arXiv:2302.12937v1 [cs.LO] 24 Feb 2023

半环上的约束优化*

A. Pavan r◯[1]、KuldeepS. Meel r◯[2]、N. V. Vinodchandran r◯[3]和 Arnab Bhattacharyya[4]

[1]爱荷华州立大学
pavan@cs.iastate.edu[2]新加坡国立大学，新加坡*meel@comp.nus.edu.sg*[3]内布拉斯加大学林肯分校
vinod@cse.unl.edu[4]新加坡国立大学，新加坡*arnabb@nus.edu.sg*

2023年2月28日

摘要

对半环（布尔半环除外）上的逻辑公式进行解释，在计算机科学的多个领域都有应用，包括逻辑、人工智能、数据库和安全。这类解释提供的信息比陈述为真或为假更为丰富。这类半环包括维特比半环、最小值—最大值半环或访问控制半环、热带半环和模糊半环。

本文研究半环上的约束优化问题的复杂性。我们研究的一般优化问题如下：给定一个 φ 关于 n变量的命题公式和一个半环（$K, +, \cdot, 0, 1$），求 φ关于 K的所有可能解释中的最大值。这可以看作是著名的可满足性问题的推广（当且仅当一个命题公式在布尔半环上的所有解释/赋值中的最大值为 1 时，该公式才是可满足的）。一个相关问题是寻找能够达到最大值的解释。在本文中，我们首先关注维特比半环上的这些优化问题，并将其称为 optConfVal 和 optConf。

我们首先证明，对于否定范式中的一般命题公式，optConfVal 和 optConf 属于 $\mathrm{FP}^{\mathrm{NP}}$。随后，我们研究输入公式 φ 以合取范式表示时的 optConf。对于合取范式公式，我们首先根据最大可满足子句的数量，推导 optConf 值的一个上界。具体而言，我们证明：如果 r 是包含 m 个子句的合取范式公式中可满足子句的最大数量，那么其 optConf 值至多为$1/4^{m-r}$。在此基础上，我们确立了合取范式公式的 optConf 对复杂性类 $\mathrm{FP}^{\mathrm{NP}[\log]}$而言是困难的。我们还设计了多项式时间近似算法，并证明了 optConfVal 的不可近似性。对于在其他半环（包括热带半环、模糊半环和访问控制半环）上的这些优化问题，我们也得到了类似的复杂性结果。

*作者决定放弃按作者姓名字母顺序排列这一旧惯例，转而采用随机排序，以 r◯表示。随机排序的公开可验证记录可在 https://www.aeaweb.org/journals/policies/random-author-order/search 查阅。本文的删节版曾发表于 AAAI 2023。

1 引言

经典地，命题公式在布尔半环 $\mathbb{B} = (\{\mathrm{F}, \mathrm{T}\}, \vee, \wedge, \mathrm{F}, \mathrm{T})$ 上进行解释；该半环是逻辑真值的标准语义。在这一设定下，变量取两个值之一：T（真）或 F（假）。然而，将语义扩展到其他半环是很自然的。这里的思想是：当变量取半环 $\mathbb{K} = (K, +, \cdot, 0, 1)$ 上的值时，对逻辑公式进行解释。这类解释能够提供超越命题真或假的更丰富信息，并已应用于数据库、人工智能、逻辑和安全等多个领域（参见 [ILJ89、FR97、Zim97、CWW00、Cui02、GT20] 及其中的参考文献）。特别是，半环溯源分析已成功应用于多个软件系统，例如 Orchestra 和 Propolis（参见例如 [ADT11、DMRT14、FGT08、Gre11、Tan13]）。

文献中研究的半环包括维特比半环、模糊半环、最小-最大半环或访问控制半环，以及热带半环。在维特比半环 $\mathbb{V} = ([0,1], \text{最大}, \cdot, 0, 1)$ 上的语义具有数据库溯源方面的应用，其中 $x \in [0,1]$ 被解释为置信度分数 [GT20, GKT07, Tan17, GM21]，也应用于概率句法分析、概率约束满足问题和隐马尔可夫模型 [Vit67, KM03, BMR95]。访问控制半环可作为安全规范中的工具 [GT20]。其他值得关注的半环包括热带半环，它用于成本分析以及最短路径算法的代数表述 [Moh02]；此外，模糊半环还用于模糊约束满足问题的相关场景 [BMR95]。

基于布尔解释的优化问题一直是许多应用领域和基础领域的核心。事实上，经典的可满足性问题是确定公式 $\phi(x_1, \cdots, x_n)$ 是否在布尔半环上具有一种使其值为真的解释/赋值。尽管半环语义自然地出现在各种应用中，但除布尔半环之外，针对半环的优化问题尚未受到太多关注。

本文介绍并研究了半环语义下优化问题的复杂性。设 $\mathbb{K} = (K, +, \cdot, 0, 1)$ 是一个在 K 上具有全序的半环，且 φ 是定义在变量集 X 上的命题公式。$\mathbb{K}$-解释 π 是一个从 X 到 K 的函数。此类解释可以自然地扩展到公式 φ，记作 $\mathrm{Sem}(\varphi, \pi)$。我们研究以下计算问题：给定一个定义在变量集 X 上、处于否定范式的命题公式 φ，求所有可能解释 π 下 $\mathrm{Sem}(\varphi, \pi)$ 的最大值。我们称此问题为 optSemVal。一个相关问题记作 optSem，要求求出一个使 $\mathrm{Sem}(\varphi, \pi)$ 最大化的解释 π。关于这些问题的精确定义，请参见第 2 节。

关于在半环上定义约束满足问题（CSP）的概念，并在一般框架下研究局部一致性算法，已有大量工作 [Bis04, BG06, BMR95, BMR97, BMR+99, MRS06]。这些工作并未涉及解释，也没有关注上述问题的计算复杂性。相关地，近期有研究探讨了半环上的和积问题的计算复杂性 [EK21]。然而，他们研究的问题与我们的问题不同。据我们所知，我们所考虑的半环上的优化问题 optSem 和 optSemVal 此前尚未得到研究，也没有关于其计算复杂性的刻画。

1.1 我们的成果

我们从算法和复杂性理论视角，全面研究 optSem 及相关问题 optSemVal 在多种半环上的计算复杂性，例如维特比半环、热带半环、访问控制半环和模糊半环。当底层半环为维特比半环时，我们将这些问题称为 optConf 和 optConfVal。我们的结果可概括如下：

1. 我们证明，optConf 和 optConfVal 均属于复杂性类 $\mathrm{FP}^{\mathrm{NP}}$。关键的观察是，尽管 π 将 X 映射为范围 $[0,1]$ 内的实数，optConfVal 的解仍可用多项式数量的比特表示。随后，我们利用与 Farey 数列的联系，推导出一种只需多项式次 NP 调用的算法（定理 3.2）。

2. 对于合取范式公式，我们建立了 optConfVal 关于最大可满足子句数量的上界（定理 3.7）。

3. 我们还建立了 optConfVal 和 optConf 的复杂性下界。具体而言，我们证明这两个问题对于复杂性类 $\mathrm{FP}^{\mathrm{NP}[\log]}$ 都是困难的。为此，我们给出了从 MaxSATVal 到 optConfVal 的归约；该归约关键依赖于上述 optConfVal 关于最大可满足子句数量的上界（定理 3.9）。

4. 我们为 optConfVal 设计了一种多项式时间近似算法，并建立了一个不可近似性结果。具体而言，对于含有 m 个子句的 3-CNF 公式，我们设计了一种 0.716^m-近似算法，并证明除非 P = NP，否则近似因子无法改进至 0.845^m （定理 4.3 和 4.5）。

5. 最后，我们证明，对于访问控制半环，这些优化问题的复杂性等价于布尔半环上相应问题的复杂性（定理 5.3）。

备注 1. 由于维特比半环与热带半环可通过映射 $x \leftrightarrow -\ln x$ 同构，因此针对维特比半环建立的结果同样适用于热带半环。模糊半环可视为具有相同代数结构的访问控制半环的“无限细化”，因此我们针对访问控制半环建立的结果同样适用于模糊半环。

文章结构。本文其余部分的组织如下。我们在第 2 节给出必要的符号和定义。第 3 节详细介绍关于 optConf 和 optConfVal 计算复杂性的结果。第 4 节讨论近似算法以及 optConfVal 的近似困难性。在第 5 节，我们给出访问控制半环上优化问题的复杂性结果。最后，我们在第 6 节总结全文。

2 预备知识

我们假设读者熟悉半环的定义。我们用 $\mathbb{K} =(K, +, \cdot, 0, 1)$ 表示一个一般的半环，其中 K 是其基础集合。为了对 $\mathbb{K}$ 上的公式进行解释，我们将添加一个“否定”函数 $\neg : K \to K$。我们假设 $\neg$ 是一个双射，使得 $\neg(\neg(x)) = x$，且 $\neg(0) = 1$。为便于表述，我们使用最自然的否定函数（具体取决于半环）。不过，我们的许多结果对于非常一般的否定解释同样成立。最后，由于我们关注的是优化问题，我们还将假设 K 的元素上具有一个（自然的）全序。

对于一个变量集合 $X = \{x_1, x_2, \ldots x_n\}$，我们关联集合 $\overline{X} = \{\neg x_1, \ldots, \neg x_n\}$。我们将 $X \cup \overline{X}$ 称为文字，而我们所考虑的公式是在 $X \cup \overline{X}$ 上的否定范式命题公式。我们也将否定范式命题公式 φ 视为一棵有根有向树，其中每个叶节点都标记为一个文字、1 或 0，每个内部节点都标记为合取（$\wedge$）或析取 $\vee$。注意，将 φ视为树可以确保其大小与其字符串表示相近。我们将表示公式 φ 的树称为公式树，并记作 T_φ。对于否定范式命题公式 φ（x_1，$\cdots$，x_n），我们用 m表示公式的大小，即每个变量及其否定出现的总次数

。当 φ（x_1, $\cdots x_n$）为合取范式时， m 表示子句的数量。我们通过将变量映射到 K，并以自然的方式扩展，从而在半环 $\mathbb{K}$ 上解释命题公式。形式上， $\mathbb{K}$-解释是一个函数 $\pi\colon X \to K$。我们将 π 扩展到任意的否定范式命题公式 φ，并记作 Sem（φ, π）（Sem 表示“语义”），如下所示。

- $\mathsf{Sem}(x, \pi) = \pi(x)$
- $\mathsf{Sem}(\neg x, \pi) = \neg(\pi(x))$
- $\mathsf{Sem}(\alpha \vee \beta, \pi) = \mathsf{Sem}(\alpha, \pi) + \mathsf{Sem}(\beta, \pi)$
- $\mathsf{Sem}(\alpha \wedge \beta, \pi) = \mathsf{Sem}(\alpha, \pi) \cdot \mathsf{Sem}(\beta, \pi)$

2.1 优化问题与复杂性类

对于公式 φ，我们将 optSemVal(φ) 定义为

$$\mathsf{optSemVal}(\varphi) = \max_{\pi}\{\mathsf{Sem}(\varphi, \pi)\},$$

其中，max 取遍所有可能的从 X 到 K 的 $\mathbb{K}$-解释。

定义 2.1（optSem 和 optSemVal）。给定一个处于否定范式的命题公式 φ，optSemVal 问题是计算 optSemVal(φ)。optSem 问题是计算一个能使 optSemVal(φ) 达到最大值的 $\mathbb{K}$-解释，即输出 π^*，使得 optSemVal(φ) $=$ Sem(φ, π^*)。

请注意，当 $\mathbb{K}$ 为布尔半环（采用 $0 < 1$ 阶以及标准的否定解释）时，optSemVal 就是众所周知的可满足性问题：当且仅当 optSemVal(φ) $= 1$ 时，公式 φ 可满足。此外，如果公式 φ 可满足，optSem 问题就是输出一个满足赋值。

在本文中，我们考虑以下半环。

1. 维特比半环 $\mathbb{V} = ([0,1], \max, \cdot, 0, 1)$。如前所述，维特比半环应用于数据库溯源，其中 $x \in [0,1]$ 被解释为置信度分数；它还应用于概率句法分析、概率约束满足问题和隐马尔可夫模型。

2. 热带半环 $\mathbb{T} = (\mathbb{R} \cup \{\infty\}, \min, +, \infty, 0)$。
   热带半环可通过映射 $x \leftrightarrow -\ln x$ 与维特比半环同构。

3. 模糊半环 $\mathbb{F} = ([0,1], \max, \min, 0, 1)$。

4. 访问控制半环 $\mathbb{A}_k = ([k], \max, \min, 0, k)$。直观地说，每个 $i \in [k]$ 都与一个具有自然顺序的访问控制级别相关联。这里，0 表示公开访问，而 n 表示完全无法访问。$[k]$ 是集合 $\{0 < 1 < \cdots < k\}$。

我们的重点将放在 Viterbi 半环上 optSem 和 optSemVal 问题的复杂性。我们分别将相应的计算问题称为 optConf 和 optConfVal。在这种情况下，我们将扩展后的解释函数称为 Sem asConf。

定义 2.2（最大可满足性和最大可满足值）。给定一个合取范式形式的命题公式 φ，最大可满足性问题是求出一个赋值 φ，使其满足的子句数量最多。给定一个合取范式形式的命题公式 φ，最大可满足值问题是求出 φ 中可满足的子句最大数量。

我们需要一个函数问题之间归约的概念。我们采用 Krentel [Kre88] 引入的度量归约概念。

定义 2.3（度量归约）。对于两个函数 f、g：$\{0,1\}^* \to \{0,1\}^*$，如果存在多项式时间可计算函数 h_1 和 h_2，其中 h_1：$\{0,1\}^* \to \{0,1\}^*$（归约函数），且 h_2：$\{0,1\}^* \times \{0,1\}^* \to \{0,1\}^*$，使得对于任意 x，$f(x) = h_2(x, g(h_1(x)))$，我们称 f 度量归约到 g。

定义 2.4。对于函数 t：$\mathbb{N} \to \mathbb{N}$，$\mathrm{FP}^{\mathrm{NP}[t(n)]}$ 表示能够在多项式时间内通过对 NP 预言机进行 $O(t(n))$ 次查询而求解的函数类，其中 n 是输入的大小。当 $t(n)$ 是某个多项式时，我们用 $\mathrm{FP}^{\mathrm{NP}}$ 表示该类。

度量归约用于定义函数类 $\mathrm{FP}^{\mathrm{NP}}$和 $\mathrm{FP}^{\mathrm{NP}[\log]}$的完全性和困难性概念。Krentel 的以下结果 [Kre88] 刻画了最大可满足值问题的复杂性。

定理 2.5（[Kre88]）。最大可满足值在度量归约下对于 $\mathrm{FP}^{\mathrm{NP}[\log]}$ 是完全的 .

以下 命题是我们结果的一个基本组成部分。它可以使用基本微积分加以证明 s.

命题 1。设 $f(x) = x^a(1-x)^b$，其中 a、b为非负整数，则 $f(x)$ 在定义域 $[0,1]$ 上的最大值在 $x = \frac{a}{a+b}$时取得。该函数的最大值为 $\left(\frac{a}{a+b}\right)^a \left(\frac{b}{a+b}\right)^b$。

3 置信度最大化的计算复杂性

对于维特比半环上的语义，我们假设标准的封闭世界语义，并使用否定函数 $\neg(x) = 1 - x$。因此，我们有 $\mathrm{Conf}(\neg x, \pi) + \mathrm{Conf}(x, \pi) = 1$。不过，我们的上界证明对任何合理的否定函数都成立。我们将在注释 2 中讨论这一点。

由于 $\mathrm{Conf}(\varphi, \pi)$ 可以在多项式时间内计算，optConf 至少与 optConfVal 一样困难。下面的观察表明，计算 optConfVal 和 optConf 都是 NP 困难的。

观察3.1。对于公式 φ，$\mathrm{optConfVal}(\varphi) = 1$ 当且仅当 φ 可满足。因此，optConf 和 optConfVal 都是 NP 困难的。

虽然 optConf 和 optConfVal 都是 NP 困难的，但我们希望了解它们与其他最大化问题之间的关系。在优化问题的研究中，复杂性类 $\mathrm{FP}^{\mathrm{NP}}$ 和 $\mathrm{FP}^{\mathrm{NP}[\log]}$ 发挥着关键作用。在本节中，我们将研究这些问题相对于复杂性类 $\mathrm{FP}^{\mathrm{NP}}$ 和 $\mathrm{FP}^{\mathrm{NP}[\log]}$ 的上界和下界。

一个示例。我们先给出一个示例，以说明上界背后的思路。考虑公式 $\phi(x_1, x_2) = (x_1) \wedge (x_2) \wedge (\neg x_1 \vee \neg x_2)$。显然，该公式不可满足。在维特比半环上，$\mathrm{optConfVal} = \max_{x_i \in [0,1]} \{x_1 x_2 (1 - x_1)\ x_1 x_2 (1 - x_2)\}$ 由分配.....

活性。根据命题 1，当 $x_1 = 1$ 和 $x_2 = 0.5$ 或 $x_1 = 0.5$ 和 $x_2 = 1$时，该值达到最大，最优值为 0.25。在下一节中，我们将说明 optConfVal 的计算可归约为

在一组多项式项上进行最大化，其中每个多项式项都对应一个我们定义的证明树。尽管多项式项的数量可能呈指数级增长，我们使用 NP 预言机通过二分搜索找到能给出最大值的项。

3.1 一般公式的上界

我们证明，optConfVal 和 optConf 可以通过对 NP 语言进行预言机查询，在多项式时间内计算。

定理 3.2. 否定范式公式的 optConfVal 属于 $\mathrm{FP}^{\mathrm{NP}}$。

证明思路：为了证明 optConfVal 属于 $\mathrm{FP}^{\mathrm{NP}}$，我们使用基于 NP 语言的二分搜索策略。一个难点是，置信度值可能是 $[0,1]$ 中的任意实数，因此事先我们可能无法限制二分搜索查询的次数。然而，我们首先论证，对于任意大小为 m、含有 n 个变量的公式 φ，optConf(φ) 是形如 A/B 的分数，其中 $1 \le A \le B \le 2^{nm\log m}$。这种形式的有序分数称为阶为 $2^{nm\log m}$ 的 Farey 数列（记作 $\mathcal{F}_{2^{nm\log m}}$）。因此，我们的任务是在 $\mathcal{F}_{2^{nm\log m}}$ 上进行二分搜索，其时间复杂性为 $O(nm\log m)$。然而，对于 Farey 数列 $\mathcal{F}_N$ 中的未知元素，一般而言，尚不清楚如何以 $O(\log N)$ 的时间复杂性进行二分搜索。我们通过使用 NP 预言机辅助二分搜索来克服这一困难。下面将给出详细说明。

定义 3.3. 设 $\varphi(x_1,\cdots,x_n)$ 是一个大小为 m 的否定范式命题公式。设 T_φ 是其公式树。T_φ 的一棵证明树 T 是通过以下过程得到的子树：对于每个 OR 节点v，从 v 的子树中选取一棵。对于每个 AND 节点 v，保留所有子树。

注意，在证明树中，每个 OR 节点都只有一个子节点。

定义 3.4. 设 $\varphi(x_1,\cdots,x_n)$ 是一个否定范式命题公式，设 T 是一棵证明树。我们通过对每个节点 v 处的子树（记作 p_v）归纳定义一个多项式，进而定义证明树多项式 p_T：如果节点 v 是变量 x_i，则该多项式为 x_i；如果它是 $\neg x_i$，则该多项式为 $(1-x_i)$。如果 v 是一个 AND 节点，其子节点为 v_1、.....、v_s，那么 $p_v = \prod_{i=1}^{s} p_s$。如果 v 是一个 OR 节点，且有一个子节点u，那么 $p_v = p_u$。

断言 3.4.1. 设 $\varphi(x_1,\cdots,x_n)$ 是一个否定范式下的命题公式，并设 T 是 φ 的证明树。

1. 证明树多项式 p_T 具有以下形式

$$\prod_{i=1}^{n} x_i^{a_i}(1-x_i)^{b_i}$$

其中 $0 \le a_i + b_i \le m$.

2. 对于一个 $\mathbb{V}$-解释 π,

$$\mathsf{Conf}(T,\pi) = p_T\left(\pi(x_1),\ldots,\pi(x_n)\right).$$

3. optConf(T) 和 optConfVal(T) 均可在多项式时间内计算。

4. $\mathsf{optConfVal}(T) = \Pi_{i=1}^{n}\left(\frac{a_i}{a_i+b_i}\right)^{a_i}\left(\frac{b_i}{a_i+b_i}\right)^{b_i}$.

证明。条目 (1) 可由证明树多项式的定义、一个简单的归纳法，以及公式 φ 的大小为 m 这一事实得出。条目 (2) 则由定义得出。

注意，最大化多项式 $\pi_{i=1}^{n} x_i^{a_i}(1-x_i)^{b_i}$，可以通过分别最大化各个项 $x_i^{a_i}(1-x_i)^{b_i}$ 来实现。根据命题 1，这种形式的多项式在 $x_i = \frac{a_i}{a_i+b_i}$ 处取得最大值。因此，解释 $\pi(x_i) = \frac{a_i}{a_i+b_i}$ 是一个最优的 $\mathbb{V}$-解释，并且可以在多项式时间内计算。由于 $0 \le a_i + b_i \le m$，optConfVal 也可以在多项式时间内计算。条目 (4) 可由条目 (3) 推出：将值 $\pi(x_i)$ 代入多项式 p_T 即可。 □

下一个断言将公式 φ 的 optConf 与其证明树的 optConf 联系起来。该断言的证明直接来自证明树的定义和标准归纳法。

断言 3.4.2。对于公式 φ,

$$\mathsf{optConfVal}(\varphi) = \max_{T} \mathsf{optConfVal}(T)$$

其中，最大值取遍 T_φ 的所有证明树 T。若 T^* 是使 optConf(T) 达到最大值的证明树，则 optConf(T^*) = optConf(φ)。

上述命题指出，可以通过遍历 φ 的所有证明树 T，并计算 optConf(T)，来求出 optConf(φ)。由于证明树的数量可能呈指数级增长，这一过程将耗费指数时间。我们的任务是证明这一过程可以在 $\mathrm{FP}^{\mathrm{NP}}$ 中完成。为此，我们将提出一个命题，以限制 optConfVal(φ) 可能取到的值。我们需要用到 Farey 数列的概念。

定义 3.5. 对于任意正整数 N，阶为 N 的 Farey 数列（记作 $\mathcal{F}_N$）是由所有满足 $0 < p < q \le N$ 的既约分数 p/q 按递增顺序排列而成的集合。

命题 3.5.1. 1. 对于命题公式 $\varphi(x_1, \cdots, x_n)$，optConfVal($\varphi$) 属于 Farey 数列 $\mathcal{F}_{2^{nm\log m}}$。

2. 对于任意两个分数 u和 v，取自 $\mathcal{F}_{2^{nm\log m}}$，$|u-v| \ge 1/2^{2nm\log m}$

证明。根据断言 3.4.2，对于某个证明树 T，optConfVal(φ) 等于 optConfVal(T)。根据断言 3.4.1 的第 (4) 项，该值是若干分数的乘积，其中每个分数的分母都具有 $(a_i+b_i)^{a_i+b_i}$ 的形式，且 a_i 和 b_i 是非负整数。由于 $a_i + b_i \le m$，每个分母都至多为 m^m，因此乘积的分母有上界 $m^{nm} = 2^{nm\log m}$。由于分子至多等于分母，结论得证。

为证明第二部分，设 $u = p_1/q_1$ 和 $v = p_2/q_2$、$u > v$。现在 $u - v = (p_1q2 - p_2q_1)/q_1q_2$。由于 q_1、$q_2 \le 2^{nm\log m}$，我们有 $u - v > p_1q_2 - p_2q_1/2^{2nm\log m}$。由于 p_1、p_2、q_1、q_2 都是整数，$p_1q_2 - p_2q_1 \ge 1$。因此 $|u-v| \ge 1/2^{2nm}$对数m。

□

考虑以下语言

$$L_{opt} = \{\langle \varphi, v\rangle \mid \mathsf{optConfVal}(\varphi) \ge v\}$$

断言 3.5.2。 L_{opt} 属于 NP。

证明。考虑以下非确定性机器 M。在输入 φ, M 时，猜测一个关于φ的证明树 T：对于每个 OR 节点，非确定性地选取其中一棵子树。对于 T，计算 optConfVal(T)，并在 optConfVal(T) $\ge v$ 时接受。利用断言 3.4.1 的第 (3) 项，可以在多项式时间内完成此操作。该算法的正确性由断言 3.4.2 得出。

□

我们需要一种方法，给定两个分数 u 和 v 以及一个整数 N，输出一个分数 p/q： $u \le p/q \le v$，且 $p/q \in \mathcal{F}_N$。我们给出一个 $\mathrm{FP}^{\mathrm{NP}}$ 算法，通过向 NP 预言机发起 $O(N)$ 次查询来实现这一目标。我们首先定义 NP 语言 L_{farey}。为此，我们固定一种使用二进制字母表对分数进行编码的标准方式。对于 $\mathcal{F}_N$ 中的任意分数，这种编码的位表示长度为 $O(\log\ N)$。

$$L_{farey} = \{\langle N, u, v, z\rangle \mid \exists z'; u \le zz' \le v \ \&\ zz' \in \mathcal{F}_N\}$$

以下命题很容易看出。

命题 3.5.3. $L_{farey} \in \mathrm{NP}$。

现在我们可以证明定理 3.2 了。

证明。（定理 3.2）。该算法在范围 $[0,1]$ 内执行二分搜索，通过自适应地向 NP 语言 L_{opt} 查询 $\langle\varphi$、$v\rangle$，并从 $v = 1$开始。在二分搜索的每次迭代中，我们都有一个区间 $I = [I_l, I_r]$，且满足不变式 $I_l \le \mathrm{optConfVal}(\varphi) < I_r$。当区间 $[I_l, I_r] = 1/2^{2nm\log m}$的大小达到时，二分搜索便停止。由于每次迭代都会将区间大小缩小一半，因此搜索在向 L_{opt}进行 $2nm\log\ m$ 次查询后停止。不变式确保 $\mathrm{optConfVal}(\varphi)$ 位于该区间内。此外，$\mathrm{optConfVal}(\varphi) \in \mathcal{F}_{2^{nm\log m}}$（根据命题 3.5.1 的第 (1) 项），且该区间内没有其他来自 $\mathcal{F}_{2^{nm\log m}}$的分数（根据命题 3.5.1 的第 (2) 项）。现在，通过使用 $N = 2^{nm\log m}$、$u = I_l$、$v = I_r$向 L_{farey}进行 $O(nm\log\ m)$ 次查询，我们可以构造出 $\mathcal{F}_{2^{nm\log m}}$中位于 I_l和 I_r之间的唯一分数的二进制表示，该分数即为 $\mathrm{optConfVal}(\varphi)$。

□

接下来我们将证明，通过查询 NP 预言机，也可以在多项式时间内计算最优的 $\mathbb{V}$-解释。

定理 3.6. 对于否定范式中的公式，optConf 可以在 $\mathrm{FP}^{\mathrm{NP}}$ 中计算。

证明。设 φ 是一个否定范式中的命题公式。我们使用 NP 语言对 φ 的证明树编码进行前缀搜索，以确定一棵证明树 T，使得 $\mathrm{optConfVal}(\varphi) = \mathrm{optConfVal}(T)$。为此，我们固定一种 φ 的证明树编码。考虑以下 NP 语言 L_{pt}:

$$\{\langle\varphi, v, z\rangle \mid \exists z' : zz' \text{encodes a proof tree } T \text{ of } \varphi \\ \& \ \mathsf{optConfVal}(T) = v\}$$

命题 3.6.1. L_{pt} 属于 NP。

证明。考虑一台非确定性机器，它猜测一个 z'，验证 zz' 是否编码了一棵φ的证明树 T，并在 $\mathrm{optConfVal}(T) = v$ 时接受。根据断言 3.4.1 的第 (3) 项，可以在多项式时间内计算 $\mathrm{optConfVal}(T)$。

□

为完成定理 3.6 的证明，给定一个命题公式 φ，我们首先使用定理 3.2 中的 $\mathrm{FP}^{\mathrm{NP}}$ 算法计算 $v^* = \mathrm{optConfVal}(\varphi)$。现在，我们可以使用语言 L_{pt} 进行前缀搜索，构造一棵证明树 T，使得 $\mathrm{optConfVal}(T) = v^*$。接着，根据断言 3.4.1，我们可以计算一个 $\mathbb{V}$-解释 π^*，使得 $\mathrm{Conf}(T, \pi^*) = v^*$。因此，根据断言 3.4.2，$\pi^*$ 是 φ 的一个最优 $\mathbb{V}$-解释。 □

注释 2。我们重新审视否定语义。如前所述，假设采用封闭世界语义，我们有 $\neg(x) = 1 - x$。我们注意到，要使上述证明成立，这一假设并非严格必要。回顾断言 3.4.1 的第 (1) 项，其中指出证明树多项式具有 $\prod x_i^{a_i}(1 - x_i)^{b_i}$ 的形式。对于一般的否定函数 $\neg$，证明树多项式具有 $\prod x_i^{a_i}(\neg(x_i))^{b_i}$ 的形式。现在，如果能够找到项 $x^a(\neg(x))^b$ 的最大值，例如当 $\neg$ 是一个显式的可微函数时，结论便成立。

3.2 与合取范式公式的最大可满足性的关系

本节研究合取范式公式的 optConfVal 问题，并阐明它与最大可满足性问题的关系。我们首先利用可满足子句的最大数量，给出 optConfVal(φ) 的一个上界。在此基础上，我们将在第 3.3 节证明，合取范式公式的 optConfVal 对复杂性类 $\mathrm{FP}^{\mathrm{NP}[\log]}$ 是困难的。

我们首先定义一些将在本小节及下一小节中使用的符号。令 $\varphi(x_1,\cdots x_n) = C_1 \wedge \cdots \wedge C_m$ 为一个合取范式公式，并令 π^* 为一个最优的 $\mathbb{V}$-解释。对于 φ 中的每个子句 C，令 $\pi^*(C)$ 为该解释取得的值，即 $\pi^*(C) = \mathsf{Conf}(C,\pi^*)$。注意，由于 C 是文字的析取，$\pi^*(C) = \max_{\ell\in C}\{\pi^*(\ell)\}$。对于子句 C，令

$$\ell_C = \mathrm{argmax}_{\ell\in C}\{\pi^*(\ell)\}$$

在上述情形中，如果存在多个最大值，我们取最小的文字作为 ℓ_C（假设存在一个阶 $x_1 < \neg x_1 < x_2 < \neg x_2 \cdots < x_n < \neg x_n$）。注意，由于我们是在维特比半环上运算，$\mathsf{Conf}(C,\pi^*) = \pi^*(\ell_C)$。若 $\ell_C = \ell$，则文字 ℓ 是子句 C 的最大化文字。

由于 φ 是一个合取范式公式，对于任意 $\mathbb{V}$-解释 π $\mathsf{Conf}(\varphi,\pi)$ 都具有 $\Pi_{i=1}^{m}\mathsf{Conf}(C_i,\pi)$ 的形式。给定来自 φ 的子句集合 $\mathcal{D}$，$\mathcal{D}$ 对 $\mathsf{Conf}(\varphi,\pi)$ 的贡献定义为 $\Pi_{c\in\mathcal{D}}\mathsf{Conf}(C,\pi)$。

下述定理利用最大可满足值给出了 optConfVal(φ) 的一个上界。这是本节的主要结果。

定理 3.7. 设 $\varphi(x_1,\cdots,x_n)$ 是一个含有 m 个子句的合取范式公式。设 r 是可满足的子句数的最大值。那么 $\mathsf{optConfVal}(\varphi) \le 1/4^{(m-r)}$。

证明。设 π^* 是 φ 的一个最优 $\mathbb{V}$-解释。若子句 C 满足 $\pi^*(C) < 1/2$，则称其为低子句；若 C 满足 $\pi^*(C) > 1/2$，则称其为高子句；若 C 满足 $\pi^*(C) = 1/2$，则称其为中性子句。分别用 L、H 和 N 表示低子句、高子句和中性子句的数量。

我们先给出以下断言，它将中性子句的数量和高子句的数量与 r联系起来。

断言 3.7.1. $\frac{N}{2} + H \le r$

证明。假设低子句的数量严格小于 $m-r$，那么高子句的数量就大于 r。

对于变量 x，令

$$p_x = |\{C \mid C \text{ is neutral and } \ell_C = x\}|$$

and

$$q_x = |\{C \mid C \text{ is neutral and } \ell_C = \neg x\}|$$

也就是说，p_x 是以 x 为最大化文字的中性子句数量，而 q_x 是以 $\neg x$为最大化文字的中性子句数量。

考虑根据以下三条规则构造的真值赋值：(1) 对于每个高子句 C，将 ℓ_C 设为真，将 $\neg\ell_C$ 设为假；(2) 对于每个变量 x，如果 p_x 或 q_x 不为零，那么若$p_x \ge q_x$，就将 x设为真；否则将 x 设为假。(3) 将所有剩余变量一致地任意赋值为真或假。

我们认为这是一个一致的赋值：即，对于每个文字 ℓ，ℓ 和 $\neg\ell$ 不会被赋予相同的真值。考虑一个文字 ℓ。如果存在一个高子句 C，使得 $\ell = \ell_C$，则该文字被赋予真值真，而 $\neg\ell$被赋予假。在这种情况下，由于 $\pi^*(\ell) > 1/2$，$\pi^*(\neg\ell) < 1/2$。因此，$\neg\ell$不可能

成为任何高子句的最大化文字，因此规则（1）不会将真赋予 $\neg\ell$。此外，由于$\pi^*(\ell) > 1/2$，不存在中性子句 D，使得 $\ell = \ell_D$ 或 $\neg\ell = \ell_D$。因此，规则（2）不会为 ℓ 或 $\neg\ell$中的任何一个赋予真值。由于 ℓ 和 $\neg\ell$已被赋予真值，规则（3）不会为 ℓ或 $\neg\ell$赋予真值。

考虑一个变量 x，其中 p_x 或 q_x 至少有一个不为零。在这种情况下，x 或 $\neg x$ 是某个中性子句的最大化文字。因此 $\pi^*(x) = \pi^*(\neg x) = 1/2$，并且 x 和 $\neg x$都不是高子句的最大化文字。因此，规则（1）不会为 x 或 $\neg x$赋予真值。现在，x 当且仅当 $p_x \geq q_x$时为真，因此赋予 x（以及 $\neg x$）的真值是一致的。由于规则（3）会为规则（1）和规则（2）未涵盖的文字一致地赋予真值，因此所构造的赋值是一致的赋值。

对于每个高子句 C，文字 ℓ_C 都被设为真。因此，该赋值满足所有高子句。考虑一个变量 x，并令 $\mathcal{D}$ 为这样的中性子句的（非空）集合：其中 x 或 $\neg x$是最大化文字。由于仅当 $p_x \geq q_x$时 x 才被赋为真，因此 $\mathcal{D}$ 中至少一半的子句得到满足。因此，该赋值至少满足 $H + \frac{N}{2}$个子句。由于 r 是可满足子句数量的最大值，结论得证。 □

对于一个 文字 ℓ，令 a_ℓ 为满足 ℓ 是最大化文字的低子句 C 的数量，即 e,

$$a_\ell = |\{C \mid C \text{ is a low-clause and } \ell_C = \ell\}|,$$

and

$$b_\ell = |\{C \mid C \text{ is a high-clause and } \ell_C = \neg\ell\}|,$$

我们证明 a_ℓ 与 b_ℓ之间存在如下关系。

命题 3.7.2. 对于每个文字 ℓ，$a_\ell \leq b_\ell$。

证明。

$$\mathsf{Conf}(\varphi, \pi) = \Pi_i \mathsf{Conf}(\varphi_{|x_i}, \pi) \tag{1}$$

现在假设对于某个文字 ℓ，$a_\ell > b_\ell$。令 x_j 为与文字 ℓ对应的变量。注意

$$\mathsf{Conf}(\varphi_{|x_j}, \pi^*) = \pi^*(\ell)^{a_\ell} \times (1 - \pi^*(\ell))^{b_\ell}$$

其中 $\pi(\ell) < 1/2$。考虑一个新的解释 π'，其中 $\pi'(\ell) = 1 - \pi^*(\ell)$，并且对于所有其他文字，$\pi'$ 的值与 π^* 的值相同。现在

$$\begin{aligned}
\frac{\mathsf{Conf}(\varphi_{|x_j}, \pi')}{\mathsf{Conf}(\varphi_{|x_j}, \pi^*)} &= \frac{\pi'(\ell)^{a_\ell} \times (1 - \pi'(\ell))^{b_\ell}}{\pi(\ell)^{a_\ell} \times (1 - \pi(\ell))^{b_\ell}} \\
&= \frac{(1 - \pi(\ell))^{a_\ell} \times \pi(\ell)^{b_\ell}}{\pi(\ell)^{a_\ell} \times (1 - \pi(\ell))^{b_\ell}} \\
&= \left(\frac{(1 - \pi(\ell)}{\pi(\ell)}\right)^{a_\ell - b_\ell} > 1
\end{aligned}$$

最后一个不等式成立是因为 $\pi(\ell) < 1/2$，以及 $a_\ell > b_\ell$ 的假设。由于对于每个 $x \neq x_j$，$\mathsf{Conf}(\varphi_{|x}, \pi^*) = \mathsf{Conf}(\varphi_{|x}, \pi')$，将上述不等式与方程（1）结合，我们得到 $\mathsf{Conf}(\varphi, \pi') > \mathsf{Conf}(\varphi, \pi^*)$，因此 π^* 不是最优的 $\mathbb{V}$-解释。这就产生了矛盾。因此 $a_\ell \leq b_\ell$ □

接下来，我们将界定中性子句和低子句对 $\mathsf{optConfVal}(\varphi)$ 的贡献。对于每个中性子句C，$\pi^*(C) = 1/2$，因此我们有如下观察。

观察 3.8. 中性子句对 Conf(φ, π^*) 的贡献恰好为 $1/2^N$。

我们证明以下断言。

断言 3.8.1。

$$\mathsf{Conf}(\varphi, \pi^*) = \prod_{\ell} \left(\pi^*(\ell)^{a_\ell} \times (1 - \pi^*(\ell))^{b_\ell} \right) \times \frac{1}{2^N}$$

证明。根据观察 3.8，中性子句对 Conf(φ, π^*) 的贡献为 $1/2^N$。接下来我们证明，所有高子句和低子句的贡献恰好为。

$$\prod_{\ell} \pi^*(\ell)^{a_\ell} \times (1 - \pi^*(\ell))^{b_\ell}.$$

为此，我们首先断言，在上述乘积中，ℓ 和 $\neg\ell$ 恰有一个作出贡献。只需证明，对于每个文字 ℓ，a_ℓ（分别为 b_ℓ）或 $a_{\neg\ell}$（$b_{\neg\ell}$）恰有一个为零。假设 $a_\ell \neq 0$，在这种情况下，$\neg\ell$ 不可能是任何低子句的最大化文字。因此 $a_{\neg\ell} = 0$。假设 $b_\ell \neq 0$，那么 $\neg\ell$ 是某个高子句的最大化文字，因此 $\pi^*(\neg\ell) > 1/2$，且 $\pi^*(\ell) \leq 1/2$。如果 $b_{\neg\ell} \neq 0$，那么 ℓ 必须是某个高子句的最大化文字；但这是不可能的，因为 $\pi^*(\ell) \leq 1/2$。因此 $b_{\neg\ell} = 0$。

令 Z 为满足 $a_\ell > 0$的文字集合 ℓ。现在，数量 $\prod_{\ell \in Z} \pi^*(\ell)^{a_\ell} \times (1 - \pi^*(\ell))^{b_\ell}$表示所有低子句和 $\sum_{\ell \in Z}$ 个高子句的贡献。对于其余所有高子句，都存在一个文字 ℓ，使得 $\ell \notin Z$ 且 $b_\ell \neq 0$。所有剩余高子句的贡献为$\prod_{\ell \notin Z}(1 - \pi(\ell))^{b_\ell}$。由于 $a_\ell = 0$ 对 $\ell \notin Z$成立，该数量等于 $\prod_{\ell \notin Z} \pi^*(\ell)^{a_{\ell)}} \times (1 - \pi(\ell))^{b_\ell}$。 □

最后，我们可以完成定理 3.7 的证明了。对于每个文字 ℓ，根据命题 3.7.2，$a_\ell \leq b_\ell$。令 $b_\ell = a_\ell + c_\ell$，$c_\ell \geq 0$。考虑以下不等式。

$$\begin{aligned}
\mathsf{optConfVal}(\varphi) &= \mathsf{Conf}(\varphi, \pi^*) \\
&= \prod_{\ell} \left(\pi^*(\ell)^{a_\ell} \times (1 - \pi^*(\ell))^{b_\ell} \right) \times \frac{1}{2^N} \\
&= \prod_{\ell} \left(\pi^*(\ell)^{a_\ell} \times (1 - \pi^*(\ell))^{a_\ell + c_\ell} \right) \times \frac{1}{2^N} \\
&\leq \prod_{\ell} \left(\pi^*(\ell)^{a_\ell} \times (1 - \pi^*(\ell))^{a_\ell} \right) \times \frac{1}{2^N} \\
&\leq \prod_{\ell} \left(\frac{1}{4^{a_\ell}} \right) \times \frac{1}{2^N} = \frac{1}{4^{L + N/2}}
\end{aligned}$$

上述推导中，第 2 行的等式由断言 3.8.1 得出。第 4 行的不等式成立，因为 $(1 - \pi^*(\ell)) \leq 1$。最后一个不等式成立，因为 $x(1 - x)$ 在 $x = 1/2$ 处取得最大值。最后一个等式由此可得$\sum a_\ell = L$。注意，子句的数量 $m = N + H + L$，并且根据断言 3.7.1 $H + N/2 \leq r$。由此可得 $L + N/2 \geq m - r$。因此，optConfVal(φ) = Conf(φ, π^*) $\leq \frac{1}{4^{L+N/2}} \leq \frac{1}{4^{m-r}}$。 □

3.3 $\mathrm{FP}^{\mathrm{NP[log]}}$- 困难性

在本小节中，我们证明 optConfVal 对 $\mathrm{FP}^{\mathrm{NP[log]}}$ 类而言是困难的。我们通过将 MaxSatVal 归约到 optConfVal 来证明这一点。由于 MaxSatVal 是 $\mathrm{FP}^{\mathrm{NP[log]}}$ 的完全问题，因此结论成立。我们还证明，可以使用同一归约从一个最优的 $\mathbb{V}$-解释中计算出一个 MaxSat 赋值。

定理 3.9. 对于合取范式公式，MaxSatVal 度量可归约到 optConfVal。因此，对于合取范式公式，optConfVal 对 $\mathrm{FP}^{\mathrm{NP}[\log]}$ 而言是困难的。

证明。设 $\varphi(x_i, \ldots, x_n) = C_1 \wedge \ldots \wedge C_m$ 为一个公式，它在变量 $x_1, \ldots, x_n$ 上有 m 个子句。考虑公式 φ'，它包含额外的 m 个变量 $y_1, \ldots, y_m$，构造如下：对于 φ 中的每个子句 C_i，在 φ' 中加入子句 $C'_i = C_i \vee y_i$。再加入 m 个单位子句 $\neg y_i$。即

$$\varphi' = (C_1 \vee y_1) \wedge \ldots \wedge (C_m \vee y_m) \wedge \neg y_1 \wedge \cdots \wedge \neg y_m$$

命题 3.9.1。optConfVal(φ') $= \frac{1}{4^{m-r}}$，其中 r 是在 φ 中能够被满足的最大子句数。

证明。我们先证明 optConfVal(φ') $\leq \frac{1}{4^{m-r}}$，并给出一个解释 π^*，使得 Conf(φ, π^*) $= \frac{1}{4^{m-r}}$，从而证明该命题。我们声称，若 r 是在 φ 中能够被满足的最大子句数，则 $m+r$ 是在 φ' 中能够被满足的最大子句数。我们将通过反证法论证这一点。设 a 是一个在 φ' 中满足 $> m + r$ 个子句的赋值。设 s 为被设为假的 y_i 的数量。该赋值将满足 $m - s$ 个形如 $C_i \vee y_i$ 的子句。然而，被满足的形如 $C_i \vee y_i$ 的子句总数为 $> m + r - s$。因此，有 $> r$ 个形如 $C_i \vee y_i$ 的子句被满足，其中 y_i 被设为假。将该赋值限制到 x_i 后，将满足 φ 中超过 r 个子句。因此产生矛盾。

因此，根据定理 3.7，可得 optConfVal(φ') $\leq \frac{1}{4^{m-r}}$。现在我们给出一个解释 π^*，使得 Conf(φ, π^*) $= \frac{1}{4^{m-r}}$。考虑一个赋值 a $= a_1, \ldots, a_n$，它满足 r 个子句。考虑 φ' 的变量上的以下解释 π^*：若 $a_i =$ 为真，则 $\pi^*(x_i) = 1$；若 $a_i =$ 为假，则 $\pi^*(x_i) = 0$。$\pi^*(y_i) = 0$ 当且仅当 C_i 被 a 满足。否则，$\pi^*(y_i) = 1/2$。对于每个可满足的子句 C_i，Conf($C_i \vee y_i, \pi^*$) $= 1$，且 Conf($\neg y_i, \pi^*$) $= 1$。对于 φ' 中的所有其他子句 C，Conf(C, π^*) $= 1/2$。由于有 r 个子句得到满足，因此 Conf(C, π^*) $= 1/2$ 的子句数为 $2m - 2r$。因此，Conf(φ', π^*) $= \frac{1}{4^{(m-r)}}$。由此，optConfVal(φ') $= \frac{1}{4^{m-r}}$。 □

由于 optConfVal(φ') $= 1/4^{m-r}$，因此通过已知 optConfVal 可以计算 φ 的最大可满足值。 □

虽然上述定理表明可以根据 optConfVal 计算最大可满足值，但下一个定理表明，可以根据最优的 $\mathbb{V}$ 解释计算 amaxsat 赋值。

定理 3.10。最大可满足性度量可归约为 optConf。

证明。考虑与前一个定理相同的归约。我们的任务是：给定针对 φ' 的最优 $\mathbb{V}$ 解释 π，为 φ 构造一个最大可满足性赋值。根据前面的定理，Conf(φ', π) $= \frac{1}{4^{m-r}}$，其中 r 是 φ 的可满足子句的最大数量。

接下来，我们将就 $\pi(y_i)$ 和 $\pi(x_i)$ 所取的值建立一系列断言。

断言 3.10.1。对于所有 y_i；$\pi(y_i) \in \{0, 1/2\}$。

证明。考虑一个子句 $C'_i = (C_i \vee y_i)$，其中 $\ell_{C'_i} = y_i$。现在，C'_i 和子句 $\neg y_i$ 对 Conf(φ', π) 的贡献为 $\pi(y_i) \times (1 - \pi(y_i))$。由于不存在满足 $\ell_{C'_j} = y_i$ 的子句 C'_j，因此当 $\pi(y_i) = 1/2$ 时，上述值最大。现在考虑一个子句 $C'_j = (C_j \vee y_j)$，其中 $\ell_{C'_j} \neq y_j$。C'_j 和子句 $\neg y_j$ 对 Conf(φ', π) 的贡献为 $\pi(\ell_{C'_j}) \times \pi(\neg y_j)$。由于 $\ell_{C'_j} \neq y_j$，且不存在其他同时包含 y_j 或 $\neg y_j$ 的子句，因此当 $\pi(\neg y_j) = 1$ 时，上述表达式最大，从而 $\pi(y_j) = 0$。 □

C断言 3.10.2. 对于每个 i，如果 y_i 不是子句 C'_i 的最大化文字，则 $\pi(y_i) = 0$.

证明。设 C_i' 为一个子句，且 y_i 不是最大化文字。设 ℓ_j 为最大化文字。我们首先考虑 $\pi(\ell_j) < 1/2$的情况。根据前面的断言， $\pi(y_i) \in \{0, 1/2\}$；如果 $\pi(y_i) = 1/2$，则 ℓ_j 不可能是子句 C_i' 的最大化文字。因此， $\pi(y_i) = 0$。现在考虑 $\pi(\ell_j) \geq 1/2$的情况。假设$\pi(y_i) = 1/2$。此时，子句 C_i' 和 $\neg y_i$ 对 $\mathsf{Conf}(\varphi, \pi)$ 的贡献为 $\pi(\ell_j)/2$。然而，如果我们将 $\pi(y_i) = 0$，则这些子句的贡献将变为 $\pi(\ell_j)$，这将与 π的最优性矛盾。因此，根据断言 3.10.1， $\pi(y_i) = 0$。 □

断言 3.10.3。对于所有 x_i，如果 x_i 或 $\neg x_i$ 是最大化文字，则 $\pi(x_i) \in \{0, 1, 1/2\}$

证明。我们讨论 x_i 是最大化文字的情形。 $\neg x_i$ 是最大化文字的情形可通过类似论证得到。假设 x_i 是最大化文字，且 $\pi(x_i) < 1/2$ 以及 $\pi(x_i)$ 既不为 0 也不为 1。必然有 $\neg x_i$ 也是最大化文字，否则令 $\pi(x_i) = 1$ 将提高信任值。假设对于 a 个子句， x_i 是最大化文字；对于 b个子句， $\neg x_i$ 是最大化文字。如果 $a > b$，则通过将 $\pi(x_i)$ 与 $\pi(\neg x_i)$ 交换，我们可以得到一个 $\mathbb{V}$-解释。如果 a 等于b，则 $\pi(x_i)$ 必须等于 $1/2$，因为 $x^a(1-x)^a$ 针对 $x = 1/2$取得最大值。因此 $a < b$。对于每个子句C_j'，若 x_i 或 $\neg x_j$ 是最大化文字，则根据断言 3.10.2，必然有 $\pi(y_j) = 0$。令$\mathcal{C}$ 为所有子句 C_j' 与 $\neg y_j$的集合，其中 x_i 或 $\neg x_i$ 是最大化文字。这些子句对 $\mathsf{Conf}(\varphi, \pi)$ 的贡献为 $\pi(x_i)^a \times (1 - \pi(x_i))^b \times 1^{a+b}$。

现在我们构造一个新的 $\mathbb{V}$-解释 π'，它将与 π 的最优性相矛盾。对于每个以 x_i 为最大化文字的子句 $C_j' \in \mathcal{C}$， $\pi'(y_i) = 1/2$，且 $\pi'(x_i) = 0$。现在，从 $\mathcal{C}$ 到 $\mathsf{Conf}(\varphi, \pi')$ 的子句贡献为 $(\frac{1}{2})^a \times 1^b \times (\frac{1}{2})^a \times 1^b$

由于 $x^a(1-x)^b < 1/4^a$ （当 $a < b$ 时），

$$(\frac{1}{2})^a \times 1^b \times (\frac{1}{2})^a \times 1^b > \pi(x_i)^a \times (1 - \pi(x_i))^b \times 1^{a+b}$$

因此 $\mathsf{Conf}(\varphi, \pi') > \mathsf{Conf}(\varphi, \pi)$，这就产生了矛盾。因此，如果 $\pi(x_i) < 1/2$，那么 $\pi(x_i) = 0$；类似的论证表明，如果 $\pi(x_i) > 1/2$，那么 $\pi(x_i) = 1$。 □

命题 3.10.4. 对于每个满足 $\pi(x_i) = 1/2$ 的 x_i， x_i 和 $\neg x_i$ 恰好在相同数量的子句中是最大化文字。

证明。令 $\mathcal{C}$ 为满足以下条件的子句集合： x_i 或 $\neg x_i$ 是最大化文字。假设 x_i 是 a 个子句的最大化文字，而 $\neg x_i$ 是 b 个子句的最大化文字。如果 $a \neq b$， $\pi(x_i) = \frac{a}{a+b} \notin \{0, 1, 1/2\}$，这与命题 3.10.3 矛盾。 □

我们将展示如何根据 π 构造一个最大可满足性赋值：如果 $\pi(x_i) = 0$，则将 x_i 的真值设为假；否则将 x_i 的真值设为真。

根据断言 3.10.3， $\pi(x_i) = \{0, 1/2, 1\}$。令 H 为最大化文字ℓ 是一个 x-变量且 $\pi(\ell) = 1$的子句数。注意，上述真值赋值将满足所有 H 子句。令 N为最大化文字 ℓ 是一个 x-变量且 $\pi(\ell) = 1/2$的子句数。根据命题 3.10.4，恰好有 $N/2$ 个子句中正文字是最大化文字，因此我们的真值赋值会满足所有这些 $N/2$子句。因此，真值赋值满足的子句总数为 $N/2 + H$。令 Y为 y_i 是最大化文字的子句数。根据命题 3.10.1，当 y_i 是最大化文字时， $\pi(y_i) = 1/2$。因此

$$\mathsf{Conf}(\varphi', \pi) = 1^H \times (\frac{1}{2})^N \times (\frac{1}{2})^{2Y} = \frac{1}{4^{N/2+Y}} = \frac{1}{4^{m-r}}$$

最后一个等式由命题 3.9.1 得出。因此 $m-r=N/2+Y$，将其与 $m=H+N+Y$结合，我们得到 $N/2+H=r$。因此，所构造的真值赋值将满足 r 个子句，故它是一个 MaxSat 赋值。

□

4 近似 optConfVal

我们研究如何高效地近似 optConfVal。下文中， k-SAT 公式是指每个子句中恰有 k 个不同变量的合取范式公式。我们从以下命题开始。

引理 4.1. 设 $a_1,\cdots a_n$ 是一个赋值，它满足合取范式公式 $\varphi(x_1,\cdots x_n)$ 的 r 个子句。存在一个解释 π，使得 $\mathsf{Conf}(\varphi,\pi)$ 为 $\left(\frac{m-r}{m}\right)^{m-r}\left(\frac{r}{m}\right)^{r}$

证明. 若 $a_i=1$，则设 $\pi(x_i)=(1-\epsilon)$；若 $a_i=0$，则设 $\pi(x_i)=\epsilon$。对于每个满足的子句，我们得到最大值 $(1-\epsilon)$；对于每个不满足的子句，最大值为 $\geq\epsilon$。因此，此赋值所得的 optConf 为 $(1-\epsilon)^r\epsilon^{m-r}$，根据命题 1，当 $\epsilon=\frac{m-r}{m}$ 时该值最大。

□

因此，例如，若 φ 是一个 3-SAT 公式，由于随机赋值在期望意义下满足 $7/8$ 比例的子句，对于随机赋值 $r\geq 7m/8$，根据引理 4.1，有 $\mathsf{optConfVal}(\varphi)>0.686^m$。下面的引理表明，对于合取范式公式，可以根据子句大小得到 optConfVal 的更好下界。

引理 4.2. 对于每个合取范式公式 φ，$\mathsf{optConfVal}(\varphi)\geq e^{-\sum_i\frac{1}{k_i}}$，其中 k_i 是 φ中第 i 个子句的元数。

证明。考虑解释 π，它为每个变量 x_i 在区间$[0,1]$中均匀随机地选取一个值。设 φ 中的子句为 C_1、...、C_m。于是：

$$\begin{aligned}
\log\mathbb{E}[\mathsf{Conf}(\varphi,\pi)] &\geq \mathbb{E}\log\mathsf{Conf}(\varphi,\pi)\ \text{(Jensen's Inequality)}\\
&=\sum_i\mathbb{E}\left[\log\max_{\ell\in C_i}\pi(\ell)\right]\\
&=-\sum_i\int_{-\infty}^{0}\Pr\left[\log\max_{\ell\in C_i}\pi(\ell)\leq t\right]dt\\
&=-\sum_i\int_{-\infty}^{0}\Pr\left[\max_{\ell\in C_i}\pi(\ell)\leq e^t\right]dt\\
&=-\sum_i\int_{-\infty}^{0}e^{k_it}dt=-\sum_i\frac{1}{k_i}
\end{aligned}$$

因此，存在一种 π 的取值方式能够达到该信任值。 □

由此得到一种概率算法。例如，如果 φ 是一个 3-SAT 公式，那么 $\mathsf{optConfVal}(\varphi)>0.716^m$，从而改进了引理 4.1 的结果。事实上，我们可以设计一种确定性多项式时间算法，找到一个达到引理 4.2 所保证信任值的解释；该算法使用广为人知的“条件期望法”对证明中的构造进行去随机化（例如，参见 [AS08,GW94]）。

定理 4.3. 对于 optConf，存在一种多项式时间的 $e^{-m/k}$-近似算法，其中输入公式是含有 m 个子句的 k-合取范式公式。

证明。任意排列变量 $x_1, x_2, \ldots, x_n$ 后，思路是依次设定 $\pi^*(x_1)$ $\pi^*(x_2) \ldots, \pi^*(x_n)$，确保对每个 i:

$$\mathop{\mathbb{E}}_{\pi \leftarrow_U [0,1]^n} [\log \mathsf{Conf}(\varphi, \pi) \mid \pi(x_j) = \pi^*(x_j)\ \forall j \le i] \ge -\sum_i \frac{1}{k_i}. \tag{*}$$

假设已固定 $\pi^*(x_1) \ldots, \pi^*(x_{i-1})$，我们说明如何选择满足上述条件的 $\pi^*(x_i)$。我们用 $\pi_{<i}$ 表示 $\pi(x_1) \cdots \pi(x_{i-1})$。对于子句 C，令 $\alpha = \max_{\ell \in C \cap \{x_j, \bar{x}_j : j<i\}} \pi^*(\ell)$，并假设 $x_i \in C$。那么:

$$\begin{aligned}
&= -\int_{-\infty}^{0} \Pr_{\pi} \left[\log \max_{\ell \in C} \pi(\ell) \le t \mid \pi_{<i} = \pi^*_{<i}, \pi^*(x_i) = p \right] dt \\
&= -\int_{\log \max(\alpha, p)}^{0} \Pr_{\pi} \left[\log \max_{\ell \in C \cap \{x_j, \bar{x}_j : j>i\}} \pi(\ell) \le t \right] dt \\
&= -\frac{1}{k'} \left(1 - \max(\alpha, p)^{k'} \right)
\end{aligned}$$

其中 k' 是子句 C 中涉及变量 $x_{i+1}, \ldots, x_n$ 的文字数。类似地，可以在 $\bar{x}_i \in C$ 和 $C \cap \{x_i, \bar{x}_i\} = \emptyset$ 的情况下计算条件期望。

将所有子句 C 求和，可得

$$\mathop{\mathbb{E}}_{\pi} [\log \mathsf{Conf}(\varphi, \pi) \mid \pi_{<i} = \pi^*_{<i}, \pi^*(x_i) = p]$$

是 p 的连续函数，并且在至多 m 个区间上是分段多项式。在多项式时间内[1]，我们可以找到使该函数达到最大值的一个 p。由关于 i 的归纳法可知，该函数的最大值至少为 $-\sum_i \frac{1}{k_i}$，因此 (*) 成立。算法的描述至此完成。

□

接下来，我们证明近似因子 $e^{-m/k}$ 无法得到显著改进。我们将使用 Hastad [H˚as01] 关于 MaxSat 近似困难性的以下结果。

定理 4.4（[H˚as01]）。对于任意 $\varepsilon > 0$ 和任意 $k \ge 3$，区分可满足的 k-SAT 公式与具有 k 个可满足子句的 $< m$-SAT 公式 $1 - 2^{-k} + \varepsilon$ 是 NP 困难的。

现在我们可以证明以下结论。

定理 4.5。除非 P = NP，否则不存在适用于 k-SAT 公式的 optConf 多项式时间 $\frac{1}{4^{m(2^{-k}-\varepsilon)}}$-近似算法。

证明。假设存在这样的近似算法 A，我们将与 Hastad 的定理（定理 4.4）矛盾。考虑以下算法 A'：它对输入的 a k-SAT 公式 φ 运行 $A(\varphi)$。如果 A 输出的值是 $\ge \frac{1}{4^{m(2^{-k}-\varepsilon)}}$，则 A' 输出 YES，否则输出 NO。假设 φ 可满足，则 optConf(φ) = 1。因此 A 将输出一个值 $\ge \frac{1}{4^{m(2^{-k}-\varepsilon)}}$。于是 A' 输出 YES。假设 φ 的最大可满足子句数为 $\le m(1 - 2^{-k} + \varepsilon)$。根据定理 3.7,

[1] 为简便起见，我们在此忽略精度问题，但误差可以做到关于 n 的逆多项式级别。

$$\mathsf{optConf}(\varphi) < \frac{1}{4^{m-m(1-2^{-k}+\varepsilon)}} = \frac{1}{4^{m(2^{-k}-\varepsilon)}}$$

因此， A的输出为$< \frac{1}{4^{m(2^{-k}-\varepsilon)}}$，所以 A' 将输出 NO。

因此， A' 与定理 4.4 矛盾，除非 P $=$ NP。 □

因此，例如对于 3-SAT 公式，虽然我们有一个多项式时间的 0.716^m-近似算法（根据定理 4.3），但根据上述结果，除非 P 等于 NP，否则我们不能指望存在高效的 0.845^m-近似算法。确定该问题能通过多项式时间算法达到的最优近似比，仍是一个有趣的开放问题。

5 AccessMaximization 的复杂性

本节中，我们研究访问控制半环 $\mathbb{A}_k = ([k], \max, \min 0, k)$ 上的优化问题。我们将相应的计算问题称为 optAccessVal 和 optAccess。本节首先假设否定函数是模 k 的加法逆元。也就是说， $\neg(a) = b$ 使得 $a + b \equiv 0 (\mathrm{mod}\ k)$。

定理 5.1。设 $\varphi(x_1, \cdots x_n)$ 是一个处于否定范式的命题公式，且 $\mathbb{A}_k = ([k], \max, \min 0, k)$。以下陈述成立。

- 如果 φ可满足，则 optAccessVal(φ) $= k$。
- 如果 φ 不可满足，则 optAccessVal(φ) $= \lfloor \frac{k}{2} \rfloor$。

证明。我们首先证明 φ 为合取范式的情形，即 $\varphi = C_1 \wedge \cdots \wedge C_m$。假设该公式可满足，且 $a_1 \cdots a_n$ 是变量 x_1、 x_2、 $\cdots$、 x_n的满足赋值。考虑如下定义的解释 π： 如果 a_i 为真，则 $\pi(x_i) = k$；否则 $\pi(x_i) = \neg(k)$。考虑一个子句C，由于该公式可满足，因此在 C 中存在一个文字 ℓ_i （对于某个 i，其为 x_i 或 $\neg x_i$ ），使得 ℓ_i 被设为真。如果 $\ell_i = x_i$，则 $\pi(x_i) = k$ 且 Sem(x_i, π) $= k$。如果 $\ell_i = \neg x_i$，则 $\pi((x_i) = \neg(k) = 0$ 且 Sem($\neg x_i, \pi$) $= \neg(0) = k$。由于 C 是析取，Sem(C, π) $= k$。因此，对于每个子句 C_i，Sem(C_i, π) $= k$。由于 φ 是 C_1、 $\cdots C_m$的合取，因此 Sem(φ, π) $= k$。

对于第二项的证明，先假设 k 是偶数；当 k 是奇数时，证明过程非常相似。注意，在这种情况下， $\neg(k/2) = k/2$。设 $\varphi = C_1 \wedge \cdots \wedge C_m$ 是一个不可满足公式。考虑一个解释 π，其中对每个 $1 \le i \le n$， $\pi(x_i) = k/2$ 。显然，对于这个解释，Sem(φ, π) $= k/2$。假设 π' 是一个解释，Sem(φ, π') $> k/2$。考虑以下满足赋值：若 $\varphi'(x_i) > k/2$，则 a_i 为真；否则 a_i 为假。注意，这是一个一致的赋值。我们将证明该赋值满足 φ。由此可得，optAccessVal(φ) $= k/2$。

注意，对于每个子句 $C_j, 1 \le j \le m$， Sem(C_j, π') $> k/2$。固定一个子句 C。由于 Sem(C, π') $> k/2$，存在一个文字 ℓ_i 属于 C，使得 Sem(ℓ_i, π') $> k/2$。若 $\ell_i = x_i$，则 Sem(x_i, π') $> k/2$，这意味着 $\pi'(x_i) > k/2$。因此 a_i 为真，且赋值满足子句 C 。若 $\ell_i = \neg x_i$，则 Sem($\neg x_i, \pi'$) $> k/2$。因此 $\neg(\pi'(x_i)) > k/2$。根据 $\neg$ 的定义，我们有 $\pi'(x_i) < k/2$。因此 a_i 被赋值为假。于是子句 C 可满足。这证明赋值 $a_1, \cdots, a_n$ 满足公式$\varphi(x_1, \cdots, x_n)$。

一般公式处于否定范式的情形，可借助证明树的概念，沿用维特比半环情形中的类似思路得出。

□

对于一般的否定函数，我们可以建立一个类似的定理。为此，我们定义否定指数的概念。给定一个否定函数 ¬，其指数记为 Index(¬)，是满足以下条件的最大 ℓ：存在 $a \in [k]$，使得 a 和 ¬(a) 均至少为 ℓ。

定理 5.2。设 $\varphi(x_1, \cdots x_n)$ 是一个处于否定范式的命题公式，且 $\mathbb{A}_k = ([k], \max, \min, 0, k)$。以下陈述成立。

- 如果 φ可满足，则 optAccessVal(φ) $= k$。
- 如果 φ不可满足，则 optAccessVal(φ) $= Index(\neg)$。

以下推论由上述结果及其证明得出，它指出，访问控制半环上的优化问题的复杂性与布尔半环上的优化问题的复杂性等价。

定理 5.3。问题 optAccessVal 与 SAT 在度量归约下等价。同样，问题 optAccess 与计算给定布尔公式的一个满足赋值的问题在度量归约下等价。

6 结论

在本研究中，我们从算法和复杂性理论视角，全面研究了 optSem 及相关问题 optSemVal 在多种半环上的计算复杂性，包括维特比半环、热带半环、访问控制半环和模糊半环。近年来，CSP/SAT 求解领域令人振奋的发展之一，是 LexSAT 求解器的出现，它旨在寻找公式 [MSAGL 11] 的字典序最小满足赋值。在这方面，定理 3.2 为未来开发 optConf 的高效技术开辟了令人振奋的方向。

7 致谢

我们感谢 Val Tannen 将我们引入半环语义的世界，并在项目初期与我们进行了有益的交流。我们感谢 AAAI-23 的匿名评审提供宝贵意见。本研究由国家研究基金会通过 NRF 奖学金计划 [NRF-NRFFAI1-2019-0004] 以及研究卓越与技术企业园区（CREATE）计划资助。Bhattacharyya 的部分研究工作由 NRF 奖学金计划 [NRF- NRFFAI1-2019-0002] 和亚马逊研究奖资助。Vinod 的部分研究工作由美国国家科学基金会 CCF-2130608 和美国国家科学基金会 HDR:TRIPODS-1934884 项目资助。Pavan 的部分研究工作由美国国家科学基金会 CCF-2130536 和美国国家科学基金会 HDR:TRIPODS-1934884 项目资助。

参考文献

[ADT11] Yael Amsterdamer、Daniel Deutch 和 Val Tannen。聚合查询的溯源。载于 PODS 论文集，第 153–164 页，2011年。[AS08] Noga Alon 和 Joel H. Spencer。《概率方法》（第三版）。Wiley-Interscience 离散数学与优化丛书。Wiley，2008年。

[BG06] Stefano Bistarelli 和 Fabio Gadducci。增强基于半环的形式化体系中的约束操作。载于 ECAI，第 141 卷，第 63–67 页，2006年。[Bis04] Stefano Bistarelli。《用于软约束求解与编程的半环》，第 2962 卷。施普林格科学与商业媒体，2004年。[BMR95] Stefano Bistarelli、Ugo Montanari 和 Francesca Rossi。基于半环的约束求解。载于 IJCAI (1)，第 624–630 页。Citeseer，1995年。[BMR97] Stefano Bistarelli、Ugo Montanari 和 Francesca Rossi。基于半环的约束满足与优化。J. ACM，44(2):201–236，1997年。[BMR+99] Stefano Bistarelli、Ugo Montanari、Francesca Rossi、Thomas Schiex、G´erard Verfaillie 和 H´elene Fargier。基于半环的约束满足问题与赋值约束满足问题：框架、性质与比较。Constraints，4(3):199–240，1999年。[Cui02] Yingwei Cui。数据仓库中的谱系追踪。博士论文，斯坦福大学，2002年。[CWW00] Yingwei Cui、Jennifer Widom 和 Janet L Wiener。在数据仓库环境中追踪视图数据的谱系。ACM Transactions on Database Systems (TODS)，25(2):179–227，2000年。[DMRT14] Daniel Deutch、Tova Milo、Sudeepa Roy 和 Val Tannen。用于 Datalog 溯源的电路。载于 ICDT 论文集，第 201–212 页。OpenProceedings.org，2014年。[EK21] Thomas Eiter 和 Rafael Kiesel。半环上的和积问题的复杂性。载于 AAAI 论文集，第 6304–6311 页。AAAI Press，2021年。[FGT08] J Nathan Foster、Todd J Green 和 Val Tannen。带注释的 XML：查询与溯源。载于第二十七届 ACM SIGMOD-SIGACT-SIGART 数据库系统原理研讨会论文集，第 271–280 页，2008年。[FR97] Norbert Fuhr 和 Thomas R¨olleke。用于整合信息检索与数据库系统的概率关系代数。ACM Transactions on Information Systems (TOIS)，15(1):32–66，1997年。[GKT07] Todd J. Green、Gregory Karvounarakis 和 Val Tannen。溯源半环。载于 PODS 论文集，第 31–40 页，2007年。[GM21] Erich Gr¨adel 和 Lovro Mrkonjic。半环语义中的初等等价与同构。载于 ICALP 论文集，LIPIcs 第 198 卷，第 133:1–133:20 页，2021年。[Gre11] Todd J Green。带注释关系上的合取查询包含性。Theory of Computing Systems，49(2):429–459，2011年。[GT20] Erich Gr¨adel 和 Val Tannen。逻辑与博弈的溯源分析。Moscow Journal of Combinatorics and Number Theory，9(3):203–228，2020年。[GW94] Michel X. Goemans 和 David P. Williamson。最大可满足性问题的新型 3/4近似算法。SIAM J. Discret. Math.，7(4):656–666，1994年。

[H˚as01] Johan H˚astad. 一些最优的不可近似性结果。J. ACM, 48(4):798–859, 2001年。[ILJ89] Tomasz Imieli´nski and Witold Lipski Jr. 关系数据库中的不完整信息。载于《人工智能与数据库读物》，第342–360页。Elsevier, 1989。[KM03] Dan Klein and Christopher D. Manning. A* 句法分析：快速精确的维特比句法分析选择。载于 Marti A. Hearst 和 Mari Ostendorf 编，《HLT-NAACL 论文集》。计算语言学协会，2003。[Kre88] Mark W. Krentel. 优化问题的复杂性。J. Comput. Syst. Sci., 36(3):490–509, 1988。[Moh02] Mehryar Mohri. 最短距离问题的半环框架与算法。J. Autom. Lang. Comb., 7(3):321–350, 2002年1月。[MRS06] Pedro Meseguer, Francesca Rossi, and Thomas Schiex. 软约束。载于《人工智能基础》，第2卷，第281–328页。Elsevier, 2006年。[MSAGL11] Joao Marques-Silva, Josep Argelich, Ana Grac¸a, and In^es Lynce. 布尔字典序优化：算法与应用。《数学与人工智能年刊》，62(3):317–343, 2011。[Tan13] Val Tannen. 复杂查询中的来源传播。载于《探寻计算理论与实践之美》，第483–493页。Springer, 2013。[Tan17] Val Tannen. FOL 模型检验的来源分析。ACM SIGLOG News, 4(1):24–36, 2017。[Vit67] Andrew Viterbi. 卷积码的错误界及渐近最优译码算法。IEEE 信息论汇刊，13(2):260–269, 1967。[Zim97] Esteban Zim´anyi. 概率关系数据库中的查询求值。《理论计算机科学》，171(1-2):179–219, 1997年。