arXiv:2512.14269v1 [cs.SC] 16 Dec 2025

More isLess：通过添加多项式加快 NLSAT 中的解释

Valentin Promies[1][0000−0002−3086−9976]✉、Jasper Nalbach[1][0000−0002−2641−1380]、Erika Ábrahám[1][0000−0002−5647−6134] 和 PaulWagner[1]

德国亚琛亚琛工业大学 {promies, nalbach, abraham}@cs.rwth-aachen.de

摘要。 为了检查（非线性）实数算术公式的可满足性，现代可满足性模理论（SMT）求解算法（如 NLSAT）高度依赖于单细胞构造：将样本点推广为 $\mathbb{R}^n$ 的一个连通子集（单元），该子集包含此样本点，并且给定的一组多项式在其上符号不变。本文提出通过动态扩展所考虑的多项式集合，加入更多线性多项式，从而加快计算并简化所得单元的表示。尽管这会增加算法运行过程中生成的（更小）单元总数，但我们的实验表明，借助合适的启发式方法，这种做法可以受益于与布尔推理的相互作用。

关键词： 单细胞构造 · 柱面代数分解 · 实代数 · 可满足性模理论求解

1 引言

可满足性检查 处理的是判定一阶逻辑公式是否存在解的问题。理论模可满足性（可满足性模理论） 求解器使用专门的算法，针对不同理论解决这一问题。尽管目标问题通常很难（命题逻辑中的问题是 NP 完全的，而整数算术中的问题甚至是不可判定的），现代可满足性模理论 求解器却非常高效，并被广泛用作集成引擎，例如用于自动演绎 [5,17]。

本文聚焦于非线性实数算术（*NRA*）的无量词片段，称为$QF\text{-}NRA$。其公式是带有有理系数和实值变量的多项式约束的布尔组合。柱面代数分解（CAD）方法 [11], 通常是一种针对NRA的量词消去过程，也是首个用于解决QF-NRA可满足性问题的可处理技术。CAD将变量的搜索空间划分为有限数量的单元，使得输入公式中的所有多项式在每个CAD 单元内都具有符号不变性，从而输入公式也具有真值不变性。因此，我们可以通过检查每个单元中的一个点来判定可满足性问题。尽管 Mc- Callum [23,24], 、Lazard [21], 和 Brown [8],进行了重大改进，CAD 的扩展性仍然较差，这通常是由于结式计算代价高昂所致。

受CAD启发，最成功的SMT求解QF-NRA 技术之一是$NLSAT$ [19]„ 它是由 Jovanic 和 de Moura 提出的模型构造可满足性演算（$MCSAT$） [25] 的一个实例。NLSAT 将DPLL+CDCL风格的命题推理扩展为一种针对理论的对偶方法。命题部分包括判定约束的真值、布尔传播以及布尔冲突消解。相应地，理论部分判定理论变量的实数值，并通过理论传播确保理论赋值与布尔赋值一致地计算约束，同时进行理论冲突消解。两部分的步骤交错执行，维护一致的部分赋值，以相互引导并最终找到解。

如果某个未赋值理论变量的所有可能取值都会与某个约束的布尔赋值矛盾，那么当前理论赋值就无法一致地扩展。可以使用解释后端来解决这种情况：它会将冲突的原因推广为一个引理（称为解释），该引理不仅排除当前的冲突赋值，还能排除未来搜索中更多类似的情况。例如，给定公式（$x_1^2 + x_2^2 - 1 < 0 \ \vee \ x_2 > 0$），NLSAT 可以将第一个约束判定为真，然后赋值 $x_1 = 2$。此时， x_2 的任何取值都无法满足 $x_1^2 + x_2^2 - 1 < 0$，因为在 $x_1 = 2$下它会化简为 $x_2^2 < -3$ 。然而，如果 $x_1^2 + x_2^2 - 1 < 0$ 成立，那么任意 $x_1 > 1$ 都要求 x_2^2 为负。因此，我们可以通过学习引理（$\neg\,(x_1^2 + x_2^2 - 1 < 0) \ \vee \ x_1 \leq 1$），将取值$x_1 = 2$ 推广为 $x_1 > 1$ 。

解释应当能够高效计算，并尽可能充分地泛化。请注意，学习到的子句可能包含输入公式中不存在的文字（如上方的 $x_1 < 1$ ），因此泛化技术对完备性至关重要。NLSAT 使用柱面代数分解 技术，将样本点泛化为其周围的单个单元。然而，NLSAT会针对样本点和约束的子集局部 计算单元，这为简化计算和构造更大的单元提供了可能。Brown 和 Košta [9], 、Li 和 Xia [22], 以及 Nalbach 等人 [28]所做的改进利用了这一优势，并避免了某些结式计算，但单细胞构造仍然是影响运行时间的主要因素。

贡献。 计算两个多项式的结式，其成本取决于它们的次数。如果其中一个多项式是线性多项式，通常可以较为廉价地计算结式。本文在单元构造过程中动态插入线性多项式，从而有效地低估单元的界。这通过用更简单的计算替代昂贵的结式计算，降低了构造所需的工作量。这也会影响NLSAT中单元的质量：其表示更简单，但覆盖的搜索空间更小。本文是在扩展摘要 [29], 的基础上进一步展开的；我们在该摘要中简要介绍了这一想法。具体而言，我们的贡献包括以下几点：

– 我们将 [29] 中的思想加以推广，并给出清晰的算法化表述。– 我们通过一个示例，进一步阐明使用修改后的单细胞构造时 NLSAT 可能不终止的原因，并调整该方法以保证终止。– 我们探讨了该方法的若干变体，包括构造附加多项式的不同方式，以及决定何时插入这些多项式的准则。

– 我们使用在 可满足性模理论 求解器 SMT-RAT [12,30] 中的实现，对这些变体进行了全面的实验评估。

结构。 第 2 节回顾一些背景知识，包括逐层单元构造；我们在第 3 节对其进行调整，以用于下近似。第 4 节探讨我们方法的变体，并在第 5 节讨论实验。第 6 节总结全文，并展望未来工作。

相关工作。 一些不完备但速度较快的 NLSAT解释后端并非基于柱面代数分解，而是基于傅里叶-莫茨金变量消去 [18], 虚拟代换 [2], 或区间约束传播 [20]。

此前已有研究探索将线性近似或抽象用于量词自由非线性实数算术。例如，ksmt 演算 [6] 会将每个公式转换为一组线性算术子句和一组非线性约束，然后逐步构造模型，主要进行线性推理。对于会使非线性约束为假的部分赋值，利用非线性函数的局部线性近似将其推广为线性约束合取式。增量线性化 [10] 通过将所有乘法替换为未解释函数，计算输入公式的线性抽象，从而有可能仅凭线性推理推导出不可满足性。随后，通过为各个乘法添加线性算术公理，逐步细化该抽象。这两种方法都不完备，不过已证明 ksmt 是 δ-完备的。

2 预备知识

我们假设读者具备一些关于多变量多项式、逻辑和可满足性模理论求解的基础知识。相关介绍可参见例如 [4,13]。

令 $\mathbb{N}$、 $\mathbb{Q}$和 $\mathbb{R}$ 分别表示自然数（包括 0）、有理数和实数的集合。对于 $k \in \mathbb{N}$，令 $[k] := \{1, \ldots, k\}$；对于 $r \in \mathbb{R}^k$和 $i \in [k]$，令 r_i为 r中的第 i个元素， $r_{[i]} := (r_1, .., r_i)$，以及$r_{[0]} = ()$。在本文中，我们固定一些 $n \in \mathbb{N} \setminus \{0\}$ 和有序实值变量 $x_1 \prec \ldots \prec x_n$。

多项式。对于 $i \in [n]$，令 $\mathbb{Q}[x_1, \ldots, x_i]$ 表示由有理系数构成的、关于x_1、 . . .、 x_i的所有多项式的集合（对于 $i = 0$，该集合为 $\mathbb{Q}$）。任意$p \in \mathbb{Q}[x_1, \ldots, x_i]$ 都可以写成关于 x_i 的一元多项式 $p = c_d x_i^d + c_{d-1} x_i^{d-1} + \ldots + c_1 x_i + c_0$，其 $d = 0$ 或 $c_d \neq 0$，次数为 $\mathsf{deg}_{x_i}[p] := d$，系数为 $\mathsf{coeff}_{x_i}[p] := \{c_0, \ldots, c_d\} \subset \mathbb{Q}[x_1, \ldots, x_{i-1}]$，首项系数为 $\mathsf{ldcf}_{x_i}[p] := c_d$。给定 $r \in \mathbb{R}^i$，我们用 $p(r)$表示 $p(r_1$、 . . .、 $r_i) \in \mathbb{R}$的求值。给定 $r \in \mathbb{R}^{i-1}$ 和 $r' \in \mathbb{R}$，令 $p(r, x_i)$表示由 p通过将 r_1、 . . .、 r_{j-1}代入x_1、 . . .、 x_{j-1}所得的结果，并将 $p(r_1$、 . . .、 $r_{j-1}, r')$记作 $p(r, r')$。

实根。 设 $p \in \mathbb{Q}[x_1, \ldots, x_i]$。 p的（实）根 是满足$p(r) = 0$的点 $r \in \mathbb{R}^i$； p 的簇 是其根构成的集合。单变量多项式的根构成（实）代数数 $\mathbb{Q}$的集合：
$= \{r \in \mathbb{R} \mid \exists q \in \mathbb{Q}[x].q(r) = 0\}$。

给定 $r \in \bar{\mathbb{Q}}^{i-1}$，可以计算 realRoots$(p, r) := \{r_j \in \mathbb{R} \mid p(r, r_j) = 0\}$，即 p 在 r上的根 。如果 realRoots$(p, r) = \mathbb{R}$，那么我们称 p 在 ..... 上被零化 r。否则，realRoots(p, r) 是一个有限的代数数集合。

设 $j \in \mathbb{N}$。带索引的根表达式$\text{root}_{x_i}[p, j]$： $\mathbb{R}^{i-1} \to \mathbb{R} \cup \{\bot\}$ 将每个 $r \in \mathbb{R}^{i-1}$ 映射为 p 在 r 上的第 j 个根（若存在），否则映射为 $\bot$：

$$\text{root}_{x_i}[p, j](r) := \begin{cases} \bot & \text{if realRoots}(p, r) = \mathbb{R} \text{ or } j > |\text{realRoots}(p, r)|, \text{ and else} \\ z_j & \text{where realRoots}(p, r) = \{z_1, \ldots, z_k\}, z_1 < \ldots < z_k. \end{cases}$$

我们用 $\xi.p$ 表示带索引的根表达式 $\xi = \text{root}_{x_i}[p, j]$ 的多项式，并称 ξ 的层为 i。层为 i 的带索引的根表达式集合记为 IRE(i)。给定 $P \subseteq \mathbb{Q}[x_1, \ldots, x_i]$ 和 $r \in \bar{\mathbb{Q}}^{i-1}$，可以计算在 r 上定义的带索引的根表达式：irExp(P, r)：$= \{\xi \in \text{IRE}(i) \mid \xi.p \in P \text{ 且 } \xi(r) \neq \bot\}$。

两个多项式 p, $q \in \mathbb{Q}[x_1, \ldots, x_i]$ 关于 x_i 的结式是多项式 $\text{res}_{x_i}[p, q] \in \mathbb{Q}[x_1, \ldots, x_{i-1}]$，满足对所有 $r \in \mathbb{R}^{i-1}$：若存在 $r' \in \mathbb{R}$ 使得 $p(r, r') = 0 = q(r, r')$，则 $\text{res}_{x_i}[p, q](r) = 0$。$p$ 的判别式为 $\text{disc}_{x_i}[p] := \text{res}_{x_i}[p, p']$，其中 p' 是 p 关于 x_i 的导数。

公式。（非线性）实数算术无量词片段（量词自由非线性实数算术）中的公式是由形如 $p \sim 0$ 的（多项式）约束 构成的布尔组合，其中 $p \in \mathbb{Q}[x_1, \ldots, x_n]$ 和 $\sim \in \{<, >, =, \leq, \geq, \neq\}$。扩展约束 的形式为 $x_i \sim \xi$，其中 $\xi \in$IRE(i) 是一个带索引的根表达式。

单元。单元 是一个非空连通集合 $S \subseteq \mathbb{R}^i$，其中 $i \in [n]$。如果S 是由一组约束和扩展约束的合取式定义的解集，我们就称其为（半）代数 。如果 p 在 S 上符号不变，即对所有点 $r \in S$， $p(r)$ 的符号都相同（也就是说，对于固定的 $\sim \in \{<, =, >\}$， $\forall r \in S.\ p(r) \sim 0$）。如果所有 $p \in P$ 在 S 上都符号不变，我们就称 S 对 $P \subset \mathbb{Q}[x_1, \ldots, x_i]$ 符号不变。

2.1 逐层单细胞构造

给定约束集 C 和赋值 $s \in \mathbb{R}^i$，如果 s 的所有扩展都会使 C 中的某些约束求值为假，那么我们称 s与 C不一致 。在NLSAT中，如果理论赋值与布尔赋值判定为真的约束 C 不一致，那么我们将 s 推广为一个单元$S \subseteq \mathbb{R}^i$，其所有点都与 C 不一致。为此，我们推导出一组 $P \subset \mathbb{Q}[x_1, \ldots, x_i]$投影多项式，使得 $s \in S$ 以及 P 在 S上的符号不变性保证 S 中的所有点都与 C不一致。于是，学习到的解释为（$\neg C \vee \neg \varphi_S$），其中 φ_S 是定义 S的扩展约束的合取。

定义1. 给定 $i \in [n]$、 $P \subset \mathbb{Q}[x_1, \ldots, x_i]$ 和 $s \in \mathbb{R}^i$，单细胞构造 （*SCC*）问题是计算代数单元$S \subseteq \mathbb{R}^i$ 的描述，使得 $s \in S$ 和所有 $p \in P$ 在 S上均符号不变。

我们现在回顾 Nalbach 等人提出的逐层SCC 方法 [28]，该方法总结于算法1中，我们将在第 3 节对其进行修改。

算法1： levelwise-scc$(P,\ s)$

输入 ：一个有限的 $P \subseteq \mathbb{Q}[x_1,\ldots,x_i]$，以及 $s \in \mathbb{R}^i$（包含 $i \in [n]$）。
输出： 一个关于 P的符号不变单元的描述（$I_1 \wedge \ldots \wedge I_i$），其中包含 s

对于 $j = i,\ \ldots\ldots,\ 1$ **执行**
  $P_j :=P \cap (\mathbb{Q}[x_1,\ldots,x_j] \setminus \mathbb{Q}[x_1,\ldots,x_{j-1}])$
  如果 实根$(p,\ s_{[j-1]}) = \mathbb{R}$ 对某个 $p \in P_j$ **则返回** 失败
  $\{\xi_1,\ldots,\xi_k\} :=$ irExp$(P_j,\ s_{[j-1]})$ 使得 $\xi_1(s_{[j-1]}) \leq \ldots \leq \xi_k(s_{[j-1]})$
  如果 $k=0$ **则** $I_j :=(true)$
  否则，如果 $s_j = \xi_\ell(s_{[j-1]})$ 对某个 ℓ **则** $I_j :=(x_j = \xi_\ell)$
  否则，如果 $s_j > \xi_k(s_{[j-1]})$ **则** $I_j :=(\xi_k < x_j)$
  否则，如果 $s_j < \xi_1(s_{[j-1]})$ **则** $I_j :=(x_j < \xi_1)$
  否则 $I_j := (\xi_\ell < x_j \wedge x_j < \xi_{\ell+1})$ 对于带有 $\xi_\ell(s_{[j-1]})$ 的 $\ell < s_j < \xi_{\ell+1}(s_{[j-1]})$
  添加到 P 判别式和系数中，以确保可描绘性
  添加到 P 结式中，以确保符号不变性
返回 $(I_1 \wedge \ldots \wedge I_i)$

令 $i \in [n]$、$s \in \mathbb{R}^i$、$P \subset \mathbb{Q}[x_1,\ldots,x_i]$，并且对于 $j \in [i]$，令 P_j为 P 中最大变量为 x_j 的多项式（即包含 x_j，但不包含 x_{j+1}、.....、x_i 的多项式）。对于每个维度 $j = i$、.....、1，算法确定一个符号区间I_j，其形式为（$x_j = \xi$）、（$x_j < \xi$）、（$x_j > \xi$）或（$\xi < x_j \wedge x_j < \xi'$），其中 ξ、$\xi' \in$ IRE(j)，用于界定 x_j 相对于较低变量 x_1、.....、x_{j-1}的值。对于所有满足 $\xi(r) \neq \bot \neq \xi'(r)$ 的 $r \in \mathbb{R}^{j-1}$，I_j 定义一个具体区间 $I_j(r) \subseteq \mathbb{R}$，该区间分别为 $\{\xi(r)\}$、$(-\infty, \xi(r))$、$(\xi(r)\ \infty)$或$(\xi(r)\ \xi'(r))$。

由 $I_1 \wedge \ldots \wedge I_i$描述的最终单元是局部柱状的，也就是说，$I_1$定义了一个具体区间 $S_1 \subseteq \mathbb{R}$，并且对于 $j = 2,\ldots,i$，I_j中的根表达式在 S_{j-1}上处处有定义，并且它们确定了该单元

$$S_j = \{(r, r') \in \mathbb{R}^j \mid r \in S_{j-1} \wedge r' \in I_j(r)\}.$$

要确定 I_j，我们将 $x_1,\ldots,x_{j-1}$赋值为底层样本点 $s_{[j-1]}$，并针对所有 $p \in P_j$计算 realRoots$(p, s_{[j-1]})$。这些根为算法1中的带索引的根表达式提供依据。不大于（或等于） s_j 的最大根和不小于（或等于） s_j 的最小根给出了区间边界（如果它们不存在，则使用 $-\infty$ 和 ∞）。因此，P_j 中的多项式在 $\{s_{[j-1]}\} \times I_j(s_{[j-1]})$上符号不变。

现在的思路是，待构造的底层单元 $S_{j-1} \subseteq \mathbb{R}^{j-1}$ 将是 $s_{[j-1]}$ 周围的一个邻域，在该邻域上，I_j 中的根表达式定义了总连续函数，使得 S_j 是包含 $s_{[j]}$ 的关于 P_j 的符号不变单元。为了获得具有所需性质的底层单元，我们将使用可描绘性和阶不变性（符号不变性的强化形式）这两个概念:

定义*2*（可描绘性 **[11]**）。设 $j \in [n-1]$ 和 $S \subseteq \mathbb{R}^j$ 是一个单元。如果存在 $k \geq 0$个连续函数 θ_1、...、θ_k：$S \to \mathbb{R}$ 以及常数 m_1、$\ldots m_k \in \mathbb{N}$，使得对所有 $r \in S$均有 $\theta_1(r) < \ldots < \theta_k(r)$，*realRoots*$(p, r) = \{\theta_1(r)$ ...、$\theta_k(r)\}$，并且对于所有 $\ell = 1$、...、k，根 $\theta_\ell(r)$ 在 $p(r, x_{j+1})$ 中的重数为 m_ℓ，则称非零多项式 $p \in \mathbb{Q}[x_1,\ldots,x_{j+1}]$ 在 S上具有可描绘性。

定义3（阶不变性 [23]）。 $p \in \mathbb{Q}[x_1,\ldots,x_j]$ 在 $r \in \mathbb{R}^j$ 处的阶，记为 $ord(p, r)$，是使得 p 的某个总阶数为 k 的偏导数在 k 处的值不为0 的最小值（或者，如果所有偏导数的值均为0，则为 ∞）。如果对于所有 r、$r' \in R$ 均有 $ord(p, r) = ord(p, r')$，则称 p 在 $R \subseteq \mathbb{R}^j$ 上具有阶不变性。

p 在算法1中确定的带索引的根表达式见证了 θ 函数。p 的可描绘性保证了这些函数定义良好且连续。该方法利用了以下事实：如果 $\mathsf{disc}_{x_j}[p]$ 在 S_{j-1} 上是序不变的，并且 $\mathsf{ldcf}_{x_j}[p]$ 在 S_{j-1} 上是符号不变的，那么 p 在底层单元 S_{j-1} 上是可描绘的；因此，它将这些多项式添加到 P 中，并确保它们在下一层上满足相应性质，从而限制 I_{j-1} 及其以下各层。

该方法仍需确保没有根函数跨越单元边界（ I_j 中的根表达式）而经过 S_{j-1}，因为这将意味着某个多项式在 S_j 内部发生符号变化。为此，我们利用如下事实：对于任意两个 $\xi,\ \xi' \in \mathsf{irExp}(P_j,\ S_{\lfloor j-1 \rfloor})$ 和 $\sim \in \{<, =\}$，若 $\xi(s_{\lfloor j-1 \rfloor}) \sim \xi'(s_{\lfloor j-1 \rfloor})$，则有：如果 $\mathsf{res}_{x_j}[\xi.p, \xi'.p]$ 在 S_{j-1} 上是序不变的，那么对于所有 $r \in S_{j-1}$，$\xi(r) \sim \xi'(r)$。

由于只有根与单元边界的交点才相关，因此只需维护根函数的偏序，这可由某些结式保证（算法1）。例如，如果 $\xi_1,\ldots,\xi_k$ 如算法1所示，且 $I_j := (\xi_\ell < x_j \wedge x_j < \xi_{\ell+1})$，那么我们可以添加 $\{\mathsf{res}_{x_j}[\xi_{\ell'}.p, \xi_\ell.p] \mid \ell' < \ell\} \cup \{\mathsf{res}_{x_j}[\xi_{\ell+1}.p, \xi_{\ell'}.p] \mid \ell + 1 < \ell'\}$，从而确保 $\xi_1,\ldots,\xi_{\ell-1}$ 保持在 ξ_ℓ 之下（或与其相等），而 $\xi_{\ell+2},\ldots,\xi_k$ 保持在 $\xi_{\ell+1}$ 之上（或与其相等）。利用传递性，还可以采用其他偏序，因此也可以采用其他结式集合；这属于一种启发式决策。不过，界的结式 $\mathsf{res}_{x_i}[\xi_\ell.p, \xi_{\ell+1}.p]$ 始终会被添加，以确保 S_j 的连通性。

请注意，如果任何遇到的多项式在底层样本点处被零化，则此方法会失败，因为此时无法以相同方式确保可描绘性。该方法可以检测到这种情况并返回“失败”，然后改用另一种完备方法。为进一步确保多项式 $p \in P_j$ 不会在 S_{j-1} 中的任何其他点处被零化，算法1还会将某个满足 $c(s_{\lfloor j-1 \rfloor}) \neq 0$ 的 $c \in \mathsf{coeff}_{x_j}[p]$ 添加到 P 中。将所有必需的多项式添加到 P 后，如果 $j > 1$，则该方法会以相同方式继续构造 I_{j-1}。

例 1. 图1展示了一个给定样本点 $s \in \mathbb{R}^2$ 和多项式 $P = \{p_1, p_2, p_3\} \subset \mathbb{Q}[x_1, x_2]$ 的示例。标记为 p_1 的直线表示 p_1 的簇，即满足 $p_1(r) = 0$ 的那些点 $r \in \mathbb{R}^2$；对于 p_2，p_3，情况也类似。

我们从第2层开始，其中 $P_2 = P$。在 $x_1 = s_1$ 处，p_2 下方有一个根，且 s_2 上方每个多项式各有一个根。最接近 s_2 的根定义了符号区间 $I_2 := (\xi_1 < x_2 \wedge x_2 < \xi_2)$（图1a）。为确保对于底层单元中 x_1 的所有取值，该区间均具有正确性（底层单元将在第1层计算），将 p_1、p_2、p_3 的判别式和首项系数添加到 P 中（图1b中的点划线）。此外，加入 $\mathsf{res}_{x_2}[p_3, p_1]$ 和 $\mathsf{res}_{x_2}[p_3, p_2]$（图1b中的虚线）可确保在 I_1 上，根函数均不会穿过由 p_1 定义的上区间边界。注意，ξ_3 与 ξ_4 的交叉并不重要，因此避免计算 p_1 与 p_2 的相应结式。在

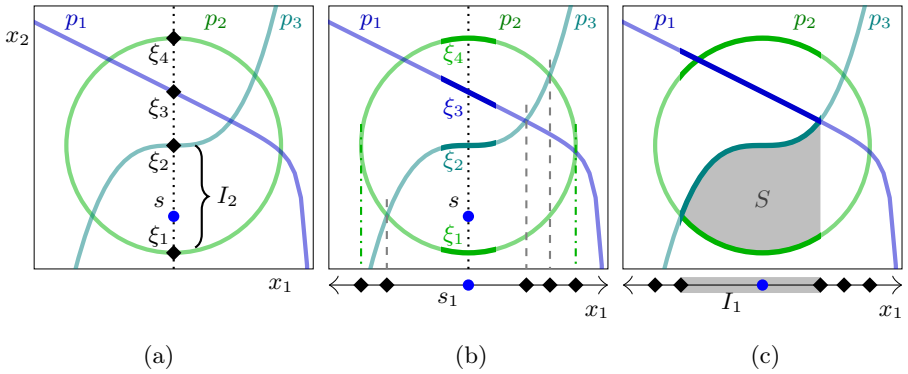

图 1：例 1 中所述逐层构造的示意图。

在第 1 层，我们隔离这些多项式的根，并使用最接近 s_1 的根作为区间边界，从而得到阴影单元（图 1c）。

3 添加多项式以避免昂贵的结式

SCC 的运行时间主要由判别式和结式计算所决定。给定 p、$q \in \mathbb{Q}[x_1, \ldots, x_n]$，其中对于所有 $j \in [n]$，$d_j =$的次数为 $\deg_{x_j}[p]$，$e_j =$的次数为 $\deg_{x_j}[q]$，p 与 q 的结式需要进行 $\mathcal{O}(d_n e_n)$ 次多项式乘法；在最坏情况下，其关于任意 x_j 的次数为 $d_n e_j + d_j e_n$。给定 $P \subset \mathbb{Q}[x_1, \ldots, x_n]$，其任意变量的最大次数为 d，在最坏情况下，SCC 中结式的次数和时间复杂度呈双重指数增长（d^{2^n}），因为结式的结式会被迭代计算。逐层方法已经减轻了结式计算的工作量，例如避免涉及高次数多项式。然而，有些结式无法避免，例如，定义区间边界的多项式总会被纳入某些结式计算中。

我们的方法如下：如果高次多项式 p 将定义符号区间 I_j 的边界，那么我们向 P 添加一个新的线性多项式 $p_* = (x_j - c)$，其根 $c \in \mathbb{Q}$ 严格位于该边界与样本s_j之间。将其用作 I_j 的新下界，可以把 p 与某些 $q \in P$ 的昂贵结式替换为 p_* 与 q 的结式；后者只需在 q中用 x_j 代入 c 即可简单计算，其次数不超过 q 的次数。对 c 的选择确保：（1）所得单元仍包含样本；（2）所有其他根仍位于单元外部；（3）底层层级仍可推广到某个更大的单元。为实现后者，c 不应等于任何多项式的根，否则该单元只能在下一级推广为截面$I_{j-1} = (x_{j-1} = \xi)$，因为该多项式与$p_*$ 的结式将在 $s_{[j-1]}$ 处有一个根。

例 *2.* 在例 1（如图 2a 所示）中，逐层方法无法避免计算结式 $\mathrm{res}_{x_2}[p_2, p_3]$ 和 $\mathrm{res}_{x_2}[p_3, p_1]$；当 p_1, p_2

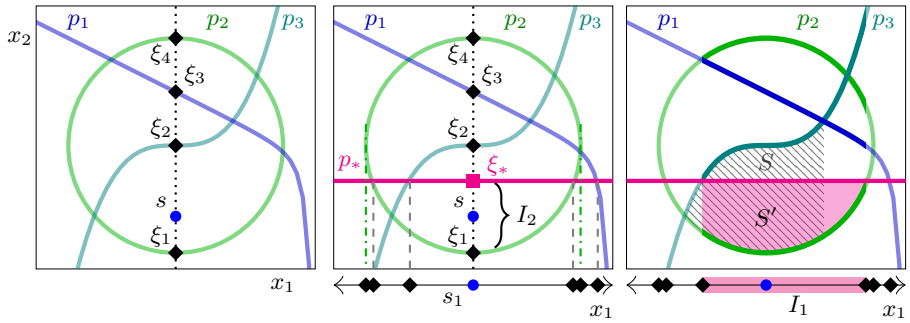

图2：使用新的线性多项式近似得到的单元

和 p_2 的次数较高时，计算这些结式的代价很高。添加一个线性多项式 p_*，使其根 ξ_* 位于s 和 ξ_2之间，就可以将 ξ_* 用作 I_2的上界，因此只需计算 p_* 与 p_1、p_2、p_3 的低成本结式（如图 2b 所示）。图 2c 展示了由此得到的单元 S'（粉色阴影部分）和原单元 S（阴影线部分）。重要的是，p_1、p_2 和 p_3 在 S'上符号不变。请注意，我们也可以对 I_2的下界进行低估，或对 I_1的界进行低估，从而得到更小的单元，但也会减少计算量。

我们将这一方法加以推广并修改 levelwise-scc，使其能够在每一层动态地用任意多项式扩展工作集 P，从而得到我们在算法2中展示的新方法 apx-scc。该方法 apx-criteria 通过检查符号区间 I_j 是否由次数较高的多项式定义等方式，判断添加新多项式是否有益。如果满足判据，apx-polys 就会计算一组新的多项式（称为辅助 多项式），并将其添加到P中。辅助多项式的根位于有利的位置，从而使需要计算的结式集合更简单。我们将在第4节讨论实现 apx-criteria 和 apx-polys 的不同方法。

添加辅助多项式会使给定样本点周围可能的最大符号不变单元变小，因此我们计算的是某种欠近似。然而，如图2c所示，apx-scc 计算出的单元不一定是 levelwise-scc 计算出的单元的子集，因为收紧 I_j 的界可能允许放宽 I_{j-1} 的某些界。无论如何，原始单元和近似单元（S 和 S'）都是 P 的最大符号不变单元 S_{max} 的子集。虽然 $S = S_{max}$ 可能成立，但 $S' \subsetneq S_{max}$ 始终成立。

我们的改进在NLSAT中的应用有两大优势：（1）避免代价高昂的结式计算，意味着可以更快地计算解释；（2）在后续计算中，NLSAT 需要在更多样本点处隔离单元边界的根，以检查给定样本是否落在排除胞中，而使用次数较低的多项式（甚至一次多项式）可以大幅减少这类计算所需的工作量。另一方面

欠近似单元可能会导致整个搜索过程中生成更多单元，甚至导致不终止，正如我们将在第3.1节中看到的那样。

需要注意的是，我们的修改并未完全消除次数的大幅增长，因为可描绘性所需的判别式也是结式。此外，对于截面 $I_j = (x_j = \xi)$，我们无法计算有意义的近似值，因此只能退回默认方法。

定理1（正确性）。设 $P \subset \mathbb{Q}[x_1, \ldots, x_i]$ 为有限集且 $s \in \mathbb{R}^i$。如果 *apx-scc*(P, s) 得出单元 $S \subseteq \mathbb{R}^i$，那么 $s \in S$，且 P 在 S 上符号不变。

证明。其思路是，在输入经过适当修改后，原方法也能得出相同的单元。对于每个 $j \in [i]$，令 $Q_j \subset \mathbb{Q}[x_1, \ldots, x_j]$ 为 apx-scc 在第 j 层添加的多项式集合，并令 $Q := Q_1 \cup \ldots \cup Q_i$。考虑由 levelwise-scc$(P \cup Q, s)$ 计算出的单元 $S' \subseteq \mathbb{R}^i$。由于该方法是正确的，$s \in S'$，且 $P \cup Q$ 在 S' 上符号不变。根据符号不变的定义，这意味着 P 在 S' 上符号不变。

我们展示 $S = S'$ 以完成证明：Q_j 中的多项式不会影响层 $j+1$，...，i的计算，因为 $Q_j \cap P_k = \emptyset$ 对于每个 $k \neq j$都成立。因此，levelwise-scc 和 apx-scc 都仅基于$P \cup \bigcup_{k=j}^{i} Q_k$计算每个层 j。由于它们的计算过程没有差异（除了添加 Q_j之外），它们计算出的区间和投影完全相同。□

3.1 不完备性

与 levelwise-scc 一样，我们的方法在零化情况下会失败，因此作为独立过程是不完备的。不过，可以检测到零化，并对该单元采用完备构造。更为关键的是，在使用 apx-scc 生成解释时，NLSAT 的终止性不再得到保证。

例*3*（不终止）。我们继续例 2。将近似单元从搜索中排除后，NLSAT 会为 y 选择一个新值。该值可能位于辅助单元边界与 p_3 的根之间，从而得到图3a所示的样本点 s'。由于 s' 和 s 位于 p_1、p_2、p_3 的同一个极大符号不变单元中，因此会与相同的约束发生冲突，并且

算法2： apx-scc(P, s)

输入：一个有限的 $P \subseteq \mathbb{Q}[x_1, \ldots, x_i]$ 和 $s \in \mathbb{R}^i$（带有 $i \in [n]$）。
输出： 一个符号不变单元的描述 $(I_1, \ldots, I_i)$，该单元包含 P s

1 **对** $j = i, \ldots, 1$ **执行**
2 　$P_j := P \cap (\mathbb{Q}[x_1, \ldots, x_j] \setminus \mathbb{Q}[x_1, \ldots, x_{j-1}])$
3 　**如果** 实根$(p, s_{[j-1]}) = \mathbb{R}$ 对于某个 $p \in P_j$ **则返回** 失败
4 　**如果** apx-criteria$(P_j, s_{[j]})$ **则** $P := P \cup$ apx-polys$(P_j, s_{[j]})$
5 　**计算** I_j，与之前相同（算法1中的第2-9行）
6 　**添加**到 P 多项式中，以确保可描绘性和符号不变性（算法1中的直线第10—11行）
7 **返回** $(I_1, \ldots\ldots, I_i)$

p_1 p_2 p_3

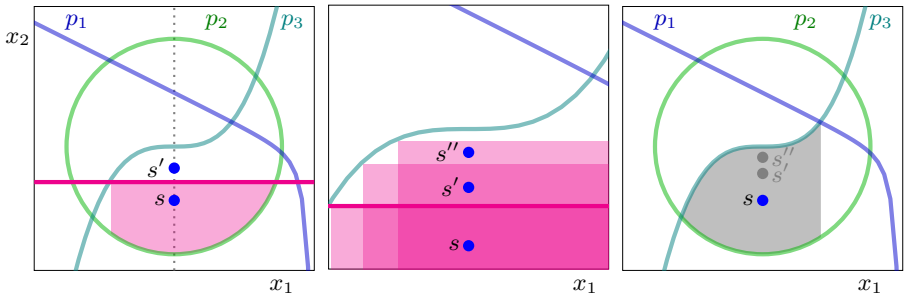

图3：如例3所述，使用欠近似单元时 NLSAT 的不终止情况。图（b）放大了 p_3 与近似区域之间的部分。

因此，apx-scc 会使用相同的多项式被调用。计算新的解释时，会在 s' 和 p_3 之间引入另一个辅助边界，这种行为不断重复，从而得到图3b所示的样本点 s''。这种情况可能无限重复，NLSAT 会一再遇到相同的冲突，却始终无法覆盖整个搜索空间。正如图3c所示，使用原始构造时不会出现这种行为。

请注意，NLSAT 并不总会陷入这种情况。它也可能选择距离更远的样本点，从而避开冲突。由于公式的布尔结构，剩余的搜索空间仍可能被针对其他冲突的近似单元覆盖，而这些冲突涉及不同的多项式集合。

我们也可以用 [19] 中的形式术语来表达这一点：NLSAT 的终止依赖于使用有限基解释函数。也就是说，对于每个输入公式 φ，都存在一个有限集合 $B(\varphi) \subset \mathbb{Q}[x_1, \ldots, x_n]$，使得在 NLSAT 的所有可能运行中，所有解释都只使用来自 B 的多项式。由于存在这样的 NLSAT 运行：我们的解释会产生无限多个不同的文字，因此 apx-scc 不会产生有限基解释函数。

保持完备性。 如果我们确保在每次 NLSAT运行中只添加有限个辅助多项式，那么终止性就能再次得到保证。理由是，经过某个时刻后，apx-scc 的输出将始终等于原始的逐层构造 levelwise-scc。由于这是一个有限基解释函数，因此可以保证终止。

引理 1（终止）。 使用 *apx-scc* 的 $NLSAT$ 运行终止，当且仅当 *apx-criteria(*$P_j^{s[j]}$*)* 在该次运行中仅返回有限次 *true*。

为了利用这一引理，我们向 apx-criteria 提供额外信息，例如迄今为止近似单元的数量 n_{cells}。这样，当 n_{cells} 超过某个固定阈值时，apx-criteria 就可以返回 false，从而满足引理 1 的条件，并因此保证终止。

然而，最优阈值会因输入公式而异。一种更灵活的做法是，随着近似单元数量的增加，逐步加强判据。例如，只有当单元边界由某个 $p \in P$ 定义，且其次数为 $\deg_{x_p}[j] \geq c \cdot n_{cells} + d$，其中 $c,\ d \in \mathbb{Q}_{\geq 0}$ 时，我们才添加一个多项式。

这避免了例3中的情况，因为 $c \cdot n_{cells} + d$ 最终会超过相关多项式的次数，但其他结式开销更大的单元仍会被近似。重要的是，只有在添加的多项式为线性时，这才能确保终止；否则，构造中导出的多项式次数可能会无限增长，从而始终满足该判据。

也可以通过其他判据来提供保证，例如，针对相关多项式分别使用计数器。

4 变体

现在，我们介绍 apx-polys 和 apx-criteria 的若干种具体实现。前三种方法近似单元边界：给定样本点 $s_{[j]}$ 和某个 $p \in P$，其根 ξ 将成为 I_j 的边界，我们构造某个 p_*，使其根 $c \in \mathbb{Q}$ 位于 s_j 和 b 之间：$= \xi(s_{[j-1]})$。

简单方法。 在第 3 节开头，我们已经介绍了添加多项式 $x_j - c$ 的想法，这些多项式定义了 x_j 上的常数界 c。现在，我们进一步说明如何选择 c。虽然选择接近 b 的 c 对 I_j 的限制较小，但根据 p 的簇的形状，它可能会缩小底层单元。图3展示了这一点：近似边界越接近实际边界，I_1 就越小（即 x_1 的区间越小）。因此，选择什么并不显而易见。更重要的是，这种方法可能产生位数 log 较大的数 $c = {}^{num}/_{den}$（num）$+log$（den），从而显著增加将 c 代入高次多项式等操作的开销。因此，我们采用基于斯特恩-布罗科树 [7,31]的方法，选择位数最小的 c。

以下两种方法试图提供更好的单元边界近似，希望提高该单元对 NLSAT 的质量。

泰勒。 我们希望构造 p_*，使其在根 $(s_{[j-1]},\ c)$ 处的梯度等于 p 在其根 r 处的梯度：$= (s_{[j-1]},\ b)$：

$$p_* \quad = \quad \frac{\partial p}{\partial x_j}(r) \cdot (x_j - c) + \sum\nolimits_{k \in [j-1]} \left(\frac{\partial p}{\partial x_k}(r) \cdot (x_k - s_k) \right)$$

这是对 p 在 r处的一阶泰勒展开 所做的轻微修改，区别在于省略了常数项 $p(r)$，因为它始终为零；并且在第一项中使用 $(x_j - c)$ 而非 $(x_j - r_j)$。这样可确保 p_* 的根位于 $(s_{[j-1]},\ c)$，而不是 r。现在，显然

$$p_*(s_{[j-1]}, c) = 0 \text{ and } \frac{\partial p_*}{\partial x_k}(s_{[j-1]}, c) = \frac{\partial p}{\partial x_k}(r) \text{ for all } k \in [j].$$

其思路是， p和 p_* 的根函数在$s_{[j-1]}$附近的表现相似，如图 4a 所示。虚线是 p_3 在 $(s_{j-1},\ b)$ 处切线的根，然后将其平移，使其经过 c。

遗憾的是，如果某个样本坐标 s_k 是无理数，那么（1）我们不能使用术语（$x_k - s_k$），因为我们要求系数为有理数；（2）梯度$\partial p/\partial x_k$（r）更难计算，也可能是无理数。对于（1），我们省略当 s_k 为无理数时与 x_k 对应的 k 的加项；对于（2），我们通过求梯度的有理近似来解决：每个（无理）代数数都可以通过一个开区间隔离，该区间包含其定义多项式的单个根；通过二分法可以任意细化这个区间。对于 $k \in [j]$，设 $g_k \approx \partial p/\partial x_k$（$r$）为一个有理近似，则我们得到

$$p_* \quad = \quad g_j \cdot (x_j - c) + \sum_{k \in [j-1] \text{ s.t. } s_k \in \mathbb{Q}} \Big(g_k \cdot (x_k - s_k) \Big)$$

这两种补救措施都会降低近似的质量，因为某些梯度只是近似相等，或者根本没有被计入。

分段线性。我们不再只在单个点处近似边界，而是可以使用分段线性插值。为此，我们确定一个围绕 s_{j-1} 的区间 $D \subseteq \mathbb{R}$，使得 p 在该区间上可描绘

$$\{s_{[j-2]}\} \times D = \{(s_{[j-2]}, s') \in \mathbb{R}^{j-1} \mid s' \in D\} \subseteq \mathbb{R}^{j-1},$$

也就是说，$\mathrm{disc}_{x_j}[p]$ 以及 p 的某些系数在该集合上是符号不变的。现在我们知道，ξ是定义在 $\{s_{[j-2]}\} \times D$上的一个整体连续函数，而这是推导出有意义的插值所必需的。

然后，我们从 D 中选择 $k \in \mathbb{N}$ 个支撑点 $d_1 < \ldots < d_k$，使得 $s_{j-1} \in \{d_1, \ldots, d_k\}$；对于每个 $\ell \in [k]$，我们计算根函数在该支撑点处的值 $\xi(s_{[j-2]},\ d_\ell)$（实际上，我们只需在一个有理数区间内将其隔离）。通过选择一个与边界 $\xi(s_{[j-2]},\ d_\ell)$“接近”的值 $d'_\ell \in \mathbb{Q}$，可以对这些值分别进行下近似，使得 $(s_{[j-2]},\ d_\ell,\ d'_\ell) \in \mathbb{R}^j$位于该单元内。我们要求其中一个支撑点 d_ℓ 等于 s_{j-1}，因为这保证了 $(s_{[j-2]},\ d_\ell,\ d'_\ell)$（对于这个特定的 ℓ）位于该界与样本点之间。

因此，近似界由 $k-1$ 段组成，这些线段连接点 $\{(s_{[j-2]}, d_\ell, d'_\ell) \in \mathbb{R}^j \mid \ell \in [k]\}$，并由以下多项式的根定义：

$$p_*^{(\ell)} := (d_{\ell+1} - d_\ell)(x_j - d'_\ell) - (d'_{\ell+1} - d'_\ell)(x_{j-1} - d_\ell), \quad \ell \in [k-1].$$

也就是说，p_* 是一个分段函数，使得对于 $r \in \mathbb{R}^j$，有 $p_*(r) = p_*^{(\ell)}(r)$，如果 $d_\ell \le r \le d_{\ell+1}$（或者如果 $d_k \le r$，则有 $\ell = k$）。我们可以按照 [33,32], 中所述，通过对线性函数取最大值再取最小值来编码这个函数，从而推导出一个QF-NRA 公式。此外，我们可以采用 [26]中所述的技术，对后续的单元构造进行调整，使其能够处理此类复合区间界：我们计算每个 $p_*^{(\ell)}$与区间下方或上方多项式的结式，并

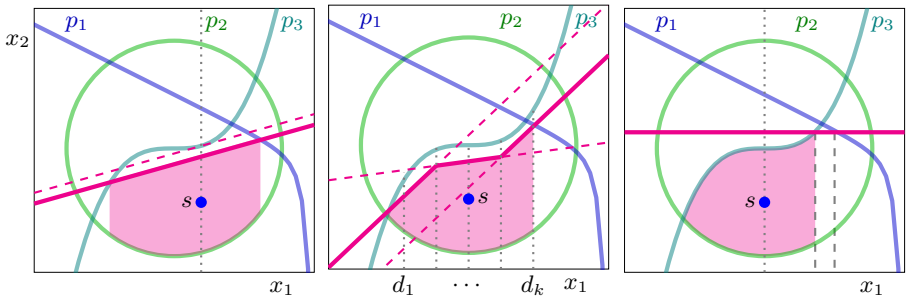

图4：修改后构造的变体：(a) 使用泰勒展开，(b) 使用分段线性界，以及 (c) 在单元外添加根。

筛除能够证明存在伪交点的结式的根。图 4b 展示了这一方法，其中虚线与其他对象的交点被筛除。

有两种情况无法应用这种方法（因此需要应用例如简单方法）：(1) 与泰勒方法类似，如果 s_{j-1} 是无理数，我们无法构造 p_*，因为对于某个 $\ell \in [k]$，$s_{j-1} = d_\ell$，因此无法使用术语 $x_{j-1} - d_\ell$。(2) 如果 $D = \{s_{j-1}\}$，即 D 只包含一个点，我们就无法选择多个支撑点。然而，构造近似边界的一段至少需要两个点。例如，当 p （原始边界定义多项式）的判别式在 $s_{[j-1]}$ 处有一个根 ξ_{disc} 时，就会出现这种情况。这意味着下一层无论如何都会坍缩为截面 $I_{j-1} = (x_{j-1} = \xi_{disc})$，因此样本点周围 p_* 的行为就没那么重要了。

单元外的根。 我们还可以引入多项式 p_*，其根函数 ξ_* 位于单元边界 ξ 与单元外的某个其他根 ξ' 之间。我们不计算 $\mathsf{res}_{x_j}[\xi.p, \xi'.p]$，而是计算 $\mathsf{res}_{x_j}[\xi.p, p_*]$ 和 $\mathsf{res}_{x_j}[p_*, \xi'.p]$，然后利用传递性，这可能会简化投影。虽然这样做的优点是顶层 I_i 保持不变，但底层单元仍可能受到新结式的限制。图 4c 展示了这一点。

近似准则。 除了改变如何 计算辅助多项式之外，我们还可以通过采用 apx-criteria 的不同实例，调整何时 计算它们。

由于我们的目标是避免代价高昂的结式计算，因此我们关注会影响当前层结式复杂度的量。一个简单（但在实践中有效）的判据取决于边界定义多项式 $p \in \mathbb{Q}[x_1, \ldots, x_i]$ 的次数：如果 $\mathsf{deg}_{x_i}[p]$ 超过固定阈值，就认为该界限已被近似。这种方法存在一些局限：（1）可能会出现

levelwise-scc 不会用 p 计算任何（代价高昂的）结式，但该判据仍会建议插入辅助多项式。虽然这违背了避免计算代价高昂结式的初衷，但并非毫无用处，因为它仍能简化单元描述。（2）对于不同类型的输入问题，最优阈值各不相同，而且很难事先选定合适的阈值。

针对第一个问题，我们还可以检查是否会计算 p 与另一个非线性多项式的结式。然而，在我们的初步实验中，这并未提升性能。第二个问题通过第 3.1 节中的动态终止准则得到（部分）解决；该准则中的次数阈值会随近似单元数量的增加而增大。

除了次数 $\deg_{x_i}[p]$，我们也可以考虑类似的度量，例如 p 的单项式的总次数之和；该度量已用于柱面代数分解投影排序 [15]。

迁移至 CAlC。与NLSAT类似，柱状代数覆盖（*CAlC*）方法 [1,27] 试图将部分赋值 $s \in \mathbb{R}^i$ 扩展为完整模型。它使用与逐层构造相同的理论框架，推导出应从搜索中排除的符号区间。特别是，它也通过使 $\text{res}_{x_{i+1}}[\xi.p, \xi'.p]$ 在序上保持不变，确保某些根函数 ξ，ξ' 不相交。我们可以应用该技术，动态引入一个在 ξ, ξ' 之间有根的多项式，以两个较简单的结式替代一个计算开销较大的结式，但这会缩小底层单元。这会引发类似的终止问题，可采用上文针对 SCC 所述的类似方法解决。

5 实验

我们在 SMT-RAT 求解器 [12,30],中实现了我们方法的若干变体，使我们能够使用其现有的逐层 SCC 和 NLSAT 实现。SMT-RAT 还采用基于傅里叶-莫茨金变量消去（FM） [18], 、区间约束传播（ICP） [20],和虚拟代换（VS） [2], 的解释后端。这些后端速度很快，但可能无法提供解释，尤其是对于次数为 3 或更高的多项式。后端按顺序调用（FM、ICP、VS、SCC），因此只有在其他所有后端都失败时，才需要进行单细胞构造。我们比较以下 SCC 变体：

基线：原始的逐层 SCC。
Simple-j：对于 j $\in \{3,4,5,6\}$，由次数
为 j 或更高的多项式定义的单元界，使用形式为 $x_j - c$ 的简单多项式进行近似。
将近似单元的数量限制为 50，以确保终止。Simple-*：Simple-j 变体的虚拟最佳组合方案。
Dy动态：采用第3.1节中的动态终止准则。如果定义多项式的次数大于或等于 $1/5 \cdot n_{cells} + 3$，则用形式为 $x_j - c$ 的简单多项式近似单元边界，其中 n_{cells} 是迄今为止已近似的单元数量。
泰勒：与动态类似，但使用第4节中介绍的泰勒近似。

PWL-j：对于 j $\in \{2, 4, 6\}$，与动态类似，但使用第4节中介绍的由 j 段组成的分段线性近似。
PWL-*：PWL-j 变体的虚拟最佳组合方案。
外部：如果下（分别为上）单元边界以及该边界下方（分别为上方）的另一个根函数由两个非线性多项式定义，且其中一个满足动态判据，那么我们会添加一个多项式（形式为$x_j - c$），其根位于单元外部、边界与另一个根之间，具体如第4节所述。

此外，我们还与当前最先进的 cvc5 [14]求解器（版本1.2.1）进行了比较，该求解器采用增量线性化和柱形代数覆盖。

我们使用了来自 SMT-LIB [3], 的 量词自由非线性实数算术 基准测试集，但只考虑其中 SMT-RAT 至少调用过一次 SCC 的 1684 个实例，因为在其他实例上各变体之间没有差异。测试在配置完全相同的 Intel®Xeon®8468 Sapphire CPU 上进行，每个核心的频率为 2.1 吉赫兹；每个实例的时间限制为 1 分钟，内存限制为 4GB。

结果汇总于图 5。虽然 cvc5 通常在整个量词自由非线性实数算术集合上求解的实例多于 SMT-RAT ，但仅基线求解器在我们的受限基准集合上的表现就已经优于 cvc5。

简单近似。 Simple-j 变体求解的实例数已经比基线多约 45 个。不过，这些变体（部分）各自在不同实例上表现出色：虚拟组合 Simple-* 求解的实例数至少比每个 Simple-j 变体多 18 个。

动态方法的表现几乎与之相当，求解的实例数比基线多 58 个；事实上，它是我们测试中表现最佳的（非组合）变体。有趣的是，已求解的可满足实例与不可满足实例数量的差距几乎相同（可满足实例多 30 个，不可满足实例多 28 个）。这表明，我们的解释不仅能帮助 NLSAT 更快地找到模型，而且不可满足问题的布尔结构通常仍允许通过欠近似单元覆盖搜索空间。

图6展示了动态方法与基线方法之间更详细的对比。从运行时间来看，动态方法可以在一秒内求解许多实例，而基线方法则需要数秒，甚至会超时。另一方面，动态方法明显更慢的实例极少，尽管其 SCC调用次数通常高得多。这是意料之中的：近似单元的计算速度更快，但覆盖的搜索空间更小。此外，我们的方法通常能显著降低所计算结式的最高次数。不过，我们没有发现判别式和系数的次数有明显降低。

其他变体 泰勒和 PWL-j 变体的表现略好于基线，但不如简单近似。看来，由此产生的开销超过了更优近似可能带来的收益。具体而言，我们的研究经常发现，辅助多项式及其结式中的有理系数位数很大。另一个原因可能是

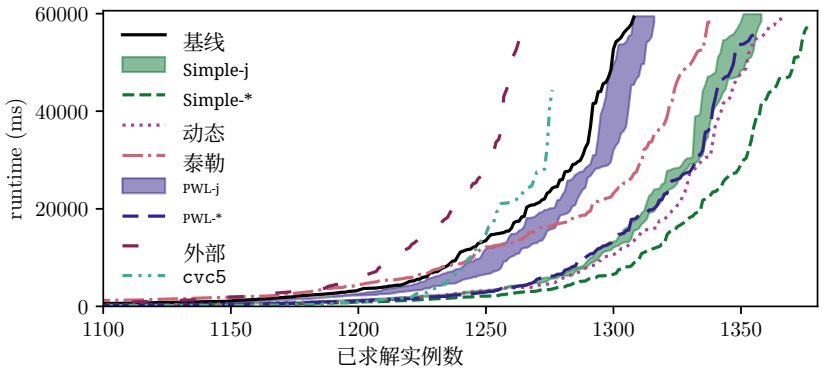

图 5：性能概况。紫色区域涵盖所有 PWL-j 变体，绿色区域涵盖所有 Simple-j 变体。

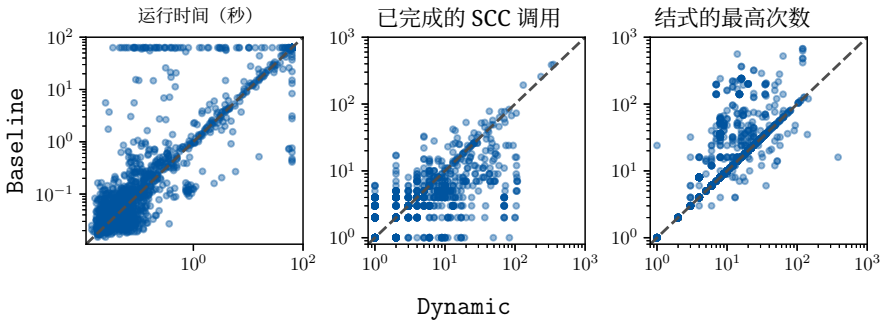

图 6：比较动态方法与基线的散点图

单元表示不够方便；这也得到了外部方法表现不佳的印证，它解决的实例比基线更少。与其他变体一样，它确实避免了一些结式，但并未简化单元表示。因此，这似乎是一个重要因素。

迁移到 *CAlC* 方法。我们还在 柱状代数覆盖（*CAlC*）方法的背景下测试了我们的修改 [27]，该方法同样在SMT-RAT 中实现。现在考虑 QF_NRA 基准测试集中的全部 12154 个实例，基线 CAlC 实现（求解了 9964 个实例）与我们的修改（求解了 9975 个实例）之间没有显著差异。再次，这可能表明NLSAT 尤其受益于更简单的单元描述（CAlC 中未使用），并且可以通过布尔推理弥补较小单元的不足。

关于不终止的注记。在第3.1节中，我们展示了我们的方法可能导致不终止，并出现某种“循环”行为，也提出了

应对这种情况的方法。自然，我们会想知道这种情况在实践中出现的频率。由于这种行为很难可靠地检测，我们没有具体数据。不过，关闭对近似单元数量的限制后，Simple-j 变体超时的情况显著增多（比设置限制时少解决约 40 个实例），这表明我们的终止准则十分重要。但在实践中还有其他影响因素：采样新值时，SMT-RAT 会优先选择简单数值，例如优先选择整数而非有理数，因此，即使没有硬性上限，这些变体也可能（暂时）摆脱循环：当实际单元边界与近似值之间的差距变小时，便可能出现这种情况。之后，它仍可能在计算的其他部分超时，或陷入另一个循环。

6 结论

我们修改了NLSAT 和 CAlC 的逐层单元构造，通过动态地将线性多项式插入投影中，避免在结式计算和对所得单元的描述中使用昂贵的高次多项式，代价是得到更小的单元——这些单元被视为“欠近似”。我们引入了多种变体来插入此类多项式，也提出了标准，以缓解NLSAT 和 CAlC 可能出现的不终止问题。

在实验中，我们的方法显著缩短了NLSAT的运行时间。值得注意的是，相对简单的欠近似效果最好，而更复杂的近似并未带来收益，这表明性能提升主要归功于NLSAT中更简单的根隔离。

未来研究有几个方向：首先，我们推测，更精细的 apx-criteria 变体或许能进一步提升效率。这并非易事：在此未展示的实验中，其他一些方法（例如基于多项式总次数之和，或涉及某一层所有多项式的次数）并未取得更好的结果。其次，我们的方法只能减少结式的计算量，尚不能减少判别式的计算量，而后者的影响更大。第三，对 CAlC 方法的基本修改影响甚微，或许还有改进空间。最后，我们通过在某个阶段恢复原始的逐层构造，来保证NLSAT 终止。另一种做法是改为考虑 δ-完备性（例如在 [16,6], 中所采用的做法），并将单元覆盖到一定精度 δ。

数据可用性。 我们的实现、实验结果以及用于复现实验的工具可在 https://doi.org/10.5281/zenodo.14916587 获取。

致谢。 J. Nalbach 和 V. Promies 获得了德国研究联合会项目 SMT-ART (AB 461/9-1) 的资助；J. Nalbach 和 P. Wagner 获得了德国研究联合会研究生院 2236 UnRAVeL 的资助。计算资源由亚琛工业大学项目 rwth1560 提供。

利益披露。 作者声明不存在与本文内容相关的竞争性利益。

参考文献

1. Ábrahám, E., Davenport, J.H., England, M., Kremer, G.：通过采用柱面代数覆盖的冲突驱动搜索，判定非线性实数算术约束的一致性。《逻辑与代数方法程序》**119**，100633（2021）。https://doi.org/10 100633 2 1016 2020 . /J.JLAMP. . . Ábrahám, E., Nalbach, J., Kremer, G.：将虚拟代入法嵌入模型构造可满足性演算框架。载于：England, M., Ganesh, V.（编）SC²17论文集。CEUR研讨会论文集，第1974卷。CEUR-WS.org（2017），https://ceur-ws.org/Vol-1974/EAb.pdf 3. Barrett, C., Stump, A., Tinelli, C.：SMT-LIB标准——2.0版。载于：SMT '10论文集（2010），http://theory.stanford.edu/~barrett/pubs/BST10.pdf 4. Barrett, C., Tinelli, C.：理论模可满足性，第305–343页。施普林格（ 2018 10 1007 978 3 319 10575 8_11 ）。https://doi.org/ . / - - - - . 5 Blanchette, J.C., Böhme, S., Paulson, L.C.：使用 SMT 求解器扩展 sledgehammer。载于：Bjørner, N.S., Sofronie-Stokkermans, V.（编）CADE'11论文集。LNCS，第6803卷，第116–130页。施普林格（2011）。https://doi.org/10. 1007 978 3 642 22438 6_11 10 1007 978 3 642 22438 6_11 / - - - - , 6 https://doi.org/ . / - - - - . Braußе, F., Korovin, K., Korovina, M., Müller, N.：一种用于求解非线性约束的CDCL风格演算。载于：Herzig, A., Popescu, A.（编）《组合系统前沿》。第131–148页。施普林格（2019）7. Brocot, A.：《齿轮的近似计算：新方法》。《计时评论》。《钟表匠科学与实践杂志》**3**，186–194（1861年），https://gallica.bnf.fr/ark:/12148 1661912 8 6/bpt k . Brown, C.W.：改进柱面代数分解的投影。《符号计算杂志》**32**0463 9 5 447 465 2001 10 1006 2001 ( )，– ( )。https://doi.org/ . /JSCO. . . Brown, C.W., Košta, M.：在柱面代数分解中构造单个单元。《符号计算杂志》**70**，14–48（2015）。https://doi.org/10.1016/J.JSC. 2014.09.024 10. Cimatti, A., Griggio, A., Irfan, A., Roveri, M., Sebastiani, R.：针对非线性算术和超越函数的理论模可满足性与验证的增量线性化。《ACM计算逻辑汇刊》**19**(3)，19:1–19:52（2018）。https: //doi.org/10 3230639 11 1145 3230639, 10 1145 . / https://doi.org/ . / . Collins, G.E.：通过柱面代数分解消去实闭域中的量词。载于：Barkhage, H.（编）《自动机理论与形式语言》，第2届GI会议，凯撒斯劳滕，1975年5月20日至23日。LNCS，第33卷，第134–183页。施普林格（ 1975 10 1007 3 540 07407 4_17 ）。https://doi.org/ . / - - - . 12 Corzilius, F., Kremer, G., Junges, S., Schupp, S., Ábrahám, E.：SMT-RAT：用于策略性与并行SMT求解的开源C++ 工具箱。载于：SAT'15论文集。LNCS，第9340卷，第360–368页。施普林格（2015）。https://doi.org/10.1007/978-3-319-24318-4 26 13 . Cox, D.A., Little, J., O'Shea, D.：《理想、簇与算法——计算代数几何与交换代数导论》（第2版）。《数学本科生教材》，施普林格（1997年）。https://doi.org/10.1007/978-3-662-41154-4 14. cvc5：用于理论模可满足性（smt）问题的高效开源自动定理证明器，https://cvc5.github.io/ 15. Dolzmann, A., Seidl, A., Sturm, T.：用于柱面代数分解的高效投影序。载于：Gutierrez, J.（编）ISSAC'04论文集。第111–118页。ACM（2004）。https://doi.org/10.1145/1005285.1005303, https://doi.org/10.1145/1005285.1005303

16. Gao, S., Kong, S., Clarke, E.M.: dreal：一种用于实数域非线性理论的 SMT 求解器。载于：Bonacina, M.P.（编）CADE'13 论文集。LNCS，第 7898 卷，第 208–214 页。施普林格（2013）。https://doi.org/10.1007/978-3-642-38574-2_1417. Giesl, J., Aschermann, C., Brockschmidt, M., Emmes, F., Frohn, F., Fuhs, C., Hensel, J., Otto, C., Plücker, M., Schneider-Kamp, P., Ströder, T., Swiderski, S., Thiemann, R.: 使用 aprove 自动分析程序终止性和复杂度。自动推理期刊**58**(1), 3–31 (2017)。https://doi.org/10.1007/S10817-016-9388-Y, https://doi.org/10.1007/s10817-016-9388-y 18. Jovanovic, D., Barrett, C., de Moura, L.: 模型构造可满足性演算的设计与实现。载于：FMCAD'13 论文集。第 173–180 页（2013）。https://doi.org/10.1109/FMCAD.2013.7027033 19. Jovanovic, D., de Moura, L.M.: 求解非线性算术问题。载于：IJ- CAR'12 论文集。LNCS，第 7364 卷，第 339–354 页。施普林格（2012）。https://doi.org/10.1007/ 978-3-642-31365-3 27_20. Kremer, G.: 面向非线性算术问题的柱面代数分解。博士论文，亚琛工业大学，德国（2020），https://publications. rwth-aachen.de/record/792185 21. Lazard, D.: 柱面代数分解的改进投影，第 467–476 页。施普林格（1994）。https://doi.org/10.1007/978-1-4612-2628-4_29 22. Li, H., Xia, B.: 通过样本点-单元投影求解多项式公式的可满足性。CoRR **abs/2003.00409**（2020），https://arxiv.org/abs/2003.00409 23. McCallum, S.: 柱面代数分解的改进投影运算。载于：Caviness, B.F.（编）EUROCAL'85 论文集。LNCS，第 204 卷，第 277–278 页。施普林格（1985）。https://doi.org/10.1007/3-540-15984-3_27724. McCallum, S.: 带等式约束的基于CAD的量词消去中的投影。载于：Geddes, K.O., Salvy, B., Dooley, S.S.（编）ISSAC '99 论文集。第 145–149 页。ACM（1999）。https://doi.org/10.1145/309831.309892 25. de Moura, L.M., Jovanovic, D.: 一种模型构造可满足性演算。载于：Giacobazzi, R., Berdine, J., Mastroeni, I.（编）VMCAI'13 论文集。LNCS，第 7737 卷，第 1–12 页。施普林格（2013）。https://doi.org/10.1007/978-3-642-35873-9_126. Nalbach, J., Ábrahám, E.: 在单个单元构造过程中合并相邻单元。载于：Boulier, F., Mou, C., Sadykov, T.M., Vorozhtsov, E.V.（编）CASC'24 论文集。LNCS，第 14938 卷，第 252–272 页。施普林格（2024）。https://doi.org/10.
1007 978 3 031 69070 9_15 10 1007 978 3 031 69070 9_15 / - - - - ,
27
https://doi.org/ . / - - - - . Nalbach, J., Kremer, G.: 面向量词的柱面代数覆盖方法扩展。CoRR **abs/2411.03070**（2024）。https://doi.org/10.48550/ARXIV. 2411.03070, https://doi.org/10.48550/arXiv.2411.03070 28. Nalbach, J., Ábrahám, E., Specht, P., Brown, C.W., Davenport, J.H., England, M.: 单个柱面代数单元的逐层构造。符号计算杂志 **123**, 102288（2024）。https://doi.org/10.1016/j.jsc.2023.102288 29. Promies, V., Nalbach, J., Ábrahám, E.: 单个代数单元的下近似（扩展摘要）。载于：Brown, C.W., Kaufmann, D., Nalon, C., Steen, A., Suda, M.（编）IJCAR'24 论文集。CEUR研讨会论文集，第 3717 卷，第 132–136 页。CEUR-WS.org（2024），https://ceur-ws.org/Vol-3717/short3.pdf 30. SMT-RAT，一款用于策略性并行求解理论模可满足性问题的工具箱，https://github.com/ths-rwth/smtrat 31. Stern, M.: 关于一个数论函数。纯粹与应用数学杂志（克雷尔杂志）**1858**(55), 193–220（1858年1月）。https: // doi.org/10.1515/crll.1858.55.193

32. Xu, J., Van Den Boom, T.J.J., De Schutter, B., Luo, X.L.: 连续分段仿射函数的最小合取范式表达式。IEEE 自动控制汇刊**61**(5), 1340–1345（2016年5月）。 https://doi.org/10. 1109 2465212 33 2015/TAC. . . Xu, J., van den Boom, T.J., De Schutter, B.: 不可约冗余格分段仿射表示及其在显式模型预测控制中的应用。载于：决策与控制（CDC 2014）。第 4416–4421 页。IEEE（2014年12月）。 https://doi.org/10.1109/CDC.2014.7040078