arXiv:2304.09422v1 [cs.CC] 19 Apr 2023

通过合并归结揭示 CDCL 学习的局限

Marc Vinyals Chunxiao Li Noah Fleming Antonina Kolokolova Vijay Ganesh

2023年4月20日

摘要

在其开创性工作中，Atserias 等人以及 Pipatsrisawat 和 Darwiche 于 2009 年分别证明，CDCL 求解器能够以多项式开销模拟归结证明。然而，先前的工作并未探讨这种模拟的紧致性，也就是说，并未研究这一开销究竟需要多大。本文聚焦于采用标准学习方案的 CDCL 求解器所生成证明的一项重要性质，以此探讨这一问题：学习子句的推导中至少有一次推理，其两个前提中都出现同一个文字（即合并文字）。具体而言，我们证明，这类证明能够以至多线性开销模拟归结证明；但也存在必须付出这种开销的公式。更确切地说，存在一些公式，其归结证明长度为线性规模，却需要二次规模的 CDCL 证明。

1 引言

在过去二十年中，CDCL SAT 求解器对软件工程的诸多领域产生了深远影响 [CGP+08]、安全领域 [DVT07、XA05], 以及人工智能 [BF97]。这得益于它们能够求解规模极其庞大的现实世界公式，这些公式包含数百万个变量和子句 [MLM21]。理论研究者和实践工作者都投入了大量精力，以理解 CDCL 算法，以及它在实际应用中表现出非凡效能的原因。尽管已经取得了显著进展，仍有许多问题尚未得到解答。

用于理解 CDCL 算法的工具中，最为成功的一类或许来自证明复杂性。其中一项极具影响力的成果表明，CDCL 的理想化模型可以多项式模拟归结证明系统；这一结果由 Atserias、Fichte 和 Thurley [AFT11], 以及 Pipatsrisawat 和 Darwiche [PD11], 在 Beame 等人 [BKS04] 和 Hertel 等人 [HBPV08]的早期成果基础上，分别独立证明。（另请参阅 Beyersdorff 和 Böhm [BB21]最近给出的另一种证明。）这类模拟结果非常有用，因为它让我们确信，只要某个公式存在简短的归结证明，采用恰当启发式方法的 CDCL 就能复现它。

近期的模型采用了更接近实际求解器的假设，但为此需要在模拟中付出多项式开销。一系列论文致力于研究这些模拟成立所需的假设，并常常在此过程中使用或引入归结的各种改进形式。例如，重启是否必要这一问题虽然仍未解决，但已得到广泛研究；池归结 [Van05]和 RTL [BHJ08]证明系统旨在刻画不进行重启的 CDCL 求解器所产生的证明。近年来，人们也研究了决策启发式方法的重要性，相关结果表明，单靠静态 [MPR20] 或类似 VSIDS 的 [Vin20] 变量排序，都不足以在一般情况下模拟归结（除非定期清除 VSIDS 分数 [LFV+20]）。对于静态排序，人们使用（半）有序归结证明系统 [MPR20] 来研究这类 CDCL 求解器的变体。

但即使我们采用理想化模型，CDCL 模拟归结的效率究竟如何仍不明确。Pipatsrisawat 和 Darwiche 的分析给出了 $O(n^4)$ 的开销——也就是说，如果一个公式

超过 n 个变量，并且有一个长度为 L 的分辨率反驳，那么就存在一个步数不超过 $O(n^4L)$ 的 CDCL 证明。Beyersdorff 和 Böhm [BB21] 将开销改进到了 $O(n^3)$，但我们并不知道最优值是多少。此外，据我们所知，在我们的论文发表之前，人们甚至不知道能否完全消除这种开销。

1.1 CDCL 中的学习方案及其与合并的联系

CDCL 求解器的一个共同特点是采用 1-赋能学习方案 [PD08、AFT11]：也就是说，它们只学习能够实现此前无法进行的单元传播的子句。一个广受欢迎的 1-赋能学习方案是 1UIP 学习方案 [MS99]。为刻画这种行为，我们基于 1-赋能与合并之间的联系展开 [And68], , 即涉及含有共享文字的子句的归结步骤。

如今，几乎所有 CDCL 求解器都采用第一唯一蕴含点（1UIP）学习方案。在该方案中，冲突分析从一个被求解器当前状态赋值为假的子句开始，随后依次将其与导致冲突的单元传播所对应的子句进行消解，直到该子句成为断言子句，也就是说，回跳后立即成为单位子句。

早期对 CDCL 求解器实现的描述 [MS99、MMZ+01] 已经指出，学习断言子句十分重要，因为这会促使求解器转向搜索空间的另一部分。因此，早期的替代学习方案探索了学习多种类型的断言子句。首先要注意，冲突分析可以扩展为生成其他断言子句；在冲突分析过程中，这些子句出现在1UIP 之后，例如中间 UIP 和最后 UIP [BS97]。早期的求解器 GRASP 甚至可以从单次冲突中学习多个 UIP子句。尽管有实证证据表明，通常最好在 1UIP 处停止冲突分析 [ZMMM01],近期研究已经发现，在某些条件下，继续分析下去更有优势 [FB20]（另请参见其中关于学习方案的讨论）。

Ryan [Rya04，§2。5] 还通过实证观察到，子句质量与冲突分析推导的长度呈负相关，并考虑了相反的方法，即在冲突分析过程中，除了学习1UIP 之外，还学习出现在1UIP 之前的子句。据称，这种方法对某些实证基准测试很有用，但与任何学习多个子句的方案一样，与只学习1UIP 的方案相比，它会减慢布尔约束传播（BCP）。

后续研究从更具理论导向的角度，深入理解1UIP 的优势，并研究学习出现在1UIP [DHN07、PD08]之前的子句。特别是，与我们的讨论高度相关的是，Pipatsrisawat 和 Darwiche 确定了 1-赋能是断言子句的一项基本属性。此外，他们还发现了 1-增强子句与合并之间的联系，并利用检查合并的简便性，将其作为 1-赋能的一种近似。

一种正交方法是扩展1UIP 推导，将其与冲突分析中通常不会使用的子句进行归结 [ABH+08]。一个典型例子是子句最小化 [SB09], , 即通过与适当的输入子句进行归结来消除1UIP 子句中的文字，而不考虑这些输入子句在冲突中的作用；因此，实际学习到的子句是1UIP 的更短、因而更强的版本。

此外，在知识编译的背景下，也有人观察到合并与单元消解完备性之间存在关联 [dV94]。最后，人们提出将可直接从公式中推得的合并数量（即通过单个归结步骤推得的合并数量）作为一种名为可合并性的度量，用以解释公式的难度；这一度量既基于受控实验，也基于对实际问题实例的分析 [ZMW+18]。

总而言之，合并之所以与 CDCL 学习方案相关，原因如下：所有实际的 CDCL 学习方案要么生成一个 1-增强子句，要么扩展一个 1-增强子句，而由于 1-增强子句的推导中始终包含合并，因此我们得出所有实际的

学习方案会生成一个子句，其推导中包含合并，这正是我们下面介绍的证明系统所要求的性质。

1.2 我们的贡献

如前所述，我们以 1-赋能与合并之间的联系为基础 [PD08，AFT11]，并引入一种名为 RMA（“带合并祖先的分辨率”）的证明系统，其中包含采用任意 1-赋能学习方案的 CDCL。该系统名称中的“合并祖先”源于这样一个事实：对于任何 1-赋能子句，其归结推导中至少有一步必须消解两个共享同一文字的子句：这就是 合并 步骤，按 [和68]的含义而言。只要子句最小化过程应用于 1-赋能子句之上，也可以用 RMA 对其进行建模。

我们证明，一方面，RMA 能够以线性开销模拟归结。另一方面，我们证明归结与 RMA 之间存在二次分离：也就是说，存在一些公式，其归结证明长度为线性，却需要长度为二次的 RMA 证明。换言之，我们表明，由于标准学习方案的性质，CDCL 的表现可能比归结差多项式倍；但这些性质导致的开销增长不会超过线性。

我们还考虑更弱的证明系统，它们都包含 1UIP（且以更细的粒度实现），但不一定包含其他断言式学习方案。一个值得关注的技术问题是，我们研究的证明系统可以证明并不在限制下封闭；这在证明复杂性中并不常见。这一事实迫使我们的证明利用证明系统的句法性质，而不是依赖更方便的语义性质。

2 预备知识

文字要么是变量 $x^1 = x$，要么是其否定 $x^0 = x$。子句是文字的析取，合取范式公式是子句的合取。子句或 vars(C) 的支撑是其所包含变量的集合。从公式 F 出发的归结推导是一个子句序列 $\eta = C_1, \ldots, C_L$，其中 C_i 要么是 F 中的公理，要么是应用归结规则所得的结论

$$\mathrm{Res}(A \vee x, B \vee \overline{x}) = A \vee B$$

该规则作用于两个前提C_j, C_k，其中 $j,k < i$。在归结推理的前提中以相反符号出现的变量x 称为归结变量。如果*literal* 同时出现在A和B 中，则归结式称为合并。若允许C_i 是C_j 和C_k在语义上蕴含的任意子句，而不要求它由句法推理得出，即使C_j 和C_k 可能无法进行归结，我们就称η 为语义归结推导。如果一个推导的最后一个子句是空子句$\bot$，则它是一个反驳。我们将 $\eta[a,b] = \{C_i \in \eta \mid i \in [a, b]\}$ 记作……。

我们假设，推导中的每个子句都标注了其获取所依据的前提，这使我们能够将证明视为一个有向无环图，其中顶点是子句，边从前提指向结论。当这个有向无环图是一棵树时，我们称该推导为树状推导；当它是一棵蜈蚣树（即最大程度不平衡的树）时，我们称其为输入型推导。

如果每次推理中至少有一个前提是由单个文字组成的单子句，则称该推导为单子句推导。由于输入型归结和单子句归结都不是完备的证明系统，我们分别写作 $F \vdash_i C$（分别为 $F \vdash_1 C$），以表示存在从 F到C的输入型归结推导（分别为单子句归结推导）。

子句 C 在推导 η 中语法依赖于公理 A，如果在 η 的有向无环图表示中存在一条从A 到 C 的路径。这并不意味着推导C需要A，因为不同的推导可能不使用A。

对变量的限制是一个映射 $\rho : X \to X \cup \{0, 1\}$，并依次扩展到文字、子句、公式和反驳，必要时进行化简。我们将 $\rho(x) = *$ 作为 $\rho(x) = x$的简写。众所周知，若 η 是从 F 得出的归结推导，且 ρ 是一个限制，则 $\eta\restriction_\rho$ 是从 $F\restriction_\rho$得出的语义归结推导。

在将限制应用于另一个推导所得到的推导中保留满足子句是很方便的，这样我们就可以使用相同的索引来指代这两个推导。为此，我们使用符号 1，并将其视为一个始终计算为真的子句，它不受任何集合支撑，不依赖于任何子句，也不能在句法上与任何子句进行消解。

通过忽略不必要的子句，可以将语义推导转化为句法推导。形式化地，若 η 是语义归结推导，我们将其句法等价物 $\mathrm{s}(\eta)$ 定义为如下句法归结推导。令 $\mathrm{C} \in \eta$，A 和 B 为 C 的父节点。若 $\mathrm{s}(\mathrm{A}) \vDash \mathrm{C}$，则令 $\mathrm{s}(\mathrm{C}) = \mathrm{s}(\mathrm{A})$；对于 $\mathrm{s}(\mathrm{B})$，类似处理。否则，令 $\mathrm{s}(\mathrm{C}) = \mathrm{Res}(\mathrm{s}(\mathrm{A}),\mathrm{s}(\mathrm{B}))$。不难看出，对每个 $\mathrm{C}_i \in \eta$，都有 $\mathrm{s}(\mathrm{C}_i) \vDash \mathrm{C}_i$。

2.1 CDCL

我们需要定义 CDCL证明中的几个概念。有关深入介绍，请参阅《可满足性手册》 [BN21]。固定一个合取范式 F，也称为子句数据库。轨迹 τ 是元组序列 $(\mathrm{x}_{i} = \mathrm{b}, \mathrm{C}_i)$，其中 C_i要么是 F 中的子句，要么是表示决策的特殊符号 d。我们用 $\alpha_{<i}$ 表示赋值 $\{\mathrm{x}_{i} = \mathrm{b} \mid \mathrm{i}' < \mathrm{i}\}$，并用 $\mathrm{dl}(\mathrm{i}) = \mathrm{dl}(\mathrm{i} - 1) + [\![\mathrm{C}_i = \mathrm{d}]\!]$ 表示位置 i 处的决策层级，即截至 i 的决策数量。我们用 i^*标记轨迹中最后一个决策的位置。

如果对于每个i 位置（该位置不是决策），都有$C_i\restriction_{\alpha_{<i}} = x^{b}{}_{j_i}$，并且对于每个决策$i$，对于满足 $C \in F$的每个子句$C\restriction_{\alpha_{<i}} = x_b$，文字$x_b$都在轨迹中出现在$i$之前，则称轨迹有效。特别地，对于每个位置$i' < i$，若 $\mathrm{dl}(i') < \mathrm{dl}(i)$，则有$C_i\restriction_{\alpha_{<i'}} \neq x_{b\,j_i}$。

一个子句 C 如果在轨迹中的最后一个决策处是单元的，则称其为断言子句，即 $C\restriction_{\alpha_{<i^*}} = x^b$。如果一个子句 C 由 F 蕴含，并且将其添加到 F 后能够引发新的单元传播，则称其为 1-增强子句。也就是说，存在一个文字 $\ell \in C$，使得对于某个 $A \in \{\bot, \ell\}$，有 $F \wedge C \setminus \ell \vdash_1 A$。如果一个子句不是 1-增强子句，则称其被 F 吸收。

给定一个被轨迹 τ 否定的子句 $D_{|\tau|}$，冲突推导是一个输入推导 $D_{|\tau|}, \ldots, D_{k+1}, D_k$，其中，如果 $x_{j_i} \in D_i$，则 $D_{i-1} = \mathrm{Res}(D_i, C_i)$，否则 $D_{i-1} = D_i$。推导中第一个（即索引最大的）断言子句称为 1UIP。注意，D_{i^*} 始终是断言子句（因为对于 $i^* \leq i \leq |\tau|$，D_i 被 $\alpha_{\leq i}$ 否定，而 D_{i^*} 未被 $\alpha_{<i^*}$ 否定），因此我们可以假设 1UIP 的索引始终至少为 i^*。

如果一系列输入推导中，输入结构化指每个推导的最后一个子句都可在后续推导中用作公理。除最后一个推导外，每个推导的最后一个子句都称为引理。CDCL 推导是由冲突推导组成的输入结构化序列，其中学习子句是引理。此定义与带输入引理的归结树 [BHJ08],的定义相似，不同之处在于，该序列只需按顺序排列，而无需对整体证明施加任何额外的树结构。

以下引理通过将合并与 1UIP、断言子句和 1-增强子句联系起来，凸显了合并在实践中的重要性。

引理 2.1 ([PD08，命题 2])。 如果一个子句是断言子句，那么它就是 *1*-增强的。[1]

引理 2.2 ([AFT11，引理 8])。 如果 $A \vee x$和 $B \vee \overline{x}$均被吸收，但 $A \vee B$ 是 *1*-增强的，那么$A \vee B$ 就是一次合并。特别地，如果一个子句是 *1*-增强的，那么它的推导中包含一次合并。

引理 2.3. *1UIP* 子句是一个合并。

[1]原始结果并未证明 1-一致性，但证明过程是类似的。

证明。 设 $D^j = \mathrm{Res}(C^{j+1}, D_{j+1})$ 为 1UIP。一方面，由于轨迹中的每个子句都包含至少两个处于其出现所在决策层级的文字，C^{j+1} 包含两个处于最后一个决策层级的文字。另一方面，任何不在轨迹中的子句也包含两个处于最后一个决策层级的文字，尤其是$D^{|\tau|}$。由于 $|D_{i+1} \setminus D_i| \leq 1$ 且D^{j+1} 不是断言子句，它也包含两个处于最后一个决策层级的文字。

我们计算了前提 D^j 中最后一个决策层级上的 4 个文字，其中有 2 个因作为枢轴文字而未出现在结论中。为了使 D^j 在最后一个决策层级上只包含一个文字，其余两个文字必须相等。

□

3 证明系统

我们用输入结构化框架来定义证明系统。每个归结证明都可以看作是输入结构化的：将其视为一系列单位长度输入归结，并将每个子句视为一个引理；对允许作为引理的子句施加不同限制，便可得到不同的证明系统。图1 中的示意图有助于梳理这些证明系统。

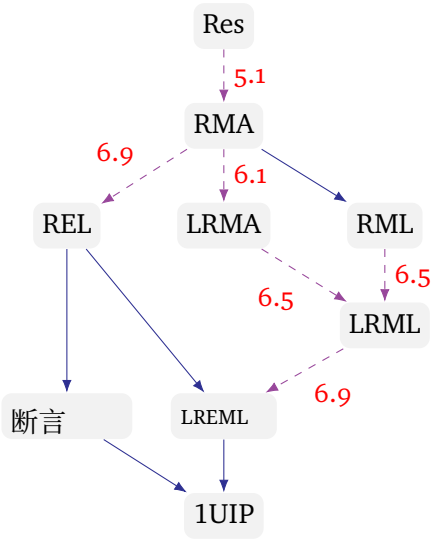

图1：证明系统之间的关系。实线箭头 A B 表示 A可以模拟 B 且无需开销。虚线箭头 A B 表示 A可以模拟 B 且无需开销，但 B 需要线性开销才能模拟A。文中标注了证明分离的论述出处。

Andrews 对合并归结的定义 [并68]考虑满足额外限制的树状证明：每次推理中，至少一个前提是公理或合并。他还指出，这类推导可以构造成输入结构化形式。

观察 3.1 ([并68])。 树状合并分辨率推导可以分解为一个输入结构化序列，其中所有引理都是合并。

处理这类推导时，这一观察至关重要；第 4 节和 A 节显然体现了这一点，因此我们也将其作为定义合并归结的一种替代方式。

Andrews 的主要结论是，合并限制不会影响树状归结。

引理 3.2 ([And68，引理 5])。 若存在一个长度为 L 的树状归结推导，其结论为 C，且至多只有根节点是合并式的，则存在某个 $C' \subseteq C$ 的输入分辨率推导，其长度至多为 L。

定理 3.3 ([And68，定理 1])。 若存在一个长度为 L 的树状归结推导，其结论为 C，则存在某个 $C' \subseteq C$ 的树状合并分辨率推导，其长度至多为 L。

如果我们从合并分辨率证明的输入结构化视角中取消树状限制，就会得到一种介于树状与 DAG 式归结之间的证明系统。在该系统中，子句可以重复使用（即出度大于 1），当且仅当它们是合并，换言之，是输入结构化分解中的引理。我们将此证明系统称为带合并引理的分辨率，并以缩写 RML 指代。

定义 3.4. RML 推导是由单子句归结推导组成的输入结构化序列，其中所有引理都是合并。

使用以下方法的求解器所产生的 CDCL 反驳1UIP 学习方案呈 RML 形式，这是引理 2.3 的推论。我们还可以推广 RML，使其允许重用推导中任意位置包含合并的子句。我们将此证明系统称为带合并祖先的分辨率，简称 RMA。

定义 3.5。 RMA 推导是一种输入结构化序列，由单子句归结推导组成，其中除最后一个推导外，所有推导都包含一次合并。

注意，根据引理 3.2 ，要求 RMA 推导中的推导序列为输入推导，还是允许一般树，并无区别。事实上，我们的下界结果对于一种更一般的证明系统仍然成立：我们只要求每个出度大于 1 的子句都有一个作为合并的祖先。这种证明系统没有简单的输入结构，而可以被看作一系列树状归结推导，其根节点为合并，随后使用前述推导的根节点作为公理进行标准归结推导。

为了重新建立与 CDCL 的联系，我们可以定义一种称为带赋能引理的分辨率的证明系统，它能够刻画使用任意断言学习方案或 1-赋能学习方案的求解器所生成的 CDCL 反驳。

定义 3.6。 设 $C_1, \ldots, C_{L-1}$ 为一个输入结构化单位推导序列的引理。若对于所有 $i \in [1, L-1]$，该序列是公式 F 的带赋能引理的分辨率（REL）推导，即 C_i 相对于 $F \cup \{C_j : j < i\}$ 是 1-赋能的。

由引理2.1和 2.2 可知，这类反驳具有 RMA 形式。

观察 3.7。 *REL 推导是 RMA 推导。*

使用 REL 证明系统而不是基于合并的对应系统，似乎更自然，因为 REL 正是通过 1-增强性质来定义的。然而，合并性质很容易检查，因为它仅与当前推导局部相关；而要判断一个子句是否具有 1-增强性质，我们只能查看推导的完整历史，尤其是此前有哪些引理。这使得 REL 难以分析。此外，CDCL 求解器产生的反驳若在断言子句的基础上应用了子句最小化方案，可能不具有 REL 形式，但仍具有 RMA 形式。

CDCL 求解器产生的输入推导还有一个性质：一旦某个变量被消解，它之后就不会再出现在推导中。

定义 3.8. 若对于每个归结步骤 i，枢轴变量 x_i 都不属于任何子句 $C_i \in \eta[i,L]$ 的支撑，则称归结推导 η 是强正则的。若一个推导序列中的每个推导都是强正则的，则称该推导序列是局部正则的。
LRML 推导（分别地，LRMA）是局部正则的 RML 推导（分别地，RMA）。

最后，我们可以考虑具有赋能合并引理且满足局部正则性的推导。这些推导仍包括 1UIP 证明。

定义 3.9。 LREML 推导是同时满足 LRML 和 REL 的推导。

由 CDCL 对归结的模拟可知 [PD11、AFT11]，我们定义的所有（DAG 型）证明系统都能多项式模拟标准归结。在第 4 节中，我们将更精确地描述这种模拟，并证明其模拟开销可以降至线性；在第 5 节中，我们将证明这种模拟是最优的，因为存在一些公式，其归结反驳长度为线性，但所需的 RMA 反驳长度为二次。

4 模拟

作为在 RML 中模拟归结的辅助工具，我们定义集合D的输入归结闭包，记为 $\mathrm{Cl_i}(D) = \{C \mid \exists C' \subseteq C、D \vdash_i C'\}$，它是通过输入归结加上弱化从$D$ 可推导出的子句集合。众所周知，由于输入归结推导可以在不失一般性的前提下假设为强正则，我们也可以假设它们的长度至多与变量数呈线性关系。

观察 4.1。 如果 D 是一个包含 n 个变量的合取范式公式，且 $C \in \mathrm{Cl_i}(D)$，那么存在一个从 D 推导出某个 $C' \subseteq C$ 的强正则输入归结推导，其长度至多为 n。

结合定理 3.3 以及这样一个想法：为了模拟归结推导，我们不需要生成每个子句，只需完成足够的工作，使得在接下来的步骤中我们可以假装已经推导出了该子句 [PD11、AFT11], 我们可以证明，合并归结以变量数的至多线性乘法开销模拟归结。

定理 4.2。 如果 F 是一个含有 n 个变量、具有长度为 L 的归结反驳的合取范式公式，那么它具有长度为$\mathrm{O}(nL)$的 *RML* 反驳。

证明。 设 $\pi = (C_1, \ldots\ldots, C_L)$ 是一个归结反驳。我们构造集合序列 $D_0, \ldots\ldots, D_L$，使其具有以下性质。

1. $D_t \setminus F$ 是 RML 推导中的引理集合，其长度至多为 $(2n+1)t$。
2. $\pi[1,t] \subseteq \mathrm{Cl_i}(D_t)$.

这足以证明该定理：由于 $\bot \in D_t$，我们可以在长度为 $\bot$ 的情况下从D_t 中得到它，因此该反驳的总长度为 $(2n+1)L + n$。

我们通过归纳构造这些集合，从 $\mathrm{D_0} = \mathrm{F}$ 开始。假设我们已经构造了 D_t，并且有 $\mathrm{C} = \mathrm{C}_{t+1} = \mathrm{Res(A, B)}$，其中 $\mathrm{A, B} \in \pi[1, \mathrm{t}]$。如果 $\mathrm{C} \in \mathrm{Cl_i}(\mathrm{D}_t)$，则令 $\mathrm{D}_{t+1} = \mathrm{D}_t$，并且证明完毕。否则，根据归纳假设，$\mathrm{A, B} \in \mathrm{Cl_i}(\mathrm{D}_t)$，因此由观察 4.1，存在 $\mathrm{A}' \subseteq \mathrm{A}$ 和 $\mathrm{B}' \subseteq \mathrm{B}$ 的输入归结推导，长度至多为 n。由于 $\mathrm{A}' \nvDash \mathrm{C}$ 且 $\mathrm{B}' \nvDash \mathrm{C}$ 均不成立，A'和 B' 可以进行归结，因此存在一个从 D_t 导出 $\mathrm{C}' \subseteq \mathrm{C}$ 的树状推导 η，长度至多为 $2\mathrm{n}+1$。由定理 3.3，存在一个从 D_t导出 $\mathrm{C}'' \subseteq \mathrm{C}$ 的树状合并分辨率推导 η'，长度至多为 $2\mathrm{n}+1$。由观察 3.1，该推导 η'可以分解为一系列输入推导，总长度至多为 $2\mathrm{n}+1$。令 E 为该序列中的引理，并令 $\mathrm{D}_{t+1} = \mathrm{D}_t \cup \mathrm{E}$。我们有 $\mathrm{C} \in \mathrm{Cl_i}(\mathrm{F} \cup \mathrm{E}) \subseteq \mathrm{Cl_i}(\mathrm{D}_{t+1})$，并且可以在至多 $2\mathrm{n}+1$ 步内从 D_t 得到 E。因此，D_{t+1} 具有所有所需的性质。

□

如果考察η 的结构，我们可以更精确地描述模拟，然后再应用定理 3.3。令 A_M 和 B_M 分别为 A' 和B' 的输入推导中的最后一次合并，并令 $E = \{A_M、B_M\}$。

现在考虑从A' 到A_M 的输入推导片段A'，类似地考虑B'。我们有一个C' 的树状推导，其中至多根节点是合并，因此可以直接应用引理3.2，而不是定理3.3，从而得到一个$C'' \subseteq C$ 从$E \cup F$得到的输入归结推导。

如果我们还确保A'和B'的输入推导是强正则的，那么 LRML 也可以模拟归结，其开销为 O(n)，与 RML 相同。

根据以下引理，可以得到关于 LREML 的类似结果。

引理 4.3 （[PD11]）。 如果 F 吸收 $A \vee x$ 和 $B \vee \neg x$，则 $F \vdash_i C' \subseteq A \vee B$。

推论 4.4。 如果 F 是一个定义在 n 个变量上的 *CNF* 公式，且具有长度为 L 的归结反驳，那么它具有一个长度为$O(nL)$的 *LREML* 反驳。

证明。 证明遵循定理 4.2 的一般结构，但我们使用一系列步骤D_{jt}来构造 D_t。我们的归纳假设是，D_{jt}可以从 D_t通过 p 个 LREML 推理步推导得到，并且A'和B'可以从D_{jt}通过 q 步推导得到，其中 $p+q \leq 2n$。

基例 $D^0{}_t = D_t$ 是显然的。

对于归纳情形，不失一般性地假设导向A' 和B' 的输入推导是强正则的。根据引理4.3 ，A' 或B'是 1-增强的，设为A'。令C 为A'的推导中第一个 1-增强子句。根据引理2.2 ，C 是一个合并，因此我们可以取$D^{j+1}{}_t = D^j{}_t \cup \{C\}$。

□

5 分离

我们证明标准归结与 RMA 之间的如下分离。

定理 5.1. 存在一个公式族 F_n，含有$O(n\log n)$ 个变量和$O(n\log n)$ 个子句，其归结反驳长度为$O(n\log n)$，但每个 *RMA* 反驳都需要长度为 $\Omega(n^2 \log n)$。

5.1 公式

令 ℓ、m、n 为正整数。设有变量 x_i，其中 $i \in [m\ell - 1]$，以及 $w_{j,k}$，其中 $j \in [\ell]$，$k \in [n]$。为方便起见，我们定义 $x_0 = 1$ 和 $x_{m\ell} = 0$，它们不是变量。令 $X = \{x_i \mid i \in [m\ell - 1]\}$，$W_j = \{w_{j,k} \mid k \in [n]\}$，且 $W = \bigcup_{j\in[\ell]} W_j$。对于每个 $j \in [\ell]$，我们构造如下构件：

$$w_{j,k} = w_{j,k+1} \qquad \text{for } k \in [n-1] \tag{1}$$

每个等式都展开为两个子句 $B_{j,k,1} = w_{j,k} \vee w_{j,k+1}$和 $B_{j,k,0} = \overline{w_{j,k}} \vee \overline{w_{j,k+1}}$，我们将它们统称为 $\mathcal{W} = \{B_{j,k,b} \mid j \in [\ell], k \in [n-1], b \in \{0,1\}\}$。注意，第 j 个构件蕴含 $w_{j,1} = w_{j,n}$。此外，我们构造如下构件：

$$(w_{1,1} = w_{1,n}) \to x_1 \tag{2}$$

$$(w_{\hat{\imath},1} = w_{\hat{\imath},n}) \to (x_{i-1} \to x_i) \qquad \text{for } i \in [2, m\ell - 1] \tag{3}$$

$$(w_{\ell,1} = w_{\ell,n}) \to \overline{x_{m\ell-1}} \tag{4}$$

其中 $\hat{\imath} \in [\ell]$ 表示 $i \pmod{\ell}$的规范形式。每个约束都展开为以下两个子句：$A_{i,1} = w_{\hat{\imath},1} \vee w_{\hat{\imath},n} \vee x_{i-1} \vee x_i$ 和 $A_{i,0} = w_{\hat{\imath},1} \vee \overline{w_{\hat{\imath},n}} \vee x_{i-1} \vee x_i$，我们统称它们为$\mathcal{X} = \{A_{i,b} \mid i \in [m\ell], b \in \{0,1\}\}$。所得公式称为 $F_{\ell,m,n}$。

5.2 上界

不难看出，存在一个对 $F_{\ell,m,n}$ 的分辨率反驳，其长度为 $O(\ell\cdot(m+n))$。事实上，我们首先对每个 $j\in[\ell]$ 推导出表示 $w_{j,1}=w_{j,n}$ 的两个子句，这需要 $O(n\ell)$ 步：

$$\dfrac{\dfrac{\dfrac{w_{j,1}\vee\overline{w_{j,2}}\qquad w_{j,2}\vee\overline{w_{j,3}}}{w_{j,1}\vee\overline{w_{j,3}}}}{\vdots}}{\dfrac{w_{j,1}\vee\overline{w_{j,n-1}}\qquad w_{j,n-1}\vee\overline{w_{j,n}}}{w_{j,1}\vee\overline{w_{j,n}}}} \tag{5}$$

接着，我们将每个 $\mathcal{X}$ 公理与这些子句之一进行归结，并适当选择子句，从而得到形如 $w^b_{\hat{\imath},1}\vee x_{i-1}\vee x_i$（其中 $i\in[m\ell]$）的子句对；再对每对子句进行归结，得到蕴含链 $x_1,\dots,x_i\to x_{i+1},\dots,x_{n\ell-1}$，共需 $O(m\ell)$ 步。

$$\dfrac{\dfrac{w_{\hat{\imath},1}\vee\overline{w_{\hat{\imath},n}}\qquad w_{\hat{\imath},1}\vee w_{\hat{\imath},n}\vee\overline{x_{i-1}}\vee x_i}{w_{\hat{\imath},1}\vee\overline{x_{i-1}}\vee x_i}\qquad\dfrac{\overline{w_{\hat{\imath},1}}\vee w_{\hat{\imath},n}\qquad\overline{w_{\hat{\imath},1}}\vee\overline{w_{\hat{\imath},n}}\vee\overline{x_{i-1}}\vee x_i}{\overline{w_{\hat{\imath},1}}\vee\overline{x_{i-1}}\vee x_i}}{\overline{x_{i-1}}\vee x_i} \tag{6}$$

既然我们已经推导出蕴含链 $x_1, x_1\to x_2,\dots,x_{m\ell-1}\to x_{m\ell-1}, x_{m\ell-1}$，就可以再用 $O(m\ell)$ 步完成反驳。下面记录上述讨论。

引理 5.2. *$F_{\ell,m,n}$ 存在一个长度为 $O(\ell\cdot(m+n))$ 的分辨率反驳。*

在证明下界之前，我们先非正式地讨论一下在 RML 中反驳这个公式的自然方法，以便了解需要排除哪些行为。

如果我们尝试重现前面的分辨率反驳，由于无法重用表示 $w_{j,1}=w_{j,n}$ 的子句（因为它们不是合并），因此每次需要时都必须重新推导它们，这意味着推导蕴含链 $x_1,\dots,x_i\to x_{i+1},\dots,x_{n\ell-1}$ 需要 $O(mn\ell)$ 步。我们称这种方法为反驳证明 1。生成 $w^b_{\hat{\imath},1}\vee x_{i-1}\vee x_i$ 时，此反驳中发生了合并（对 $w_{\hat{\imath},1}, x_{i-1}$ 和 x_i）；生成 $x_{i-1}\vee x_i$ 时也发生了合并（对 x_{i-1} 和 x_i）。但由于我们从未重用这些子句，因此该反驳实际上是树状的。

另一种方法，我们称之为反驳 2，是改为从 $\mathcal{X}$ 公理着手。在这个证明中，我们将所有形如 $w_{j,1}\neq w_{j,n}$ 的重复约束归并在一起，其中每个 $j\in[\ell]$ 都对应一条约束，然后一次性将它们消解出去。换句话说，我们首先推导出以下约束序列

$$D_i=\left(\bigvee_{\hat{\imath}\in[\min(i,\ell)]} w_{\hat{\imath},1}\neq w_{\hat{\imath},n}\right)\vee x_i \qquad\qquad \text{for } i\in[m\ell]\ , \tag{7}$$

其中，可以从 D_{i-1} 和一对 $\mathcal{X}$ 公理 $A_{i,b}$ 中得到 D_i，然后使用这些 $\mathcal{W}$ 公理消解掉 $D_{m\ell}=\bigvee_{j\in[\ell]} w_{j,1}\neq w_{j,n}$ 中的不等式。然而，表示任意约束 D_i（其中 $i\geq\ell$）都需要 2^ℓ 个子句，这远大于 $mn\ell$；当 ℓ 足够大时，甚至是超多项式级别的，因此这种反驳同样效率不高。注意，每当我们使用 $i\geq\ell$ 推导出 D_i 时，这个反驳都会进行一次合并（涉及 W 变量）。

构造反驳的第三种、略显牵强的方法是：通过一种最后一步为合并的推导，推导出表示 $w_{j,1}=w_{j,n}$ 的子句对，以便重用。通过稍微调整反驳证明 1，在每当 $i\equiv j \pmod{\ell}$ 时，将 $x_i\to x_{i+1}$ 的每个推导替换为 $w_{j,1}\vee w_{j,n}\vee x_i\vee x_{i+1}$ 的推导，即可分别在 $O(mn\ell)$ 步内推导出这些子句，总计需要 $O(mn\ell^2)$ 步，这样最终得到的是 $w_{j,1}\vee w_{j,n}$，而不是 the

空子句。例如，可以通过对$w_{j,1} \vee w_{j,2} \vee x_i \vee x_{i+1}$与$w_{j,2} \vee w_{j,n} \vee x_i \vee x_{i+1}$进行归结，按如下方式得到这样的替换子句。

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{w_{j,2} \vee \overline{w_{j,3}} \quad w_{j,3} \vee \overline{w_{j,4}}}{w_{j,2} \vee \overline{w_{j,4}}}}{\vdots}}{w_{j,2} \vee \overline{w_{j,n-1}}} \qquad w_{j,n-1} \vee \overline{w_{j,n}}}{w_{j,2} \vee \overline{w_{j,n}}} \qquad \dfrac{w_{i,1} \vee w_{i,n} \vee \overline{x_{i-1}} \vee x_i \qquad \dfrac{w_{i,1} \vee \overline{w_{i,2}} \quad \overline{w_{i,1}} \vee \overline{w_{i,n}} \vee \overline{x_{i-1}} \vee x_i}{\overline{w_{i,2}} \vee \overline{w_{i,n}} \vee \overline{x_{i-1}} \vee x_i}}{w_{i,1} \vee w_{i,2} \vee \overline{x_{i-1}} \vee x_i}}{w_{i,1} \vee \overline{w_{i,n}} \vee \overline{x_{i-1}} \vee x_i} \tag{8}$$

在推导出$w_{j,1} = w_{j,n}$作为合并后，我们遵循反驳1的后续步骤，并在$O(m\ell)$个步骤内完成反驳。我们将其称为反驳 *3*。

注意，推导出表示$w_{j,1} = w_{j,n}$的子句所需的最小长度仅为$O(n)$，即使在 RML 中也是如此。因此，如果我们只利用反驳3 包含这些子句这一信息，那么只能将其长度界定为 $\Omega(\ell \cdot (m+n))$。因此，当我们计算推导某个子句的难度时，不仅需要考虑其语义，还需要考虑它在句法上是如何得到的。

5.3 下界

在正式证明下界之前，我们先作两个有用的观察 .

引理 5.3。 设 η 是一个仅依赖于 $\mathcal{W}$ 公理的归结推导。那么 η 不包含任何合并，并且所有子句都支撑在 W 上。

证明。 我们通过归纳证明 η 中的每个子句都具有以下形式：$w_{j,k} \vee \overline{w_{j,k'}}$，其中$k \neq k'$。对于公理，这一点显然成立。根据归纳假设，对$w_{j,k}$ 进行的一般归结步骤具有以下形式：

$$\dfrac{w_{j,k} \vee \overline{w_{j,k'}} \qquad \overline{w_{j,k}} \vee w_{j,k''}}{w_{j,k''} \vee \overline{w_{j,k'}}} \tag{9}$$

并且尤其不是一次合并。 □

引理 5.4. 设 η 是一个子句 C 的归结推导，该子句由 W 变量支撑，且使用一个 $\mathcal{X}$公理。那么 η 对每个 $i \in [m\ell]$至少使用一个 $A_{i,b}$ 公理。

证明。 我们证明其逆否命题，并假设存在一个被使用的公理 $A_{i,b}$ ，且以下两种情况之一成立：要么 $A_{i+1,0}$ 和 $A_{i+1,1}$ 均未被使用，要么 $A_{i-1,0}$ 和 $A_{i-1,1}$ 均未被使用。在第一种情况下，文字 x_i出现在从 $A_{i,b}$ 到 C的路径上的每个子句中，这与 C 由 W 个变量支撑相矛盾。第二种情况下，文字 x_{i-1} 也有类似结论。

□

我们朝着证明下界迈出的第一步，是排除反驳 2 规模可能很小的情况。为此，我们证明宽子句几乎无法带来任何进展。这是证明复杂性中的一个常见主题，标准工具是对一个简短的反驳应用随机限制，以获得一个窄反驳。然而，正如我们稍后在推论 5.12中证明的那样，RMA 在限制下并不封闭，因此我们需要分别论证哪些合并得以保留。

让我们定义所使用的限制类，以及这些限制需要如何尊重公式的结构。限制是一个自足赋值 [MS 85] ，相对于子句集合 D 而言，如果它满足其所触及的每个子句；换句话说，对于每个子句 $C \in D$ ，要么 $C\restriction_\rho = 1$ ，要么 $C\restriction_\rho = C$。如果限制相对于 $\mathcal{W}$ 公理是一个自足赋值，则称其为 k-尊重的；我们有 $F_{\ell,m,n}\restriction_\rho \cong F_{k,m,n}$ 直到变量

重命名，并且每个X变量都映射到一个X变量。我们对窄子句的定义也针对当前公式进行了调整，并统计子句C所提及的不同W-块的数量。形式化地说，$\mu(C) = |\{j \in [\ell] \mid \exists x_{j,k} \in \mathrm{vars}(C)\}|$。

引理 5.5. 设 π 是 $F_{\ell,m,n}$ 的一个长度为 $L = o((4/3)^{\ell/8})$的分辨率反驳。存在一个$\ell/4$-相容的限制 ρ，使得 $\pi\restriction_\rho$ 中的每个子句都有 $\mu(C) \le \ell/8$。

证明。 我们使用概率方法。考虑以下在 $\{0,1,*\}^\ell$上的分布 $\mathcal{J}$：每个坐标独立地按 $\Pr[J_i = 0] = \Pr[J_i = 1] = 1/4$、$\Pr[J_i = *] = 1/2$ 的概率选取。给定一个按此分布采样的随机变量 $J \sim \mathcal{J}$，我们如下导出一个随机限制 ρ： $\rho(w_{j,i}) = J_j$, $\rho(x_i) = *$，如果 $J_i = *$；否则 $\rho(x_i) = \rho(x_{i-1})$（其中 $\rho(x_0) = 1$）。

注意，$F_{\ell,m,n}\restriction_\rho \cong F_{|J^{-1}(*)|,m,n}$ 在变量重命名下等价，并且根据切尔诺夫界，有 $\Pr[|J^{-1}(*)| < \ell/4] \le e^{-\ell/16}$。

对于每个子句 $C \in \pi$ 且满足 $\mu(C) > \ell/8$，我们还有：

$$\Pr[C\restriction_\rho \neq 1] \le (3/4)^{\mu(C)} \le (3/4)^{\ell/8} \ . \tag{10}$$

因此，由并集界可知，$|J^{-1}(*)| < \ell/4$ 或任意子句满足 $\mu(C\restriction_\rho) > \ell/8$ 的概率严格小于 1，因此我们得出结论：存在一个限制 ρ，满足该引理的结论。

□

注意，$s(\pi\restriction_\rho)$ 是F_n的归结反驳，$\ell\restriction_\rho$但不一定是 RMA 反驳，因此我们无法控制哪些子句可以重复使用[2]。尽管如此，我们可以找出$s(\pi\restriction_\rho)$中的一个片段，在其中我们仍然掌握足够的信息。

引理 5.6。 存在一个整数 t，使得 $\psi = s(\pi[1,t]\restriction_\rho)$ 是一个归结推导，其结论是一个由 W 变量支撑的子句，该子句依赖于一个 $\mathcal{X}$ 公理，且没有任何由 W 变量支撑的子句被重复使用。

证明。 令 $C_t \in \pi$ 为第一个依赖于一个 $\mathcal{X}$ 公理且满足 $D_t = s(C_t\restriction_\rho)$ 由 W 支撑的子句；这样的子句确实存在，因为 $\bot$ 就是这样一个子句。

根据 t 的定义，每个由 W 变量支撑的祖先 $D_k \in \psi$ of D_t，都对应于 C_k 中的一个子句 π，且该子句仅依赖于 $\mathcal{W}$ 公理，因此根据引理 5.3 C_k 不是合并。根据 RMA 的定义，C_k 没有被重复使用；根据 $s(\cdot)$ 的构造，D_k也没有被重复使用。

还需证明 D_t 依赖于一个 $\mathcal{X}$ 公理。由于 C_t 依赖于一个 $\mathcal{X}$ 公理，其前驱 C_p 和 C_q 中至少有一个也依赖于公理；不妨设为 C_p。根据 t 的定义，$D_p = s(C_p\restriction_\rho)$ 不由 W 支撑，因此根据引理 5.3，D_p 依赖于一个 $\mathcal{X}$ 公理，或者 $D_p = 1$。类似地，如果 C_q 也依赖于一个 $\mathcal{X}$ 公理，那么 $D_q = s(C_j\restriction_\rho)$ 也依赖于公理（或者它为 1），于是结论成立。否则，C_q 的形式为 $w_{j,k} \vee w_{j,k'}$，并且要么被 ρ 满足，要么未受影响。在这两种情况下，我们都有 $D_q \not\models C_t\restriction_\rho$（第一种情况显然成立；第二种情况是因为 D_q 包含归结变量，而 C_t 不包含），因此 D_t 依赖于 D_p。

□

注意，C_t 在语义上可能蕴含于 $\mathcal{W}$ 公理，并且像反驳3中那样具有较短的推导，因此我们不得不使用句法论证来说明，使用一个C_t 公理 $\mathcal{X}$ 需要许多归结步骤。

下一步是将 ψ 划分为 m 个（可能相交的）部分，每个部分大致对应于 ψ 中使用 $\mathcal{X}$ 其变量位于长度为 ℓ 的区间内的公理（根据引理5.4，我们可以假设 ψ 包含来自每个区间的公理）。为此，我们使用以下为 $i \in [n]$定义的限制族：

$$\sigma_i(x_{i'}) = \begin{cases} 1 & \text{if } i' \le i\ell \\ * & \text{if } i\ell < i' \le (i+1)\ell \\ 0 & \text{if } (i+1)\ell < i' \end{cases} \qquad \sigma_i(w_{i',j}) = * \tag{11}$$

[2]回想一下，$s(\pi)$是 π的句法等价物。

令 $X_i = X \cap \sigma^{-1}{}_i(*)$，并注意 $F_{\ell,\, m,\, n}\restriction_{\sigma_i} \cong F_{\ell,\, 1,\, n}$。

含有 ψ 许多 X 变量的子句可能难以分类，但直观上，只需考察最小的正文字和最大的负文字就足够了，因为它们最难消除。因此，我们将 $r(C)$ 定义为对一个子句进行如下操作：保留关于 W变量的文字不变，删除除最小者之外的所有正 X 文字，并删除除最大者之外的所有负 X文字。形式化地，

$$r\left(\bigvee_{i\in A} x_i \vee \bigvee_{i\in B} \overline{x_i} \vee \bigvee_{(i,j)\in C} w_{i,j}^{b_{i,j}}\right) = x_{\min A} \vee \overline{x_{\max B}} \vee \bigvee_{(i,j)\in C} w_{i,j}^{b_{i,j}} \tag{12}$$

其中，如果 A（分别为 B）为空，则省略 $x_{\min A}$（分别为 $x_{\max B}$）。

我们需要 $r(C)$具有以下性质。

引理 5.7。 如果 $C\restriction_{\sigma_i} \neq 1$ 且 $\mathrm{vars(r(C))} \cap X_i = \emptyset$，那么 $C\restriction_{\sigma_i}$ 由 W 变量支撑。

证明。 $\mathrm{vars(r(C))} \cap X_i = \emptyset$ 这一假设意味着，C 中最小的正 X 文字要么不大于 $i\ell$，要么大于 $(i+1)\ell$；但 $C\restriction_{\sigma_i} \neq 1$ 这一假设排除了第一种情况。因此，所有正 X 文字都被 σ_i满足。类似地，最大的负 X 文字不大于 $i\ell$，所有负 X 文字也都被满足。

□

我们定义每个部分 ψ_i 由满足以下条件的所有子句 $C \in \sigma$ 组成：C 是

1. 一个未被 σ_i满足的 $\mathcal{X}$ 公理；或
2. 一个归结变量属于 X_i 的推理结论；或
3. 一个归结变量属于 W 的推理结论，且该推理依赖于一个 $\mathcal{X}$ 公理，前提是 $r(C)$包含一个属于 X_i 的变量；或
4. 一个归结变量属于 W 的推理结论，且该推理不依赖于 $\mathcal{X}$ 公理，前提是 唯一的直接后继 C 属于 ψ_i。

这是证明中至关重要的一步：我们在此利用原始推导具有 RMA 形式这一点。由于不依赖于 $\mathcal{X}$ 公理的子句不是合并，它们只有一个后继，因此该定义是良构的。

理想情况下，我们希望证明这些部分 ψ_i 两两不相交。虽然这并不完全成立，但它们的重叠程度仍然不大。

引理 5.8。令 ψ 和 $\{\psi_i \mid i \in [\ell]\}$ 如上所述。则 $2|\psi| \geq \sum_i |\psi_i|$。

证明。 公理最多可能出现在两个不同的 ψ_i 中，而与X 归结变量进行归结后得到的子句只会出现在一个中。其他依赖于 $\mathcal{X}$ 公理且可能出现在不同ψ_i 中的子句，只有与W 归结变量进行归结后得到的子句；但由于$r(C)$ 只包含两个X 变量，这类子句最多只会出现在两个不同的 ψ_i 中。最后，不依赖于 $\mathcal{X}$公理的子句，会与前述类型中的某个子句出现在同一 ψ_i 中，因此最多出现在两个不同部分。

□

为了完成证明，我们需要论证每个 ψ_i 都很大。直观上，这是因为 ψ_i 必须为每个$j \in [(i\ell,(i+1)\ell]$ 使用一个 $\mathcal{X}$公理，这会从每个W块中引入一对W 变量；但由于没有子句包含多于 $\ell/8$ 个这样的变量，我们需要使用足够多的 $\mathcal{W}$ 公理来消去上述W 变量。形式上，该结论可由以下两个引理推出。

引理 5.9. 对于每个 $i \in [\ell]$，都存在一个整数 t_i，使得 $s(\psi_i[1,t_i]\restriction_{\sigma_i})$ 是一个归结推导，其子句由 W 变量支撑，且依赖于一个 $\mathcal{X}$ 公理。

证明。令 C_{t_i} 为 ψ_i 中第一个依赖于 $\mathcal{X}$ 公理的子句，并且 $C_{t_i}\restriction_{\sigma_i}$ 由 W 变量支撑。我们证明 t_i 定义良好，$\psi_i[1,t_i]\restriction_{\sigma_i}$ 是一个有效的语义归结推导，并且 $D^{t_i} = s(\psi_i\restriction_{\sigma_i})$ 依赖于 $\mathcal{X}$ 公理。

我们的归纳假设是：对于 $k \le t_i$（或者在 t_i 不存在时，对于任意 k），如果子句 $C_k \in \psi$ 依赖于一个 $\mathcal{X}$ 公理且不被 σ_i 满足，那么存在一个子句 $C_{k'} \in \psi_i$，其 $k' \le k$ 蕴含 C_k（模 σ_i），即 $C_{k'}\restriction_{\sigma_i} \vDash C_k\restriction_{\sigma_i}$，并且依赖于一个 $\mathcal{X}$ 公理（在 ψ 上）。

如果归纳假设成立，那么 t_i 就是良定义的：由于 C_t 不被 σ_i 满足且依赖于一个 $\mathcal{X}$ 公理，因此存在一个子句 $C_{t'} \in \psi_i$，它依赖于一个 $\mathcal{X}$ 公理，并且满足 $C_{t'}\restriction_{\sigma_i} \vDash C_t\restriction_{\sigma_i} = C_t$，且其支撑在 W 变量上。

基例是 C_k 为一个未被满足的 $\mathcal{X}$ 公理，此时我们可以取 $C_{k'} = C_k$。对于归纳情形，令 C_p 和 C_q 为 ψ 中 C_k 的前提。如果恰好有一个前提（设为 C_p）未被满足，并且还依赖于一个 $\mathcal{X}$ 公理，那么根据归纳假设，我们可以取 $C_{k'} = C_{p'}$。否则，我们需要考虑几个子情形。如果归结变量是一个 X 变量，那么两个前提都依赖于一个 $\mathcal{X}$ 公理（根据引理 5.3），因此两个前提都未被满足。由此可知，归结变量未被 σ_i 赋值，因此我们可以取 $C_{k'} = C_k$。

如果归结变量是一个 W 变量，那么由于 σ_i 只对 X 变量赋值，两个前提均不满足。我们分两种情况讨论：如果恰好有一个前提依赖于一个 $\mathcal{X}$ 公理，不妨设为 C_p，那么 $C_{p'}$ 存在于 ψ_i 中；根据 ψ_i 的构造，另一个前提 C_q 存在于 ψ_i 中，当且仅当结论 C_k 存在于其中。如果两个前提都依赖于一个 $\mathcal{X}$ 公理，那么 $C_{p'}$ 和 $C_{q'}$ 都存在于 ψ_i 中。

因此，在后面两种情况下，只需证明 $C_k \in \psi_i$，因为这样我们就可以取 $C_{k'} = C_k$，并且有 $C_k\restriction_{\sigma_i}$ 可由一个有效的语义归结步骤推出。事实上，根据引理 5.7，$C_k\restriction_{\sigma_i}$ 是一个由 W 变量支撑的子句；根据 C_{t_i} 的定义，这意味着 $k = t_i$。然而，由于归结变量是一个 W 变量，$C_{p'}\restriction_{\sigma_i}$ 同样由 W 变量支撑；再结合 $C_{p'}$ 依赖于一个 $\mathcal{X}$ 公理这一事实，这与 C_{t_i} 是第一个此类子句相矛盾。

这就完成了第一个归纳论证，并证明 $\psi[1,t_i]\restriction_{\sigma_i}$ 是一个有效的语义推导；接下来还需证明 D^{t_i} 依赖于关于 $s(\psi_i\restriction_{\sigma_i})$ 的 X 公理。我们通过第二个归纳论证证明：对于每个子句 $D_k \in s(\psi_i[1,t_i]\restriction_{\sigma_i})$，如果 C_k 依赖于一个 $\mathcal{X}$ 公理，那么 D_k 也依赖于一个公理。基例中，D_k 是一个公理，结论显然成立。

对于归纳情形，固定 C_k、$E_k = C_k\restriction_{\sigma_i}$ 和 $D_k = s(E_k)$，并令 $E_p = C_p\restriction_{\sigma_i}$ 和 $E_q = C_q\restriction_{\sigma_i}$ 为 $\psi_i\restriction_{\sigma}$ 中 E_k 的前提。如果 C_p 和 C_q 都依赖于一个 X 公理，那么根据归纳假设，D_p 和 D_q 也都依赖于一个 X 公理，结论得证。我们只需论证一个前提 C_p 依赖于一个 X 公理，而另一个前提 C_q 不依赖于 X 公理的情形。在这种情况下，由于 σ_i 只影响 X 变量，用于推导 C_q 的所有公理都不受 σ_i 影响，因此有 $s(\sigma_i(C_q)) = C_q$，其中包含用于推导 C_k 的归结变量，因此它不蕴含 $s(\sigma_i(C_k))$。根据 $s(\cdot)$ 的构造，$s(\sigma_i(C_k))$ 依赖于 $s(\sigma_i(C_p))$。

□

引理 5.10。设 η 是从 $F_{\ell,1,n}$ 出发的一个归结推导，其中子句 C 由 W 变量支撑，并且依赖于一个 $\mathcal{X}$ 公理。那么 $|\eta| \ge (n-2)(\ell - \mu(C))/2$。

证明。根据引理 5.4，我们可以假设 η 对每个 $j \in [\ell]$ 都至少使用了一个 $A_{j,b}$ 公理。

令 $J = \{j \in [\ell] \mid \exists w_{j,k} \in \mathrm{vars}(C)\}$ 为 C 中提及的 W 块的集合。我们证明，对于每个 $j \in J = [\ell] \setminus J$，至少有 $(n-2)/2$ 个关于 W_j 中变量的公理出现在 η 中，因此至少有 $(n-2)|J|/2 = (n-2)(\ell - \mu(C))/2$ 个公理。

固定 $j \in J$，并为导出矛盾而假设 $(n-2)/2$ 个关于 W_j 中变量的公理出现在 η 中的数量少于该值。于是存在 $k \in [2, n-1]$，使得变量 $w_{j,k}$ 不出现在 η 中。按如下方式重命名变量：对于 $k' < k$，将 $w_{j,k'} \mapsto$ 重命名为 $y_{k'}$；对于 $k' > k$，将 $w_{j,k'} \mapsto$ 重命名为 $y_{k'-n}$。然后，类似于引理 5.3 的证明，我们可以通过归纳证明，从公理 $A_{j,b}$ 推导出的每个子句都具有 $y_{k'} \vee y_{k''} \vee D$ 的形式，其中 D 是由 W_j 外部的文字支撑的。由于这包括 C，因此与我们关于 $j \notin J$ 的假设矛盾。

□

为了完成定理 5.1 的证明，我们只需将各个部分组合起来。

定理的证明 5.1。我们选取公式族 $F_{\ell=48\log n,n,n}$，根据引理 $O(n\log n)$，存在一个长度为 5.2 的归结反驳。

为证明下界，我们假设存在一个长度为 L 的 RMA 反驳 π，且 $L \le n^3 = 2^{16\ell} = o((4/3)^{8\ell})$；否则下界显然成立。我们应用引理 5.5 给出的限制到 π，并使用引理 5.6 得到一个子句的归结推导 ψ，该子句由 W 变量支撑且使用一个公理 $\mathcal{X}$。然后，我们将 ψ 分成 m 个部分 ψ_i，根据引理 5.9 和引理 5.10，每部分的大小至少为 $n\ell/16$。最后，由引理 5.8 可得 $|\pi| \ge |\psi| \ge mn\ell/32 = \Omega(n^2 \log n)$。

□

5.4 结构性推论

定理 5.1 立即给出了 RML 和 RMA 的两个结构性质之一。其一是，引入弱化规则时，证明长度可能会缩短。

推论 5.11. 存在一个公式族，其包含 $O(n\log n)$ 个变量和 $O(n\log n)$ 个子句；这些公式具有长度为 $O(n\log n)$ 的 *RML* 弱化反驳，但每个 *RMA* 反驳都需要长度为$\Omega(n^2 \log n)$。

证明。考虑公式 $F_n \wedge z$，其中 F_n 是定理 5.1 给出的公式，而 z 是一个新变量。若将每个子句 $C \in F_n$ 弱化为 $C \vee z$，那么我们可以在 $O(n \log n)$ 步 RML 推理中推出 $F \vee z \vdash z$，因为每一步推理都是一次合并。然而，如果不能进行弱化，那么 z 就无法与 F_n 中的任何子句进行消解，定理 5.1 的下界也就适用。

□

第二个性质是，RML 和 RMA 并非自然的证明系统（按 [BKS04]的定义），因为限制可能会导致证明长度增加。

推论 5.12。存在一个限制 ρ 以及一个公式族，其包含 $O(n\log n)$ 个变量和 $O(n \log n)$ 个子句；这些公式具有长度为 $O(n\log n)$ 的 *RML* 反驳，但每个针对 $F_n\restriction_\rho$的 *RMA* 反驳都要求长度为 $\Omega(n^2 \log n)$。

证明。考虑公式 $G_n = (F_n \vee z) \wedge z$，其中 F_n 是定理 5.1 给出的公式，$F \vee z = \{C \vee z \mid C \in F\}$，且 z 是一个新变量。与推论 5.11 的证明类似，存在一个长度为 $O(n \log n)$ 步的 z 的 RML 推导，而 $G_n\restriction_\rho = F_n$。

□

6 进一步的分离

我们可以通过对$F_{\ell、m、n}$ 进行几种变体，分离我们引入的不同类型的合并归结；在每个$i \in [\ell]$中，我们都添加常数个冗余子句。我们将这些不同的子句视为 $\mathcal{W}$的一部分。

上界的证明遵循相同的模式。我们首先逐一说明如何将$w_1 w_n$ 和 $w_1 w_n$ 作为引理得到，然后按照第 5.2 节中的方法继续。

为了证明下界，我们将推广定理 5.1 证明中的下界部分，使其也适用于这些变体。幸运的是，我们只需要进行少量局部修改。F

首先，我们需要证明引理 5.3 的一个等价形式，这一点我们将逐一分类证明。s.

其次，我们需要说明k-respecting 限制可以扩展到新变量。对于每个块J_i，由于新子句被$w_{i,1} = w_{i,2}$语义蕴含，因此存在一种将新变量映射到$w_{i,1}$ 和$w_{i,2}$ 中的方式，使得限制的结果与我们一开始就使用子句$w_{i,1} \vee w_{i,2}$ 和$w_{i,1} \vee w_{i,2}$时相同，而这些子句已经是 $\mathcal{W}_i$的一部分。也就是说，经过引理5.6之后，我们处理的公式是一个未被修改的$F_{\ell',m',n'}$ 公式的副本。

下界中唯一依赖于归结的特定子系统的部分是引理 5.6；之后我们使用的所有信息，都是没有重复使用以 W 变量为支撑的子句。此外，我们在引理 5.6 的证明中所用到的该子系统的唯一性质，是引理 5.3 适用。因此，我们刚才概述的修改足以使下界结论成立。

6.1 RMA 与 LRMA 的分离

命题 6.1。 存在一个公式族，其包含 $O(n\log n)$ 个变量和 $O(n\log n)$ 个子句，并且具有长度为 $O(n\log n)$ 的 *RMA* 反驳；但每个 *LRMA* 反驳都需要长度为 $\Omega(n^2\log n)$。

分离公式是 $F^{(1)}{}_{m,n,\ell}$，其中我们向 $F_{m,n,\ell}$ 子句中添加了子句

$$\overline{w_{i,1}} \vee w_{i,2} \vee \overline{z_i}, \tag{C1}$$
$$\overline{w_{i,2}} \vee z_i, \tag{C2}$$
$$w_{i,1} \vee \overline{w_{i,2}} \vee \overline{y_i}, \tag{C3}$$
$$w_{i,2} \vee y_i, \tag{C4}$$

对于每个 $i \in [\ell]$。新变量可以赋值为 $z_i = w_{i,1}$ 和 $y_i = w_{i,1}$，从而还原原公式。

上界由以下引理得出。

引理 6.2. 子句 $w_{i,1} \vee w_{i,n}$ 和 $\overline{w_{i,1}} \vee \overline{w_{i,n}}$ 可以在 *RMA* 中从 $F^{(1)}{}_{m,n,\ell}$ 推导为引理，长度为 $O(n)$。

证明。 我们先用 (C2) 消解子句 $w_{i,1} \vee w_{i,2}$，再用 (C1) 消解，以获得 $w_{i,1} \vee w_{i,2}$ 作为合并，然后推导出 $w_{i,1} \vee w_{i,n}$；它以一个合并作为祖先，因此可以被记住。类似地，从 $w_{i,1} \vee w_{i,2}$、(C3) 和 (C4) 出发，我们可以得到 $w_{i,1} \vee \overline{w_{i,n}}$ 作为一个引理。 □

以下观察对证明下界很有用。

引理 6.3. 设 C 和 D 是含有两对相反文字的子句。那么，C 和 D 不可能同时出现在同一个局部正则输入推导中。

证明。 设 $C = x \vee y \vee C'$ 和 $D = \overline{x} \vee \overline{y} \vee D'$。不失一般性，假设 C 是推导中先于 C 和 D出现的子句。如果在 D之前将 x 或 y 用作归结变量，那么局部正则条件会阻止将 D 用作公理。否则，自从使用 C 以来，推导中就会出现 $x \vee y$，这同样会阻止使用 D。

□

引理 5.3 的对应结论如下。

引理 6.4. 设 η 是一个仅依赖于 $\mathcal{W}$ 公理的 *LRMA* 推导。那么，η 中的子句都不能重复使用。

证明。 我们只能通过 (C1) 或 (C3) 之一得到合并。不失一般性，假设 (C1) 是推导中最先使用的这两者之一。根据引理 6.3，(C2) 和 (C3) 都不会出现在推导中。我们可以通过归纳证明，只能得到如下形式的子句：$w_{i,j} \vee w_{i,j'} \vee z_i$ 或 $y_i \vee w_{i,j} \vee z_i$，绝不会以合并的方式得到。

□

6.2 RML/LRMA 与 LRML 之间的分离

命题 6.5. 存在一个公式族，其包含 $O(n\log n)$ 个变量和 $O(n\log n)$ 个子句，并且具有长度为 $O(n\log n)$ 的 *RML* 和 *LRMA* 反驳；但是，每个 *LRML* 反驳都需要长度为$\Omega(n^2\log n)$。

分离公式为$F^{(2)}{}_{m,n,\ell}$，其中我们为$F_{m,n,\ell}$ 添加子句

$$z_i \vee \overline{w_{i,1}} \vee w_{i,2}, \tag{C1}$$
$$\overline{z_i} \vee \overline{w_{i,1}} \vee w_{i,2}, \tag{C2}$$
$$y_i \vee w_{i,1} \vee \overline{w_{i,2}}, \tag{C3}$$
$$\overline{y_i} \vee w_{i,1} \vee \overline{w_{i,2}}, \tag{C4}$$

对每个$i \in [\ell]$均如此。新变量可赋值为$z_i = 1$和$y_i = 1$，从而还原原公式。

这些上界分别由以下引理得出。

引理 6.6. 子句 $w_{i,1} \vee w_{i,n}$ 和 $\overline{w_{i,1}} \vee \overline{w_{i,n}}$ 可以在 *RML* 中以长度 $O(n)$ 从 $F^{(2)}_{m,n,\ell}$ 推导为引理。

证明。我们首先消解子句 $w_{n-1} \vee \overline{w_{i,n}}$, $w_{i,n-2} \vee \overline{w_{i,n-1}}, \ldots, w_{i,2} \vee \overline{w_{i,3}}$ (C1)，得到 $z_i \vee w_{i,1} \vee w_{i,n}$。接着，我们通过输入推导与 (C2) 进行消解，得到 $w_{i,1} \vee w_{i,2} \vee w_{i,n}$。然后，我们与 $w_{i,2} \vee w_{i,3}, w_{i,3} \vee w_{i,4}, \ldots, w_{i,n-1} \vee w_{i,n}$ 进行消解，得到 $w_{i,1} \vee \overline{w_{i,n}}$，这是对 $w_{i,n}$ 的否定。类似地，我们可以得到 $w_{i,1} \vee w_{i,n}$。 □

引理 6.7。 子句 $w_{i,1} \vee w_{i,n}$ 和 $w_{i,1} \vee \overline{w_{i,n}}$ 可以作为引理从 $F^{(2)}_{m,n,\ell}$ 中推导出来，其在 *LRMA* 中的长度为 $O(n)$。

证明。 我们对 (C1) 和 (C2) 中的子句进行消解，得到 $w_{i,1} \vee w_{i,2}$，这是一个合并，然后推导出 $w_{i,1} \vee w_{i,n}$，其祖先中有一个合并，因此可以将其用作引理。类似地，从 (C3) 和 (C4) 开始，我们可以得到 $w_{i,1} \vee w_{i,n}$ 作为引理。 □

引理 5.3 的等价表述如下。

引理 6.8。 设 η 是一个仅依赖于 $\mathcal{W}$ 公理的 *LRML* 推导。则 η 中的任何子句都不能被重复使用。

其证明思路是，我们能够得到的唯一合并涉及z_i或y_i变量。如果我们仅针对这样的变量对两个子句进行消解，就会得到一个已经存在的子句，因此这是无效的。否则，我们会将 w_2 之一消解掉，而在将z_i消解掉时，该变量又会被重新引入，而 SR条件不允许这样做。

证明。 我们只能通过使用某个新子句 (C1)–(C4) 得到合并。如果对y_i或z_i上的任一对子句进行消解，得到的子句已经存在于该公式中，因此可以通过预处理消去这类推导。

否则，考虑推导中首次将某个新子句用作前提的步骤，不失一般性，设该子句为 (C1)。这一步与一个形如$w_{i,2} \vee w_{i,j}$的子句进行消解，得到一个形如$z_i \vee w_{i,1} \vee w_{i,j}$的子句，这不是一个合并。该子句可能会在$w_{i,j}$ $(j > 2)$上进一步消解，从而得到其他同类子句；这些子句也都不是合并。但它不能在y_i、z_i或$w_{i,1}$ 上进行消解，因为这一步会重新引入变量$w_{i,2}$。 □

6.3 LRML 与 REL 之间的分离

命题 6.9。 存在一个含$O(n \log n)$ 个变量和$O(n \log n)$ 个子句的公式族，它们具有长度为$O(n \log n)$ 的 *LRML* 反驳，但每个 *REL* 反驳都需要长度为 $\Omega(n^2 \log n)$。

分离公式为$F^{(3)}_{m,n,\ell}$，其中我们向$F_{m,n,\ell}$ 中添加子句

$$\overline{w_{i,1}} \vee \overline{w_{i,2}} \vee w_{i,3}, \tag{C1}$$
$$w_{i,1} \vee w_{i,2} \vee \overline{w_{i,3}} \tag{C2}$$

对每个$i \in [\ell]$。如果我们赋值$w_{i,2} = w_{i,1}$，就会得到$F_{m,n-1,\ell}$ 的一个副本。尽管严格来说它并不是我们最初那个公式，但这已足以满足我们的目的。

上界由以下引理得出。

引理 6.10. 子句 $w_{i,1} \vee w_{i,n}$ 和 $\overline{w_{i,1}} \vee \overline{w_{i,n}}$ 可以在长度$O(n)$内，作为引理从 $F^{(3)}_{m,n,\ell}$ 中通过 *LRML* 推导出来。

证明。我们将 (C1) 与 $w_{i,3} \vee w_{i,4}, \ldots, w_{i,n-1} \vee w_{i,n}$ 消解，得到 $w_{i,1} \vee w_{i,2} \vee w_{i,n}$，然后再与 $w_{i,1} \vee w_{i,2}$ 消解，以合并的形式得到 $w_{i,1} \vee w_{i,n}$。类似地，从 (C2) 开始，我们可以得到 $w_{i,1} \vee w_{i,n}$作为一个引理。□

引理 5.3 的等价表述如下。

引理 6.11. 设 η 是一个仅依赖于 $\mathcal{W}$ 公理的 *REL* 推导。那么，η 中没有任何子句可以被重复使用。

证明。注意，每个可推导的子句的宽度至少为 2。设 C 是任意一个可推导的子句，ℓ 是 C 中的任意一个文字。我们有 $\alpha = C \setminus \ell$ 非空。然而，给任意变量 $w_{i,j}$赋值会立即传播所有变量，因此 ℓ 不具赋能性。□

7 结论

本文探讨了 CDCL 求解器对归结证明的模拟是否紧的问题。具体而言，我们证明，RMA 以及其他 DAG 型合并归结形式，都能以至多线性的乘法开销模拟标准归结。然而，与树状情形不同，这种开销是不可避免的。我们引入的证明系统有助于解释 CDCL 模拟归结时开销的一个来源，但目前尚不清楚它们能否精确刻画这一开销。换言之，一个有趣的未来研究方向是探究 CDCL 能否以低于模拟标准归结所需开销的代价，模拟某种形式的合并归结。

致谢

作者感谢 Yuval Filmus 以及参加西蒙斯计算理论研究所项目 可满足性：理论、实践及更多的众多参与者，感谢他们进行了大量讨论。本工作部分是在作者访问西蒙斯计算理论研究所期间完成的。

树状合并归结

为完整起见，我们非正式地概述引理 3.2 和定理 3.3的证明；完整细节见 [以及68]。

引理 A.1 （引理 3.2，重述）。 如果存在一个 C 的树状归结推导，其长度为 L，且至多只有根节点是合并节点，那么存在某个 $C' \subseteq C$ 的输入归结推导，其长度至多为 L。

证明（概述）。我们通过对 $|\eta|$ 进行归纳来证明：对于每个公理 $E \in \eta$，存在一个 C' 的输入推导，该推导使用 η 中的一部分公理，其中E 是最上层公理。作为中间对象，我们允许该推导中的子句包含相反文字；稍后再将其清除。

令 $C = \mathrm{Res}(A \vee x, B \vee \bar{x})$，并令 η_1 和 η_2 为用于推导出 $A \vee x$ 和 $B \vee \bar{x}$ 的推导。不失一般性，假设 $E \in \eta_1$。由于 η_2 不包含任何合并，因此从 $B \vee \bar{x}$ 到某个公理 $D \vee \bar{x} \in \eta_2$存在唯一一条路径，其中所有子句都包含 $\bar{x}$。注意，η_2中的其他子句仍可能包含 $\bar{x}$ 或 x。我们在 η_2 中将 $D \vee \bar{x}$ 替换为 D（从而删除上述路径中所有出现的 $\bar{x}$），得到一个有效的 B 推导 η_3。我们对 η_1 和 η_3 应用归纳假设，得到两个单位推导 η_4 和 η_5，其顶层叶子分别为 E 和 D，且分别推导出 $A' \vee x \subseteq A \vee x$ 和 $B' \subseteq B$。我们在 η_5 中将 D 替换为 $D \vee A'$，得到一个 $B' \vee A'' \subseteq B' \vee A'$的单位推导 η_6。通过注意到 $\mathrm{Res}(A' \vee x, D \vee \bar{x}) = A' \vee D$，我们将 η_4 和 η_6 拼接起来，

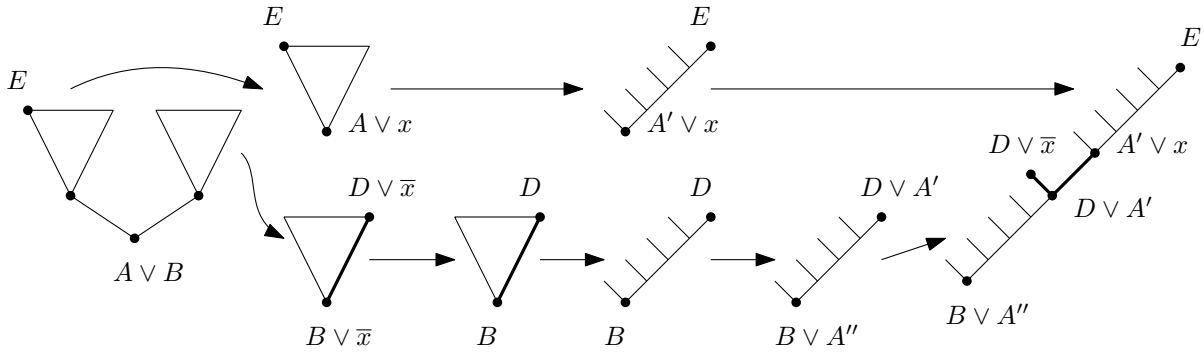

图 2：引理证明A.1

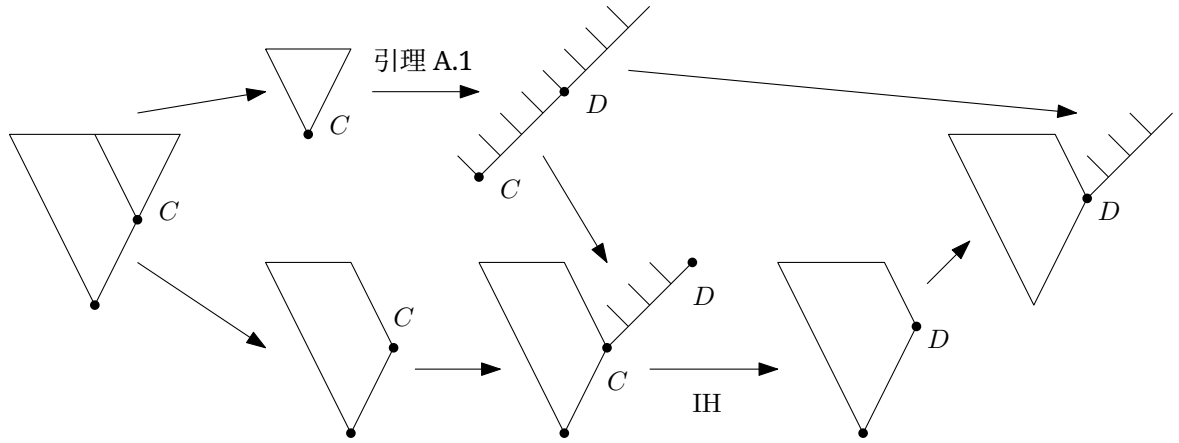

图 3：定理证明 A.2

这是 η_6 中唯一不在原始公理中的公理，并获得一个单位推导 η_7，其$B \vee A'' = C' \subseteq C$ 仅使用原始公理。

最后，在归纳论证之外，我们通过将任何包含相反文字的子句替换为 1，来去除这类子句，从而得到一个语义推导 η_8。其句法对应物$s(\eta_8)$满足引理的结论。

□

定理 A.2 （定理3.3，重述）。 如果存在一个长度为 C 的树状归结推导 L，那么就存在一个关于某个 $C' \subseteq C$ 的长度至多为 L的合并分辨率推导。

证明（概要）。证明采用关于合并次数的归纳法。当不存在合并时的基例由引理 A.1 得出。否则，设 ψ 是一个子树，其中恰好根节点 C 是一次合并。设ψ' 是由引理 A.1 给出的 C′ 的输入分辨率推导，设 D 是 ψ'中的最后一次合并，并设ω 和 ω' 分别是从 D 到 C′ 以及到 D 的 ψ' 片段。我们在 η 中用 ω 替换 ψ，从而得到一个以 D 为公理的反驳 η'（注意，在用 C′ 替换 C 时，我们可能需要剪除 η的部分内容）。由于 η' 少了一次合并，我们可以应用归纳假设，并得到一个合并分辨率推导 ψ''。最后，我们将公理 D 替换为推导 ω'。

□

参考文献

[ABH+08] Gilles Audemard、Lucas Bordeaux、Youssef Hamadi、Saïd Jabbour 和 Lakhdar Sais。《冲突分析的通用框架》。载于 Hans Kleine Büning 和 Xishun Zhao 编，《style id='3'>可满足性测试理论与应用 - SAT 2008，第11届国际

会议，SAT 2008，中国广州，2008年5月12日至15日。会议论文集，《计算机科学讲义》第4996卷，第21–27页。Springer，2008年。doi:10.1007/ 978-3-540-79719-7_3.[AFT11] Albert Atserias、Johannes Klaus Fichte 和 Marc Thurley。具有多次重启和有界宽度归结的子句学习算法。《人工智能研究杂志》，40:353–373，2011年1月。初步版本发表于 SAT '09。[And68] Peter B. Andrews。带合并的归结。《ACM学报》，15(3):367–381，1968年。[BB21] Olaf Beyersdorff 和 Benjamin Böhm。理解 QBF CDCL 求解器与 QBF 归结的相对强度。载于 James R. Lee 编，《第12届理论计算机科学创新会议》，ITCS 2021，2021年1月6日至8日，虚拟会议，《LIPIcs》第185卷，第12:1–12:20页。Schloss Dagstuhl - Leibniz-Zentrum für Informatik，2021年。doi:10.4230/LIPIcs.ITCS.2021.12。[BF97] Avrim L Blum 和 Merrick L Furst。通过规划图分析进行快速规划。《人工智能》，90(1-2):281–300，1997年。[BHJ08] Samuel R. Buss、Jan Hoffmann 和 Jan Johannsen。带引理的归结树：刻画具有子句学习功能的 DLL 算法的归结改进。《计算机科学中的逻辑方法》，4(4:13)，2008年12月。[BKS04] Paul Beame、Henry Kautz 和 Ashish Sabharwal。致力于理解和利用子句学习的潜力。《人工智能研究杂志》，22:319–351，2004年12月。初步版本发表于 IJCAI '03。[BN21] Sam Buss 和 Jakob Nordström。证明复杂性与 SAT 求解。载于 Armin Biere、Marijn Heule、Hans van Maaren 和 Toby Walsh 编，《可满足性手册》，《人工智能与应用前沿》第336卷，第7章，第233–350页。IOS Press，第2版，2021年。doi:10.3233/FAIA200990。[BS97] Roberto J. Bayardo Jr. 和 Robert Schrag。使用 CSP 回溯技术求解现实世界中的 SAT 实例。载于《第14届全国人工智能会议论文集》（AAAI '97），第203–208页，1997年7月。[CGP+08] Cristian Cadar、Vijay Ganesh、Peter M Pawlowski、David L Dill 和 Dawson R Engler。EXE：自动生成死亡输入。《ACM信息与系统安全汇刊》（TISSEC），12(2):1–38，2008年。[DHN07] Nachum Dershowitz、Ziyad Hanna 和 Alexander Nadel。探讨冲突驱动 SAT 求解器功能的更好理解。载于 João Marques-Silva 和 Karem A. Sakallah 编，《可满足性测试理论与应用 - SAT 2007，第10届国际会议》，葡萄牙里斯本，2007年5月28日至31日，会议论文集，《计算机科学讲义》第4501卷，第287–293页。Springer，2007年。doi:10.1007/978-3-540-72788-0_27.[dV94] Alvaro del Val。易处理数据库：如何通过编译使命题单元归结完备。载于 Jon Doyle、Erik Sandewall 和 Pietro Torasso 编，《第4届知识表示与推理原理国际会议论文集》（KR'94）。德国波恩，1994年5月24日至27日，第551–561页。Morgan Kaufmann，1994年。

[DVT07] Julian Dolby、Mandana Vaziri 和 Frank Tip。使用 SAT 求解器高效查找错误。载于《欧洲软件工程会议》和 ACM SIGSOFT 软件工程基础国际研讨会第 6 次联合会议论文集，第 195–204 页，2007 年。doi:10.1145/1287624.1287653。[FB20] Nick Feng 和 Fahiem Bacchus。采用全 UIP 学习进行子句大小缩减。载于 Luca Pulina 和 Martina Seidl（编），《可满足性测试理论与应用 - SAT 2020 - 第 23 届国际会议》，意大利阿尔盖罗，2020 年 7 月 3–10 日，《会议论文集》，《计算机科学讲义》第 12178 卷，第 28–45 页。Springer，2020 年。doi:10.1007/978-3-030-51825-7_3。[HBPV08] Philipp Hertel、Fahiem Bacchus、Toniann Pitassi 和 Allen Van Gelder。子句学习能够有效地 P-模拟一般命题归结。载于第 23 届全国人工智能会议（AAAI '08）论文集，第 283–290 页，2008 年 7 月。[LFV+20] Chunxiao Li、Noah Fleming、Marc Vinyals、Toniann Pitassi 和 Vijay Ganesh。迈向对 SAT 求解器中重启的复杂性理论理解。载于《可满足性测试理论与应用–SAT 2020：第 23 届国际会议》，意大利阿尔盖罗，2020 年 7 月 3–10 日，《会议论文集》23，第 233–249 页。Springer，2020 年。[MLM21] João Marques-Silva、Inês Lynce 和 Sharad Malik。冲突驱动子句学习 SAT 求解器。载于 Armin Biere、Marijn Heule、Hans van Maaren 和 Toby Walsh（编），《可满足性手册 - 第二版》，《人工智能与应用前沿》第 336 卷，第 133–182 页。IOS Press，2021 年。doi:10.3233/FAIA200987。[MMZ+01] Matthew W. Moskewicz、Conor F. Madigan、Ying Zhao、Lintao Zhang 和 Sharad Malik。Chaff：设计高效的 SAT 求解器。载于第 38 届设计自动化会议（DAC '01）论文集，第 530–535 页，2001 年 6 月。[MPR20] Nathan Mull、Shuo Pang 和 Alexander A. Razborov。论采用有序决策策略的基于 CDCL 的证明系统。载于第 23 届可满足性测试理论与应用国际会议（SAT '20）论文集，《计算机科学讲义》第 12178 卷，第 149–165 页。Springer，2020 年 7 月。[MS85] Burkhard Monien 和 Ewald Speckenmeyer。在少于 2^n 步的时间内求解可满足性问题。《离散应用数学》，10(3)：287–295，1985 年。doi:10.1016/0166-218X(85)90050-2。[MS99] João P. Marques-Silva 和 Karem A. Sakallah。GRASP：用于命题可满足性的搜索算法。《IEEE 计算机学报》，48(5)：506–521，1999 年 5 月。初步版本发表于 ICCAD '96。[PD08] Knot Pipatsrisawat 和 Adnan Darwiche。用于高效不可满足性证明的新子句学习方案。载于 Dieter Fox 和 Carla P. Gomes（编），《第二十三届 AAAI 人工智能会议论文集》，AAAI 2008，美国伊利诺伊州芝加哥，2008 年 7 月 13–17 日，第 1481–1484 页。AAAI Press，2008 年。URL：http://www.aaai.org/Library/AAAI/2008/aaai08-243.php。[PD11] Knot Pipatsrisawat 和 Adnan Darwiche。论子句学习 SAT 求解器作为归结引擎的能力。《人工智能》，175(2)：512–525，2011 年 2 月。初步版本发表于 CP '09。[Rya04] Lawrence Ryan。子句学习 SAT 求解器的高效算法。硕士论文，西蒙弗雷泽大学，2004 年。

[SB09] Niklas Sörensson 和 Armin Biere。学习子句的最小化。载于第 12 届可满足性测试理论与应用国际会议（SAT '09）会议论文集，《计算机科学讲义》第 5584 卷，第 237–243 页。Springer，2009 年 7 月。[Van05] Allen Van Gelder。池归结及其与正规归结和带子句学习的 DPLL 算法之间的关系。载于第 12 届逻辑编程、人工智能与推理国际会议（LPAR '05）会议论文集，《计算机科学讲义》第 3835 卷，第 580–594 页。Springer，2005 年。[Vin20] Marc Vinyals。常见变量决策启发式方法的困难示例。载于第 34 届 AAAI 人工智能会议（AAAI '20）会议论文集，第 1652–1659 页，2020 年 2 月。[XA05] Yichen Xie 和 Alexander Aiken。Saturn：一种基于 SAT 的错误检测工具。载于第 17 届计算机辅助验证国际会议（CAV 2005）会议论文集，第 139–143 页，2005 年。doi:10.1007/11513988_13.[ZMMM01] Lintao Zhang、Conor F. Madigan、Matthew W. Moskewicz 和 Sharad Malik。布尔可满足性求解器中的高效冲突驱动学习。载于 IEEE/ACM 计算机辅助设计国际会议（ICCAD '01）会议论文集，第 279–285 页，2001 年 11 月。[ZMW+18] Edward Zulkoski、Ruben Martins、Christoph M. Wintersteiger、Jia Hui Liang、Krzysztof Czarnecki 和 Vijay Ganesh。结构度量与合并对 SAT 求解器性能的影响。载于 John N. Hooker 编，第 24 届约束编程原理与实践国际会议（CP 2018）会议论文集，法国里尔，2018 年 8 月 27–31 日，《计算机科学讲义》第 11008 卷，第 436–452 页。Springer，2018 年。doi:10.1007/978-3-319-98334-9_29。