arXiv:2301.07472v1 [cs.LO] 18 Jan 2023

逻辑程序结构难度的刻画：是什么让环和可达性问题在树宽参数下变得困难?

Markus Hecher

2023年1月19日

摘要

答案集编程（ASP）是一种问题建模与求解框架，可用于知识表示领域中的多种问题，其工业应用日益广泛。多年来，答案集编程也一直吸引着研究计算复杂性及深入探究困难性及其来源的研究者。这些研究取得了丰硕成果，包括以复杂性类为基础的刻画、以二分式结果形式呈现的细粒度洞见，以及详尽的参数化复杂性图景。最近，一项新结果表明，对于能够刻画程序结构密度的树宽这一度量，求解著名的正规程序类预计会比判定可满足性（SAT）稍难。然而，如何利用答案集编程的这种结构能力尚不明确。本文提出了一种从 SAT 到正规答案集编程的新归约，超越了广为人知的编码：我们明确利用答案集编程的结构能力，以次线性的幅度降低树宽，而这种降低可能无法得到显著改进。与已有结果相比，这一归约以细粒度方式刻画了困难性，确立了依赖图的环长度（强连通分量大小）与树宽之间所需的函数依赖关系。

1 引言

答案集编程（答案集编程）Brewka、Eiter 和 Truszczy´nski（2011年）；Gebser 等（2012）是一种用于知识表示与推理以及一般人工智能领域的声明式问题建模与求解框架。因此，答案集编程成为该领域有效解决问题的关键形式化方法和合适的目标语言，例如 Balduccini、Gelfond 和 Nogueira（2006年）；Niemel¨a、Simons 和 Soininen（1999）；Nogueira 等（2001）；Guzi- olowski 等（2013年）；Schaub 和 Woltran（2018）；Abels 等（2019年）。这类问题由此被编码为逻辑程序，即一组通过所谓的答案集描述其解的规则——这种方法不止于像 SAT 中那样满足一组子句（规则），还要求为被认定成立的变量（原子）提供论证依据。人们投入了大量精力来提供各种扩展和丰富的建模语言，使其能够由 clasp 这类求解器高效求解（Gebser、Kaufmann 和 Schaub，2009年），或由 WASP 求解器求解（Alviano 等，2019年）。

例 1（使用答案集编程进行编码）。对公式 F 的可满足性（SAT）进行编码的经典方法是：对于每个变量 $v \in \text{at}(F)$，通过两条规则 $v \leftarrow \neg\hat{}v$ 和 $\hat{}v \leftarrow \neg v$猜测其真值。然后，对于 F中的每个子句 $l_1 \vee l_2 \vee \ldots \vee l_n$，添加一个约束来确保该子句：$\bot \leftarrow l_1$、$l_2$、.....、$l_n$ 中，若 $l_i = \neg v_i$，则 l_i 为 v_i，$\hat{}$否则为 l_i。

答案集编程的计算复杂性已得到相当充分的研究。对于一致性问题，即判定所谓的正规逻辑程序 是否存在答案集，该问题是 NP 完全的 Bido´ıt and Froidevaux (1991); Marek and Truszczy´nski (1991)。这一结果也适用于头循环无关（*HCF*）程序 Ben-Eliyahu 和 Dechter (1994)，但如果规则头中允许完全析取，复杂性就会提升至 Σ_2^P-完全 Eiter 和 Gottlob (1995)。随着时间推移，对答案集编程复杂性的研究受到越来越多的关注。针对答案集编程，已有大量更细致的研究 Truszczynski (2011)，也包括参数化复杂性方面的研究 Cygan

等（2015）；Niedermeier（2006）；Downey 和 Fellows（2013）；Flum 和 Grohe（2006），其中考虑了某些参数 Lackner 和 Pfandler（2012）。在参数化复杂性中，问题的“困难性”根据解决该问题所需的工作量来分类，例如，以某个参数为依据的运行时间依赖。对于答案集编程，针对广受研究且重要的结构参数树宽，相关研究日益增多 Jakl, Pichler, and Woltran (2009); Bichler, Morak, and Woltran (2020); Bliem et al. (2020)。直观而言，树宽会导出一种树分解，这是一种结构表示，可用于分块求解众多组合困难问题；树宽表示问题求解过程中需要考察的这些部分所包含的变量数上限。

近期的研究表明，在假设指数时间假设（*ETH*）成立的情况下——Impagliazzo、Paturi 和 Zane（2001）提出的这一假设意味着，SAT 无法 在最坏情况下于优于关于变量数量的单指数时间内求解——正规答案集编程在树宽方面似乎比 SAT 稍难 Hecher (2022)。更准确地说，(i) 对于树宽为 k、含有 n 个变量（原子）的任意逻辑程序，正规答案集编程都可以在 $2^{\mathcal{O}(k\cdot\log(k))}\cdot\text{poly}(n)$ 时间内求解；而在指数时间假设下，这种对树宽的依赖无法显著改善。困难性的根源在于程序依赖图中非常大的环（强连通分量，*SCC*）；困难性证明要求环长不受树宽限制，也就是说，环中涉及的原子数量与实例规模同阶。有趣的是，这与 (ii) SAT 形成鲜明对比，SAT 可以在 $2^{\mathcal{O}(k)}\cdot\text{poly}(n)$ 时间内判定。SAT 到答案集编程的经典归约在保持树宽的同时，不会在编码中引入任何环（见例 1）。因此，问题在于：能否构造含环的程序来归约 SAT，同时降低树宽？更具体地说，本文探讨：

- 如何在（正规）答案集编程中编码 SAT，从而使树宽降低的幅度反映（i）与（ii）之间的运行时间差异?
- 鉴于 (i) 中环长度无界，而 (ii) 中没有环，正规答案集编程与 SAT 的复杂性在环长度（强连通分量大小）方面有何差异？我们能否对树宽中的环长度设定一个界限，同时仍使正规答案集编程保持困难?
- 我们能否进一步推导其他片段或答案集编程相关扩展的运行时间结论和下界?

贡献。 我们通过一种新颖的归约来回答这些问题，它缩小了现有复杂性结果与下界之间的差距。具体而言，我们给出以下结果。

- 首先，我们建立了一种从 SAT 到正规答案集编程的新颖归约。与现有转换不同，该归约充分利用了沿环的可达性，从而将树宽从 k 降低至 $\mathcal{O}(\frac{k}{\log(k)})$。除非指数时间假设不成立，否则预计无法显著改进这一归约，即不太可能进一步大幅降低树宽。据我们所知，这是首个充分利用正规答案集编程结构能力的归约。随后，我们还研究了所构造程序的依赖图中的最大环（强连通分量大小）。
- 有趣的是，所构造程序的依赖图中的环（强连通分量大小）至多为 $2^{\mathcal{O}(k\cdot\log(k))}$。与近期困难性结果中的最大强连通分量大小相比，这是一个重大改进；后者在树宽方面没有上界。随后，我们证明，对于 ι-紧程序，即使将强连通分量大小进一步降至 $2^{\mathcal{O}(k\cdot\log(\iota))}$，仍能保持树宽方面的困难性。
- 最后，我们展示了我们的归约在计算复杂性方面还有直接的推论。对于 ι-紧程序这一类，我们证明了相应的指数时间假设紧下界。此外，相较于计算析取程序的答案集数量，计算正规程序在某个感兴趣投影下的答案集数量，预计会稍微困难一些。值得注意的是，这两个问题在同一个（经典）复杂性类中都是完备的，但就树宽而言，它们的困难程度却出人意料地不同。

相关工作。 有界偶环或奇环的程序已由 Lin 和 Zhao（2004）进行分析。此外，反馈宽度也得到了研究；它取决于打破大型强连通分量所需的原子数量（Gottlob, Scarcello, and Sideri（2002））。针对所谓的 ι-紧程序，已有改进（Fandinno and Hecher（2021））

其中 ι 小于树宽 k，从而允许运行时间达到 $2^{\mathcal{O}(k\cdot\log(\iota))}\cdot\text{poly}(n)$。对于正规程序和 HCF 程序，已有用于求解一致性、关于树宽具有略超指数复杂度的算法（Fichte and Hecher（2019））。对于析取 ASP，已有算法（Jakl, Pichler, and Woltran（2009）；Pichler et al.（2014）），其运行时间对实例大小呈线性，但对树宽呈双指数。其他问题的困难性也通过运行时间对树宽的依赖关系进行研究，例如指数级别；对此已知存在三重指数算法（Marx and Mitsou（2016）；Fichte, Hecher, and Pfandler（2020））。

已知许多从答案集编程到 SAT 的归约 Clark (1977); Ben-Eliyahu and Dechter (1994); Lin and Zhao (2003); Janhunen (2006); Alviano and Dodaro (2016); Bomanson and Janhunen (2013); Bo- manson (2017)。这些归约关注所得公式的规模和辅助变量的数量，而次二次级的膨胀不可避免 Lifschitz and Razborov (2006)。除非指数时间假设不成立，否则树宽也无法避免次二次级的膨胀 Hecher (2022)。对于 SAT，关于归结宽度和树宽的实证结果 Atserias, Fichte, and Thurley (2011) 表明，在树宽较小的实例上，SAT 求解器能够高效运行。

2 预备知识

我们假设读者熟悉图论术语，参见 Dies- tel (2012)。令 $G=(V,\ E)$ 为一个有向图。那么，$C\subseteq V$ 中的顶点集合 G 是 G 的强连通分量（*SCC*），当且仅当 C 是 $\subseteq$ 最大的集合，并且对于 C 中任意两个不同的顶点 $u,\ v$，都存在一条从 u 到 v 的有向路径，该路径位于 G中。

树分解（TD）。给定图 G=(*V*, *E*)，其树分解（TD）Robertson and Seymour (1986) 是一对 $\mathcal{T}$=(*T*, χ)，其中 T 是一棵以 root(*T*) 为根的树，且 χ 为 T 的每个节点 t 指定一个集合 $\chi(t)\subseteq V$，称为包，并满足：(i) $V=\bigcup_t$ 属于 $_T\chi$(*t*)；(ii) $E\subseteq\{\{u,v\}\mid t$在 T中，$\{u,v\}\subseteq\chi(t)\}$；(iii) 对于 T中的每个 r,s,t，若 s位于从 r到*t*的路径上，则有 $\chi(r)\cap\chi(t)\subseteq\chi(s)$。对于 T中的每个节点 t，我们用 chldr(*t*) 表示 T中 t 的子节点集合。令 width($\mathcal{T}$) := max$_t$ of $_T|\chi(t)|-1$。G 的树宽 tw (G) 是其所有树分解 $\mathcal{T}$ 中 width($\mathcal{T}$) 的最小值

G。对于 T中的节点 t，若 t 没有子节点，则称 type(*t*) 为叶节点；若 t恰有两个子节点 t' 和 t''，且满足 $t'\neq t''$，则称为合并节点；若 t 只有一个子节点，则称为内部节点。如果对于 T中的每个节点*t*，type(*t*) $\in$ {叶节点、合并节点、内部节点}，则称该树分解为良好树分解。树分解可以在线性时间内转化为良好树分解 Kloks (1994)[引理 13.1.3] 且不增加宽度。不失一般性，我们假设良好树分解的袋彼此不同。

答案集编程 (ASP)。我们假设读者熟悉命题可满足性（SAT）Biere等人（2009）；Kleine B¨uning和 Lettman（1999），其中子句、公式和赋值均采用通常含义。若对于每个 $x\in X\cap X'$，都有 $I(x)=I'(x)$，则两个赋值 $I\colon\ X\to\{0,1\}$, $I'\colon\ X'\to\{0,1\}$ 是相容的。

我们遵循命题答案集编程的标准定义 Brewka、Eiter 和 Truszczy´nski（2011）；Janhunen 和 Niemel¨a（2016）。设 ℓ, m, n 为非负整数，使得 $\ell\leq m\leq n$, $a_1, \ldots, a_n$ 是互不相同的命题原子。此外，我们将原子或其否定称为文字。程序 Π 是由以下形式的规则组成的集合：$a_1\vee\cdots\vee a_\ell\leftarrow a_{\ell+1},\ldots,a_m,\neg a_{m+1},\ldots,\neg a_n.$ 对于规则 r，令 $H_r:=\{a_1,\ldots,a_\ell\}$, $B_r^+:=\{a_{\ell+1},\ldots,a_m\}$, 且 $B_r^-:=\{a_{m+1},\ldots,a_n\}$。我们将规则 r 或程序 Π中出现的原子集合分别记作 at(*r*) := $H_r\cup B_r^+\cup B_r^-$和 at(Π) := $=\bigcup_{r\in\Pi}$at(*r*)。若对于每个 $r\in\Pi$都有 $|H_r|\leq 1$，则程序 Π 称为正规程序。程序 Π 的依赖图 D_Π 是定义在 $\bigcup_{r\in\Pi}H_r\cup B_r^+$中的原子上的有向图；对于每条规则 $r\in\Pi$，两个原子 $a\in B_r^+$和 $b\in H_r$ 之间由一条边(*a*, *b*)相连。

一个解释 I 是原子的集合。I 满足 *</style}*一条规则，当且仅当 () 或 。是 的模型*</style}*，如果它满足 的所有规则，符号表示为 。为简洁起见，我们将命题公式视为需要满足的公式集合（例如子句），并以类似方式使用解释、模型和可满足性的概念。在 下的 ***Gelfond-Lifschitz (GL)*** 化简 是程序 Π^I，它通过先从 Π 中删除所有满足 $B_r^-\cap I\neq\emptyset$ 的规则 r，再从剩余规则 r 中删除所有满足 $z\in B_r^-$ 的 $\neg z$ 而得到（Gelfond 和 Lifschitz，1991）。I 是程序 Π 的答案集，记作 $I\models\Pi$，当且仅当 I 是 Π^I 的极小模型。判定一个答案集编程程序是否有答案集的问题称为一致性，其为 Σ_2^{P}-完全问题（Eiter 和 Gottlob，1995）。若输入限于正规程序，其复杂性则降至

Marek 和 Truszczy´nski（1991年）提出的 NP 完全问题。

以下答案集的刻画常用于正规程序（Lin and Zhao, 2003）。设 $A \subseteq \text{at}(\Pi)$ 是原子集合。那么，函数 $\varphi : A \to \{0, \dots, |A|-1\}$ 是 $\text{dom}(\varphi) := A$ 上的一个序。设 I 是正规程序 Π 的模型，φ 是 I 上的一个序。若存在一条规则 $r \in \Pi$ 证明 a，则称原子 $a \in I$ 已被证明，其中 $a \in H_r$ 满足：(i) $B_r^+ \subseteq I$，(ii) $I \cap B_r^- = \emptyset$ 和 $I \cap (H_r \setminus \{a\}) = \emptyset$，以及 (iii) 对每个 $b \in B_r^+$，$\varphi(b) < \varphi(a)$。那么，I 是 Π 的答案集，当且仅当：(i) I 是 Π 的模型；(ii) I 已被证明，即每个 $a \in I$ 都已被证明。对于一个序 φ 以及两个原子 a、$b \in \in \text{at}(\Pi)$，当 b 直接紧随 a（即 $\varphi(b) = \varphi(a)+1$）时，我们记作 $a \prec_\varphi b$。$\text{dom}(\varphi)$ 为 $= \emptyset$ 的空序 φ 简记为 $\varnothing$。

原始图。我们需要图表示来对答案集编程应用树宽方法（Jakl, Pichler, and Woltran (2009)）。程序 Π 的原始图 $\mathcal{G}_\Pi$ 以 Π 的原子为顶点，并在满足以下条件时添加一条边：存在规则 $r \in \Pi$ 和 $a,\ b \in \text{at}(r)$。布尔公式 F（合取范式）的原始图 $\mathcal{G}_F$ 以 F 的变量为顶点，并在满足以下条件时用一条边连接两个顶点 $a,\ b$：F 中存在一个包含 $a,\ b$ 的子句。设 $\mathcal{T} = (T,\ \chi)$ 是 $\mathcal{G}_F$ 的一个树分解。那么，对 T 的每个节点 t，我们定义包子句 $F_t := \{c \in F \mid \text{at}(c) \subseteq \chi(t)\}$。

例 2。考虑公式 $F := \{c_1, c_2, c_3\}$，其中 $c_1 := (a \vee \neg b)$，$c_2 := (\neg a \vee c \vee d)$，$c_3 := (\neg c \vee \neg d)$。图 *1*（左）展示了原始图 $\mathcal{G}_F$，图 *1*（右）展示了 $\mathcal{G}_F$ 的一个树分解。然后注意到 $F_{t_1} = \{c_1\}$、$F_{t_2} = \{c_2, c_3\}$ 和 $F_{t_3} = \emptyset$。

ι-紧性。对于程序 Π 和原子 $a \in \text{at}(\Pi)$，我们用 $\text{scc}(a)$ 表示 D_Π 中原子 a 所在的强连通分量。因此，给定 $\mathcal{G}_\Pi$的树分解 $\mathcal{T} = (T, \chi)$，其紧性宽度为 $\max_{t \text{ of } T} \max_{x \in \chi(t)} |\chi(t) \cap \text{scc}(x)|$。$\Pi$ 的紧性树宽 ι 是 $\mathcal{O}(tw(\mathcal{G}_\Pi))$中所有宽度的树分解所对应的紧性宽度的最小值；此时我们称 Π为 ι-紧的。

命题 1(Fandinno and Hecher (2021))。假设一个正规、ι-紧的程序 Π；$\mathcal{G}_\Pi$ 的树宽为 k。那么，可以在 $2^{\mathcal{O}(k \cdot \log(\iota))} \cdot \text{poly}(|\text{at}(\Pi)|)$ 时间内判定 Π 的一致性。

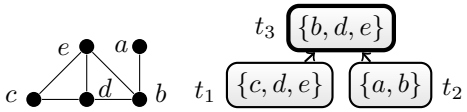

图 1：图 G（左）以及 G的树分解 $\mathcal{T}$（右）。

3 通过答案集编程降低 SAT 的树宽

本节将展示如何把一个布尔公式转换为逻辑程序，从而降低树宽，并明确利用答案集编程的结构能力。因此，与示例 1 中概述的标准转换不同，我们明确利用环以及答案集编程形式体系所提供的可达性能力。

下一小节中归约所实现的树宽具体降低幅度，将与以下观察密切相关；该观察表明，数 $k \in \mathbb{N}$ 的阶乘 $k!$ 有下界 $k^{\Omega(k)}$。

观察 1。令 $k \in \mathbb{N}$。则 $k!$属于 $2^{\Omega(k \cdot \log(k))}$。

证明。通过使用斯特林公式（例如，参见 Lokshtanov、Marx 和 Saurabh（2011年）），我们有 $\frac{k}{k^k} = e^{-\mathcal{O}(k)}!$。因此，我们推导出 $k!$ ~~对应于~~ $\frac{2^{k \cdot \log(k)}}{e^{\mathcal{O}(k)}} = \frac{2^{k \cdot \log(k)}}{2^{\log(e) \cdot \mathcal{O}(k)}} = 2^{k \cdot \log(k) - \log(e) \cdot \mathcal{O}(k)} = 2^{\Omega(k \cdot \log(k))}$。□

这一观察立即表明，$k!$ 与 $k^{\Theta(k)}$ 的量级相同，因为显然 $k!$ 属于 $k^{\mathcal{O}(k)}$。

借助可达性降低树宽

我们的主要归约思路 $\mathcal{R}$ 如下。我们取一个 F SAT 实例，也就是一个布尔公式和一个宽度为 k 的规范树分解 $\mathcal{T} = (T,\ \chi)$，其中 $\mathcal{G}_F$。然后，对于 T 中的每个节点，我们用 $k'!$ 种排序来模拟最多 2^k 个赋值，其中 k' 应当远小于 k。更准确地说，我们将树宽从 k 降至 k'，使得 $k'! \geq 2^k$。接着，由于 $k'!$ 属于 $2^{\Omega(k' \cdot \log(k'))}$（见观察 1），我们有 $2^{\Omega(k' \cdot \log(k'))}$ 至少为 2^k，因此 $k' = \mathcal{O}(\frac{k}{\log(k')}) = \mathcal{O}(\frac{k}{\log(k)})$。因此，我们的方法能够略微降低树宽，从而有效利用答案集编程和正环的能力来求解

结构开销更小的 SAT。虽然这看起来令人惊讶，但这与已知的答案集编程困难性结果相符；参见命题 1，其中 $\iota = k$。

形式上，我们通过取满足 $k'! \geq 2^k$ 的最小整数 k' 来确定 k'。对于 T 的每个节点 t，我们都定义这样一个值，其中 k'_t 是满足 $k'_t! \geq 2^{|\chi(t)|}$ 的最小整数。接着，我们定义一个由排序顶点组成的集合 V_t，其中包含 k'_t 个全新的顶点，这些顶点由包 $\chi(t)$ 唯一确定；也就是说，对于任意 TD节点 t、t'，当且仅当 $V_t = V_{t'}$ 时，$\chi(t) = \chi(t')$。利用集合 V_t，我们将元素 V_t 中至少 $2^{|\chi(t)|}$ 个排序构成的集合记为 ord(t)。此外，对于 T 的每个节点 t，我们将从排序的子集 $X \subseteq$ ord(t) 到赋值的双射映射记为 $\mathcal{I}_t$：$X \to 2^{\chi(t)}$。更准确地说，对于每个节点 t 和排序 $\varphi \in$ ord(t)，$\chi(t)$ 中原子上的 φ 对应的唯一赋值为 $\mathcal{I}_t(\varphi)$（若存在）。注意，$\mathcal{I}_t$ 是任意但固定的双射；也就是说，ord(t) 中某些未使用的排序可能没有定义。

示例 *3*。回忆示例 *2* 中的公式 F 和树分解 $\mathcal{T} = (T, \chi)$。根据定义，我们要求 $|V_{t_1}|! \geq 2^2$,$|V_{t_2}|! \geq 2^3$，并且$|V_{t_3}|! \geq 2$。因此，我们需要选择 $|V_{t_1}| = 3$, $|V_{t_2}| = 4$ 和 $|V_{t_3}| = 2$。于是，存在排序 $\alpha \in ord(t_1)$、$\beta \in ord(t_2)$，其中 $\mathcal{I}_{t_1}(\alpha)$ 和 $\mathcal{I}_{t_2}(\beta)$ 未定义。按照惯例，我们将 V_{t_1} 中的元素称为 v_1^j，将 V_{t_2} 中的元素称为 v_2^j，将 V_{t_3} 中的元素称为 v_3^j。

排序增强树分解。令 $\mathcal{T} = (T, \chi)$ 为 $\mathcal{G}_\Pi$ 的一个规范树分解。为了将 $k'!$ 个赋值彼此解耦，我们需要能够逐一访问任意模拟排序。为此，我们定义一种排序增强树分解。

定义 *1*. 设 F 是一个布尔公式，$\mathcal{T} = (T, \chi)$ 是 $\mathcal{G}_F$ 的良好树分解。接下来，我们按如下方式从 $\mathcal{T}$ 构造 $\mathcal{G}_F$ 的增广序树分解 $\mathcal{T}' = (T', \chi', \varphi, \psi)$，其中 (T', χ') 是一个树分解，φ, ψ 是从节点到序的映射。对于 T 中的每个 t，令 $\chi'(t) := \chi(t), \varphi_t := \varnothing, \psi_t := \varnothing$。对于 T 中任意两个相邻节点 t, t'，其中 $t' \in chldr(t)$ 且 $ord(t) \times ord(t') = \{(\alpha_1, \beta_1) \ldots, (\alpha_\ell, \beta_\ell)\}$，我们在 t 和 t' 之间添加一串新的节点 $t_1, \ldots, t_\ell$，使得对于每个 $1 \leq i \leq \ell$，我们定义 $\chi'(t_i) := \chi(t)$, $\varphi_{t_i} := \alpha_i$, 以及 $\psi_{t_i} := \beta_i$。对于 T 中满足 $ord(t) = \{\alpha_1, \ldots, \alpha_\ell\}$ 的每个叶节点 t，在 T' 中，我们在 t 下方串接复制节点 $t_1, \ldots, t_\ell$，其中对于每个 $1 \leq i \leq \ell$, $\chi'(t_i) := \chi(t)$，$\varphi_{t_i} := \alpha_i$, $\psi_{t_i} := \varnothing$。

请注意，该定义为逐一分析不同序下的赋值奠定了基础。此外，通过比较相邻 TD 节点的每一对排序，我们的归约之后将同步相邻的排序并确保兼容性。

例 4. 回顾前一个例子中的公式 F 和树分解 $\mathcal{T} = (T, \chi)$。随后，根据定义 *1*，可以将树分解 $\mathcal{T}$ 转化为带排序的树分解 $\mathcal{T}' = (T', \chi', \varphi, \psi)$。为此，我们在 t_1下添加一串子节点（路径）；这些节点各自处理 V_{t_1}上的一种排序 φ_{t_i}。对节点 t_2也进行类似操作。在 t_3 和 t_1 之间，我们还添加一条由新节点组成的路径，其中每个节点覆盖一组排序组合（一对排序）$\varphi_{t_3}, \psi_{t_1}$，这将在同步过程中发挥关键作用。同样地，我们也在 t_3 和 t_2之间进行此操作。

涉及的原子。为了在每个节点 t 的 T 个 |ord(t)| 种排序中进行猜测，我们通过形如 $e_{x,y}$ 的 用于表示排序边的原子来刻画每种排序 $\alpha \in$ ord(t)（以及其否定 $\hat{e}_{x,y}$），以指示对于任意两个不同的顶点 $x, y \in V_t$，是否满足 $x \prec_\alpha y$。此外，对于每个 $x \in V_t$，我们还要求使用形如 r_x 的原子，以记录 x 是否可达。对于 V_t 本身（它可能跨越多个树分解节点）以及 $y \in V_t$，我们需要一个额外的（源）可达性原子，形式为 $r_{s_{V_t}}$，以及边原子 $e_{s_{V_t}, y}$。然后，我们还使用（目标）可达性原子 $r_{d_{V_t}}$，并对每个 $y \in V_t$ 使用边原子$e_{y, d_{V_t}}$。

我们将检查，对于 T 的每个节点 t 中的每个 $x \in V_t$，包含在某个答案集中的出边至多有一条，即原子 $e_{x,y}$。为确保这一点，我们沿着树分解传递至多一条出边的信息；对于 T 的每个节点 t 以及 $x \in V_t$，我们使用一个辅助原子 o_t^x。对于源 s_{V_t} 也需要这样处理，即对于每个节点 t，我们也使用辅助原子 $o_t^{s_{V_t}}$。注意，如果 $V_t = V_{t'}$，这也包括节点 t' 的原子 $o_{t'}^{s_{V_t}}$。

为了比较排序，对于 T 的每个树分解节点 t，以及排序 α 中包含的满足 $\alpha \in \{\varphi_t, \psi_t\}$ 的元素 x，我们使用一个额外的 排序查询原子 q_t^x。如果该查询原子成立，则排序 α 至少从

排序 α 中包含的第一个原子开始，直到原子 x 为止都得到满足。直观地说，如果 q_t^y 对 α 中包含的最后一个原子 y 成立，我们最终就能确定 α 是否成立。

这些查询原子通过额外的 测试点 p_t^x 以及 初始测试点 p_ϵ^x得到支持。对于 T 中的每个节点 t，我们对其中每个 $\alpha \in\{\varphi_t, \psi_t\}$ 所包含的每个原子 x 都使用这些测试点。测试点使我们能够在每个节点中检查带查询的排序。对于 T 中的每个节点 t 和原子 $x \in V_t$，我们用 prev(x, t) 表示 x 前序 t 的测试点 。形式上，若存在某个 $t' \in$ chldr(t)，则 prev(x, t) := chldr(t)；否则为 $\{\epsilon\}$。

归约。 该归约 $\mathcal{R}$ 以一个布尔公式 F 和一个带排序的树分解 $\mathcal{T} = (T, \chi, \varphi, \psi)$ 为输入。在开始描述 $\mathcal{R}$ 之前，我们需要以下定义。对于 T 中的每个节点 t，令排序边 E_t 的集合 为其直接后继的任意固定子集。更准确地说，对于第一个元素 $a \in$ dom(φ_t) 和最后一个元素 $b \in$ dom(φ_t)，也就是说， a 没有 $\prec_{\varphi_t}$ 前驱，而 b 没有 $\prec_{\varphi_t}$ 后继，E_t 是满足 $E_t \subseteq \{e_{x,y} \mid x \prec_{\varphi_t} y\} \cup \{e_{s_{V_t},a}, e_{b,d_{V_t}}\}$ 的最大子集，使得对于 T 中任意满足 $t' \neq t$ 的节点 t'，都有 $E_t \cap E_{t'}$=∅。

例 5. 再次考虑上文的公式 F 以及经序扩充的树分解 $\mathcal{T}' = (T', \chi', \varphi, \psi)$。注意， E_t 的定义相当宽泛，但其本质要求是，每条序边恰好出现在 T'的一个节点中。例如，序 $\varphi_{t^*} = \{v_1^1 \mapsto 0, v_1^2 \mapsto 1, v_1^2 \mapsto 2\}$ 在下方的节点 t^* 中处理，该节点位于 t_1之下。我们可以设定 $E_{t^*} = \{e_{s_{V_{t_1}}, v_1^1}, e_{v_1^1, v_1^2}, e_{v_1^2, v_1^3}, e_{v_1^3, d_{V_{t_1}}}\}$。假设在 t^*的子节点 t' 中，采用另一种关于 V_{t_1} 的序 $\varphi_{t'} = \{v_1^1 \mapsto 0, v_1^3 \mapsto 1, v_1^2 \mapsto 2\}$ 。那么，考虑到 E_{t^*}，即避免边重叠，我们可以设定$E_{t'} = \{e_{v_1^1, v_1^3}, e_{v_1^3, v_1^2}, e_{v_1^2, d_{V_{t_1}}}\}$。同样，其余每条边也都唯一地归入某个节点。

总体而言，图 2 所示的归约 $\mathcal{R}$ 由五个块组成。第一个块，即公式 (1)–(7)，涉及排序的选择（通过边原子 $e_{y,x}$），并确保每个可达性原子 r_x 都成立。然后，第二个块，即公式 (8)–(10)，确保对于 T 中的每个节点 t，其每个节点 $y \in V_t$ 在答案集中至多有一条形如 $e_{y,x}$ 的出边。第三个块，即公式 (11)–(13)，确保测试点得到保留。这样，我们便能在第四个块，即公式 (14)–(16) 中恰当地定义排序查询。最后，最后一个块，即公式 (17)–(19)，确保所选排序彼此兼容，并且满足 F 中的每个子句。

块 1：选择序，公式 (1)–(7)。 第一个块涉及选择排序。注意，公式 (3) 的析取是头循环无关的，可以通过 Ben-Eliyahu 和 Dechter (1994) 的移位方法，简单地转换为正规规则。接着，公式 (1) 设置源顶点的可达性；对于 T 的每个节点 t，公式 (2) 确保 V_t 中所有顶点以及 V_t 的目标顶点均可达。公式 (3) 要求从每个可达顶点 y 选择出边（根据公式 (2)，至少一条），指向某个顶点 x。由此，公式 (4) 为 x 得到初始测试点。对于集合 V_t 的目标顶点，不需要这样的测试点，因此我们可以通过公式 (5) 立即得到可达性。测试点之间的连接和传播将由块 3 实现。最后，顶点 x 的最后一个测试点将导出 x 的可达性。若 x 未出现在 t' 的后继节点的排序中，则由公式 (6) 确保这一点；若 x 出现在根节点的排序中，则由公式 (7) 确保这一点。

块 2： ≤ 1 **出边，公式 (8)–(10)**。 此块确保每个顶点 y 至多有一条出边，其中 y 也可以是集合 V_t 中某个排序顶点的源顶点 s_{V_t}。公式 (8) 将 y 是否已决定一条通向某个节点的出边这一信息，从节点 t' 传递到其父节点 t。随后，每当在节点 t 中为 y 选定一条出边时，公式 (9) 要求 o_t^y 成立。最后，公式 (10) 禁止在节点 t 中为原子 y 选择出边（如果已在子节点 t' 中选过）。

第3块：传播测试点、公式（11）–（13）。第三块讨论在某些查询不成立时，如何传播测试点。对于空序，即 $\varphi_t = \varnothing$，在节点 t中，公式（11）直接为每个原子 $x \in V_t$ 传播测试点：依据的是每个子节点 t 中关于 x 的测试点证据（或者依据 p_ϵ^x，如果 prev(x, t) $= \{\epsilon\}$）。此外，只要 $\alpha = \varphi_t$ 或其前驱 $\alpha = \psi_t$ 的某个序关系 $y \prec_\alpha x$ 不成立，我们仍需推导出

相应的测试点，见公式（12），因为后续查询或最终推导可达性都需要这个测试点，参见公式（6）、（7）。对于 α的最后一个元素，情况也一样，见公式（13）。我们需要同时涵盖两种排序 φ_t 和 ψ_t，原因在于相邻排序需要兼容性，这一点将在下文第5块讨论。

第4块：定义排序查询、公式 (14)–(16)。这一块着重于推导查询原子，以确保某些排序成立。任意排序 $\alpha \in \{\varphi_t, \psi_t\}$ 的第一个元素均由前一个测试点推导得出，如公式 (14) 所示。然后，每当遇到 $y \prec_\alpha x$ 时，公式 (15) 便可推导出查询原子 q_t^x；它取决于 x 对应的前一个测试点以及 q_t^y。由此确保顺序 $y \prec_\alpha x$ 确实得以保留，这与上述公式 (12) 和 (13) 相反。最后，如果查询原子 q_t^x 成立，公式 (16) 会立即给出相应的测试点 p_t^x。

第5块：排序的兼容性与可满足性，公式 (17)–(19)。最后一块负责处理公式 F 中每个子句的兼容性和可满足性：排除那些对应赋值无法满足某个子句的排序。为此，公式 (17) 排除 φ_t 与 ψ_t 不兼容的情况；也就是说，为避免不一致性，禁止与 φ_t 的最后一个元素 x 以及 ψ_t 的最后一个元素 y 对应的查询原子 q_t^x，q_t^y 同时成立。最重要的是，如果与 φ_t 的最后一个元素 x 对应的查询原子 q_t^x 成立，公式 (18) 会确保排序 φ_t 对应的赋值满足 F_t 中的子句。最后，公式 (19) 用于避免可能采用未使用排序的边界情况，否则便可绕过可满足性检查。

示例6。 回顾公式 F、带排序的树分解$\mathcal{T}'$=$(T', \chi', \varphi, \psi)$，以及示例5中的 φ_{t^*} 和 E_{t^*}。我们简要概述为节点 t^*生成的规则。

---

(1) $r_{s_{V_{t^*}}} \leftarrow$

(2) $\leftarrow r_{v_1^1}$; $\leftarrow r_{v_1^2}$; $\leftarrow r_{v_1^3}$; $\leftarrow \neg r_{V_{d_{V_{t^*}}}}$

(3) $e_{s_{V_{t^*}},v_1^1} \vee \hat{e}_{s_{V_{t^*}},v_1^1} \leftarrow r_{s_{V_{t^*}}}$; $e_{v_1^1,v_1^2} \vee \hat{e}_{v_1^1,v_1^2} \leftarrow r_{v_1^1}$; $e_{v_1^2,v_1^3} \vee \hat{e}_{v_1^2,v_1^3} \leftarrow r_{v_1^2}$; $e_{v_1^3,d_{V_{t^*}}} \vee \hat{e}_{v_1^3,d_{V_{t^*}}} \leftarrow r_{v_1^3}$

(4) $p_\epsilon^{v_1^1} \leftarrow e_{s_{V_{t^*}},v_1^1}$; $p_\epsilon^{v_1^2} \leftarrow e_{v_1^1,v_1^2}$; $p_\epsilon^{v_1^3} \leftarrow e_{v_1^2,v_1^3}$

(5) $r_{v_1^3} \leftarrow e_{v_1^3,d_{V_{t^*}}}$

---

(8) $o_{t^*}^{v_1^1} \leftarrow o_{t'}^{v_1^1}$; $o_{t^*}^{v_1^2} \leftarrow o_{t'}^{v_1^2}$; $o_{t^*}^{v_1^3} \leftarrow o_{t'}^{v_1^3}$; $o_{t^*}^{d_{V_{t^*}}} \leftarrow o_{t'}^{d_{V_{t^*}}}$

(9) $o_{t^*}^{s_{V_{t^*}}} \leftarrow e_{s_{V_{t^*}},v_1^1}$; $o_{t^*}^{v_1^1} \leftarrow e_{v_1^1,v_1^2}$; $o_{t^*}^{v_1^2} \leftarrow e_{v_1^2,v_1^3}$; $o_{t^*}^{v_1^3} \leftarrow e_{v_1^3,d_{V_{t^*}}}$

(10) $\leftarrow o_{t'}^{s_{V_{t^*}}}, e_{s_{V_{t^*}},v_1^1}$; $\leftarrow o_{t'}^{v_1^1}, e_{v_1^1,v_1^2}$; $\leftarrow o_{t'}^{v_1^2}, e_{v_1^2,v_1^3}$; $\leftarrow o_{t'}^{v_1^3}, e_{v_1^3,d_{V_{t^*}}}$

---

(11) $p_{t^*}^{v_1^1} \leftarrow p_{t'}^{v_1^1}$; $p_{t^*}^{v_1^2} \leftarrow p_{t'}^{v_1^2}$; $p_{t^*}^{v_1^3} \leftarrow p_{t'}^{v_1^3}$

(12) $p_{t^*}^{v_1^2} \leftarrow p_{t'}^{v_1^2}, \neg q_{t^*}^{v_1^1}$; $p_{t^*}^{v_1^3} \leftarrow p_{t'}^{v_1^3}, \neg q_{t^*}^{v_1^2}$

(13) $p_{t^*}^{v_1^3} \leftarrow p_{t'}^{v_1^3}, \neg q_{t^*}^{v_1^3}$

---

(14) $q_{t^*}^{v_1^1} \leftarrow p_{t'}^{v_1^1}$

(15) $q_{t^*}^{v_1^2} \leftarrow p_{t'}^{v_1^2}, q_{t^*}^{v_1^1}$; $q_{t^*}^{v_1^3} \leftarrow p_{t'}^{v_1^3}, q_{t^*}^{v_1^2}$

(16) $p_{t^*}^{v_1^1} \leftarrow q_{t^*}^{v_1^1}$; $p_{t^*}^{v_1^2} \leftarrow q_{t^*}^{v_1^2}$; $p_{t^*}^{v_1^3} \leftarrow q_{t^*}^{v_1^3}$

注意，如果 $\mathcal{I}_{t^*}(\varphi_{t^*}) \not\models F_{t^*}$ 或$\mathcal{I}_{t^*}(\varphi_{t^*})$ 未定义（未使用的序），公式 *(18)* 和 *(19)* 会生成 $\leftarrow q_{t^*}^{v_1^3}$。

归约的性质与后果

接下来，我们将证明该归约确实利用了结构参数树宽，即树宽有所降低。

定理 1（树宽感知）。从布尔公式 F 和带排序的树分解 $\mathcal{T}$=（T、χ、φ、ψ）到正规程序 Π 的归约由公式 *(1)* 至 *(19)* 组成，并会略微降低树宽。准确地说，如果 k 是 $\mathcal{T}$的宽度，那么 $\mathcal{G}_\Pi$ 的树宽属于 $\mathcal{O}(k^{k^{\log(k)}})$。

证明（概要）。我们构造一个 $\mathcal{G}_\Pi$的树分解 $\mathcal{T}' = (T, \chi')$，以表明 $\mathcal{T}'$ 的宽度仅略有增加（相较于 k）。为此，设 t 是 T的一个节点，其子节点为 $= \langle t_1, \ldots, t_\ell \rangle$，并设 $\hat{t}$ 为 t 的父节点（若存在）。我们定义 (i) $R(t) := \{r_x \mid x \in V_t\} \cup \{r_x \mid x \in V_{t'}, t' \in \text{chldr}(t)\} \cup \{r_{s_{V_t}}, r_{d_{V_t}}\}$,

(ii) $E(t) := \{e_{x,y}, \hat{e}_{x,y} \mid e_{x,y} \in E_t\}$, (iii) $P(t) := \{q_t^x, p_t^x, p_{t'}^x, p_\epsilon^x \mid x \in \mathsf{dom}(\varphi_t) \cup \mathsf{dom}(\psi_t)\ t' \in \mathsf{prev}(x,t)\}$, 以及 (iv) $O(t) := \{o_t^x, o_t^{s_{V_t}} \mid x \in V_t\} \cup \{o_{t'}^x, o_{t'}^{s_{V_{t'}}} \mid t' \in \mathsf{chldr}(t)\ x \in V_{t'} \cap V_t\}$。接着，我们令$\chi'(t) := R(t) \cup E(t) \cup P(t) \cup O(t)$。注意，$\mathcal{T}'$ 是 $\mathcal{G}_\Pi$ 的一个树分解，并且根据构造，$|\chi'(t)|$ 属于 $\mathcal{O}(|V_t|)$。根据 V_t 的定义，$|V_t| \leq k'$，其中 $k'! \geq 2^k$。接着，由于 $k'!$ 属于 $2^{\Omega(k' \cdot \log(k'))}$（参见观察 1），我们有 $2^{\Omega(k' \cdot \log(k'))}$ 至少为 2^k，因此 $k' = \mathcal{O}(\frac{k}{\log(k')}) = \mathcal{O}(\frac{k}{\log(k) - \log(\log(k'))}) = \mathcal{O}(\frac{k}{\log(k)})$。□

有趣的是，我们的归约无法得到显著改进，这使得进一步降低树宽的可能性很小。

定理 2（树宽下降是最优的）。假设存在一个将公式 F 归约为正规程序 Π 的归约，其运行时间为 $2^{o(k)} \cdot \mathrm{poly}(|\mathsf{at}(F)|)$，其中 $k = tw(\mathcal{G}_F)$。那么，除非指数时间假设不成立，否则其树宽不可能属于 $o(\frac{k}{\log(k)})$。

证明。反设这样的归约存在，记为 $\mathcal{R}^*$。于是，我们对任意 F 应用该归约，得到程序 $\Pi = \mathcal{R}^*(F)$。然后，根据 Hecher (2022)，我们知道 Π 可以在 $2^{\mathcal{O}(k' \cdot \log(k'))} \cdot \mathrm{poly}(|\mathsf{at}(F)|)$ 时间内判定，其中 k' 属于 $o(\frac{k}{\log(k)})$。因此，我们有 Π，从而 F 可以在 $2^{o(\frac{k}{\log(k)} \cdot \log(k))} \cdot \mathrm{poly}(|\mathsf{at}(F)|)$ 时间内判定，而该时间属于 $2^{o(k)} \cdot \mathrm{poly}(|\mathsf{at}(F)|)$，这与指数时间假设相矛盾。□

正确性证明，公式 (1) 至 (19) 所编码的归约确实刻画了布尔公式的满足赋值。

定理 3 (正确性)。从布尔公式 F 和其树分解 $\mathcal{T} = (T, \chi)$ 到逻辑程序 Π 的归约由公式 *(1)* 至 *(19)* 构成，且该归约是正确的。具体而言，对于 F 的每个模型，都存在 Π 的一个答案集。反过来，对于 Π 的每个答案集，都存在 F 的唯一模型。

例 7. 图 *3* 展示了块 *1*、*3* 和 *4* 中规则在两个简单 *TD* 节点上的依赖图；该图说明，不兼容的查询会导致未经证明的循环依赖，因此不会发生这种情况。

4 关于答案集编程困难性的进一步见解

本节将深入探讨正规逻辑程序困难性的刻画。首先，我们讨论最大强连通分量长度所带来的影响。

无界环（强连通分量）对困难性至关重要吗?

根据第 3 节中的归约构造，并注意到 $\mathcal{R}$ 会在程序的依赖图中导致大小为 $2^{\mathcal{O}(k \cdot \log(k))}$的环（强连通分量），我们得到了关于判定正规程序一致性的新困难性结果及其精确下界。

推论 1 (最大强连通分量下界)。设 Π 是一个正规逻辑程序，其中 $\mathcal{G}_\Pi$ 的树宽为 k，且 D_Π 的最大强连通分量大小为 $2^{\mathcal{O}(k \cdot \log(k))}$。那么，除非指数时间假设不成立，否则无法在时间 $2^{o(k \cdot \log(k))} \cdot \mathrm{poly}(|\mathsf{次}(\Pi)|)$ 内判定 Π 的一致性。

推论 1 揭示了困难性所需的强连通分量大小，这与已知下界形成对比；在树宽方面，Hecher (2022) 的结果无法界定环长度或强连通分量大小。有趣的是，该推论与命题 1 中相应的上界（针对 $\iota = k$）一致。我们预计推论 1 无法得到显著加强，但下文将展示，对于 ι-紧程序，可以将强连通分量大小降至 $2^{o(k \cdot \log(\iota))}$。

这也给出了投影答案集计数问题的相应下界，而这一问题此前在 Fichte and Hecher (2019) 中仍未解决。我们的归约使我们能够弥合与上界之间的差距，表明析取程序上的该问题预计比普通计数更难。

定理 4 (投影计数下界)。设 Π 是一个正规逻辑程序，$P \subseteq \mathsf{at}(\Pi)$ 是原子，k 是 $\mathcal{G}_\Pi$ 的树宽，且 D_Π 的最大强连通分量大小属于 $2^{\mathcal{O}(k \cdot \log(k))}$。那么，在指数时间假设成立的条件下，无法在时间 $2^{2^{o(k \cdot \log(k))}} \cdot \mathrm{poly}(|\mathsf{at}(\Pi)|)$ 内计算基数 $|\{M \cap P \mid M \models \Pi\}|$。

ι-紧程序的一个指数时间假设紧下界

回顾一下 ι-紧程序这一片段，其动机是提供几乎紧的程序，使其比正规程序更容易求解。每个公式 F 都可以编译成一个（1-）紧程序，参见例 1。

因此，在下文中，我们将上一节的归约 $\mathcal{R}$ 从 $\iota = k$ 推广到情形 $\iota \geq 2$，从而得到 $\mathcal{R}'$。直观而言，这一归约 $\mathcal{R}'$ 使我们能够降低树宽 k，但不一定能将其降至正规程序树宽的最大值 $\mathcal{O}(k \log(k))$。相反，ι 为紧致性提供了精确的控制手段，从而将树宽降至 $\mathcal{O}(k \log(\iota))$。

改造后的归约。我们改造归约 $\mathcal{R}$ 的构造，得到 $\mathcal{R}'$。为此，我们取 SAT 的一个实例 F，即一个布尔公式，以及其 $\mathcal{G}_F$ 的一个增添序的树分解 $\mathcal{T} = (T, \chi, \varphi, \psi)$，其宽度为 k。然后，对 T 的每个节点，我们用至多 2^k 个排序来模拟至多 2^k 个赋值，其中排序数量为 $(\iota!)^{k'}_{\iota}$，这里 ι 是任意固定的 $2 \leq \iota \leq k'$。因此，我们将树宽从 k 降至 k'，使得 $(\iota!)^{k'}_{\iota} \geq 2^k$；上一节的特殊情形对应于 $\iota = k'$。因此，由于每个包中仍有至多 k' 个元素，而我们只有 ι 个位置，所以每个包需要 k'_{ι} 个强连通分量。结果，节点 t 的排序 ord(t) 并非全序。因此，对于排序集合 V_t，我们不再只需要一个源顶点和一个目标顶点，而是需要至多 k'_{ι} 个形如 $r_{s^j_{V_t}}, r_{d^j_{V_t}}$ 的可达性原子，对每个 $1 \leq j \leq k'_{\iota}$ 都如此。这需要对 E_t 的定义、公式 (1)、(2)、(5)，以及 (8) 和 (10) 做一些小幅调整，附录中对此作了概述。

可以证明观察 1 的一个推广，其中 $\iota < k$。

观察 2. 设 $2 \leq \iota \leq k$。那么，$(\iota!)^{k}_{\iota}$ 属于 $2^{\Omega(k \cdot \log(\iota))}$。

根据观察 2，$(\iota!)^{k'}_{\iota}$ 属于 $2^{\Omega(k' \cdot \log(\iota))}$，因此我们有 $2^{\Omega(k' \cdot \log(\iota))}$ 至少为 2^k，从而 $k' = \mathcal{O}(k \log(\iota))$。因此，$\mathcal{R}'$ 仅将树宽略微降低至 $\mathcal{O}(k \log(\iota))$。所以，对于命题 1 中给出的结果，它竟然能够得到显著改进（在指数时间假设下）实属出乎意料。更准确地说，我们得到如下下界结果。

定理 5 (下界 ι-紧致性)。设 Π 是一个 ι-紧致逻辑程序，其中 $\mathcal{G}_\Pi$ 的树宽为 k，且 D_Π 的最大强连通分量大小属于 $2^{\mathcal{O}(k \cdot \log(\iota))}$。那么，在指数时间假设下，无法在 $2^{o(k \cdot \log(\iota))} \cdot \text{poly}(|at(\Pi)|)$时间内判定 Π 的一致性。

5 讨论与结论

答案集编程的复杂性已从不同方面和角度进行了研究。最近的研究表明，在指数时间假设（ETH）成立的前提下，正规逻辑程序的求值在树宽这一结构参数方面，预计会比判定布尔公式的可满足性（SAT）稍难一些。然而，困难性证明依赖于树宽无界的大环（强连通分量）。此外，与标准归约（见例 1）相比，正规答案集编程的“困难性”尚不明晰。

本文解决了这两个问题。我们的思路是从 SAT 归约到正规答案集编程，从而主动降低树宽所体现的结构依赖。我们设计了这样一种归约，将树宽从 降低到$k \log(k)$。我们证明，在 ETH 成立的前提下，这种降低不可能得到显著改进。进一步地，借助该归约，我们改进了正规程序和树宽的现有困难性结果：所构造的环（强连通分量）不必在树宽上无界；事实上，即使树宽仅有单指数上界，困难性仍然成立。随后，我们进一步改进了 ι-紧程序这一类的上界，使我们得以弥合差距，达到已知上界；我们的结果表明，该上界是指数时间假设紧的。

最后，我们应用该归约，为正规逻辑程序建立更多指数时间假设紧的下界。我们希望，本文中的归约能够进一步揭示正规逻辑程序相关问题的困难性，并带来新的见解。鉴于 Atserias、Fichte 和 Thurley（2011年）关于树宽与分辨率宽度对应关系的已知结果（该结果应用于 SAT 求解），这或许能为答案集编程领域的相关见解铺平道路。目前，我们正在比较从 SAT 到答案集编程的不同归约，并研究它们在实践中的表现。鉴于将有向度量用于答案集编程并未取得成功（Bliem、Ordyniak 和 Woltran，2016年），介于树宽和有向变体之间的结构参数

蚂蚁可能会带来新的见解。

第1块：排序与可达性
$r_{s_{V_t}} \leftarrow$ 对于每个 t 在 T (1)$\leftarrow \neg r_y$ 对于每个 t 在 $T, y \in V_t \cup \{d_{V_t}\}$ (2)$e_{y,x} \vee \hat{} e_{y,x} \leftarrow r_y$ 对于每个 t 在 T,$e_{y,x} \in E_t$ (3)$p^x_\epsilon \leftarrow e_{y,x}$ 对于每个 t 在 $T, e_{y,x} \in E_t, \{x, y\} \subseteq V_t$ (4)$r_{d_{V_t}} \leftarrow e_{y,d_{V_t}}$ 对于每个 t 在 T,$e_{y,d_{V_t}} \in E_t$ (5) $r_x \leftarrow p^x_{t'}$ 对于每个 t 在 $T, \varphi_t = \varnothing, t' \in \mathsf{chldr}(t)$ $x \in \mathsf{dom}(\psi_{t'}) \setminus \mathsf{dom}(\varphi_{t'})$ (6)$r_x \leftarrow p^x_{\mathrm{root}(T)}$ 对于每个 $x \in V_{\mathrm{root}(T)}$ (7) 第2块：≤ 1 出边$o^y_t \leftarrow o^y_{t'}$ 对于每个 t 在 $T, t' \in \mathsf{chldr}(t)$ $y \in (V_t \cap V_{t'}) \cup \{s_{V_t} \mid V_t = V_{t'}\}$ (8) $o^y_t \leftarrow e_{y,x}$ 对于每个 t 在 T,$e_{y,x} \in E_t, y \in V_t$ (9)$\leftarrow o^y_{t'}$,$e_{y,x}$ 对于每个 t 在 $T, e_{y,x} \in E_t, t' \in \mathsf{chldr}(t)$ $y \in V_{t'} \cup \{s_{V_t} \mid V_t = V_{t'}\}$ (10) 第3块：测试点$p^x_t \leftarrow p^x_{t_1}, \ldots, p^x_{t_o}$ 对于每个 t 在 $T, x \in V_t, \varphi_t = \varnothing, \mathsf{prev}(x, t) = \{t_1, \ldots, t_o\}$, (11)$p^x_t \leftarrow p^x_{t'}, \neg q^y_t$ 对于每个 t 在 $T, \alpha \in \{\varphi_t, \psi_t\}, \mathsf{prev}(x, t) = \{t'\}, \{x, y\} \subseteq \mathsf{dom}(\alpha)$ $y \prec_\alpha x$ (12)$p^x_t \leftarrow p^x_{t'}, \neg q^x_t$对于每个 t 在 $T, \alpha \in \{\varphi_t, \psi_t\}, \mathsf{prev}(x, t) = \{t'\}, x$ 没有 $\prec_\alpha$后继 (13) 第4块：排序查询 $q^x_t \leftarrow p^x_{t'}$ 对于每个 t 在 $T, \mathsf{prev}(x, t) = \{t'\}, \alpha \in \{\varphi_t, \psi_t\}, x \in \mathsf{dom}(\alpha)$ x 没有 $\prec_\alpha$前驱 (14)$q^x_t \leftarrow p^x_{t'}, q^y_t$ 对于每个 t 在 $T, \alpha \in \{\varphi_t, \psi_t\}, \mathsf{prev}(x, t) = \{t'\}, \{x, y\} \subseteq \mathsf{dom}(\alpha)$ $y \prec_\alpha x$ (15)$p^x_t \leftarrow q^x_t$ 对于每个 t 在 T, $\alpha \in \{\varphi_t, \psi_t\}, x \in \mathsf{dom}(\alpha)$ (16) 第5块：兼容性与 SAT$\leftarrow q^x_t, q^y_t$ 对于每个 t 在 $T, \alpha \in \{\varphi_t, \psi_t\}, x \in \mathsf{dom}(\alpha)$ 没有 $\prec_\alpha$后继，(17)$y \in \mathsf{dom}(\psi_t)$ 没有 $\prec_{\psi_t}$ 后继 $\mathcal{I}_t(\varphi_t)$ 和 $\mathcal{I}_t(\psi_t)$ 不兼容$\leftarrow q^x_t$对于每个 t 在 $T, x \in \mathsf{dom}(\varphi_t)$ 没有 $\prec_{\varphi_t}$后继 $\mathcal{I}_t(\varphi_t) \not\models F_t$ (18)$\leftarrow q^x_t$对于每个 t 在 $T, x \in \mathsf{dom}(\varphi_t)$ 没有 $\prec_{\varphi_t}$后继 $\mathcal{I}_t(\varphi_t)$ 未定义 (19)

图 2：归约 $\mathcal{R}$ 将公式 F 以及与之对应的带排序的树分解 $\mathcal{T} = (T, \chi, \varphi, \psi)$ 映射到 $\mathcal{G}_F$。

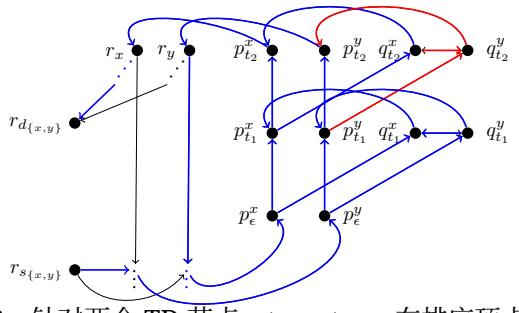

图 3：针对两个 TD 节点 t_1、t_2，在排序顶点 $V_{t_1}{=}V_{t_2}{=}\{x,y\}$ 上进行归约所得的正依赖图。这些块可能导致环（更大的强连通分量）。蓝色高亮部分是按推导顺序排列的原子，假设 y 排在 x 之前，即 $y \prec_{\varphi_{t_1}} x$。红色高亮的边不能被采用，即 q_2^y 不能被推导，因为这意味着 $x \prec_{\varphi_{t_2}} y$，与 $y \prec_{\varphi_{t_1}} x$ 矛盾；这会导致一个涉及 r_y 的未证明原子环。

致谢。

本研究得到奥地利科学基金（FWF）资助，资助项目编号为 J 4656 和 P 32830；同时也得到维也纳科学技术基金资助，资助项目编号为 WWTF ICT19-065。

参考文献

Abels, D.; Jordi, J.; Ostrowski, M.; Schaub, T.; Toletti, A.; and Wanko, P. 2019年。混合 ASP 列车调度。载于 *LPNMR*，第 11481 卷*LNCS*，第 3–17 页。施普林格。

Alviano, M.; Amendola, G.; Dodaro, C.; Leone, N.; Maratea, M.; and Ricca, F. 2019年。WASP 中析取程序的评估。载于*LPNMR'19*，第 11481 卷*LNCS*，第 241–255 页。施普林格。

Alviano, M.; 和 Dodaro, C. 2016年。《析取逻辑程序的完备化》。载于 *IJCAI'16*，886–892。IJ-CAI/AAAI出版社。Atserias, A.; Fichte, J. K.; 和 Thurley, M. 2011年。具有多次重启和有界宽度归结的子句学习算法。*J. Artif. Intell.Res.*，40：353–373。Balduccini, M.; Gelfond, M.; 和 Nogueira, M. 2006年。基于答案集的知识系统设计。《数学与人工智能年刊》，47(1-2)：183–219。Ben-Eliyahu, R.; 和 Dechter, R. 1994。析取逻辑程序的命题语义。《数学与人工智能年刊》，12(1)：53–87。Bichler, M.; Morak, M.; 和 Woltran, S. 2020年。selp：单次执行的认知逻辑程序求解器。*TheoryPract.Log.Program.*，20(4)：435–455。Bido´ıt, N.; 和 Froidevaux, C. 1991年。默认否定与不可分层逻辑程序。*Theo. Comput. Sci-ence*，78(1)：85–112。Biere, A.; Heule, M.; van Maaren, H.; 和 Walsh, T.，编。2009年。可满足性手册，第185卷，收录于人工智能前沿与应用。IOS出版社。ISBN 978-1-58603-929-5。

Bliem, B.; Morak, M.; Moldovan, M.; 和 Woltran, S. 2020年。树宽对答案集程序基化与求解的影响。*J. Artif. Intell. Res.*，67：35–80。

Bliem, B.; Ordyniak, S.; and Woltran, S. 2016年。答案集编程的团宽度与有向宽度度量。载于*ECAI'16*，第 285 卷*FAIA*，第 1105–1113 页。IOS出版社。

Bomanson, J. 2017年。lp2normal——一种用于扩展逻辑程序的规范化工具。载于*LPNMR'17*，*LNCS*第10377卷，222–228页。施普林格。

Bomanson, J.；Janhunen, T. 2013年。使用合并和排序构造对基数规则进行规范化。载于*LPNMR'13*，*LNCS*第8148卷，187–199页。施普林格。

Brewka, G.; Eiter, T.; and Truszczy´nski, M. 2011年. 答案集编程概览. 《*ACM*通讯》, 54(12): 92–103. Clark, K. L. 1977年. 失败否定. In 逻辑与数据库, 数据库理论进展, 293–322. Plemum出版社. Cygan, M.; Fomin, F. V.; Kowalik, Ł.; Lokshtanov, D.; D´aniel Marx, M. P.; Pilipczuk, M.; and Saurabh, S. 2015年.参数化算法. 施普林格. ISBN 978-3-319- 21274-6. Diestel, R. 2012年. 图论, 第*4*版, 数学研究生教材第173卷数学研究生教材. 施普林格. ISBN 978-3-642-14278-9. Downey, R. G.; and Fellows, M. R. 2013年. 参数化复杂性基础. 计算机科学丛书. 施普林格. ISBN 978-1-4471-5558-4. Eiter, T.; and Gottlob, G. 1995年. 析取逻辑编程的计算成本：命题情形.《数学与人工智能年刊》, 15(3–4): 289–323. Fandinno, J.; and Hecher, M. 2021年. 答案集编程中的树宽感知复杂性：并非所有正环都同样困难. In *AAAI'21*, 6312–6320. AAAI出版社.

Fichte, J. K.; and Hecher, M. 2019年. 树宽与计数投影答案集. In *LPNMR'19*, LNCS第11481卷, 105–119. 施普林格.

Fichte, J. K.；Hecher, M.；和 Pfandler, A. 2020年。《有界树宽量化布尔公式的下界》。载于*LICS'20*，410–424。计算机协会。

Flum, J.；和 Grohe, M. 2006年。参数化复杂性理论，理论计算机科学丛书第XIV卷。施普林格。ISBN978-3-540-29952-3。

Gebser, M.；Kaminski, R.；Kaufmann, B.；和 Schaub, T. 2012年。实践中的答案集求解。Morgan & Claypool。

Gebser, M.；Kaufmann, B.；和 Schaub, T. 2009年。投影布尔搜索问题的解枚举。载于 *CPAIOR'09*, LNCS 第 5547 卷，71–86 页。施普林格。ISBN 978-3-642-01929-6。Gelfond, M.；和 Lifschitz, V. 1991年。逻辑程序和析取数据库中的经典否定。*NewGenerationComput.*，9(3/4)：365–386。Gottlob, G.；Scarcello, F.；和 Sideri, M. 2002年。人工智能与非单调推理中的固定参数复杂性。人工智能，138(1-2)：55–86。Guziolowski, C.；Videla, S.；Eduati, F.；Thiele, S.；Cokelaer, T.；Siegel, A.；和 Saez-Rodriguez, J. 2013年。使用答案集编程对信号网络的可行逻辑模型进行穷尽表征。*Bioin-formatics*，29(18)：2320–2326。勘误见 Bioinformatics 30, 13, 1942。Hecher, M. 2022年。正规答案集编程到 SAT 的树宽感知归约——正规答案集编程终究比 SAT 更难吗？人工智能，304：103651。Impagliazzo, R.；Paturi, R.；和 Zane, F. 2001年。哪些问题具有强指数复杂性？计算机与系统科学杂志，63(4)：512–530。Jakl, M.；Pichler, R.；和 Woltran, S. 2009年。具有有界树宽的答案集编程。载于 *IJCAI'09*，第 2 卷，816–822 页。

Janhunen, T. 2006年。正规逻辑程序和命题理论的一些（不可）转换性结果。应用非经典逻辑学杂志，16(1-2)：35–86。

Janhunen, T.；和 Niemel¨a, I. 2016年。答案集编程范式。《人工智能杂志》，37(3)：13–24。

Kleine B¨uning, H.；和 Lettman, T. 1999。《命题逻辑：演绎与算法》。剑桥大学出版社。ISBN978-0521630177。

Kloks, T. 1994. 树宽。计算与近似，*LNCS*第842卷。施普林格。ISBN 3-540-58356-4。

Lackner, M.; and Pfandler, A. 2012. 用于寻找极小模型的固定参数算法。载于*KR'12*。AAAI出版社。Lifschitz, V.; and Razborov, A. A. 2006. 为什么有这么多环公式？计算机逻辑学会汇刊，7(2): 261–268。Lin, F.; and Zhao, J. 2003. 关于紧逻辑程序，以及从正规逻辑程序到命题逻辑的另一种转换。载于*IJCAI'03*，853–858。摩根·考夫曼出版社。Lin, F.; and Zhao, Y. 2004. ASSAT：通过 SAT 求解器计算逻辑程序的答案集。人工智能，157(1-2): 115–137。Lokshtanov, D.; Marx, D.; and Saurabh, S. 2011. 稍超指数参数化问题。载于*SODA'11*，760–776。工业与应用数学学会。Marek, W.; and Truszczy´nski, M. 1991. 自认知逻辑。*ACM*期刊，38(3): 588–619。Marx, D.; and Mitsou, V. 2016. 以树宽为参数的可选择性问题的双指数和三指数界。载于*ICALP'16*，信息学出版系列第55卷，28:1–28:15。达格施图尔出版社。ISBN 978- 3-95977-013-2。Niedermeier, R. 2006. 固定参数算法入门，牛津数学及其应用讲座系列第31卷。牛津大学出版社。ISBN 978-0-19-856607-6。

Niemel¨a, I.; Simons, P.; and Soininen, T. 1999. 权重约束规则的稳定模型语义。载于*LP-NMR'99*，LNCS第1730卷，317–331。施普林格。ISBN 3-540-66749-0。Nogueira, M.; Balduccini, M.; Gelfond, M.; Watson, R.; and Barry, M. 2001. 面向航天飞机的 A-Prolog 决策支持系统。载于*PADL' 01*，LNCS第1990卷，169–183。施普林格。ISBN 978-3-540-45241-6。Pichler, R.; R¨ummele, S.; Szeider, S.; and Woltran, S. 2014. 带权重约束的易处理答案集编程：有界树宽还不够。理论与实践逻辑程序，14(2)。Robertson, N.; and Seymour, P. D. 1986. 图小子 II：树宽的算法方面。算法期刊，7: 309–322。Schaub, T.; and Woltran, S. 2018. 答案集编程特刊。*KI*，32(2-3): 101–103。Truszczynski, M. 2011. 析取逻辑程序推理复杂性的三分性与二分结果。理论与实践逻辑程序，11(6): 881–904。

正确性证明

设 t 是 T的一个节点。接着，我们定义 $Q(t)$：$= \{q_t^x \mid x \in V_t\}$ 为完整的查询标记集合（针对 t）。

引理 1（可判定性）。从布尔公式 F 和 $\mathcal{G}_\Pi$ 的树分解 $\mathcal{T} = (T, \chi)$ 到由公式（*1*）至（*19*）组成的逻辑程序 Π的归约确保可判定性：对于 Π 的每个答案集 M 和 T的每个节点 t，至少存在 T 的一个节点 t'，使得 $V_{t'} = V_t$ 且$Q(t') \subseteq M$。

证明（概要）。反设存在 Π 的一个答案集 M 和 T的一个节点 t，使得不存在 T的节点 t' 满足 $V_{t'} = V_t$ 且 $Q(t') \subseteq M$。

然而，根据公式 (3)，对于每个 $y \in V_t$（目标顶点 $d_{\mathrm{root}(T)}$ 除外），都必须存在某个 $e_{y,\ x} \in M$。公式 (8)–(10) 确保：(1) 每个 $y \in V_t$ 至多只能有一条出边。假设存在矛盾：对于 $y \in V_t$，有两条边 $e_{y,\ x}$，$e_{y,\ x'} \in M$。那么，不失一般性地，我们会在某个节点 t'' 中遇到 $e_{y,\ x} \in E_{t''}$。根据公式 (9)，我们要求 $o^y_{t''}$。接着，根据公式 (8)，对于位于 t'' 与我们遇到 $e^y_{y,\ x'} \in E_{t'''}$ 的节点（比如 t'''）之间的每个 t_i，都要求满足 $o^y_{t_i} \in M$。注意，根据树分解的连通性，从 t'' 到 t''' 必须存在这样一条路径。最后，由于 $o^y_{t_i} \in M$，针对 t''' 的公式 (10) 无法满足，因此我们得出矛盾。因此，我们得出结论：(2) 每个 $y \in V_t$（目标顶点 $d_{\mathrm{root}(T)}$ 除外）至多有一条出边。结合 (1) 和 (2)，并根据公式 (2)，我们可以推得，对于每个 $y \in V_t$（$d_{\mathrm{root}(T)}$ 除外），在 M中都恰好要求有一条出边。

因此，根据 Π 的构造，在 V_t 中有一个顶点最先被到达，随后用它到达第二个顶点，依此类推。我们可以按照可达性的先后顺序将这些顶点排列在一条路径上，即 $P = v_1, \ldots, v_{|V_t|}$。然而，根据 Π 的构造，必定存在一个节点 t^*，使得 φ_{t^*} 精确反映了 P 的序。注意，$p^{v_1}{}_{\mathrm{prev}(t^*, v_1)} \in M$，它要么由公式 (4) 推导得出（如果 $\mathrm{prev}(t^*, v_1) = \epsilon$），要么由公式 (11) 推导得出（如果对于 t^* 的子节点，我们有 $\mathrm{dom}(\varphi_{\mathrm{prev}(t^*, v_1)}) = \emptyset$）。如果 $\mathrm{dom}(\varphi_{\mathrm{prev}(t^*, v_1)}) \neq \emptyset$，那么 $p^{v_1}{}_{\mathrm{prev}(t^*, v_1)} \in M$ 要么由公式 (16) 推导得出，要么通过公式 (12) 或 (13) 推导得出。无论如何，我们都可以推导出 $Q(t^*) \subseteq M$，它们恰好按照 P 的序，使用公式 (14) 推导得出，随后再应用公式 (15) 若干次。这与 t' 不存在相矛盾，因为 $t' = t^*$。 □

引理 2（兼容性）。从布尔公式 F 和树分解 $\mathcal{T} = (T, \chi)$ 到逻辑程序 Π的归约由公式 *(1)* 至 *(19)* 组成，并确保兼容性：设 M 是 Π的一个答案集。那么，对于 T中任意两个节点 t、t'，若 $Q(t)$ $Q(t') \subseteq M$，则 $\mathcal{I}_t(\varphi_t)$ 和 $\mathcal{I}_{t'}(\varphi_{t'})$ 兼容。

证明。情形 $V_t \neq V_{t'}$：反设存在这样的 Π 的答案集 M 以及 T中的两个节点 t, t'，满足 $Q(t)$ $Q(t') \subseteq M$，且 $\mathcal{I}_t(\varphi_t)$ 与$\mathcal{I}_{t'}(\varphi_{t'})$不兼容。然而，根据不相容性的定义，这两个赋值必须共享至少一个变量 x，满足 $[\mathcal{I}_t(\varphi_t)](x) \neq [\mathcal{I}_{t'}(\varphi_{t'})](x)$。因此，根据良好树分解 $\mathcal{T}$的构造，必定存在 T中的一个节点 t^*，满足 $\varphi_{t^*} = \varphi_t$ 和 $\psi_{t^*} = \varphi_{t'}$，或反之（$\varphi_{t^*} = \varphi_{t'}$ 和 $\psi_{t^*} = \varphi_t$）。但这样一来，公式 (17) 对 t^*不成立，这与 M 是 Π的一个答案集相矛盾。

情形 $V_t = V_{t'}$：反设存在这样的答案集 M of Π，其序为 α，能够证明 M，并且存在 T中的两个节点 t、t'，满足 $Q(t)$ $Q(t') \subseteq M$，其中 $\mathcal{I}_t(\varphi_t)$ 与 $\mathcal{I}_{t'}(\varphi_{t'})$ 不兼容。于是，至少存在两个顶点 x、$y \in V_t$，满足 $\varphi_t(x) > \varphi_t(y)$，但 $\varphi_{t'}(y) > \varphi_{t'}(x)$，否则 $\varphi_t = \varphi_{t'}$。根据 $\mathcal{R}$ 的构造，特别是公式 (15)，从 $q^y_{t'}$ 到 $q^x_{t'}$ 存在一条规则依赖链，这要求使用 $q^y_{t'}$ 来推导 $q^x_{t'}$，即 $\alpha(q^y_{t'}) < \alpha(q^x_t)$。另一方面，对于 t，从 q^x_t到 q^y_t 存在一条规则依赖链，即 $\alpha(q^x_t) < \alpha(q^y_t)$。

子情况 r_y 用于推导 r_x，即 $\alpha(r_y) < \alpha(r_x)$：此子情况与图 3 中带红色边的情况类似。因此，依赖链间接推导出 r_y，再推导出 p^x_ϵ；根据公式 (6) 和 (7)，这是推导 r_x 所必需的。因此，我们有 $\alpha(r_y) <\alpha(p^x_\epsilon) < \alpha(r_x)$。然后，根据公式 (14) 和 (15)，我们要求 $\alpha(p^x_\epsilon) < \alpha(q^x_t)$。此外，根据构造，例如公式 (16) 和 (11)–(13)，我们有 $\alpha(q^y_t) < \alpha(r_y)$；根据上述假设，我们有 $\alpha(q^x_t) < \alpha(q^y_t)$。因此，我们得出矛盾：$\alpha(r_y) < \alpha(r_y)$。

子情形 r_x 用于推导 r_y，即 $\alpha(r_x) < \alpha(r_y)$：这里，依赖链间接推导出 r_x，然后才推导出 p^y_ϵ；而根据公式 (6) 和 (7)，推导 r_y需要 p^y_ϵ。因此，我们有 $\alpha(r_x) < \alpha(p^y_\epsilon) < \alpha(r_y)$。

接着，根据公式 (14) 和 (15)，我们要求 $\alpha(p_\epsilon^y) < \alpha(q_{t'}^y)$。根据构造，例如公式 (16) 和 (11)–(13)，我们有 $\alpha(q_t^x) < \alpha(r_x)$；根据上述假设，我们有 $\alpha(q_{t'}^y) < \alpha(q_t^x)$。因此，我们得出了矛盾：$\alpha(r_x) < \alpha(r_x)$。□

定理 3（正确性）。从布尔公式 F 和树分解 $\mathcal{T} = (T, \chi)$ $\mathcal{G}_\Pi$ 到逻辑程序 Π的归约由公式 *(1)* 到 *(19)* 组成，且是正确的。具体而言，对于 F 的每个模型，都至少存在一个 Π 的答案集。反过来，对于 Π 的每个答案集，都存在唯一一个 F 的模型。

证明。“⇒”：令 $I\colon \mathsf{at}(F) \to \{0,1\}$ 为 F 的一个模型，即 $I \models F$。据此，我们如下构造 Π 的一个答案集 A。对于 T 中的每个 t，令 $\alpha_t := \mathcal{I}_t^{-1}(I_{|V_t})$ 为赋值 I 在包 V_t 上的限制所对应的序。我们如下构造 A'：

(i) 对于 T 中满足 $\alpha_t = \langle a_1, \ldots, a_\ell \rangle$ 且 $1 \leq i \leq \ell$ 的每个 t，我们要求 $p_\epsilon^{a_i} \in A'$ 以及 $r_{a_i} \in A'$；如果 t 是叶节点，还要求 $r_{s_t} \in A'$；(ii) 对于 T 中满足 $\alpha_t = \langle a_1, \ldots, a_\ell \rangle$ 的每个 t，对于每个 $1 \leq i < \ell$，我们要求 $e_{a_i, a_{i+1}} \in A'$；(iii) 对于 T 中满足 $\alpha_t = \langle a_1, \ldots, a_\ell \rangle$、$t' \in \mathsf{chldr}(t)$，且 $\alpha_{t'} = \langle a'_1, \ldots, a'_{\ell'} \rangle$ s.t. $\alpha_t \neq \alpha_{t'}$ 的每个 t，我们要求 $e_{a'_{\ell'}, a_1} \in A'$；(iv) 对于 T 中的每个 t、$\alpha \in \{\varphi_t, \psi_t\}$ 和 $\alpha = \langle a_1, \ldots, a_\ell \rangle$，对于每个 $1 \leq i \leq \ell$，都需要有 $p_t^{a_i} \in A'$。此外，对于与 α_t 兼容的 α ($\ell' \leq \ell$) 的最大序列 $\beta = \langle a_1, \ldots, a_{\ell'} \rangle$，我们还要求对每个 $1 \leq j \leq \ell'$ 都有 $q_t^{a_j} \in A'$。

然后，我们基于 A' 定义 A''，其中，对于 T 中满足 $x, y \in V_t$ 和 $e_{x,y} \in A'$ 的每个 t，都必须有 $o_t^y \in A''$。此外，对于 T 中满足 $y \in V_{t^*}$ 的 t 的每个父节点和祖先 t^*，我们要求 $o_{t^*}^y \in A''$。

然后，我们基于 A' 和 A'' 定义 A，其中 (vi)$A := A' \cup A'' \cup \{\hat{}e_{x,y} \mid e_{x,y} \in (\bigcup_{t\text{属于 }T} E_t \setminus A')\}$。

我们证明 A 确实是 Π的一个答案集。首先，我们在第 (1) 部分证明 A 确实满足 Π中的每条规则，然后在第 (2) 部分证明 A 确实是 Π的一个极小子集模型。第 (1) 部分：根据 (i)，A 满足公式 (1)、(2) 和 (4)。根据 (ii)、(iii) 和 (vi)，我们有由此可知，A 满足公式 (3)。根据 (v)，A 满足公式 (8)、(9) 和 (10)。根据 (iv)，A 也满足公式 (17)、(18) 和 (19)。根据 (iv)，A 满足头部含有形如 p_t^x 的变量的那些公式，即公式 (11)、(16)、(12) 和 (13)。此外，根据 (i)，A满足公式 (6) 和 (7)。最后，根据 (iv)，我们有 A满足公式 (14) 和 (15)。

第 (2) 部分：反设存在一个集合 $B \subsetneq A$，满足公式 (1)–(18)。注意，B 与 A 之间的差异不可能是由 (ii)、(iii) 或 (vi) 导致的，否则 B 就不满足公式 (3)。此外，由于公式 (1) 以及公式 (4)，要使 B 成为答案集，就必须满足定义 (i)；而对于 T 中每个 t 的每个 $y \in V_t$，只有要求 $p_\epsilon^y \in B$ 才能满足公式 (4)，参见公式 (6)、(7)。然后，为了满足公式 (8) 和 (9)，还需要定义 (v)。如果 B 没有将至少某个 p_t^x 设为真，那么 B 就无法满足公式 (11)、(16)、(12) 或 (13) 中的某一条。另一方面，如果 B 没有将某个 q_t^x 设为真，那么 B就无法满足公式 (14) 或 (15)。

“⇐”：令 M 为 Π的一个答案集。根据引理 1，对于 T 中的每个节点 t，T 中至少存在一个节点 t'，满足 $V_{t'} = V_t$ 和 $Q(t') \subseteq M$。据此，我们构造如下赋值 $I\colon \mathsf{at}(F) \to \{0,1\}$，其中对于每个 $x \in \mathsf{at}(F)$，我们定义 $I(x) := v$，只要 T 中存在一个 t，使得 $Q(t) \subseteq M$ 且 $\mathcal{I}_t(\varphi_t)(x) = v$。注意，根据引理 1，$\mathsf{at}(F)$ 中的每个变量 $x \in$ 都确实得到了处理；最后，根据引理 2 的兼容性，I 定义良好。由 I 的构造以及公式 (18) 和 (19)，可得 $I \models F$。更确切地说，反设 $I \not\models F$。那么，至少存在一个子句 $c \in F$，满足 $I \not\models \{c\}$。根据树分解的性质，必定存在 T 中至少一个节点 t，满足 $c \in F_t$。根据引理 1，对于每个节点 t，T 中至少存在一个节点 t'，满足 $V_{t'} = V_t$ 和 $Q(t') \subseteq M$。于是，根据公式 (18)（和 (19)），有 $\mathcal{I}_{t'}(\alpha_{t'}) \models F_{t'}$。因此，根据 I 的构造以及引理 2 的兼容性，有 $I \models F_{t'}$。这与假设 $I \not\models \{c\}$ 矛盾。

注意，对于每个不同的答案集 M，I 都有唯一的定义。反设存在两个不同的答案集 M，M'，它们属于 Π，且具有相同的 I（如上文定义和构造）。但是，如果 M 和 M' 都只在形如 $e_{y,\,x}$ 的变量上有所不同，那么可以证明，T 中存在一个节点 t，使得 $Q(t) \subseteq M$，但 $Q(t) \not\subseteq M'$。在引理 1 的证明中，已证明

$Q(t) \in M$ 如何精确地依赖于这些边 $e_{y,\ x}$，这反映了序 y 先于 x。因此， M 和 M' 不可能只在形如 $e_{y,\ x}$ 的变量上有所不同。于是，由于其余公式仅用于传播信息（除公式 (3) 外没有其他选择），我们得到 $M = M'$，这与我们的假设矛盾。 □

B 补充证明

引理 3 (最大强连通分量大小)。 设 F 是一个布尔公式，且 $\mathcal{T} = (T, \chi, \varphi, \psi)$ 是 $\mathcal{G}_F$ 的一个宽度为 k的带排序的树分解。那么，归约 $\mathcal{R}$ 构造出的程序，其强连通分量（*SCC*）的大小至多为 $2^{\mathcal{O}(k' \cdot \log(k'))}$，其中 $k' = {}^{k}\log(k)$，这等于 $2^{\mathcal{O}(k)}$。

证明。注意， D_Π 的最大强连通分量只能由公式 (3)–(7) 和 (11)–(16) 导致，因为这些是唯一可能形成循环的规则。该强连通分量由形如 $r_x, e_{x,y}, p^x_\epsilon, p^x_t, q^x_t$ 的变量构成，其中 $x, y \in V_{t^*}$ 对于 T 的某些 t, t^* 满足 $V_t = V_{t^*}$。因此，该强连通分量中形如 r_x 的变量至多有 k' 个，形如 $e_{x,y}$ 的变量至多有 k'^2 个。然而，由于公式 (12)–(16) 会分析任意两个排序 φ_t, ψ_t 的潜在组合，形如 p^x_t 和 q^x_t 的变量至多有 $(k'!)^2$ 个。虽然为排序 φ_t 和 ψ_t 构造的变量按定义属于不同的强连通分量（参见定义 1），但这些组合会导致对形如 p^x_t, q^x_t 的变量进行重复排序的检查，其数量仍不会超过 $(k'!)^2$ 种组合。显然，$(k'!)^2$ 主导强连通分量大小，而其上界为

$2^{\mathcal{O}(2 \cdot k' \cdot \ k')} = 2^{\mathcal{O}(k' \cdot \ k')}$ $2^{\mathcal{O}(k' \cdot \ k')^2} = \log(\ ))$ $\log(\ )) \log(\ ))$. Then, $\log(\ ))$ $= 2^{\mathcal{O}(\overline{{}^{k}{}_{k} \cdot} \ k')} = 2^{\mathcal{O}(\overline{{}^{k}{}_{k} \cdot} \ k - \ k)} = 2^{\mathcal{O}(k)} \log(\ )$ $\log(\ )) \log(\ ) (\log(\ ) \log(\log(\ )))$，由此即可得证。 □

推论 1 (LB 最大强连通分量)。 设 Π 是一个正规逻辑程序，其中 $\mathcal{G}_\Pi$ 的树宽为 k，且 D_Π 的最大强连通分量大小属于 $2^{\mathcal{O}(k \cdot \log(k))}$。那么，除非指数时间假设不成立，否则不太可能在 $2^{o(k \cdot \log(k))} \cdot \text{poly}(|\text{at}(\Pi)|)$ 时间内判定 Π 的一致性。

证明。 该结果由归约 $\mathcal{R}$ 的正确性（见定理 3）、定理 1 的树宽感知性质，以及引理 3 给出的最大强连通分量大小上界可得。反设存在一种算法可以在时间 $2^{o(k \cdot \log(k))} \cdot |\text{poly}(\text{at}(\Pi))|$ 内求解 F。那么，我们任取一个布尔公式 F 和一个宽度为 k' 的 $\mathcal{G}_F$ 的带排序的树分解 $\mathcal{T}$，并通过 $\mathcal{R}$ 构造 Π。根据定理 3，该归约是正确的，即当且仅当 F 有满足赋值时，Π 才有答案集。此外，根据引理 3，D_Π 的最大强连通分量大小有上界 $2^{\mathcal{O}(k \cdot \log(k))}$。根据定理 1，$\Pi$ 的树宽 k 有上界 $\mathcal{O}(\ {}^{k'}\log(k'))$。因此，若能在时间 $2^{o(k \cdot \log(k))} \cdot \text{poly}(|\text{at}(\Pi)|)$ 内判定 Π，就能在时间 $2^{o(\ {}^{k'}\log(k') \cdot \log(\ {}^{k'}\log(k')))} \cdot \text{poly}(|\text{at}(F)|) = 2^{o(k')} \cdot \text{poly}(|\text{at}(F)|)$ 内求解 F，这与指数时间假设相矛盾。 □

以下是关于量化布尔公式的一个已知结果：事实证明，判定量化布尔可满足性仍然需要关于原始图树宽的 ℓ 重指数时间（即使将图限制为最内层量词块中的变量所构成的图）。

命题 2 (Fichte, Hecher,and Pfandler (2020))。给定一个量词深度为 ℓ 的量化布尔公式 $Q = \forall V_1 . \exists V_2 , \ldots , \exists V_\ell . F$，其中 $\mathcal{G}_F$ 的树宽为 $k+1$，且 k 是将 $\mathcal{G}_F$ 限制在 V_ℓ 中的顶点后得到的树宽。那么，在指数时间假设成立的条件下，无法在 $\text{tow}(\ell, o(k)) \cdot \text{poly}(\text{at}(Q))$ 时间内判定 Q 的有效性。

该命题将作为推导以下结果的基础。

定理 4 (投影计数下界)。 设 Π 为正规逻辑程序，$P \subseteq \text{at}(\Pi)$ 为原子，且 k 为 $\mathcal{G}_\Pi$的树宽，并且 D_Π 的最大强连通分量大小属于 $2^{\mathcal{O}(k \cdot \log(k))}$。那么，在指数时间假设下，无法在时间 $2^{2^{o(k \cdot \log(k))}} \cdot \text{poly}(|\text{at}(\Pi)|)$ 内计算基数 $|\{M \cap P \mid M \models \Pi\}|$。

证明。 思路是应用命题 2 的结果，从 $Q = \forall V_1 . \exists V_2 . F$归约得到； $\mathcal{G}_F$的树宽为 k'。在命题 2 的证明中，作者实际上针对一种受限的 QBF 证明了一个更强的结果：存在 $\mathcal{G}_F$的良好树分解 $\mathcal{T} = (T, \chi)$，使得对于每个变量 $v \in V_1 \cup \ldots \cup V_{\ell-1}$， T中都存在唯一的节点 t满足 $v \in \chi(t)$，即 $\{v\} = \chi(t) \cap (V_1 \cup \ldots \cup V_{\ell-1})$。假设 (A)：我们假设已为 $\ell = 2$ 构造了这样一个良好树分解 $\mathcal{T}$，并将其转化为 $\mathcal{G}_F$的带排序的树分解 $\mathcal{T}' = (T', \chi', \varphi, \psi)$。

然后，我们设定 $P := V_1$，并使用归约 $\mathcal{R}$，根据 F构造程序 Π'。接着，对于每个变量 $v \in V_1$，我们构造规则 $v \leftarrow \neg \hat{} v$ (20) 和 $\hat{} v \leftarrow \neg v$ (21)

基于新原子 v 和 $\hat{}v$。这些规则负责为新变量猜测真值。此外，我们通过向 Π' 添加以下规则来构造 Π。首先，我们添加与公式 (18) 类似的兼容规则，以确保这些真值反映在归约中，具体如下：$\leftarrow v,\ q_t^x$(22)，对 T' 中每个满足以下条件的节点 t：$x \in \mathsf{dom}(\varphi_t)$，且不存在这样的 $\prec_{\varphi_t}$ 后继，使得 $\mathcal{I}_t(\varphi_t)[v] = 0$。类似地，对于 T' 中每个满足以下条件的节点 t，我们添加 $\leftarrow \hat{}v,\ q_t^x$ (23)：$x \in \mathsf{dom}(\varphi_t)$，且不存在这样的 $\prec_{\varphi_t}$ 后继，使得 $\mathcal{I}_t(\varphi_t)[v] = 1$。

根据定理 3 的正确性证明以及上述调整，不难看出，只要满足 $|\{M \cap P \mid M \models \Pi\}| = 2^{|V_1|}$，$Q$ 就会求值为真。进一步地，由于根据假设 (A)，每个变量 $v \in V_1$都唯一地出现在 T的一个节点 t 中；根据定义 1，在 T' 中，位于 t 下方、从节点 t_1、.....、t_o到节点的路径上，针对 t_i 的赋值 φ_{t_i} 中，除了 v 之外，不会出现 V_1 中的其他变量。因此，我们可以轻松修改定理 1 构造的树分解，只需将 v和 $\hat{}v$添加到恰好这些包 $\chi(t_1)$.....、$\chi(t_o)$ 中即可。因此，与定理 1 相比，树宽至多增加 2，即 k 在 $\mathcal{O}(k' \log(k'))$ 中的树宽保证仍然成立。

现在反设$|\{M \cap P \mid M \models \Pi\}|$ 可以在时间 $2^{2^{o(k\cdot\log(k))}} \cdot \mathsf{poly}(|\mathsf{at}(\Pi)|)$内计算。然而，我们可以在时间 $2^{2^{o(k'\cdot(\log(k')-\log(\log(k'))))}} \cdot \mathsf{poly}(|\mathsf{at}(F)|) = 2^{2^{o(k')}} \cdot \mathsf{poly}(|\mathsf{at}(F)|)$内判定 Q，这与指数时间假设相矛盾。 □

观察 2。设 $\iota, k \in \mathbb{N}$ 满足 $2 \le \iota \le k$。那么，$(\iota!)^{k}_{\iota}$属于 $2^{\Omega(k\cdot\log(\iota))}$。

证明。根据观察 1，$\iota!$有上界 $2^{\Omega(\iota\cdot\log(\iota))}$。因此，$(\iota!)^{k}_{\iota}$属于 $2^{\Omega(\iota\cdot\log(\iota))^{k}_{\iota}}$，而 $2^{\Omega(\iota\cdot\log(\iota))^{k}_{\iota}}$属于 $2^{\frac{k}{\iota}\cdot\Omega(\iota\cdot\log(\iota))}$，从而得到结果：$(\iota!)^{k}_{\iota}$属于 $2^{\Omega(k\cdot\log(\iota))}$。 □

C 归约 $\mathcal{R}'$ 用于 ι-紧性

下面我们描述为归约 $\mathcal{R}'$ 对 $\mathcal{R}$ 所需的修改。令 F 为一个布尔公式，$\mathcal{T} = (T, \chi, \varphi_t, \psi_t)$ 为 $\mathcal{G}_F$ 的一个宽度为 k 的带排序的树分解。回顾一下，对于固定的 $2 \le \iota \le k'$，我们将树宽从 k 降低到满足 $(\iota!)^{k'}_{\iota} \ge 2^k$ 的最小 k'。然后，对于 T的每个节点 t，k'_t 是满足 $(\iota!)^{k'_t}_{\iota} \ge 2^{|\chi(t)|}$的最小整数。与上文类似，$V_t$ 由包 $\chi(t)$唯一确定，并由 k'_t 个新元素组成。

在归约 $\mathcal{R}'$的上下文中，对于 T的每个节点 t，我们令 $\mathsf{ord}(t)$ 为上的偏序集合，其中 V_t 中的每个元素都与中同一块的 ι <style id='12'}个元素（剩余 $< \iota$ 个元素则更少）全序排列。因此，实质上，$\mathsf{ord}(t)$ 中的每个序都是至多 $\binom{k'_t}{\iota}$ 个独立全序的组合。假设这些至多含有 ι 个元素的块之间存在任意但固定的全序。对于每个 $v \in V_t$，我们用 $\mathsf{bn}(v, t)$ 表示 $1 \le j \le \binom{k'}{\iota}$ 所属的至多含有 ι 个元素的块编号。

对于T的每个节点t，令E_t为满足以下条件的最大子集：$E_t \subseteq \{e_{x,y} \mid x \prec_{\varphi_t} y\} \cup \{e_{s^j_{V_t},a} | a \in \mathsf{dom}(\varphi_t)$ 没有$\prec_{\varphi_t}$前驱 $j = \mathsf{bn}(a, t)\} \cup \{e_{b,d^j_{V_t}} | b \in \mathsf{dom}(\varphi_t)$ 没有 $\prec_{\varphi_t}$后继 $j = \mathsf{bn}(a, t)\}$ 且对于 T中满足 $t' \ne t$ 的任意节点 t'，都有 $E_t \cap E_{t'} = \emptyset$。

图4展示了取自 $\mathcal{R}$ 的略作改造的归约 $\mathcal{R}'$，其中公式 (1)、(2)、(5)、(8) 和 (10) 分别替换为公式 (1')、(2')、(5')、(8') 和 (10')。利用这一归约，我们得到以下结果。

定理6 ($\mathcal{R}'$的树宽感知)。从布尔公式 F 和序增强的树分解 $\mathcal{T} = (T, \chi, \varphi, \psi)$ $\mathcal{G}_F$ 出发，通过归约 $\mathcal{R}'$ 得到宽度为 k 的正规程序 Π，使用某个固定的 $2 \le \iota \le k^{\log(\iota)}$，树宽略有降低。具体而言，$\mathcal{G}_\Pi$ 的树宽至多为 $\mathcal{O}(k^{\log(\iota)})$。此外，存在一个树分解，表明 Π 的紧性宽度属于 $\mathcal{O}(\iota)$。

证明（概要）。我们构造一个 $\mathcal{G}_\Pi$ 的树分解 $\mathcal{T}' = (T, \chi')$，以表明 $\mathcal{T}'$ 的宽度（与 k 相比）仅略有增加。为此，令 t 为 T 中一个满足 $\mathsf{chldr}(t) = \langle t_1, \ldots, t_\ell \rangle$ 的节点，并令 $\hat{}t$ 为 t 的父节点（如果存在）。我们定义 (i) $R(t) := \{r_x \mid x \in V_t\} \cup \{r_x \mid x \in V_{t'}, t' \in \mathsf{chldr}(t)\} \cup \{r_{s^j_{V_t}}, r_{d^j_{V_t}} \mid z \in V_t, j = \mathsf{bn}(z, t)\}$, (ii) $E(t) := \{e_{x,y}, \hat{}e_{x,y} \mid e_{x,y} \in E_t\}$, (iii) $P(t) := \{q_t^x, p_t^x, p_{t'}^x, p_\epsilon^x \mid x \in \mathsf{dom}(\varphi_t) \cup \mathsf{dom}(\psi_t)\ t' \in \mathsf{prev}(x, t)\}$, 以及 (iv) $O(t) := \{o_t^x, o_t^{s^j_{V_t}} \mid x \in V_t, j = \mathsf{bn}(x, t)\} \cup \{o_{t'}^x, o_{t'}^{s^j_{V_{t'}}} \mid t' \in \mathsf{chldr}(t)\ x \in V_{t'} \cap V_t\}$。接着，令 $\chi'(t) := R(t) \cup E(t) \cup P(t) \cup$

块 1：排序与可达性

$r_{s^j_{V_t}} \leftarrow$ 对于每个 t 属于 $T, z \in V_t, j = \mathrm{bn}(z,t)$ (1')

$\leftarrow \neg r_y$ 对于每个 t 属于 $T, y \in V_t \cup \{d^j_{V_t} \mid z \in V_t,\ j = \mathrm{bn}(z,\ t)\}$ (2')

$r_{d^j_{V_t}} \leftarrow e_{y,d^j_{V_t}}$ 对于每个 t 属于 $T, e_{y,d^j_{V_t}} \in E_t$ (5')

块 2： ≤ 1 **出边**

$o^y_t \leftarrow o^y_{t'}$ for every t in $T, t' \in \mathrm{chldr}(t), y \in (V_t \cap V_{t'}) \cup \{s^j_{V_t} \mid V_t{=}V_{t'}, z \in V_t, z \in \mathrm{bn}(z,t)\}$ (8')

$\leftarrow o^{t'}_{e_{y,x}}$ 对于 t 中的每个 $T,\ e_{y,x} \in E_t, t' \in$ 子节点(t) $y \in V_{t'} \cup \{s^j_{V_t} \mid V_t{=}V_{t'},\ z \in V_t,\ z \in \mathrm{bn}(z,\ t)\}$ (10')

图4：归约 $\mathcal{R}'$ 改编自 $\mathcal{R}$，据此分别用 (1')、(2')、(5')、(8') 和 (10') 替换 (1)、(2)、(5)、(8) 和 (10)。归约 $\mathcal{R}'$ 接收一个公式 F 和一个 $\mathcal{G}_F$ 的带排序的树分解 $\mathcal{T} = (T,\ \chi,\ \varphi,\ \psi)$。

$O(t)$。注意，$\mathcal{T}'$ 是 $\mathcal{G}_\Pi$ 的一个树分解（TD），并且根据构造，$|\chi'(t)|$ 属于 $\mathcal{O}(|V_t|)$。根据 V_t的定义，$|V_t| \leq k'$，其中 $\iota! \leq \iota^{k'} \geq 2^k$。然后，由于 $\iota!$ 属于 $2^{\Omega(k'\cdot\log(\iota))}$（见观察 2），我们有 $2^{\Omega(k'\cdot\log(\iota))}$ 至少为 2^k，因此 $k' = \mathcal{O}(k/\log(\iota))$。此外，根据构造：(i) $R(t)$至多包含 D_Π的任意强连通分量中的 $2\cdot\iota$ 个元素；(ii) $E(t)$至多包含 D_Π的任意强连通分量中的 ι 个元素；(iii) $P(t)$至多与 D_Π的任意强连通分量中的 $4\cdot\iota$ 个元素相交。总之，Π 在 $\mathcal{T}'$ 上的紧性宽度为 $\mathcal{O}(\iota)$。 □

有趣的是，归约 $\mathcal{R}'$ 也无法得到显著改进。

定理7 ($\mathcal{R}'$ 树宽降低幅度最优)。 假设存在一个从公式 F 到正规逻辑程序 Π 的归约，其运行时间为 $2^{o(k)}\cdot\mathrm{poly}(|\mathrm{at}(F)|)$，其中 k是 $\mathcal{G}_F$的树宽。那么，根据指数时间假设，$\mathcal{G}_\Pi$的树宽不可能是 $o(k/\log(\iota))$，对于固定的 $2 \leq \iota \leq k/\log(\iota)$ 而言。

证明。反设这样的归约存在，记作 $\mathcal{R}^*$。于是，我们将该归约应用于任意 F和 ι，得到程序 $\Pi = \mathcal{R}^*(F)$。我们知道，根据 Fandinno和Hecher（2021），Π可在 $2^{\mathcal{O}(k'\cdot\log(\iota))}\cdot\mathrm{poly}(|\mathrm{at}(F)|)$ 时间内判定，其中 k'属于 $o(k/\log(\iota))$，因为根据定理6，Π 在某个宽度为 k'的树分解上具有紧性宽度 ι。因此，Π，从而 F，可在 $2^{o(k/\log(\iota)\cdot\log(\iota))}\cdot\mathrm{poly}(|\mathrm{at}(F)|)$ 时间内判定；该时间属于 $2^{o(k)}\cdot\mathrm{poly}(|\mathrm{at}(F)|)$，这与指数时间假设矛盾。 □

此外，我们证明了以下强连通分量界，参见引理 3。

引理 4 (最大强连通分量大小由 $\mathcal{R}'$)。 设 F是一个布尔公式，$\mathcal{T} = (T, \chi, \varphi, \psi)$ 是 $\mathcal{G}_F$的一个宽度为 k的带排序的树分解。则，对于某个固定的 $2 \leq \iota \leq k'$，归约 $\mathcal{R}'$ 构造出一个程序，其强连通分量（*SCC*）的 D_Π 大小至多为 $2^{\mathcal{O}(k'\cdot\log(\iota))}$，其中 $k' = k/\log(\iota)$，这等于 $2^{\mathcal{O}(k)}$。

证明。注意，D_Π 的最大强连通分量只能由公式 (3)、(4)、(5')、(6)、(7) 和 (11)–(16) 导致，因为这些是唯一可能构成循环的规则。该强连通分量由形如 r_x、e_x、y、p^x_ϵ、p^x_t、q^x_t 的变量构成，其中 x、$y \in V_{t^*}$ 对于某个 t、t^* 满足 T 和 $V_t = V_{t^*}$。因此，在该强连通分量中，形如 r_x 和 p^x_ϵ 的变量至多有 k' 个，而形如 e_x、y 的变量至多有 k'^2 个。然而，由于公式 (12)–(16) 会分析任意两个排序 φ_t、ψ_t 的潜在组合，形如 p^x_t 和 q^x_t 的变量至多有 $(\iota!)^{k'^2}_\iota$ 个。显然，$(\iota!)^{k'^2}_\iota$ 主导强连通分量大小，而其上界为 $2^{\mathcal{O}(k'\cdot\log(\iota))^2} = 2^{\mathcal{O}(2\cdot k'\cdot\log(\iota))} = 2^{\mathcal{O}(k'\cdot\log(\iota))}$。于是，$2^{\mathcal{O}(k'\cdot\log(\iota))} = 2^{\mathcal{O}(k/\log(\iota)\cdot\log(\iota))} = 2^{\mathcal{O}(k)}$ 得证。 □

定理 5 (下界 ι-紧致性)。 设 Π 是一个 ι-紧致逻辑程序，其中 $\mathcal{G}_\Pi$ 的树宽为 k，且 D_Π 的最大强连通分量大小属于 $2^{\mathcal{O}(k\cdot\log(\iota))}$。那么，在指数时间假设下，无法在时间 $2^{o(k\cdot\log(\iota))}\cdot\mathrm{poly}(|\mathrm{at}(\Pi)|)$内判定 Π 的一致性。

证明。 根据定理6的树宽感知性质和引理 4 给出的最大强连通分量大小界，结论成立。

反设存在矛盾，即可以在 $2^{o(k\cdot\log(\iota))}\cdot|\mathrm{poly}(\mathrm{at}(\Pi))|$ 时间内求解 Π。因此，我们取任意一个布尔公式 F 及其宽度为 k' 的带排序的树分解 $\mathcal{T}$（$\mathcal{G}_F$），并通过对 ι 使用 $\mathcal{R}'$ 构造 Π。于是，根据引理 4，D_Π 的最大强连通分量大小至多为 $2^{\mathcal{O}(k\cdot\log(\iota))}$。此外，根据定理6，$\Pi$ 的树宽 k 至多为 $\mathcal{O}(k'\log(\iota))$，并且存在一个树分解，证明了 Π 的紧性宽度为 $\mathcal{O}(\iota)$。因此，假设可以在 $2^{o(k\cdot\log(\iota))}\cdot\mathrm{poly}(|\mathrm{at}(\Pi)|)$ 时间内判定 Π，就能在 $2^{o(k'\log(\iota)\cdot\log(\iota))}\cdot 2^{o(k'\cdot F)}\cdot\mathrm{poly}(|\mathrm{at}(F)|)\cdot\mathrm{poly}(|\mathrm{at}(\ )|)$ 时间内求解 F，这与指数时间假设相矛盾。 □