收敛门消除与构造性电路下界

Marco Carmosino MIT-IBM
Watson AI Lab，美国马萨诸塞州剑桥
Ngu Dang 波士顿大学，美国马萨诸塞州波士顿
Tim Jackman 波士顿大学，美国马萨诸塞州波士顿

arXiv:2602.17942v1 [cs.CC] 20 Feb 2026

摘要

门消除是目前唯一已知能够针对不受限制的布尔电路证明显式函数复杂性下界的方法。经过数十年的努力，最好的结果仍然是线性的：Li 和 Yang 在由所有二元布尔函数组成的基 $\mathcal{B}_2$ 中证明了 $3.1n - o(1)$ 的下界（STOC 2022）；Iwama 和 Morizumi 则在仅由与、或和非函数组成的德摩根基中证明了 $5n - o(1)$ 的下界（MFCS 2002）。这些论证采用复杂且冗长的分类讨论，将精心挑选的输入代入电路，并追踪哪些门因此变得冗余——这就是“门消除”。与复杂性理论中的其他方法不同，对于通过门消除证明电路下界所面临的改进障碍，目前只有有限的刻画。

为更好地理解门消除，本工作做出了以下贡献：（1）将电路化简形式化为一个合流项图重写系统；（2）使用该系统给出一个简单且具有构造性的经典下界证明。

首先，我们证明，电路简化在德摩根基和$\{\wedge, \vee, \oplus\}$基上是一个合流项图重写系统。我们根据布尔恒等式定义局部重写规则，使得每个化简序列都会得到相同的最终结果（在电路同构或互模拟意义下）。收敛性使我们能够严谨地推理简化电路的结构性质，而不受化简顺序的影响。接着，我们证明，在$\mathcal{U}_2$ 和$\mathcal{B}_2$ 基上，不存在类似的电路简化合流形式化方法。

然后，我们使用自己的化简系统给出一个构造性电路下界，将 Schnorr 的经典结果推广到异或函数：在德摩根基中，计算该函数需要 $3(n-1)$ 个门。构造性下界 $f \notin \mathcal{C}$ 给出一种算法（称为“反驳算法”），能够高效地为每个试图计算函数 f的$\mathcal{C}$电路找出反例。Chen、Jin、Santhanam 和 Williams 指出，构造性在复杂性理论的许多长期未决问题中发挥着核心作用（FOCS 2021），因此，自然会考虑通过门消除论证来获得构造性电路下界。这表明，利用合流化简可以得到更简洁、模块化程度更高的电路下界证明。此外，在本工作之前，还没有通过门消除证明过构造性下界。

2012 ACM 学科分类 计算理论 → 重写系统；计算理论→ 电路复杂性

关键词与短语 项图重写、电路复杂性、构造性

数字对象唯一标识符 10.4230/LIPIcs.CVIT.2016.23

补充材料

© Marco Carmosino、Ngu Dang 和 Tim Jackman；依据知识共享署名 4.0 国际许可协议授权 第42届重要主题会议（CVIT 2016）。编辑：John Q. Open 和 Joan R. Access；第23号文章；页码 23:1–23:24 莱布尼茨国际信息学会议录 Schloss Dagstuhl – 莱布尼茨信息学中心，达格施图尔出版社，德国

LIPICS

1 引言

电路通过原子处理单元——逻辑“门”之间的无环导线，对固定输入长度上的布尔函数计算进行建模。为了衡量函数 f 的电路复杂性，我们固定一组门 $\mathcal{B}$，称为 基，并统计计算 f 所需的二元门数量。一些知名的基包括德摩根基（二元与门、二元或门和一元非门）、$\mathcal{U}_2$ 基（除奇偶函数和等价函数外的所有二元布尔函数），以及 $\mathcal{B}_2$ 基（所有二元布尔函数）。

香农在 1949 年——使用一个简单的计数论证——证明了，对于大多数 n 输入布尔函数而言，计算它们需要接近最大值的电路复杂性（$2^n/n$ 个门） [39]。对于任意 n 输入布尔函数，通过使用 $O(n2^n)$ 个门写出一个析取范式，都可以得到几乎相同的、微不足道的“记忆”复杂性。不幸的是，计数论证只能证明这类困难函数的存在。识别出具有高电路复杂性的具体布尔函数，是一个重大的未解决问题，并且与通过广受瞩目的困难性-随机性权衡： 持续寻找高效伪随机比特生成器密不可分：足够强的电路复杂性下界意味着 P = BPP [28, 17, 19]。过去七十年间，人们已经为限制为常数深度的电路 [11, 41, 51]证明了令人瞩目的超多项式复杂性下界。然而，高效的通用去随机化还需要针对无约束电路的复杂性下界，而这仍是一项重大的挑战。

门消除是目前已知用于证明 DeMorgan、$\mathcal{U}_2$[34, 38, 52, 23, 18] 和$\mathcal{B}_2$ 基上的无约束电路针对特定布尔函数的复杂性下界最成功的技术。其思路很简单：替换输入变量（用常量或其他函数替换），然后化简所得电路。化简过程中，会跟踪门的数量等复杂度度量以及关于电路结构的断言，并通过归纳法将其提升为下界。例如，如果限制操作得到一个理想的子函数（如在更少变量上计算同一函数），同时消去 c 个门，那么我们就可以通过归纳法论证，原电路至少有 $c\cdot n$ 个门。最简单的门消除论证严格遵循这一模式，根据基本布尔恒等式替换常量并消去门。Schnorr 关于异或$_n$ （以及 ¬异或$_n$） [38] 的紧致 3($n-1$) 下界很好地体现了这一点：将一个输入替换为常量，就能至少消去三个门，并得到一个计算异或$_{n-1}$（或 ¬异或$_{n-1}$）的电路。归纳可知，任何异或$_n$ 电路都至少需要 3($n-1$) 个门。

然而，研究进展缓慢，目前已知的最佳下界仍然是线性的：对于德摩根和$\mathcal{U}_2$ 基，大约为$5n$；对于$\mathcal{B}_2$ 基，则为 $3.1n$ [26]。为什么会这样？持续改进这些结果，并理解门消除的能力与局限性，是一个重要的研究方向。最近的一项令人振奋的进展中，突破性的 ETH 困难性结果利用门消除刻画并利用了线性规模电路的结构 [16]。结合近期关于构造性分离（能够生成输入、使错误算法出错的高效算法，定义见下文）对于广泛猜想的复杂性下界 [7],是必要的新结果，这引出了自然的问题：通过门消除得到的证明是否具有构造性？如果是，这究竟是障碍还是优势？

上述问题促成了本项工作。尽管门消除发挥着基础性作用，但我们对这一方法的理解仍停留在非正式层面，且所有已知论证都属于临时性的。我们提出两项贡献：

1. 将所有门消除论证中使用的电路简化过程形式化为一个合流项图重写系统。**2.** 通过构造性分离，将异或$_n$ 与规模为 $<3(n-1)$ 的电路分离；这是通过使用我们的合流电路简化系统重写 Schnorr 的门消除证明而得出的。

1.1 我们的结果与研究方向

用于门消除的合流系统。 在第 3 节中，我们运用项重写 [2] 和基于图的计算模型 [32]的思想，为门消除建立一个作为形式化基础的合流**项图重写系统**。第 3.1 节和第 3.2 节定义了基于德摩根基和{∧、 ¬、 ⊕} 基的布尔公式化简系统，其规则源自精心挑选的布尔恒等式。随后，应用 Knuth-Bendix 算法[20, 15] 生成合流的公式化简系统。在第 3.3 节中，这些公式重写系统通过 Plump 的项图重写框架 [32]提升为电路重写系统。所得系统是合流的，也就是说，任何有效的化简序列都会终止于唯一的范式，而与化简顺序无关。

▶ **定理** （定理 13 的非正式陈述）。 在德摩根（*{*∧， ∨， ¬*}*）基和 *{*∧， ⊕， ¬*}* 基中，电路简化是收敛的。

这种收敛性质有助于使门消除论证既简洁又可自动化：由于每一条化简序列最终都会到达同一个范式，简化后电路的结构性质在重写选择下保持不变，因此我们可以对“范式意义下”的电路进行推理，而不必为每一种可能的化简路径分别提供论证。这对于计算机辅助证明尤为重要——不终止的重写系统甚至可能无法停止，而收敛的重写则保证补全，并使穷举探索变得没有必要。我们希望，将门消除置于收敛的项图重写框架中，有助于推动复杂性理论的持续形式化（参见第 1.2 节“相关工作”）。

▶**研究方向 1.** 在标准证明助手中形式化已知的门消除论证，以开发新的计算机辅助门消除论证，并最终改进电路下界。

为了推进这一目标，我们使用该系统（附录 C）以结构化的形式重新证明了异或$_n$ [38] 的施诺尔下界，旨在简化形式化过程（附录 C.1）。下一步自然是使用 Lean4 这样的证明助手验证我们系统的收敛性以及施诺尔的论证。

非合流门消除。 我们在第4节探究了对系统进行各种调整，结果令人惊讶。经典电路复杂性论证将德摩根基和$\mathcal{U}_2$ 视为可互换。这样的处理并非毫无依据：就规模与深度界而言，对于所有非退化函数，这两个基都相同，除了 $f(x) = \neg x$。这一等价关系在$\{\neg, \wedge, \oplus\}$ 与$\mathcal{B}_2$ 之间也成立； [30] 之类的论证会先将$\mathcal{B}_2$ 门归类为“与”类和“异或”类，从而非正式地将基改为$\{\neg, \wedge, \oplus\}$。我们在附录 B 明确阐述了这些基之间的民间已知联系。然而，我们发现，将门消除形式化为重写系统时，这些基之间的等价关系会立即失效。

▶ **观察** （观察16的非正式陈述）。 U_2 与B_2 不容许合流电路化简系统。

这一失败源于多余否定门的存在：这些否定门在化简过程中需要被向上和/或向下归并，从而导致发散的化简路径，并最终产生非同构电路（见第4节中的图 2）。

即便基不同，通常也会以类似的方式处理，因为翻译往往只会使规模和深度发生常数级变化。研究者关注的是证明非线性下界，

因此，这种代价看似无关紧要。这一结果凸显了基的选择可能产生的重大影响；在德摩根基或$\{\wedge, \oplus, \neg\}$中进行论证，可能比在更容易 其$\mathcal{U}_2$或$\mathcal{B}_2$对应的基中更容易。若借助证明助手实现，这一点尤为明显——根据具体论证，在这些基中形式化可能缩小搜索空间。

将门消除表述为一个重写系统后，我们对其底层性质有了更深入的理解。除了数十年来进展有限以及已知的局限性（见“相关工作”）之外，利用门消除证明超线性电路界所面临的可证明障碍仍然未知。

▶**研究方向 2**。 进一步研究作为重写系统的门消除。改变基和规则会如何影响其能力？这种技术是否存在无法用于证明电路下界的障碍？

构造性电路下界。 陈述“布尔函数 f 不具有规模为 $s(n)$的电路”翻译成逻辑后具有量词结构 $\forall\exists$:

$$\forall n \in \mathbb{N} \ \ \forall C \text{ such that } \mathsf{size}(C) < s(n) \ \ \exists x \in \{0,1\}^n \ \ C(x) \neq f(x).$$

具体而言，每个无法计算 f 的电路都必定会在某个特定输入 x上出错。因此，针对特定函数 f 的每个电路复杂性下界都会自然地产生一个全搜索问题：给定一个规模过小的电路 C（因此必定 无法计算f），输出一个位串 x，使得 C 与 f 不一致。近期令人振奋的研究发现，构造性分离在复杂性理论中发挥着出人意料的核心作用，因为研究表明，大多数猜想中的一致分离（包括 P $\neq$ NP、ZPP $\neq$ EXP 和 BPP $\neq$ NEXP）都可以自动转换为构造性分离 [7]。

另一方面，他们还证明，如果某些众所周知的下界（包括“在单带图灵机上判定回文串需要超二次时间”， [27]）能够被构造化，那么重大的复杂性突破（例如 P $\neq$NP）将随之而来[7]。从本质上说，认为复杂性下界的“简单”证明应该“容易”实现构造化，这一直觉是极其不准确的。我们证明，至少在门消除的情况下，简单证明可以转化为构造性分离。

▶**定理**（定理 19 的非正式表述）**：** 设 C 是一个作用于 n 个输入、所含二元门少于$3(n-1)$ 的德摩根电路。存在一个高效的确定性算法，给定输入 C后，输出一个 n位字符串 x，使得 C（x）$\neq$异或$_n$（x）。

这立即引出了更多问题：对于使用门消除证明的其他电路下界，情况如何？尽管其中一些证明使用了复杂的组合对象，但它们仍然采用了简单的归纳式化简结构。其他形式的“构造性”又如何呢？上文讨论的高效错误生成算法并不是逻辑上的概念，因为证明本身并未被视为一个数学对象。后续研究表明，当我们考察与复杂性理论类别密切相关的算法意义上的皮亚诺算术弱片段中的证明时，构造性分离在算法意义上的许多特征仍然成立，例如 Cook 的 PV（它与确定性多项式时间相关） [14]。近期的研究开始转向理解建立已知定理以及解决复杂性理论开放问题所需的证明论强度，而门消除在这一图景中的位置完全未知 [29, 8]。上述讨论促使

▶**研究方向 3**。 从“算法”和逻辑证明论两个角度分析门消除。哪些门消除证明可以转化为构造性分离？弱逻辑理论（如 PV）能否通过门消除证明电路下界？

1.2 相关工作

证明并利用电路的结构刻画。 除了为特定布尔函数证明紧电路下界的问题之外，还有一个自然的设计问题：“其最优电路的结构是什么？”尽管从经验来看，这个问题通常很难回答 [50], ，但过去几十年来，人们已对少数函数和基进行了研究。这里的主要方法同样是门消除：证明任何偏离目标结构的情况都会导致消除“过多”门，从而与最优性矛盾。关于这一问题，最早的一些研究包括 [37] 和[3] ，它们考察了最优电路何时会分别独立于另一个输出计算双输出布尔函数的每个输出。后续研究刻画了在非门计入电路规模[21], 这一约定下，德摩根基中计算异或的最优电路；之后，这些思想又被推广到其他基 [22]。最近，Rahul Ilango [16]刻画了计算 $\bigvee_{i=1}^{n}(x_i \wedge y_i)$ 的最优电路结构，并利用这一结果取得了一项突破性的困难性归约：在指数时间假设（ETH）成立的前提下，部分函数最小电路规模问题（MCSP*）是困难的。

结构刻画也具有实际意义，但近期的应用凸显了更广泛的理论价值。具体而言，Carmosino、Dang 和 Jackman [6]明确致力于寻找并刻画具有线性规模电路的函数的最优电路，例如异或函数 XOR；他们证明，在相同的基下，即使非门不 计入电路规模， [21] 的结果依然成立。他们进一步证明，简单扩展 （仅增加极少电路复杂性的函数扩展）的最优电路结构，使得固定参数可解算法能够识别这些电路。总之，这些进展促使我们研究门消除，即便这种技术似乎无法证明超线性下界：了解它作为证明和利用结构定理的方法所固有的优势与局限，或许本身就有价值，而不必仅仅将其视为证明下界的工具。

这些关于门消除的新应用促使我们对其进行研究，即使这种技术似乎无法证明超线性下界。因此，我们认为，有必要进一步研究门消除这一技术本身固有的优势与弱点，而不只是将其视为证明下界的工具。

使用证明助手形式化电路复杂性。 除了理论意义之外，电路复杂性还具有明确的实际价值；电路最小化、综合和验证是硬件设计中的核心任务。这些应用推动了丰富的布尔电路设计软件工具的开发 [5, 43, 35, 47]。近期的开源工具（如 Cirbo [1] ）采用基于 SAT 的技术和最优电路数据库，从而实现高效的优化、分析与实验 [13]。这些工具虽有助于获得上界，却无法提供穷举搜索以外的下界求解方法。

定理证明器在研究电路复杂性时，主要用于验证。 [40] 提供了一个 HOL4 库，可用于验证常见的函数元件，并对由这些元件组成的复杂电路进行结构分析。Rocq 也被用于验证经典电路和量子电路 [4, 33]。少数经过形式化验证的下界之一——团无法由多项式规模的单调电路计算——已在 Isabelle/HOL 中形式化 [48]。然而，据我们所知，这是首次将电路建模为项图的工作。这种表示方式并非不适用于证明助手—— [49] 最近在 Isabelle/HOL 中形式化验证了项图优化。通过形式化研究门消除，我们希望逐步构建可验证电路下界的工具集，并缩小其与丰富的上界工具之间的差距。

门消除面临的障碍。已知会阻碍复杂度分离取得进展的相对化、自然证明和代数化障碍，并不适用于门消除 [36]。为探究其局限性， [12] 构造了一些函数，其电路对门消除具有“抵抗力”：任何一次代换都只能使其复杂性降低一个常数。由于门消除论证依赖于少量常数次代换，现有技术无法证明这些函数具有超线性电路下界。这项分析有两方面的局限：（i）这要求该类包含这些构造，但这一点并非总能满足；（ii）这一限制并未充分说明门消除能否用于电路下界以外的应用。

2 将布尔电路表示为项图

我们研究两个基上的一般电路。第一个是布尔函数的德摩根基 $\mathcal{B}_D = \{\wedge, \vee, \neg, 0, 1\}$：包含二元的 $\wedge$ 与 $\vee$、一元的 $\neg$，以及零元的（常量）1 和 0。第二个基是$\{\wedge, \oplus, \neg, 0, 1\}$，其中 $\vee$ 已替换为二元异或函数$\oplus$。对于某个固定的 n，电路以 $X = \{x_1, x_2, \ldots, x_n\}$ 中的零元变量作为输入，并且恰有一个指定的输出门。通常，电路被描述为有向无环图（DAG），其节点标记为函数符号或变量，边表示门之间的“导线”。对于任意这样的电路 C，底层图的无环性确保 C 通过按拓扑序依照各门所标记的函数逐个求值（从将具体比特代换到变量中开始），能计算出一个定义良好的 n元布尔函数，直至输出门取得一个值。

布尔电路的复杂性可以通过多种方式衡量，包括规模和深度。在本文中，电路规模 定义为二元 门 的数量（即，$\neg$ 门是“免费的”）。在这一度量下，非退化布尔函数的电路复杂性——即计算这些函数的最小电路的规模——在德摩根基和$\mathcal{U}_2$，（由除 $\oplus$ 及其补之外的所有二元布尔运算符组成的基）中相同。$\{\wedge, \oplus, \neg, 0, 1\}$ 与$\mathcal{B}_2 = \mathcal{U}_2 \cup \{\oplus, \odot\}$之间也存在相同的关系。我们在附录 B 中明确阐述了这一关系。

然而，要应用项图重写的结果，我们必须将电路描述为超图，即形如 $C = V_C, E_C,$ $\mathrm{lab}_C, \mathrm{att}_C$ 的四元组，其中 V_C 和 E_C 分别是有限的顶点（或节点）集和超边集；lab_C : $E_C \to \mathrm{B} \cup X$ 是边标记函数，用于记录每条边的类型；att_C : $E_C \to V_C^{\leq 3}$ 是附着函数，它为每条超边 e 分配一个非空的顶点字符串 $\mathrm{att}_C(e) = v_0, \ldots, v_\ell$，使得$|\mathrm{att}_C(e)| = 1 + \mathrm{arity}(\mathrm{lab}_C(e))$。我们将 v_0 称为 e 的结果节点，将 $v_1, \ldots, v_\ell$ 称为参数节点，并分别将它们记作 $\mathrm{res}(e)$ 和 $\mathrm{arg}(e)$。

超图 C 是一个项图，当且仅当：(1) 存在一个节点 root_G，从该节点可以到达每个节点；(2) G 是无环的——不存在某条路径使某个节点出现两次；(3) 每个节点都是唯一一条边的结果节点。在此设定中，超边表示逻辑门，顶点表示门之间的“导线”——本质上与将电路编码为有向无环图的标准方式对偶。

3 作为收敛项图重写系统的门消除

一个抽象重写系统就是一组对象 A 以及定义在A 上的二元关系 $\to$，称为重写关系。本节将构造一个系统$\mathcal{S}$，其中 A 由德摩根基上的布尔电路组成，并且当 C 通过一次门消除重写为 C' 时， $C \to C'$ 成立。为此，首先回顾关于抽象

重写系统的标准概念（参见 [2] 的定义 2.1.3）。对于元素 $a, a' \in A$，记作 $a \xrightarrow{*} a'$，表示存在一条由有限个重写步骤构成的路径，从 a 到 a'。

▶**定义 1**。重写关系 $\to$ 称为终止的当且仅当不存在无限路径 $a_0 \to a_1 \to \ldots$，合流的当且仅当对每个三元组 a，b，$b' \in A$，若 $a \xrightarrow{*} b$ 且 $a \xrightarrow{*} b'$，则存在 $c \in A$ 使得 $b \xrightarrow{*} c$ 和 $b' \xrightarrow{*} c$，并且收敛的当且仅当它既合流又终止。

对象 $a \in A$ 若不存在满足 $a \to b$ 的 b，则称其为范式。范式对象通常具有良好的结构性质，因为根据重写关系，它们是“最简单的”。例如，按照我们的系统$\mathcal{S}$，处于范式的电路 C 将具有：

- 没有双重否定。除非 C 计算的是常函数，否则没有常量——例如，$(x_i \vee \neg x_i)$ 规范化为
- 1。没有恒等门 —— 子电路 $(\gamma \wedge \gamma)$ 和 $(\gamma \vee \gamma)$ 不会出现。
-

$\mathcal{S}$ 的每条重写规则都会（按构造）减少或保持门的数量。因此，很容易证明：（1）系统 $\mathcal{S}$ 是终止的；（2）对于一个电路 C，将比特 $b = b_1, \ldots, b_n$ 代入所有输入变量后，使用 $\mathcal{S}$ 对 C 进行重写直至终止，就会得到 C 在输入 b 上的求值结果。这些基本性质将穿过 $\mathcal{S}$ 的路径与高效的电路操作和求值联系起来。要证明 $\mathcal{S}$ 具有“良好性质”，关键在于证明其合流性。我们将按以下思路展开。

1. 确定一组布尔恒等式 $\mathcal{E}_B$，使其足以用于门消除论证。**2.** 对 $\mathcal{E}_B$ 应用 Knuth-Bendix 算法，得到一个合流的公式 化简系统 $\mathcal{R}_B$。**3.** 通过 Plump 的项图重写理论，将 $\mathcal{R}_B$ 提升为合流的电路 化简系统 $\mathcal{S}$。

这套详细论述针对德摩根基（$\{\wedge, \vee, \neg\}$）；对于$\{\wedge, \oplus, \neg\}$ 的合流系统，论述方式类似，只是使用另一组布尔恒等式。

3.1 布尔恒等式

在通过门消除进行证明时，$\mathcal{E}_B$（定义 2）中收集的恒等式描述了电路化简的每个步骤。显然，$\mathcal{E}_B$ 对于布尔代数是有效的：对于所有 $g \in \{0, 1\}$，当把 $\approx$ 解释为布尔域上的等式时，每个恒等式都成立。因此，通过“可靠推理规则”从 $\mathcal{E}_B$ 推导出的结论都是真的。我们不对这种等式逻辑[1] 作形式化处理，因为我们会在下一小节将 $\mathcal{E}_B$ 转换为一个收敛项重写系统。

▶**定义 2**（有用的布尔恒等式）。 我们用 E_B 表示以下恒等式集合：

$1 \wedge 1 \approx 1$	$1 \vee 1 \approx 1$	$\neg 1 \approx 0$	$g \wedge 1 \approx g$	$g \wedge 0 \approx 0$	$g \wedge \neg g \approx 0$	$g \wedge g \approx g$
$1 \wedge 0 \approx 0$	$1 \vee 0 \approx 1$	$\neg 0 \approx 1$	$1 \wedge g \approx g$	$0 \wedge g \approx 0$	$\neg g \wedge g \approx 0$	$g \vee g \approx g$
$0 \wedge 1 \approx 0$	$0 \vee 1 \approx 1$	$\neg\neg g \approx g$	$g \vee 0 \approx g$	$g \vee 1 \approx 1$	$g \vee \neg g \approx 1$	
$0 \wedge 0 \approx 0$	$0 \vee 0 \approx 0$		$0 \vee g \approx g$	$1 \vee g \approx 1$	$\neg g \vee g \approx 1$	（与） （或）
（非）	（传递）	（固定）	（重言式）	（化简）		

[1] 参见 [2] 第 3 章，或参见 [31]中对 Birkhoff 定理的阐述。

有些布尔恒等式未纳入 $\mathcal{E}_B$，例如 $\wedge$ 与的交换律。这有两个原因：（1）纳入这些恒等式会导致一个不收敛的重写系统；（2）交换律恒等式不会“化简”布尔表达式——等式右边的布尔运算符数量并不比左边少。虽然 $\mathcal{E}_B$ 并不能完全刻画布尔代数，但足以涵盖门消除的步骤。恒等式 $\mathcal{E}_B$以机器可读形式作为补充材料提供:
`https://feasible-math.org/2026-FSCD-80-supplementary.zip`

3.2 布尔公式的收敛项重写

现在，我们将$\mathcal{E}_B$ 机械地转换为布尔公式上的抽象项重写系统，并证明其收敛性。项重写是抽象重写系统的一个经典特例，其理论动机源自代数、逻辑和编程语言理论。项是由函数符号、常量和变量构成的树状表达式。我们将通过一个中间步骤，从树状表达式过渡到类似有向无环图的表达式：电路。下面定义一般的项，尤其是德摩根公式，并介绍指定适当重写关系所需的一些辅助概念。

▶ **定义 3** (作为项的德摩根公式)。 设 Σ 是一个有限的函数符号和常量符号元组，其元数为 $\vec{d} \in \mathbb{N}^{|\Sigma|}$，并设 Z 表示一个无限变量集。$T(\Sigma, Z)$ 表示由 Σ 和 Z 上的所有项构成的集合，其归纳定义如下：

- 每个变量 $z \in Z$ 都是一个项。将函数符号 $f_i \in \Sigma$ 应用于 d_i 个项 t_1、……、t_{d_i}，得到形如$f(t_1, \ldots\ldots, t_{d_i})$ 的表达式，也是一个项。设 $B = \langle \wedge, \vee, \neg, 0, 1 \rangle$ 的元数分别为 $\langle 2, 2, 1, 0, 0 \rangle$，并称其为德摩根签名。我们用$F = T(B, X)$ 表示 $X = \{g, x_1, x_2, \ldots\ldots\}$ 构成的集合，即德摩根公式的集合。

一个代换 σ 是项之间的一种映射，可以用另一个项替换任意有限个变量，但必须保持常量和函数应用不变。因此，可以将代换写作 $\sigma = \{x_i \mapsto t\}$。一个项重写规则 是由两个项组成的有序对$\langle \ell, r \rangle$ ，记作 $\ell \to r$，满足：(1) ℓ 不是变量；(2) r 中的变量集合是 ℓ中变量集合的子集。定义在$\mathcal{T}(\Sigma, Z)$上的项重写系统 （TRS）是由项重写规则组成的集合$\mathcal{R}$ ，其中每一对项都来自$\mathcal{T}(\Sigma, Z)$。最后，我们有：

▶ **定义 4** （项重写， [32]中的定义 4.1）。 由项重写系统 R 诱导的 T 上的重写关系 $\to_{\mathcal{R}}$ (Σ, Z)定义如下：
$t \to_{\mathcal{R}} u$ 若 R 中存在规则 $\ell \to r$ 以及代换 σ ，使得
1.规则的左部与 t — $\sigma(\ell)$ 相匹配，即它是 t的一个子项。2.右部“生成” u — u 由 t 替换 $\sigma(\ell)$ 的一个出现位置 $\sigma(r)$

现在我们可以给出$\mathcal{E}_B$的精确定义：它是由来自 F 的项组成的有序对集合。我们的目标是将$\mathcal{E}_B$ 转化为一个收敛项重写系统$\mathcal{R}_B$。我们不必手动将方程改写为项重写规则（例如 $g \wedge 1 \approx g \Longrightarrow g \wedge 1 \to g$），再从头证明其收敛性，而是采用一种众所周知、旨在自动完成这一过程的方法：Knuth-Bendix 完备化算法 [20, 15]。

▶ **定理 5** (Knuth-Bendix，[45]). 给定输入为 $T(\Sigma, Z)$ 上的一组恒等式 E，若 *Knuth-Bendix* 终止，则它输出一个与 E 具有相同推论的、$T(\Sigma, Z)$ 上的收敛项重写系统 R。

从高层来看，Knuth-Bendix 完备化通过确保重叠的规则对不会产生歧义来工作。此外，该算法可以添加新规则和/或简化规则，以创建合流系统。在我们的情形中，手工运行该算法需要检查 $\binom{25}{2}$ 对方程——尽管并非每一对都可能重叠。虽然这在技术上可以手工完成，但我们改用开源的 Knuth-Bendix Completion Visualizer 软件（KBCV，[45]）对输入 $\mathcal{E}_B$ 运行该算法，从而得到

▶ **定义 6**（$\mathcal{R}_B$ 项重写系统）。

$0 \to \neg 1$	$g \wedge \neg 1 \to \neg 1$	$g \wedge 1 \to g$	$g \wedge \neg g \to \neg 1$
$\neg\neg g \to g$	$\neg 1 \wedge g \to \neg 1$	$1 \wedge g \to g$	$\neg g \wedge g \to \neg 1$
$g \wedge g \to g$	$g \vee 1 \to 1$	$g \vee \neg 1 \to g$	$g \vee \neg g \to 1$
$g \vee g \to g$	$1 \vee g \to 1$	$\neg 1 \vee g \to g$	$\neg g \vee g \to 1$
（归一化）	（固定）	（传递）	（重言式）

▶ **引理 7**。R_B 是针对德摩根公式 F的收敛项重写系统。

证明。对输入 $\mathcal{E}_B$ 运行 Knuth-Bendix 算法后，算法终止并输出了系统$\mathcal{R}_B$，即上文定义 6 中列出的系统。因此，$\mathcal{R}_B$ 是收敛的，并且与 $\mathcal{E}_B$具有相同的推论。补充材料中附有一份以认证问题格式（CPF）提供的机器可验证证明，可使用 CeTA [44]等工具进行验证。KBCV 以 CPF 2 格式输出证明，而 CeTA 要求使用 CPF 3。我们使用以下工具将输出转换为 CPF 3：`http://cl-informatik.uibk.ac.at/software/ceta/cpf3.tgz`。◀

根据结构及其对公式的影响，$\mathcal{R}_B$ 的规则分为若干组。注意，$\mathcal{R}_B$ 比$\mathcal{E}_B$包含的规则更少，因为 Knuth-Bendix 算法进行了一些化简，包括删除真值表恒等式。唯一新增的规则 $0 \to \neg 1$尤为突出。它是系统$\mathcal{R}_B$ 中唯一会增加公式中布尔运算符数量的重写规则。这与门消除相容，因为：(1) 将 0 重写为 $\neg 1$ 后，公式会变得更简单，即只出现一种类型的常量；(2) 将变量代入 0 可通过额外的一步重写为 $\neg 1$来实现。项重写规则$\mathcal{R}_B$ 以机器可读形式收录于补充材料中。

剩下的就是将这个用于公式的重写系统提升为用于电路的重写系统。

3.3 布尔电路的收敛项图重写

回顾 Plump 的定义，我们可以将项重写系统提升为项图重写系统，方法是将模式匹配和生成的概念推广如下 [32]。

▶ **定义 8** (超图态射)。对于超图 G 和 H，一个超图态射f：$G \to H$ 是一对函数 f_E：$E_G \to E_H$ 和 f_V：$V_G \to V_H$，它们保持标签，因此 f_E 将 γ 中的每条边映射到 G 中的一条边，且标签相匹配：H

$\forall \gamma \in E_G$ 实验室$_G(\gamma)$ = 实验室$_H(f_E(\gamma))$ 与

附着关系，因此对于 G 中的每条边 γ，f_V 是一个保序映射，它将附着于 γ 的顶点映射到 H 中附着于 $f_E(\gamma)$ 的顶点——回顾一下，$att(\gamma)$ 是一个字符串：$\forall \gamma \in E_G\ \forall i \in |att_G(\gamma)|\ \ f_V(att_G(\gamma))[i] = att_H(f_E(\gamma))[i]$.

项图可以用一种自然的方式转换为项：通过“展开”项图，使每条边只被读取一次。

23:10 收敛门消除与构造性电路下界

▶ 定义 *9*（项图的项表示（ [32] 中的 *3.2*））。项图G 中的节点 v 表示项 $term_G(v) = lab_G(e)(term_G(v_1), \ldots, term_G(v_\ell))$，其中 e 是满足 $res(e) = v$ 的唯一边，且 $arg(v) = v_1, \ldots v_\ell$。将 $term_G$(根$_G$)记为 $term(G)$。

同样，项 t 可以通过将函数符号 t 的每次应用编码为一条超边，转换为项图；由此得到的超图本质上就是 t的解析树。为表述模式匹配，Plump 将这一简单思路扩展为擦除特定变量：对于项 t，定义 $\diamond t$ 为用超图编码的 t 的解析树，并将所有重复变量坍缩为“开放”顶点——也就是说，对于每个 x_i，删除边 x_i，但保留唯一的结果顶点，并由解析树中原先连接到 x_i的每条边引用该顶点。如果存在图态射 $\diamond l \to L$，将 $\diamond l$ 的根映射到 L的根，那么项图 L 就是项 l 的一个实例 。给定项图 G 中的一个节点 v 以及一条项重写规则 $r \to \ell$，若从 v可达的 G 的子图（记作 $G[v]$）是 ℓ的一个实例，则二元组 $\langle v, \ell \to r \rangle$ 是一个归约项 。最后，我们回顾 Plump 图重写的一步操作：本质上，将与某条规则左部匹配的子图切出，并用右部替换。

▶ **定义 10 (**项图重写（ [32]的定义 1.4.5））。 设 G 是一个包含归约项 $\langle v, l \to r \rangle$ 的项图。存在一个适当的重写步骤，从 G 到 H，其中 H 由以下方式构造：

1. $G_1 \leftarrow G - \{e\}$ 其中 e 是满足以下条件的唯一边：$\mathrm{res}(e) = v$**2.** $G_2 \leftarrow$ 与 G_1 的不交并，其中 $\diamond r$ 将v 与 $\mathrm{root}(\diamond r)$中所有标记为变量的边，都按照将 ℓ匹配到 G的态射进行识别。

3. 垃圾回收： H 由 G_2 删除所有无法从根到达的节点和边后得到。

我们将考虑超图在两种不同等价概念下的重写。一种是超图同构，另一种是

▶ **定义 11 (**双模拟等价)。 若两个项图 G 和 H 是双模拟等价的，记作 $G \sim H$，则$\mathrm{term}(G) = \mathrm{term}(H)$。

对我们的应用而言，关键在于，两个双模拟等价的电路计算相同的函数。因此，尽管双模拟等价弱于同构，但在电路复杂性领域仍然很有价值：它蕴含函数等价。Plump 证明，在这两种等价概念下，都可以利用收敛项重写构造收敛项图重写。

▶ **定理 12** （的推论 1.7.4 和定理 1.7.12））。 如果 *R* 是一个收敛项重写系统，那么

1. 由 $\mathcal{R}$ 诱导的正规重写步骤以及**坍缩规则** 所构成的项图重写系统在超图同构意义下是收敛的，并且**2.** 仅由正规重写步骤构成的项图重写系统（**普通重写**）由 $\mathcal{R}$ 诱导，在超图双模拟等价类意义下是收敛的。

坍缩是另一条项图重写规则，允许我们在两个有根子超图之间存在保根超图态射时将它们合并。对于电路而言，这一操作相当于找出冗余子电路并将其合并为一个。这是一个自然的化简步骤；事实上，如果我们的目标是优化电路，那么任何缺少这条规则的系统都不够完善。在门消除证明中，

坍缩规则基本无关紧要——论证从次优电路出发，并对其进行化简，直到得到矛盾，从而证明电路复杂性的下界。

然而，使用坍缩仍然值得，因为它能提供一项强有力的保证：电路化简对电路的 同构类 作用完全相同。若要寻求在同构意义下成立的结构刻画，这自然是首选。然而，坍缩规则能否高效实现尚不明确。因此，在需要构造性论证时（例如定理 19），我们使用普通重写系统（不含坍缩规则）。

将此提升定理应用于我们的项重写系统 $\mathcal{R}_B$ ，用于化简德摩根公式，可得到用于门消除的系统。

▶**定理 13**。 $\mathcal{R}_B$ 诱导出两个用于德摩根 电路的项图重写系统：**1.** 系统 $\mathcal{S}$ 包含坍缩规则，并且在电路同构意义下是收敛的；**2.** 系统 $\mathcal{S}'$ 不包含坍缩规则，并且在电路双模拟意义下是收敛的。

3.4 显式电路简化系统

本节将系统 $\mathcal{S}$ 和 $\mathcal{S}'$ “编译”为具体描述德摩根电路化简的方法，以补充定理 13 完全抽象的陈述。

$\mathcal{S}$包含三个部分：(1) 子电路冗余和模式匹配的概念（定义 14 和 15）；(2) 一组以模式对形式给出的电路重写规则（附录 A 中的图 3–6）；(3) 在电路中进行模式代换的过程（算法 1）。随后，系统 $\mathcal{S}$ 是由以下条件诱导的电路上的二元关系：设定 $C \to C'$，当 (a) C 与某条规则 $\langle l \mapsto r \rangle$ 的左部 l 匹配，且 C' 是将模式 r 代换为 C中的 l 所得的结果；或者 (b) C 含有一个在 C'中“坍缩”的冗余子电路时，关系成立。系统 $\mathcal{S}'$ 与之完全相同，只是省略了坍缩规则。

▶ **定义 14** (冗余子电路的坍缩)。 若存在一个非单射超图态射 $C \mapsto D$ ，将根$_C$ 映射到根$_D$，记作 $C \succ D$，则电路 C 坍缩 为电路 D 。

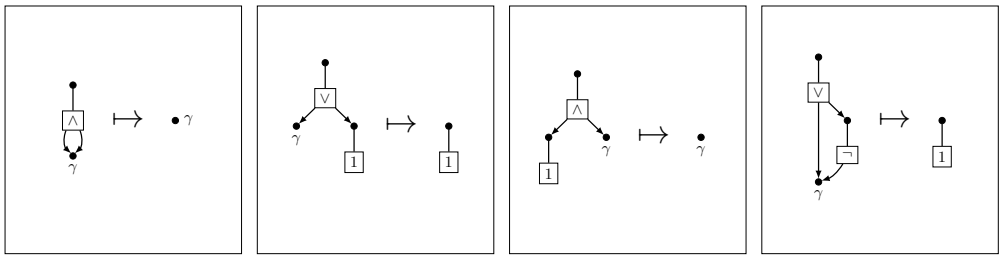
▶ **定义 15** （电路中的模式与约简）。 模式就是一种电路 L ，其中顶点γ 没有唯一的门 g ，因此可能出现 $\mathrm{att}_L\,(g)\,[0] = \gamma$ ；我们称这些顶点为开放顶点。如果存在一个态射 p： $L \to D$ ，将根$_L$映射到根$_D$，那么电路 D 就是模式 L 的一个实例。于是，给定电路 C 中的一个顶点 α 和一条规则 $L \mapsto R$，如果以 α 为根的 C 子电路（记作 $C[\alpha]$）是 L 的一个实例，那么二元组 $\langle \alpha, L \mapsto R \rangle$ 就是一个可约表达式。

$\mathcal{S}$ 的大多数重写规则都会使用标记为 γ 的开放顶点来匹配“子电路D的其余部分”。也就是说，态射 p 会将 γ 映射到 D 中未明确提及的每个节点，以此“匹配”该模式。随后，每一组规则在门消除中发挥不同的作用。

归一化确保每个电路都不包含“重复”的门，例如双重否定或平凡恒等。

固定适用于某个门因一个输入而计算常函数的情况。**传递**适用于某个门因一个输入而计算恒等函数的情况。**重言式**会移除冗余（成本高昂）的门，这些门显然计算的是常量。

固定、传递和重言式需要采用不同的规则来处理左输入和右输入，因为将“与具有交换性”和“或具有交换性”纳入重写规则会使合流性不可能实现 [42]。最后，我们将抽象的 适当重写步骤（定义 10）描述为一种具体的 操作，用于变换电路（算法 1）。



(a) 与去重 **(b)** 或修正（左） **(c)** 与传递（右） **(d)** 或重言式（左）

图 1 适当电路重写步骤的选择；完整系统请参见附录 A 中的图 3–6。

算法 1 电路简化系统$\mathcal{S}$的步骤，定义 $C \to C'$

要求： C 是一个包含可约表达式$\langle \alpha, R \mapsto L \rangle$的电路，其中 $p:\ R \to C[\alpha]$ 见证态射
$C_1 \leftarrow C - \{g\}$，且 $g = res^{-1}(\alpha)$ ▷ 移除输出线为 α
$C_2 \leftarrow C_1 + R$ ▷ 的唯一门 C不交并：将匹配规则的右侧与 $C_3 \leftarrow$ 合并 α 将顶点 $_R$ 与 C_2 ▷ 的根相连 R 连接到 C 的适当元素；如果 $\gamma \in R$，则 ▷ R 是否重用了一个子电路？$C_4 \leftarrow$ 将顶点 $p\,(\gamma)$ 与 γ 的 C_3 ▷ 相连是；将 R 的匹配元素接入 C否则$C_4 \leftarrow C_3$ ▷ R 不重用任何子电路——不执行任何操作
$C' \leftarrow$ 垃圾回收：移除从根不可达的所有顶点和边C_4

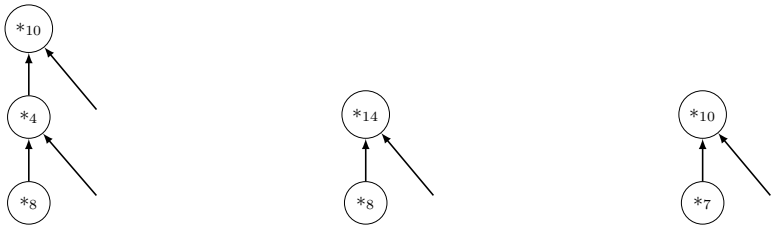
4 $\mathcal{U}_2$ 与$\mathcal{B}_2$ 电路的化简不具合流性

本节将证明，$\mathcal{U}_2$ 中任何类似的门消除表述都不具合流性。由于$\mathcal{U}_2 \subset \mathcal{B}_2$，，同样的论证表明 $\mathcal{B}_2$ 中的门消除也不具合流性。为此，我们构造一个电路和一个代换；经过应用不同的一系列重写后，它们会得到两个不同的电路（即非同构）。我们在表 1 中回顾构成$\mathcal{U}_2$ 的 14 种二元运算符。

表 1 $\mathcal{U}_2$ 基：除 $\oplus$ 及其取反 $\odot$之外的所有二元布尔运算

p	q	$*_1$	$*_2$	$*_3$	$*_4$	$*_5$	$*_6$	$*_7$	$*_8$	$*_9$	$*_{10}$	$*_{11}$	$*_{12}$	$*_{13}$	$*_{14}$
T	T	T	F	T	F	T	F	T	F	T	F	T	F	T	F
T	F	T	F	T	F	F	T	T	F	F	T	F	T	T	F
F	T	T	F	F	T	T	F	F	T	T	F	F	T	T	F
F	F	T	F	F	T	F	T	T	F	T	F	F	T	F	T

门 $*_4$ 和 $*_6$ 分别计算其第一个和第二个输入的否定。如果它们不是电路中唯一的门，那么它们就是多余的。我们可以通过重新标记它们的输入门，或重新标记它们读取的每个门，将它们移除。图 2 展示了“推动”这些多余否定的两种方式。



(a) $*_4$ 可以通过两种方式移除 **(b)** 向上推： $*_{10} \to *_{14}$ **(c)** 向下推： $*_8 \to *_7$

图 2 对 $*_4$ 的两种化简，证明在 $\mathcal{U}_2$ 和 $\mathcal{B}_2$ 上进行门消除并不收敛

如果我们的系统中同时包含这两条重写规则，那么正如示例所示，系统就不是合流的。不幸的是，我们**必须**包含既向上又向下推动否定的重写规则：如果不这样做（而是选择始终沿一个方向推动否定），那么多余的

$*_4$ 和 $*_6$ 门可能会出现在电路的两端。此外，即使我们只考虑最优电路，也无法绕开这个问题。这是因为该重写系统中的“传递”规则必然会生成中间否定门。

例如， $p *_8 q$ 等价于 $\neg p \wedge q$；如果我们想代换 $q \leftarrow 1$，就希望应用一条等价的“传递”规则。然而， $*_8$ 的输出线应传递到 $\neg p$，而它并不是电路中的实际门。因此，这种代换不会移除 $*_8$，而是必须将 $*_8$ 重新标记为 $*_4$。随后，可以通过将 $*_4$ 向上或向下“推”来消除它，从而得到两个非同构电路。同样，我们并不总能“向下推”，因为p 可能是输入，而不是二元门。从这些例子中，我们看到

▶**观察16.** 在U_2 或B_2 基中，门消除并非合流，即使将范围限制在最优电路内也是如此。

5 通过合流门消除构造电路下界

证明 通过门消除证明电路 C 具有性质P的论证通常如下。

1. 为引出矛盾，假设 C 不**具有**性质P。2. 选择一个变量 x_i 和常数 α，使用 ¬P 以及 C 的结构进行代换。3. 在代换 下化简 $C\{x_i \mapsto \alpha\}$，得到无常数电路 C'。4. 论证 P′ 对 C' 的关键性质会导致矛盾。

为了证明该系统足够强大，能够处理门消除论证，我们使用系统 $\mathcal{S}$重新考察 Schnorr 关于奇偶函数 XOR 的下界。首先，我们需要回顾奇偶函数 XOR_n:

▶ **定义 17**。 对于一个 n 位输入 $\vec{x} = (x_1, \ldots\ldots, x_n) \in \{0, 1\}^n$，$\mathrm{XOR}_n(\vec{x}) = (\sum_{i=1}^{n} x_i) \bmod 2$。

我们将 $(\neg)f$定义为“f或 $\neg f$”。对于一个（部分）赋值 α，我们将 $f\restriction_\alpha$定义为在 α下受限的函数。现在我们回顾关于 $(\neg)\mathrm{XOR}_n$的两个有用事实。

▶ **事实 1.** 异或$_n$ 是自归约的，即，对于任意 $i \in [n]$，对于任意赋值 $b \in \{0,1\}$，

$\mathrm{XOR}_n\restriction_{x_i \leftarrow b} \equiv (\neg)\mathrm{XOR}_{n-1}$.

▶ **事实 2.** 异或$_n$ 是最大敏感的，即，对于所有 $i \in [n]$，对于所有赋值 α，

异或$_n(\alpha) \neq$ 异或$_n(\alpha^{\oplus_i})$ 其中 $\alpha^{\oplus_i}$ 表示将其第 i 位翻转后的 α。

Schnorr 的证明是典型的门消除证明。我们在附录 C 中详细介绍如何使用我们的系统重新证明该结果，这里仅作简要概述。

▶ **定理 18** (Schnorr， [38])。异或$_n$ 在德摩根基中需要$3(n-1)$ 个门。

证明概要。令 C 为计算异或$_n$ （其中 $n \geq 2$）的最优电路，我们的目标是找到一个输入超边的代换，使得在重写步骤中至少有 3 条高代价门超边（标记为 $\wedge$ 或 $\vee$）被消除。首先固定一个拓扑序，并令 h 为其中第一个高代价门。于是， h 有两个参数，其项分别为（¬）x_i 和（¬）x_j。如果读取（¬）x_i的高代价门只有 h ，我们可以代换 x_j ，通过固定规则消除 h ，从而使 x_i 断开连接。然而，所得电路根据自约化性（事实 1）计算（¬）异或$_{n-1}$，不再依赖于 x_i。这与异或$_{n-1}$ 的非退化性（事实 2）矛盾。

因此，另一个高代价门 $f \neq h$ 读取 (¬)x_i。此外，(¬)f 不能是输出，因为我们可以通过代换 x_i 将其固定，这再次与事实 1 矛盾。令 f' 为读取 f 的任意二元门。接着，我们选择能固定 f 的 x_i 的代换。经过化简后，这最终至少消除了 3 个高代价门 h， f， f'，且所得电路根据事实 1 计算 XOR$_{n-1}$ 。最后，我们应用归纳法得到所需的下界。◀

Schnorr 的证明很简单，但若从形式化方法的角度来看，很难不将其视为一种分解异或电路的算法；我们的附录 C.1 中的结构化证明尤其如此。门消除通过论证：如果一个电路不具备某种最小结构，那么它就无法计算其底层函数，从而得到电路下界。由于函数是由其在输入上的行为定义的，因此我们似乎有理由期待能够推导出该电路在哪些输入上会出错。我们证明，至少对于 Schnorr 的证明而言，这一直觉是正确的：我们给出了首个针对通过门消除证明的电路下界的构造性分离。

▶ **定理 19.** 设 C 是一个以 n 为输入、规模为 $s < 3(n-1)$ 的德摩根电路，声称计算的是异或$_n$。我们可以在关于 n 的多项式时间内构造一个输入 $x \in \{0,1\}^n$ ，使得 $C(x) \neq$ 异或$_n(x)$。

证明。 设 C 是一个以 n 为输入、规模为 $s < 3(n-1)$ 的德摩根电路，并设 f 是它计算的函数。若 $n = 2$，则我们可以在常数时间内穷举评估所有四种可能的输入，从而找到一个反例。对于较大的 n，我们可以运行算法 2。概括而言，该算法将遵循 Schnorr 的证明；在每次迭代中，它会尝试找到一种代换，至少消去三个门。如果做不到，那么该电路的函数就会违反事实 2，我们可以轻易提取出一个反例。如果成功，它则会生成一个规模太小、无法在少一个变量的情况下计算异或的电路，并对其递归处理。这个过程会持续下去，直到最终剩余的电路声称计算（¬）异或$_2$ ，且门数少于三个；此时我们再次可以直接穷举其四种补全。

算法 2 跟踪（i）部分限制 α 以及（ii）一份列表 V，其中列出电路仍未限制的变量。每次迭代中，它都会考虑 C_i，这是一个计算 $f\restriction_\alpha$的**规范化**电路。在第 4–5 行，算法可以识别出与事实 2 明显矛盾的情况。如果 C_i 相对于 V 中的某些变量退化，那么 $f\restriction_\alpha$ 就不依赖于某个x_j（对于某个 $j \in V$）。令 $\hat{}\alpha$为 α的任意补全，则 $C(\hat{}\alpha) = f(\hat{}\alpha) = f(\hat{}\alpha^{\oplus_j}) = C(\hat{}\alpha^{\oplus_j})$；由于 XOR$_n(\hat{}\alpha) \neq$ XOR$_n(\hat{}\alpha^{\oplus_j})$，因此 C在这两个输入中的一个上出错。

否则，在第 7 行，算法会选择 C_i 中拓扑极小的二元门，该门必须读取 (¬)x_p 和 (¬)x_q ，其中 $i \neq j$ ，因为 C_i 已规范化。与 Schnorr 一样，(¬)x_p 必须由除 h 之外的另一个二元门 f 读取。如果第 9 行的情况并非如此，那么我们就遇到了与前一个 DEGEN 情况类似的情形：在加入 x_q 的正确代换后

代换到 α 并进行归一化后，$f\restriction_\alpha$ 将不再依赖于 x_p，而且对于任意补全 $\hat{}\alpha$ 或 $\hat{}\alpha^{\oplus_p}$，C 都必须出错。此外，$(\neg)f$ 不能是输出，否则我们就可以在第 15–17 行将电路固定为常量。若它是输出，我们就可以取任意补全 $\hat{}\alpha$，而它或 $\hat{}\alpha^{\oplus_q}$ 将成为反例，因为 f 不再依赖于 x_q。

如果算法没有提前返回，那么按照 Schnorr 的做法，存在某个 f'，它读取 $(\neg)f$。算法 2 为 x_p 选择一种代换，使得化简移除 h，f和f'，并在每次循环迭代中至少将电路规模缩小三。因此，得到的C_{i+1}太小，无法计算异或$_{|V|}$。在最坏情况下，该算法会执行 $n-2$ 次迭代，而根据事实 1，得到的 C_{n-2} 据称会计算 $(\neg)$异或$_2$。然而，其规模至多为 $s-3(n-2) < 3(n-1) - 3(n-2) = 3$。因此，我们只需对 α 的所有四种补全进行穷举，而 C 必定会在其中至少一种情况下出错。

很容易看出，该过程可在多项式时间内运行（关于 n）。循环迭代次数至多为线性，每个条件都可以在多项式时间内检查。由于我们使用的是不发生坍缩的系统 $\mathcal{S}'$，每次迭代中的代换和化简都可以在多项式时间内完成。要提取一个反例，我们只需计算电路在常数个输入上的输出，而每次求值都可以在多项式时间内完成。◀

算法 2 高效搜索坏限制 α，以证明 C 无法计算异或 .

要求： C 是一个具有 n 个输入变量、s 个门的电路，且有 $n > 3$ 和 $s < 3(n-1)$

```
1:  i ← 0；  C_0 ← 规范化 C；  α ← ∅；  V ← [n]
2:  轮数；工作电路；工作限制；活跃变量 ◁
3:  当 |V| > 2时执行
4:      如果 某个变量 x_j 对某个 j ∈ V 未在 C_i 中读取
5:          返回 "DEGEN 在 x_j 之下 α"
6:      断言： C_i 是范式，且并非显然无法计算异或_{|V|}
7:      h ← 拓扑极小的{∧, ∨}-门，属于 C_i
8:      断言： h 有输入 (¬) x_p，(¬) x_q，对于 p, q ∈ V和 p ≠ q
9:      ▷该断言成立，因为 C_i 是范式 ◁
10:     如果 (¬) 的扇出为x_p 为1，那么
11:         添加 fixer(h, x_q) ↦ x_q to α
12:         ▷ fixer(h, x_q) 是对 x 的代换 _q 通过固定规则移除 h 的 ◁
13:         返回 "DEGEN 中 x _p 在 α" 下
14:     else
15:         f ← 不同的门读取 x _p 除了 h，因此 f ≠ h
16:     如果(¬)f 是 C_i 的输出门那么
17:         添加 fixer(f, x_p) ↦ x_p to α
18:         返回 "CONST位于 α"
19:     else
20:         f' ← C_i中 f 的拓扑极小高代价后继节点
21:     添加 fixer(f, x_p) ↦ x_p to α
22:     C_{i+1} ← 代入 x _p ← fixer(f, x_p) 中的 C_i 并化简 ▷消去 h、f 和 f'
23:     ▷根据系统 S' 的收敛性，此步骤是确定性的——对……进行贪心选择
          化简在 C_{i+1} 的互模拟意义下相同，这正是我们所需要的 ◁
24:     V ← V | {p}
25:     i ← (i + 1)
26:     断言： C_{i+1} 是一个规模过小、无法计算 (¬) 异或_{|V|}的规范化电路
27: 返回 "FAILS低于 α"  ▷ "失败搜索"，通过穷举很容易找到错误
```

参考文献

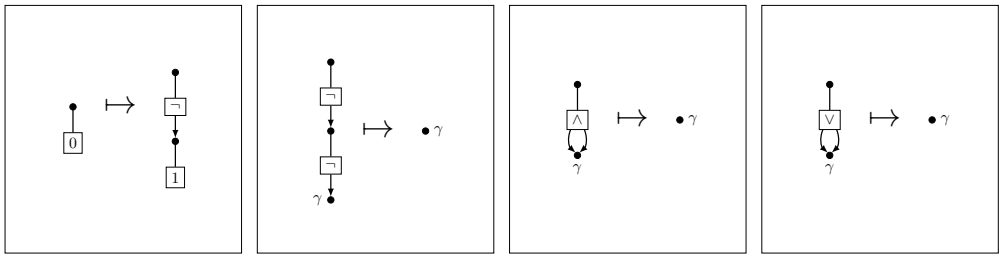
1 Daniil Averkov、Tatiana Belova、Gregory Emdin、Mikhail Goncharov、Viktoriia Krivogornitsyna、Alexander S. Kulikov、Fedor Kurmazov、Daniil Levtsov、Georgie Levtsov、Vsevolod Vaskin 和 Aleksey Vorobiev。Cirbo：一种用于布尔电路分析与综合的新工具。载于 Toby Walsh、Julie Shah 和 Zico Kolter 编，AAAI-25，由人工智能促进协会赞助，2025年2月25日至3月4日，美国宾夕法尼亚州费城，第11105–11112页。AAAI出版社，2025年。网址：https://doi.org/10.1609/aaai.v39i11.33207,doi:10.1609/AAAI.V39I11.33207。2 Franz Baader 和 Tobias Nipkow。《项重写及相关内容》。剑桥大学出版社，1998年。3 Norbert Blum 和 Martin Seysen。对同时计算与和或非的所有最优网络的刻画。《计算机学报》21：171–181，1984年。 doi:10.1007/BF00289238。4 Thomas Braibant。Coquet：用于验证硬件的 Coq 库。载于 Jean-Pierre Jouannaud 和 Zhong Shao 编，《认证程序与证明——第一届国际会议，CPP 2011》，台湾垦丁，2011年12月7日至9日。论文集，《计算机科学讲义》第7086卷，第330–345页。施普林格，2011年。 doi:10.1007/978-3-642-25379-9_24。5 Robert K. Brayton 和 Alan Mishchenko。ABC：一种学术界与工业界通用的验证工具。载于 Tayssir Touili、Byron Cook 和 Paul B. Jackson 编，《计算机辅助验证》第22届国际会议，CAV 2010，英国爱丁堡，2010年7月15日至19日。论文集，《计算机科学讲义》第6174卷，第24–40页。施普林格，2010年。 doi:10.1007/978-3-642-14295-6_5。6 Marco Carmosino、Ngu Dang 和 Tim Jackman。用于多项式时间异或的简单电路扩展。arXiv预印本 arXiv:2511.16903。将发表于 STACS 2026，2025年。7 Lijie Chen、Ce Jin、Rahul Santhanam 和 Ryan Williams。构造性分离及其推论。TheoretiCS，3，2024年。网址： https://doi.org/10.46298/theoretics.24.3, doi:10.46298/THEORETICS.24.3。8 Lijie Chen、Jiatu Li 和 Igor C. Oliveira。复杂性下界的逆向数学。载于《第65届IEEE计算机科学基础年会》，FOCS 2024，美国伊利诺伊州芝加哥，2024年10月27日至30日，第505–527页。IEEE，2024年。 doi:10.1109/FOCS61266.2024.00040。9 Evgeny Demenkov 和 Alexander S. Kulikov。关于仿射分散器电路复杂性的 3n − o(n) 下界的一个初等证明。载于 Filip Murlak 和 Piotr Sankowski 编，《计算机科学数学基础 2011——第36届国际研讨会》，MFCS 2011，波兰华沙，2011年8月22日至26日。论文集，《计算机科学讲义》第6907卷，第256–265页。施普林格，2011年。 doi:10.1007/978-3-642-22993-0_25。10 Magnus Gausdal Find、Alexander Golovnev、Edward A. Hirsch 和 Alexander S. Kulikov。显式函数电路复杂性的优于 3n 的下界。载于《2016年第57届IEEE计算机科学基础年会》（FOCS），第89–98页，2016年。doi:10.1109/FOCS.2016.19。11 Merrick L. Furst、James B. Saxe 和 Michael Sipser。奇偶函数、电路与多项式时间层次。《数学系统理论》17(1)：13–27，1984年。 doi:10.1007/BF01744431。12 Alexander Golovnev、Edward A. Hirsch、Alexander Knop 和 Alexander S. Kulikov。门消除的局限性。《计算机与系统科学杂志》96：107–119，2018年。 doi:10.1016/j.jcss.2018.04.005。13 Mikhail Goncharov、Alexander S. Kulikov 和 Georgie Levtsov。更小的比特加法电路。arXiv预印本 arXiv:2509.13966。将发表于 STACS 2026，2025年。14 Stefan Grosser 和 Marco Carmosino。学生—教师构造性分离与有界算术中的（不可）证明性：间隙见证。载于 Michal Koucký 和 Nikhil Bansal 编，《第57届ACM理论计算机科学年会论文集》，STOC 2025，捷克布拉格，2025年6月23日至27日，第1341–1347页。ACM，2025年。 doi:10.1145/3717823.3718216。

15 Gérard Huet. Knuth-Bendix 完备化算法正确性的完整证明。《计算机与系统科学杂志》，23(1):11–21，1981。URL: https://www.sciencedirect.com/science/article/pii/0022000081900027, doi:10.1016/0022-0000(81)90002-7. 16 Rahul Ilango. 常数深度公式版本和部分函数版本的最小电路规模问题都是困难的。SIAM计算期刊，0(0):FOCS20–317–FOCS20–367，2020。 arXiv:https://doi.org/10.1137/20M1383562, doi:10.1137/20M1383562. 17 Russell Impagliazzo 和 Avi Wigderson. 若 E 需要指数规模电路，则 P = BPP：异或引理的去随机化。载于 Frank Thomson Leighton 和 Peter W. Shor 编，《第二十九届 ACM 计算理论年会论文集》，美国得克萨斯州埃尔帕索，1997 年 5 月 4–6 日，第 220–229 页。ACM，1997。 doi:10.1145/258533.258590. 18 Kazuo Iwama 和 Hiroki Morizumi. 布尔电路的显式 5n - o(n) 下界。载于 Krzysztof Diks 和 Wojciech Rytter 编，《2002 年计算机科学数学基础会议》，第 27 届国际研讨会，MFCS 2002，波兰华沙，2002 年 8 月 26–30 日，论文集，《计算机科学讲义》第 2420 卷，第 353–364 页。施普林格，2002。 doi:10.1007/3-540-45687-2_29. 19 Valentine Kabanets 和 Russell Impagliazzo. 对多项式恒等式测试进行去随机化意味着证明电路下界。《计算复杂性》，13(1-2):1–46，2004。URL: https://doi.org/10.1007/s00037-004-0182-6, doi:10.1007/S00037-004-0182-6. 20 DONALD E. KNUTH 和 PETER B. BENDIX. 泛代数中的简单字问题艹本文所述研究部分由美国海军研究办公室资助。载于 JOHN LEECH 编，《抽象代数中的计算问题》，第 263–297 页。培格曼出版社，1970。URL: https://www.sciencedirect.com/science/article/pii/B978008012975450028X, doi:10.1016/B978-0-08-012975-4.50028-X. 21 Yu A Kombarov. 线性布尔函数的最小电路。《莫斯科大学数学通报》，66(6):260–263，2011。 22 Yu A Kombarov. 奇偶函数电路的复杂度与结构。《数学科学杂志》，233:95–99，2018。 23 Oded Lachish 和 Ran Raz. 布尔电路的显式 4.5n - o(n) 下界。载于《第三十三届 ACM 计算理论年会论文集》，STOC '01，第 399–408 页，美国纽约州纽约市，2001。美国计算机协会。 doi:10.1145/380752.380832. 24 Leslie Lamport. 如何撰写证明。《美国数学月刊》，102(7):600–608，1995。 25 Leslie Lamport. 如何撰写 21 世纪的证明。《不动点理论与应用杂志》，11:43–63，2012。 26 Jiatu Li 和 Tianqi Yang. 显式函数的 3.1n - o(n) 电路下界。载于 Stefano Leonardi 和 Anupam Gupta 编，STOC '22：第 54 届 ACM SIGACT 计算理论年会，意大利罗马，2022 年 6 月 20–24 日，第 1180–1193 页。ACM，2022。doi:10.1145/3519935.3519976. 27 Wolfgang Maass. 确定性与非确定性单带图灵机的二次下界（扩展摘要）。载于 Richard A. DeMillo 编，《第 16 届 ACM 计算理论年会论文集》，1984 年 4 月 30 日至 5 月 2 日，美国华盛顿哥伦比亚特区，第 401–408 页。ACM，1984。doi:10.1145/800057.808706. 28 Noam Nisan 和 Avi Wigderson. 困难性与随机性。《计算机与系统科学杂志》，49(2):149–167，1994。 doi:10.1016/S0022-0000(05)80043-1. 29 Igor C. Oliveira.《SIGACT 新闻》复杂性理论专栏第 124 期：计算复杂性理论的元数学。《SIGACT 新闻》，56(1):41–68，2025。 doi:10.1145/3726856.3726862. 30 Wolfgang J. Paul. 布尔函数组合复杂度的 2.5 n 下界。载于《第七届 ACM 计算理论年会论文集》，STOC '75，第 27–36 页，美国纽约州纽约市，1975。美国计算机协会。 doi:10.1145/800116.803750.

31 David A. Plaisted. 等式推理与项重写系统，第 274–364 页。牛津大学出版社，美国，1993。 32 Detlef Plump. 项图重写。《图文法与图变换计算手册：第 2 卷：应用、语言与工具》，第 3–61 页，1999。 33 Robert Rand、Jennifer Paykin 和 Steve Zdancewic. QWIRE 实践：在 Coq 中对量子电路进行形式化验证。载于 Bob Coecke 和 Aleks Kissinger 编，《第 14 届量子物理与逻辑国际会议论文集》，QPL 2017，荷兰奈梅亨，2017 年 7 月 3–7 日，EPTCS 第 266 卷，第 119–132 页，2017。 doi:10.4204/EPTCS.266.8. 34 NP Red'kin. 函数元件构成的电路最小性证明。《系统理论研究：控制论问题》，第 85–103 页，1973。 35 Franz-Xaver Reichl、Friedrich Slivovsky 和 Stefan Szeider. 基于量化布尔公式的精确综合实现电路最小化。载于 Brian Williams、Yiling Chen 和 Jennifer Neville 编，《第三十七届 AAAI 人工智能会议，AAAI 2023；第三十五届人工智能创新应用会议，IAAI 2023；第十三届人工智能教育进展研讨会，EAAI 2023》，美国华盛顿哥伦比亚特区，2023 年 2 月 7–14 日，第 4087–4094 页。AAAI出版社，2023。URL: https://doi.org/10.1609/aaai.v37i4.25524, doi:10.1609/AAAI.V37I4.25524. 36 Hanlin Ren 和 Rahul Santhanam. 元复杂性的相对化视角。载于 Petra Berenbrink 和 Benjamin Monmege 编，《第 39 届理论计算机科学专题国际研讨会》，STACS 2022，法国马赛（线上会议），2022 年 3 月 15–18 日，LIPIcs 第 219 卷，第 54:1–54:13 页。Schloss Dagstuhl - 莱布尼茨信息学中心，2022。URL: https://doi.org/10.4230/LIPIcs.STACS.2022.54, doi:10.4230/LIPICS.STACS.2022.54. 37 Jürgen Sattler. 布尔函数同步计算网络。载于 Peter Deussen 编，《理论计算机科学：第五届 GI 会议》，德国卡尔斯鲁厄，1981 年 3 月 23–25 日，论文集，《计算机科学讲义》第 104 卷，第 32–40 页。施普林格，1981。URL: https://doi.org/10.1007/BFb0017293, doi:10.1007/BFB0017293. 38 Claus-Peter Schnorr. 布尔函数复杂度的两个线性下界。《计算》，13(2):155–171，1974。 doi:10.1007/BF02246615. 39 Claude. E. Shannon. 二端开关电路的综合。《贝尔系统技术杂志》，28(1):59–98，1949。 doi:10.1002/j.1538-7305.1949.tb03624.x. 40 Sumayya Shiraz 和 Osman Hasan. 使用 HOL定理证明器进行组合电路验证的库。《IEEE计算机辅助设计集成电路与系统汇刊》，37(2):512–516，2018。 doi:10.1109/TCAD.2017.2705049. 41 Roman Smolensky. 布尔电路复杂性下界理论中的代数方法。载于 Alfred V. Aho 编，《第 19 届 ACM 计算理论年会论文集》，1987，美国纽约州纽约市，第 77–82 页。ACM，1987。 doi:10.1145/28395.28404. 42 Rolf Socher-Ambrosius. 布尔代数不存在收敛项重写系统。载于 Ronald V. Book 编，《重写技术与应用》，第 4 届国际会议 RTA-91，意大利科莫，1991 年 4 月 10–12 日，论文集，《计算机科学讲义》第 488 卷，第 264–274 页。施普林格，1991。 doi:10.1007/3-540-53904-2_102. 43 Mathias Soeken、Heinz Riener、Winston Haaswijk 和 Giovanni De Micheli. EPFL逻辑综合库。CoRR，abs/1805.05121，2018。URL: http://arxiv.org/abs/1805.05121,arXiv:1805.05121. 44 Christian Sternagel 和 René Thiemann. Knuth-Bendix序与 Knuth-Bendix 完备化的形式化。载于 Femke van Raamsdonk 编，《第 24 届重写技术与应用国际会议》，RTA 2013，荷兰埃因霍温，2013 年 6 月 24–26 日，LIPIcs 第 21 卷，第 287–302 页。Schloss Dagstuhl - 莱布尼茨信息学中心，2013。URL: https://doi.org/10.4230/LIPIcs.RTA.2013.287, doi:10.4230/LIPICS.RTA.2013.287. 45 Thomas Sternagel 和 Harald Zankl. KBCV - Knuth-Bendix Completion Visualizer。载于 Bernhard Gramlich、Dale Miller 和 Uli Sattler 编，《自动推理——第六届国际会议

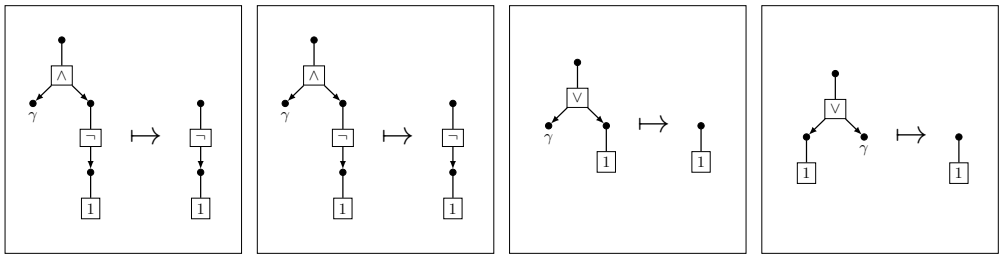
联合会议 IJCAR 2012，英国曼彻斯特，2012年6月26日至29日。论文集，卷 7364，《计算机科学讲义》，第 530–536 页。施普林格，2012年。 doi:10.1007/978-3-642-31365-3_41。46 Larry J. Stockmeyer。关于某些对称布尔函数的组合复杂度。《数学系统理论》，10:323–336，1977年。 doi:10.1007/BF01683282。47 Ruofei Tang、Xuliang Zhu、Xinyi Zhang、Lei Chen、Xing Li、Mingxuan Yuan 和 Jianliang Xu。EDGE：用于 GIG 综合的 DBMS 赋能布尔分解。载于第 62 届 ACM/IEEE 设计自动化会议，DAC 2025，美国加利福尼亚州旧金山，2025年6月22日至25日，第 1–7 页。IEEE，2025年。 doi:10.1109/DAC63849.2025.11133306。48 René Thiemann。团无法由多项式规模的单调电路求解。形式化证明档案，2022年5月。 https://isa-afp.org/entries/Clique_and_Monotone_Circuits.html，形式化证明开发。49 Brae J. Webb、Ian J. Hayes 和 Mark Utting。使用 Isabelle/HOL 验证项图优化。载于 Robbert Krebbers、Dmitriy Traytel、Brigitte Pientka 和 Steve Zdancewic（编），第 12 届 ACM SIGPLAN 认证程序与证明国际会议论文集，CPP 2023，美国马萨诸塞州波士顿，2023年1月16日至17日，第 320–333 页。ACM，2023年。doi:10.1145/3573105.3575673。50 Ingo Wegener。《布尔函数的复杂度》。Wiley-Teubner，1987年。51 Ryan Williams。非一致 ACC 电路下界。《J. ACM》，61(1):2:1–2:32，2014年。doi:10.1145/2559903。52 Uri Zwick。基于单边二元布尔函数基的某些对称布尔函数组合复杂度的 4n 下界。SIAM计算期刊，20(3):499–505，1991年。

门消除规则



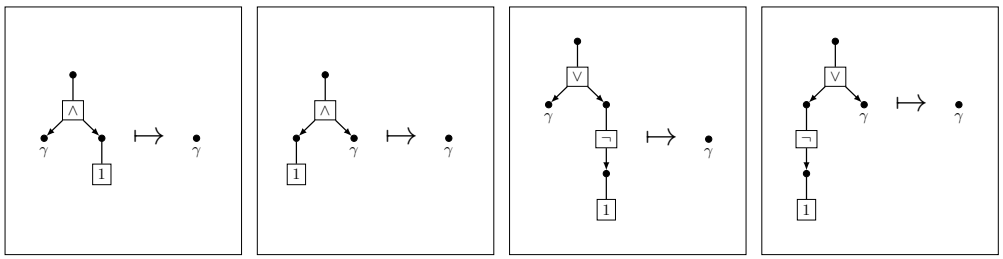
(a) 零消除 (b) 双重取反消除 (c) 与去重 (d) 或去重

图 3 规范化规则



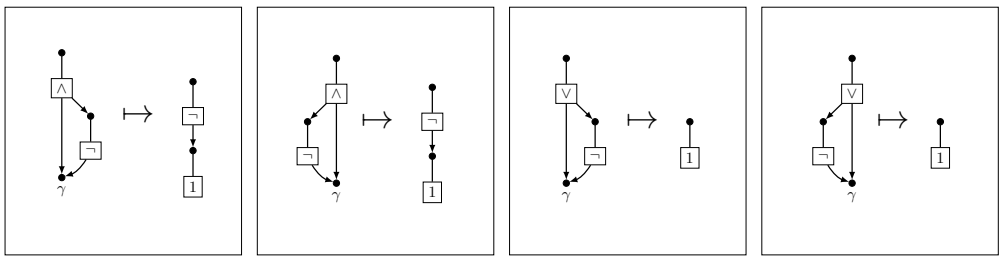
(a) AND左规则 (b) AND右规则 (c) OR左规则 (d) OR右规则

图 4 修正规则



(a) AND左规则 (b) AND右规则 (c) 或左规则 (d) 或右规则

图 5 传递规则



(a) AND左规则 (b) AND右规则 (c) OR左规则 (d) OR右规则

图 6 重言式规则

B NOT 门免费时的电路规模：比较不同的基

回想一下，$\mathcal{U}_2$ 是由除二元异或 $\oplus$ 和等价 $\odot$ 之外的所有二输入布尔函数组成的基，而德摩根基为 $\{\wedge, \vee, \neg\}$。我们明确证明，当 $\neg$ 门不计入这些复杂度度量时，这些基在规模和深度方面等价。当后一种基中的 $\neg$ 门免费时，$\mathcal{B}_2$ 与 $\{\wedge, \oplus, \neg\}$ 等价的证明完全相同。

▶ **引理 20** (民间结果)。 设 f 是除 $f(x) = \neg x$之外的任意非退化布尔函数。那么，在 $_2$ 门免费的条件下， f 在 U_2 (B_2) 基和德摩根（$\{\wedge, \oplus, \neg\}$）基中的规模与深度复杂度相等。

证明。 令 $CC_U(f)$ 表示 f在 $\mathcal{U}_2$ 基中的电路规模复杂度。令 $CC_D(f)$表示 f 在非门免费的德摩根基中的电路规模复杂度。

我们首先通过将任意最优的$\mathcal{U}_2$ 电路转换为规模相同的德摩根电路，来论证 $CC_D(f) \leq CC_U(f)$。我们回顾一下构成表 2 中$\mathcal{U}_2$的 14 种二元布尔运算。

p	q	$*_1$	$*_2$	$*_3$	$*_4$	$*_5$	$*_6$	$*_7$	$*_8$	$*_9$	$*_{10}$	$*_{11}$	$*_{12}$	$*_{13}$	$*_{14}$
T	T	T	F	T	F	T	F	T	F	T	F	T	F	T	F
T	F	T	F	T	F	F	T	T	F	F	T	F	T	T	F
F	T	T	F	F	T	T	F	F	T	T	F	F	T	T	F
F	F	T	F	F	T	F	T	T	F	T	F	F	T	F	T

表 2 $\mathcal{U}_2$ 基由除异或$_2$ 和 $\neg$异或$_2$以外的所有二元布尔运算组成

我们声称，对于 f，不存在使用 $*_i$ 门的最优电路，条件是 $i \in [6]$。对于 $*_1$ 和 $*_2$，将这些门替换为常量 0 和 1 会减小电路的规模。同样， $*_3$ 和 $*_5$ 门不是最优的，因为它们分别只是计算第一个和第二个输入。如果电路中出现 $*_3$ 门，我们可以改为将它的第一个输入直接接到其输出端，效果相同。门 $*_4$ 和 $*_6$ 分别只是对第一个和第二个输入取反。同样，如果内部出现 $*_4$ 或 $*_6$，我们可以将第一个和第二个输入直接接到这些门的输出端，但必须更改这些门的标签。我们在下面的表 3 中记录这些变换：

		$*_7$	$*_8$	$*_9$	$*_{10}$	$*_{11}$	$*_{12}$	$*_{13}$	$*_{14}$
$\neg p$	q	$*_{12}$	$*_{11}$	$*_{13}$	$*_{14}$	$*_8$	$*_7$	$*_9$	$*_{10}$
p	$\neg q$	$*_{13}$	$*_{14}$	$*_{12}$	$*_{11}$	$*_{10}$	$*_9$	$*_7$	$*_8$

表 3 移除多余内部 $*_4$ 或 $*_6$ 门时的门替换

例如，如果一个 $*_4$ 门是 $*_7$的第一个输入，那么我们可以移除 $*_4$ 门，将其第一个输入接入 $*_7$ 门的第一个输入，然后将 $*_7$ 门重新标记为 $*_{12}$ 门。如果一个 $*_4$ 或 $*_6$ 门是电路的输出，那么它必定是在对剩余的 8 个布尔函数之一取否定，因为 $f(x) \neq \neg x$。我们可以应用表4中列出的变换。

	$\neg(*_7)$	$\neg(*_8)$	$\neg(*_9)$	$\neg(*_{10})$	$\neg(*_{11})$	$\neg(*_{12})$	$\neg(*_{13})$	$\neg(*_{14})$
等价于	$*_8$	$*_7$	$*_{10}$	$*_9$	$*_{12}$	$*_{11}$	$*_{14}$	$*_{13}$

表4 用于消除输出端多余否定的门替换

最后，基中剩余的 8 个二元布尔函数都可以仅用一个二元门表示为德摩根基：

	$*_7$	$*_8$	$*_9$	$*_{10}$	$*_{11}$	$*_{12}$	$*_{13}$	$*_{14}$
等价于	$p \vee \neg q$	$\neg p \wedge q$	$\neg p \vee q$	$p \wedge \neg q$	$p \wedge q$	$\neg p \vee \neg q$	$p \vee q$	$\neg p \wedge \neg q$

表5 $_2$ 与德摩根公式之间的翻译，适用于 $*_7$ 到 $*_{14}$

为证明 $CC_U(f) \leq CC_D(f)$，我们说明如何将任意最优德摩根电路转换为一个规模相同的等价 $\mathcal{U}_2$ 电路。为此，若任何 $\neg$ 门的扇出为 m 且$m > 1$，我们就将该 $\neg$ 门拆分为 m 个副本，每个副本的扇出恰好为 1。我们还会消除所有双重否定。这两种变换都不会改变电路的规模。

然后，按照递增的拓扑序，我们将每个 $\wedge$ 与 $\vee$ 门替换为$\mathcal{U}_2$ 门之一，具体取决于其左侧或右侧输入是否被取反。共有 8 种可能的配置，对应关系见表5。最后，如果整个电路被取反（即最终输出门是 $\neg$），我们就利用表4中列出的变换，将其并入下方的 $\mathcal{U}_2$ 门。

由于 $CC_U(f) \leq CC_D(f)$ 且 $CC_D(f) \leq CC_U(f)$，我们得到 $CC_U(f) = CC_D(f)$。注意，这些变换也都不会改变电路的深度，因此这些基在深度复杂度方面也等价。◀

C 电路简化实战：证明施诺尔下界

本节将给出定理 18 的完整证明，即施诺尔关于 XOR$_n$ 的下界证明。我们将使用系统 $\mathcal{S}$ 来说明，该系统足以涵盖传统的门消除论证。提醒一下，该定理指出，任何计算 XOR$_n$ 的德摩根电路C 都至少需要 3$(n-1)$ 个高代价门 [38]。

简单回顾一下，通过门消除进行证明时，常用以下论证来证明电路 C 具有性质 P。

1. 假设 C 不具有性质 P。2. 选择一个变量 x_i 和常数 α，利用 ¬P 及 C的结构进行代换 $\{x_i \mapsto \alpha\}$。3. 在代换 $\{x_i \mapsto \alpha\}$ 下化简 C，得到一个无常数电路 C'。4. 论证 C' 的关键性质 P′ 会导致矛盾，因此 C 必定具有性质 P。

系统 $\mathcal{S}$ 形式化了上文第三步中使用的化简过程。通常，这并无必要：关键性质大致是 P′ = “化简消除了四个门”，而且显然，每一系列化简步骤最终都会得到一个具有性质 P′的电路 C'。然而，为了以算法方式处理门消除的结果，我们必须断言更复杂的化简后性质，例如 P′ = “输入 x_j 的扇出为一”，其中 x_j 是 x_i 在 C中的兄弟节点。每次化简终止后，这些更微妙的性质并不那么容易看出是否成立。此外，我们通常希望仔细安排并分析门消除的最初几步，然后在不详细考察的情况下应用“所有剩余的化简”。关键在于，无论消除步骤按何种顺序进行，都必须得到 P′。

电路的收敛简化过程$\mathcal{S}$ 具有以下性质：对 C 中任意输入 x_i 代入 α 后，$\mathcal{S}$ 的每次有效运行都会终止于相同的电路

C'，在互模拟意义下。因此，要执行上述门消除论证模板，只需展示$\mathcal{S}$ 的一次特定运行，并论证所得的 C' 具有关键性质 P′。

在附录 C.1 中，我们采用 Lamport [24, 25] 的结构化证明格式，给出定理 18 的另一种证明。由于其中存在分类讨论以及反复使用反证法，这种替代格式可能更易于验证。此外，我们认为这种呈现方式使我们在 Schnorr 证明中出现的复杂分类讨论更加明确、更易阅读。此外，这种证明风格更便于使用计算机进行验证。

定理 18 的证明。设 C 是异或$_n$ 的一个最优电路，且具有 $n \geq 2$。在原始证明中，目标是找到某个输入节点的某种设定，使得门消除至少移除 3 个高代价（$\wedge$ 与 $\vee$）门。除记号差异外，我们的目标保持不变。我们将找到对一条输入超边的代换，使得在重写过程中至少移除 3 条高代价门超边（标记为 $\wedge$ 或 $\vee$ 的那些）。遵循 Schnorr，我们通过反复反证，在某个输入周围局部构建电路；我们执行代换与重写，以与 C的最优性或异或$_n$ 的向下自归约性相矛盾，从而迫使 C 具有所需结构。

为了平滑地从将电路视为图过渡到将其视为项图，我们将直接把输入超边称为输入，把门超边称为门。我们将从较高层次描述代换和重写步骤。

我们希望先按“拓扑序”对门排序。在传统的电路视图中，电路是有向无环图（其中门是节点），这个概念很直观。不过，由于我们的门是边，因此必须间接完成排序。我们可以对 C 的顶点进行拓扑排序；由于每个节点都是唯一一条边的结果节点，接着便可根据结果节点对门排序。从现在起，凡是提到按拓扑序对输入或门排序，严格来说指的都是这一过程。

固定一个拓扑序，并令 h 为 C中第一个高代价门。在传统的电路视图中，我们会得出结论：对于某些 i、$j \in [n]$，h 以 x_i（或 $\neg x_i$）和 x_j（或 $\neg x_j$）为输入。严格来说，这意味着 h 有两个参数节点，它们对应的项分别为 x_i（或 $\neg x_i$）和 x_j（或$\neg x_j$）。不过，在继续之前，我们先简化一下表述。我们通过定义简写（$\neg$）f来表示$\neg$门的可能性，其含义是“f（或 $\neg f$）”。如果 f ′有一个参数节点，其对应的项为（$\neg$）f，我们就说 f输入到 f'，并称 f'是 f的后继。最后，如果门 f 的标签为 $\wedge$ 或 $\vee$，我们就称 f 是高代价的。综合这些说法，我们可以改为说，对于某些 i、$j \in [n]$，h 是两个输入 x_i 和 x_j 的高代价后继——这样既保留了我们体系的形式化表述，又更贴近原证明。

我们断言 $i \neq j$。否则，对于某个 $\diamond \in \{\wedge, \vee\}$，$h \equiv (\neg)x_i \diamond (\neg)x_i$。若出现这种情况，我们就可以应用归一化规则或重言式规则；随后，重写 C 就会删除 h。由于 h 是一个高代价门，这会减小 C 的规模，与 C的最优性矛盾。

现在我们要说明，x_i 的扇出至少为 2，其中扇出定义为一个门或输入所拥有的高代价后继节点的数量。再次反设：h 是 x_i唯一的高代价后继节点。于是我们可以代入 $x_j = \alpha$，其中 α 的取值使得重写过程中我们可以对 h应用固定规则。这意味着 $C|_{x_j=\alpha}$ 不依赖于 x_i，从而违反事实 2（(¬)异或的所有子函数均不退化）。

令 f 为 x_i 的另一个高代价后继节点。我们可以得出，(¬)f 不是电路的输出（即根节点）。如果它是输出，那么我们就可以在重写过程中替换 $x_i = \beta$ 并固定 (¬)f。这会固定电路的输出，使 $C|x_i = \beta$ 成为常量，这与事实 1（(¬)异或是完全 DSR）矛盾。

23:24 收敛门消除与构造性电路下界

令 $f_\prime$ 为 f 的一个高代价后继节点，并注意 $h \neq f_\prime$，因为在我们的拓扑序中 $f < f'$，且 h 是序列中的第一个高代价门。现在我们注意到，如果我们设置 $x_i = \beta$，使得 f 被固定，那么在重写过程中，我们会消除 h （使用固定规则或传递规则）、f （使用固定规则）以及 f' （使用固定规则或传递规则）。这是因为一旦 f 被固定，固定后的 (¬)1 就会馈入 f'。在这种情况下，我们称 (¬)1 在应用此重写后作为后继节点“继承”了 f'，因此还会应用另一条重写规则来删除 f'。

因此，XOR$_n$ 所需的高代价门至少比 XOR$_{n-1}$多 3 个。由于 XOR$_1$ 不需要任何高代价门，根据归纳法可知，XOR$_n$ 至少需要 3$(n-1)$ 个。◄

C.1 Schnorr 的结构化证明

下面，我们采用 Lamport [24, 25]提出的结构化证明来证明定理 18。我们认为，这种呈现方式能让证明中复杂的分类讨论更加清晰易读。此外，这种证明风格也更便于使用计算机进行验证。

定理 18 的结构化证明。设 C 是 XOR$_n$ 的一个最优电路，其中 $n \geq 2$。1. 设 h 是 C 在拓扑序中的第一个与门或或门，因此 h 以 (¬)x_i、(¬)x_j 作为 i、$j \in \mathbb{N}$的输入。2. $i \neq j$a. 假设并非如此，则 $i = j$b. 那么 $h \equiv (\neg)x_i \diamond (\neg)x_i$，其中 $\diamond \in \{\wedge, \vee\}$ c. 因此，门消除 TGRS 中的一条归一化规则或重言式规则适用于 hd. 重写 C 后发现 h 已删除 e. 这与 C的最优性矛盾

3. .....的扇出 x_i 至少为 2。a. 反设不然，则 x_i 的扇出为 1。b. **将** $x_j = \alpha$ 代换到 C 中，以固定 hc. **重写** C，发现 x_i 的扇出现在为 0d. 因此，$C\mid_{x_j=\alpha}$ 不依赖于 x_ie. 这与事实 2 矛盾。

4. 令 f 为另一个以 x_i 为输入的门，因此 $f \neq h$5. (¬)f 不是 C的输出门。 a. 假设它是，则 (¬)f 是 C的输出门。 b. 将 $x_i = \alpha$ 代换到 C 中，以固定 fc. 重写 C，发现 C 的输出是常数 d. 因此，$C\mid_{x_i=\alpha}$ 是一个常函数 e. 这与事实 1 矛盾。 6. 令 f' 为 C中 f 的一个高代价后继节点，这样的节点必然存在。 7. 通过一次代换消去三个不同的门。 a. 将 $x_i = \alpha$ 代换到 C 中，以固定 fb. 重写 C，发现至少删除了 f、h、f' c. 论证：注意到 x_i “接触”了要消去的门 h，并且它固定了 f，而 f “接触”了门 f'd. 存在一种 1 位限制，可以消去 ≥ 3 个门 8. 得出结论：XOR$_n$ 至少比 XOR$_{n-1}$多需要 3 个高代价门。 9. 注意到 XOR$_1$ 不需要高代价门。 10. 用归纳法证明 XOR$_n$ 至少需要 3$(n-1)$。

◄