arXiv:2303.07825v1 [math.GR] 14 Mar 2023

语言、群与方程

LAURA CIOBANU 和 ALEX LEVINE

Abstract. 本文综述了过去十年间以形式语言刻画群中方程解的相关研究。首先介绍 Ciobanu、Diekert 和 Elder 的工作：他们证明了自由群中方程组的解，用约化字表示时可以表示为 EDT0L语言。本文概述了他们的算法，并说明自由群中的结果如何推广到双曲群。将解刻画为 EDT0L语言的方法具有很强的稳健性，正如 Levine 所展示的，许多群构造都能保留这一性质。

该领域最近的进展针对的是不具有负曲率的群，例如几乎阿贝尔群、整数海森堡群或可解鲍姆斯拉格-索利塔尔群；在这些群中，描述解的方法有别于负曲率群。在几乎阿贝尔群中，解集实际上是有理的，并且可以将其表示为 m-正则集。在海森堡群中，求出单个方程的解可归结为理解二次丢番图方程的解，并运用数论方法。在鲍姆斯拉格-索利塔尔群中，所用方法是组合性的，重点在于利用正规形之间的相互作用来求解特定类型的方程。

总而言之，EDT0L语言为许多重要的群类中看似具有很高复杂度的集合，提供了一种有效而简单的组合刻画。

1. Introduction

本综述旨在展示如何使用形式语言来描述群中的方程的解。方程是群的两个 Dehn 判定问题——字问题和共轭问题——的推广。群 G 中的一个方程是恒等式 $w = 1$，其中 w 是由 G 上的字以及一个有限变量集及其逆元组成的。方程的一个解是将群 G 中的一个元素赋给每个变量，使得将这些元素代入 w后得到一个与 1 等价的字。因此，群 G 中的共轭问题可以看作是求解方程 $X^{-1}gXh^{-1} = 1$，其中所有 $g, h \in G$。由于一般情况下字问题和共轭问题都是不可判定的，在任意群中求解方程也将不可判定，因此一个重要问题是：在哪些群类中能够成功求解方程。

自20世纪50年代以来，研究如何寻找算法来判定各种不同类别的群和幺半群中的方程是否有解，一直是一个活跃的研究领域。最早的重要肯定性结果由马卡宁于20世纪70年代取得，他证明了判定自由群或自由幺半群中的有限方程组是否可满足是可判定的 [27–29]。此后，马卡宁的工作得到拓展，研究者证明了双曲 [12, 33] 群以及某些相对双曲群 [11], 、可解鲍姆斯拉格-索利塔群 [24], 和直角阿廷群 [16], 中的方程可满足性是可判定的，等等。

虽然马卡宁的工作可以判定一个方程是否有解，却无法描述解集。后来，拉兹博罗夫提出了一种方法，可以构造自由群 [31, 32] 中方程组的解。由于解集通常是无限的，因此

2020 *Mathematics Subject Classification.* 03D05, 20F10, 20F65, 68Q45。*Key words and phrases.* 群中的方程，EDT0L语言。

表示解集的方式有多种，也可能根本无法表示。一种描述解的方法是将解集表示为一种语言，然后定义生成该语言的文法或自动机。形式语言是有限字母表上的任意字符串集合，而对语言进行分类的一种标准方法是乔姆斯基层级，它根据复杂度将语言分为正则语言、上下文无关语言、上下文有关语言和递归可枚举语言。由于上下文无关语言无法描述这些解集（见第 2.4 节），而上下文有关语言过于宽泛，无法以最佳方式刻画解的结构，因此需要一种不同的类别；本综述聚焦的正是这一类别。

EDT0L语言与解。2016年，第一作者、Diekert 和 Elder 成功地运用形式语言描述了自由群中方程组的解集[5]。所使用的语言类是 EDT0L语言。Diekert 和 Elder 将这一结果推广到了几乎自由群 [14], ； Diekert、Je˙z 和 Kufleitner 则将其推广到了直角阿廷群 [15]。随后又扩展到了双曲群 [6, 7], 、几乎阿贝尔群 [19] 以及双曲群的几乎直积 [26] 。

20世纪60年代，Lindemayer 引入了一系列称为 L系统的语言类，这些语言类最初用于研究生物体的生长。EDT0L（扩展确定性表0交互林德迈尔）语言是 L系统之一，由 Rozen- berg 于 1973 [36]年引入。20世纪70年代和80年代初，计算机科学领域对 L系统（包括 EDT0L语言）进行了深入研究；研究最多的 L系统族是带下标语言的子类。这些语言尽管历史上不属于乔姆斯基层级，却恰好处于上下文无关语言与上下文有关语言之间（见图1）。此前，人们只知道如何将自由幺半群上的二次字方程的解集描述为 EDT0L语言 [20]。自第一作者、Diekert 和 Elder 使用 EDT0L语言研究自由群中的方程以来，许多其他研究也使用 EDT0L语言（或称为 ET0L语言的类似语言类） [3, 4, 8] 来描述群中除方程解以外的集合。ET0L语言是 EDT0L语言和上下文无关语言的推广。此处省略其定义，因为本文不会直接用到它们，但后文会提及（定义见 [35]）。

将解写成 EDT0L语言时，一个内在环节是确立这些解的书写约定，并建立一种用字符串表示群元素的方法。假设 G 由有限集合 X 生成，并且固定了一种标准形 η，也就是说，固定了一种将元素统一表示为 X 上的字符串的方法（见定义 2.11）。我们总体采用的约定（对于几乎阿贝尔群，约定更加灵活）是：对于 G中的一个关于 n 个变量的方程，任意一个解（$g_1, \ldots, g_n$），其中 $g_i \in G$，都将转换为一个标准形元组（$g_1\eta, \ldots, g_n\eta$），然后转换为字母表 $X \cup \{\#\}$ 上的单个字符串（$g_1\eta$）# $\cdots$ #($g_n\eta$)，其中 # 是一个不属于 X的符号。用自然标准形来书写解，而不是用表示群元素的任意字符串，这是使方程相关算法和结果具有挑战性的因素之一。我们在表 1 中按大致的时间顺序汇总了该领域最初取得的成果。

许多研究考察了构造方程解的 EDT0L 系统所需的空间复杂度，其中输入是方程组的规模；在所有考察的情形中，均已证明该复杂度属于非确定性多项式空间。请注意，具有非确定性 $\mathcal{O}(n)$空间复杂度的算法，其时间复杂度为非确定性$\mathcal{O}(\exp(n))$。我们在表 1、表 3、表 2 和表 4 中总结了这些结果，包括空间复杂度（其中 n 是输入长度，即方程组的总长度）。对于表 3 中的构造， f 和 g 指的是用于构造这些结果的群的非确定性空间复杂度函数。

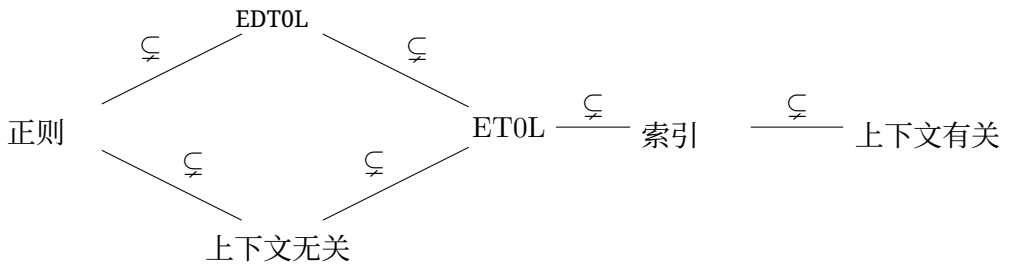

FIGURE 1. 从左向右阅读，可得语言类之间的包含关系 s

群类	生成集	解为	语言	非确定性空间复杂度	
Free	自由基	自由约化词	EDT0L	n对数 n	[5]
Free	any	唯一拟测地线	EDT0L	n对数 n	[7]
Free	any	拟测地线	ET0L	n对数 n	[7]
Free	自由基	所有词	ET0L	n对数 n	[7]
几乎自由	某些	某些拟测地线	EDT0L	n^2 对数 n	[14]
几乎自由	any	唯一拟测地线	EDT0L	$n^2 \log n$	[7]
几乎自由	any	拟测地线	ET0L	$n^2 \log n$	[7]
无挠双曲群	any	唯一拟测地线	EDT0L	n^2 对数 n	[7]
无挠双曲群	any	拟测地线	ET0L	n^2 对数 n	[7]
含挠双曲群	any	唯一拟测地线	EDT0L	n^4 对数 n	[7]
含挠双曲群	any	拟测地线	ET0L	$n^4 \log n$	[7]

TABLE 1. 自由群和双曲群的结果概述

自由群与双曲群。表 1 中的结果都针对双曲群；事实上，最早证明方程求解可判定的正是这类群：先是马卡宁对自由群 [27–29], ，随后是 Rips 和 Sela 对无挠双曲 [33], ，以及 Dahmani 和 Guirardel [12] 对所有双曲群。双曲群先于其他类群得到研究，粗略而直观地说，是因为负曲率有助于限制双曲凯莱图中与薄三角形相关的路径数量和长度：求解三角方程组时，测地三角形至关重要，而双曲群中的任意方程都可以通过将其转化为三角方程组来处理。此外，双曲群尤其适合用正规形来表达解，因为最自然的正规语言（所有测地线、拟测地线、短字典序正规形等）对于任意生成集都是正则的。第一作者和 Elder 在其研究中大量运用了双曲凯莱图的几何性质以及正规形的正则性。

其他类别的群。在负曲率之外，研究的主要群类别是（几乎）阿贝尔群和幂零群。对于几乎阿贝尔群，无论是算法复杂度还是形式语言方面，都可以取得不少成果（见第 5 节）；但关于幂零群的结果非常有限，因为在所有非阿贝尔自由幂零群中，求解方程组的不可判定性早已出现 [17]。不过，考察幂零群中的单个单变量方程（例如海森堡群中的方程）却颇有成果，并且

由此与二次丢番图方程建立了一些优美的联系（见第 6 节）。这些结果汇总于表 2。尽管在所有自由幂零群中，方程组的可满足性都是不可判定的，但对于单个方程，仍有一些正面结果。Duchin、Liang 和 Shapiro [17] 证明，在所有交换子子群几乎循环的二类幂零群（包括所有海森堡群）中，单个方程的可满足性是可判定的。Roman'kov证明，并非所有二类幂零群都满足这一性质 [34]。对于幂零级更高的群，即便只求解单个方程，在3阶且秩足够大的自由幂零群中也会出现不可判定性 [17]。

群类	生成集	解作为	语言	非确定性空间复杂度	
自由阿贝尔群	自由阿贝尔基	标准正规形	EDT0L	n^2	[13, 19]
几乎阿贝尔群	标准生成集	横截正规形	EDT0L	n^2	[19]
海森堡群（单变量）	马尔采夫生成元	马尔采夫正规形	EDT0L	n^8(对数 n)^2	[25]
BS(1,n)（仅限特定方程）	标准生成集	参见第 8 节	EDT0L		[18]
右角阿廷群	标准生成集	标准正规形	EDT0L	n对数 n	[15]
二面体阿廷群	继承生成集	继承正规形（准测地）	EDT0L	$n^2 \log n$	[26]
虚拟直积双曲群的	继承生成集	继承正规形（准测地）	EDT0L	$n^4 \log n$	[26]

TABLE 2. 其他群的结果摘要

关于群中方程求解，一个最自然的问题是：可判定性或语言复杂度是否会在群构造下保持不变？第二作者近期的研究在这方面得出了一些关于解集的肯定结果，这些结果汇总于表 3，并在第 7 节中作了进一步阐述。

群构造	生成集	解为	语言	非确定性空间复杂度	
直积	标准生成集	标准正规形	EDT0L	$f + g$	[26]
有限指数子群	施莱尔生成元	施莱尔正规形	EDT0L	f	[26]
广 wreath 积带有限群	标准生成集	标准正规形	EDT0L	f	[26]

TABLE 3. 群构造结果概要

利用表 3 中的构造，第二作者证明，几乎是双曲群直积的群中的方程组，其解语言是 EDT0L [26]。由于二面体 Artin 群几乎是自由群的直积，因此这些群中的方程的解可以表示为 EDT0L语言，尽管其正规形并非标准形式。关于鲍姆斯拉格-索利塔群 [18] 的最新研究表明，某些类别的方程（包括乘法表）可以相对于一种自然标准形写成 EDT0L语言。这项研究的作者猜想，并非所有相对于这种标准形的方程组都具有 EDT0L 解语言，因此这可能是首个具有可判定丢番图问题、但相对于自然正则标准形不具有 EDT0L 解语言的群的例子。这些结果汇总于表 2。

在证明自由群中的方程组的解是 EDT0L 时，作者首先将问题归结为证明带反演的自由幺半群中的方程组的解是 EDT0L [5]。对于直角阿廷群，也需要采用类似的方法，将问题归结为迹幺半群 [15]。这些关于幺半群的结果概述于表 4。

幺半群类别	生成集	解作为	语言	非确定性空间复杂度	
自由幺半群	标准基	字符串	EDT0L	n对数 n	[5]
迹幺半群	standardgen 集	标准正规形	EDT0L	n对数 n	[15]

Table 4. 幺半群结果概要

约束。添加约束，也就是要求解属于特定集合，在证明解语言属于 EDT0L 时也曾得到考虑。添加有理约束可以将特定变量限制在某些集合中，例如有限生成子群或共轭类。可识别约束的限制更严格，但可用于强制解属于有限指数子群的陪集。在任何一类群中，只要方程组的解语言属于 EDT0L，带有可识别约束的方程组也同样如此 [26]。然而，有理约束带来了更多挑战。

已证明，带有理约束的方程组在自由群 [5] 和几乎自由群 [14]中具有 EDT0L 解语言。对于双曲群，研究表明，将一种称为准等距嵌入有理约束、限制稍严的约束类添加到方程组中，仍能保持具有 EDT0L 解这一性质 [7]。有理约束也可以添加到几乎阿贝尔群中 [26]。对于其他群类，这可能比较困难。例如，在海森堡群中，是否存在一种算法来检查给定元素是否属于给定有理集，目前仍未解决。

Contents

1. 引言 1 2. 预备知识 6 2.1. 群方程 6 2.2. 形式语言 7 2.3. 正则语言 7 2.4. 解语言 8 2.5. 空间复杂度 9 2.6. 带约束的方程 10 3. EDT0L语言 10 4. 自由群和双曲群中的解 12

4.1. 自由群算法概述。13 4.2. 无挠双曲群算法概述。16 5. 几乎阿贝尔群 18 6. 海森堡群与整数环上的丢番图方程 19 7. 闭包性质 22 8. 鲍姆斯拉格-索利塔群 24 致谢 25 参考文献 25

2. Preliminaries

记号 2.1. 我们将函数写在其自变量右侧，写作 xf 或 $(x)f$，而非$f(x)$。这意味着阅读由复合函数组成的长串时，我们从左向右阅读。

2.1. 群方程。我们在此定义群中的一个方程组。

定义 2.2. 设 G 是一个群， $\mathcal{X}$ 是一个有限变量集。 G 中以变量 $\mathcal{X}$为变量的有限方程组，是 $\mathcal{E}$的一个有限子集，其中 $G * F(\mathcal{X})$ 是由有限集 $\mathcal{X}$生成的自由群。如果 $\mathcal{E} = \{w_1, \ldots, w_n\}$，我们将 $\mathcal{E}$视为一个方程组，并写作 $w_1 = w_2 = \cdots = w_n = 1$。方程组 $w_1 = \cdots = w_n = 1$的一个解是一个同态 ϕ: $F(\mathcal{X}) \to G$，且满足 $w_1\phi = \cdots = w_n\phi = 1_G$，其中 ϕ是将 ϕ扩展为从 $G * F(\mathcal{X})$ 到 $\to G$的同态，¯由以下规则定义：对所有 $g \in G$， $g\phi = g$。

注 2.3. 对于变量为 X_1、...、 X_n 的方程，其解通常表示为由群元素组成的元组 $(x_1, \ldots, x_n)$，而不是同态。我们可以通过对每个 i 定义 $X_i \mapsto x_i$ ，从该元组得到同态。

下面给出一些群中的方程的入门示例。

例 2.4. $\mathbb{Z}$ 中的方程是整数上的线性方程，可以用初等线性代数来判断其可满足性，并描述其解。

例如，若我们将 a 写作 $\mathbb{Z}$ 的自由生成元，那么

$$a^3Xa^{-5}Y^{-1}a^7Ya^{-1}X = 1 \tag{1}$$

是 $\mathbb{Z}$ 中的一个方程 。我们可以用加法记号将 (1) 改写为（并用 3 代替 a^3）

$$3 + X - 5 - Y + 7 + Y - 1 + X = 0. \tag{2}$$

利用 $\mathbb{Z}$ 是阿贝尔群这一事实，我们可以对 (2) 进行变形，得到以下具有相同解集的方程：

$$2X + 4 = 0.$$

因此，我们的解集（写成元组形式）为

$$\{(2, y) \mid y \in \mathbb{Z}\}.$$

例 2.5。可以通过一个方程来检查某个元素是否为交换子。在群G中，方程 $XYX^{-1}Y^{-1} = g$（其中 X和 Y是变量，且 $g \in G$）有解，当且仅当 g 可以写成交换子的形式。这是一个重要的二次方程，在自由群中，可以使用尼尔森变换完整地枚举其解（例如参见 [21]）。

2.2. 形式语言。我们简要介绍语言的概念。

定义 2.6. 字母表是一个有限集合。字母表 Σ 上的一个字是由 Σ中的元素组成的有限序列，通常记作 $a_1 \cdots a_n$，而不是 $(a_1, \ldots, a_n)$。Σ上的一个语言是 Σ上任意一组字。

例 2.7.

(1) 英语，也就是说，英语词典中收录的字符串集合，是字母表 $\{a, \ldots, z\}$上的一种形式语言。
(2) 字母表 $\{a, b, a^{-1}, b^{-1}\}$ 上的自由约化词集合（也就是说，不含形如 $aa^{-1}, a^{-1}a, bb^{-1}$ 或 $b^{-1}b$的子字符串的字符串集合）也是一种语言。

记号 2.8.

(1) 我们用 ε 表示任意字母表上的空词；(2) 字母表 Σ 上的所有词构成的集合记为 Σ^*；(3) 如果 Σ 是群的一个有限子集，我们有时会使用 $\Sigma^{\pm} = \Sigma \cup \{a^{-1} \mid a \in \Sigma\}$，尤其是在 Σ 是群的有限生成集时，此时每个群元素都可以表示为 $\Sigma^{\pm}$中的一个字。不过，我们有时也会直接说明某个生成集“对逆封闭”，而不使用 $\pm$，以简化记号。

2.3. 正则语言。我们将在全文中用到正则语言的定义，包括在 EDT0L 语言的定义中，以及在可添加到方程组中的约束类型的定义中。我们先从定义开始。

定义 2.9. 有限状态自动机（FSA）是一个 4 元组 $\mathcal{A} = (\Sigma, \mathcal{G}, q_0, F)$，其中

(1) Σ 是一个（有限）字母表；(2) $\mathcal{G}$ 是一个有限有向图，其边以 $\Sigma \cup \{\varepsilon\}$标记；(3) $q_0 \in V(\mathcal{G})$ 称为初始状态；(4) $F \subseteq V(\mathcal{G})$ 称为接受状态集合。

如果 $w \in \Sigma^*$ 可被 $\mathcal{A}$ 接受，那么在 $\mathcal{G}$ 中存在一条从 q_0 到 F中某一点的路径，使得沿着该路径前进时，依次读取各条边的标签即可得到 w。$\mathcal{A}$ 所接受的语言是 $\mathcal{A}$所接受的所有词组成的集合。

如果某个有限状态自动机接受语言 $L \subseteq \Sigma^*$，则称该语言为正则语言。

例 2.10。

(1) 语言 $\{a, b\}^*$ 是正则的。它被图 2 中的自动机接受 .

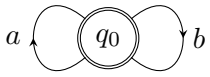

FIGURE 2. 关于 $\{a,\ b\}^*$ 的有限状态自动机，初始状态为 q_0，接受状态为 q_0

（2）语言 $\{w \in \{a,\ b,\ a^{-1},\ b^{-1}\}^* \mid w$ 是自由约化的$\}$ 是正则的。它被图 3 中的有限状态自动机接受

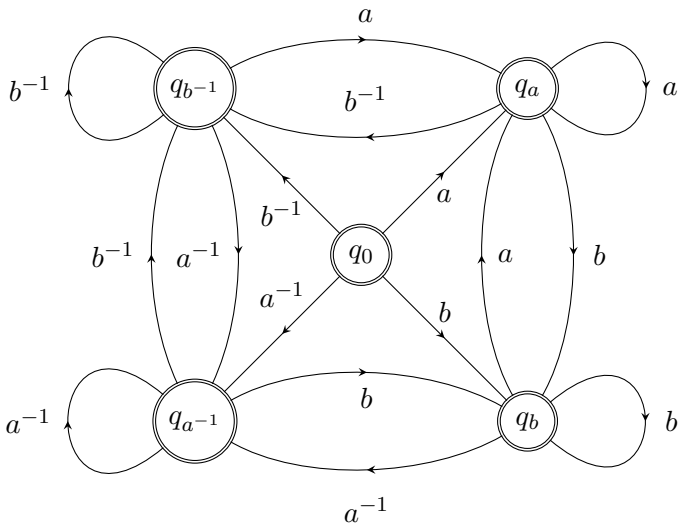

FIGURE 3. 关于 $\{w \in \{a,\ b,\ a^{-1},\ b^{-1}\}^* \mid w$ 的有限状态自动机，其自由约化为$\}$，其中 q_0 是初始状态，且每个状态都是接受状态

(3) 语言 $\{a^n\#a^n \mid n \in \mathbb{Z}_{\geq 0}\}$ 不是正则的。证明这一点需要用到抽引引理（例如参见 [23]）。

2.4. 解语言。现在我们说明如何将解集表示为语言。首先为群中的元素定义一种正规形。

定义 2.11. 设 G 是一个群，Σ 是 G的有限生成集。关于 Σ，G的正规形是一个函数 η: $G \to (\Sigma^{\pm})^*$，它保持 $\Sigma^{\pm}$不变，并且对于任意 $g \in G$，字 $g\eta$ 表示元素 g。

正规形 η 称为

(1) 若 $\mathrm{im}\,\eta$ 是关于 $\Sigma^{\pm}$的正则语言，则称其为正则；(2) 若 $\mathrm{im}\,\eta$ 仅由 G中的测地词构成，则称其为测地，相对于 Σ而言；也就是说，对所有 $g \in G$，$|g\eta| = |g|_{(G,\Sigma)}$；(3) 若存在 λ、$\mu > 0$，使得对所有 $g \in G$，$|g\eta| \leq \lambda|g|_{(G,\Sigma)} + \mu$，则称其为 (λ, μ)-拟测地线。若相关常量并不重要，我们就称其为拟测地线。

注意，由于 η 是一个函数，根据定义，每个元素都有唯一的代表。我们常常滥用记号，将正规形 η 视为 $\operatorname{im}\eta \subseteq (\Sigma^{\pm})^*$的一个子集，而不是一个函数，因为这样有时更方便。

下面给出一些标准形的典型例子。

例 2.12。

(1) 由生成元 $\{a,b\}$ 表示的 $\mathbb{Z}^2$ 的标准正规形是 $\{a^n b^m \mid m,n \in \mathbb{Z}\}$ over$\{a,a^{-1},b,b^{-1}\}$。这种正规形是测地的且为正则。(2) 由生成元 $\{a,b\}$ 表示的自由群 $F(a,b)$ 的标准正规形是由 $\{a,a^{-1},b,b^{-1}\}$ 上的约化字组成的集合。这种正规形是测地的且为正则，如例 2.10 (2) 所示。(3) 更一般地，双曲群 G 在带有顺序的生成集上，其短字典序代表元构成一种测地且正则的正规形（详情见 [23] ）。(4) 鲍姆斯拉格-索利塔群 BS(1, k) 中元素所用的一种标准正规形见引理 8.1。这种正规形是正则的，但不是测地的。

现在，我们可以根据给定的正规形，将解表示为语言。

定义 2.13. 设 G 是一个群，其生成集为 Σ，并设 $\eta\colon G \to (\Sigma^{\pm})^*$ 是关于 Σ的 G的一个正规形。设 $\mathcal{E}$ 是 G 中的一个方程组，其解集为 $\mathcal{S}$。$\mathcal{E}$ 的解语言是指以下语言

$$\{(g_1\eta)\#\cdots\#(g_n\eta) \mid (g_1,\ \ldots,\ g_n) \in \mathcal{S}\}$$

在 $\Sigma^{\pm} \sqcup \{\#\}$上。

注 2.14. 请注意，通常情况下，上下文无关语言并不适用于按照上述约定描述方程的解。即使在 $\mathbb{Z}$ 中，方程组 $\{X = Y, Y = Z\}$ 相对于标准正规形的解语言也是 $\{a^x\#a^x\#a^x \mid x \in \mathbb{Z}\}$，而这并不是上下文无关语言。

2.5. 空间复杂度。我们简要定义空间复杂度。关于复杂度的更全面引言，包括标准的大 O 记号 $\mathcal{O}$，可参见 [30]。

定义 2.15. 设 $f\colon \mathbb{Z}_{\geq 0} \to \mathbb{Z}_{\geq 0}$ 是一个函数。若某个算法可以由满足以下条件的非确定性图灵机执行，则称该算法运行于非确定性空间复杂度 f:

(1) 只读输入带；(2) 只写输出带；(3) 读写工作带，使得图灵自动机在输入长度为 n 时的一条计算路径至多使用 $\mathcal{O}(nf)$ 个工作带单元。

如果某个算法在 NSPACE(f) 中运行，其中 $f\colon \mathbb{Z}_{\geq 0} \to \mathbb{Z}_{\geq 0}$ 是某个线性（分别为二次、分别为多项式）函数，则称该算法在非确定性线性（分别为二次、分别为多项式）空间内运行。

注 2.16. 我们通常会说，定义语言的文法或自动机可在非确定性空间复杂度(f)内构造。这意味着存在一个在非确定性空间复杂度(f)内运行的算法，它接受一个指定的输入（有时是其他文法或自动机），并输出所需的文法或自动机。

2.6. 带约束的方程。在群 G 中，带有有理（可识别）约束 $\mathcal{E}$ 的有限方程组是一个有限方程组 $\mathcal{F}$，其变量为 $X_1, \ldots, X_n$，并附有一个由 G 的有理（可识别）子集 $R_1, \ldots, R_n$ 组成的元组。$\mathcal{E}$ 的解是 $\mathcal{F}$ 的一个解 ϕ，且对所有 i， $X_i\phi \in R_i$ 均成立。

这里介绍幺半群的有理子集和可识别子集的基本定义。这两种类型都用作群中方程的变量约束；我们将利用可识别约束证明：对于方程组的解构成 EDT0L 语言的群类，在取有限指数子群后仍然封闭。虽然我们主要关注群的有理子集，但定义 EDT0L 语言时需要用到幺半群的有理子集。

定义2.17。设 S 是一个幺半群（例如，一个群），且 Σ 是 S的一个（幺半群）生成集。定义 π: $\Sigma^* \to S$ 为自然同态。我们称子集 $A \subseteq S$ 为

(1) 可识别的，如果 $A\pi^{-1}$ 是 Σ上的正则语言；(2) 有理的，如果存在 Σ 上的正则语言 L，使得 $A = L\pi$。

请注意，这个定义与生成集的选择无关。

注2.18. 可识别集是有理的。

例2.19. 任意幺半群的有限子集都是有理的。群中的有限子集 G 是可识别的，当且仅当 G 是有限的 [22]。任意群的有限指数子群都是可识别的，因此也是有理的。

Herbst 和 Thomas 证明，群 G 中的可识别集总是其 G [22]某个有限指数正规子群的陪集的有限并。这个结果被用于证明许多关于可识别集的事实。

3. EDT0L languages

最初的定义归功于 Rozenberg [36], ；然而，有理控制往往能让 EDT0L语言的处理容易得多，这种用法则归功于 Asveld [2]。

定义3.1. EDT0L系统是一个元组 $\mathcal{H}$ =(Σ, C, ω, $\mathcal{R}$)，其中

(1) Σ 是一个字母表，称为（终结）字母表；(2) C 是 Σ的一个有限超集，称为 $\mathcal{H}$的扩展字母表；(3) $\omega \in C^*$ 称为初始字（或种子字）；(4) $\mathcal{R}$ 是 End(C^*) 的一个有理子集，称为 $\mathcal{H}$的有理控制。

$\mathcal{H}$ 接受的语言为

$$L(\mathcal{H}) = \{\omega\phi \mid \phi \in \mathcal{R}\} \cap \Sigma^*.$$

由 EDT0L系统接受的语言称为 EDT0L语言。

记号3.2。在为某个扩展字母表 C定义 C^* 的自同态时，作为 EDT0L系统定义的一部分，我们通常会通过说明每个自同态将 C中的每个字母映射到哪里来定义该自同态。如果在自同态的定义中没有为某个字母指定像，我们就称该字母在此自同态下保持不变。

下面给出一些 EDT0L语言的例子和反例。

例3.3。

(1) 语言 $L = \{a^{n^2} \mid n \in \mathbb{Z}_{>0}\}$ 是字母表 $\{a\}$ 上的 EDT0L语言。可以将 $C = \{\perp,\ s,\ t,\ u,\ a\}$ 视为一个接受 L 的 EDT0L系统的扩展字母表，以 $\perp$ 为起始词，并使用图4中的有限状态自动机来定义有理控制。

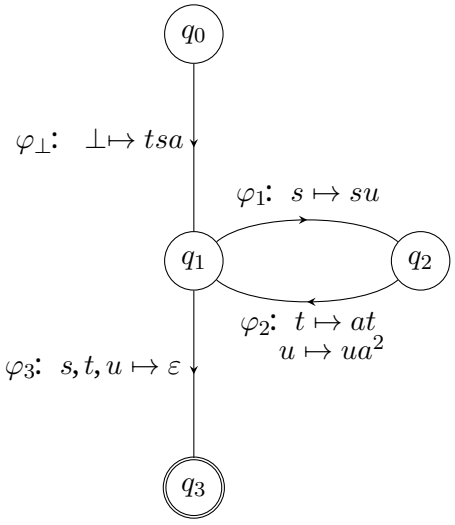

FIGURE 4. $L = \{a^{n^2} \mid n \in \mathbb{Z}_{>0}\}$的有理控制，初始状态为 q_0，接受状态为 q_3。

有理控制也可以写作 $\varphi_\perp(\varphi_1\varphi_2)^*\varphi_3$。该语言不是上下文无关语言。

(2) 考虑带有表示 $\langle a \mid \rangle$ 的方程 $XY^{-1} = 1$ in $\mathbb{Z}$。相对于标准正规形的解语言为 $L = \{a^n\# a^n \mid n \in \mathbb{Z}\}$，其字母表为$\{a,\ a^{-1},\ \#\}$。语言 L 是 EDT0L；我们的方程组将具有扩展字母表 $\{\perp, \#\ \ a,\ a^{-1}\}$，以及由图 5 定义的有理控制。

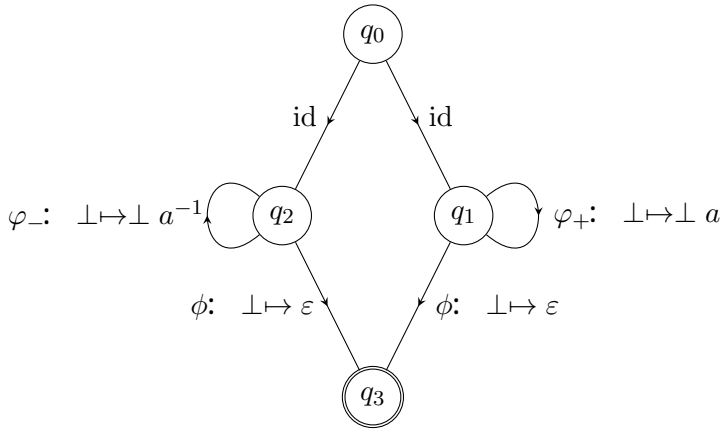

FIGURE 5. $L = \{a^n\#a^n \mid n \in \mathbb{Z}\}$ 的有理控制，初始状态为 q_0，接受状态为 q_3。

有理控制也可以用有理表达式 $\{\varphi_-^*,\ \varphi_+^*\}\phi$ 表示。这个语言不是正则的，这可以用抽引引理（[23], 定理 2.5.17）证明，但它是上下文无关的。

(3) 自由群 $F(a,b)$ 中关于生成元 a,b 的方程 $XY^{-1}=1$，相对于标准正规形（即自由约化词）的解语言等于

$$L=\{w\#w \mid w \text{ freely reduced over } \{a,a^{-1},b,b^{-1}\}\}.$$

语言 L 是 EDT0L（与 [5] 的例 1.3 类似），但不是上下文无关的。这一结论可推广到任意有限秩自由群。

我们为 L 构造一个 EDT0L系统。我们的字母表是 $\Sigma=\{a,\ b,\ a^{-1},\ b^{-1},\#\}$，扩展字母表为 $C=\Sigma\cup\{\perp\}$，起始词为 $\perp\ \#\ \perp$。对于每个 $x\in\{a,\ b,\ a^{-1},\ b^{-1}\}$，定义 $\varphi_x\in$ End(C^*) 为 $\perp\ \varphi_x=x\ \perp$。令 $\mathcal{R}\subseteq$ End(C^*) 为这样得到的有理集：将图 3 中有限状态自动机里每条边上的 $x\in\{a,\ b,\ a^{-1},\ b^{-1}\}$ 标签替换为 φ_x。这样我们得到

$$\{(\perp\ \#\ \perp)\phi \mid \phi\in\mathcal{R}\}=\{w\perp\ \#w\ \perp \mid w \text{ freely reduced over } \{a,\ b,\ a^{-1},\ b^{-1}\}\}.$$

将有理控制与由 $\perp\ \psi=\varepsilon$ 定义的映射 $\psi\in$ End(C^*) 后置串接（可通过添加一个新的接受状态、将其设为唯一接受状态，并从每个原接受状态添加一条带有 ψ 标签的边指向该新状态来实现），即可得到新的有理集 $\mathcal{R}'\subseteq$End(C^*)，使得

$$L=\{(\perp\ \#\ \perp)\phi \mid \phi\in\mathcal{R}'\}.$$

(4) 语言

$$\{w\in\{a,\ b,\ a^{-1},\ b^{-1}\} \mid w=_{F(a,\ b)}1\}$$

是上下文无关的，但不是 EDT0L [8]。

(5) 语言

$$\{w\in\{a,\ b\}^* \mid |w|=n^2 \text{ for some } n\in\mathbb{Z}_{\geq 0}\}$$

不是 EDT0L（ [35]的推论 IV.3.4）。不难证明，它是 ET0L，但不是上下文无关的。

定理 3.4（[26], 引理 2.15）。EDT0L语言类在以下运算下封闭：

(1) 有限并；(2) 与正则语言的交集；(3) 连接；
(4) 克莱尼星闭包；(5) 在自由幺半群同态下的像。

应用这些运算中的任意一种，都不会影响为涉及的语言构造方程组所需的空间复杂度，前提是（2）中的正则语言和（5）中的同态都能在常数空间内构造。

4. Getting the solutions in free and hyperbolic groups

本节概述在自由群和双曲群中生成 EDT0L 描述的算法。我们重点介绍自由群中的算法，随后利用该算法求解无挠双曲群中的方程。从带挠性的双曲群约化到几乎自由群也有类似的方法，但过程更为复杂。由于挠性给相关算法和论证增加了很大的难度，为了让本综述更易于理解，我们请读者参阅 [14] 和 [7]，了解带挠情形的详情。同样地，我们将讨论范围限制在群中的方程，不

涉及不等式或约束，以使论述更加简明。

算法概述。求解自由群或幺半群中方程的算法，核心在于构造一个有限标记图：该图（或称自动机）具有 (i) 若干顶点，顶点标记为大小有界的方程，这些方程是输入方程的变体，并附有一些额外数据；(ii) 有向边，对应于应用于方程的变换；边则标记为自由幺半群 C^* 的自同态，其中 C 是一个包含群或幺半群生成元的有限字母表。将该图视为一个（非确定性）有限状态自动机（有限状态自动机），它会生成 C^* 上自同态的有理语言。在 [5] 中，我们证明，将所有此类自同态应用于某个特定“种子”字，所得集合恰好是输入方程的全部解（以约化字表示）。因此，根据 Asveld 的定义 [2], ，解集是 EDT0L语言，因而也是索引语言。此外，只需检查该图是否为空，就能判定解的数量是零、有限还是无限。

下面我们介绍自由群算法，并请读者参阅 [5]了解自由幺半群情形。它与自由群情形相似（需要采用不同的记号，但方法并无不同）。

4.1. 自由群算法概述。以下总结了求解自由群中方程的主要步骤。完整细节及补充说明见第一作者、Diekert 和 Elder 的论文 [5]。在下一小节中，我们将把该算法称为 CDE 算法。

令 $F(A)$ 为由逆封闭集 A （见记号 2.8）生成的自由群，并令 $(M(A)^-)$ 为 A 上的带对合的自由幺半群，定义如下。集合A 上的对合是一个映射 $x \mapsto \overline{x}$ ，满足对所有 $x \in A$，都有 $\overline{x} = x$ ；该映射可通过应用 $\overline{xy} = \overline{y}\,\overline{x}$扩展为 A 上的自由幺半群上的对合。因此，$(M(A)^-)$ 就是在 A 上生成的自由幺半群，再加上一个对合；该对合在自由群 $F(A)$中起到逆元的作用，但这里不允许消去。

预处理。算法接下来将输入方程（在自由群中）转换为带对合的自由幺半群中的方程，如图6所示。预处理的第一步是通过引入新变量，将变量集 $\mathcal{X} = \{X_1, \ldots, X_k\}$ 上的方程 $U = 1$ in $F(A)$ 转换为由某些 $s > 1$ 个方程 $U_j = V_j$, $1 \le j \le s$ 组成的三角系统 $\Delta_{F(A)}$ ；也就是说，$\Delta_{F(A)}$ 中的所有方程都满足 $|U_j| = 2$ 和 $|V_j| = 1$，对所有 j 均如此，并且 $U = 1$ 的初始变量 $X_1, \ldots, X_k$ 是系统 $U_j = V_j$ 中出现的变量集的子集（详情见 [5, Section 4] ）。

第二步中，我们离开群的框架：从三角自由群系统 $\Delta_{F(A)}$ 出发，在带对合的自由幺半群 $(M(A)^-)$ 上构造一个由某些 $s' > 1$ 个方程$U'_j = V'_j$, $1 \le j \le s'$ 组成的三角系统 $\Delta_{M(A)}$ 。这是利用了如下事实：在自由群 $F(A)$ 中，若三角方程 $XY = Z$ （其中 X, Y, Z 是变量或常量），那么它在 A 上的约化字中有解，当且仅当存在字 P, Q, R，满足 $X = PQ, Y = Q^{-1}R, Z = PR$，且 P 与 Q、Q^{-1} 与 R、以及 P 与 R 之间均不发生消去（见 [5, Lemma 4.1]）。此时，我们引入有理约束，要求系统 $\Delta_{M(A)}$ 在 $M(A)$ 中的解为自由约化词，也就是说，不允许出现 $a\bar{a}$ 或 $\bar{a}a$ 。

最后，三角系统 $\Delta_{M(A)}$ 可以完全编码为一个关于 $(M(A)^-)$ 的单个方程 $U' = V'$，使得可以从解 $U' = V'$ 中获得自由群中初始 $U = 1$ 的解（见 [5, 第 4] 节）。为此，我们使用标记符 #（其中 # $\notin A$ 且 # $= \#$）

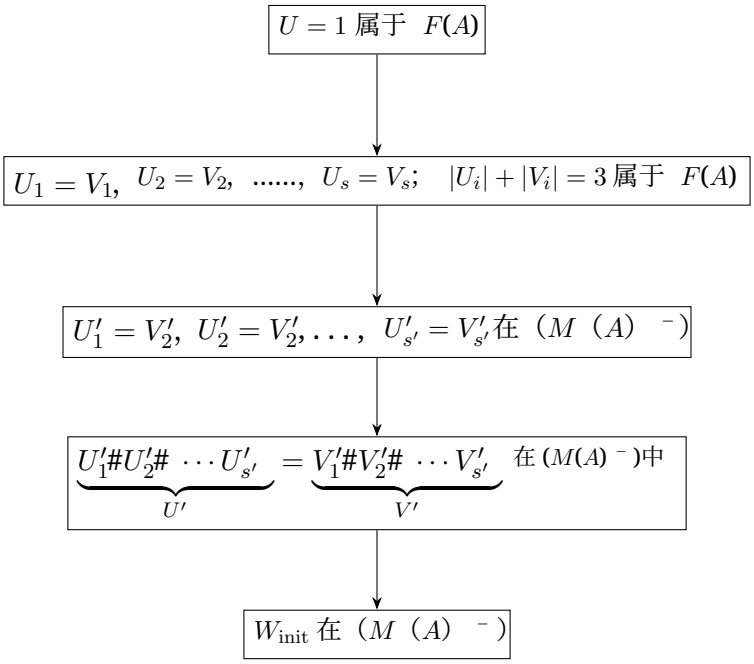

FIGURE 6. 预处理图

以定义一个单词方程 $U' = V'$，其中 $U', V' \in (A \cup \{\#\})^*$，其中

$$U' = U'_1\# \cdots \#U'_{s'} \text{ and } V' = V'_1\# \cdots \#V'_{s'}. \tag{3}$$

加上「$U' = V'$的解中不含 #」这一约束后，$U' = V'$的解与方程组 $U'_j = V'_j$的解一致；再加上不允许出现 $a_i\bar{a}_i$ 和 $\bar{a}_i a_i$（其中 $a_i \in A$）这一约束，便可得到 $U_j = V_j$的解，因此也得到作为约化字的 $U = 1$ 的解。$U' = V'$ 中的变量集包含出现在 $U = 1$中的变量，因此，将解从 $U' = V'$ 转移到 $U = 1$，指的是将解限制在相应的变量集上，也就是仅限于出现在 $U = 1$ 中的变量。

从现在起，我们聚焦于单词方程 $U' = V'$，它属于 $(M(A)^-)$。我们将生成集 A 中的字母 a_i 以及出现在 $U' = V'$中的变量 X_i 添加到 U'和 V'中，并以 # 分隔；将该方程连同这些数据记作一个单词 W_{init}:

$$W_{\text{init}} = \#X_1\# \ldots \#X_\ell\#U'\#V'\#\overline{U'}\#\overline{V'}\#\overline{X_\ell}\# \ldots \#\overline{X_1}\#.$$

简单计算可知，尽管 $|W_{\text{init}}| > |U|$，W_{init}仍与 U的大小呈线性关系；但 W_{init} 的长度将作为输入的大小（有关 W_{init}的更多细节，请参见 [5, 第 3.1] 节）。方程 $U' = V'$ 有解，当且仅当 $W_{\text{init}} = \overline{W_{\text{init}}}$ 有解。通过添加变量，将方程 $U' = V'$ 转化为字 W_{init}，使我们能够同时追踪应用于方程和变量的变换，因此我们不仅从种子字 W_{init}得到一个 EDT0L语言，还能自动追踪整个算法中变量的变化。

主要算法。我们构造一个有限有向图（即一个非确定有限状态自动机）$\mathcal{G}$，其边以幺半群同态为标签：该有限状态自动机将作为施加于种子字 W_{init} 的有理控制。这些同态定义在带对合的自由幺半群 $M(C, {}^-)$ 上，其中 C 是一个包含 A 的扩展字母表；这使我们能够引入

新字母（并在需要时将其移除），以便进行替换和压缩。 C 的大小由 $|A|$ 限定。

求解。构造好图 $\mathcal{G}$ 后，关键（而且起初有些违背直觉）的一点是：为了生成解，我们的算法会从终态向初始状态反向沿路径行进，并按照弧上标记的变换从最后一个到第一个依次应用，从而生成解。因此，用于求解的每条弧的方向，都与构造图时标记该弧的同态方向相反。

$\mathcal{G}$的顶点。图中的每个顶点（状态）大致包含以下数据：(i) W_{init}的一个变体，即一个长度有界的字 W （具体而言为 $|W| \leq 40|W_{\text{init}}|$），通过对 W_{init}应用变换得到；(ii) 关于当前字母表的信息（该字母表是 C的一个动态子集 B，也就是说， B 取决于具体状态），以及 W中出现的变量；(iii) 对 W 中变量施加的有理约束（这些约束通过到有限幺半群的同态实现）；(iv) 变量和字母的“类型”，用于帮助压缩 W 中字母的高次幂（参见 [5, 定义 6.1、6.2]）。自动机 $\mathcal{G}$ 的初始状态对应于 W_{init}，并包含有理约束的初始化；接受状态则对应于从 W_{init} 得到的任意有界长度的字 W，使得 $W = \overline{W}$且 W 不包含任何变量（此外还有一些用于记账的技术性要求，见第 6.2节 [5,]）。

一个关键事实是，图 $\mathcal{G}$ 是有限的：状态中字符串 W 的长度受输入 W_{init}长度的限制，因此状态的数量有限。

$\mathcal{G}$的边。就状态之间的转移（或边）而言，主要有两类：替换弧和压缩弧。这两类边都由 $M(C, ^{-})$ 的自同态标记。主要的替换弧类型最为直观，即 $X \to aX$ 或 $X \to Xa$，也就是说，变量 X 被替换为 aX 或 Xa，其中 $a \in C$。在某些有限的情形下，会使用形式为 $X \to 1$ （移除变量 X）、$X \to XX'$（拆分变量 X）的替换，或者通过允许变量与字母 a 交换来为压缩做“准备”（参见 [5, 第 3.3] 节中“类型”的含义）。压缩弧会转换常量，但不影响变量。主要类型是成对压缩弧和块压缩弧；此外，在不丢失信息时，也允许进行字母重命名和字母表缩减。成对压缩弧由形式为 $h(c) = u$ 的映射标记，其中 $|u| = 2$，因此会将某个字母映射为两个不同的字母；块压缩弧的形式为 $h(c) = c^2$ 或 $h(c) = ac$，其中 c 是 a的幂。

简而言之，考虑到解是沿着上述弧的反向遍历而得到的，我们会遇到以下变换：

– 弹出操作变量： $X \to aX$、 $X \to Xa$或 $X \to 1$ （最后一种仅当约束允许时）– 压缩常量对 $ab \to c$，其中 c 是一个新常量。– 压缩字母块 $aa \ldots a \to a_\ell$，其中 a_ℓ是一个新常量。– 最后，代入所有变量 $X \to 1$、 $Y \to 1$ 等，使两个仅由常量组成的字符串保留下来。如果这两个字符串相同，我们就接受新状态及其变换；否则就将其舍弃。

请注意，第一步（弹出操作）会增加方程的长度，但会使我们更接近解。两种压缩操作会在扩大常量集合的同时缩短方程。不过，我们会在每个状态对方程和当前字母表的大小施加界限，因此长度或字母表大小的增长是受控的。

自动机 $\mathcal{G}$ 以非确定性方式构造：通过猜测每个顶点上的约束以及边的标签类型，生成满足可允许顶点定义的顶点（ W 的大小有界、约束适当等），并检查给定一对顶点之间适用哪些类型的弧。尽管 $\mathcal{G}$ 相对于 W_{init}的规模呈指数级增长，但算法的工作空间中只存储顶点对及其间的弧，因此该算法的空间复杂度较低，即非确定性空间复杂度($n\log\ n$)（见[5, 第 3.8] 节）。此外，还会对自动机进行非常仔细的裁剪。

例 4.1。求解方程 $aXXba = YYaX$ over $A = \{a, b\}$ 的简化算法如下：

– 首先猜测变量的首字母和末字母，然后进行弹出操作/代入：

$$\begin{array}{ll} X \to Xa & aXaXaba = YYaXa \\ X \to bX & abXabXaba = YYabXa \\ Y \to aY & abXabXaba = aYaYabXa \\ Y \to Yb & abXabXaba = aYbaYbabXa \end{array}$$

– 代入 $X \to 1$ 和 $Y \to 1$，使得只含常量的两个字符串留下来。如果这两个字符串相同，就将新顶点设为接受状态；否则丢弃该顶点。

回想一下，该算法处理的是 W_{init}，其形式为

$$\#X\#Y\#aXXba\#YYaX\#\overline{aXXba}\#\overline{YYaX}\#\overline{Y}\#\overline{X}\#.$$

假设我们将第一次弹出操作记为 τ_1: $X \to Xa$，第二次记为 τ_2: $X \to bX$，第三次记为 τ_3: $Y \to aY$，第四次记为 τ_4: $Y \to Yb$。这些弹出操作将 X 转换为 bXa，并将 Y 转换为 aYb，因此当前方程以 $\#bXa\#aYb\ldots$ 开头。然后，如果我们应用 τ_5: $X \to 1, Y \to 1$，当前字符串就变为 $\#ba\#ab\#abababa\#abababa\#$ ...，并且我们在中间得到了相同的字符串（不涉及任何变量）。最后，我们进行压缩 f: $ba \to c_1, ab \to c_2$。令$C = A \cup \{c_1, c_2\}$.

现在将所有方向与 τ_i、 $1 \le i \le 5$相反的映射记为 h_i，注意到它们在 C^*上是恒等变换，并且 f 的逆是 h_6: $c_1 \to ba$、 $c_2 \to ab$。对种子 $\#c_1\#c_2\#\ldots$ 应用映射 h_1、...、h_6，即可得到 X、 Y 的解，也就是 $(c_1)h_6 \ldots h_1$、$(c_2)h_6 \ldots h_1$，并得到 $X = ba$、 $Y = ab$ 满足初始方程。

构造出图 $\mathcal{G}$ 后，难点在于证明所有解都能由它得到，而且这一过程不会引入错误的解。我们使用概率论证来说明，对方程和字母表大小所施加的界限并不构成限制，并且确实能由 $\mathcal{G}$ 得到所有解（见 [5, 引理 3.34]）。

4.2. 无挠双曲群算法概述。设 G 是一个无挠双曲群，其有限生成集 S 对逆封闭；设 $F(S)$ 是以 S 为基的自由群，且 π 是从 $F(S)$ 到 G 的自然投影。给定常量 λ、 $\mu \ge 0$，若一个字是 S 上的字，并且在 G 中表示一个 (λ, μ)-拟测地线，则称其为一个 (G, S, λ, μ)-拟测地线。

输入一个长度为 n、变量集为 $\mathcal{X} = \{X_1, \ldots, X_k\}$ 的方程。与自由群的情形一样，我们首先将输入方程转化为由 G 中方程组成的三角系统 Φ，也就是说，该方程组中的每个方程长度均为 3。具体做法是引入新变量，并由此生成 $\mathcal{O}(n)$ 个方程（详见第 4 节 4.1 和 [5, 第 4] 节）。

从现在起，假设方程组 Φ 由 $q \in \mathcal{O}(n)$ 个形如 $X_jY_j = Z_j$ 的方程组成，其中 $1 \le j \le q$。在双曲群中，无法像此前从自由群到带对合的自由幺半群那样，将三角方程直接化为无消去方程组。对于方程$XY = Z$，不再是在 G 中寻找测地解（g_1, g_2, g_3），即寻找由 G 的凯莱图中的测地三角形表示的解（如图7左侧所示）；而是通过其典范代表 $\theta(g_i)$ 来寻找解 g_i。无挠双曲群中元素的典范代表由 Rips 和 Sela 在 [33], 中定义。其构造详见 [33]，基本介绍则见 [9, 附录 C]。

设 $g \in G$。对于一个固定整数 $T \ge 1$，称其为判据，关于 T的 g的典范代表 $(g)\theta_T$是 S上的一个 (G, S, λ, μ)-拟测地词，并满足 $(g)\theta_T =_G g$、$(g\theta_T)^{-1} = (g^{-1})\theta_T$。如果 T选取得当，一些组合稳定性使得关于 T 的典范代表特别适合用于求解双曲群中的三角方程。我们的主要结果中的语言刻画要求，关于 T的典范代表是 (G, S, λ, μ)-拟测地线，其中 λ和 μ只取决于 δ，而不取决于 T。因此，我们写作 $(g)\theta$，而不是 $(g)\theta_T$。

根据 [33]，存在 $y_i, c_i \in F(S)$，使得方程 $(g_1)\theta = y_1c_1y_2$、$(g_2)\theta = y_2^{-1}c_2y_3$、$(g_3)\theta = y_3^{-1}c_3y_1^{-1}$都是无消去方程（$y_1$ 与 c_1、c_1 与 y_2 等之间不会发生消去），且定义在 $F(S)$上。此外，应将 y_i视为重合的“长”前缀和后缀，而将$c_1c_2c_3$ 视为一个“较小”的内圆，其周长为 $\mathcal{O}(n)$，如图7所示。

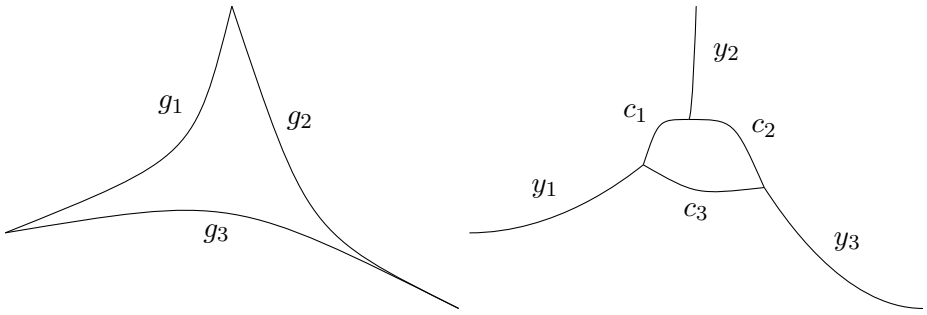

FIGURE 7. 双曲群中 $XY = Z$ 的解。

将 Φ 提升到自由群 $F(S)$中。令 b 为一个常数，它取决于 δ，并且如 [11]所示，与 q 呈线性关系。我们按字典序遍历所有可能的字元组

$$\mathbf{c} = (c_{11}, c_{12}, c_{13}, \ldots, c_{q1}, c_{q2}, c_{q3})$$

满足 $c_{ji} \in S^*$ 和 $|c_{ji}|_S \le b$。

然后，对每个元组 $\mathbf{c}$，我们使用 Dehn算法检查 $c_{j1}c_{j2}c_{j3} =_G 1$是否成立；如果对所有 j 都成立，我们就构造一个方程组 $\Phi_{\mathbf{c}}$，其中的方程形式如下

$$X_j = P_jc_{j1}Q_j,\ Y_j = Q_j^{-1}c_{j2}R_j,\ Z_j = P_jc_{j3}R_j,\quad 1 \le j \le q, \tag{4}$$

其中 P_i、Q_i、Z_i 是新变量。这个新方程组的大小为 $\mathcal{O}(n^2)$。为了避免指数级的复杂度，我们每次只写出一个方程组 $\Phi_{\mathbf{c}}$，因此这一步所需的空间为$\mathcal{O}(n^2)$。令 $\mathcal{Y} \supset \mathcal{X}$、$|\mathcal{Y}| = m$为新的变量集，其中包括所有 P_i、Q_i、Z_i。

注意，$\Phi_{\mathbf{c}}$ 在自由群 $F(S)$ 中的任意解，显然也是原双曲群 G 中 Φ 的解，只需限制到原有变量 $\mathcal{X}$ 即可。也就是说，如果 $(w_1, \ldots, w_m) \subseteq F^m(S)$

是 $\Phi_{\mathbf{c}}$ 的解，那么 $((w_1)\pi, \ldots, (w_k)\pi)$ 就是 Φ 在 G 中的解。这证明了该算法的可靠性。

现在令 $\lambda_G \geq 1$、$\mu_G \geq 0$ 为 [11, 命题 3.4] 给出的常量。

在 $F(S)$ 中求解带约束的方程。现在，我们运行 [5] 中用于求解带有有理约束的自由群方程的算法（我们称之为 tCDE 算法）。该算法的输入 $\Phi_{\mathbf{c}}$ 大小为 $\mathcal{O}(n^2)$，此外，对于每个 $Y \in \mathcal{Y}$，还输入一条有理约束，规定 Y 的解是一个 (G, S, λ_G, μ_G)-拟测地线。由于这些约束的大小为常量（仅取决于 G，而不取决于方程组 Φ），因此它们不会增加 CDE 算法输入的 $\mathcal{O}(n^2)$ 大小。

我们修改 CDE 算法，确保算法打印的每个节点都包含额外的标签 c。（这确保我们为每个方程组 $\Phi_{\mathbf{c}}$ 打印的有限状态自动机各不相同。）由于 c 的大小属于 $\mathcal{O}(n^2)$，这不会影响复杂度。令 $\mathcal{C}$ 为所有元组 c 组成的集合。我们运行上述修改后的 CDE 算法（为打印出的每个状态添加标签 c），为每个 $\Phi_{\mathbf{c}}$ 打印一个有限状态自动机（可能为空）。该有限状态自动机是一个 EDT0L 文法的有理控制；该文法生成 $F(S)$ 中所有作为自由约化字的解，这些解对应于双曲群中同一方程组 $\Phi_{\mathbf{c}}$ 的 (G, S, λ_G, μ_G)-拟测地线解。

注意，若 $(g_1, \ldots, g_k) \in G^k$ 是原双曲群中的一个解，则 [33] 保证存在典范代表 w_i，它们是 (G, S, λ_G, μ_G)-拟测地线，满足 $w_i =_G g_i$；对于 $1 \leq i \leq m$，它们的自由约简形式为 $u_i =_G w_i$，而我们的构造保证能够捕获任意这样的字符串集合。更具体地说，若 $(w_1, \ldots, w_r)$ 是 Φ 在典范代表中的一个解，那么 $(u_1, \ldots, u_k, \cdots u_{|\mathcal{Y}|})$ 将包含在 CDE 算法生成的 $\Phi_{\mathbf{c}}$ 的解中，其中 w_i 对于 $1 \leq i \leq m$的约简形式为 u_i。这表明，将所有方程组 $\Phi_{\mathbf{c}}$ 对应的文法取并集后，完备性即可得到保证。

为了生成 G中的解，我们添加一个新的初始状态，并用标签为 c 的边连接到各个有限状态自动机的初始状态，其中 $\mathbf{c} \in \mathcal{C}$。由此得到一个自动机，它是输入方程的解语言 L 的有理控制。该有理控制使 L 成为由 (G, S, λ, μ)-拟测地线构成的 EDT0L语言，其中 $\lambda := \lambda_G, \mu := \mu_G$。所需空间恰好等于自由群上 CDE 算法在输入 $\mathcal{O}(n^2)$ 时所需的空间，即非确定性空间复杂度为 $n^2 \log n$。

还需要进一步的步骤，才能用短字典序代表元表示这些解（见 [7,定理 7.4(2)]），但这些步骤不会影响语言或空间复杂度。

5. Virtually abelian groups

几乎阿贝尔群这一类群在本文讨论的群中十分特殊，因为我们可以对解集给出更强的结论。这是由于几乎阿贝尔群与自由阿贝尔群十分接近，因而可以运用线性代数方法；而且解的结构有若干种等价的代数、几何和组合描述。

更具体地说，我们将几乎阿贝尔群中的解集描述为 m-正则语言，这比将其描述为 EDT0L 严格更强。m-正则语言的定义与第 2.3 节中介绍的正则语言定义相似。设 A 为一个集合。我们将 A^n 记作 $A \times \cdots \times A$ 个 n 的笛卡尔积，并将 ε 记作空词。

定义5.1。（异步） n变量有限状态自动机（n-有限状态自动机） $\mathcal{A}$ 是一个四元组（Σ，Γ，s_0，F），其中

(1) Σ 是一个有限字母表；(2) Γ 是一个有限有向图，其边的标签取自（$\Sigma \cup \{\varepsilon\}$），^{n} 且至多有一个非$\varepsilon$ 项；(3) $s_0 \in V$（Γ）是指定的起始顶点／状态；(4) $F \subseteq V$（Γ）是接受顶点／终结顶点的集合。

如果存在一条从 s_0 到 $s \in F$ 的有向路径，使得沿路径读取标签并删除每个坐标中所有 ε 的出现后得到w，那么 n元组w $\in(\Sigma^*)^n$ 就被 $\mathcal{A}$ 接受。由 n有限状态自动机接受的语言，简称为 n正则语言。

注5.2。

(1) 注意，定义5.1给出的是一个非确定有限状态自动机，因为它允许 (i) 标签为完全由 ϵ坐标组成的元组的边，以及 (ii) 从同一个顶点出发、标签相同的两条边。若不允许 (i) 和 (ii)，则得到一个确定有限状态自动机 n。(2) 当 $n = 1$时，定义5.1与正则语言的通常定义一致。然而，对于 $n > 1$，确定与非确定 n有限状态自动机所接受的语言类并不相同。例子见 [23, 练习 2.10。3]

We 下面汇总几乎阿贝尔群中解集的刻画。

Th理定理 5.3。 设 G 是一个有限生成的几乎阿贝尔群。那么其上的一个解集 G

(1) 是 G^n 的一个有理子集，其中某个 $n \geq 1$ [10, 由定理 1.1 (2)] 给出；(2) 可描述为由测地代表元组成的集合，该集合是某个 $m \geq 1$ 下的 m-正则语言，因此也是 EDT0L 语言，其中 G [10, 的任意生成集都适用（定理 1.3 (4)]）；(3) 依照生成元集 $X = \Sigma \cup T$ 上的自然标准形来描述（其中 Σ 是 G 的有限指数自由阿贝尔子群的对称生成元集，T 是该子群的有限陪集代表集），可描述为 m-正则语言，因此也是 EDT0L 语言。该 EDT0L 语言的 EDT0L 系统可在非确定性空间复杂度 NSPACE(n^2) 下构造（ [19] 的推论3.12 和命题3.15，或 [10] 的定理 1.4 (4)）。

据我们所知 ，几乎阿贝尔群是唯一一类解集为有理的群 l.

以下命题与引理 2.20(1)类似， [19], 为 m-正则语言与 EDT0L 语言之间搭起了桥梁，这一桥梁出现在定理 5.3(2)和(3)的陈述中。

命题 5.4. 设 $L \subset (\Sigma^*)^n$ 是一个 n-正则语言。若 $\theta_\#$: $(\Sigma^*)^n \to (\Sigma \cup \{\#\})^*$ 是映射 $(w_1, w_2, \ldots, w_n) \mapsto w_1\#^{\epsilon_1} w_2\#^{\epsilon_2} \cdots \#^{\epsilon_n} w_n$，其中 $\epsilon_i \in \{0, 1\}$ 对所有 i成立，该映射会在某些元组坐标之间插入一个 #，则 $(L)\theta_\# \subset (\Sigma \cup \{\#\})^*$ 是 EDT0L。

注记 5.5。考虑情形 $n = 1$ 可知，命题 5.4 的逆命题不成立（因为存在非正则的 EDT0L语言），因此，n-正则性严格强于 EDT0L 性质。

6. The Heisenberg group and Diophantine equations over the ring of integers

定义 6.1. 海森堡群 $H(\mathbb{Z})$ 是由以下表示定义的二类幂零群

$$H(\mathbb{Z}) = \langle a,\ b,\ c \mid c = [a,\ b],\ [a,\ c] = [b,\ c] = 1 \rangle.$$

生成元 c 是多余的，不过我们仍将其列入，因为使用生成集 $\{a,\ b,\ c\}$ 通常比使用 $\{a,\ b\}$更方便。

海森堡群的 Mal'cev生成集为集合 $\{a,\ b,\ c\}$。

定义 6.2。海森堡群的 Mal'cev正规形是将元素 $g \in H(\mathbb{Z})$ 映射为唯一一个形如 $a^i b^j c^k$ 的字的正规形，其中 $i,\ j,\ k \in \mathbb{Z}$，该字表示 g。

众所周知，这是 $H(\mathbb{Z})$ 的一个（唯一）正规形（例如参见 $[17]$）。

虽然无法判定海森堡群中的一个方程组是否有解[17], ，但存在一种算法可以判定单个方程是否有解 $[17]$。即使在一般的二类幂零群中，单个方程的可满足性也是不可判定的 $[34]$, ，因此海森堡群中的单个方程处于可判定性的"边界"附近。海森堡群中单个方程的解语言是否属于 EDT0L 仍是一个开放问题。不过，在单变量情形下，答案是肯定的：

定理6.3（$[25]$）。设 L 是海森堡群中关于 Mal'cev生成集和正规形的单个单变量方程的解语言。那么

(1) 语言 L是 EDT0L；(2) L 的 EDT0L系统可在非确定性空间复杂度 NSPACE($n^8(\log n)^2$) 内构造。

定理6.3的证明首先将问题归约为整数环中的二元二次方程。证明的大部分内容随后用于说明这些方程具有 EDT0L 解语言。

定理6.4。二元二次方程 $\mathcal{E}$的解语言：

$$\alpha X^2 + \beta XY + \gamma Y^2 + \delta X + \epsilon Y + \zeta = 0$$

在整数中是 EDT0L，可由一个 EDT0L系统接受；该系统可在非确定性空间复杂度 NSPACE($n^4 \log n$) 内构造，其中输入规模取为 max($|\alpha|, |\beta|, |\gamma|, |\delta|, |\epsilon|, |\zeta|$)。

定理6.4的证明采用拉格朗日的方法，将任意二元二次方程归约为佩尔方程。佩尔方程是整数环中的一种方程，形式如下：

$$X^2 - DY^2 = 1,$$

其中 X 和 Y 是变量，且 $D > 0$ 不是完全平方数。我们要求 $D > 0$，因为如果 $D \leq 0$，它只有有限多个解；我们还要求 D 不是完全平方数，因为在这种情况下，我们可以将方程因式分解，从而证明它同样只有有限多个解。为此，必须先将其化为广义佩尔方程，其形式为

$$X^2 - DY^2 = N,$$

其中 $N \in \mathbb{Z}$。拉格朗日方法表明，整数范围内任意二元二次方程 $\mathcal{E}$ 的解集等于

$$\left\{ \left(\frac{\alpha x + \beta y + \gamma}{\delta},\ \frac{x - \lambda}{\mu} \right) \;\middle|\; (x,\ y) \text{ is a solution to } X^2 - DY^2 = N \right\},$$

其中 $N,\ D \in \mathbb{Z}$ 可由 $\mathcal{E}$计算得出。证明佩尔方程的解属于 EDT0L语言相当直接，我们在下文给出证明。此证明可轻易修改以适用于广义佩尔方程。更难的是证明这些解的线性组合（即形如 $\alpha x + \beta y + \gamma$、其中 $\alpha,\ \beta,\ \gamma \in \mathbb{Z}$ 的整数）构成一种 EDT0L语言。归约中的"除法"部分

通过构造 EDT0L系统来完成，不需要数论方法。

下面给出证明：任意佩尔方程的解构成一种 EDT0L语言。

引理6.5. 设 $X^2 - DY^2 = 1$ 是佩尔方程的一个实例，其中 $D \in \mathbb{Z}_{>0}$ 不是完全平方数。那么

$$\{a^x \# a^y \mid x^2 - Dy^2 = 1\}$$

是字母表 $\{a,\ a^{-1}, \#\}$上的一种 EDT0L语言。

证明 佩尔方程 $X^2 - DY^2 = 1$ 的解具有如下形式

$$\{(x_n,\ y_n) \mid n \in \mathbb{Z}_{\geq 0}\},$$

其中 $(x_0,\ y_0) = (1, 0)$，$(x_1,\ y_1)$ 是除 $(1, 0)$ 外的最小解，而对于 $n > 1$，$(x_n,\ y_n)$ 由下式递归定义

$$x_n = x_1 x_{n-1} + D y_1 y_{n-1}, \quad y_n = y_1 x_{n-1} + x_1 y_{n-1}.$$

这一事实是众所周知的（例如参见 [1] 第3章）。已有大量研究采用不同方法计算 $(x_1,\ y_1)$，效率各不相同；然而，为了证明该语言是 EDT0L，我们无需进行这项计算。若要构造 EDT0L系统，则需要进行计算；而要证明构造 EDT0L系统所需的空间复杂度良好，也需要采用这些方法之一。首先注意，只需证明非负解的集合是 EDT0L。为此，我们将证明

$$L = \{a^x \# b^y \mid x,\ y \in \mathbb{Z}_{\geq 0},\ x^2 - Dy^2 = 1\}$$

是 EDT0L。然后注意，$(x,\ y)$ 是一个解，当且仅当 $(x,\ -y)$ 是一个解；当且仅当 $(-x,\ y)$ 是一个解；当且仅当 $(-x,\ -y)$ 是一个解。因此，利用自由幺半群同态将 $(a,\ b)$ 映射为 $(a,\ a)$、$(a^{-1},\ a)$、$(a,\ a^{-1})$ 或 $(a^{-1},\ a^{-1})$ 中任意一个，并应用定理 3.4，我们可以证明位于任意象限内的解的集合是 EDT0L。接着，我们可以应用定理 3.4 证明这些语言的并集是 EDT0L。由于这就是所有解的集合，证明至此完成。

现在定义我们的 EDT0L系统 L。我们的扩展字母表为 $C = \{a,\ b, \#\ \ a_x,\ a_y,\ b_x,\ b_y\}$，起始词为 $a_x \# b_x$。定义 $\varphi \in$End(C^*) 如下：

$$a_x\varphi = a_x^{x_1} a_y^{Dy_1} \qquad b_x\varphi = b_x^{x_1} b_y^{Dy_1}$$
$$a_y\varphi = a_x^{y_1} a_y^{x_1} \qquad b_y\varphi = b_x^{y_1} b_y^{x_1}.$$

然后定义 $\theta \in$End(C^*) 如下：

$$a_x\varphi = a \qquad b_x\varphi = \varepsilon$$
$$a_y\varphi = \varepsilon \qquad b_y\varphi = b.$$

我们的有理控制是 $\varphi^*\theta$（见图 8）。根据构造，如果 $n \in \mathbb{Z}_{\geq 0}$，那么 $(a_x \# b_x)\varphi^n$ 将具有 $u \# v$ 的形式，其中 u 恰好包含 x_n 个 a_x，以及 y_n 个 a_y；而 v 恰好包含 x_n 个 b_x，以及 y_n 个 b_y。因此 $(a_x \# b_x)\varphi^n\theta = a^{x_n} \# b^{y_n}$，由此我们证明了我们的 EDT0L系统接受 L。

□

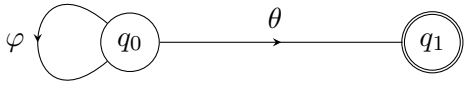

FIGURE 8. 对于 $L = \{a^x \# b^y \mid (x,\ y) \in \mathbb{Z}^2_{\geq 0}$，有理控制是 $X^2 - DY^2 = 1\}$ 的一个解，初始状态为 q_0，接受状态为 q_1。

7. CLOSURE PROPERTIES

我们总结了能够保持方程组的解语言具有 EDT0L 性质的群论构造。许多其他构造是否也能保持这一性质，仍是一个开放问题，例如将群扩张为有限指标超群。

定理 7.1（[26], 定理 A）。设 G 和 H 为群，其方程组的解语言分别关于正规形 η_G 和 η_H 属于 EDT0L，并且这些 EDT0L 系统可在非确定性空间复杂度 NSPACE(f) 内构造，其中 f。那么，在以下群中，方程组的解属于 EDT0L，且可在非确定性f空间内构造 EDT0L 系统：

(1) $G \wr F$，其中 F 为任意有限群；
(2) $G \times H$；(3) G 的任意有限指数子群；

在以下群中，方程组的解属于 EDT0L，且可在非确定性空间复杂度 NSPACE($n^4 \log n$) 内构造 EDT0L 系统：

(4) 任意几乎为双曲群直积的群；(5) 二面体 Artin 群。

如果 η_G 和 η_H 都是拟测地的或正则的，那么 (1)、(2) 和 (3) 中使用的正规形也具有相同性质。对于几乎为双曲群直积的群 (4) 以及 (5) 中的二面体 Artin 群，可以选取正则且拟测地的正规形。

证明 (1) 和 (2) 依赖于一个关于 EDT0L语言的技术引理：其中每个字中都含有相同数量的字母 #。该引理表明，可以用一个性质良好的 EDT0L系统生成这类 EDT0L系统；这个系统允许将两个这样的语言“并行连接”，而这正是构造直积所需要的。我们必须先正式定义这类 EDT0L系统必须具备的性质，以便允许并行连接。

定义 7.2. 设 Σ 是一个字母表，且 #₁、……、$\#_n \in \Sigma$。一个 (#₁、……、$\#_n$)-分离 EDT0L 系统是一个 EDT0L 系统 $\mathcal{H}$，其起始词具有 $\omega_0 \#_1 \omega_1 \#_2 \cdots \#_n \omega_n$ 的形式，其中 $\omega_i \in$(Σ\{#₁、……、$\#_n$})* 对所有 i 都成立，并且对所有 i 以及有理控制中的每个 ϕ，$\#_i \phi^{-1} = \{\#_i\}$。

引理 7.3（[26], 引理 3.4）。设 L 是一个 EDT0L语言，且其中每个字都恰好包含 n 个字母 #，其中 $n \in \mathbb{Z}_{\geq 0}$。令 $f\colon\ \mathbb{Z}_{\geq 0} \to \mathbb{Z}_{\geq 0}$。那么

(1) 存在一个接受 L的（$\#\ldots,\#$）-分离 EDT0L 系统 $\mathcal{H}$。(2) 如果能在非确定性空间复杂度 NSPACE(f) 内构造出 L 的 EDT0L 系统，那么就能在非确定性空间复杂度 NSPACE(f)内构造出 $\mathcal{H}$ 。

利用引理 7.3，可以直接证明如下结果。该结果表明，直积保持方程组具有 EDT0L 解语言这一性质。有限群的广 wreath 积则需要稍多一些工作；详情见 [26] 。

引理 7.4（[26],引理3.5）。设 L和 M是 EDT0L 语言，且 $L\cup M$中的每个字都恰好包含 n 个字母 #。设 $f\colon\ \mathbb{Z}_{\geq 0}\to\mathbb{Z}_{\geq 0}$。那么

(1) 语言

$$N=\{u_0v_0\#\cdots\#u_nv_n\mid u_0\#\cdots\#u_n\in L,\ v_0\#\cdots\#v_n\in M\},$$

是 EDT0L；(2) 如果可以在非确定性空间复杂度 NSPACE(f)内构造 L 和 M 的 EDT0L 系统，那么可以在非确定性空间复杂度 NSPACE(f)内构造 N 的 EDT0L 系统。

在讨论有限指数子群（有限生成群的）也保持方程组具有 EDT0L 解语言这一性质的证明之前，我们必须先定义生成集和正规形。根据 Schreier 引理，我们知道可以显式构造一个（有限）生成集，而该生成集会导出一个正规形。下面给出施莱尔生成元的定义，但证明请参见 [26] 。

定义 7.5。设 G 是一个群，由有限集 Σ生成；设 H 是 G的有限指数子群；设 T 是 H 在 G中的右横截。对于每个 $g\in G$，令 $\bar{g}$ 为 T 中位于陪集 Hg内的唯一元素。关于 T 和 Σ的 H的施莱尔生成集定义为

$$Z=\{tx\overline{tx}^{-1}\mid t\in T,\ x\in\Sigma\}.$$

给定一个群及其有限生成集的正规形，Schreier 引理的证明还利用原群的正规形，为有限指数子群构造了一个显式的正规形。下面给出该正规形。转而使用此正规形会保留正则性和拟测地性。

定义 7.6。设 G 是一个群，由有限集 Σ生成；设 H 是 G的有限指数子群；设 T 是 H 在 G中的右横截。令 Z 为 H的施莱尔生成集。固定关于（G，Σ）的正规形 η。

我们将相对于 η的 (H, Z) 的施莱尔正规形定义为 ζ，如下所示。令 $h\in H$，并假设 $h\eta=a_1\cdots a_n$，其中 $a_1,\ldots,a_n\in\Sigma^{\pm}$。令 t_0 为 $T\cap H$ 中唯一的元素，并定义$t_i=\overline{a_1\cdots a_i}$。将 $h\zeta$ 定义为

$$h\zeta=(t_0a_1\overline{t_0a_1}^{-1})(t_1a_2\overline{t_1a_2}^{-1})\cdots(t_{n-1}a_n\overline{t_{n-1}a_n}^{-1}). \tag{5}$$

这确实定义了 H 的一个元素，且等于 h，这一事实是 Schreier 引理证明的一部分。

以下证明依赖于用 G 的生成元和一个固定右横截中的元素为扩展字母表中的字母编索引，以便追踪我们的元素所在的陪集，确保我们始终处于正规形之内。

命题 7.7（[26], 命题 6.3）。设 $f\colon\ \mathbb{Z}_{\geq 0} \to \mathbb{Z}_{\geq 0}$。设 G 是一个群，其中相对于正规形 η，方程组的解属于 NSPACE(f) 中的 EDT0L 类。那么，对于 G 的任意有限指数子群，相对于从 η 继承的施莱尔正规形，同样的结论也成立。

8. Baumslag-Solitar groups

Duncan、Evetts、Rees 和 Holt [18] 证明，按照标准正规形，若干可解鲍姆斯拉格-索利塔尔群中的方程具有 EDT0L语言，包括乘法“三元组”和固定元素的中心化子。鲍姆斯拉格-索利塔尔群是由以下表示定义的群

$$\mathrm{BS}(j,\ k) = \langle a,\ b \mid b^{-1}a^j b = a^k \rangle,$$

其中 j、 $k \in \mathbb{Z}\backslash\{0\}$满足某些条件。当且仅当 $j = 1$ 或 $k = 1$时，这些群是可解的，因此我们将（不失一般性地）假设 $j = 1$。如果 $k = \pm 1$，那么该群几乎阿贝尔，因此这里无需考虑。此外，[18] 中的作者假设 $k > 1$，不过他们猜想，当 $k < -1$时，除了某些技术细节外，证明仍然成立。

现在我们定义一种正规形，用来证明某些方程的解语言是 EDT0L语言。

引理 8.1（[18],第 3 节）。BS(1, k) 中的每个元素都可以唯一地表示为

$$b^t \alpha^{i_m} b^{-1} \alpha^{i_m - 1} b^{-1} \cdots \alpha^{i_1} b^{-1} \alpha^s,$$

其中 $t \in \mathbb{Z}$, m, $s \in \mathbb{Z}_{\geq 0}$, $i_j \in \{0,\ \ldots,\ k-1\}$ 对所有 j都成立，且当 $m > 0$时， $i_m \neq 0$。

定理 8.2（[18]）。设 $G = B(1,\ k)$，其中 k。那么，相对于引理 8.1 中的正规形，

(1) $XY = Z$ 的解语言是 EDT0L；(2) $XYZ = 1$ 的解语言是 EDT0L；(3) 若 $g \in G$，则 $Xg = gX$ 的解语言是 EDT0L。

相对于这种正规形，Duncan、Evetts、Holt 和 Rees 推测，存在解语言不是 EDT0L 的方程。这并不一定意味着对于其他正则正规形，所有解都不是 EDT0L；然而，迄今尚未发现一个具有可判定丢番图问题、但相对于任意“合理”正规形都存在非 EDT0L 解语言的群，因此，利用 EDT0L 语言理解方程的解方面若能取得进展，将意义重大。他们推测，方程

$$XY = YX$$

不是 EDT0L。通过将该解语言与正则语言 $b^*a\#b^*a^*$ 求交，证明该解语言不是 EDT0L，就归结为证明语言

$$\{b^r a \# b^{rn} a^{\frac{k^{rn}-1}{k^r-1}} \mid r,\ n \geq 0\}$$

不是 EDT0L。

Acknowledgements

作者谨此感谢 George Metcalfe 和伯尔尼大学在撰写本综述最后阶段给予的款待与支持。第一作者感谢瑞士国家科学基金会提供的科学交流基金（编号 IZSEZ0 213937），第二作者感谢海尔布隆数学研究所在本研究期间给予的支持。

最后，若没有许多数学家和珍贵合作者对这一领域作出的重要贡献，本综述便无从写成：Volker Diekert、Andrew Duncan、Murray Elder、Alex Evetts、Derek Holt、Sarah Rees 等等。

References

[1] Titu Andreescu 和 Dorin Andrica, *Quadratic Diophantine equations*, 《数学进展》第40卷, 施普林格, 纽约, 2015 3362224 。前言由 Preda Mihailescu 撰写。MR Peter R.J. Asveld, , 备忘录／特温特大学应用数学系, 1976年（英文）。[3] Alex Bishop 和 Murray Elder, *Bounded automata groups are co-ET0L*, 《语言与自动机理论及应用》, 2019年, 第82–94页。MR3927687[4] Tara Brough、Laura Ciobanu、Murray Elder 和 Georg Zetzsche, *Permutations of context-free ET0L and indexed languages* **17** 2016 3, 167 178 3529277 （）, 第–期。MR[5] L. Ciobanu、V. Diekert 和 M. Elder, *Solution sets for equations over free groups are EDT0L languages*, 《国际代数与计算杂志》 **26** 2016 5, 843 886 3536438 （）, 第–期。MR L. Ciobanu 和 M. Elder, [6] *Solutions sets to systems of equations in hyperbolic groups are EDT0L in PSPACE*, 第46届自动机、语言与程序设计国际研讨会, 2019, 110, 15 3984927 第页, 文章编号。MR, 《以色列数学杂志》（）, 第2期, 869 920 4358266 –。MR[8] L. Ciobanu、M. Elder 和 M. Ferov, *Applications of L systems to group theory*, 《国际代数与计算杂志》28（2018）, 第2期, 309–329页。MR3786421[9] Laura Ciobanu 和 Murray Elder, [7] ———— *The complexity of solution sets to equations in hyperbolic groups* **245** 2021 *Solutions sets to systems of equations in hyperbolic groups are edt0l in pspace*, arXiv 预印本（2019）。arXiv:1902.07349。[10] Laura Ciobanu 和 Alex Evetts, *Rational sets in virtually abelian groups: languages and growth*, arXiv 预印本（2022）。arXiv:2205.05621。[11] F. Dahmani, *Existential questions in (relatively) hyperbolic groups*, 《以色列数学杂志》173（2009）, 91–124页。MR2570661[12] Fran cois Dahmani 和 Vincent Guirardel, , , 《拓扑学杂志》（）, 第–期。MR[13] *Foliations for solving equations in groups: free virtually free and hyperbolic groups* **3** 2010 2, 343 404 2651364 *More than 1700 years of word equations* V. Diekert, , 《代数信息学》, 2015, 22 28 3448027 第–页。MR V. Diekert 和 M. Elder, [14] *Solutions to twisted word equations and equations in virtually free groups* 《国际代数与计算杂志》 **30** 2020 4, 731 819 4113851 （）, 第–期。MR V. Diekert、A. Jez 和 M. Kufleitner, [15] *Solutions of word equations over partially commutative structures*, 第43届自动机、语言与程序设计国际研讨会, 2016, 127, 14 3577188 第页, 文章编号。MR Volker Diekert 和 Anca Muscholl, [16] *Solvability of equations in graph groups is decidable* 《国际代数与计算杂志》 **16** 2006 6, 1047 1069 2286422 （）, 第–期。MR Moon Duchin、Hao Liang 和 Michael Shapiro, [17] *Equations in nilpotent groups*, 《美国数学会会刊》143卷（2015 11, 4723 4731 3391031 ）, 第–期。MR Andrew Duncan、Alex Evetts、Derek F. Holt 和 Sarah Rees, [18] *Using EDT0L systems to solve some equations in the solvable Baumslag-Solitar groups* 2023 2204 13758 , arXiv 预印本（）。arXiv:。。[19] Alex Evetts 和 Alex Levine, *Equations in virtually abelian groups: languages and growth*, 《国际代数与计算杂志》 **32** 2022 3, 411 442 4417480 （）, 第–期。MR Julien Ferte、Nathalie Marin 和 Geraud Senizergues, [20] *Word-mappings of level 2*, 《计算机系统理论》54卷（2014 1, 111 148 3149083 ）, 第–期。MR R. I. Grigorchuk 和 P. F. Kurchanov, [21] *Some questions of group theory related to geometry*, 《代数》第七卷, 1993年, 第167–232、233–240页。P. M. Cohn 从俄文翻译。MR1265271

[22] Thomas Herbst 和 Richard M. Thomas, *Group presentations formal languages and characterizations of one-*, 《理论计算机科学》（），第–*counter groups* **112** 1993 2, 187 213 1216320 期。MR[23] *Groups languages and automata* D. F. Holt、S. Rees 和 C. E. Rover，，伦敦数学会学生教材第88卷，剑桥大学出版社，剑桥，2017年。MR3616303[24] Olga Kharlampovich、Laura L´opez 和 Alexei Myasnikov, *The Diophantine problem in some metabelian groups*, 《数学计算》 **89** 2020 325, 2507 2519 4109575（），第–期。[25] *Formal languages* MR Alex Levine, *quadratic Diophantine equations and the Heisenberg group*, arXiv 预印本（2022年）。arXiv:2203.04849。
[26] ——— *EDT0L solutions to equations in group extensions* **619** 2023 860 899 4534696，《代数学杂志》（），–。MR G.
[27] *Systems of equations in free groups* **13** 1972 587 595 S. Makanin, 《西伯利亚数学杂志》（），–。MR 0318314
[28] ——— *The problem of the solvability of equations in a free semigroup* **103 145** ，《数学文集》（新系列）（（ 0470107 1977 2, 147 236, 319），第–期。MR[29] ——— ，
Equations in a free group **46** 1982 6, 1199 1273, 1344 682490，《苏联科学院消息》数学系
[30] *Computational complexity* 1994 列（），第–期。MR Christos H. Papadimitriou，Addison-Wesley 出版公司，马萨诸塞州雷丁，。MR1251285[31] A. A. Razborov, *On systems of equations in free groups*, 博士论文，1971年。俄文。[32]， *On systems of equations in free groups*, 《组合群论与几何群论》（爱丁堡，~~1993年~~），1995, 269 283 1320290 第–页。MR E. Rips 和 Z. Sela,
[33] *Canonical representatives and equations in hyperbolic groups*, 《数学发明》120卷（1995年），第 3, 489 512 1334482 –期。MR
[34] Vitaly A. Roman'kov, *Diophantine questions in the class of finitely generated nilpotent groups*, 《群论杂志》 **19** 2016 3, 497 514 3510840（），第–期。MR G. Rozenberg 和 A. Salomaa, *The mathematical theory of L systems*, 纯粹与应
[35] 用数学第90卷，学术出版社公司 [ ], 1980 561711
[36] *Extension of tabled -systems and languages* Harcourt Brace Jovanovich 出版社，纽约—伦敦，。MR Grzegorz Rozenberg，OL，《国际计算机与信息科学杂志》2（1973年），311–336页。MR0413614

Department of Mathematics, Heriot-Watt University, and the Maxwell Institute for Mathematical Sciences, Edinburgh EH14 4AS, UK

Email address: L.Ciobanu@hw.ac.uk

Department of Mathematics, Alan Turing Building, The University of Manchester, Manchester M139PL, UK

Email address: alex.levine@manchester.ac.uk