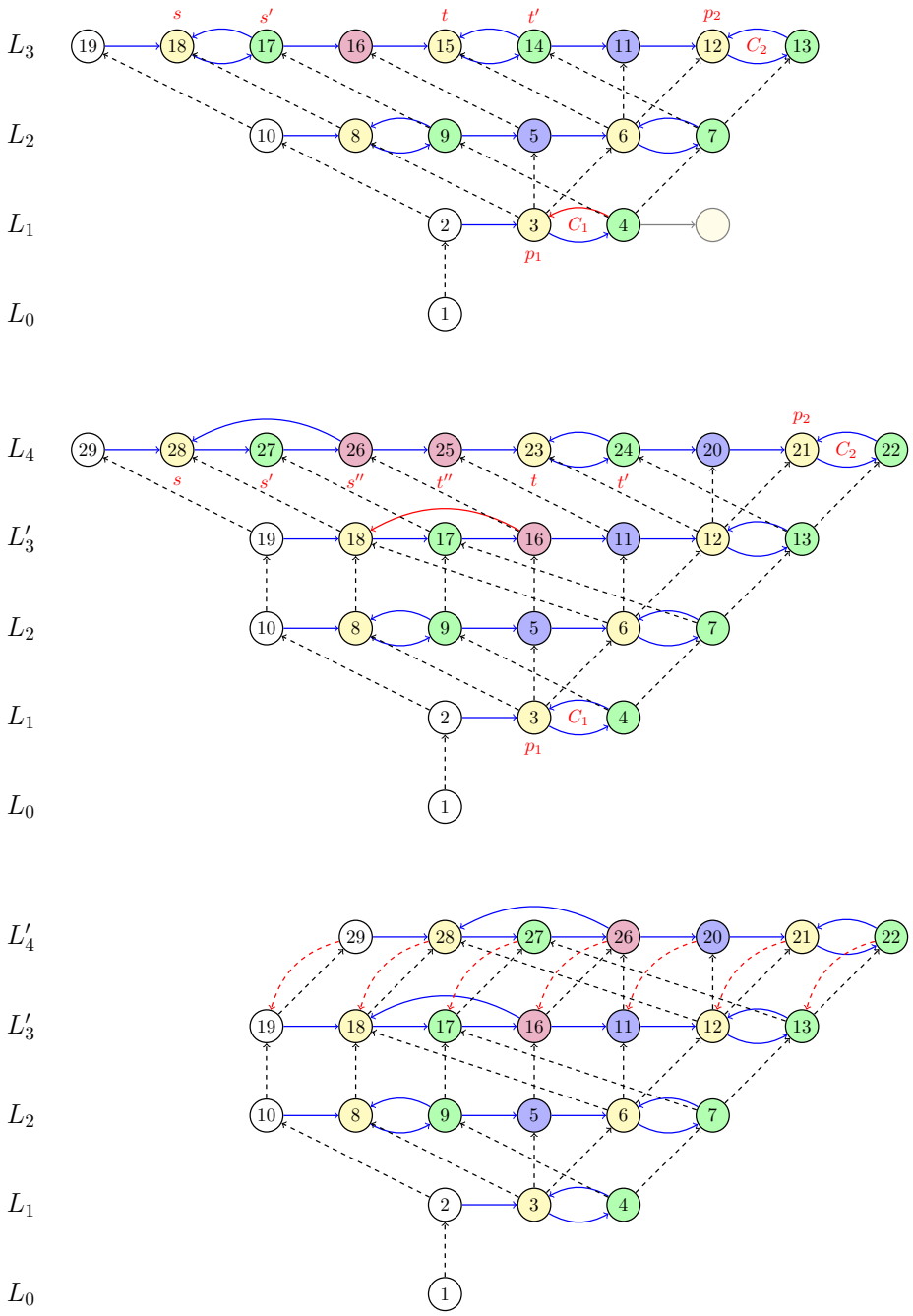


图12。公式$\Box(\Diamond D \wedge \Diamond c) \supset \bot$，带有 s 第 $D = (a \supset b) \supset \bot$。对最上层序列应用循环饱和步骤，所得饱和结果表示在中间序列中。然后，也对中间序列应用循环饱和，所得结果为最下层序列。

- $\mathfrak{G}, x \colon \Box B^{\circ}$：根据 x 的幸福性，存在 y 和 z 使得 $x \leq_{\mathfrak{G}} y$，$y R_{\mathfrak{G}} z$，以及 $\mathfrak{G}, z \colon B^{\circ}$。根据归纳假设，$m_{\mathfrak{G}}$，$z \nVdash B$。因此，$m_{\mathfrak{G}}$，$x \nVdash \Box B$。 □

C. 第6节中的证明

我们从一个不在正文中但能简化许多论证的命题开始。

命题 A.4. 结构饱和序列 $\mathfrak{G}$ 相对于 $R^{\leftrightarrow}_{\mathfrak{G}}$ 也是饱和的：

（F） 若 $x R^{\leftrightarrow}_{\mathfrak{G}} y$ 且 $x \leq_{\mathfrak{G}} z$，则存在 u 使得 $y \leq_{\mathfrak{G}} u$ 且 $z R^{\leftrightarrow}_{\mathfrak{G}} u$。

证明。必然存在一个标签序列 x_0，...，x_n，使得 $n \geq 0$，$x_0 = x$，$x_n = y$，并且对于每个 $0 \leq i \leq n-1$，$x_i R_{\mathfrak{G}} x_{i+1}$ 或 $x_{i+1} R_{\mathfrak{G}} x_i$。我们对 n 使用归纳法。如果 $n = 0$，即 $x = y$，那么 $u := z$ 即可。否则，根据归纳假设，存在 v，使得 $x_{n-1} \leq_{\mathfrak{G}} v$ 且 $z R^{\leftrightarrow}_{\mathfrak{G}} v$。如果 $x_n R_{\mathfrak{G}} x_{n-1}$，则根据 ($F_1$)-结构饱和，存在 u，使得 $x_n \leq_{\mathfrak{G}} u$ 且 $u R_{\mathfrak{G}} v$。如果 $x_{n-1} R_{\mathfrak{G}} x_n$，则根据 ($F_2$)-结构饱和，存在 u，使得

$x_n \leq_{\mathfrak{G}} u$ 且 $v R_{\mathfrak{G}} u$。无论哪种情况，都有 $y \leq_{\mathfrak{G}} u$ 且 $z R^{\leftrightarrows}_{\mathfrak{G}} u$。 □

命题 6.3. 对于分层的结构饱和序列 $\mathfrak{G}$，关系 $\leq$ 是其各层上的序关系。

证明. 为了证明自反性，任取一层 L。对于任意标签 $x \in L$，由 (≤rf)-结构饱和性可得 $x \leq_{\mathfrak{G}} x$。因此，根据定义 6.2，$L \leq L$。

为了证明传递性，设 $L_1 \leq L_2$ 和 $L_2 \leq L_3$ 分别是 L_1、L_2 和 L_3 的层。根据定义 6.2，对于某些标签 $x \in L_1$、y'、$y \in L_2$ 和 $z \in L_3$，有 $x \leq_{\mathfrak{G}} y'$ 和 $y \leq_{\mathfrak{G}} z$。由于 $y' R^{\leftrightarrows}_{\mathfrak{G}} y$，由 (F)-结构饱和性（命题 A.4）可知，存在一个标签 z'，使得 $y' \leq_{\mathfrak{G}} z'$ 且 $z' R^{\leftrightarrows}_{\mathfrak{G}} z$。后者意味着 $z' \in L_3$。由 (≤tr)-结构饱和性，$x \leq_{\mathfrak{G}} z'$。因此，根据定义 6.2，$L_1 \leq L_3$。

关于反对称性，设 $L_1 \leq L_2$ 和 $L_1 \neq L_2$。根据定义 6.2，存在标签 $x \in L_1$ 和 $x' \in L_2$，使得 $x \leq_{\mathfrak{G}} x'$。对于任意标签 $y \in L_1$ 和 $y' \in L_2$，我们有 $x R^{\leftrightarrows}_{\mathfrak{G}} y$ 和 $x' R^{\leftrightarrows}_{\mathfrak{G}} y'$。由于 $L_1 \neq L_2$，我们有 $x \neq x'$。因此，根据定义 6.2 的 2)，$y' \not\leq_{\mathfrak{G}} y$。由于对于任意 $y' \in L_2$和 $y \in L_1$都有 $y' \not\leq_{\mathfrak{G}} y$，所以不可能有 $L_2 \leq L_1$。 □

命题 6.7. 对于结构饱和序列 $\mathfrak{G}$，关系 $R_{\mathfrak{G}}$ 是一个序，$\leq_{\mathfrak{G}}$ 在其簇上是一个预序。如果 $\mathfrak{G}$ 是分层的，那么 $\leq_{\mathfrak{G}}$ 也是一个序。

证明。对于 $R_{\mathfrak{G}}$ 的自反性，考虑任意簇 C。对于任意标签 $x \in C$，根据 (Rrf)-结构饱和，$x R_{\mathfrak{G}} x$。因此，根据定义 6.6，$C R_{\mathfrak{G}} C$。对于 $R_{\mathfrak{G}}$ 的传递性，设 $C_1 R_{\mathfrak{G}} C_2$ 和 $C_2 R_{\mathfrak{G}} C_3$ 分别对应簇 C_1、C_2，以及 C_3。根据定义 6.6，$x R_{\mathfrak{G}} y$，并且对于某些标签 $x \in C_1$、y、$u \in C_2$和 $z \in C_3$，有 $u R_{\mathfrak{G}} z$。由于 $y R_{\mathfrak{G}} u$，根据 (Rtr)-结构饱和，$x R_{\mathfrak{G}} z$。因此，根据定义 6.6，$C_1 R_{\mathfrak{G}} C_3$。对于 $R_{\mathfrak{G}}$ 的反对称性，设 $C_1 R_{\mathfrak{G}} C_2$ 且 $C_2 R_{\mathfrak{G}} C_1$。根据定义 6.6，存在标签 x、$y \in C_1$ 和 x'、$y' \in C_2$，使得 $x R_{\mathfrak{G}} x'$ 且 $y' R_{\mathfrak{G}} y$。对于任意标签 $u \in C_1$ 和 $v \in C_2$，我们有 $u R_{\mathfrak{G}} x$，并且 $x' R_{\mathfrak{G}} v$，因此根据 (Rtr)-结构饱和，$u R_{\mathfrak{G}} v$。类似地，由于 $v R_{\mathfrak{G}} y'$ 和 $y R_{\mathfrak{G}} u$，$v R_{\mathfrak{G}} u$。由于对于所有 $u \in C_1$ 和 $v \in C_2$，$u R_{\mathfrak{G}} v$ 且 $v R_{\mathfrak{G}} u$，这两个簇中的所有标签关于 $R_{\mathfrak{G}} \cap R_{\mathfrak{G}}^{-1}$ 构成一个等价类，即 $C_1 = C_2$。

关于 $\leq_{\mathfrak{G}}$ 的自反性，考虑任意簇 C。对于每个标签 $y \in C$，根据（≤rf）-结构饱和性，$y \leq_{\mathfrak{G}} y$。因此，根据定义 6.6，$C \leq_{\mathfrak{G}} C$。关于 $\leq_{\mathfrak{G}}$ 的传递性，令 $C_1 \leq_{\mathfrak{G}} C_2$ 和 $C_2 \leq_{\mathfrak{G}} C_3$ 分别为簇 C_1、C_2 和 C_3。根据定义 6.6，对于每个 $z \in C_3$，存在 $y \in C_2$ 使得 $y \leq_{\mathfrak{G}} z$。接着，对于这个 y，存在 $x \in C_1$ 使得 $x \leq_{\mathfrak{G}} y$。根据（≤tr）- 结构饱和性，$x \leq_{\mathfrak{G}} z$ 对于这个 x。因此，根据定义 6.6，$C_1 \leq_{\mathfrak{G}} C_3$。所以，$\leq_{\mathfrak{G}}$ 是一个预序。

现在进一步假设 $\mathfrak{G}$ 是分层的。为证明 $\leq_{\mathfrak{G}}$ 的反对称性，令 $C_1 \leq_{\mathfrak{G}} C_2$ 和 $C_1 \neq C_2$，即 $C_1 \cap C_2 = \varnothing$。为证明 $C_2 \not\leq_{\mathfrak{G}} C_1$，我们任取一个标签 $x \in C_1$，并证明对于所有 $y \in C_2$ 都有 $y \not\leq_{\mathfrak{G}} x$。根据定义 6.6，对于每个 $y \in C_2$，存在某个标签 $x' \in C_1$ 使得 $x' \leq_{\mathfrak{G}} y$。由于它们属于同一个簇，所以有 $x' R^{\leftrightarrows}_{\mathfrak{G}} x$，$y R_{\mathfrak{G}} y$，因此根据（$R$rf）-结构饱和性，有 $y R^{\leftrightarrows}_{\mathfrak{G}} y$；又由于它们来自不相交的簇，有 $x' \neq y$。因此，根据定义 6.2，有 $y \not\leq_{\mathfrak{G}} x$。

由于 y 是任意选取的，因此可推出 $C_2 \not\leq_{\mathfrak{G}} C_1$，并且 $\leq_{\mathfrak{G}}$ 是一个序。 □

D. 第 7 节中的证明

为简便起见，当 $w \leq v$ 时，我们称 w 是 v 的过去，并称 v 是 w 的未来。

引理 7.2。以下所有陈述均成立：

a) 如果集合 $\mathfrak{S}$ 是稳定的，并且 $\mathfrak{S} \rightsquigarrow_{\mathsf{s}} \mathfrak{S}'$，那么 $\mathfrak{S}'$ 是稳定的。b) 在有限集合 $\mathfrak{S}$ 上，重写关系 s 是终止的。c) 集合 $\mathfrak{S}$ 是半饱和的，当且仅当它是稳定的，并且关于 s 处于范式。

证明。为证明 a)，假设集合 $\mathfrak{S}$ 是有限的，且所有序列式 $\mathfrak{G} \in \mathfrak{S}$ 都是稳定的，即具有树分层和树簇化性质（特别是分层的并且结构饱和的），且具有良好内部层。$\mathfrak{S}'$ 的大小至多比 $\mathfrak{S}$ 大一个序列式，因此，$\mathfrak{S}'$ 也是有限的。由于来自 $\mathfrak{S}$ 的序列式的内部层中的所有标签都是良好的，所以所有新添加的带标签公式的标签，无论是标签 x 本身，还是从 x 可由 $R_{\mathfrak{G}}$ 访问的标签，都必须位于某个来自 $\mathfrak{S}$ 的序列式的最顶层。鉴于该序列式是分层的，所有新带标签公式的标签都没有非自反未来，因此对它们而言，(mon•) 是平凡的。由于重写既不引入新标签，也不添加关系原子，结构饱和的所有其他条件，以及树分层和树簇化性质，仍然成立。由于没有带标签公式被移除，且所有序列式的内部层都保持不变，因此内部层仍然是良好的（见注 6.10）。因此，$\mathfrak{S}'$ 中的所有序列式都是稳定的，$\mathfrak{S}'$ 本身也是如此。

为证明 b），注意到当且仅当公式尚未出现在 $\mathfrak{G}$ 中时，每次重写才会向序列式 $\mathfrak{G} \in \mathfrak{S}$ 添加一个带标签公式。此外，添加的公式都是序列式中公式的子公式，且不会引入新标签。由于每个 $\mathfrak{G}$ 都只包含有限多个公式和有限多个标签，因此只有有限多个公式能够添加到序列式中。因此，如果 $\mathfrak{S}$ 中的序列式只有有限多个，关系 s 就是终止的。

最后，为证明 c），先假设 $\mathfrak{S}$ 是半饱和的。那么根据定义，$\mathfrak{S}$ 是有限的，$\mathfrak{S}$ 中的所有序列式都是稳定的，特别是都具有良好内部层，并且其最顶层是朴素满足的。由于这些有限多个序列式都是稳定的，$\mathfrak{S}$ 是稳定的。由于它们所有的层都（至少）是朴素满足的，因此没有重写步骤适用于 $\mathfrak{S}$，所以相对于 s，$\mathfrak{S}$ 处于范式。反过来，假设 $\mathfrak{S}$ 是稳定的，且相对于 s 处于范式。那么 $\mathfrak{S}$ 是一个由稳定序列式构成的有限集合。由于没有重写适用于 $\mathfrak{S}$ 中的任何序列式，这些序列式的所有（最顶层）层都是朴素满足的。因此，$\mathfrak{S}$ 中有限多个序列式都是半饱和的，$\mathfrak{S}$ 本身也如此。 □

以下两个引理未出现在正文中，但证明引理 7.6 时需要用到。

引理 A.5。以下命题成立：

i) 如果序列 $\mathfrak{G}$ 是半饱和的，且 $\mathfrak{G} \rightsquigarrow_{\diamond} \mathfrak{G}'$，那么序列 $\mathfrak{G}'$ 是稳定的。

ii) 如果序列 $\mathfrak{G}$ 是半饱和的，并且 $\mathfrak{G} \rightsquigarrow_{\diamond} \mathfrak{G}'$ 是根据选项 1 得到的，那么序列 $\mathfrak{G}'$ 是半饱和的。

证明。为证明 i)，假设序列 $\mathfrak{G}$ 是半饱和的，特别地，它是稳定的（即具有结构饱和性、树分层性，并且具有良好内部层的树簇化性）。令 $\mathfrak{G} \rightsquigarrow_{\diamond} \mathfrak{G}'$。对于定义 7.5 中的任一选项，不满足的标签 y 只能位于 $\mathfrak{G}$的最顶层。选项 1 中的两个标签 x 或选项 2 中的 z都属于同一个最顶层。因此，在这两种情形下，内部层中的标签以及大多数带标签公式都不受影响，因而根据注 6.10，它们仍然是良好的。对于选项 2，不会移除任何公式或标签，因此内部层中的所有白色 $A \supset B^{\circ}$ 和 $\Box A^{\circ}$ 公式仍然是良好的。对于选项 1，内部层中的公式 $A \supset B^{\circ}$ 仍然是良好的，因为 y 没有过去。最后，某个内部层中的 $\mathfrak{G}, k{:}\Box A^{\circ}$ 可能由于 $\mathfrak{G}, y{:}A^{\circ}$, $k \leq_{\mathfrak{G}} w$以及 $w R_{\mathfrak{G}} y$而是良好的。但是由于 $x \sim y$, $\mathfrak{G}', x{:}A^{\circ}$也成立。注意， $w \neq y$ ，因为 y 没有过去。因此，$k \leq_{\mathfrak{G}'} w$ 和 $w R_{\mathfrak{G}'} x$ 意味着 $\mathfrak{G}', k{:}\Box A^{\circ}$ 仍然是良好的。对于选项 1，不存在新标签，并且 y 在 $\mathfrak{G}$中没有过去，因此 $\leq$关系原子唯一的变化是 $y \leq y$ 变为 $x \leq x$；对于选项 2，唯一的新标签是 z ，其中 $z \leq z$ 是唯一新的 $\leq$关系原子。无论哪种情形，($\leq$rf)-和 ($\leq$tr)-结构饱和性都得到保证，并且非自反$\leq$关系原子的集合保持不变。后者意味着， $\mathfrak{G}'$ 的 (mon)-、(F_1)-和 (F_2)-结构饱和性由 $\mathfrak{G}$的相同性质推出。(Rtr)-结构饱和性得到明确保证。对于选项 2，(Rrf)-结构饱和性也是如此，因为新增标签 z需要这一性质。这就完成了 $\mathfrak{G}'$ 具有结构饱和性的证明。由于重写不影响序列的层结构，$\mathfrak{G}'$ 具有树分层性立即由 $\mathfrak{G}$的相同性质推出。为证明 $\mathfrak{G}'$ 是树簇化的，我们分别考察两个选项。在定义 7.5 的选项 1 中，重写将标签 y 替换为标签 x ，并应用传递闭包。考察为新的 $R_{\mathfrak{G}'}$链接提供依据的最短链，并结合 $\mathfrak{G}$和 $\mathfrak{G}'$ 的结构饱和性以及 $x R_{\mathfrak{G}} y$，可得：

$$u R_{\mathfrak{G}'} x \implies u R_{\mathfrak{G}} y; \tag{5}$$

$$x R_{\mathfrak{G}'} v \implies x R_{\mathfrak{G}} v; \tag{6}$$

$$u R_{\mathfrak{G}'} v \text{ but not } u R_{\mathfrak{G}} v \implies u R_{\mathfrak{G}} y \& x R_{\mathfrak{G}} v. \tag{7}$$

Let C'_x 是 $\mathfrak{G}'$ 中包含 x的簇。对于 $\mathfrak{G}$中的任意簇C， $y \in C$ 和 $C \setminus \{y\} \subseteq C'_x$ ，或 $y \notin C$ 以及$C \subseteq C'_x$ 或 C 中的任一个，仍然是 $\mathfrak{G}'$中的簇。 $y \in C$的情形很简单。假设 $y \notin C$，它不是 $\mathfrak{G}'$中的簇。由于所有与 y 无关的 $R_{\mathfrak{G}}$-链接都保留在 $\mathfrak{G}'$中，簇 C只能扩大。因此，必定存在某些标签 $u \in C$ 和$v \notin C$，使得 $u R_{\mathfrak{G}'} v$、 $v R_{\mathfrak{G}'} u$，但要么 $u R_{\mathfrak{G}} v$不成立，要么 $v R_{\mathfrak{G}} u$不成立。根据 (7)，在前一种情形下， $u R_{\mathfrak{G}} y$ 和 $x R_{\mathfrak{G}} v$，从而 $u R_{\mathfrak{G}'} x$ 和 $x R_{\mathfrak{G}'} v$；而在后一种情形下， $v R_{\mathfrak{G}} y$ 和$x R_{\mathfrak{G}} u$，从而 $v R_{\mathfrak{G}'} x$ 和 $x R_{\mathfrak{G}'} u$。无论哪种情形，根据 $\mathfrak{G}'$的结构饱和性， $u \in C'_x$，且 $C \subseteq C'_x$。为证明 $\mathfrak{G}'$ 是树簇化的（定义 7.5 的选项 1），我们从 $\mathfrak{G}'$中取三个簇C_1、 C_2和 C'，使得 $C_1 R_{\mathfrak{G}'} C'$和 $C_2 R_{\mathfrak{G}'} C'$，并证明 C_1 和 C_2 是 $R_{\mathfrak{G}'}$-可比较的。如果任意两个其中任意两个相等，这显然成立，因此假设它们两两不同。如果它们都不是 C'_x，那么它们也是 $\mathfrak{G}$中的簇，只需证明存在 $\mathfrak{G}$中的某个簇 C^* ，使得 $C_1 R_{\mathfrak{G}} C^*$ 和 $C_2 R_{\mathfrak{G}} C^*$。于是，由于 $\mathfrak{G}$ 是树簇化的且该连接在 $\mathfrak{G}'$中仍然存在， C_1 和 C_2 是 $R_{\mathfrak{G}}$可比较的，这意味着它们也是 $R_{\mathfrak{G}'}$可比较的。如果 $C_1 R_{\mathfrak{G}} C'$ 和 $C_2 R_{\mathfrak{G}} C'$都成立，则 $C^* = C'$。如果 $C_1 R_{\mathfrak{G}} C'$ 和 $C_2 R_{\mathfrak{G}} C'$都不成立，则由 (7) 可得 $C_1 R_{\mathfrak{G}} C_y$和 $C_2 R_{\mathfrak{G}} C_y$ ，其中 $\mathfrak{G}$ 中包含 y 的簇为 C_y 。因此$C^* = C_y$。最后，若不失一般性地有 $C_1 R_{\mathfrak{G}} C'$ 但没有 $C_2 R_{\mathfrak{G}} C'$，则根据 (7)，$C_2 R_{\mathfrak{G}} C_y$ 和 $C_x R_{\mathfrak{G}} C'$ ，其中 $\mathfrak{G}$ 中包含 x的簇为 C_x 。由于 $\mathfrak{G}$ 是树簇化的， $C_x R_{\mathfrak{G}} C_1$ 或 $C_1 R_{\mathfrak{G}} C_x$。在后一种情形下， $C_1 R_{\mathfrak{G}} C_y$，因此 $C^* = C_y$。在前一种情形下，我们直接由 $C_2 R_{\mathfrak{G}} C_y$ 和 $C_x R_{\mathfrak{G}} C_1$推出 $R_{\mathfrak{G}'}$连接。接下来考虑三个簇中恰有一个是 C'_x 、另外两个是 $\mathfrak{G}$的簇的情形。如果 $C' = C'_x$，那么根据 (5)， $C_1 R_{\mathfrak{G}} C_y$ 和 $C_2 R_{\mathfrak{G}} C_y$，因此使用上述关于 $C^* = C_y$ 的论证即可。最后，若不失一般性地有 $C_1 = C'_x$，则根据 (6)， $C_x R_{\mathfrak{G}} C'$ 。如果 $C_2 R_{\mathfrak{G}} C'$，则由于 $\mathfrak{G}$ 是树簇化的，C_x 与 C_2 是 $R_{\mathfrak{G}}$可比较的，而C'_x 与 C_2是 $R_{\mathfrak{G}'}$可比较的。如果 $C_2 R_{\mathfrak{G}'} C'$ 但没有 $C_2 R_{\mathfrak{G}} C'$，则由 (7) 得到 $C_2 R_{\mathfrak{G}} C_y$ ，并且 $C_2 R_{\mathfrak{G}'} C'_x$。定义 7.5 的选项 2 中的情形要简单得多：在那里，当且仅当 $C = \{z\}$ 或$C R_{\mathfrak{G}} C_y$时，才会向 $\mathfrak{G}$ 和 $C R_{\mathfrak{G}'} \{z\}$ 的簇中加入一个新的簇 $\{z\}$。唯一需要考虑的非平凡新情形是$C' = \{z\}$、 $C_1 \neq \{z\}$和 $C_2 \neq \{z\}$，在此情形下，$C_1 R_{\mathfrak{G}} C_y$和 $C_2 R_{\mathfrak{G}} C_y$以及 C_1 和 C_2 是 $R_{\mathfrak{G}'}$可比较的，因为它们是 $R_{\mathfrak{G}}$可比较的。这就完成了 $\mathfrak{G}'$ 为树簇化且总体稳定的证明。

为证明 ii)，由于 i) 中刚刚证明了大部分条件，因此只需进一步证明 $\mathfrak{G}'$ 的所有最顶层都朴素满足。由于选项 1 既不添加新公式，也不添加新标签，大多数类型公式的（朴素）幸福性都继承自 $\mathfrak{G}$。唯一的例外是 $\Box^{\bullet}$-公式和 $\diamond^{\circ}$-公式。两者的论证相同。下面以 $\Box^{\bullet}$-公式为例。假设 $u R_{\mathfrak{G}'} v$ 且 $\mathfrak{G}', u{:}\Box C^{\bullet}$。由于选项 1 不添加公式，因此也有 $\mathfrak{G}, u{:}\Box C^{\bullet}$。有两种可能：要么 $u R_{\mathfrak{G}} v$ 已在 $\mathfrak{G}$ 中，要么它是为确保 $\mathfrak{G}'$的结构饱和而添加的。在前一种情形下，根据 $\mathfrak{G}$中这一层的朴素幸福性，有 $\mathfrak{G}', v{:}C^{\bullet}$ 和 $\mathfrak{G}', v{:}\Box C^{\bullet}$ 。在后一种情形下，我们有 $u R_{\mathfrak{G}'} v$ 但没有 $u R_{\mathfrak{G}} v$，因此根据 (7)，有 $u R_{\mathfrak{G}} y$ 和 $x R_{\mathfrak{G}} v$。因此，根据 $\mathfrak{G}$中这一层的朴素幸福性，有 $\mathfrak{G}, y{:}\Box C^{\bullet}$ ，所以有 $\mathfrak{G}, x{:}\Box C^{\bullet}$ ，因为 $x \sim y$ 在$\mathfrak{G}$中；此外，由于朴素幸福性，还有 $\mathfrak{G}, v{:}C^{\bullet}$ 和 $\mathfrak{G}, v{:}\Box C^{\bullet}$。在 $\mathfrak{G}'$ 中， $C^{\bullet}$ 和 $\Box C^{\bullet}$ 都仍位于 v 。

□

引理 A.6。 设 $\mathfrak{G}$ 是一个半饱和序列，设 $\mathfrak{G} \rightsquigarrow_{\diamond} \mathfrak{G}'$，并设 $\mathfrak{S}^*$ 是 $\{\mathfrak{G}'\}$的半饱和。则所有 happy $\diamond^{\bullet}$- 公式在从 $\mathfrak{G}$ 到 $\mathfrak{S}^*$的转换中都得到保留，即对于所有序列 $\mathfrak{G}^* \in \mathfrak{S}^*$ 以及 u 中出现的所有标签 $\mathfrak{G}^*$，如果 $\mathfrak{G}, u{:}\ \diamond C^{\bullet}$ 是 happy，那么 $\mathfrak{G}^*$也同样如此，$u{:}\ \diamond C^{\bullet}$。

证明。注意，半饱和不会移除公式或标签，也不会添加 R-关系原子。因此，选项 2 中的半饱和不会破坏任何 $\diamond^{\bullet}$-公式的幸福性；只需证明， $\mathfrak{G}$ 中没有幸福的 $\diamond^{\bullet}$-公式会在 $\mathfrak{G}'$ 中变得不满足。对于选项 2，这显然成立，因为

同样，没有任何内容会被移除。对于选项 1，唯一的问题与移除 y 有关；若$\mathfrak{G}$、u: $\Diamond C^\bullet$ 因 $uR_{\mathfrak{G}}y$ 和 $\mathfrak{G}$、y: $C^\bullet$而幸福，则可能出现这种情况。然而在此情形下，$uR_{\mathfrak{G}'}x$ 和 $\mathfrak{G}'$、x: $C^\bullet$ 因为 $x \sim y$ 在 $\mathfrak{G}$中，所以幸福性得以保持。 □

引理 7.6。以下所有陈述均成立：

a) 如果集合 $\mathfrak{S}$ 是半饱和的，并且 $\mathfrak{S} \rightrightarrows_\diamond \mathfrak{S}'$，那么 $\mathfrak{S}'$ 是半饱和的。b) 重写关系 $\diamond$ 是终止的。c) 集合 $\mathfrak{S}$ 是饱和的，当且仅当它是半饱和的，并且关于 $\diamond$ 处于范式。

证明。为证明 a)，假设集合 $\mathfrak{S}$ 是有限的，并且 $\mathfrak{S}$ 中的所有序列式都是半饱和的（特别是稳定的）。注意到，对于某个 $\mathfrak{G} \rightsquigarrow_\diamond \mathfrak{G}'$，$\mathfrak{S}' = (\mathfrak{S} \setminus \{\mathfrak{G}\}) \cup \mathfrak{S}^*$，其中 $\mathfrak{S}^*$ 如引理 A.6 中所述。在选项 1 中，我们将一个半饱和序列式 $\mathfrak{G}$ 替换为另一个序列式 $\mathfrak{G}'$，根据引理 A.5.ii)，后者是半饱和的。因此，所得集合是半饱和的。在选项 2 中，根据引理 A.5.i)，序列式 $\mathfrak{G}'$ 是稳定的。根据引理 7.2，集合 $\mathfrak{S}^*$ 是半饱和的。因此，$\mathfrak{S}'$ 也是如此。

为证明 b)，考虑集合序列 $\mathfrak{S} \rightrightarrows_\diamond \mathfrak{S}_1 \rightrightarrows_\diamond \mathfrak{S}_2 \rightrightarrows_\diamond \dots$。我们将该序列中出现的所有标签分为旧标签（即出现在 $\mathfrak{S}$ 中的标签）和新标签（即之后引入的标签）。我们也根据标签将不满足 $\Diamond^\bullet$公式分为旧公式和新公式。旧的不满足 $\Diamond^\bullet$公式的数量永远不会增加，并且每次 3 转换都会至少使一个 $\Diamond^\bullet$公式（旧的或新的）变为满足。因此，只需证明不可能无限地继续创建新标签。利用树聚类性，我们将新标签 x （来自某个集合中的序列的最顶层）的新簇深度定义为满足以下条件的最短链$oR_{\mathfrak{G}}x_1R_{\mathfrak{G}} \dots R_{\mathfrak{G}}x_nR_{\mathfrak{G}}x$ 的长度： o 是旧标签，而标签 x_0、……、x_n 是新标签。特别地，任何旧标签 o 的新簇深度都是 0。为证明终止性，我们考虑所有来自 $\mathfrak{S}_i$ 的新的不满足 $\Diamond^\bullet$公式的新簇深度的多重集上的 Derschowitz–Manna 型序。我们将这样的整数多重集记为 $\mathrm{unh}_\diamond(\mathfrak{S}_i)$，并将其与来自 $\mathfrak{S}_i$ 的旧的不满足 $\Diamond^\bullet$公式的数量配对。令 $\mathfrak{S}_i \rightrightarrows_\diamond \mathfrak{S}_{i+1}$ 基于标签 y 处一个新簇深度为 k 的不满足$\Diamond A^\bullet$ 公式执行。如果 $k = 0$，即 y 是旧标签，则旧的不满足 $\Diamond^\bullet$公式的数量会减少。通过对 i 进行归纳，容易证明，对于新的 y，即对于 $k > 0$，标签 y 形成一个单元素簇 $\{y\}$，y 的任何非平凡 R子节点都必须是相应最顶层中的 R叶节点。因此，由于根据 A.6，y 之外的所有 $\Diamond^\bullet$公式都保持满足性，y 中的所有公式都会消失，而其他包含不满足 $\Diamond^\bullet$公式的簇会保持其新簇深度，多重集 $\mathrm{unh}_\diamond(\mathfrak{S}_{i+1}) \subsetneq \mathrm{unh}_\diamond(\mathfrak{S}_i)$，也就是说，至少一个不满足 $\Diamond^\bullet$公式（即标签 y 处的 $\Diamond A^\bullet$ ）消失了，没有新增不满足 $\Diamond^\bullet$公式，也没有不满足$\Diamond^\bullet$公式的新簇深度降低。或者，如果 $\mathfrak{S}'$ 通过选项 2 获得，那么同样根据 A.6，出现在 $\mathfrak{S}_i$ 中的标签不会产生额外的不满足 $\Diamond^\bullet$公式，并且其中一些不满足公式可能变为满足。然而，半饱和可能会在新创建的标签 z 中产生新的不满足 $\Diamond^\bullet$公式，该标签具有簇深度 $k+1$ （容易看出其他标签不受半饱和影响）。

因此，$\mathrm{unh}_\diamond(\mathfrak{S}_{i+1}) \subseteq \mathrm{unh}_\diamond(\mathfrak{S}_i) \setminus \{k\} \cup \{k+1, \dots, k+1\}$，其中新增 $k+1$ 的重数取决于半饱和后 z 中包含多少个不满足 $\Diamond^\bullet$公式。为确保终止性，还需注意不满足 $\Diamond^\bullet$公式的新簇深度存在一个全局上界。事实上，令 M 为 $\mathfrak{S}$ 的子公式数量。我们证明，没有不满足 $\Diamond^\bullet$公式的新簇深度能够超过 $2^M + 1$。这是因为根据子公式性质，分配给一个标签的不同公式集合最多有 2^M 个。选项 2 中创建的所有新标签 z 都没有过去，并且此类 z 的子节点中的 $\Diamond^\bullet$公式在 z 几乎满足之前不会用于饱和。考虑两个满足 $z_1R_{\mathfrak{G}}z_2$ 的新标签 z_1 和 z_2。假设选择了 z_2 中的一个不满足 $\Diamond A^\bullet$ 公式用于下一次饱和步骤。那么 $\{z_2\}$ 是一个单元素簇，其所有非平凡父节点（包括 z_1 ）都几乎满足。如果 $z_2 \sim z_1$，则选项 1 会要求将 z_2 与 z_1 识别。因此，为使 z_2 保持为独立簇，它必须不同于其新标签的严格前置簇中的所有标签。于是，对于任意一列互不相同的新标签簇 $C_1R_{\mathfrak{G}} \dots R_{\mathfrak{G}}C_k$，都必须存在标签 $z_1R_{\mathfrak{G}} \dots R_{\mathfrak{G}}z_k$，使得对于任意 $j \neq l$ 都有 $z_j \not\sim z_l$。这意味着，连续创建至多 $k \geq 2^M$ 个新标签簇后，（半饱和的）新标签 z 将等价于某个先前创建的标签，而选项 1 会阻止继续延长该序列。由于所有新的不满足$\Diamond^\bullet$公式都会朝着新簇深度增加的方向推进，而新簇深度不能超过全局上界 $2^M + 1$，且旧的不满足 $\Diamond^\bullet$公式的数量只会减少，我们得出结论：3 终止。

要证明 c)，注意到根据定义，$\mathfrak{S}$ 饱和的含义是：它是半饱和的，并且其序列式的所有最顶层都是几乎快乐的。当且仅当某一层是朴素满足的，且其中出现的所有 $\Diamond A^\bullet$-公式都是满足的时，该层才是几乎快乐的。只有当半饱和集具有一个不满足的 $\Diamond A^\bullet$-公式（位于其最顶层）时，才能对其应用饱和 $\diamond$。因此，$\mathfrak{S}$ 饱和，当且仅当它是半饱和的，并且关于 $\diamond$ 处于范式。 □

引理 7.15。如果 $\mathfrak{G}$ 饱和，则 $\mathfrak{G}\uparrow$ 稳定。

证明。假设 $\mathfrak{G}$ 是饱和的。那么它是稳定的，即结构饱和、树分层，并且树簇化且具有良好内部层；此外，其最顶层几乎快乐。根据定义，$\mathfrak{G}\uparrow$ 由 $\mathfrak{G}$ 构成，并且对于每个不满足公式 x: F°（其中 $F = A \supset B$ 或$F = \Box A$ 出现在 $\mathfrak{G}$ 的某个未模拟最顶层 L中），我们都添加一个最顶层ˆ层 L（这些原先的最顶层会在 $\mathfrak{G}\uparrow$中变成内部层）。如果ˆ$\{x\}$ 是一个单元素簇，那么这样的层 L 就是 L 的副本（若 $F = \Box A$，还会额外添加标签 z ）；如图 5 所示，如果ˆx 属于非单元素簇 C_x，层 L 还会额外包含 C_x 的副本和标签ˆx （若 $F = \Box A$，还会额外添加标签 z）。由于每个新增层都是最顶层，因此可以分别验证每一层都满足所有条件。如前所述，构造 7.8（以及针对附加标签 z的构造 7.9）明确保证 $\mathfrak{G}\uparrow$ 是分层的，并且 $\mathfrak{G}\uparrow$ 的新层ˆL 满足（Rrf）-和（Rtr）-

结构饱和，并且 $\mathfrak{G}\uparrow$ 满足（$\leq$rf）-、（$\leq$tr）-、（mon$^\bullet$）-、（F_1）-和（F_2）-结构饱和。$\mathfrak{G}\uparrow$ 是树分层的这一事实，源于 $\mathfrak{G}$ 是树分层的，并且对每个新增的最顶层 L，都有 $L' < L$ 当且仅当$L' \leq L$。在 x 属于单元素簇的情形下，证明$\hat{}L$ 是树簇化的显而易见，因为 L 的 R-结构要么与树簇化的 L 同构，要么是在其基础上添加一个相对于 R为叶节点的单元素簇 $\{z\}$ 而得。在 x 属于非单元素簇 C_x的情形下，从构造 7.8 和构造 7.9 很容易看出，L 被划分为以下簇：$\hat{}C'_x = \{\hat{}x'_j \mid x_j \in C_x\}$、$\hat{}C''_x = \{\hat{}x''_j \mid x_j \in C_x\}$、$\{\hat{}x\}$，我们称之为 x-簇；此外，对 L的每个簇 $C \neq C_x$，还有簇 $C := \{\hat{}y_j \mid y_j \in C\}$，我们称之为 y-簇；若 $F = \Box A$，还额外有簇 $\{z\}$（见图 5）。同样容易看出，对于任意的$\hat{}$ y-簇 C 和 C^*：

a) $C R_{\mathfrak{G}\uparrow} C^*$ 当且仅当 $C R_{\mathfrak{G}} C^*$，b) $C R_{\mathfrak{G}\uparrow} \hat{}C'_x$ 当且仅当 $C R_{\mathfrak{G}\uparrow} \hat{}C''_x$ 当且仅当 $C R_{\mathfrak{G}\uparrow} \{\hat{}x\}$ 当且仅当 $C R_{\mathfrak{G}} C_x$（在存在 z的情况下，$\hat{}$当且仅当 $C R_{\mathfrak{G}\uparrow} \{z\}$），c) $\hat{}C'_x R_{\mathfrak{G}\uparrow} \hat{}C$ 当且仅当 $\hat{}C''_x R_{\mathfrak{G}\uparrow} \hat{}C$ 当且仅当 $\{\hat{}x\} R_{\mathfrak{G}\uparrow} \hat{}C$ 当且仅当 $C_x R_{\mathfrak{G}} \hat{}C$（但绝不会是 $\{z\} R_{\mathfrak{G}\uparrow} \hat{}C$），$\hat{}\hat{}\hat{}\hat{}$ d) $C'_x R_{\mathfrak{G}\uparrow} C''_x$、$C'_x R_{\mathfrak{G}\uparrow} \{\hat{}x\}$、$\{\hat{}x\} R_{\mathfrak{G}\uparrow} C''_x$ 是 x-簇中仅有的非自反关系，$\hat{}$ e) 在存在 z的情况下：$C'_x R_{\mathfrak{G}\uparrow} \{z\}$、$\{\hat{}x\} R_{\mathfrak{G}\uparrow} \{z\}$和$\{z\} R_{\mathfrak{G}\uparrow} \{z\}$ 是涉及$\{z\}$的仅存关系。

从$\hat{}L$ 中任取满足 $C_1 R_{\mathfrak{G}\uparrow} C_3$ 和 $C_2 R_{\mathfrak{G}\uparrow} C_3$的三个簇 C_1、C_2 和 C_3。我们需要证明 C_1 和 C_2 在 $\mathfrak{G}\uparrow$中是可比较的。略去一些平凡情形，假设这三个簇两两不同。根据这些簇是否为 x-簇、y-簇等，我们区分不同情形。$C_3 = \{z\}$ 的情形可归结为$C_3 = \{\hat{}x\}$，因为 $C_i R_{\mathfrak{G}\uparrow} \{z\}$ 当且仅当 $C_i R_{\mathfrak{G}\uparrow} \{\hat{}x\}$ 或 $C_i = \{z\}$，后一种情形是平凡的。所有其他涉及 $\{z\}$ 的情形也都是平凡的。当 C_1 和 C_2 都是 x-簇时，该情形是平凡的，因为它们在 $\mathfrak{G}\uparrow$中两两可比较。因此，只需考虑$\hat{}$中的簇均非 $\{z\}$，且不妨设 C_1 是一个 y-簇的情形。$L\hat{}$中的所有非 $\{z\}$ 簇在 L中都有起源簇：对于一个 y-簇 C，其起源簇为 C；对于每个 x 簇，其起源簇为 C_x。我们分别将 C_1、C_2和 C_3 簇的起源称为C_1^o、C_2^o和 C_3^o。考虑 C_2 和 C_3 分别为 x-簇或 y-簇的四种可能情形，可以看出$C_1^o R_{\mathfrak{G}} C_3^o$和 $C_2^o R_{\mathfrak{G}} C_3^o$。由于 $\mathfrak{G}$ 是树簇化的，C_1^o 和 C_2^o 在 $\mathfrak{G}$中是可比较的。这意味着在四种情形中的每一种情形下，C_1和 C_2 在 $\mathfrak{G}\uparrow$ 中都是可比较的。由此证明了 $\mathfrak{G}\uparrow$ 是树簇化的。

最后，由于 $\mathfrak{G}$ 是饱和的，其所有最顶层都是几乎快乐的，也就是说，所有不属于 $A \supset B^\circ$或 $\Box A^\circ$ 形式的公式都是快乐的。通过添加新的最顶层而不移除任何公式或任何标签，我们使它们全部变得快乐，同时不会使 $\mathfrak{G}$ 中的任何其他公式变得不满足（这里我们依赖于 $\mathfrak{G}$的幸福性的局部表述，x: $A \supset B^\bullet$见定义 5.3）。因此，$\mathfrak{G}\uparrow$ 的所有内部层都是快乐的，并且 $\mathfrak{G}\uparrow$是稳定的。 □

引理 7.19。成立以下结论：

a) 如果 $\mathfrak{G}$ 是饱和的，并且 $\mathfrak{G} \rightrightarrows_{\circlearrowleft} \mathfrak{G}'$，那么 $\mathfrak{G}'$ 是饱和的。b) 重写关系 $\circlearrowleft$ 是终止的

。证明。为证明 a)，假设 $\mathfrak{G}$ 已饱和。考虑一个满足 $\mathfrak{G} \rightsquigarrow_{\circlearrowleft} \mathfrak{G}'$ 的序列 $\mathfrak{G} \in \mathfrak{G}$。那么，$\mathfrak{G}$ 包含一个不满足的 R- 三角循环或一个不满足的 U- 三角循环。通过将标签 s 替换为标签 t 的所有出现，然后对 R 施加传递性闭包（定义 7.18），可由 $\mathfrak{G}$ 得到序列 $\mathfrak{G}'$。根据假设，$\mathfrak{G}$ 是稳定的，即它是结构饱和的、树分层的，并且具有良好内部层的树簇化结构。无论 $\mathfrak{G}$ 中出现哪一种三角循环，标签 s 和 t 都属于该序列的同一个最顶层。因此，内部层中的标签和带标签公式都不会受到重写的影响，并且根据注 6.10，对于大多数公式而言，其幸福性得以保持。有两类公式可能是例外。我们分别考察它们。对于第一类，如果某个内部层中的 $\mathfrak{G},k\colon A \supset B^\circ$ 因为 $k \leq_{\mathfrak{G}} t$, $\mathfrak{G},t\colon A^\bullet$ 且 $\mathfrak{G},t\colon B^\circ$ 而具有幸福性，那么 $\mathfrak{G}',s\colon A^\bullet$ 和 $\mathfrak{G}',s\colon B^\circ$ 都因为 $s \sim t$ 而具有幸福性。因此，$\mathfrak{G}',k\colon A \supset B^\circ$ 仍然因为 $k \leq_{\mathfrak{G}'} s$ 而具有幸福性。对于另一类，考虑某个内部层中的 $\mathfrak{G},k\colon \Box A^\circ$ 因为 $\mathfrak{G},v\colon A^\circ$, $k \leq_{\mathfrak{G}} w$ 而具有幸福性的情形，其中 $w R_{\mathfrak{G}} v$ 表示标签 w 和 v 中有一个或两个是 t。如果只有 $v = t$，那么 $\mathfrak{G}',s\colon A^\circ$ 因为 $s \sim t$ 而成立，因此 $k \leq_{\mathfrak{G}'} w$ 和 $w R_{\mathfrak{G}'} s$ 使 $\mathfrak{G}',k\colon \Box A^\circ$ 具有幸福性。如果只有 $w = t$，那么 $k \leq_{\mathfrak{G}'} s$ 和 $s R_{\mathfrak{G}'} v$ 使 $\mathfrak{G}',k\colon \Box A^\circ$ 具有幸福性。最后，如果 $v = w = t$ 都成立，那么 $\mathfrak{G}',s\colon A^\circ$ 因为 $s \sim t$，并且 $k \leq_{\mathfrak{G}'} s$ 与一个 $s R_{\mathfrak{G}'} s$ 使 $\mathfrak{G}',k\colon \Box A^\circ$ 具有幸福性。在所有情形下，内部层中的所有公式都保持幸福。重写不会引入新标签，而 $\leq$-关系原子中唯一的变化是 $t \leq t$ 变为 $s \leq s$，并且对于任意 $k \neq t$，若 $k \leq_{\mathfrak{G}} t$，则 $k \leq_{\mathfrak{G}'} s$。$\mathfrak{G}'$ 的（Rtr）- 结构饱和性得到明确施加，而（$\leq$rf）- 和（Rrf）-结构饱和性不受移除标签 t 的影响。对于（$\leq$tr）- 结构饱和性，唯一非平凡的情形是 $w \leq_{\mathfrak{G}'} k$ 和 $k \leq_{\mathfrak{G}'} s$，因为 $k \leq_{\mathfrak{G}} t$，其中 $k \neq s$。那么 k 和 w 属于未受影响的层，因此 $w \leq_{\mathfrak{G}} k$。于是，根据 $\mathfrak{G}$ 的（$\leq$tr）- 结构饱和性，$w \leq_{\mathfrak{G}} t$ 成立，而 $w \leq_{\mathfrak{G}'} s$ 根据构造成立。（mon$^\bullet$）-结构饱和性由 $s \sim t$ 得出。最后，对于 $\mathfrak{G}'$ 的（F_1）-和（F_2）-结构饱和性，最非平凡的情形是，对于二者之一，$k R_{\mathfrak{G}'} w$ 和 $k \leq_{\mathfrak{G}'} s$，因为 $k \leq_{\mathfrak{G}} t$，其中 $k \neq s$。同样，k 和 w 属于未受影响的层，因此 $k R_{\mathfrak{G}} w$。根据 $\mathfrak{G}$ 的同一结构饱和性质，存在一个标签 z，使得 $w \leq_{\mathfrak{G}} z$ 和 $t R_{\mathfrak{G}} z$。如果 $t \neq z$，那么 $s R_{\mathfrak{G}'} z$ 和 $w \leq_{\mathfrak{G}'} z$，因此同一个标签 z 满足这些条件。如果 $t = z$，那么 $w \leq_{\mathfrak{G}'} s$；这与 $s R_{\mathfrak{G}'} s$ 一起意味着 s 满足这些条件。剩余情形是对称的。由此证明 $\mathfrak{G}'$ 是结构饱和的。由于重写不影响该序列的层结构，我们立即得出 $\mathfrak{G}'$ 是树分层的。证明 $\mathfrak{G}'$ 是树簇化的过程，与引理 7.6 的情形 a）在定义 7.5 的选项 1 下的证明相同。由此证明 $\mathfrak{G}'$ 是稳定的。根据假设，$\mathfrak{G}$ 是饱和的。这意味着 $\mathfrak{G}$ 的所有最顶层都是几乎快乐的，即所有不具有 $A \supset B^\circ$ 或 $\Box A^\circ$ 形式的公式都是快乐的。还需证明 $\mathfrak{G}'$ 的所有最顶层都是几乎快乐的。带标签公式的几乎幸福性

命题公式的 $\mathfrak{G}'$ 是局部的，因此不受新的 R-和 $\leq$-链接的影响。- 内部层中具有幸福性。如果 $F^\circ = A \supset B^\circ$，则必有 $\mathfrak{G}_i{:}z{:}A^\bullet, \mathfrak{G}_i{:}z{:}B^\circ$，并且 $x' \leq_{\mathfrak{G}_i} z$。然后，根据 $\mathfrak{G}$的($\leq$tr)-结构饱和性，$x \leq_{\mathfrak{G}} z$，这使得 $\mathfrak{G}{:}x{:}F^\circ$ 具有幸福性。最后，如果 $F^\circ = \Box A^\circ$，则必有 $\mathfrak{G}_i{:}z{:}A^\circ$, $x' \leq_{\mathfrak{G}_i} u$，以及 $uR_{\mathfrak{G}_i}z$。然后，根据 $\mathfrak{G}$ 的($\leq$tr)-结构饱和性，$x \leq_{\mathfrak{G}} u$，并且根据构造有 $uR_{\mathfrak{G}}z$，这使得 $\mathfrak{G}{:}x{:}F^\circ$ 具有幸福性。这就完成了 $\mathfrak{G}$ 是一个具有幸福性的序列的证明。 □

和 $\Diamond^\circ$-公式的幸福性证明与引理 7.6 的情形 a）的证明相同。

为证明 b)，注意到，与 $\rightsquigarrow\rightsquigarrow\Diamond$ （定义 7.5）中的情形不同，$\rightsquigarrow\rightsquigarrow_{\circlearrowleft}$ 从不在其所操作的序列中引入新标签。每当将 $\rightsquigarrow_{\circlearrowleft}$ 应用于序列 $\mathfrak{G}$时，$\mathfrak{G}$某个最顶层的标签数会减少一个，因为 $s \neq t$。因此，对于 L 中包含 $n \geq 1$ 个标签的任意最顶层$\rightsquigarrow\rightsquigarrow_{\circlearrowleft}$，变换 $n-1$ 至多可以对其中出现的标签应用 L次。由于饱和集 $\mathfrak{G}$ 是有限的，重写$\rightsquigarrow\rightsquigarrow_{\circlearrowleft}$是终止的。 □

E. 第 8 节中的证明

定理 8.1。 如果图 7 所示的算法在步骤 4 终止，那么公式 F 不是 IS4 的定理。

证明。如果算法在步骤 4 终止，那么对于某个非公理序列 $\mathfrak{G}_i \in \mathfrak{G}_i$，$\mathfrak{G}_i{\uparrow} = \mathfrak{G}_{i0}$。容易看出，标签 r 是 $\mathfrak{G}_0$ 中唯一的标签，算法从不会将其移除，因为它始终保留在每个集合中每个序列的根层的根簇中。事实上：半饱和不会移除标签；饱和不会从根簇中移除标签；提升饱和不会移除标签；最后，循环饱和不会从根层中移除标签。因此，$\mathfrak{G}_i{:}r{:}F^\circ$。对于 $\mathfrak{G}_i$ 的每个不满足的最顶层 L，都存在一个模拟 L 的内部层 L' （见定义 7.11），其模拟关系为 S_L。令 $\mathfrak{G}$ 为如下集合：首先向 $\mathfrak{G}_i$ 添加所有满足 $x'\mathsf{S}_L x$ 的关系原子 $x \leq x'$，其中 L 是某个不满足的最顶层，然后对结果取 $\leq$的传递闭包。只需证明 $\mathfrak{G}$ 是满足的。根据定理 5.8，$\mathsf{M}_{\mathfrak{G}}, r \not\Vdash F$。因此，根据定理 2.5，$F$ 不是 IS4 中的定理。

我们需要证明 $\mathfrak{G}$ 是结构饱和的，并且其中的所有公式都具有幸福性。由于没有添加新标签，($\leq$rf)-和(Rrf)-结构饱和性得以保持。由于没有添加新的 R-链接，(Rtr)-结构饱和性得以保持。($\leq$tr)-结构饱和性得到了明确强制。为了证明(mon$^\bullet$)-、(F_1)-和(F_2)-结构饱和性得以保持，只需对传递闭包之前添加的每个 $\leq$-链接证明这些性质即可。对于(F_1)-和(F_2)-结构饱和性，它们恰好是定义 7.11 中的模拟条件 S1 和 S2。对于(mon$^\bullet$)-模拟，这由以下事实推出：对于所有 L，$\mathsf{S}_L \subseteq \sim$，这意味着每当 $\mathfrak{G}{,}x{:}A^\bullet$ 且 $x \leq_{\mathfrak{G}} x'$ 时，由于对于某个 L有 $\mathsf{S}_L x$，我们有 $x \sim x'$，因此 $\mathfrak{G}{,}x'{:}A^\bullet$。这就完成了 $\mathfrak{G}$ 是结构饱和的证明。由于没有移除任何公式或链接，$\mathfrak{G}_i$ 中所有具有幸福性的公式在 $\mathfrak{G}$中仍然具有幸福性（这里我们再次依赖定义 5.3 中针对$\mathfrak{G}{,}x{:}A \supset B^\bullet$ 的幸福性的局部表述）。饱和序列 $\mathfrak{G}_i$ 的内部层中的所有公式都具有幸福性，最顶层中除可能有一些形如 $A \supset B^\circ$ 或$\Box A^\circ$ 的 F° 之外，所有公式也都具有幸福性。对于来自 $\mathfrak{G}_i$的某个最顶层 L 中的任何此类不满足的 $\mathfrak{G}_i{,}x{:}F^\circ$，存在一个内部层 L'，并且根据命题 7.12，存在一个标签 $x' \in L'$，使得 $x'\mathsf{S}_L x$，因此$x \leq_{\mathfrak{G}} x'$。同时，$x \sim x'$，所以 $\mathfrak{G}_i{,}x'{:}F^\circ$ 在其

F. 第9节中的证明

命题 9.4. labIS4$^\star_\leq$ 中的每条规则 r 都可在 labIS4$'_\leq$中导出；若 $\mathfrak{G}$ 是 r 的某个实例的结论，则每个前提都具有 $\mathfrak{G} + \mathfrak{H}$ 的形式，其中 $\mathfrak{H}$。

证明. 每当 labIS4$_\leq$ 中与 r 对应的规则删除主公式时，我们先使用收缩规则创建一个副本。接着，对于 labIS4$^\star_\leq$ 中将公式添加到前件的规则，我们用 labIS4$_\leq$ 中相应的规则模拟，再使用 mon$^\bullet$ 将前件公式向上传播。

自反关系原子 $x \leq x$ 在 $\supset^\bullet_{\mathsf{k}}$和 $\Box^\bullet$ 中均已出现于结论中，这是因为根据假设，结论是一个适当序列。

labIS4$^\star_\leq$ 中的复合规则 $\Diamond^\bullet_\star$ 可通过以下方式模拟：使用 $\Diamond^\bullet$创建标签 $\hat{}x$；接着依次应用一系列 F_2 规则，从 U 创建其他标签 $\hat{}$（这些规则必须按照与层上的序相容的顺序执行，即不得跳过中间层）；然后对所有这些新建标签应用 Rrf 和 $\leq$rf；接着应用 $\leq$tr 的实例，添加所有必需的 $\leq$-关系原子（依据结论序列的 $\leq$tr-结构饱和）；再应用 Rtr 的实例，添加所有必需的 R-关系原子；最后应用可容许的 mon$^\bullet$，将A 向上传播。

复合规则 $\supset^\circ_\star$ of labIS4$^\star_\leq$ 通过使用 $\supset^\circ$创建标签 $\hat{}x$来模拟，随后使用一系列 F_1 和 F_2 规则 $\hat{}$从 L 创建其他标签（务必按照与 x 在层 L中的距离递增顺序执行这些规则，即不得跳过中间标签），然后对所有这些创建的标签使用 Rrf 和 $\leq$rf，接着使用 Rtr 的实例添加所有必需的 R-关系原子（依赖于层 L中的 Rtr-结构饱和），再使用 $\leq$tr 的实例添加所有必需的 $\leq$-关系原子，最后使用可容许的 mon$^\bullet$ 实例，将 L 中的所有后件公式提升到 L中。

复合规则 $\Box^\circ_\star$ 的 labIS4$^\star_\leq$ 以相同方式进行模拟，只是使用 $\Box^\circ$ 来创建$\hat{}x$ 和$\hat{}z$ 来代替 $\supset^\circ$。注意，在此处，只要结论序列中存在 Rtr 的实例就会额外添加$\hat{}uR\hat{}z$，因为uRx 出现在结论序列中时$\hat{}uR\hat{}x$ 会按照 $\supset^\circ_{\mathsf{k}}$的情形创建，而$\hat{}xR\hat{}z$ 则由规则 $\Box^\circ$创建。 □

命题 9.8。设 D 是一个整洁推导。如果 D 的结论是适当的，那么 D 中的每个序列也都是适当的。

证明。我们还证明，对于 D 中一元规则或二元规则 r $\notin \{\supset^\circ_\star, \Box^\circ_\star\}$ 的每个实例，主公式的标签位于（结论和前提中的）最顶层。

命题和该陈述通过对给定的整洁推导进行互归纳得到证明。只需

分别考察 labIS4$^{\star}_{\leq}$ 的每条规则 r。设 $\mathfrak{G}$ 是 D 中 r 的一个实例的结论，$\mathfrak{H}$ 是其前提之一。

r $\in \{\wedge^{\bullet}_{k}, \wedge^{\circ}_{k}, \vee^{\bullet}_{k}, \vee^{\circ}_{k}, \supset^{\bullet}_{k}, \Box^{\bullet}, \Diamond^{\circ}, 4^{\bullet}, 4^{\circ}\}$ 这些规则既不会创建新标签，也不会创建新的关系原子，并且根据（4）有 f($\mathfrak{H}$) = f($\mathfrak{G}$)。因此，$\mathfrak{H}$ 的大多数适当性方面都从 $\mathfrak{G}$中平凡地继承而来，包括垂直性、没有非单元素簇、树簇化性和树分层性，以及结构完备性的大多数方面；例外是 mon$^{\bullet}$，因为向内部层添加公式可能会破坏这一性质。然而，D 的整洁性意味着所有添加的公式都被添加到了允许的标签上，而在受限序列中，这些标签只能位于最顶层。换言之，每当规则 r 添加一个带标签公式 $x{:}A^{\bullet}$ 时，标签 x 都没有非平凡的未来，即不存在满足 $x{\leq_{\mathfrak{H}}}y$ 的标签 $y \neq x$。因此，序列 $\mathfrak{H}$ 同样是 mon$^{\bullet}$-饱和的，因而也是适当的。最后，由于对于所有这些规则，主公式的标签都与添加公式所对应的标签位于同一层（对于 $\wedge^{\bullet}_{k}$、$\wedge^{\circ}_{k}$、$\vee^{\bullet}_{k}$、$\vee^{\circ}_{k}$ 和 $\supset^{\bullet}_{k}$，是同一标签；对于 $\Box^{\bullet}$、$\Diamond^{\circ}$、$4^{\bullet}$ 和 4°，是父标签），所以主公式也位于最顶层。

r $= \Diamond^{\bullet}_{\star}$ 令 $x{:}\Diamond A^{\bullet}$ 为该规则的主公式。尽管该规则创建了一个新标签 y，形成一个单元素簇，但它被添加到现有的层 L_x 中，并且在 $\mathfrak{H}$ 中没有过去（也没有非平凡未来）。因此，树分层性、垂直性和单例簇条件都得以保持。注意，根据（4），新标签 y 必须被允许出现在 $\mathfrak{H}$中；根据受限性的定义，这意味着 L_x 是最顶层的（在 $\mathfrak{G}$ 和 $\mathfrak{H}$中都是如此）。该规则添加的关系原子显然确保了新创建标签的树簇性、Rtr-、Rrf-以及 $\leq$rf-结构完备性，而 $\leq$tr-结构完备性则是显然的。由于没有向 $\mathfrak{G}$中已有的标签添加公式，mon$^{\bullet}$-结构完备性得以满足；而唯一的新 $\leq$-关系原子是$y{\leq_{\mathfrak{H}}}y$，对于该原子而言，mon$^{\bullet}$-结构完备性是显然的。对于这个新原子$y{\leq_{\mathfrak{H}}}y$，F$_1$-和 F$_2$-结构完备性同样是显然的。至于新的 R-原子，它们全都具有 $zR_{\mathfrak{H}}y$ 的形式，其中 z 和 y 都位于 $\mathfrak{H}$的最顶层 L_x 中。由于二者都没有非平凡未来，F$_1$-和 F$_2$-结构完备性得以满足。

r $\in \{\supset^{\circ}_{\star}, \Box^{\circ}_{\star}\}$ 令 $x{:}A^{\circ}$ 为规则的主公式，并令 x' 为该规则创建的新标签之一，使得 $x{\leq_{\mathfrak{H}}}x'$。该规则创建的所有标签都与 x' 处于同一层 $L_{x'}$，因此 $L_x \leq_{\mathfrak{H}} L_{x'}$，并且这一层成为新的最顶层。所有新创建的标签（在 $\Box^{\circ}_{\star}$的情形下，除一个标签外）都有过去。显然，新创建标签的树分层性、垂直性、F$_1$-、F$_2$-、mon$^{\bullet}$-、$\leq$rf-和$\leq$tr-结构完备性都得到了保证。新层的树聚类性，以及该新层 $L_{x'}$ 的 Rtr-和 Rrf-结构完备性，继承自 L_x 的相应性质（除外加的标签 y 外；该标签由 $\Box^{\circ}_{\star}$ 以一种确保所有这些性质的方式创建）。注意，由于该规则只触及新创建的标签，而这些标签是在新的最顶层中创建的，因此不必要求 x 处于最顶层。 □

为了证明引理 A.9，我们还需要另一个定义和一个辅助引理。

定义 A.7。设推导 D$_n$ 是稳定集 $\mathfrak{S}$的 n-展开，其前提 $\mathfrak{G}_1$、...、$\mathfrak{G}_k$是特定序列 $\mathfrak{G} \in \mathfrak{S}$的 n-展开。设 U$_i \subseteq \ell(\mathfrak{G}) \times \ell(\mathfrak{G}_i)$ 是展开关系 n，用以证明对于每个 $i = 1..k$，$\mathfrak{G}_i$ 是 $\mathfrak{G}$的 n-展开。若对于每个 $i = 1..k$，满足 $\hat{}x \in \ell(\mathfrak{G}_i)$ 的每个标签 $\hat{}x$ U$_i\hat{}x$ 在 $\mathfrak{G}_i$ 中都是允许的，则称 D$_n$ 对于标签 $x \in \hat{}\ell(\mathfrak{G})$ 是允许的。

引理 A.8。假设 $\mathfrak{S} \rightsquigarrow_{\mathfrak{H}} \mathfrak{S}'$，其中 $\mathfrak{G} \in \mathfrak{S}$ 是重写步骤中被替换的序列，并且 $x{:}\ C$ 是不满足公式，该公式根据定义 7.1 进行展开。如果$\mathfrak{S}$ 是 F-可展开的，使得对于每个 $N \in \mathbb{N}$ 都存在一个N-展开 $\mathcal{D}_N$，其为 $\mathfrak{S}$ 的展开，并且对于 x是允许的，那么 $\mathfrak{S}'$ 也是F-可展开的。

证明。我们需要证明 $\mathfrak{S}'$ 是 F-可展开的，即对于每个 n，都存在一个整洁推导 D$'_n$，其结论为 $\mathfrak{G}_0(F)$，并且 D$'_n$ 的每个前提都是 $\mathfrak{S}'$ 中某个序列的 n-展开。

只需考虑 $\mathfrak{S}'$ 通过定义 7.1 中列出的步骤从 $\mathfrak{S}$ 获得的每一种方式。对于定义 7.1 的每种情形，我们选取某个 $N \geq n$，并将 D$_N$ 扩展为 D$'_n$。$\mathfrak{S}$ 中唯一未出现在 $\mathfrak{S}'$ 中的序列是 $\mathfrak{G}$；根据所采用的步骤，它在 $\mathfrak{S}'$ 中要么被序列 $\mathfrak{G}'$ 所替代，要么被两个序列 $\mathfrak{G}'$ 和 $\mathfrak{G}''$ 所替代。因此，D$_N$ 中唯一需要扩展的前提是 $\mathfrak{G}$的 N-展开。所有其他前提都是 N-展开，因此也是 $\mathfrak{S}$ 中某个仍保留在 $\mathfrak{S}'$中的（稳定）序列的 n-展开。

由于 $\mathfrak{G}$属于 D$_N$，且是 N-展开，因此对于每个前提 $\mathfrak{G}$ 对 D$_N$ 是允许的，因此对

2）当 $\mathfrak{G}$、$x\colon A\vee B^{\circ}$ 以及 $x\colon A^{\circ}$ 以及 $x\colon B^{\circ}$ 被添加时，情形类似，其中用 $\vee_{\mathsf{k}}^{\circ}$ 代替 $\wedge_{\mathsf{k}}^{\bullet}$。

3) 假设 $\mathfrak{G}$，$x\colon \Box A^{\bullet}$。我们进一步将此步骤拆分为若干子步骤，每个子步骤只为一个标签 y（满足 $xR_{\mathfrak{G}}y$）补上缺失的 $y\colon A^{\bullet}$ 和 $y\colon \Box A^{\bullet}$。注意，由于 $uR_{\hat{\mathfrak{G}}}u$ 适当，$\hat{}$ $\hat{}$每个标签 $u\in\ell(\mathfrak{G})$ 都满足 $\mathfrak{G}$。规则序列取决于 x 相对于 y: 的位置。

a) 如果 $x=y$，则使用 $N=n$ 并令 $\mathsf{U}':=\mathsf{U}$。再次需要修正的唯一问题是 (U1)。每个 $\hat{}x$（满足 $x\mathsf{U}\hat{}x$）都是允许的，并且根据 $\mathfrak{G}$ 的 (U1)，我们有 $\mathfrak{G},\hat{}x\colon\Box A^{\bullet}$。为获得 $\mathfrak{G}'$，我们通过以下规则扩展 D_n 的前提 $\mathfrak{G}$：

$$\Box^{\bullet}\ \frac{\mathcal{R},\hat{x}R\hat{x},\Gamma,\hat{x}\colon\Box A,\hat{x}\colon A\Longrightarrow\Delta}{\mathcal{R},\hat{x}R\hat{x},\Gamma,\hat{x}\colon\Box A\Longrightarrow\Delta}$$

对每个满足 $x\mathsf{U}\hat{}x$ 的 $\hat{}x$。因此，只要 $x\mathsf{U}\hat{}x$，就有 $\hat{}x\sim_{\mathfrak{G}'}x$。b) 如果 x 和 y 不在同一个簇中，则再次使用 $N=n$ 并令 $\mathsf{U}':=\mathsf{U}$。根据 (U5)–(U6)，存在 $\hat{}x$，使得 $x\mathsf{U}\hat{}x$。由于 D_n 对 x 是允许的，标签 $\hat{}x$ 在 $\mathfrak{G}$ 中是允许的。根据 (U2)，对于每个满足 $\hat{}y\mathsf{U}\hat{}y$ 的 $\hat{}y$，都有 $\hat{}xR_{\hat{\mathfrak{G}}}\hat{}y$。根据定义 9.6(a)，所有这些 $\hat{}y$ 在 $\mathfrak{G}$ 中也都是允许的。根据 $\mathfrak{G}$ 的 (U1)，我们有 $\mathfrak{G},\hat{}x\colon\Box A^{\bullet}$。再次，(U1) 是唯一需要修正的问题。为获得 $\mathfrak{G}'$，我们通过以下规则扩展 D_n 的前提 $\mathfrak{G}$：

$$\Box^{\bullet},4^{\bullet}\ \frac{\mathcal{R},\hat{x}R\hat{y},\Gamma,\hat{x}\colon\Box A,\hat{y}\colon\Box A,\hat{y}\colon A\Longrightarrow\Delta}{\mathcal{R},\hat{x}R\hat{y},\Gamma,\hat{x}\colon\Box A\Longrightarrow\Delta}$$

对于每个标签 $\hat{}y$ 使得 $y\mathsf{U}\hat{}y$。这确保了 $\hat{}y\sim_{\mathfrak{G}'}y$ 每当 $y\mathsf{U}\hat{}y$。

c) 如果 $x\neq y$ 位于同一个簇 C 中，则它不是单元素簇。令 $|C|=k\geq 2$。使用 $N=n+1$。$\hat{}$ 根据 $\mathfrak{G}$ 的 (U6)，序列 $\mathfrak{G}$ 包含 $n+1$ 从簇 C 中所有标签的展开副本 C_1、……、C_{n+1}（这些展开本身不是簇，因为 $\mathfrak{G}$ 是一个适当序列）。特别地，存在 $\hat{}x_1\in C_1\subseteq\ell(\mathfrak{G})$ 使得 $\hat{}\hat{}$ $x\mathsf{U}\hat{}x_1$，并且对于 $j=2..n+1$，存在 $\hat{}y_j\in C_j\subseteq\ell(\mathfrak{G})$ 使得 $y\mathsf{U}\hat{}y_j$，其中 $j=2..n+1$。由于 $\mathsf{D}_{n+1}\hat{}$ 对 x 是允许的，标签 $\hat{}x_1$ 在 $\mathfrak{G}$ 中是允许的。根据 $\mathfrak{G}$ 的 (U1)，我们有 $\mathfrak{G},\hat{}x_1\colon\Box A^{\bullet}$。由于适当序列 $\mathfrak{G}$ 必须是 Rtr-饱和的，根据 $\mathfrak{G}$ 的 (U6)，可得 $\hat{}x_1R_{\hat{\mathfrak{G}}}\hat{}y_j$for $j=2..n+1$。因此，根据定义 9.6(a)，对于 $j=2..n+1$ 的所有 $\hat{}y_j$ 也都是允许的。在此，我们仅对展开 y 的 $\hat{}n$ 个标签 $\hat{}y_2$、……、$\hat{}y_{n+1}$ 恢复 (U1)。为了得到 $\mathfrak{G}'$，我们通过规则扩展 D_{n+1} 的前提 $\mathfrak{G}$

$$\Box^{\bullet},4^{\bullet}\ \frac{\mathcal{R},\hat{x}_1R\hat{y}_j,\Gamma,\hat{x}_1\colon\Box A,\hat{y}_j\colon\Box A,\hat{y}_j\colon A\Longrightarrow\Delta}{\mathcal{R},\hat{x}_1R\hat{y}_j,\Gamma,\hat{x}_1\colon\Box A\Longrightarrow\Delta}$$

对于每个 $j=2..n+1$。由于$\hat{}x_1R_{\hat{\mathfrak{G}}}\hat{}y_1$通常不一定成立，对于标签$\hat{}y_1$以及 y的所有其他展开，无法保证（U1）成立，因此必须将它们从展开关系中移除，即：

$$\mathsf{U}':=\mathsf{U}\setminus\Big\{(y,\hat{u})\mid\hat{u}\notin\{\hat{y}_2,\dots,\hat{y}_{n+1}\}\Big\}.$$

对于 $\mathfrak{G}'$ 的每个单元素簇，$'$ 的 (U5) 可由 $\mathfrak{G}$ 中同一簇对应的 (U5) 推出。除 C 外，$\mathfrak{G}'$ 的所有非单元素簇都具有 $n+1$ 由 U6) 对 $(n+1)$-展开生成的展开副本，且任意 n 其中的副本都满足 U6)，这是n-展开所需的 $\mathfrak{G}'$。最后，对于 C 中的簇 $\mathfrak{G}'$，标签$\hat{}$ $\hat{}$集合 C_2，...，C_{n+1}可作为所需的展开副本，特别是$\hat{}_{\mathfrak{G}'}\hat{}y_j\sim_{\mathfrak{G}'}y$对应于 $j=2..n+1$。

4）当 $\mathfrak{G}$、$x\colon\Diamond A^{\circ}$ 以及 $y\colon A^{\circ}$ 以及 $y\colon\Diamond A^{\circ}$ 被添加到某个标签y 时，使得 $xR_{\mathfrak{G}}y$ 的情形是类似的，其中使用 $\Diamond^{\circ}$ 和 4° 来分别代替 $\Box^{\bullet}$ 和 $4^{\bullet}$。

5）假设 $\mathfrak{G}$、$x\colon A\vee B^{\bullet}$ 以及 $x\colon A^{\bullet}$ 被添加以获得 $\mathfrak{G}'$和 $x\colon B^{\bullet}$ 被添加以获得 $\mathfrak{G}''$。规则序列取决于簇的大小，而x 属于 $\mathfrak{G}$:

a）如果 $\{x\}$ 是单例簇，则使用 $N=n$ 并设置 $\mathsf{U}':=\mathsf{U}$。根据 $\mathfrak{G}$的（U5），存在唯一的 $\hat{}x$，使得 $\hat{}$ $x\mathsf{U}x$，且 $\hat{}x$ 在 $\mathfrak{G}$中被允许。根据 $\mathfrak{G}$的（U1），我们 $\hat{}$有 $\mathfrak{G}$、$\hat{}$ $x\colon A\vee B^{\bullet}$，对于这个 $\hat{}x$也是如此。这里仍然只有（U1）需要修正；对于此规则，可以通过两种方式完成：要么添加 $A^{\bullet}$ 以得到 n-展开的 $\mathfrak{G}'$，要么添加 $B^{\bullet}$ 以得到 n-展开的 $\mathfrak{G}''$。$\hat{}$我们将 D_n 的前提 $\mathfrak{G}$ 扩展一条规则

$$\vee_{\mathsf{k}}^{\bullet}\ \frac{\mathcal{R},\Gamma,\hat{x}\colon A\vee B,\hat{x}\colon A\Longrightarrow\Delta\quad\mathcal{R},\Gamma,\hat{x}\colon A\vee B,\hat{x}\colon B\Longrightarrow\Delta}{\mathcal{R},\Gamma,\hat{x}\colon A\vee B\Longrightarrow\Delta}$$

对于唯一标签 $\hat{}x$。由此得到的左前提 $\hat{}_{\mathfrak{G}'}$和右前提 $\hat{}_{\mathfrak{G}''}$ 很容易看出分别是n-展开 $\mathfrak{G}'$ 和 $\mathfrak{G}''$ 的折叠，特别是，

$$\hat{}_{\mathfrak{G}'}\hat{x}\sim_{\mathfrak{G}'}x\qquad\text{and}\qquad\hat{}_{\mathfrak{G}''}\hat{x}\sim_{\mathfrak{G}''}x.$$

b) 如果 $x\in C$ 对于某个簇 C 满足 $|C|=k\geq 2$，则使用 $\hat{}$ $N=2n-1$。根据 $\mathfrak{G}$的 (U6)，序列式 $\mathfrak{G}$ 包含来自$\hat{}$ $\hat{}$ 簇 C的所有标签的 $2n-1\hat{}$ $\hat{}$ 个展开副本 C_1，……，C_{2n-1}。特别地，存在 $\hat{}x_j\in C_j\subseteq\ell(\mathfrak{G})$，使得对于$j=1..2n-1$，有 $x\mathsf{U}\hat{}x_j$，其中 $j=1..2n-1$。由于 D_{2n-1} 对于 x是允许的，因此 x_j 的所有标签 $\hat{}$ 对于 $j=1..2n-1$ $\hat{}$ $\hat{}$ 都在 $\mathfrak{G}$ 中是允许的。根据 $\mathfrak{G}$的 (U1)，我们有 $\mathfrak{G}$，$\hat{}x_j\colon A\vee B^{\bullet}$，其中 $j=1..2n-1$。我们无法为所有 $2n-1$ 标签 $\hat{}x_j$修正 (U1)，而只能为其中（至少）n 个修正。我们将 $\mathfrak{T}_0:=\{\hat{}\mathfrak{G}\}$ 设为当前推导的 $2^0=1$ 个前提的集合。注意，$\mathfrak{G}$ 是形如以下序列式的特殊情形（其中 $l=0$）

$$\mathcal{R},\Gamma,\hat{x}_1\colon D_1,\dots,\hat{x}_l\colon D_l\Longrightarrow\Delta\qquad(8)$$

其中 $D_i\in\{A,B\}$ 对每个 $i=1..l$ 以及$\hat{}x_j\colon A\vee B\in\Gamma$ 对所有 $j=1..2n-1$。从 $k=0$ 开始并重复此过程，直到 $k=2n-2$，我们将每个形如（8）的 2^k 推导前提扩展为 $l=k$ 中的 $\mathfrak{T}_k$ 如下

$$\vee_{\mathsf{k}}^{\bullet}\ \frac{\mathcal{R},\Gamma,,\hat{x}_1\colon D_1,\dots,\hat{x}_k\colon D_k,\hat{x}_{k+1}\colon A\Longrightarrow\Delta\quad\mathcal{R},\Gamma,,\hat{x}_1\colon D_1,\dots,\hat{x}_k\colon D_k,\hat{x}_{k+1}\colon B\Longrightarrow\Delta}{\mathcal{R},\Gamma,\hat{x}_1\colon D_1,\dots,\hat{x}_k\colon D_k\Longrightarrow\Delta}$$

由此，得到形式为 (8) 的 $l=k+1$ 的 2^{k+1} 推导前提，它们构成集合 $\mathfrak{T}_{k+1}$。最后，一个前提 $\hat{}\mathfrak{G}$ 产生了集合 $\mathfrak{T}_{2n-1}$，其中包含形式为 (8) 的 2^{2n-1} 关于 $l=2n-1$ 的前提。接下来需要说明，如何针对来自 $\mathfrak{T}_{2n-1}$的每个前提限制关系 U，从而得到一个 n 展开

或 $\mathfrak{G}'$，或 $\mathfrak{G}''$。考虑任意一个 $\mathfrak{H} \in \mathfrak{T}_{2n-1}$。它的形式为

$$\mathcal{R}, \Gamma, \hat{x}_1{:}D_1, \ldots, \hat{x}_{2n-1}{:}D_{2n-1} \Longrightarrow \Delta$$

由于所有 $2n-1$ 公式 D_j 都是仅两个不同公式 A 和 B的重复，根据鸽巢原理，这两个公式中的一个必定至少重复n 次。换言之，存在 $E \in \{A, B\}$，使得存在 n 个指标 $1 \leq i_1 < \cdots < i_n \leq 2n-1$，对于所有 $j = 1..n$都有 $D_{i_j} = E$。在此情形下，对于此前提 $\mathfrak{H}$，我们设

$$\mathsf{U}_{\mathfrak{H}} := \mathsf{U} \setminus \Big\{(x, \hat{u}) \mid \hat{u} \notin \{\hat{x}_{i_1}, \ldots, \hat{x}_{i_n}\}\Big\}.$$

令 $\mathfrak{G}^{\mathfrak{H}}$ 表示 $\mathfrak{G}'$，当 $E = A$ 时；当 $E = B$时表示 $\mathfrak{G}''$。对于 $\mathfrak{G}^{\mathfrak{H}}$的每个单元素簇，$\mathsf{U}_{\mathfrak{H}}$ 的 (U5) 可由 U 的 $\mathfrak{G}$ 中同一簇的 (U5) 得出。除 C 外，$\mathfrak{G}^{\mathfrak{H}}$ 的所有非单元素簇，根据 $\mathfrak{G}$的 $(2n-1)$-展开中的 (U6)，都有 $2n-1$个展开副本，其中任意 n 个都满足 $\mathfrak{G}^{\mathfrak{H}}$的$n$-展开所需的 (U6)。最后，对于 $\mathfrak{G}^{\mathfrak{H}}$的簇 C，标签为 C_{i_1}、……、C_{i_n}的集合充当 $\mathfrak{G}^{\mathfrak{H}}$中 C所需的 n个展开副本，特别是，$\mathfrak{H}\hat{}^{x_{i_j}} \sim \mathfrak{G}^{\mathfrak{H}}{}^{x}$ 对于$j = 1..n$成立。因此，$\hat{}\,\mathfrak{G}$ 产生的所有 2^{2n-1} 前提，都是 $\mathfrak{G}'$ 或 $\mathfrak{G}''$的n-展开。

6）添加 $\mathfrak{G}$、$x{:}A \wedge B^{\circ}$ 和 $x{:}A^{\circ}$ 以得到 $\mathfrak{G}'$，以及添加 $x{:}B^{\circ}$ 以得到 $\mathfrak{G}''$ 的情形是类似的，其中使用 $\wedge^{\circ}_{\mathsf{k}}$ 代替 $\vee^{\bullet}_{\mathsf{k}}$。7）添加 $\mathfrak{G}$、$x{:}A \supset B^{\bullet}$ 和 $x{:}A^{\circ}$ 以得到 $\mathfrak{G}'$，以及添加 $x{:}B^{\bullet}$ 以得到 $\mathfrak{G}''$ 的情形也类似，其中使用 $\supset^{\bullet}_{\mathsf{k}}$ 代替 $\vee^{\bullet}_{\mathsf{k}}$。

对于每个步骤，我们都将给定的整洁推导$\mathcal{D}_N$ 扩展为推导 $\mathcal{D}'_n$，使得没有标签、禁用标签或关系原子被修改。由于所有添加的规则只涉及允许标签，因此所得的推导是整洁的。因此，推导 $\mathcal{D}'_n$ 关于 $\mathfrak{G}_0(F)$ 是一个n-展开，展开对象为 $\boldsymbol{\mathfrak{G}}'$。此外，只要 $\mathcal{D}_N$ 对于某个标签 z是允许的，$\mathcal{D}'_n$ 也同样允许，因为在前提中没有标签改变其禁用/允许状态。

□

引理 A.9. (i) 若 $\boldsymbol{\mathfrak{G}}'_0 \approx\!\!\Rightarrow^*_{\mathsf{s}} \mathfrak{T}$，则 $\mathfrak{T}$ 是 F-可展开的。(ii) 对每个 i，若 $\boldsymbol{\mathfrak{G}}_i$ 是 F- 可展开的，且对于某个半饱和的 $\mathfrak{T}$，有 $\boldsymbol{\mathfrak{G}}'_{i+1} \approx\!\!\Rightarrow^*_{\mathsf{s}} \mathfrak{T}$，则 $\mathfrak{T}$ 是 F-可展开的。

证明。(i) 若 $\boldsymbol{\mathfrak{G}}'_0 \approx\!\!\Rightarrow^*_{\mathsf{s}} \mathfrak{T}$，则 $\boldsymbol{\mathfrak{G}}'_0 = \mathfrak{T}_0 \approx\!\!\Rightarrow_{\mathsf{s}} \mathfrak{T}_1 \approx\!\!\Rightarrow_{\mathsf{s}} \cdots \approx\!\!\Rightarrow_{\mathsf{s}} \mathfrak{T}_k = \mathfrak{T}$ 对于稳定集 $\mathfrak{T}_0$、……、$\mathfrak{T}_k$ 对于某个 $k \geq 0$。注意，这些集合中出现的所有序列式都是适当的，且具有 $\mathfrak{G}$，因为半饱和步骤既不会给唯一的序列式 $\mathfrak{G}_0(F)$ 添加标签（该序列式 $\mathfrak{G}_0(F)$ 属于 $\mathfrak{T}_0$，且只有一个标签 r），也不会创建簇。为证明归纳的基础，即证明 $\mathfrak{T}_0 = \{\mathfrak{G}_0(F)\}$ 是 F-可展开的，只需考虑 $\mathcal{D}^0$ 对 $\mathfrak{G}_0(F)$ 受限版本的一个整洁推导，其中允许使用标签 r，且不应用任何规则（注意，每个序列式中唯一的标签 r 构成唯一的层，因此该层位于最上方，而不是内部层）。如前文所述，对于每个 $n \in \mathbb{N}$，该推导都是 $\mathfrak{T}_0$ 的 n-展开，这通过二元关系 $\mathsf{U} := \{(r, r)\}$ 实现，而该关系对于每个 $n \in \mathbb{N}$ 都是 n-展开关系。注意，D^0 显然对 r是允许的。

归纳步骤。设推导 $\mathcal{D}^i_n$ 属于 $\mathfrak{G}_0(F)$ 对于 $n \in \mathbb{N}$是 n-展开 $\mathfrak{T}_i$，且对 r允许。对于半饱和步骤 $\mathfrak{T}_i \approx\!\!\Rightarrow_{\mathsf{s}} \mathfrak{T}_{i+1}$，我们可以对 $x = r$ 应用引理 A.8，从而得到推导 $\mathcal{D}^{i+1}_n$属于 $\mathfrak{G}_0(F)$ 对于 $n \in \mathbb{N}$，它们是 n-展开 $\mathfrak{T}_{i+1}$，且对 r允许。[5]

（ii）回顾算法的步骤 5）、定义 7.14 和构造 7.10，$\boldsymbol{\mathfrak{G}}'_{i+1} = (\boldsymbol{\mathfrak{G}}_i \setminus \{\mathfrak{G}\}) \cup \{\mathfrak{G}\!\uparrow\}$对于某个完全饱和的非公理 $\mathfrak{G} \in \boldsymbol{\mathfrak{G}}_i$，其中

$$\mathfrak{G}\!\uparrow = \mathfrak{G} + \mathfrak{G}\!\uparrow^{x_1:F_1} + \cdots + \mathfrak{G}\!\uparrow^{x_h:F_h}$$

以及 $x_1{:}\ F_1^{\circ}$、……、$x_h{:}\ F_h^{\circ}$是某些集合的最顶层中的所有不满足公式，$\mathfrak{G}$ 并且每个 F_j° 要么是一个 $\supset$-公式，要么是一个 $\Box$-公式。还要注意，由于 $\boldsymbol{\mathfrak{G}}_i$ 是半饱和的，我们只需要对 $\{\mathfrak{G}\!\uparrow\} \subseteq \boldsymbol{\mathfrak{G}}'_{i+1}$ 进行半饱和处理即可得到 $\mathfrak{T}$。

将（ii）拆分成若干更细粒度的陈述似乎很自然，这些陈述说明展开可以通过更小的步骤得以保持，其中包括引理 A.8 已涵盖的许多步骤，并增加对应于构造 7.9 的新步骤；这些新步骤基于 $x_j{:}\ F_j^{\circ}$中的某一个创建一个新层。不幸的是，这些中间阶段无法表述为展开，因为序列会因稳定性丧失而不再稳定：一旦某一层中的第一个 $x_j{:}\ F_j^{\circ}$变得满足，该层就会变成内部层，因此，直到该层中其余所有 $x_j{:}\ F_j^{\circ}$ 都变得满足之前，序列才会再次稳定。

与此同时，应用 $\supset^{\circ}_{\star}$ 和 $\Box^{\circ}_{\star}$ 的所有必要实例以获得 $\boldsymbol{\mathfrak{G}}'_{i+1}$ 的 n-展开，然后如情形（i）那样继续并反复应用引理 A.8，会遇到另一个技术问题。这会违反整洁性条件（见定义 9.7）：每个 $\supset^{\circ}_{\star}$-和 $\Box^{\circ}_{\star}$-实例都会创建一个新层，并使先前 $\supset^{\circ}_{\star}$-或 $\Box^{\circ}_{\star}$-实例创建的层变为禁用，因此不再能应用展开这些层的半饱和步骤所需的规则。为此，我们必须在每个 $\supset^{\circ}_{\star}$-或 $\Box^{\circ}_{\star}$-实例之后，应用展开半饱和所需的规则。

因此，我们仍将按特定顺序逐步进行较小的步骤，并在每一步之后为所有 $n \in \mathbb{N}$ 保持条件（U1）–（U7），但不得不将这些步骤合并到（ii）中；到这一步结束时，所得结果便又可以称为一个展开。

设 L_1、.. 、L_h 是由 $\mathfrak{G}\!\uparrow^{x_1:\ F_1}$、. . .、$\mathfrak{G}\!\uparrow^{x_h:\ F_h}$ 分别引入的 $\mathfrak{G}\!\uparrow$ 的层。由于 $\mathfrak{T}$ 是半饱和的，且 s 的每个重写步骤都局部作用于某一层，因此不失一般性地，我们可以假设重写序列 $\boldsymbol{\mathfrak{G}}'_{i+1} \approx\!\!\Rightarrow^*_{\mathsf{s}} \mathfrak{T}$ 按层添加公式，即，所有添加到层 L_j

[5]由于集合 $\mathfrak{T}_i$ 中涉及的所有序列式 n都是适当的，对引理 A.8 的进一步观察表明，一个推导可以充当所有 $n \in \mathbb{N}$ 的 $\mathfrak{T}_i$ -展开。

的公式都会在添加公式到层 $L_{j'}$ 之前添加，只要 $j < j'$。

对于每个使下一层 L_j 半饱和的步骤，我们都以与情形（i）相同的方式保留所有 $n \in \mathbb{N}$ 的 (U1)–(U7)，这是利用以下事实实现的：L_j 由允许标签创建，而这些标签在展开半饱和步骤期间不会变为禁用。不过，严格来说，由于前文提到的稳定性缺失，我们不能依赖引理 A.8。相反，我们只需重复引理 A.8 的证明中的论证，并注意到其证明中用到的稳定性只有 Rtr-结构完备性，而在创建层 L_j时，这一性质确实成立。

唯一尚待涵盖的步骤是创建层 L_j。这只需对 $x_j{:}F_j^\circ$ 应用 $\supset_\star^\circ$-规则或 $\Box_\star^\circ$-规则即可。不过，为了保留 (U5) 和 (U6) 条件，我们必须先从 $\mathfrak{S}_i$ 的 $(2n+1)$-展开开始，以获得 $\mathfrak{T}$ 的 n-展开。对于每个不包含 x_j 的提升簇，都会丢弃展开中的额外 $n+1$ 个副本。但如果 x_j 位于某个簇中，那么 L_j 就包含该簇的 2 个副本（见图 5），其 n 次重复由提升的副本 $1..n$ 和 $n+1..2n+1$ 提供。在第 n 个副本中，只有 x_j 的提升属于展开关系。

容易验证，在此过程结束时，F可展开性的所有条件都会恢复。 □

下一步是证明 $\diamond$ 保持可展开性。为此，我们需要重复证明中的部分内容。为了简化重复过程，我们使用嵌入这一概念。

定义 A.10 (嵌入)。设 $\mathfrak{G}$ 和 $\mathfrak{H}$ 是受限的适当序列。一个嵌入 $\mathsf{e}\colon \mathfrak{G} \to \mathfrak{H}$ 是一个单射函数 $\mathsf{e}\colon \ell(\mathfrak{G}) \to \ell(\mathfrak{H})$，对所有 x，$y \in \ell(\mathfrak{G})$满足以下条件：

(e1) $x \sim \mathsf{e}(x)$；(e2) $xR_{\mathfrak{G}}y$ 当且仅当 $\mathsf{e}(x)R_{\mathfrak{H}}\mathsf{e}(y)$；(e3) $L_x \le_{\mathfrak{G}} L_y$ 当且仅当 $L_{\mathsf{e}(x)} \le_{\mathfrak{H}} L_{\mathsf{e}(y)}$。(e4) x 在 $\mathfrak{G}$ 中没有过去，当且仅当 $\mathsf{e}(x)$ 在 $\mathfrak{H}$ 中没有过去；(e5) $x \in \mathsf{f}(\mathfrak{G})$ 当且仅当 $\mathsf{e}(x) \in \mathsf{f}(\mathfrak{H})$；

命题 A.11。嵌入在复合下封闭，并保持展开。

证明。首先，我们证明两个嵌入的复合仍是嵌入。设 $\mathfrak{G}$、$\mathfrak{H}$和 $\mathfrak{J}$ 为受限的适当序列。设 $\mathsf{e}\colon \mathfrak{G} \to \mathfrak{H}$ 和 $\mathsf{e}'\colon \mathfrak{H} \to \mathfrak{J}$ 分别为从 $\mathfrak{G}$ 到 $\mathfrak{H}$、从 $\mathfrak{H}$ 到 $\mathfrak{J}$ 的嵌入。我们需要证明 $\mathsf{e}'\mathsf{e}$ 是从 $\mathfrak{G}$ 到 $\mathfrak{J}$ 的嵌入，其中对于任意 $x \in \ell(\mathfrak{G})$，$(e'e)(x) := \mathsf{e}'(\mathsf{e}(x))$。由于 $\mathsf{e}\colon \ell(\mathfrak{G}) \to \ell(\mathfrak{H})$ 和 $\mathsf{e}'\colon \ell(\mathfrak{H}) \to \ell(\mathfrak{J})$ 是单射函数，因此显然 $\mathsf{e}'\mathsf{e}\colon \ell(\mathfrak{G}) \to \ell(\mathfrak{J})$ 是单射函数。对于所有 $x, y \in \ell(\mathfrak{G})$，有：

(e1) ${}_{\mathfrak{G}}x \sim {}_{\mathfrak{H}}\mathsf{e}(x) \sim {}_{\mathfrak{J}}\mathsf{e}'(\mathsf{e}(x))$。（e2） $xR_{\mathfrak{G}}y$ 当且仅当 $\mathsf{e}(x)R_{\mathfrak{H}}\mathsf{e}(y)$ 当且仅当 $\mathsf{e}'(\mathsf{e}(x))R_{\mathfrak{J}}\mathsf{e}'(\mathsf{e}(y))$。（e3） $L_x \le_{\mathfrak{G}} L_y$ 当且仅当 $L_{\mathsf{e}(x)} \le_{\mathfrak{H}} L_{\mathsf{e}(y)}$ 当且仅当 $L_{\mathsf{e}'(\mathsf{e}(x))} \le_{\mathfrak{J}} L_{\mathsf{e}'(\mathsf{e}(y))}$。（e4） x 在 $\mathfrak{G}$ 中没有过去，当且仅当 $\mathsf{e}(x)$ 在 $\mathfrak{H}$ 中没有过去，当且仅当 $\mathsf{e}'(\mathsf{e}(x))$ 在 $\mathfrak{J}$ 中没有过去。

(e5) $x \in \mathsf{f}(\mathfrak{G})$ 当且仅当 $\mathsf{e}(x) \in \mathsf{f}(\mathfrak{H})$ 当且仅当 $\mathsf{e}'(\mathsf{e}(x)) \in \mathsf{f}(\mathfrak{J})$。

ˆ其次，设 $\mathsf{U} \subseteq \ell(\mathfrak{G}) \times \ell(\mathfrak{G})$ 是一个 n-展开关系，ˆ它见证了 $\mathfrak{G}$ 是 $\mathfrak{G}$ 的一个 n-展开。我们证明，二元ˆˆ关系 $\mathsf{eU} \subseteq \ell(\mathfrak{G}) \times \ell(\mathfrak{H})$ 见证了 $\mathfrak{H}$ 是 $\mathfrak{G}$ 的一个 n-展开，其中

$$x(\mathsf{eU})\hat{y} \qquad \Longleftrightarrow \qquad (\exists \hat{x})\Big(x\mathsf{U}\hat{x} \text{ and } \hat{y} = \mathsf{e}(\hat{x})\Big).$$

令x（eU）e（ˆx），因为 $x\mathsf{U}\hat{x}$ 且 x'（eU）e（ˆx'），因为 $x'\mathsf{U}\hat{x}'$。

(U1) ${}_{\mathfrak{G}}x \sim_{\mathfrak{G}} \hat{x}$ 根据（U1）对U和${}_{\mathfrak{G}}\hat{x} \sim {}_{\mathfrak{H}}\mathsf{e}(\hat{x})$ 根据（e1）对e。

（U2）令x 和 x' 来自不同的簇。则 $xR_{\mathfrak{G}}x'$ 当且仅当 $\hat{x}R_{\mathfrak{G}}\hat{x}'$根据（U2）对U，而后者当且仅当e（ˆ$x$）$R_{\mathfrak{H}}$e（ˆ$x'$）根据（e2）对e。

（U3）$L_x \le_{\mathfrak{G}} L_{x'}$ 当且仅当 $L_{\hat{x}} \le_{\hat{\mathfrak{G}}} L_{\hat{x}'}$ 根据（U3）对U，而后者当且仅当 $L_{\mathsf{e}(\hat{x})} \le_{\hat{\mathfrak{H}}} L_{\mathsf{e}(\hat{x}')}$ 对 e 应用 (e3)。

ˆ（U4） x 在 $\mathfrak{G}$ 中没有过去当且仅当 ˆx 在 $\mathfrak{G}$ 中没有过去，依据（U4）对U。ˆ且后者当且仅当e（ˆx）在 $\mathfrak{H}$ 中没有过去，依据（e4）对e。

（U5）如果 x 在 $\mathfrak{G}$中形成单元素簇，则ˆx 是唯一的标签，在ˆ$\mathfrak{G}$ 中满足$x\mathsf{U}\hat{x}$ 依据（U5）对U。因此，e（ˆx）是在ˆ $\mathfrak{H}$中满足x（eU）e（ˆx）的唯一标签。

(U6) 如果 C 是 $\mathfrak{G}$ 中带有 $|C| = k \ge 2$ 的非单元素簇，ˆ那么根据 U 的 (U6)，存在 $x_i \in \ell(\mathfrak{G})$ 和 $\hat{}x_{i,\,j} \in \ell(\mathfrak{G})$，分别对应 $i = 1..k$ 和 $j = 1..n$，使得 $C = \{x_1, \dots, x_k\}$，ˆ且 $x_i\mathsf{U}\hat{}x_{i,\,j}$ 对于 $i = 1..k$、$j = 1..n$，并且在 $\mathfrak{G}$ 中

$$\hat{x}_{1,1}R\cdots R\hat{x}_{k,1}R\hat{x}_{1,2}R\cdots R\hat{x}_{k,2}R\cdots R\hat{x}_{1,n}R\cdots R\hat{x}_{k,n}.$$

在这种情形下，根据 e 的 (e2)，对于 $i = 1..k$、$j = 1..n$，有 x_i (eU)e(ˆ$x_{i,\,j}$)，且ˆ在 $\mathfrak{H}$ 中

$$\mathsf{e}(\hat{x}_{1,1})\,R\cdots R\,\mathsf{e}(\hat{x}_{k,1})\,R\,\mathsf{e}(\hat{x}_{1,2})\,R\cdots R\,\mathsf{e}(\hat{x}_{k,2})\,R\cdots R\,\mathsf{e}(\hat{x}_{1,n})\,R\cdots R\,\mathsf{e}(\hat{x}_{k,n}).$$

ˆ因此，e(ˆ$x_{i,\,j}$) $\in \ell(\mathfrak{H})$ 对于 $i = 1..k$、$j = 1..n$ 提供了 C 在 $\mathfrak{H}$中各标签所需的ˆ展开。

(U7) 如果 $\mathsf{e}(\hat{}x) = \mathsf{e}(\hat{}x')$，那么 $\hat{}x = \hat{}x'$ 根据 e 的单射性，并且 $x = x'$ 根据 U7) 对于 U。 □

引理 A.12（嵌入引理）。假设我们有一个整洁推导 D，其末序列为受限的适当末序列 $\mathfrak{G}$，前提为 $\mathfrak{G}_1$、……、$\mathfrak{G}_n$。如果存在一个嵌入 $\mathsf{e}\colon \mathfrak{G} \to \mathfrak{H}$到受限的适当序列 $\mathfrak{H}$中，那么存在一个整洁推导 D′，其结论为 $\mathfrak{H}$，前提为 $\mathfrak{H}_1$、……、$\mathfrak{H}_n$，使得存在嵌入 $\mathsf{e}_i\colon \mathfrak{G}_i \to \mathfrak{H}_i$，对应于 $i = 1..n$。

证明。首先考虑 D 仅由一个整洁推理规则实例 r 构成的情形。设 $\mathfrak{G}_i$ 是 r 的一个前提。那么，根据命题 9.4，对于某个$\mathfrak{G}'_i$，我们有 $\mathfrak{G}_i = \mathfrak{G} + \mathfrak{G}'_i$。如果 r 不添加新标签，那么 $\ell(\mathfrak{G}_i) = \ell(\mathfrak{G})$，并且我们有 $\ell(\mathfrak{H}_i) = \ell(\mathfrak{H})$ 以及 $\mathsf{e}_i = \mathsf{e}$。这给出了一个满足该引理的正确规则应用。否则，如果 r 添加了新标签，那么 r 是 $\diamond_\star^\bullet$、$\supset_\star^\circ$ 或 $\Box_\star^\circ$ 之一，只有一个前提，且新标签只出现在 $\mathfrak{G}'_1$ 中。

- r 是 $\diamond_\star^\bullet$：该规则应用于 x：$\diamond A^\bullet$ 在 $\mathfrak{G}$ 中，创建一个新标签 y，我们也可以将同一规则应用于 $\mathsf{e}(x)$：$\diamond A^\bullet$在 $\mathfrak{H}$ 中，创建一个新标签 y'。我们可以定义 $\mathsf{e}_1(y) = y'$，对于所有其他 $u \in \ell(\mathfrak{G}_1)$，我们定义 $\mathsf{e}_1(u) = \mathsf{e}(u)$。

- r 是 $\supset^\circ_\star$ 或 $\Box^\circ_\star$：该规则应用于 $x\colon F^\circ$ 在 $\mathfrak{G}$ 中，创建一个新层 L'，其标签全部只出现在 $\mathfrak{G}'_1 = \mathfrak{G}{\uparrow^x}\colon F$ 中。我们可以将同一规则应用于 e(x)：F° 在 $\mathfrak{H}$ 中，创建一个新层 L'' 并将 $\mathfrak{H}'_1 = \mathfrak{H}{\uparrow}\mathsf{e}(x)\colon F$ 添加到 $\mathfrak{H}$ 中。一般而言，L'' 可以包含比 L'更多的标签。但 e 定义了从 L_x 到 $L\mathsf{e}(x)$ 的嵌入，并由此规范地定义了一个映射 e′：$L' \to L''$，如下所示：对于 $y' \in L'$，当且仅当以下任一条件成立时，我们定义 e′(y') $= y'' \in L''$：存在一个 $v \in L_x$，满足$v \leq_{\mathfrak{G}_1} y'$ 且 e(v)$\leq_{\mathfrak{H}_1} y''$；或者 y' 在 $\mathfrak{G}_1$ 中没有过去，且y'' 在 $\mathfrak{H}_1$ 中没有过去。然后，我们可以将嵌入 e1：$\mathfrak{G}_1 \to \mathfrak{H}_1$ 定义为：如果 $u \in L'$，则 e1(u) $=$ e′(u)；否则，e1(u) $=$e(u)。

现在，我们可以通过对 D 的高度进行直接归纳来证明任意 D 的引理。 □

引理 A.13. 对于半饱和序列 $\mathfrak{G}$，如果算法在步骤 1）中通过将 $\mathfrak{G}$ 中的标签 y 替换为标签 x，并对 R 应用传递闭包，来应用定义 7.5 的选项 1），从而得到满足 $\mathfrak{G} \rightsquigarrow_\diamond \mathfrak{G}'$ 的 $\mathfrak{G}'$，则标签 y 在 $\mathfrak{G}$ 中形成一个单元素簇。

证明。由于应用选项 1）意味着 y 在 $\mathfrak{G}$中没有过去，因此算法创建了 y 作为一个单元素簇（除非 $y = r$，在这种情况下，它从一开始就已经存在，但最初是一个单元素簇）。不难检查，每当算法创建一个新的非单元素簇时，要么 z 对于簇中的所有标签 z都有过去，要么所有带标签公式 $z\colon \diamond B^\bullet$ 对于簇中的所有标签 z 都变得满足。由于 $y\colon \diamond A^\bullet$ 必须是不满足的且没有过去，才能应用选项 1），因此标签 y 不是非平凡簇的一部分。 □

定义 A.14. 对于序列 $\mathfrak{G}$ 的标签 x，我们定义

$$\overrightarrow{x} := \{z \mid xR_{\mathfrak{G}}z\}.$$

^^^引理 A.15. 如果 $\boldsymbol{\mathfrak{G}} \rightrightarrows_\diamond \boldsymbol{\mathfrak{G}}$ 且 $\boldsymbol{\mathfrak{G}}$ 是半饱和的，并且 ^^^F- 可展开，那么 $\boldsymbol{\mathfrak{G}}$ 是 F- 可展开的。

^^^ 证明。回顾定义 7.5，根据该定义，$\boldsymbol{\mathfrak{G}}$ 是集合 $(\boldsymbol{\mathfrak{G}}\backslash\{\mathfrak{G}\})\cup\{\mathfrak{G}'\}$ 的半饱和，其中存在某个 $\mathfrak{G} \in \boldsymbol{\mathfrak{G}}$ 使得 $\mathfrak{G} \rightsquigarrow_\diamond \mathfrak{G}'$。注意，根据引理 A.5(i)，$\mathfrak{G}'$ 是一个稳定序列式。半饱和过程按照引理 A.9 中的方式处理，利用 $\mathfrak{G}$ 已经是半饱和的这一事实，因此半饱和过程将要触及的唯一标签是一个新标签，该标签按照允许的方式创建。因此，剩下需要解释的是，如何将 $\mathfrak{G}$ 的 N- 展开转换为 $\mathfrak{G}'$ 的 n- 展开，其中 $N \geq n$ 适当。根据定义 7.5，有两种选择：

1) 在选项 1 中，我们有 $\mathfrak{G}{,}y\colon\diamond A^\bullet$，其中 y 根据引理 A.13 构成一个单元素簇；并且我们有来自最顶层的某个 x，其包含 $x \sim y$，$xR_{\mathfrak{G}}y$ 和 $x \neq y$，使得所有 $u \in \overrightarrow{x}$ 都没有过去。然后，通过用 x 替换 y 并对 R-传递闭包，得到 $\mathfrak{G}'$。^^^ 注意，所得的 $\mathfrak{G}'$ 是半饱和的，因此，$\boldsymbol{\mathfrak{G}} = (\boldsymbol{\mathfrak{G}}\backslash\{\mathfrak{G}\})\cup\{\mathfrak{G}'\}$。考虑 D 的任意一个前提 $\hat{}$，它是经由 U_1 对 $\mathfrak{G}$ 进行 N-展开所得。根据 (U5)，存在唯一的 $\hat{}y$，使得 $y\mathsf{U}_1\hat{}y$。N 的选取以及 D_N 的扩展，取决于 x 在 $\mathfrak{G}$中所属簇的大小：

a) 如果 $\{x\}$ 是单元素簇，则设 $N := n$。根据 (U5)，存在唯一的 $\hat{}x$，使得 $x\mathsf{U}_1\hat{}x$。根据 (U2)，$\hat{}xR_{\hat{\mathfrak{G}}}\hat{}y$。由于按假设$x$ 没有过去，根据 (U4)，$\hat{}x$也没有过去。由$\hat{}\mathfrak{G}$ 的树分层性可知，$\overrightarrow{\hat{}x}$ 中的所有标签，包括$\hat{}y$，都没有过去。因此，$\overrightarrow{\hat{}x}$中的所有标签（$\hat{}x$ 自身除外）都是由规则$\diamond^\bullet_\star$创建的。令 $\mathfrak{K}_1$ 为 D^n 中通往$\hat{}\mathfrak{G}$ 的分支上，第一个（即最靠下的）规则 $\diamond^\bullet_\star$ 实例的前提，该实例创建了$\hat{}x$的一个非平凡子节点；将这个子节点记作 u_1。由于 D_n 是整洁的，标签 u_1 是 $\mathfrak{K}_1$中唯一允许的标签。令 $\hat{}x\colon\diamond C^\bullet$ 为此规则 $\diamond^\bullet_\star$实例的主公式。则 $\mathfrak{K}_1{,}u_1\colon C^\bullet$ 是 u_1中唯一带标签的 $\hat{}$ 公式。由于 $\mathfrak{K}_1{,}\hat{}x\colon\diamond C^\bullet$，分支上方也有 $\mathfrak{G}{,}\hat{}x\colon\diamond C^\bullet$，因此根据 (U1)，$\mathfrak{G}{,}x\colon\diamond C^\bullet$。因此，$\hat{}\mathfrak{G}{,}y\colon\diamond C^\bullet$，因为 $x \sim y$ 且根据 (U1)，$\mathfrak{G}{,}\hat{}y\colon\diamond C^\bullet$。由于$\hat{}$ $\hat{}y$ 位于 $\mathfrak{G}$的最顶层，因此可以对 $\mathfrak{G}$ 应用整洁规则 $\diamond^\bullet_\star$，其主公式为$\hat{}y\colon\diamond C^\bullet$。令此规则的前提为 $\mathfrak{K}_2$，并令 u_2 为该规则创建的新标签，它是 $\mathfrak{K}_2$中唯一允许的标签。考虑以下函数 $\mathsf{e}_1\colon \ell(\mathfrak{K}_1) \to \ell(\mathfrak{K}_2)$：

$$\mathsf{e}_1 := \{(\hat{x},\hat{y}),(u_1,u_2)\}\cup\{(w,w) \mid w \in \ell(\mathfrak{K}_1)\backslash\{\hat{x},u_1\}\}.$$

显然，e_1 是一个嵌入。根据引理 A.12，可以通过此嵌入将以 $\mathfrak{K}_1$ 为根的整洁推导子树应用于 $\mathfrak{K}_2$，从而得到一个整洁推导；根据命题 A.11，该扩展的所有前提都是 n展开，与 D_n 的相应前提来自 $\boldsymbol{\mathfrak{G}}$ 的同一组序列。特别地，对于该子树的前提 $\mathfrak{G}$，嵌入 e_1 将扩展为嵌入，映射到扩展推导的某个前提 $\mathfrak{G}_2$，其中 $\mathsf{e}_1(\hat{}y) = \hat{}y_2$，使得 $\hat{}_{\mathfrak{G}_2}\hat{}y_2 \sim_{\hat{\mathfrak{G}}}\hat{}y$（注意，$\hat{}_{\mathfrak{K}_2}\hat{}y \sim_{\hat{\mathfrak{G}}}\hat{}y$）。根据命题 A.11，我们可以构造一个 n展开 U_2，将 $\mathfrak{G}$ 展开到 $\mathfrak{G}_2$ 中，使得对于所有标签$\hat{}w \in \ell(\mathfrak{K}_1) \setminus \{\hat{}x, u_1\} = \hat{}\ell(\mathfrak{G}) \setminus \{\hat{}x\}$，包括$\hat{}x$的所有非平凡父节点，$w\mathsf{U}_1\hat{}w$ 当且仅当 $w\mathsf{U}_2\hat{}w$。此外，$x\mathsf{U}_2\hat{}y$，并且每当 $xR_{\mathfrak{G}}v$ 且 $v\mathsf{U}_2\hat{}v$ 时，我们都有$\hat{}yR_{\hat{\mathfrak{G}}_2}\hat{}v$。特别地，存在某个$\hat{}y_2$，使得 $y\mathsf{U}_2\hat{}y_2$，且$\hat{}$ $\hat{}yR_{\hat{\mathfrak{G}}_2}\hat{}y_2$和 $\mathfrak{G}_2{,}\hat{}y_2\colon\diamond C^\bullet$。

我们可以应用 $\diamond^\bullet_\star$ 来获得 $\mathfrak{K}_3$ 带有一个新标签的u_3，取一个嵌入

$$\mathsf{e}_2 := \{(\hat{y},\hat{y}_2),(u_2,u_3)\}\cup\{(w,w) \mid w \in \ell(\mathfrak{K}_2)\backslash\{\hat{y},u_2\}\}.$$

从 $\mathfrak{K}_2$ 到 $\mathfrak{K}_3$ 并重复这一过程。

经过 $n-1$ 次这样的重复后，我们将得到一个$\hat{}$前提 $\mathfrak{G}_{n+1}$ 以及一个将 $\mathfrak{G}$ 展开为$\hat{}\mathfrak{G}_{n+1}$ 的 n-展开U_{n+1}，使得对于所有标签$\hat{}\hat{}w \in \ell(\mathfrak{G}) \setminus \{\hat{}x\}$，包括$\hat{}x$的所有非平凡父节点，都有$w\mathsf{U}_1\hat{}w$ 当且仅当 $w\mathsf{U}_{n+1}\hat{}w$。此外，$x\mathsf{U}_{n+1}\hat{}y_n$，并且每当 $xR_{\mathfrak{G}}v$ 且$v\mathsf{U}_{n+1}\hat{}v$时，我们都有$\hat{}y_nR^{\hat{}\mathfrak{G}_{n+1}}\hat{}v$。特别地，对于某个 y_{n+1}，有 $y\mathsf{U}_{n+1}\hat{}y_{n+1}$，使得$\hat{}y_nR^{\hat{}\mathfrak{G}_{n+1}}\hat{}y_{n+1}$。

再一次，该推导中其他所有展开 n 次的前提，都是来自 $\boldsymbol{\mathfrak{G}}$ 的 n- 展开序列式。$\hat{}$ 现在还需要将 $\mathfrak{G}_{n+1}$ 转换为 $\mathfrak{G}'$ 的 n- 展开。为此，我们修改 $\mathfrak{G}'$ 中新创建的簇 $C_x = \{u \neq y \mid xR_{\mathfrak{G}}uR_{\mathfrak{G}}y\}$ 所对应的 U_{n+1}，并保留

$\mathfrak{G}$ 的所有其他标签的展开，如 U$_{n+1}$ 中所示。对于每个标签 $u \in C_x$ 和每个 $j = 1..n$，至少存在一个标签 $\hat{}u_j$，使得 uUj $\hat{}u_j$ 和 $\hat{}y_{j-1} R_{\hat{\mathfrak{G}}_{n+1}} \hat{}u_j R \hat{}_{\mathfrak{G}_{n+1}} \hat{}y_j$。因此，对于每个 $u \in C_x$，这些标签 $\hat{}u_1$、……、$\hat{}u_n$ 可以作为 C_x 所需的 n 展开副本。事实上，只要 $wR_{\mathfrak{G}}x$ 且对于某个 $w \neq x$ 有 wU$_{n+1}\hat{}w$，如前所述，就有 $\hat{}wR_{\hat{\mathfrak{G}}_{n+1}}\hat{}xR_{\hat{\mathfrak{G}}_{n+1}}\hat{}u_j$。另一方面，只要 $xR_{\mathfrak{G}}v$ 且对于某个 $v \notin C_x \cup \{y\}$ 有 vU$_{n+1}\hat{}v$，如前所述，就有 $\hat{}u_j R_{\hat{\mathfrak{G}}_{n+1}} \hat{}y_n R \hat{}_{\mathfrak{G}_{n+1}} \hat{}v$。注意，$x$ 的子节点若不在 $\mathfrak{G}'$ 中的 C_x 内，就必须展开为所有 $\hat{}u_j$ 的子节点所对应的标签；这正是我们需要创建 n 个成功展开而不是 $n+1$ 个的原因：前 n 个展开用于创建 C_x 的 n 个展开副本，而最后一个（第 $n+1$ 个）则为不在其簇中的 x 的子节点提供展开。

b) 若x 属于一个非单元素簇 C_x，我们采用与刚才讨论的单元素簇情形类似的扩展。因此，我们只概述其中的差异。首先，我们不从 D$_n$ 开始，而是从 D$_N$ 开始，其中N = 为 max(n,2)。$N \geq n$ 这一事实保证：对于除 $\mathfrak{G}$ 之外的所有前提，以及 $\mathfrak{G}$ 中除新创建的簇 C'_x之外的所有簇（该簇包含 C_x 以及 C_x 和 y 之间除 y以外的所有标签），都存在n-展开。

$\hat{}$ 在由 (U6) 得到的 $\mathfrak{G}$ 中 x 的 $N \geq 2$ 个展开副本中，取倒数第二个，即第（$N-1$）个展开副本作为 $\hat{}x$，并将 y 的唯一展开作为 $\hat{}y$。对这些 $\hat{}x$ 和 $\hat{}y$，按照单元素簇情形进行扩展。与该情形唯一的区别在于，如何修改最后的展开 U$_{n+1}$ 以纳入新创建的簇C'_x。对于 C_x之外的所有标签，构造保持不变。对于每个标签 $u \in C_x$，其最后一个，即第 N个来自 U$_1$ 的展开副本，位于重复出现的标签所在片段中（请注意，第（$N-1$）个展开副本不一定也满足这一点，这正是我们必须提前一个展开副本开始的原因）。因此，现在将这个第 N个副本重复 $n+1$ 次，并将其中前 n 个作为 u的新展开。

2) 在选项 2 中，我们有 $\mathfrak{G}$，y: $\Diamond A^\bullet$ 并创建一个子节点 z，它属于 y，使得 $\mathfrak{G}'$，z: $A^\bullet$。这种情形类似于引理 A.8 中半饱和所采取的步骤；如前所述，在 $\mathfrak{G}$ 被替换为 $\mathfrak{G}'$之后，也采用这些步骤进行半饱和。下面简要说明如何将 $\hat{}$展开成一个 $\mathfrak{G}$，使其成为 N对 $\mathfrak{G}$ 的展开，再将其展开成一个 n对 $\mathfrak{G}'$的展开。和之前一样，具体过程取决于 y在 $\mathfrak{G}$中所属的簇大小：

a) 如果 $\{y\}$ 是一个单例簇，则使用 $N = n$。根据 (U5)，存在唯一的 $\hat{}y$，使得 yU$\hat{}y$。根据 (U1)，对于这个 $\hat{}y$，$\mathfrak{G}$、$\hat{}y$:$\Diamond A^\bullet$。只需将 $\Diamond^\bullet_\star$ 的一个整洁实例应用于这个 $\hat{}y$:$\Diamond A^\bullet$，创建一个新标签 $\hat{}z$，该标签是前提中唯一允许使用的标签，并将集合 U$' :=$ U $\cup \{(z, \hat{}z)\}$。很容易检查出，所有展开条件都得到满足

这样得到的前提就是 $\mathfrak{G}'$ 的 n-展开。b) 如果对于某个簇 C 有 $|C| = k \geq 2$，则使用 $\hat{}N = n+1$。根据 (U6)，序列 $\mathfrak{G}$ 包含 $j = 1..n+1$ 中的 $\hat{}y_j$ 的展开副本 $n+1$，使得 $y =$ U($\hat{}y_j$) 对于 $\hat{}j = 1..n+1$ 成立。根据 (U1)，$\mathfrak{G}$、$\hat{}y_{n+1}$:$\Diamond A^\bullet$。这里我们将 $\Diamond^\bullet_\star$ 的一个整洁实例应用于这个 $\hat{}y_{n+1}$:$\Diamond A^\bullet$，创建一个新标签 $\hat{}z$，该标签是前提中唯一允许使用的标签，并按如下方式修改 U:

- $\hat{}z$ 被设为 z的唯一展开；
- 来自 $\ell(\mathfrak{G})\backslash C$ 的所有标签的展开均保持不变；
- 对于 C 的展开，我们只保留由 (U6) 保证的前 n个展开，以供 U 使用，并移除其余展开。

最后一项中移除多余展开，是为了确保 $\hat{}z$ 是簇 C所有展开的子节点。再次检查可知，所有展开条件均得到满足，因此所得前提是一个 n对 $\mathfrak{G}'$的展开。

□

$\hat{}\hat{}\hat{}$引理 A.16。如果 $\mathfrak{G} \rightsquigarrow_{\circlearrowleft} \mathfrak{G}$ 并且 $\mathfrak{G}$ 是饱和的，且 F- $\hat{}\hat{}\hat{}$ 可展开，那么 $\mathfrak{G}$ 是 F- 可展开的。

证明。假设 $\mathfrak{G} \rightsquigarrow_{\circlearrowleft} \mathfrak{G}'$ 和 $\mathfrak{G} \in \mathfrak{G}$ 是饱和的，$\mathfrak{G}$ 是F-可展开的，并且 $\mathfrak{G}' = \mathfrak{G}\backslash\{\mathfrak{G}\}\cup\{\mathfrak{G}'\}$。我们希望为 $\mathfrak{G}'$构造一个 n-展开。$\mathfrak{G}'$ 是通过闭合一个不满三角循环从 $\mathfrak{G}$ 得到的。令 L_1、L_2、L'、C_1、C_2、p_1、s、t如定义 7.16 和 7.17 中所述。我们还有一个 p_2，其中有 $p_2 \in C_2$和 $p_2 \sim p_1$，将在证明中使用。

如果 s 和 t 位于同一个簇中，那么 $\mathfrak{G}$ 的每个 n-展开也都是 $\mathfrak{G}'$的 n-展开。现在，假设 s 和 t不在同一个簇中。

如前一引理的情形 1b) 所示，我们从 N-展开 D$_N$ of $\mathfrak{G}$ 开始，其中 N = max(n, 2)，并令 $\mathfrak{G}$ 为 D$_N$ 的一个前提，该前提是 $\mathfrak{G}$ 的展开。令 U 为相应的展开关系，并令 $\hat{}$ L_1、$\hat{}L_2$、$\hat{}L'$ 分别为 L_1、L_2、L' 的展开。

现在令 T 为 D$_N$ 中以序列 $\hat{}\hat{}$ 为根的子树，其中 L' 首次出现。令 $\mathfrak{G}_0$ 为该序列，令 r 为引入 $\hat{}L'$ 的推理规则（一个 $\Box^\circ_\star$ 或 $\supset^\circ_\star$）的实例，令 $\mathfrak{H}$ 为该规则实例的结论，并令 $\hat{}p_1$ 为 $\mathfrak{H}$ 中包含该规则所作用公式的标签。于是 p_1U$\hat{}p_1$。令 $\hat{}p_2$ 为 $\mathfrak{G}$ 中满足 p_2U$\hat{}p_2$ 的一个标签。若 p_2 是单例，则 $\hat{}p_2$ 唯一确定。否则，我们将 $\hat{}p_2$ 选为簇 C_2 的第 k 次重复，其中 k 是 $\hat{}p_1$ 出现于其中的 C_1 的重复。我们可以对 $\hat{}p_2$ 应用相同的规则 r，以得到层 L''。令新序列为 $\mathfrak{G}_1$。

我们还令$\hat{}p'_1$为$\hat{}p_1$中的未来，$\hat{}$ L'，并令 p_2 为 L''中 p_2 的未来。由此我们可以定义一个嵌入 e: $\hat{}$ $\mathfrak{G}_0 \to \hat{}\mathfrak{G}_1$，它在所有层上都表现为恒等映射，除了$\hat{}L'$ $\hat{}$嵌入到 L'' 中，将每个标签映射到其唯一的未来，但 $\rightarrow\hat{}p_1$ 被嵌入到 $\rightarrow\hat{}p_2$中。现在我们将应用嵌入引理 A.12 来重复证明树 T。更确切地说，令 $\mathfrak{H}_1, \ldots, \mathfrak{H}_m$ 为 T 的叶节点，并令 $\mathfrak{H}_1, \ldots, \mathfrak{H}_l$为对于某个 $l \leq m$ 而言是 N-$\hat{}$展开的 T 的叶节点（ $\mathfrak{G}$ 也在其中）。每个 $\mathfrak{H}_i$ 都有一个,i 最顶层层 L_2，它是 N的 L_2-展开，并且该层具有

标签 $\hat{s}_i$, $\hat{}t_i$，以及 $\hat{}p_{2,i}$，它们分别对应于 s, t和 p_2 在展开中（如果 s 位于一个簇中，我们为 $\hat{s}_i$ 选择第一次重复；$\hat{}$如果 t 位于一个簇中，我们为 t_i 选择最后一次重复）。

现在，我们仿照引理 A.15 的证明，在每个叶节点 $\mathfrak{H}_1, \ldots, \mathfrak{H}_l$ 处插入子树 T 的副本。注意，每当创建新层，并使用 $\supset^\circ_\star$ 和 $\Box^\circ_\star$ 规则提升前一层时，旧三角形与新三角形之间的间隙也会随之提升（见图 13）。我们称 $\hat{}\hat{}L_{3,i}$ 为新的最顶层（它是 $L_{2,i}$ 的副本，并带有额外部分 $\hat{}\hat{}L^*_{3,i}$ ；后者是 $L_{2,i}$ 中标签的副本，但这些标签在 L_1 中没有对应项）。如果 T 中的某个规则实例向满足 $s\, R_{\mathfrak{h}_i}\, u\, R_{\mathfrak{h}_i}\, t$ 的某个标签 u 添加公式，那么我们也需要向 $L^*_{3,i}$ 中对应的标签添加相同的公式。形式上，对于每个 i，再次应用引理 A.12，即可通过第二个嵌入来实现这一点。

如同前面的引理 A.15，我们现在重复这一构造 n 次，从而获得所需的 n- $\mathfrak{G}'$展开。

□

引理 9.10 （展开引理）。 由图 7 中的算法生成的所有序列式集合 $\mathfrak{G}_i$ 均为 F-可展开的。

证明。集合 $\mathfrak{G}'_0 = \{\mathfrak{G}_0(F)\}$ 显然是 F-可展开的。根据前面的引理，该性质在用于构造序列 $\mathfrak{G}_0$、$\mathfrak{G}_1$、.....的所有步骤中均得以保持。

□

定理 9.11。如果图 7 所示的算法在步骤 2 终止，那么公式 F 是 IS4 的定理。

证明。根据前面的引理可知，导致算法终止的公理集 $\mathfrak{G}_k$ 是 F-可展开的。特别地，我们在 labIS4$^\star_\leq$ 中有一个以$\mathfrak{G}_0(F)$为结论的推导，其中所有前提都是 $\mathfrak{G}_k$中元素的 1-展开，而这些元素全都是公理性的。因此，我们可以使用 id 规则和 $\bot^\bullet$规则的实例，在 labIS4$^\star_\leq$中给出 $\mathfrak{G}_0(F)$的完备证明。根据命题 9.4，在 labIS4$'_\leq$ 中还存在 $\mathfrak{G}_0(F)$的证明。我们可以在底部用规则 Rrf 和 $\leq$rf 将其接上，从而得到 $\Longrightarrow r{:}F$ 在 labIS4$'_\leq$中的证明。根据推论 4.3 和定理 4.1，我们得到 F 是 IS4 的定理。

□

G. 第10节的证明

引理 10.1。某个 $\mathfrak{G}_i$ 的序列中出现的标签大小至多为 n。关于 $\sim$，标签共有 2^n 个等价类。

证明。 只有 F 的子公式才能出现在序列中。此外， F中每个子公式出现的位置决定了该子公式能否出现在 $\Longrightarrow$的左侧或右侧。此外，算法在序列中引入的带标签公式，都是序列中尚未出现过的（定义 7.1）。因此，在给定标签处，公式出现的次数至多为 n 。另请注意， $x \sim y$ 当且仅当 x 与 y包含 F的相同子公式集合时成立。因此， 2^n在标签上有 $\sim$ 个不同的等价类，它们可能出现在 $\mathfrak{G}_i$中。

□

引理 10.2。某个 $\mathfrak{G}_i$ 的序列中出现的簇的大小至多为 2^n。此外，关于 $\sim$，非空簇有 $2^{2^n}-1$ 个等价类。

证明。我们将证明，根据构造，簇中的所有标签关于 $\sim$ 都彼此不同。引入非单元素簇有两种方式：(i) 通过不满足 $\Diamond^\bullet$- 公式（定义 7.5 的选项 1）：若此类簇中存在等价标签，该簇就会在算法的较早步骤中创建。(ii) 通过不满足R三角形或U-三角循环（定义7.16、7.17和7.18）：若创建时从 s 到 t 的链中有重复标签，那么这些重复标签会立即合并，因为它们同样是满足相同不满足三角形条件的 s 和 t 的有效选择。因此，当 $\rightsquigarrow\rightsquigarrow\circlearrowright$ 终止时，所有新簇中的标签关于 $\sim$都彼此不同。这将簇的大小限制为 2^n，并且在 $\mathfrak{G}_i$ 中可能出现的非空簇至多有 $2^{2^n}-1$ 个不同的等价类。

□

引理 10.3。 集合中一个序列内一层中的分支长度 $\mathfrak{G}_i$ 有界，且该界由 F确定。

证明。若对于所有 $x' \in M'$ 都存在一个 $x \in M$ 使得 $x' \leq_{\mathfrak{G}} x$，则称分支 M' 是序列$\mathfrak{G}$中分支 M 的过去。在这种情况下，我们记作 $M' \leq_{\mathfrak{G}} M$。对于分支 M，我们用 $\check{}M$表示 M 中具有过去的$\bar{}$标签，用 M 表示那些没有$\check{}\bar{}$ $\check{}\bar{}$过去的标签。于是 $M = M \cup M$且 $M \cap M = \varnothing$。现在考虑最顶层 L 中的一个分支 M 及其所有过去，即满足 $M' \leq_{\mathfrak{G}} M$的分支 M'。

令 M_0、M_1、..、M_k为 M的所有过去，它们满足$M_i \neq \varnothing$，并且当 $0 \leq i \leq j \leq k$ 时满足 $M_i \leq_{\mathfrak{G}} M_j$ （见图 14）。对于所有 i，都有 $|\bar{M}_i| \leq 2^n + 1$，因为 M_i 中的每个标签都是通过定义 7.5 的选项 2 创建的，并且对应于一个不满足的 $\Diamond^\bullet$-公式（但 M_i中的第一个标签除外；该标签可能是通过层提升创建的，对应于一个不满足的 $\Box^\circ$ 公式（构造 7.9），也可能是算法步骤 0 中创建的初始标签）。但无论如何，至多经过 2^n 个这样的步骤后，我们会遇到一个与此前某个标签等价的标签（另见引理 7.6.b 的证明）。

现在令 $M^i_j = \{y \in M_i \mid \exists x \in \bar{M}_j.\, x \leq_{\mathfrak{G}} y\}$ 为 M_i 中 M_j 的未来的$\bar{}$集合，其中某个 $j < i$ （见图 14）。由于层提升（构造 7.8）中簇的复制，我们可以有 $|M^i_j| > |\bar{M}_j|$。我们可以限制 M^i_j 的大小，因为在遇到 U-三角循环之前，一个簇只能被复制有限次。然后，在簇重复之前和之后，从 s 和 t 出发的分支长度再次由 $2 \cdot (2^n + 1)$ 界定，否则 U-三角循环将不满足。（我们必须将 $2^n + 1$ 取两次，因为不允许 suricata 标签出现在 s 和 t 之间。）由于一个簇的大小由 2^n 界定，且 $\bar{M}_j$ 中簇的数量由 2^n 界定，因此 M^i_j 的大小由 $2 \cdot 2 \cdot 2^n \cdot 2^n \cdot 2^n = 2^{3n+2}$ 界定。（这是一个非常朴素的论证，该数量还可以限制为小得多的值。

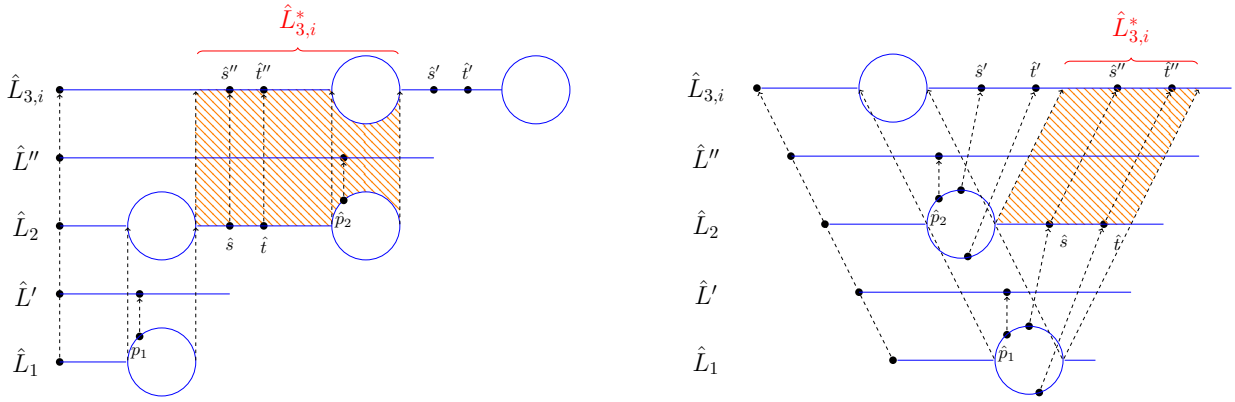


图 13。不满足的R-三角循环的展开（左）；不满足的U三角循环的展开（右）

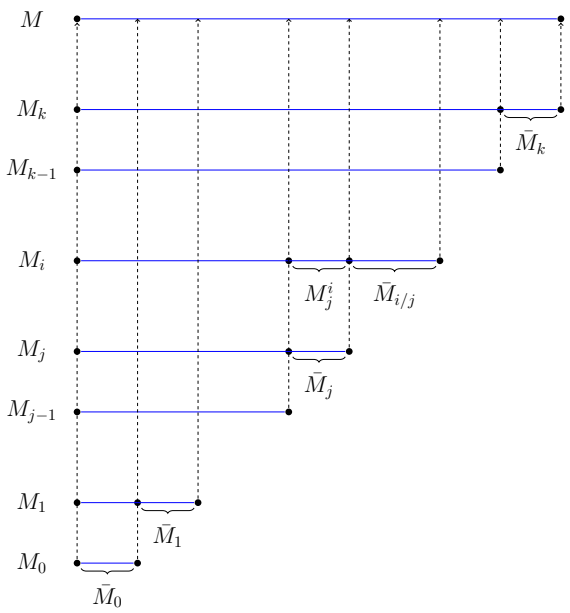


图 14。一个分支及其过去

然而，这篇论文已经足够技术性了，而我们在这里只关心这样一个界限存在这一事实。）

现在我们来为 k设定一个上界。非常直观地，在 $k_{\max} = 2^{2^n}$ 之后我们会看到一个R-三角循环（因为我们需要重复簇 C_1，见定义 7.16 和引理 10.2）。令 M_j 为这个能看到 M_i重复的分支。定义 $M_{i/j}$（对 $j < i$而言）为 M_i 中那些在 M_j中没有过去的标签集合（见图 14）。$|\bar{M}_{i/j}| \geq 2 \cdot 2^n = 2^{n+1}$。那么，若（与上文论证相同），R-三角循环便是不满足的。由于，$2^{3n+2} > 2^{n+1}$ 因此我们有$|M| \leq k_{\max} \cdot |M^i_j| = 2^n \cdot 2^{2^n} \cdot 2^{3n+2} = 2^{2^n+4n+2}$。

请注意，一旦我们观察到不满三角循环，新创建的簇的大小就是$|C| \leq 2^n$。 □

定理 10.4（终止性）。图 7 所示的证明搜索算法是终止的。

证明。 根据前一个引理，层中一个分支的大小受限于序列中的公式 F 因此，算法访问的集合$\mathfrak{S}_i$ 中序列可能出现的层数也受到限制。这限制了序列中层树的高度，因为最终会出现由前一个层模拟的情况。 □