申请上海交通大学博士学位论文

向量加法系统模型研究

博 士 研 究 生：杨启哲

学　　号：016037910001

导　　师：傅育熙教授

申 请 学 位：工学博士

学　　科：软件工程

所 在 单 位：电子信息与电子工程学院软件工程系

答 辩 日 期：2022 年 4 月 19 日

授予学位单位：上海交通大学

Dissertation Submitted to Shanghai Jiao Tong University
for the Degree of Doctor

THE STUDY OF VECTOR ADDITION SYSTEM

Candidate:	Qizhe Yang
Student ID:	016037910001
Supervisor:	Prof. Yuxi Fu
Academic Degree Applied for:	Doctor of Engineering
Speciality:	Software Engineering
Affiliation:	Depart of SEIEE
Date of Defence:	April 19, 2022
Degree-Conferring-Institution:	Shanghai Jiao Tong University

向量加法系统模型研究

摘　要

Petri 网，自 1962 年由 Carl Petri 提出以来，已经成为了研究并发系统中最重要的一个理论模型之一。向量加法系统，作为一个与 Petri 网等价的模型，其由于更简单的形式受到了研究者们的青睐，并且随着研究的开展，一些相关的衍生模型也映入研究者的眼帘，这都说明对向量加法系统这一模型的研究是十分有意义且必要的。

在向量加法系统的研究中，可达性问题是其中的核心。所谓可达性问题，即询问两个格局之间是否存在一条路径，该问题由于能建模很多现实中并发程序中的验证问题，从而受到研究者们的关注。与之相关的还有可覆盖性问题和有界性问题，前者询问是否可以到达一个比目标格局更大的格局，而后者则询问所能达到的格局是否是有限的，这些验证问题都因为在实际中有着大量的应用而产生了重要的理论研究意义。

在并发理论中，非确定计算的结构是相当复杂的。即使是在只有 τ 动作的模型 $\mathbb{CCS}^{\mu}$ 中其计算对象的结构也是不平凡的。对其进行定量的分析，比如有限状态的不同的计算对象有多少个是十分有意义的，因为这体现了对其复杂性的刻画，而随着 Fu 提出了对其的抽象表示 C-图，进行这方面的研究有了扎实的基础。

本文首先聚焦于向量加法系统，对其上的验证问题进行研究。我们将在归纳总结这些验证问题的已有工作的基础上，阐述一些新的理解，这些想法对日后的进一步研究会起到指导的作用。其次我们将聚焦于非确定计算，我们将给出第一个从计数方面进行研究的工作，通过计数的角度来理解非确定性的复杂性。本文的主要贡献如下：

- 首先我们系统归纳总结了近 60 年来向量加法系统的验证问题的研究，包括可达性问题，可覆盖性问题和有界性问题。尤其在可达性问题的下界，我们介绍了其中用到的不同技术和方法，也比较了这些方法的差异与不同，相信对其他模型的相关验证问题有很好的指导意义。
- 其次，对于可达性问题中的 KLMST 算法，我们提出了一种全新的理解方式，即通过分解-降维-分解-$\cdots$ 的形式来理解，即我们希望通过联系 $n+1$ 维的向量加法系统和 n 维向量加法系统的可达性问题之间的关系来理解 KLMST 算法，该理解方式被相信将有助于研究固定维度下的向量加法系统的可达

性问题。

- 对于有限状态计算对象在 $\mathbb{CCS}^{\mu}$ 的抽象刻画：C-图，我们引入了高度的概念，该概念能够反映一个进程在计算过程中所要经过的不同状态的个数。并且我们以高度为参数，给出了 C-图个数随高度变化的递推式，从而证明了随着高度的变化，C-图个数的增长速度是非初等的，这一结果可以反映由非确定计算所带来的复杂性。

关键词：向量加法系统，可达性，可覆盖性，有界性，非确定性，C-图

THE STUDY OF VECTOR ADDITION SYSTEM

ABSTRACT

Petri net, proposed by Carl Petri in the year 1962, has been one of the most famous formal models in the analysis of parallel processes. As an equivalent model of Petri net, vector addition system (VAS) or vector addition system with states (VASS) has been heavily investigated in theoretical setting.

Given a VASS and an initial configuration, the reachability problem asks whether there exists a sequence of valid execution steps that reaches a given final configuration. Reachability problem capture various verification issues in concurrent program, and many problems can reduce to it. In many situations it is the central issue. There are other verification problems, such as the coverability problem and boundedness problem. The former asks whether a configuration can reach a configuration above a target configuration, and the latter asks whether the set of reachable configurations is finite. All of them have a wide range of applications, and have been studied by researchers in the past 60 years.

In concurrency theory, the structure of nondeterministic computations is extremely complicated. Even in the variant $\mathbb{CCS}^{\mu}$ of CCS that admits only τ actions, these objects already demonstrate nice and rich structures that we understood very little before. Research on the quantitative aspect of process such as asking how many unequal n-state processes is meaningful since it helps to see the complexity of nondeterminism. Fu proposed an abstract representation of the computational objects called C-graphs which let it possible to do some research from a quantitative aspect.

This thesis first focuses on the vector addition system. Based on the

discussion of the existing work, we propose some new understanding of the reachability problem, which will help future research. Next, we focus on the nondeterministic computation, we will carry out quantitative study of the complexity of nondeterminism. AS far as we know, this is the first of it kind. The main contributions of the thesis are the follows:

- First we summarize the existing work of VAS in the recent 60 years including reachability problem, coverability problem and boundedness problem. Especially for the lower bound of reachability problem, we compare with different techniques, which will help some related verification problems in other models.
- Next for the famous KLMST algorithm, we offer a new understanding of the algorithm, i.e. by the form of decomposition-dimension reduction-decomposition-$\cdots$. In this way we try to build a relationship between the reachability problem for $(n+1)$-dimensional VAS and the reachability problem for n-dimensional VAS, which we believe will contributes to efforts in the reachability problem for fixed dimension
- Finally for the abstract presentation of finite state computational objects C-graphs, we introduce the concept of height, which characterizes the maximal number of steps a computational object may engage. Then we give a recursive equality of the number of the C-graphs of height , and prove that its growth rate is non-elementary. This result classifies the worst case branching time complexity.

KEY WORDS: vector addition system, reachability, coverability, boundedness, nondeterminism, C-graph

目 录

插图索引

第一章　绪论

1.1　研究背景

随着现代计算机的发展，软件得到了越来越广泛的应用，但同时伴随了软件中层出不穷的漏洞与隐患，并且有些漏洞很难通过测试寻找出来，比如上世纪 80 年代著名的 Therac-25 事件。

用一些形式化的模型来刻画具体的程序，再在其模型上研究一些验证问题来保证程序的安全性等性质是一种常见的解决方案。注意到在并发系统中，很多行为都是不确定的，因而非确定模型进入了研究者们的视角。Petri 网，自 1962 年由 Carl Petri 在 [1] 中提出以来，已经成为了研究并发系统最重要的数学模型之一。除去在形式化研究中的重要作用外，国内外的研究者们也将 Petri 网运用到生物、化学、金融、网络、安全等不同领域的建模和分析当中[2-19]，比如在 [2] 中 Ball 等人定义了并行库系统 (parameterized library system, PLS) 并实现了一个多线程程序的模型检测工具，其上的验证问题被证明与 Petri 网上的相关问题等价；在业务流程管理当中 Petri 网被广泛用于建模用来设计和分析业务流程[15-16]；Angeli 等人用 Petri 网建模了化学反应网络对其进行动力学分析[12]；在生物学方面，Petri 网也会被用来建模研究代谢途径，具体可以参考 Baldan 等人的调研 [13]；北京交通大学 Lei Lei 教授等人用 Petri 网对无线网络系统建模以实现对无线网络的性能的有效分析[5]；清华大学大学林闯教授等人在 Petri 网的基础上定义了私有 Petri 网，对恶意软件造成的私有信息泄露行为进行分析[6] 等。

向量加法系统 (vector addition system) 在 [20] 中被提出，是一个与 Petri 网等价的一个数学模型，其将 Petri 网进一步简化，将令牌 (token) 的变迁抽象成了向量的加减法，从而具有了非常简洁的描述。该模型上的验证问题：可达性问题，即给定两个格局，问两个格局之间是否存在一条路径，随着研究者们发现许多验证问题上的活性性质都可以归约到可达性问题[21]，同时不仅关于程序验证，很多其他方面比如形式语言理论、逻辑、并发系统里的很多问题都可以规约至可达性问题来解决[22-32]，其成为了向量加法系统中最核心的研究问题。目前而言，随着近两年的进展，研究者对该问题已经有了相对精确的认识，例如对一般情况下的可达性问题已有了完备的结论，但仍然有许多神秘的面纱尚未揭开，比如对固定维的可达性问题的上下界依旧有很大的鸿沟。同时，针对其一些特殊的子类，该问题也有了对应完备的结论[33-39]。本文主要研究一般情况下的向量加法系统上的相关验证问题。

同时一些相关的衍生模型也开始进入研究者们的视角。Hopcroft 在 [40] 中提出了带状态的向量加法系统，并且证明 $d+3$ 维的向量加法系统可以模拟 d 维带状态的向量加法系统，这一额外开销是紧的，从而也说明当维度作为输入的时候，两者的可达性问题是等价的。下推向量加法系统 (pushdown vector addition system)，简单来说是一个带栈的向量加法系统，因此其为研究同时具有递归以及整数变量的程序语言提供了一个简洁的数学模型[41-45]。类似的，还有带测 0 的向量加法系统模型 (vector addition system with zero test)[46-49]，交替向量加法系统模型 (alternating vector addition system)[50-51]，分支向量加法系统模型 (branching vector addition system)[52-58] 等。

另一方面来讨论一些计算 (computation) 的概念。在传统的计算模型比如图灵机[59]，λ-演算[60]，递归论模型[61] 定义了封闭的不交互的项，从而计算的概念可以理解为对一个 f，给定一个输入 x，然后输出结果 $f(x)$。而 f 和 g 相等则可以定义为：如果对于输入 x，存在 f 或者 g 有定义，则有 $f(x)=g(x)$，否则 f 和 g 在 x 都没有定义。

注意到在这些模型中，计算都是确定的，但是当在并发理论中考虑计算的时候则发生了变化，因为此时既有交互又有计算，因此需要考虑非确定计算 (nondeterministic computation)。在这类模型理论所定义的进程中[62-63]，进程最主要的任务是进行交互，计算则是协同帮助交互的一部分，因而不能使用上述定义的相等来刻画两个相等的对象。互模拟 (bismulation) 成为了在并发模型中两个进程相等的重要性质，Milner 和 Sangiorgi 率先在 [64] 用其刻画了对所有并发模型都成立的相等概念，随后又有 [56, 65-66] 等相关工作。Fu 则在 [67] 中系统性的提出了关于模型独立的交互理论。本文将专注于其中的非确定计算，即并发模型中的计算对象。

1.2　研究现状

1.2.1　可达性问题可判定性研究

向量加法系统的可达性问题自提出以来，其可判定性是最先攻克的一个问题。可达性问题指的是给定一个向量加法系统 V 和两个格局 $\mathbf{c},\mathbf{c}'$，问是否存在 V 上的一条路径 π 使得 $\mathbf{c}\xrightarrow{\pi}\mathbf{c}'$？在上世纪 70 年代，Hopcroft 和 Leeuwen 分别在 [40] 和 [68] 对低维的 VAS 的可达性问题给出了可判定的证明，Sacerdote 和 Tenney 在 1977 年提出了一个对一般情况下 VAS 的可达性问题可判定的证明[69]，但并不完整。而到了 1980-1990 早期这段时间，Mayr,Kosaraju 和 Lambert 对这个证明进行了补充[70-72]，最终完成了可达性可判定性的证明。其采用的方法被合起来称之为

KLMST 分解算法。由于证明复杂，后人依旧对可判定性的证明做着简化的研究。Leroux 在 2011 年提出了一个由 Presburger 递归不变量的方法给出的可判定性证明[73]，其通过给出一个不可达的证据（witness）存在性给出了两个半可判定的算法来证明 VAS 可达性的可判定性。这个证明相比于早期的 KLMST 简洁易懂，但是最新的研究表明[74] 一些特定的向量加法系统里，如果有两个格局其之间可达的路径是比较远的话，那么其中两个不可达的格局的证据也会很大，从而表明从寻找不可达的证据出发，似乎也不能减少可达性问题的复杂性。

1.2.2 可达性问题复杂性研究

在获取可判定性结论的同时，对于 VAS 可达性问题的复杂性研究也在大量的进行当中。首先是对于固定维数的 VAS 的可达性问题复杂性研究。Hopcroft 在 [40] 证明了 5 维向量加法系统的可达集是一个半线性集合（semi-linear set），而维度 ≥ 5 时则不是，并给出了一个具体的反例。通过这个基础，研究者目前已经在 1 维带状态的向量加法系统和 2 维带状态的向量加法系统上获得了完备的结论。Hasse 在 [75] 证明了 1 维上的可达性问题是 NP 完备的，Leroux 在 [76] 证明了 2 维的向量加法系统有平坦性（flatness）的性质，即任何一条路径都可以被有大小限制的圈所刻画。Blondin 最后在 [77] 提出了线性路径策略（linear path scheme）的思想从而获得了多项式空间的算法，并且通过有界单计数器自动机模型上可达性问题的规约，获得了 **PSPACE**-完备的结论，在 [78] 中也提出了一个新的证明证明该问题是 **PSPACE**-完备的结论。然而对于更高的维度 (≥ 3)，目前还没有完备的结论。但随着诸多研究的推动，比如针对一般情况下的向量加法系统的可达性研究[79-82]，以及对于一些特定的向量加法系统的研究，比如在 [74] 中所提到的拥有 2-指数长度的路径的 4 维带状态的向量加法系统，可以发现维度的增加基本可以极大程度的提高可达性问题的复杂性。

对于任意维度的向量加法系统的可达性问题复杂性问题上也有了不少不错的进展。首先是其上界，上界获得的技术基本上是从对 KLMST 算法的分析而获得。Leroux 和 Schmitz 在 [83-84] 通过给出基于理想（ideal）的分解算法重新解释了 KLMST 算法，并且给出了可达性性问题的第一个上界 $\mathbf{F}_{\omega^3}$，这个算法通过使用了 Karp-Miller 覆盖树[20] 因此会有一个 $\mathbf{F}_{\omega}$ 的下界。随后 Schmitz 在 [85] 将上界改善成了 $\mathbf{F}_{\omega^2}$。最新的结论是 Leroux 在 [82] 给出的的 $\mathbf{F}_{\omega}$ 上界，即 **Ackermann** 上界，其通过对可泵性（pumpability）判定的改进-提出了一个指数空间的算法和引入了一个新的秩函数极大地改善了上界的复杂度。注意到其是一个非原始递归的上界，因此这依旧是个非常复杂的算法。

在下界方面，下界的证明技术比较统一。在这种高复杂性类里，研究领域有两个非常基本的问题：被限制住时间的图灵机（Turing machine）停机问题以及被限制住计数器大小的技术程序（counter machine）接收问题[86-88]。首先的一个结论是 1976 年 Lipton 在 [89] 证明了指数空间难 (**EXPSPACE**) 的下界，其成功将一个被限制在 2^{2^n} 大小的计数程序接收问题规约至可达性问题。Lipton 提出了一个非常重要的思路，因为对于向量加法系统和计数程序来说，最大的差别在于计数器的能力上; 计数程序的计数器能力会更强，能够去执行测试 0 的操作，因而在规约的过程中，最大的难点在于如何用不能测试 0 的计数器去模拟那些能测试 0 的计数器，当然这些被模拟的计数器有大小的限制，否则一旦存在两个以上不受限的计数器，其问题便是不可判定的。Lipton 提出了通过维持 $x + \hat{x} = B$ 的这样的形式，即构造一个计数器的补，使两者的和始终维持等于 B 不变，这样对 x 测试 0 便相当于计数器 $\hat{x}$ 可以连续进行 B 次减法，从而模拟了测试 0 这一操作。

然而在此后的 40 年里，可达性问题的下界并没有获得任何改进，直到最近两年，Czerwiński 在 [74] 将下界提升到了非初等下界（non-elementary）。能产生这一重大改进的原因是 Czerwiński 提出了一个放大器（amplifier）的概念，他能将一个计数器限制在 k! 大小下的计数器机的接受问题转化成一个计数器限制在 k 大小下的计数器机的接收问题，通过对这个放大器的迭代，其证明了 VAS 可达性问题的下界至少是非初等的。这里使用了一个核心技巧，将一些需要测 0 的命令一起执行，具体来说，在利用 Lipton 提到的测 0 的转换思路上，引入三个计数器 b, c, d 使其满足其中的值为：$v(b) = B$, $v(d) = v(b) \cdot v(c)$，则当需要对被模拟的计数器 x 测 0 时，我们可以在一边对 $\hat{x}$ 作减法的同时，将 d 也做对应次数的减法，整个过程完成后再将 c 也作一次减法；这样做的好处是并不需要在这次模拟测 0 时确保 $\hat{x}$ 正好做了 B 次减法，因为如果没有做足，最后不可能使得 d, c 里的值都变为 0，即会在模拟的最后一步失效。采用这样一个技术的好处是极大的增加了被模拟测 0 计数器的限制，但是也使得其不能像 Lipton 的方法那样获得其他验证问题比如可覆盖性问题，有界性问题的下界结论。

上述方法可以简单理解为当没有按设想的去模拟测 0 时留下一些可以在最后验证的副作用，或者说将一些测 0 操作合并起来最后一起验证。随着这一想法的提出，原本可达性问题上巨大的上下界鸿沟已经从 **EXPSPACE**-难至 **Ackermann**-难缩小至了 **Non-Elementary**-难至 **Ackermann**-难。而这一想法也打开了研究者们对如何更好的模拟测 0 的思路。随后，Czerwiński，Leroux，Lasota 等人分别各自在 [80-82, 90] 提出了更好的模拟测 0 的思路：Czerwiński 通过将更大一部分模拟测 0 的命令汇聚到一起验证其正确性获得了新的下界 **Ackerman**-难；Lasota 则通

过限制测 0 的次数，从而解放了被模拟测 0 计数器的大小限制，因而也获得了 **Ackermann**-难的下界；Leroux 则通过进一步放宽放大器这一概念，提出了预生成器 (pre-amplifier) 这一概念，不仅获得了 **Ackermann**-难的下界，也获得了目前最好的固定维可达性问题的下界。至此，任意维度下的向量加法系统的可达性问题已经被证明是 **Ackermann**-完备的。需要注意的是，尽管对任意维的可达性问题已经有了完备的结论，但是对固定维 $d(\geq 3)$ 的可达性问题，目前还有着 $F_{\frac{d-4}{2}}$-难到 F_{d+4} 上界的巨大鸿沟，这里不作过于严谨的表述，因为当 d 较小时比如 $= 3, 4$ 等，目前还有只有 2 维的平凡推论，即 **PSPACE**-难。

1.2.3 其他验证问题

在向量加法系统的验证问题中，除了可达性问题以外，还有许多其他的验证问题也发挥着重要的作用。

- 可覆盖性问题 (coverability problem) 和有界性问题 (boundness problem)。这两个问题可以视作是可达性问题的一个松弛版本。前者指的是给定向量加法系统 V 和两个格局 $\mathbf{c}, \mathbf{c}'$，问是否存在一个格局 $\mathbf{c}''$ 使得 $\mathbf{c} \xrightarrow{*} \mathbf{c}'' \geq \mathbf{c}'$; 后者问的则是在 V 中 $\mathbf{c}$ 可以到达的格局是否是有限的？这两个问题最先在 [20] 中被证明是可判定的，其用到的核心技术：Karp-Miller 树被广泛运用到各个模型的验证问题当中[52, 91-92]。随后其在 [89, 93] 中两个问题被证明都是 **EXPSPACE**-完备的，最近也有一个新的关于该问题的完备证明[94]。

  而关于固定维度上的这两个问题，首先在 [95] 中其证明当维度 $d \geq 4$ 时这两个问题都是 **PSPACE**-完备的；随后在 [96] 中被证明 3 维的可覆盖性问题以及有界性问题也是 **PSPACE**-完备的；当维度 $d = 1$ 时则在 [97] 中被证明是 **NP**-完备的；最后在关于 2 维带状态向量加法系统可达性问题的研究中[77] 证明了其在维度 $d = 2$ 的时候也是 **PSPACE**-完备的，从而给出了整个问题完整的答案。此外，近些年也有对有界性问题变种的一些研究，比如有助于时间线型逻辑检测的可逆有界性 (reversal boundedness)[98-100] 以及状态有界性 (place boundedness)[20] 和选择无界性 (selective unboundedness)[101] 等问题。
- 可达集全集问题。如果我们在定义一个带状态的向量加法系统中类似自动机那样定义每条边的标记和初始接收状态，则就可以定义一个全集问题，即在 V 中由初始格局 $\mathbf{c}$ 出发可达所有到达接收状态的路径的对应串是否构成了自目标的全集？该问题某种程度上也刻画了可达性的描述，其在 [102-103] 中被证明了就算仅是 1 维，该问题也是 **Ackermann**-完备的。而在 [104] 其

考虑了一个无歧义性 (unambiguous) 的性质，即对于任何一个串，最多只对应的一条接收路径的限制下，证明了该问题是 **EXPSPACE**-完备的，并且对于固定维的无歧义向量加法系统的全集问题，其也证明了当维度 $d = 1$ 时是 co-**NP**-难的，而对于维度 $d \geq 2$ 则是 **PSPACE**-难的。

- 一些不可判定的问题。最后再简单的介绍一些其上的不可判定问题。
  - 可达集包含问题。可达性问题的一个简单拓展就是研究其可达集的关系。所谓可达集，即从一个格局 **c** 出发所能达到的所有格局的集合。在 [105] 中，其通过将希尔伯特第十问题 (Hilbert tenth problem)[106] 规约至该问题获得了不可判定的结论，同时也证明了可达集相等问题也是不可判定的。
  - 强互模拟等价问题。强互模拟等价是并发系统中非常值得研究的问题。Petri 网作为其上非常重要的一个模型，对于该问题却只有否定的答案，Jančar 在 [107-108] 中证明了该问题的不可判定性。

1.2.4 其他相关模型

随着向量加法系统的研究进展，一些相关的衍生模型也开始映入研究者的眼帘。

- 带测 0 的向量加法系统。在 [87] 中其证明了只要有两个计数器，其停机问题便是不可判定的。因此如果给向量加法系统中的两维增加测 0 的能力，其可达性问题便是不可判定的。一个很自然的问题是如果只有一维可以测 0，我们称其为带测 0 的向量加法系统，该模型上的可达性问题是否可以判定? 在 [109] 中 Reinhardt 第一次给出了其可判定性的证明，而在 2011 年 Bonnet 通过 Presburger 递归不变量这一技术给出了一个更为简洁的证明[46]。而其的复杂性结论目前只有平凡的向量加法系统的下界的直接推论。而这方面最新的一个研究是 Leroux 在 2020 年发表的 [47]，其考察了维度为 2 的带测 0 的向量加法系统上的可达性问题，其通过使用在 [77] 中介绍的线性路径策略技术证明了 **PSPACE** 的上界，从而获取了其可达性问题是 **PSPACE**-完备的结论。
- 下推向量加法系统。这里还有一个与之几乎相同的模型：语法控制的向量加法系统（grammar-controlled vector addition system，GVAS），其对应关系可以参考下推系统（pushdown system）与上下文无关语法（context-free grammar）的转换关系可得。关于该模型下的验证问题，研究者也有了不少的研究。Leroux 等人在 [41] 证明了其终止性和有界性都是可判定的，但是对于更受关注的可覆盖性问题和可达性问题上，目前还没有可判定性的结

论，目前仅仅只知道其有非初等的下界 [55]. 有趣的是，相别于向量加法系统的可达性问题和可覆盖性问题的巨大差别，该模型的可达性和可覆盖性在不固定维数的时候是一致的；Leroux 在 [42] 中证明了 n 维下推向量加法系统的可达性问题是可以规约至 $(n+1)$ 维下推向量加法系统的可覆盖性问题。而在固定的维度上，研究者们也有一些进展。对于 1 维的下推向量加法系统来说，Leroux 等人在 [42-44] 中证明了其可覆盖性问题蕴含在 **NP**-难到 **EXPSPACE** 之间，而有界性问题在 **NP**-难到 **EXPTIME** 之间。

- 交替向量加法系统。该模型是由 Lincoln 等人在研究命题线性逻辑时提出来的一个 VAS 的扩展模型[50]。它引入了分叉规则，使得其作用在同一格局时，下一格局会存在两种情况。很遗憾的是关于其可达性的结论是不可判定的。但如果我们就可达性问题削弱为状态可达性问题，即只要求状态可达，则 Courtois 等人在 [51] 证明了其是双指数完备的，并且如果维数固定时，状态可达性问题难度降低为指数时间完备的。另一方面，该问题令人感兴趣的一点是其状态可达性问题可以有效的规约到基本并发进程（Basic Parallel Process,BPP）, 带状态的向量加法系统与有限状态系统的模拟问题上，从而可以期待得到新的下界结论。
- 分支向量加法系统。该模型将向量加法系统线性的计算过程扩展成了一颗计算树[52]，其非叶子节点的向量值被刻画成其儿子节点的向量值以及对应的规则之和。该模型可以有效刻画计算语言学、逻辑乃至 XML 中的一些核心问题，因为也引起了研究者们的兴趣。Lazic 和 Demri 在 [53-54] 中对其一系列验证问题进行了研究，得到了该模型的有界性问题和可覆盖性问题是 2-**EXPTIME**-完备的，而其可达性问题是 2-**EXPSPACE**-难的，随后在 [55] 中进一步将一般的可达性问题提升到了非初等的下界，但是目前依旧不清楚其是否可判定。非常有意思的是，在维度较低时，分支向量加法系统的验证问题体现出了非常优秀的低复杂性。在 2016 年 Göller 等人证明了在一进制编码的时候，一维分支向量加法系统上的可达性、可覆盖性、有界性问题都是多项式时间完备的[56]；而在二进制编码的时候一维分支向量加法系统的可达性问题在被证明是多项式空间完备的[58]。在 2019 年 Mzaowiecki 等人考虑了有界的分支向量加法系统，并且得到了二维有界分支向量加法系统可达性问题是指数时间完备的结论[57]。

1.2.5 非确定计算

目前对进程的研究当中，存在非常多的从定性的角度去研究的成果[62-63, 110-113]，但是从定量的角度去研究进程的性质的研究目前还十分稀缺。这样的研究同样是有意义的，比如询问一个 n 个状态可以在 a, b, c 三个信道交互的不同的 CCS 进程有多少个可以在某种程度上反映出非确定性带来的复杂性。Fu 在 [114] 中提出应当率先在简单的模型：只有 τ 动作的 $\mathbb{CCS}^{\mu}$ 中进行对应的研究，其原因有二：首先这样的限制能够仅仅关注于内部动作的不确定性所产生的复杂性；其次尽管是如此简单的模型，其结构依旧是不平凡的，我们可以构造出任意多个不相等的计算对象。并且其在里面定义了计算对象的图的抽象表示：C-图，为作定量的分析打下了坚实的基础。

1.3 本文贡献

本文首先将着重介绍向量加法系统上的相关验证问题，其主要贡献如下：

- 首先归纳总结了向量加法系统上一些验证问题的研究结果以及研究技术，包括可覆盖性问题和有界性问题 (第三章)，任意维度下的可达性问题上界算法 (第四章和第五章)，以及固定维度下的可达性问题 (第七章)，尽管在过去的时间中已经有了很多这样的工作[22, 47, 86, 115-118]，但是随着近两年技术的改进导致其结论的一再更新，给予这些问题这些年的研究的一个总结是很有意义的。
- 其次在第四章，本文提出了有别于 Leroux 提出的 KLMST 算法的理解思路的算法解释，该解释可以简单的用分解-降维-分解-降维-$\cdots$ 所来表示，并且相信其能反应上不同维度的向量加法系统的可达性问题之间的关系，从而对研究固定维度的向量加法系统的可达性问题起帮助的作用。
- 对于可达性问题的下界证明所用的模拟测 0 技术，本文归纳总结了目前所有的技巧，并且阐述了其中的差别和为什么其能拥有更强的能力，相信此对于研究一些其他模型相关验证问题的下界可以产生指导的作用。

此外，本文研究了并发理论中的一类进程-有限状态计算对象。在只有 τ 动作的 $\mathbb{CCS}^{\mu}$ 模型中，引用了 Fu 在 [114] 对有该类进程的一种的抽象刻画：C-图，并做了如下研究：

- 本文定义了 C-图的高度这一概念，并通过该参数计算得到了不同高度下 C-图个数的一个递推式，给出了不同高度下 C-图个数的大小以及增长速度，该大小有助于理解非确定计算分支的计算复杂性，也是第一个以定量角度来研究类似问题的研究。

1.4 章节安排

本文后续的安排章节如下：

- 第二章将介绍本书所要用到的一些基本知识，包括向量加法系统的基本概念，基本的良拟序理论基础，快速增长层级复杂性理论介绍和线性方程组正整数解的基本理论。
- 第三章则将介绍向量加法系统的可覆盖性问题和有界性问题，本文将首先介绍被广泛运用的 Karp-Miller 树技巧，以此来获得这两个问题的可判定性结论。随后进一步介绍如何获取一个在指数空间内的算法，从而获得其是 **EXPSPACE**-完备的结论。
- 第四章开始至第七章将开始介绍可达性问题。首先在第四中本文将介绍可达性问题的最新上界算法-KLMST 算法，并且给出一个新的解释来理解该算法，该解释可以将维度之间的变化联系起来。
- 第五章将介绍另一个可达性问题的判定性算法-Presburger 不变量方法。
- 第六章将介绍可达性问题的下界研究。本文将众多的模拟测 0 技术归纳起来，企图可以对其他相关模型的相关验证问题的下界产生帮助。
- 第七章则将介绍固定维度下带状态向量加法系统的可达性问题的研究。首先将介绍 1 维带状态的向量加法系统的可达性问题是 **NP**-完备的，其次将介绍 2 维带状态的向量加法系统的可达性问题是 **PSPACE**-完备的，最后则讨论目前更高维度的带状态向量加法系统的可达性问题的困难之处。
- 第八章将从计数的方式研究有限状态计算对象的个数，对于其等价的抽象表示：C-图，文章将给出以高度为参数得到的 C-图个数的递推式，并且给出其相应的增长速度，这一结论有助于研究计算对象分支下的时间复杂性。
- 第九章则是全文总结和未来展望。

第二章　预备知识

2.1　向量加法系统

本章将开始介绍向量加法系统的基本概念，以及本文中所需要用到的一些其他的基本知识。

2.1.1　基本记号

本文用 u, v, w 等表示整数，$|u|$ 表示 u 的绝对值。令 $\mathbb{N}, \mathbb{Z}$ 分别表示自然数和整数集合，用下标表示他们所处的范围，比如 $\mathbb{Z}_{\geq 0}$ 表示非负整数集合。特别的，用 $\mathbb{N}_\omega$ 表示集合 $\mathbb{N} \cup \{\omega\}$, 这里 ω 可视作一个比任何自然数都要大的数，即 $\forall n,\ n < \omega$，并且满足对于 $\forall n \in \mathbb{N}$ 有 $n + \omega = \omega$。关于 ω 的具体信息，可以在后面章节 2.3.1获取。对于 $\mathbb{N}_\omega$，定义序关系 $\sqsubseteq$，即称 $x \sqsubseteq y$ 如果 $y \in \{x, \omega\}$。此外，用 $[d]_0$ 表示不超过 d 的自然数，即 $[d] = \{0, 1, 2, \ldots, d\}$，用 $[d]$ 表示不超过 d 的正整数，即 $[d] = \{1, 2, \ldots, d\}$。

给定一个集合 X，用 $|X|$ 来表示集合里元素的个数。对于两个集合 X, Y，称 $X \times Y = \{(x, y) | x \in X,\ y \in Y\}$ 为 X, Y 的笛卡尔积。给定一个集合 X，用 X^d 表示 d 个 X 的笛卡尔积，也即 X 上所有 d 维向量的集合。令 $\mathbf{u}, \mathbf{v}, \mathbf{w}$ 表示向量，对于一个 d 维向量 $\mathbf{u}$，用 $\mathbf{u}[i]$ 表示其第 i 维的大小，用 $||\mathbf{u}||_p$ 表示 p-范数，这里 p-范数的定义如下：

$$||\mathbf{u}||_p \stackrel{\text{def}}{=} (|\mathbf{u}[1]|^p + \ldots + |\mathbf{u}[d]|^p)^{\frac{1}{p}} \tag{2–1}$$

特别的，用 $\mathbf{0}_d$ 表示全为 0 的 d 维向量，在向量空间不会引起疑义的情况下，也会用 $\mathbf{0}$ 来表示零向量。

例 2.1　考虑一个 3 维向量 $\mathbf{u} = (2, -1, 3)$，其 1-范数、2-范数、∞-范数分别为：

$$||\mathbf{u}||_1 = 6,\ ||\mathbf{u}||_2 = \sqrt{14},\ ||\mathbf{u}||_\infty = 3. \tag{2–2}$$

给定一个 d 维向量 $\mathbf{u}$ 和一个集合 $I \subseteq [d]$，定义 $\mathbf{u}$ 在 I 上的射影 $\mathbf{u}_I$ 为:

$$\mathbf{u}_I[i] \stackrel{\text{def}}{=} \begin{cases} \mathbf{u}[i], & \text{if } i \in I, \\ \omega, & \text{o.w.} \end{cases}$$

现在将序关系 $\sqsubseteq$ 扩展到 $\mathbb{N}_\omega^d$ 上，即对于 $\mathbb{N}_\omega^d$ 上的两个向量 $\mathbf{u}, \mathbf{v}$，称 $\mathbf{u} \sqsubseteq \mathbf{v}$，如果对于任何一维 $i \in [d]$ 都有 $\mathbf{u}[i] \sqsubseteq \mathbf{v}[i]$。此外，我们自然的定义向量上的 $\leq$ $(<)$ 关

系，即对于两个 d 维向量 $\mathbf{u},\mathbf{v}$，如果对于任意的 $i \in [d]$ 都有 $\mathbf{u}[i] \leq (<)\mathbf{v}[i]$，则称 $\mathbf{u} \leq (<)\mathbf{v}$。向量的字典序关系 $<_{lex}$ 则定义如下：即对于两个 d 维向量 $\mathbf{u},\mathbf{v}$，如果最小的满足 $\mathbf{u}[i] \neq \mathbf{v}[i]$ 的 $i \in [d]$ 满足 $\mathbf{u}[i] < \mathbf{v}[i]$，则称 $\mathbf{u} \leq_{lex} \mathbf{v}$，不难发现 $\leq$ 是一个偏序而 $<_{lex}$ 是一个全序。

例 2.2 考察三个 $\mathbb{N}_\omega^d$ 上的向量 $\mathbf{u} = (1,\omega,3)$, $\mathbf{v} = (\omega,\omega,3)$, $\mathbf{w} = (\omega,\omega,4)$，有 $\mathbf{u} \sqsubseteq \mathbf{v}$, 但是 $\mathbf{u} \not\sqsubseteq \mathbf{w}$, $\mathbf{v} \not\sqsubseteq \mathbf{w}$。

给定集合 X，一个定义在其上面的关系 R 是 $X \times X$ 的一个子集，其对应的逆关系 R^{-1} 定义为 $R^{-1} = \{(y,x)|(x,y) \in R\}$。给定 X 上的两个关系 R_1 和 R_2，定义其关系的复合 $R = R_1 \circ R_2 = \{(x,z)|\exists y \in X,\ (x,y) \in R_1 \wedge (y,z) \in R_2\}$。令 $R^{(n)}$ 表示关系 R 自身的 n 次复合，特别的 $R^0 = \{\epsilon\}$。我们称 $R^* = \bigcup_{i\in\mathbb{N}} R^{(i)}$ 表示 R 的克莱尼星闭包。

注 一般来讲，关系的复合定义与本文给出的是相反的，即 $R = R_1 \circ R_2 = \{(x,z)|\exists y \in X,\ (x,y) \in R_2 \wedge (y,z) \in R_1\}$。但是为了后文的使用方便，本文使用了将 R_1, R_2 对换过来的关系复合的定义方式。

给定一个集合 X，集合 X 上的一个串是 $\xi = x_1x_2\ldots x_n \in X^*$，用 $|\xi| = n$ 来表示串的长度，用 $\xi[i]$ 表示串上第 i 位的元素，用 $\xi[i,j] = x_i\ldots x_j$ 表示 ξ 上的一个子串，特别的如果 $i > j$，有 $\xi = \epsilon$ 即为空串。对于两个串 $\xi_1 = x_1\ldots x_n, \xi_2 = y_1\ldots y_m$，用 $\xi = \xi_1\xi_2$ 表示其连接，即 $\xi = x_1\ldots x_ny_1\ldots y_m$。

2.1.2 不带状态的向量加法系统

本节开始介绍不带状态的向量加法系统 (Vector Addition Systems, VAS)[20]。作为一个更数学化的并发程序模型，不带状态的向量加法系统显得十分简洁。为了叙述方便，以下用向量加法系统作为简称。

定义 2.1 (d-VAS) 一个 d 维的向量加法系统 V 是一个 d 维整数向量的集合 $\mathbf{A}_V \subseteq \mathbb{Z}^d$。称 $\mathbf{A}_V$ 里的元素是 V 的一个迁移动作（action）。

对于 d 维向量加法系统 V，令

$$|V| \stackrel{\text{def}}{=} d \cdot |A_V| \cdot \max_{\mathbf{a}\in\mathbf{A}_V} \log_2(||\mathbf{a}||_\infty)$$

表示 V 的大小，特别的如果省略掉其中的 $\log_2$，则称其为 V 在一进制编码的大小，并用 $||A_V||$ 表示 $\max_{\mathbf{a}\in\mathbf{A}_V} ||\mathbf{a}||_\infty$。一个格局（configuration）$\mathbf{u} \in \mathbb{N}^d$ 是一个 d 维自然数

向量。对于其中的每一个动作 $\mathbf{a} \in \mathbf{A}_V$，定义其在 $\mathbb{N}^d$ 上的一个关系：$\xrightarrow{\mathbf{a}} = \{(\mathbf{x}, \mathbf{y}) | \mathbf{y} = \mathbf{x} + \mathbf{a} \wedge \mathbf{x}, \mathbf{y} \in \mathbb{N}^d\}$，也记作 $\mathbf{x} \xrightarrow{\mathbf{a}} \mathbf{y}$，称 $\mathbf{x}$ 做了一步迁移动作 $\mathbf{a}$ 到 $\mathbf{y}$，也称格局 $\mathbf{x}$ 能触发迁移规则 $\mathbf{a}$。特别的，定义 $\xrightarrow{\mathbf{A}_V} = \bigcup_{\mathbf{a} \in \mathbf{A}_V} \xrightarrow{\mathbf{a}}$。对于自然数向量 $\mathbf{x}_0, \ldots, \mathbf{x}_n \in \mathbb{N}^d$ 和 $\mathbf{a}_1, \ldots, \mathbf{a}_n \in A_V$，如果存在如下关系：

$$\mathbf{x}_0 \xrightarrow{\mathbf{a}_1} \mathbf{x}_1 \xrightarrow{\mathbf{a}_2} \cdots \xrightarrow{\mathbf{a}_n} \mathbf{x}_n, \tag{2–3}$$

称 $\pi = \mathbf{a_1 a_2} \ldots \mathbf{a_n}$ 是格局 $\mathbf{x}_0$ 到格局 $\mathbf{x}_n$ 的一条路径，也称 $\mathbf{x}_0$ 能触发路径 π、π 对于 $\mathbf{x}_0$ 是合法的；将 $\rho = (\mathbf{x}_0, \pi, \mathbf{x}_n)$ 称为一个运行，有时也会将其拆开来记，即 $\rho = (\mathbf{x}_0, \mathbf{a}_1, \mathbf{x}_1) \ldots (\mathbf{x}_{n-1}, \mathbf{a}_n, \mathbf{x}_n)$，令 $\Delta(\pi) = \sum_{i=1}^{n} \mathbf{a}_i$ 表述路径 π 的增量，串 $\mathbf{x}_0 \mathbf{x}_1 \ldots \mathbf{x}_n$ 称为运行 ρ 的格局串，并且用 $path(\rho)$ 表示其对应的路径 π。此外称 $\mathbf{x}_0$, $\mathbf{x}_n$ 为该运行的起点和终点，记作 $src(\rho)$ 和 $tgt(\rho)$。格局 $\mathbf{x}_n$ 对于 $\mathbf{x}_0$ 是可达的（reachable），反之如果不存在这样一条路径，则称格局 $\mathbf{x}_n$ 对于 $\mathbf{x}_0$ 是不可达的。关系 $\xrightarrow{V} = (\xrightarrow{\mathbf{A}_V})^*$ 称为 V 上的可达关系，在考虑的向量加法系统 V 没有疑义的情况下，将可达关系简写为 $\xrightarrow{*}$。如下集合：

$$\mathrm{Reach}_V(\mathbf{x}) \stackrel{\text{def}}{=} \{\mathbf{y} | (\mathbf{x}, \mathbf{y}) \in \xrightarrow{V}\} \tag{2–4}$$

称为格局 $\mathbf{x}$ 在向量加法系统 V 上的可达集，简称 $\mathbf{x}$ 的可达集。

我们同样可以介绍在任意空间下的可达关系。给定一个向量空间 $\mathbb{V}$，如果对于其上的向量 $\mathbf{x}_1, \ldots \mathbf{x}_n \in \mathbb{V}$，存在一条路径 $\pi = \mathbf{a}_1 \mathbf{a}_2 \ldots \mathbf{a}_n$ 满足：

$$\mathbf{x}_0 + \mathbf{a}_1 = \mathbf{x}_1, \ldots, \mathbf{x}_{n-1} + \mathbf{a}_n = \mathbf{x}_n. \tag{2–5}$$

则称 $\mathbf{x}_n$ 对于 $\mathbf{x}_1$ 在 $\mathbb{V}$ 上是可达的，记作 $\mathbf{x}_0 \xrightarrow{V}_{\mathbb{V}} \mathbf{x}_n$，$\mathbf{x}_i$ 称为在 $\mathbb{V}$ 上的格局。一步可达关系也可类似的定义为 $\xrightarrow{\mathbf{A}_V}_{\mathbb{V}}$，其他概念也可以类似定义。这里最常用的不在 $\mathbb{N}^d$ 上的可达关系是在整数向量空间上的可达关系，即 $\mathbb{V} = \mathbb{Z}^d$。

接下来介绍 Parikh 像的概念。给定一个路径 π，其 Parikh 像是一个函数 $\mathrm{Par}_\pi : \mathbf{A}_v \to \mathbb{N}$，定义为动作 $\mathbf{a}$ 在路径 π 出现的次数，即 $\mathrm{Par}_\pi(\mathbf{a}) = |\{i : \pi(i) = \mathbf{a}\}|$，这里用 $\pi(i)$ 表示路径 π 上的第 i 个元素，而 $|\mathrm{Par}_\pi| \stackrel{\text{def}}{=} \sum_{\mathbf{a} \in \mathbf{A}_V} \mathrm{Par}_\pi(\mathbf{a})$ 则表示其 Parikh 像的大小。

例 2.3　考虑如下一个二维向量加法系统 V，其动作集合 $\mathbf{A}$ 定义如下：

$$\mathbf{A} = \{(3, -2), (-3, 3)\} \tag{2–6}$$

则路径 $(3, -2)(-3, 3)$ 是格局 $(0, 2)$ 到 $(0, 3)$ 的一条路径，对于格局 $(0, 2)$ 来说，其可达集为：$\mathrm{Reach}_V((0, 2)) = \{(3m, k) | 3m + k \geq 2 \wedge m, k \in \mathbb{N}\}$。而对于格局 $(0, 1)$ 来说，其可达集为空，即不存在任何一个以格局 $(0, 1)$ 为起点的运行。

2.1.2.1 向量加法系统上的验证问题

本节来定义一些在向量加法系统上的重要验证问题。

问题 2.1 (可达性问题 (reachability)) 给定一个向量加法系统 V 和两个格局（向量）$\mathbf{u}$, $\mathbf{v}$，问格局 $\mathbf{v}$ 对于格局 $\mathbf{u}$ 是否是可达的?

可达性问题是向量加法系统验证问题中最重要的一个问题，很多并发程序的验证问题都可以规约至此。该问题在很长一段时间有着巨大的鸿沟（下界是 **EXPSPACE**-难[89], 上界是可判定[69-72]），直到近些年才被证明是 **Ackermann**-完备[79-82]。我们将在后面的章节四,五,六,七作详细的介绍。

问题 2.2 (可覆盖性问题 (coverability)) 给定一个向量加法系统 V 和两个格局（向量）$\mathbf{u}$, $\mathbf{v}$，问是否存在一个格局 $\mathbf{v}' \geq \mathbf{v}$，使得格局 $\mathbf{v}$ 对于格局 $\mathbf{u}$ 是可达的?

问题 2.3 (有界性问题 (boundedness)) 给定一个向量加法系统 V 和一个格局（向量）$\mathbf{u}$，问格局 $\mathbf{u}$ 的可达集是否是有限的?

可覆盖性问题和有界性问题都是向量加法系统验证问题中非常重要的问题，其在 1978 年被 Rackoff 证明为了是 **EXPSPACE**-完备的[93]。得益于其相对复杂性低的结论，以及可以用来刻画活性 (liveness), 安全性 (safety) 等重要的并发程序性质，其在验证中发挥了巨大的作用。我们将在章节三作详细的介绍。

问题 2.4 (可达集包含问题 (inclusion)) 给定两个个向量加法系统 V_1, V_2 和两个格局（向量）$\mathbf{u}_1, \mathbf{u}_2$，问格局 $\mathbf{u}_1$ 在 V_1 的可达集是否是格局 $\mathbf{u}_2$ 在 V_2 的可达集的子集?

对于向量加法系统上可达集性质的研究则有一些否定的结论。类似的问题还有判断两个可达集是否相等的相等问题，以及判断某个可达集在某个子空间的投影是否是另一个可达集的子集的子空间包含问题都在 1976 年被证明为是不可判定的[105]。

2.1.3 带状态的向量加法系统

这一节将介绍向量加法系统的一个简单变形，带状态的向量加法系统 (Vector Addition System with states, VASS)。简单的来说，前面介绍的向量加法系统可以视作只有一个状态，而带状态的向量加法系统可以认为是被某种正则语言在控制其做的动作顺序，或者说是被一张有向图所控制。令人惊奇的是，在不固定维度的时候带状态的向量加法系统和不带状态的向量加法系统有着相同的能力。

首先来介绍最基本的带状态的向量加法系统的定义。

定义 2.2 (d-VASS)　一个 d 维带状态的向量加法系统 V 是一个边上被 d 维整数向量标记的有向图，记作 $V = (Q_V, T_V, A_V)$。其中 Q 是一个有限的状态集合，$A_V \subseteq \mathbb{Z}^d$ 是动作集合，$T_V \subseteq Q_V \times A_V \times Q_V$ 是边的集合，也被称作为迁移规则。

注　如果 $d = 1$，我们也将其称为一维计数器网 (one-counter net)。

对于一个 d 维带状态的向量加法系统 V，定义

$$|V| \stackrel{\text{def}}{=} |Q_v| + d \cdot |T_v| \cdot \max_{t=(p,\mathbf{a},q)\in T_v} \log_2(||\mathbf{a}||_\infty)$$

表示其大小，并且用 $||T_v||$ 表示 $\max_{t=(p,\mathbf{a},q)\in T_v} ||\mathbf{a}||_\infty$。一个格局 $\mathbf{c}$ 是由一个状态和 d 维自然数向量组成的两元对，记作 $\mathbf{c} = (q, \mathbf{u})$，有时也写为 $q(\mathbf{u})$。特别的，对于格局 $\mathbf{c} = q(\mathbf{u})$，用 $\text{State}(\mathbf{c}) = q$ 表示其状态，用 $\text{Value}(\mathbf{c}) = \mathbf{u}$ 表示其上面的值。同样的，如果 $\mathbf{u} \in \mathbb{V}$，那称 $q(\mathbf{u})$ 是在向量空间 $\mathbb{V}$ 上的格局。类似于不带状态的向量加法系统上的可达关系，我们也可以定义 V 上的可达关系。给定两个格局 $q_1(\mathbf{u})$, $q_2(\mathbf{v})$，如果存在 $t = (q_1, \mathbf{a}, q_2) \in T_v$ 满足 $\mathbf{v} = \mathbf{u}+\mathbf{a}$，则称 $q_1(\mathbf{u}) \xrightarrow{T_V} q_2(\mathbf{v})$，也记作 $q_1(\mathbf{u}) \xrightarrow{\mathbf{a}} q_2(\mathbf{v})$ 或者即格局 $q_2(\mathbf{v})$ 对于 $q_1(\mathbf{u})$ 在 V 上是一步可达的。令 $\xrightarrow{V} = (\xrightarrow{T_V})^*$ 表示 V 上的可达关系，如果 $p(\mathbf{u}) \xrightarrow{V} q(\mathbf{v})$，则存在一条路径 $\pi = t_1 \dots t_n = (q_0, \mathbf{a_1}, q_1) \dots (q_{n-1}, \mathbf{a_n}, q_n)$ 和一串自然数向量 $\mathbf{v}_0, \dots \mathbf{v}_n$ 满足：

$$p(\mathbf{u}) \xrightarrow{\mathbf{a}_1} q_1(\mathbf{v_1}) \xrightarrow{\mathbf{a}_2} \cdots \xrightarrow{\mathbf{a}_{n-1}} q_{n-1}(\mathbf{v_{n-1}}) \xrightarrow{\mathbf{a}_n} q(\mathbf{v}) \tag{2–7}$$

其中 $p(\mathbf{u}) = q_0(\mathbf{v}_0)$, $q(\mathbf{v}) = q_n(\mathbf{v}_n)$ 并且对于 $i \in [d]$ 有 $\mathbf{v}_i = \mathbf{v}_{i-1} + \mathbf{a}_i$。与不带状态的向量加法系统相同，称 $\Delta(\pi) = \sum_{i=1}^{n} \mathbf{a}_i$ 是路径 π 上的增量，$\rho = (p(\mathbf{u}), \pi, q(\mathbf{v})) = (p(\mathbf{u}), t_1, q_1(\mathbf{v}_1)) \dots (q_{n-1}(\mathbf{v}_n - 1), t_n, q(\mathbf{v}))$ 是一个运行，$path(\rho) = \pi$ 表示运行对应的路径，并且对于一个格局 $\mathbf{c}$ 可以定义在 V 上的可达集 $\text{Reach}_V(\mathbf{c})$ 以及关于一个运行 ρ 上的起点 $src(\rho) = p(\mathbf{u})$ 终点 $tgt(\rho) = q(\mathbf{v})$ 和格局串 $p(\mathbf{u})q_1\mathbf{v}_1 \dots q_{n-1}(\mathbf{v}_{n-1})q(\mathbf{v})$。有的时候也称该路径 π 为 $\mathbf{a}_1 \dots \mathbf{a}_n$，同样还可以定义路径 π 上的 Parikih 像 $\text{Par}_\pi : T_V \to \mathbb{N}$。

注　与上一节相同，我们也可以类似定义在任何向量空间 $\mathbb{V}$ 上的可达关系 $\xrightarrow{V}_{\mathbb{V}}$，这里不再多作阐述。

例 2.4　考虑一个二维带状态的向量加法系统 $V = (Q, T, A)$，其定义如2–1所示：考虑格局 $q_1((0,0))$，$q_3(0,2)$ 是其一个可达的格局，一个可达的路径为：

$$q_1((0,0)) \xrightarrow{t_1} q_2((1,0)) \xrightarrow{t_3} q_2((0,2)) \xrightarrow{t2} q_1((0,2)) \xrightarrow{t_4} q_3((0,2))$$

而 $q_3((0,1))$ 则是一个不可达的格局。

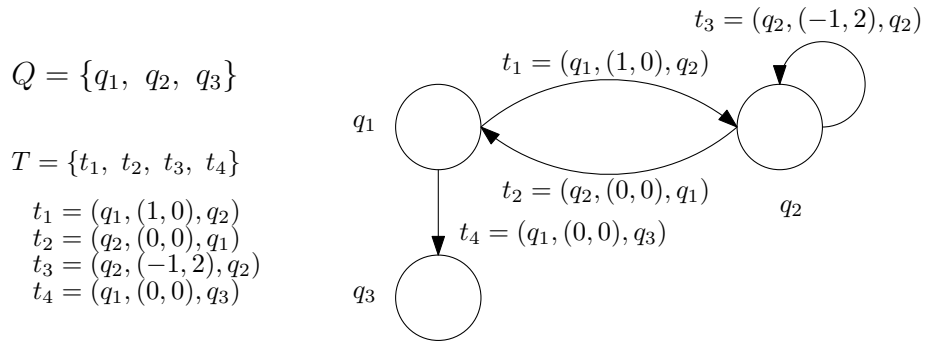


图 2–1 一个两维带状态的向量加法系统例子

Figure 2–1 An Example of 2-VASS

注 可以看出带状态的向量加法系统可以看作一张有向图，Q 表示图上的节点，T 表示对应的边，A 则表示边上的标记，所以在之后有时也会直接用图上的一些概念来直接描述 V 上的内容。

接下来再给出由向量加法系统定义出的语言。为了叙述方便，首先将给出一个略微复杂的定义，需要注意的是该定义只是为了描述其定义出来的语言因此添加了初始状态接受状态以及边上的标签。

定义 2.3 (带标签和初始接受状态的 d-VASS) 给定一个有限的字母表 Σ，一个 d 维带状态向量加法系统 V 是一个六元组 $V = (\Sigma, Q, T, A, q_0, F)$，其中 Q 是有限的状态集合，$A \subseteq \mathbb{Z}^d$ 是动作集合，q_0 是一个初始状态，$F \subseteq Q$ 是接受状态集合，$T \subseteq Q \times \Sigma \times A \times Q$ 是一个迁移规则集合。

称 $q_0(\mathbf{0}_d)$ 为其初始格局，一个接收格局 $q(\mathbf{u})$ 满足 $q \in F$, $\mathbf{u} \in \mathbb{N}^d$。给定一条路径 π，称 $\omega(\pi) \in \Sigma^*$ 是这条路径的对应串，即这些迁移规则的标签组成的字符串。一个到达接收格局的路径被称为接收路径。给定一个格局 $\mathbf{c}$，称由其定义的语言 $\mathcal{L}(V, \mathbf{c})$ 是所有由 $\mathbf{c}$ 出发的接收路径所对应的串的集合，即 $\mathcal{L}(V, \mathbf{c}) = \{\omega(\pi) | \mathbf{c} \xrightarrow{\pi} q(\mathbf{v}) \wedge q \in F\}$。特别的，如果 $\mathbf{c}$ 是初始格局 $q_0(\mathbf{0})$，称 $\mathcal{L}(V, \mathbf{c})$ 是由向量加法系统 V 定义的语言，并简写为 $\mathcal{L}(V)$。如果对于一个串 $\omega \in \Sigma^*$，从初始格局至多只有一条路径能够通向接收格局，则称这样的向量加法系统是无歧义的 (unambiguous)。

注 一般来说，当不需要考虑其定义的语言时，定义2.2已经足够使用，所以在后文不涉及到关注其语言方面的时候，本文还是采用较为简单的定义2.2。

例 2.5 依旧考虑图2–1中的向量加法系统，令字母表为 $\Sigma = a, b$，补充定义初始状态为 q_1，接收状态集合为 $F = \{q_3\}$。将 t_1, t_4 赋标签 a，将 t_2, t_3 赋标签 b。则可

以看到 aba 是该向量加法系统接收的一个语言，而 abb 则不是。进一步来说，该向量加法系统所接收的语言是没有连续的 a 出现，并且以 a 结尾，连续出现 b 的个数之和不超过前面出现 a 的次数多 1 的这样的串。

注　关于其接收的语言，还有如下几点补充说明：

1. 可以看到例子2.5中定义出来的语言不是正则语言。
2. 向量加法系统无法接收形如 $\{a^n b^n | n \geq 1\}$ 的前缀闭包这样的语言。
3. 由于不带状态的向量加法系统可以视作只有一个状态，因此也可以定义在其上接收的语言。

2.1.3.1　带状态的向量加法系统上的验证问题

现在来介绍一下带状态的向量加法系统上的验证问题。事实上，对于格局 $q(\mathbf{u})$，可以定义其上的序关系 $(Q \times \mathbb{N}^d, \leq)$，即如果两个格局 $\mathbf{c}_1 = q_1(\mathbf{u})$, $\mathbf{c}_2 = q_2(\mathbf{v})$ 满足 $q_1 = q_2$, $\mathbf{u} \leq \mathbf{v}$，则称 $\mathbf{c}_1 \leq \mathbf{c}_2$。基于此，我们可以几乎一致的定义其上的可达性问题，可覆盖性问题，有界性问题，可达集包含问题。特别的，还可以定义其上语言接收的全集性问题 (universality)。

问题 2.5 (可达性问题 (reachability))　给定一个向量加法系统 $V = (Q, T, A)$ 和两个格局 $\mathbf{c}_1$, $\mathbf{c}_2$，问格局 $\mathbf{c}_2$ 对于格局 $\mathbf{c}_1$ 是否是可达的?

问题 2.6 (可覆盖性问题 (coverability))　给定一个向量加法系统 $V = (Q, T, A)$ 和两个格局 $\mathbf{c}_1$, $\mathbf{c}_2$，问是否存在一个格局 $\mathbf{c}'$ 满足 $\mathbf{c} \leq \mathbf{c}'$，使得格局 $\mathbf{c}'$ 对于格局 $\mathbf{c}_1$ 是可达的?

问题 2.7 (有界性问题 (boundedness))　给定一个向量加法系统 $V = (Q, T, A)$ 和一个格局 $\mathbf{c}$，问格局 $\mathbf{c}$ 的可达集是否是有限的?

问题 2.8 (可达集包含问题 (inclusion))　给定两个个向量加法系统 V 和两个格局 $\mathbf{c}_1, \mathbf{c}_2$，问格局 $\mathbf{c}_1$ 的可达集是否是格局 $\mathbf{c}_2$ 的可达集的子集?

问题 2.9 (全集问题 (universality))　给定向量加法系统 $V = (\Sigma, Q, T, A, q_0, F)$，问由 V 定义的语言是否是全集，即 $\mathcal{L}(V)$ 是否等于 Σ^*?

有意思的是，尽管带状态的向量加法系统多了状态来控制迁移规则，但是在不固定维度的时候这两者的问题复杂度是一样的，关于这一点将在下一小节进行说明。而关于其语言问题，在 2021 年全集问题[104] 被证明是 **Ackermann** 完全的，同时对于无歧义的向量加法系统，该问题是 **EXPSPACE** 完备的。

2.1.3.2 和不带状态的向量加法系统的关系

在这一节将介绍带状态的向量加法系统和不带状态的向量加法系统之间的关系。事实上只需要 3 个额外的维度就可以模拟其状态，即 $d+3$ 维的向量加法系统可以模拟 d 维的向量加法系统，这一证明首先由 hopcroft 在 1976 年给出[40]。其基本思想是，通过多出来的三维来模拟状态，即前 d 维依旧是正常的向量的迁移，而后面三维只有可能有有限种情况，用来模拟所处在的状态。具体的构造以及证明如下：

命题 2.1 (Hoprcroft[40]) 给定一个 d 维带状态的向量加法系统 $V=(Q,T,A)$，存在一个 $d+3$ 维的不带状态的向量加法系统 $V'=A_{V'}$ 和一个函数 $f: Q\times\mathbb{N}^d\to\mathbb{N}^{d+3}$，使得如果格局 $\mathbf{c}_2$ 相对于格局 $\mathbf{c}_1$ 在 V 是可达的当且仅当格局 $f(\mathbf{c}_2)$ 相对于格局 $f(\mathbf{c}_1)$ 在 V' 是可达的。

证明 令 $Q=\{q_1,\ldots,q_m\}$，$T=\{t_1,\ldots,t_m\}$，其中 $t_m=(q_{i_m},\mathbf{a}_m,q_{o_m})$，$A=\{\mathbf{a}_1,\ldots\mathbf{a}_m\}$。我们来构造对应的 V' 以及函数 f。定义如下两个有限数列 $\{a_k|k\in[n]\}$ 和 $\{b_k|k\in[n]\}$，其中 $a_k=k$，$b_k=(n+1)(n+1-k)$。函数 f 定义成 $f(q_i(\mathbf{u}))=(\mathbf{u},a_i,b_i,0)$。对于 V 中的一条规则 $t=(q_i,\mathbf{a},q_j)$，在对应的 $A_{V'}$ 中添加如下三条规则：

- $\mathbf{t}_{i_1}=(\mathbf{0}_d,-a_i,a_{n+1-i}-b_i,b_{n-i+1})$，
- $\mathbf{t}_{i_2}=(\mathbf{0}_d,b_i,-a_{n+1-i},a_i-b_{n-i+1})$，
- $\mathbf{t}_{ij}=(\mathbf{a},a_j-b_i,b_j,-a_i)$。

则在 V 中 $q_i(\mathbf{u})\xrightarrow{t}q_j(\mathbf{u}+\mathbf{a})$ 可以在 V' 中由如下的路径模拟：

$$(\mathbf{u},a_i,b_i,0)\xrightarrow{\mathbf{t}_1}(\mathbf{u},0,a_{n+1-i},b_{n+1-i})\xrightarrow{\mathbf{t}_2}(\mathbf{u},b_i,0,a_i)\xrightarrow{\mathbf{t}_3}(\mathbf{u}+\mathbf{a},a_j,b_j,0),\quad(2\text{–}8)$$

即 $f(q_i(\mathbf{u}))\xrightarrow{V'^*}f(q_j(\mathbf{u}+\mathbf{a}))$。接下来只需说明 V' 不会产生新的路径。称 V' 中存在三类迁移规则，分别代表了上述的第一第二和第三条。一个形如 $\mathbf{v}=(\mathbf{u},a_i,b_i,0)$ 的格局在 V 中只能触发迁移规则 $\mathbf{t}_{i_1}=(\mathbf{0}_d,-a_i,a_{n+1-i}-b_i,b_{n-i+1})$，原因如下：

- 在第二类规则 $\mathbf{t}_2$ 中，$\mathbf{t}_2[d+3]<a_n-b_n<0$，因此 $\mathbf{v}+\mathbf{t}_2\notin\mathbb{N}^{d+3}$。
- 在第三类规则 $\mathbf{t}_3$ 中，$\mathbf{t}_3[d+3]<0$，因此 $\mathbf{v}$ 也不能触发第三类规则。
- 对于第一类规则中的其他规则，比如 $\mathbf{t}_{j_1}=(\mathbf{0}_d,-a_j,a_{n+1-j}-b_j,b_{n-j+1})$，我们分成两类情况讨论。如果 $i>j$，则有 $(\mathbf{v}+\mathbf{t}_{j_1})[d+2]=b_i+a_{n+1-j}-b_j<0$；如果 $i<j$，则有 $(\mathbf{v}+\mathbf{t}_{j_1})[d+1]=a_i-a_j<0$，因此 $\mathbf{v}$ 不能触发。

同样的原因，我们能证明在格局 $\mathbf{v}$ 触发迁移规则 $\mathbf{t}_{i_1}$ 后得到的格局 $\mathbf{v}'=(\mathbf{u},0,a_{n-i+1},b_{n-i+1})$ 只能触发迁移规则 $\mathbf{t}_{i_2}=(\mathbf{0}_d,b_i,-a_{n+1-i},a_i-b_{n-i+1})$ 变成格局

$\mathbf{v}'' = (\mathbf{u}, b_i, 0, a_i)$。并且仅当 $\mathbf{u} + \mathbf{a} \in \mathbb{N}^d$ 的情况下，格局 $\mathbf{v}''$ 能触发规则 $\mathbf{t}_{ij} = (\mathbf{a}, a_j - b_i, b_j, -a_i)$，由此完成了该命题的证明。 □

注　可以说明这是一个紧的构造，这里有一个很好的说法，3 维带状态的向量加法系统的可达集不一定是一个半线性集 (semi-linear set)，而 5 维不带状态的向量加法系统的可达集则被证明了是半线性集[40]，但这都说明，在将维度作为参数的前提下，两者的可达性问题，可覆盖性问题，有界性问题，可达集的包含问题的难度是一致的，在后文无特别说明，我们不会区分其在不固定维度的时候的模型。

最后再用一个例子说明一下命题2.1。

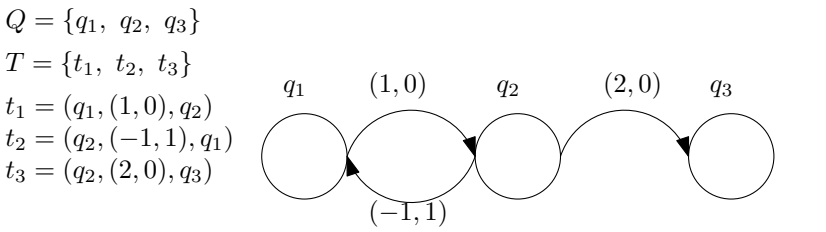


图 2–2 VAS 与 VASS 转化的例子

Figure 2–2 An Example of 2-VASS simulated by 5-VAS

例 2.6　如图2–2是一个 2 维带状态向量加法系统 V(上半部分) 被 5 维不带向量加法系统 V' 模拟的例子 (下半部分)。在 V 中一个可达的路径 $q_2(1,1) \xrightarrow{t_2} q_1(0,2)$ 可以被 V' 中如下一条路径唯一的模拟:

$$(1,1,2,8,0) \xrightarrow{t_{2_1}} (1,1,0,2,8) \xrightarrow{t_{2_2}} (1,1,8,0,2) \xrightarrow{t_{2^1}} (0,1,1,12,0)$$

2.1.4　和 Petri 网的关系

这一小节将简单介绍下 Petri 网，以及向量加法系统是 Petri 网一更为抽象的数学模型这一事实。Petri 网作为并发程序验证里面最重要的模型之一，于 1962 年被提出[119]，下面给出其形式化定义:

定义 2.4 (Petri 网)　一个 Petri 网是一个五元组 $N = (P, T, F, Pre, Post)$，其中:

- P, T 分别是一个有限的库所 (place) 集合和一个有限的迁移规则 (transition) 集合。
- $F \subseteq Q \times P \cup P \times Q$ 是一个有限的边的集合。
- $Pre,\ Post : T \to \mathbb{N}^P$ 是一个将迁移规则映射到库所的多重集的映射函数。

我们将库所里的元素称为令牌 (Token)，Petri 网的一个格局 $\mathbf{m}$ 是一个库所的多重集，库所的数量代表了在该库所里令牌的个数。用 $\mathbf{m}(p)$ 表示库所 p 中的令牌数，以此来定义 Petri 网上的一步可达关系 $\xrightarrow{T}$。如果对于两个格局 $\mathbf{m}_1, \mathbf{m}_2$ 存在一条迁移规则 $t \in T$ 满足 $Pre(t) \subseteq \mathbf{m}_1$ 和 $\mathbf{m}_2 = \mathbf{m}_1 - Pre(t) + Post(t)$，则称 $\mathbf{m}_1 \xrightarrow{T} \mathbf{m}_2$。而可达关系 $\xrightarrow{N}$ 则是 $\xrightarrow{T}$ 上的自反传递闭包。

例 2.7 如图2–3是一个有 3 个库所和 1 个迁移规则的 Petri 网。边上的赋值表示 $Pre, Post$ 对应的库所在多重集中的个数，例如 $Post(t_1) = \{p_3, p_3\}$。考虑多重集 $\mathbf{m}_1 = \{p_1, p_1, p_1, p_2, p_2, p_2, p_3\}$ 是其上的一个格局，它能触发迁移规则 t_1 至格局 $\mathbf{m}_2 = \{p_1, p_3, p_3, p_3\}$。

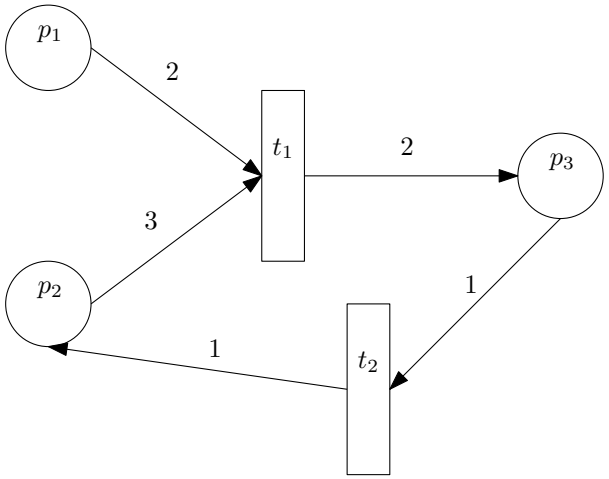


图 2–3 一个 Petri 网的例子

Figure 2–3 An Example of Petri net

我们简单叙述下为什么可以用向量加法系统来模拟 Petri 网。事实上，假设 Petri 网有 n 个库所，则可以用一个 n 维向量加法系统来模拟，第 i 维代表着第 i 个库所里存有的令牌数，而每次迁移规则则就可以用一个 n 维向量来表示，因此可以用一个 n 维向量加法系统来模拟 Petri 网的运行，即：

命题 2.2 给定一个有 k 个库所的 Petri 网 $N = (P, T, F, Pre, Post)$，存在一个 n 维向量加法系统 V 以及一个函数 $f : P^{\mathbb{N}} \to \mathbb{N}^k$，对于 N 上的两个格局 $\mathbf{m}_1$，$\mathbf{m}_2$，$\mathbf{m}_1 \xrightarrow{N} \mathbf{m}_2$ 当且仅当 $f(\mathbf{m}_1) \xrightarrow{V} f(\mathbf{m}_2)$。

2.2　良拟序理论基础介绍

本节将介绍一些基本的良拟序理论，其在后面的验证问题证明中起到了重要的作用。首先回顾一下关系上的基本性质。给定集合 X 上的一个序关系 $R = (X, \leq)$，常见的性质有：

- 自反性 (reflexive)，即对于 $x \in X$，有 xRx。
- 对称性 (symmetric)，即对于 $x, y \in X$，如果有 xRy，则有 yRx。
- 传递性 (transitive)，即对于 $x, y, z \in X$，如果有 xRy, yRz，则有 xRz。
- 反对称性 (antisymmetric)，即对于 $x, y \in X$，如果有 xRy, yRx，则有 $x = y$。

如果一个关系是自反对称传递的，则称这个关系是等价关系 (equivalence relation)。如果一个序关系是自反传递反对称的，则称这是一个偏序关系 (partial order)；如果任两个元素 $x, y \in X$，都有 xRy 或者 yRx，并且其是自反传递反对称的，则称这是一个全序关系 (total order)。对于 $x, y \in X$，如果存在 xRy 或者 yRx，则称 x 和 y 是可比较的，否则称之为不可比较的。如果一个序关系是自反传递的，则称这是一个拟序关系 (quasi order)。令 $(X, \leq)$ 是 X 上的一个拟序关系，则 $(X, \geq)$ 定义为 $(X, \geq) = \{(x, y) | y \leq x,\ x, y \in X\}$，则 $(X, \leq) \cap (X, \geq)$ 是一个等价关系。特别在如果对于集合 X 没有异议的情况下，用 $\leq$ 和 $\geq$ 来简写。

下面来介绍良拟序 (well quasi order) 的概念。

定义 2.5 (well quasi order)　给定一个集合 X 和其上的一个拟序关系 $(X, \leq)$。如果对于 X 上任何一个无穷序列 $s_1, s_2, \ldots, s_n, \ldots$，都存在一个升序对 (i, j), $i < j$, $i, j \in \mathbb{N}$ 满足 $s_i \leq s_j$，则称 $\leq$ 是一个良拟序。

例 2.8　在自然数 $\mathbb{N}$ 上的 $\leq$ 关系是一个良拟序；而在整数 $\mathbb{Z}$ 上的 $\leq$ 关系则不是，它存在一个无穷递降的序列 $0, -1, -2, -3, \ldots$。

良拟序有着很多很多等价的定义，下面还列举了一些不同的良拟序的定义，给定 X 上的一个序关系 $(X, \leq)$，如果满足下列条件之一，则 $\leq$ 是一个良拟序：

- 对一个无穷序列 $s_1, s_2, \ldots, s_n, \ldots$ 都存在一个无穷递增子序列 $s_{i_1}, s_{i_2}, \ldots$，即对于 $k < l$，有 $s_{i_k} < s_{i_l}$。
- X 上不存在无穷多个不能互相比较的元素并且不存在一个无穷递降的序列。

特别的，如果 X 上的序关系只满足不存在一个无穷递降的序列，则称这个序列是良基序 (well founded)。最后我们再简单介绍一下如何构造一个良拟序。给定两个序关系 $(A, \leq_A)$ 和 $(B, \leq_B)$，扩展定义如下两个新的序关系：

- $(A \times B, \leq_{A \times B})$，即 $(x_1, y_1) \leq_{A \times B} (x_2, y_2)$ 当且仅当 $x_1 \leq_A x_2$, $y_1 \leq_B y_2$。

- $(A^*, \leq_{A^*})$，即 A 上的两个串 $s = x_1x_2 \ldots x_n$ 和 $s' = y_1y_2 \ldots y_m$ 满足 $s \leq_{A^*} s'$ 当且仅当存在 $1 \leq i_1 < i_2 \ldots < i_n \leq m$ 满足 $x_k \leq_A y_{i_k}$。

Dickson 引理和 Higman 引理告诉了我们如果原本的序关系是一个良拟序，则新产生的这两种序关系也是良拟序。

引理 2.1 (Dickson 引理 [120]) 给定两个良拟序 $(A, \leq_A)$ 和 $(B, \leq_B)$，则 $(A \times B, \leq_{A\times B})$ 是一个良拟序。

引理 2.2 (Higman 引理 [121]) 给定一个良拟序 $(A, \leq_A)$，则 $(A^*, \leq_{A^*})$ 是一个良拟序。

2.3 快速增长层级复杂性介绍

在向量加法系统的验证问题中，很多问题的复杂性都超过了人们的想象，因此我们需要介绍一些复杂度更高的复杂性类来描述这些问题。Schmitz 在 [86] 介绍了一类快速增长层级复杂性类，本文将在此节做一个基本的介绍。为此首先需要借助一些序数 (ordinal)[122] 的概念。

2.3.1 序数介绍

序数可以视作对自然数的一种拓展，一般用良序集来表示。这里一个良序集指的是对于一个集合 X 和其上的一个序关系 $\leq$ 满足以下三个条件：

- 三分性 (trichotomy)，即对于任何两个元素 x, y，$x < y$, $x = y$, $x > y$ 之一必然成立。
- 传递性，即如果 $x < y$, $y < z$，就有 $x < z$。
- 良基性，即不存在无穷递降的序列。

例 2.9 我们来介绍 von Neumann 序数[123]，即一个序数是所有比它更小的序数组成的良序集。具体来说，定义

$$0 \stackrel{\text{def}}{=} \{\}, \ 1 \stackrel{\text{def}}{=} \{0\} = \{\{\}\}, \ 2 \stackrel{\text{def}}{=} \{0, 1\} = \{\{\}, \{\{\}\}\}, \ldots \tag{2–9}$$

其上的序关系定义为集合的属于 $\in$ 关系，显然该关系满足上述三个性质。

一个后继序数 (successor ordinal)，记作 $\alpha + 1$，指的是下一个序数。在例子2.9中则是指 $\alpha \cup \{\alpha\}$。如果一个序数不是 0 也不是一个后继序数，则称其为极限序数 (limit ordinal)，其可被视作一个良序集的上确界，或者说，对于一个极限序数 γ，其对

于任何一个序数 $\beta < \gamma$，都有 $\beta + 1 < \gamma$。ω 记为最小的极限序数，它可以视作所有自然数的上确界，下一个极限序数则是 $\omega \cdot 2$。

任何一个序数都能表示成康托尔范式 (Cantor Normal Form,CNF)，即对于序数 α，其可以表示为下列形式：

$$\alpha = \omega^{\alpha_1} \cdot c_1 + \omega^{\alpha_2} \cdot c_2 + \cdots + \omega^{\alpha_n} \cdot c_n \tag{2–10}$$

其中 c_i, $i \in [n]$ 是正整数，$\alpha \geq \alpha_1 > \alpha_2 > \cdots > \alpha_n \geq 0$ 是一串序数。特别的，在本文中只关心能被满足 $\alpha > \alpha_1$ 这样的康托尔范式表示的序数。令 ϵ_0 表示满足 $\omega^x = x$ 的序数，显然 ϵ_0 是上述形式能表示的范数的上确界。

最后介绍基本序列 (fundamental sequence) 的概念。对于一个极限序数 λ，其基本序列 $\{\lambda(x)_{x<\omega}\}$ 由下面的方式递归定义如下：

$$(\lambda + \omega^{\beta+1})(x) \overset{\text{def}}{=} \lambda + \omega^{\beta}(x+1) \qquad (\lambda + \omega^{\lambda})(x) \overset{\text{def}}{=} \lambda + \omega^{\lambda(x)}. \tag{2–11}$$

比如 $\omega(x) = x + 1$。

2.3.2　快速增长层级复杂性

在有了序数的概念以后，我们可以定义一系列以序数为下标的快速增长函数。这里以 Grzegorczyk 层级快速增长函数[124-125] 来定义需用到的一些复杂性类。定义这样一类以序数为下标的函数 $\{F_\alpha\}_{\alpha<\epsilon_0}$：$F_0, F_1, F_2, F_3, \ldots, F_\omega, \ldots : \mathbb{N} \to \mathbb{N}$ 满足：

- $F_0(x) = x + 1$。
- $F_{\alpha+1}(x) = F_\alpha^{\omega(x)}(x)$。
- $F_\lambda(x) = F_{\lambda(x)}(x)$。

简单计算可知，$F_1(x) = 2x + 1$, $F_2(x) = 2^{x+1}(x+1) - 1$，而 F_3 则是一个非初等函数，它要比塔 (Tower) 函数 $\text{Tower}(x) = 2^{\overset{\cdot^{\cdot^{\cdot^{2}}}}{\}x}}$ 还要大。对于 $\alpha < \omega$，F_α 是一个原始递归函数 (primitive recursive)，而当 $\alpha = \omega$ 的时候，F_α 是著名的 Ackermann 函数[126]，即是一个可计算但不是原始递归的函数。

我们可以用这类函数定义一些被有限资源限制的可计算函数[127]。对于 $\alpha \geq 2$ 来说，$\mathscr{F}_\alpha$ 表示确定图灵机 (deterministic Turing machine) 对于输入大小为 n 的输入能在 $O(F_\alpha^c(x))$ 计算出来的函数集合，其中 $c \in \mathbb{N}$ 是一个常数，即：

$$\mathscr{F}_\alpha \overset{\text{def}}{=} \bigcup_{c<\omega} \mathbf{FDTIME}(F_\alpha^c(n)). \tag{2–12}$$

特别的，定义 $\mathscr{F}_{<\alpha} = \bigcup_{\beta<\alpha} \mathscr{F}_\beta$。$\mathscr{F}_\alpha$ 有些很好的性质，比如它在复合运算下是封闭的，并且每个在里面的函数都可以在被其中的一个函数限制内的时间算出来[127-128]等。

据此可以定义一类可判定问题的复杂性类 $\mathbf{F}_\alpha$[86]，即本节开始所说的快速增长复杂性类。该复杂性类定义如下：

$$\mathbf{F}_\alpha \stackrel{\text{def}}{=} \bigcup_{p\in\mathscr{F}_{<\alpha}} \mathbf{DTIME}(F_\alpha(p(n))) \tag{2–13}$$

这里 **DTIME** 表示确定图灵机对于一个判定问题所用的时间。可以发现，如果要说明一个问题是 $\mathbf{F}_\alpha$-难的，我们可以用一个规模为 $\mathscr{F}_{<\alpha}$ 的规约去证明。

$\mathbf{F}_\alpha$ 定义了许多著名的复杂性类，比如 $\mathbf{F}_3 \stackrel{\text{def}}{=} \textsc{Tower}$ 是一个在初等函数下封闭的复杂性类；$\mathbf{F}_\omega \stackrel{\text{def}}{=} \textsc{Ack}$ 是向量加法系统可达性所在的复杂性类，其可以使用原始递归函数规模的规约；$\mathbf{F}_{\omega^\omega} \stackrel{\text{def}}{=} \textsc{HAck}$ 是一些向量加法系统扩展模型上验证问题所处的复杂性类等等。

最后介绍相对快速增长复杂性类。令 $h : \mathbb{N} \to \mathbb{N}$ 是一个单调递增的函数，定义基于 h 的函数类 $F_{h,\alpha}$：

$$F_{h,0}(x) \stackrel{\text{def}}{=} h(x),\ F_{h,\alpha+1}(x) \stackrel{\text{def}}{=} F^{\omega(x)}_{h,\alpha}(x),\ F_{h,\lambda}(x) \stackrel{\text{def}}{=} F_{h,\lambda(x)}(x) \tag{2–14}$$

定义复杂性类：$\mathbf{F}_{h,\alpha}$：

$$\mathbf{F}_{h,\alpha} \stackrel{\text{def}}{=} \bigcup_{p\in\mathscr{F}_{<\alpha}} \mathbf{DTIME}(F_{h,\alpha}(p(n))) \tag{2–15}$$

当 $h : \mathbb{N} \to \mathbb{N}$ 是初等时间内可构造的严格单调递增函数时，也可以在初等时间内构造 $F_{h,\alpha}$，Schmitz 在在 [86] 中证明了该定理：

定理 2.1 (Schmitz[86]) 令 $h : \mathbb{N} \to \mathbb{N}$ 是个严格单调增函数和 α 是个序数，如果 h 是初等时间内可构造的，则 $F_{h,\alpha}$ 也是初等时间内可构造的。

当 h 是严格单调递增的时候，有 $\mathbf{F}_\alpha \subseteq \mathbf{F}_{h,\alpha}$，Schmitz 在 [86] 中证明了如下定理，给出了其关系更精确地表述：

定理 2.2 (Schmitz[86]) 令 $h : \mathbb{N} \to \mathbb{N}$ 是个严格单调增函数，α, β 是两个序数，则有：

- $h \in \mathscr{F}_\beta$ 蕴含 $\mathbf{F}_{h,\alpha} \subseteq \mathbf{F}_{\beta+1+\alpha}$。
- $h \leq \mathbf{F}_\beta$ 蕴含 $\mathbf{F}_{h,\alpha} \subseteq \mathbf{F}_{\beta+\alpha}$。

2.3.3　基于序数的长度函数控制定理

本节将介绍证明算法上界的一类方法：基于序数的长度函数控制定理[129]。首先来回顾一下良拟序的概念，不难得到，对于任何一个良拟序，其不存在无限长的降序序列，并且称降序序列为坏序列 (bad sequence)。考虑一个简单的例子 $(\mathbb{N}, \leq)$，如下的序列显然是一个坏序列：

$$n, n-1, n-2, \ldots, 1, 0 \tag{2–16}$$

不难得出，在 $(\mathbb{N}, \leq)$ 中以 n 为开头的坏序列的长度至多为 $n+1$。但是当选择的良拟序越来越复杂时，坏序列的长度便会变得越来越难以估计，比如考虑 $(\mathbb{N}^2, \leq_{lex})$ 中的一个序列：

$$(1,0), (0,n), (0,n-1), \ldots, (0,1), (0,0) \tag{2–17}$$

以 $(1,0)$ 为开头的坏序列长度则可以有 $n+2$，因此需要一种方式来计算坏序列的长度。首先我们需要对序列作一定的限制，因为如果不加任何限制的话，比如在序列2–17中 $(1,0)$ 能接上任意的 $(0,m)$，也就有着任意长的坏序列长度。为此需要介绍控制函数 (control function) 和对应的长度函数 (length function)。

给定一个良拟序 $(X, \leq)$，定义在 X 上的范数 $|.|_X X \to \mathbb{N}$ 满足：对任意的 $n \in \mathbb{N}$ 满足范数 $\leq n$ 的 X 的元素是有限的。令 $g : \mathbb{N} \to \mathbb{N}$ 是一个函数是单调且可膨胀的 (montone and expansive)，即对任意的 $x \leq x' \in \mathbb{N}$ 有 $g(x) \leq g(x')$, $x \leq g(x)$，则对于 X 上的任何一个序列 $x_0, x_1, \ldots$，如果其满足 $|x_i|_X \leq g^i(n_0)$，则称该序列是 (g, n_0)-控制的，并且令 $L_{g,X}(n)$ 表示所有 (g,n)-控制的序列中，坏序列的最大长度。

注意到可以用序数来标记任何一个良序的集合 (用 0 表示最小元素，1 往后，依次类推)，我们称未用到的最小序数 α 为该集合的序数类型，接下来令 X 的序数类型为 α，下面给出关于某些控制函数 g 的 $L_{g,X}(n)$ 的界。首先介绍两个同样用到序数定义的层级函数：Hardy 层级 $\{h^\alpha\}_{\alpha<\epsilon_0}$ 和 Cichoń 层级 $\{h_\alpha\}_{\alpha<\epsilon_0}$[130]，其定义如下：

- $h^0(x) = x$, $h^{\alpha+1} = h(h^\alpha(x))$, $h^\lambda(x) = h^{\lambda(x)}(x)$。
- $h_0(x) = 0$, $h_{\alpha+1} = 1 + h(h_\alpha(x))$, $h_\lambda(x) = h_{\lambda(x)}(x)$。

这里同样可以用 Cichoń 层级和 Hardy 层级来定义上述的复杂性类，令 $h(x) = H(x) = x+1$，则有:$H^{H_{\omega^\alpha}(x)}(x) + x = H^{\omega^\alpha}(x) = F_\alpha(x)$，因此可以同样定义复杂性类 $\mathbf{F}_\alpha$:

$$\mathbf{F}_\alpha \stackrel{\text{def}}{=} \bigcup_{p \in \mathcal{F}_{<\alpha}} \mathbf{DTIME}(H^{\omega^\alpha}(p(n))) \tag{2–18}$$

如果 h 是单调可膨胀的，则有 h_α, h^α 都是单调可膨胀的[130-132]。但是序数之间的大小却不能表示函数之间的大小。比如有 $H^\omega(x) = x+1 \leq H^{x+2}(x) = x+2$，但是 $\omega > x+2$。为了保持序数间的单调性，对于任意 x 定义序关系 $\prec_x$[131] 为 $\alpha \prec_x \alpha+1$, $\lambda(x) \prec_x \lambda$ 的传递闭包，为了方便有时将 $\alpha \prec_x \beta$ 记作 $\alpha \in \beta[x]$，则有：

- $\prec_0 \subseteq \prec_1 \subseteq \cdots \subseteq \prec_x \subseteq \cdots \subseteq \leq$。
- $\alpha \prec_x \beta$ 蕴含了 $h_\alpha(x) \leq h_\beta(x)$。

然后来介绍 $L_{g,X}(n)$ 的界。对于一个序数 $\alpha = \omega^{\alpha_1} \cdot c_1 + \omega^{\alpha_2} \cdot c_2 + \cdots + \omega^{\alpha_n} \cdot c_n$，定义其范数 $N\alpha$ 为：

$$N\alpha \stackrel{\text{def}}{=} \max\{c_1, \ldots c_n,\ N\alpha_1, \ldots N\alpha_n\} \tag{2–19}$$

则下述定理说明 (g,n)-控制的坏序列的最大长度大约是 $g_\alpha(n)$。

定理 2.3 (Schmitz[129]) 令 $\alpha < \epsilon_0$ 并且满足 $N\alpha \leq n$，则在良拟序关系 $(\alpha, \leq)$ 上的最长坏序列的长度为 $L_{g,\alpha}(n) = g_\alpha(n)$。

2.4 线性方程组介绍

本节将介绍一些线性方程组[133] 上非零整数解上的性质。首先补充说明一些基本的标记。令 $\mathbf{x}_1, \mathbf{x}_2 \ldots \mathbf{x}_n$ 为向量空间 $\mathbb{V}$ 上的一组向量，如果存在一组非零整数 $m_1, m_2, \ldots, m_n \in \mathbb{Z}$ 满足：

$$m_1\mathbf{x}_1 + m_2\mathbf{x}_2 + \ldots m_n\mathbf{x}_n = \mathbf{0} \tag{2–20}$$

则称这组向量是线性相关的，否则称其是线性无关的。一个向量组的秩 (rank) 定义为其中极大线性无关向量的个数。令 $\mathrm{A} = \{a_{ij}\}_{m\times k}$ 是一个 $m \times k$ 的矩阵，其列向量组定义为 $\{\mathbf{a}_i = (a_{1i}, a_{2i}, \ldots a_{mi})\}_k$ 矩阵上的秩定义为其列向量组的的秩。下面来定义矩阵 A 的范数，用向量的 p-范数2–2来定义矩阵的 p-范数 $||\mathrm{A}||_p$，即：

$$||A||_p = \sup_{\mathbf{x}\neq\mathbf{0}} \frac{||\mathrm{A}\mathbf{x}||_p}{\mathbf{x}_p} \tag{2–21}$$

例 2.10 计算可得，给定一个 $m\times k$ 的矩阵 A，其 1-范数为:$||\mathrm{A}||_1 = \max_i \sum_{j=1}^m |a_{ji}|$，其 ∞-范数则为：$||\mathrm{A}||_\infty = \max_{i,j} |a_{ij}|$。

接下来介绍线性方程组的概念。一个非齐次线性方程组是形如 $\Phi : \mathrm{A}\mathbf{x} \leq \mathbf{b}$ 这样的不等式组，其中 A 是一个 $m \times k$ 的整系数矩阵，$\mathbf{b}$ 是一个在 $\mathbb{N}$ 上的 m 维向量，

其目标是求出一个 $\mathbb{N}^k$ 上的向量 $\mathbf{x}$ 满足这些不等式，注意到如果只要求 $\mathbf{x}$ 是 $\mathbb{Z}^d$ 里的向量，那么这个问题可以在被多项式时间内解决 (高斯消元法 [133])，然而求其非负整数解却是一个 **NP** 完全问题。

这个问题的研究会被转化成研究齐次线性方程组的解。一个齐次线性方程组是形如 $\Phi_0 : A\mathbf{x} = \mathbf{0}$ 这样的方程组，它的解是一个 k 维向量，考察其解集中所有极小的解，其个数是有限的，将所有这样的解组成的集合称为 Φ_0 的希尔伯特基 (Hilbert basis)[134]，记作 $\mathcal{H}(\Phi_0)$，Φ_0 的每一个解都能表示成 $\mathcal{H}(\Phi_0)$ 中的解的线性组合。Pottier 在 [135] 中证明了 $\mathcal{H}(\Phi_0)$ 中任何一个解的大小都是被方程组大小的指数限制住的，即：

定理 2.4 (Pottier[135])　令齐次线性方程组 $\Phi_0 : A\mathbf{x} = \mathbf{0}$，其中 A 是一个 $m \times k$ 的矩阵，令其秩为 r，则对于任意的 $\mathbf{m} \in \mathcal{H}(\Phi_0)$，有：

$$||\mathbf{m}|| \leq (1 + k \cdot ||A||_\infty)^r \tag{2–22}$$

而关于非齐次线性方程组 $\Phi : A\mathbf{x} = \mathbf{r}$ 的解的情况，考虑齐次线性方程组 $\Phi_0' : [A; -\mathbf{r}]\mathbf{x} = \mathbf{0}$，这里 $[A; -\mathbf{r}]$ 是一个 $m \times (k+1)$ 维矩阵，即在 A 的最后一列再加上 $\mathbf{r}$。显然 Φ_0' 中那些最后一维为 1 的解 $\mathbf{x}$ 的前 k 维是方程组 Φ 的解。记 $\mathcal{H}(\Phi_0')$ 中所有最后一维为 1 的解的前 k 维向量组成的集合为 $\mathbb{S}$，Φ 的任何一个解都可以由 $\mathbb{S}$ 中的一个解和 Φ_0 中的解组合而成，因此有：

推论 2.1　令齐次线性方程组 $\Phi : A\mathbf{x} = \mathbf{r}$，其中 A 是一个 $m \times k$ 的矩阵，令其秩为 r，则对于任意的 $\mathbf{m} \in \mathcal{H}(\Phi_0) \cup \mathbb{S}$，有：

$$||\mathbf{m}| \leq (1 + k \cdot ||A||_\infty + k \cdot ||\mathbf{r}||_\infty)^{r+1} \tag{2–23}$$

同样对于不等式组 $\Phi_1 : A\mathbf{x} \leq \mathbf{r}$ 来说也有如下结论：

推论 2.2　令齐次线性方程组 $\Phi_1 : A\mathbf{x} \leq \mathbf{r}$，其中 A 是一个 $m \times k$ 的矩阵，令其秩为 r，如果 Φ_1 存在非平凡的非负整数解，则存在一个解 $\mathbf{m} \in \mathbb{N}^k$ 满足：

$$||\mathbf{m}| \leq (1 + k \cdot ||A||_\infty + k \cdot ||\mathbf{r}||_\infty)^{r+1} \tag{2–24}$$

第三章　向量加法系统的可覆盖性问题与有界性问题

在这一章中将介绍向量加法系统中两个 **EXPSPACE** 完备的问题，即可覆盖性问题和有界性问题。这两类验证问题相比于可达性问题而言，很早就有了完备的结果[93-94]，但是依旧有着非常广泛的应用，在模型检测[136-139] 和异步程序验证[27, 94, 140] 方面都有着重要的作用。其中用到的技术 Karp-Miller 树对于其他模型上比如分支向量加法系统[52] 和良结构转移系统 (well structured transistion system,WSTS)[91-92] 上的验证问题也发挥了重大的效果。有界性问题也有着很多的变种，可逆有界性 (reversal boundedness)[98-99] 是一个允许某些维度在非升或者非降的变化之间切换，其有助于线性时间逻辑 (linear temporal logic) 检测上的一些方法[100]；状态有界性 (place boundedness)[20] 则研究状态的有界性。这些问题可以定义成一类选择无界性 (selective unboundedness)，在 [101] 有详细的介绍。

首先回顾一下两个问题的定义。

问题 3.1 (可覆盖性问题 (coverability))　给定一个向量加法系统 V 和两个格局（向量）$\mathbf{u}$, $\mathbf{v}$，问是否存在一个格局 $\mathbf{v}' \geq \mathbf{v}$，使得格局 $\mathbf{v}$ 对于格局 $\mathbf{u}$ 是可达的?

问题 3.2 (有界性问题 (boundedness))　给定一个向量加法系统 V 和一个格局（向量）$\mathbf{u}$，问格局 $\mathbf{u}$ 的可达集是否是有限的?

更直观的来说，可覆盖性问题问的是初始格局能不能到达一个比目标格局更大的格局；而有界性问题则问的是一个格局的可达集是不是有限的，也称某个向量加法系统对于某个格局是有界的。在本章中，首先将介绍 Karp-Miller 树这一概念，其能够帮助获得可覆盖性问题和有界性问题的可判定性结果，然后将介绍这两种问题的指数空间算法，具体来说将在第二节介绍可覆盖性问题的算法，而第三节将介绍有界性问题的算法，这两个问题的指数空间算法是由 Rackoff 提出的[93]。而关于这两个问题 **EXPSAPACE** 下界的证明，鉴于其是 Lipton 证明可达性问题是 **EXPSPACE** 难[89] 的这一结论的直接推论，将在章节六作详细的介绍。在不做特别说明的情况下，本章固定一个 d 维向量加法系统 $V = \mathbf{A}_v$，大小为 n，对于可覆盖性问题还固定两个格局 $\mathbf{u}$, $\mathbf{v}$，其中 $\mathbf{u}$ 初始格局，$\mathbf{v}$ 是目标格局，而对于有界性问题则固定一个初始格局 $\mathbf{u}$。

3.1　Karp-Miller 树

本节将介绍 Karp-Miller 树[20]。直观上来说，Karp-Miller 树是一颗模拟着向量加法系统从某个格局开始所有运行的树，正常来说这样的树是会无限大的，但是 Karp-Miller 树最后却能被证明是有限的。其中的关键点是如果一个格局 $\mathbf{w}$ 经过了某个路径到达了格局 $\mathbf{w}'$ 满足 $\Delta_w = \mathbf{w}' - \mathbf{w} \geq \mathbf{0}_d$，那么对于所有 Δ_w 里大于 0 的维度来说，格局可以一直重复走这条路径使其变大，在 Karp-Miller 树中就用 ω 来表示这种维度，从而可以在这条路径上忽略这些维度产生的变化。换句话说在 Karp-Miller 树中使用 $\mathbb{N}_\omega^d$ 里的向量来描述 V 上一个格局的可达格局，并且证明这样的树不会是无限大的，即对于任何一条路径它都有一个终点。

接下来首先给出 Karp-Miller 树的形式化定义。首先需要的是一些关于树结构的定义作准备。

定义 3.1 (树)　一个树 (Tree) 是一类特殊的有向图 $T = (N, E, v_0)$，其中 N 是顶点集合，$E \subseteq N \times N$ 是有向边的集合，满足如下三点：

- v_0 被称作根节点 (root)，其入度为 0，即没有任何一条边指向它。
- 对于 N 中的其他不是根节点的点，其入度为 1，即有且只有一条边指向它。
- 任何一个点 $v \in N$ 对于 v_0 都是可达的，并且存在一条唯一的 v_0 到 v 的路径。

我们再补充一些上面的术语。如果对于树 T 上的两个点 v 和 v'，如果存在一条由 v 到 v' 的路径，则称 v 是 v' 的祖先 (ancestor)，记作 $v \prec v'$，特别的如果 v 和 v' 存在一条边，则称 v' 是 v 的后继 (successor)，一个没有边出去的点我们则称为叶子节点 (leaf)。如果树上的节点是有限的，那么称这棵树是有限的 (finite)。

接下来定义 Karp-Miller 树的概念。

定义 3.2 (Karp-Miller 树 [20])　给定一个向量加法系统 $V = \mathbf{A}_V$ 和一个格局 $\mathbf{u}$，一个对应的 Karp-Miller 树是一个二元组 $K = (T_{V,\mathbf{u}}, L)$，其中 $T_{V,\mathbf{u}} = (N, E, r)$ 是一棵树，$L : N \to \mathbb{N}_\omega^d$ 则是一个将每个节点标识一个 $\mathbb{N}_\omega$ 上的 d 维向量的标识函数，K 具体可以有如下的递归构造获得：

- $L(r) \stackrel{\text{def}}{=} \mathbf{u}$。
- 令 u 是树上的一个顶点，如果存在一个其祖先 v 满足 $L(u) = L(v)$，则令 u 为叶子节点，否则根据 $\mathbf{A}_V$ 来构造其后继节点，对于每个 $\mathbf{a} \in \mathbf{A}_V$，如果 $L(u) + \mathbf{a} \geq \mathbf{0}$，则构造一个 u 的后继节点 $u_\mathbf{a}$，并按如下的方式定义 $L(u_\mathbf{a})$：

– 如果存在 $u_{\mathbf{a}}$ 的一个祖先 v 满足 $L(u)+\mathbf{a} \geq L(v)$，则定义 $L(u_{\mathbf{a}})$：

$$L(u_{\mathbf{a}})[i] \stackrel{\text{def}}{=} \begin{cases} \omega, & \text{if } (L(u)+\mathbf{a})[i] > L(v)[i] \\ (L(u)+\mathbf{a})[i], & \text{if } (L(u)+\mathbf{a})[i] = L(v)[i]. \end{cases}$$

– 如果不存在这样的祖先，则 $L(u_{\mathbf{a}}) \stackrel{\text{def}}{=} L(u)+\mathbf{a}$。

例 3.1 考虑一个 Karp-Miller 树的具体例子来理解这个定义。如图3–1是一个 3 维向量加法系统 $V = \mathbf{A}_V$ 上由格局 $(0,2,0)$ 生成的 Karp-Miller 树，每个点上的标注则是 L 对其的函数值。事实上，由 $(0,2,0)$ 出发的任何一条运行都会体现在 Karp-Miller 树上的一条路径里，并且可以看到其可达的路径前两维只有 $(1,1)(2,0)(0,2)$ 这三种情况。

有一点需要注意的是，在 V 中 $(0,2,0)$ 可达的格局里，ω 所在第三维是可以到达一切自然数的，但是在正常的情况下 ω 只能代表格局在这一维可以无限大，但不代表任意大的值都是可达的。

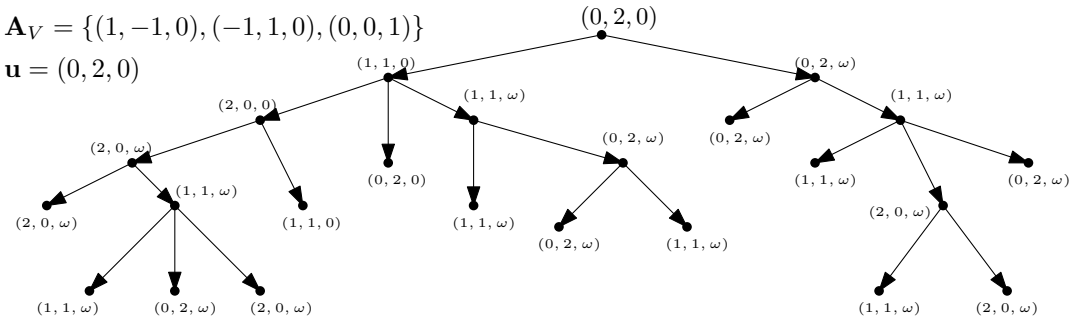


图 3–1 一个三维向量加法系统上由 $(0,2,0)$ 出发的 Karp-Miller 树的例子

Figure 3–1 An Example of Karp-MIller tree related a VAS with configuration$(0,2,0)$

接下来证明对于任何一个向量加法系统和任何一个格局，其产生的 Karp-Miller 树都是有限的。为此首先需要用一个有关树的引理。

引理 3.1 (König infinity lemma[141]) 给定一颗树 T，如果对于其中任何一个顶点 v 都只有有限个后继节点，并且从根节点出发任何一条路径都是有限长的，则这棵树 T 是有限的。

引理3.1给出了判断一棵树是否有限的方法，我们以此可以获得 Karp-Miller 树的有限性。

定理 3.1 (Karp and Miller[20])　任给定一个 d 维向量加法系统 $V = \mathbf{A}_V$ 和一个格局 $\mathbf{u}$，其产生的 Karp-Miller 树 $K = (T, L)$ 都是有限的。

证明　因为 $\mathbf{A}_V$ 是有限的，所以根据构造 K 上的每一个节点 v 的后继节点都是有限的。因此只需要说明 K 上不存在一条无限长的路径。

假设 K 上存在一条无限长的路径 $v_1, \ldots, v_n, \ldots$，显然 $(\mathbb{N}^d_\omega, \leq)$ 是一个良拟序，因此其存在一条无限长的子路径 $v_{i_1}, \ldots, v_{i_n}, \ldots$ 满足：

$$L(v_{i_1}) \leq L(v_{i_2}) \leq \cdots \leq L(v_{i_n}) \leq \cdots \tag{3–1}$$

由构造，对任意的 $j \in \mathbb{N}$，我们有 $L(v_{i_j}) \neq L(v_{i_{j+1}})$，否则路径停止。另一方面，如果存在一条路径上的两个点 u, v 满足 $L(u) < L(v)$，则依据构造 $L(v)$ 中为 ω 的维数的数量至少比 $L(v)$ 多 1，而 $L(u)$ 至多只有 d 个维度能为 ω，因此在 K 不存在一条无限长的路径，即 K 是有限的。 □

接下来将说明构造出的 Karp-Miller 树能够展现出的关于可达集的一些性质，以此来说明可覆盖性问题和有界性问题都是可判定的。具体来说，对于 V 上的任意一条路径及其经过的格局，我们都可以找到一条 Karp-Miller 树上的路径来覆盖住它，并且反过来对于 Karp-Miller 树上的一条路径，我们也可以构造出符合其标识的在 V 上的一条可达路径。因此对于可覆盖性问题来说，只需要在 Karp-Miller 树上找到一个节点的标识能够覆盖住目标节点，而对于有界性问题来说，只需要证明 Karp-Miller 树上所有节点的标识都是一个 d 维自然数向量即可。形式化的来说，该性质可被表示如下：

定理 3.2 (Karp and Miller[20])　给定一个格局 $\mathbf{u} \in \mathbb{N}^d$，对于向量加法系统 $V = \mathbf{A}_V$ 和由其和格局 $\mathbf{u}$ 产生的 Karp-Miller 树 $T = (K, L)$ 中，以下的两个叙述是等价的：

1. 存在 $\mathbf{y} \in \mathrm{Reach}_V(\mathbf{u})$ 满足 $\mathbf{x} \leq \mathbf{y}$。
2. 存在 T 中的一个节点 v 满足 $\mathbf{x} \leq L(v)$。

证明　在证明之前我们引入一个函数 $L_E : E^* \to \mathbf{A}_V^*$，这是一个树上路径的标识函数，即对于一条边 $e = (v_1, v_2)$ 来说，如果 e 是由规则 $a \in \mathbf{A}_V$ 生成的，则令 $L_E(e) = \mathbf{a}$。特别的对于边的串 $e_1 e_2 \ldots e_n$，有 $L(e_1 e_2 \ldots e_n) = L(e_1)L(e_2)\ldots L(e_n)$。

接下来先证明 (1) $\Rightarrow$ (2)。考虑 $\mathbf{u}$ 到 $\mathbf{y}$ 的最短路径 $\pi = \mathbf{a}_1 \ldots \mathbf{a}_m$，显然其不会经过两个相同的格局，然后我们从树 K 上的根节点 r 出发也寻找标识为 π 的路径，这会有两种情况：

- 存在一条树上的路径 $r v_1 v_2 \ldots v_m$ 满足 $L((r, v_1)(v_1, v_2)\ldots(v_{n-1}, v_m)) = \pi$。那由构造有 $L(v_i) \geq \mathbf{u} + \Delta(\pi[1, i])$，因此有 $L(v_n) \geq \mathbf{y} \geq \mathbf{x}$。

- 只存在一条树上的路径 $rv_1v_2\dots v_i$ 满足 $L((r,v_1)(v_1,v_2)\dots(v_{i-1},v_i)) = \pi[1,i]$。显然 v_i 是一个叶子结点，如果 $L(v_i) \geq \mathbf{y}$ 则 (2) 已满足，否则由构造存在点 v_j, $j \in [i]$ 满足 $L(v_j) = L(v_i) \geq \mathbf{u} + \Delta(\pi[1,i])$。继续以 v_j 作为起点，$\pi[i+1,m]$ 作为标识路径，寻找树 K 上的路径，重复上述过程直到标识路径全部走完，最终获得节点 v，显然有

$$L(v) \geq \mathbf{u} + \Delta(\pi) = \mathbf{y} \geq \mathbf{x}$$

最后来证明 (2) $\Rightarrow$ (1)。这里一个简单的思路是要根据树 K 上的路径构造一条 V 中的路径，其中关键点是保持那些不是 ω 的维度不变，而对于树上已经变成 ω 的部分，可以构造一个路径让其不停的上涨到足够大的地方。

严格来说假设节点 v_m 满足 $L(v_m) \geq \mathbf{x}$，令根节点到其的路径为 $rv_1v_2\dots v_m$，对应边的标识路径则为 $\pi = \mathbf{a}_1\dots\mathbf{a}_m$。为了说明存在一个可达的格局 $\mathbf{y} \geq \mathbf{x}$，我们准备用 K 上的路径去构建一条 V 中可达的路径，如果对于任意的 $i \in [m]$，$L(v_i)$ 中不存在某项为 ω 的维度，则 $(\mathbf{u}, \pi, L(v_m))$ 便是一个运行，$L(v_m)$ 对于 $\mathbf{u}$ 是可达的，(1) 正确。否则令 $I(j) \subseteq [d]$ 表示 $L(v_j)$ 中大小为 ω 的维度集合，特别的 $I(0)$ 用来表示根节点 r 上的标识上大小为 ω 的维度集合。显然有：

$$I(0) = \emptyset \subseteq I(1) \subseteq \cdots \subseteq I(m)$$

考虑使其严格增大的下标 i_k，即满足 $I(i_k - 1) \subsetneq I(i_k)$，这样的下标数最多只有 $k \leq d$ 个。由定义，对于 v_{i_k} 存在一个之前路径的点 v_{j_k} 满足：$L(v_{i_k}) \geq L(v_{j_k})$，且对于 $h \in I(i_k) \setminus I(i_k - 1)$ 有 $L(v_{i_k})[h] > L(v_{j_k})$，令 v_{j_k} 到 v_{i_k} 的路径的标识路径 $\pi[j_k + 1, i_k]$ 为 p_k，p_k 满足以下两点：

- 对于 $h \in I(i_k) \setminus I(i_k - 1)$，有 $\Delta(p_k)[h] > 0$。
- 对于 $h \notin I(i_k)$，有 $\Delta(p_k)[h] = 0$。

记 $i_0 = 0$, $i_{k+1} = m$，将路径 $\mathbf{a}_{i_k+1}\mathbf{a}_{i_k+2}\dots\mathbf{a}_{\mathbf{i_{k+1}}}$ 记为 s_k。下面证明路径 $\pi' = s_0p_1^{n_1}s_1p_2^{n_2}\dots p_k^{n_k}s_k$ 是一条格局 $\mathbf{u}$ 能触发的路径，且 $\mathbf{u} + \Delta(\pi') \geq \mathbf{x}$。这里 $n_i \in \mathbb{N}$ 满足：

$$n_i \geq ||x||_\infty + n \cdot (m + \sum_{j=i+1}^{k} n_j \cdot mn) \tag{3–2}$$

其中 n 是定义的向量加法系统 V 的大小。那么对于 $h \in I(i_l) \setminus I(i_l - 1)$，有：

$$(\mathbf{u} + \Delta(s_0p_1^{n_1}s_1p_2^{n_2}\dots p_l^{n_l})[h] \geq \mathbf{x}[h] + n \cdot (m + \sum_{j=l+1}^{k} n_j \cdot mn) \tag{3–3}$$

而对于在这之前的路径上的格局：

$$(\mathbf{u}+\Delta(s_0p_1^{n_1}s_1p_2^{n_2}\dots s_{l-1})[h]=(\mathbf{u}+\Delta(s_0s_1\dots s_{l-1})[h]\geq 0 \tag{3–4}$$

对于之后路径的格局：

$$(\mathbf{u}+\Delta(\pi'))[h]\geq \mathbf{x}[h]+n\cdot(m+\sum_{j=i+1}^{k}n_j\cdot mn)-n\cdot|s_lp_{l+1}^{n_{l+1}}\dots p_k^{n_k}s_k|\geq \mathbf{x}[h] \tag{3–5}$$

因此路径 π' 是 $\mathbf{u}$ 可触发的一条路径，$\mathbf{u}+\Delta(\pi')\in \mathrm{Reach}_V(\mathbf{u})$ 并且 $\mathbf{u}+\Delta(\pi')\geq\mathbf{x}$。□

有了定理3.2很自然的能获得可覆盖性问题和有界性问题的可判定性结果，具体如下：

定理 3.3　向量加法系统上的可覆盖性问题是可判定的。

证明　只要计算出初始格局 $\mathbf{u}$ 所对应的 Karp-Miller 树后，验证是否存在某个节点的标识比目标格局 $\mathbf{v}$ 大，如果存在这样的节点回答正确，否则拒绝。由定理3.1和定理3.2，该算法是正确的并且可终止的。□

定理 3.4　向量加法系统的有界性问题是可判定的。

证明　只要计算出初始格局 $\mathbf{u}$ 所对应的 Karp-Miller 树后，验证是否存在某个节点的标识 $L(v)\in\mathbb{N}_\omega^d\setminus\mathbb{N}^d$，如果不存在这样的节点回答正确，否则拒绝。由定理3.1和定理3.2，该算法是正确的并且可终止的。□

3.2　可覆盖性算法

通过上一节的 Karp-Miller 树可以获得可覆盖性问题的可判定结论，但是因为其树的大小虽然能被证明是有限的，但是我们无法给出其大小的界，所以它不能获得复杂性上的结果。而在本节将介绍 Rackoff 给出的可覆盖性问题上的指数空间算法[93]。在向量加法系统 V 上有一个非常简单的性质-单调性 (monotonicity)：对于两个格局 $\mathbf{u}_1\leq\mathbf{u}_2$，如果格局 $\mathbf{u}_1$ 能触发路径 π，那么格局 $\mathbf{u}_2$ 也能触发路径 π，即：

$$\mathbf{u}_1+\Delta(\pi)\in\mathrm{Reach}_V(\mathbf{u}_1)\qquad\Rightarrow\qquad\mathbf{u}_2+\Delta(\pi)\in\mathrm{Reach}_V(\mathbf{u}_2) \tag{3–6}$$

这给了一个非常好的直观：为什么可覆盖性问题会比可达性问题要简单。事实上可达性问题的一个显著难的地方是说就算我们知道比如一个相对较小格局 $\mathbf{u}$ 能够到达 $\mathbf{v}$，我们也无法通过此来获知 $\mathbf{u}$ 大的格局 $\mathbf{u}'$ 是否能够达 $\mathbf{v}$，而只能够获

悉 $\mathbf{u}'$ 能够通过相同的路径到达一个比格局 $\mathbf{v}$ 更大的格局 $\mathbf{v}'$。而这对于可覆盖性问题是有帮助的，因为可覆盖性问题只关心格局 $\mathbf{u}$ 是否能到达一个比 $\mathbf{v}$ 更大的格局 $\mathbf{v}'$，因此如果存在一个比格局 $\mathbf{u}$ 小的 $\mathbf{u}'$ 能够到达格局 $\mathbf{v}$，那由单调性就能知道 $\mathbf{u}$ 能够到达一个比格局 $\mathbf{v}$ 更大的格局 $\mathbf{v}'$。

另一方面基于此，如果不考虑整颗 Karp-Miller 树，而是专注于具体的路径去寻找一些性质，比如如果能找到一个增量严格大于 $\mathbf{0}$ 的路径的话就能得到可覆盖性的正面结论，而较小的格局向量 (也就是每一维都受限) 的个数是有限的，所以可以想象如果回答是正确的话，我们可以在一个不太长的路径中找到一个证据 (witness)，然后只要枚举所有这样子的路径，就可以获得可覆盖性问题的判定算法。接下来依据这个思路来说明如果是可覆盖的，则一定可以找到一个不太长的路径说明其是可覆盖的。

为了更好的研究可覆盖性问题，首先放宽对自然数向量的限制，考虑整数向量上的可达关系。

定义 3.3 给定一个整数向量 $\mathbf{w} \in \mathbb{Z}^d$，如果对于 $i \in [d]$，$\mathbf{w}$ 满足对于 $j \in [i]$ 有 $[\mathbf{j}] \geq 0$，则称 $\mathbf{w}$ 是 i-有界的，如果 $\mathbf{w}$ 还满足 $\mathbf{w}[j] < r$，则称 $\mathbf{w}$ 是 (i,r)-有界的。对于 $\mathbf{Z}^d$ 上的一个串 $p = \mathbf{w}_1, \ldots \mathbf{w}_n$，如果每个 $\mathbf{w}_i$ 都是 i-有界的 ((i,r)-有界的)，则称整个序列 p 是 i-有界的 ((i,r)-有界的)。

遵循上面的思路，假设问题的回答是正确的，即从 $\mathbf{u}$ 出发有一条路径能够覆盖 $\mathbf{v}$，我们希望说明可以找到一条不太长的路径可以验证这一点。为此首先考虑要求降低一些的情况；即将空间从自然数向量放宽到整数向量，并且首先只要求第一维能够被覆盖住；这种路径在可覆盖性问题回答正确的情况下肯定是存在的，而找到这种答案以后，在这样的路径基础上一步步在扩大能被覆盖的维度，最终获取能回答可覆盖性问题的证据。

形式化的来说，给定一个 $\mathbb{Z}^d$ 上的串 $p = \mathbf{w}_1 \ldots \mathbf{w}_n$，如果对于 $i \in [d]$ 每个 $\mathbf{w}_j$, $j \in [n]$ 都满足 $\mathbf{w}_j[k] \geq \mathbf{v}[k]$, $k \in [i]$，则称 p 是一个 i-覆盖的序列，特别的如果 p 是某个在 $\mathbb{Z}^d$ 上的运行 ρ 的格局串，称 p 是一个由 $src(\rho)$ 出发的 i-覆盖的格局串，对应的路径则记为 $path(p) = path(\rho)$。可覆盖性问题其实便是寻找一个由 $\mathbf{u}$ 出发 d-有界 d-覆盖的格局串。对于 $\mathbf{x} \in \mathbf{Z}^d$，定义函数 $m(i, \mathbf{x})$：

$$m(i,\mathbf{x}) \stackrel{\text{def}}{=} \begin{cases} \min_{p\text{是一个 }\mathbf{x}\text{ 出发的 }i\text{-有界 }i\text{-覆盖的格局串}} |p|, & \text{如果存在这种格局串,} \\ 0, & \text{如果不存在这种格局串} \end{cases} \tag{3–7}$$

函数 $f(i) \stackrel{\text{def}}{=} \max_{\mathbf{v}\in\mathbb{Z}^d} m(i,\mathbf{v})$ 则表示着 i-有界 i-覆盖的格局串的上界，通过求出 $f(k)$ 的上界便可获得可覆盖证据路径的上界。

引理 3.2 (Rackoff [93])　函数 f 满足下列性质：

1. $f(0) = 1$。
2. 对于 $i \in [d-1]_0$，有 $f(i+1) \leq (2^n f(i))^{i+1} + f(i)$。
3. 存在某个常数 c 满足 $f(d) \leq 2^{2^{cn \log n}}$。

证明　$f(0) = 1$ 是显然的，接下来证明第 2 点。假设存在一条由 $\mathbf{x}$ 出发 $(i+1)$ 有界 $(i+1)$ 覆盖的格局串，分两种情况讨论。

情况一：　存在一条由 $\mathbf{x}$ 出发 $(i+1, 2^n f(i))$ 有界 $(i+1)$ 覆盖的格局串。这种情况是非常简单的，因为在这个格局串里，不会存在两个格局在前 $i+1$ 维相同，否则删除这一段，依旧能获得一条由 $\mathbf{x}$ 出发 $(i+1, 2^n f(i))$ 有界 $(i+1)$ 覆盖的格局串。因此则条格局串的长度不会超过 $(2^n f(i))^{i+1}$。

情况二：　不存在这样的格局串。由假设存在一条由 $\mathbf{x}$ 出发 $(i+1)$ 有界 $(i+1)$ 覆盖的格局串 $p = p_1 p_2$ 满足 p_1 是 $(i+1, (2^n f(i)))$-有界的，令 $p_2[1] = \mathbf{w} \in \mathbb{N}^d$，则 $\mathbf{w}$ 不是 $(i+1, 2^n f(i))$-有界的，不妨令 $\mathbf{w}[i+1] \geq 2^n f(i)$。由情况一和假设有 $|p_1| \leq (2^n f(i))^{i+1}$。注意到 p_2 也是一个 i-有界 i-覆盖的格局串，因此存在一个以 $\mathbf{w}$ 出发的长度小于 $f(i)$ 的 i 有界 i 覆盖的格局串 p_2'，接下来说明由 $\mathbf{x}$ 开始，触发路径 $\pi = path(p_1)path(p_2')$ 所得到的格局串，是 $(i+1)$ 有界 $(i+1)$ 可覆盖的。事实上，只需要考虑第 $i+1$ 维，考虑到如下两个条件：

- $(\mathbf{x} + \Delta(path(p_1)))[i+1] \geq 2^n f(i)$。
- $path(p_2') \leq f(i)$。

我们有：

$$(\mathbf{x} + \Delta(\pi))[i+1] \geq 2^n f(i) - 2^n (f(i) - 1) = 2^n \geq \mathbf{v}[i+1] \tag{3–8}$$

这是因为任何一个规则在第 $i+1$ 维的减少量不会超过 2^n，这里 n 是 V 的大小。因此在走完路径 $path(p_2')$ 后，第 $(i+1)$ 维最多只被减少了 $2^n(f(i)-1)$。因此由 $\mathbf{x}$ 触发路径 π 形成的格局串是 $(i+1)$ 有界 $(i+1)$ 可覆盖的，所以有：

$$f(i+1) \leq |\pi| = (2^n f(i))^{i+1} + f(i)$$

最后来说明第 3 点，事实上由第 2 点，很容易可以获得对于 $i \geq 2$ 有 $f(i) \leq (2^n f(i-1))^{3n}$，因此：

$$f(d) \leq 2^{3n^2} \cdot (f(d-1))^{3n} \leq 2^{3n^{d+2}} \cdot (f(0))^{(3n)^d} \leq 2^{3n^{n+2}} \leq 2^{2^{3n \log n}} \tag{3–9}$$

□

引理 3.2直接可覆盖性问题的指数空间算法。

定理 3.5 (Rackoff [93]) 向量加法系统可覆盖性问题是在指数空间内可解决的。

证明 由引理 3.2, 可以构造一个非确定性的图灵机让其猜一条长度小于 $2^{2^{3n\log n}}$ 的路径并验证其是否是 d-有界并且最终格局比目标格局 $\mathbf{v}$ 大，图灵机接收当且仅当存在这样一条满足要求的路径。显然这是一个指数空间的算法。 □

3.3 有界性算法

本节将介绍有界性问题的指数空间算法[93]。Karp-Miller 树给了直观上很好判定有界性问题的一个方法-定理3.4，即向量加法系统 V 对于某个格局是无界的当且仅当其 Karp-Miller 树中存在一个节点的标识出现了 ω，这说明在根节点到其的一个运行出现了两个节点有序关系 $\leq$, 或者说有一个节点覆盖住了自己的祖先。这是个非常好的性质，但是与可覆盖性一样的原因，Karp-Miller 树的大小无法被一个较小的界给限制住导致这个判定方法没有好的复杂性结果。

与可覆盖性问题相同，Rackoff 通过类似的思路分析了这种路径的长度，其证明了如果 V 对于某个格局是无界的，则可以在一个不太长的路径中找到一条能够自己覆盖住自己的证据，本节就将介绍这个指数空间的算法，我们会沿用章节3.2的一些定义，同时补充定义上述说到的自覆盖 (self-covering) 的概念。

定义 3.4 给定一个 $\mathbb{Z}^d$ 上的格局串 $p = \mathbf{w}_1 \dots \mathbf{w}_n$, 如果存在两个格局满足 $\mathbf{w}_i < \mathbf{w}_j$, 则称 p 是自覆盖的。

有了这个概念，可以将定理3.4重新改写成:

定理 3.6 (Karp and Miller[20]) V 对于格局 $\mathbf{u}$ 是有界的当且仅当存在一条由 $\mathbf{u}$ 出发的 d 有界、自覆盖的格局串。

接下来将介绍由 Rackoff 给出的证明，其证明如果确实是无界的，那么一定存在一条双指数的自覆盖运行。其基本思想与章节3.2相同，是希望通过寻找 i 有界、自覆盖的最短格局串和 $(i+1)$ 有界、自覆盖的最短格局串之间的递推关系。这里的一个关键点是发现如果一条自覆盖的路径很长，那么在其用到的规则里可以找到很多组和为 $\mathbf{0}$ 的线性组合，而由于单调性的存在其实可以删除很多组这样的规则，因为这对有界性的判断不构成影响，从而唯一需要验证的就是删除了这些规则之后该串还能是一个 i 有界的串，即中间的格局不会落入 0 之下的位置，这点则可以由前面讲到的单调性3–6来保证。

具体来说，定义函数 $M(i,\mathbf{x})$

$$M(i,\mathbf{x}) \stackrel{\text{def}}{=} \begin{cases} \min_{p\text{是一个}\mathbf{x}\text{出发的}i\text{-有界、自覆盖的格局串}} |p|, & \text{如果存在这种格局串,} \\ 0, & \text{如果不存在这种格局串} \end{cases} \tag{3–10}$$

令 $g(i) \stackrel{\text{def}}{=} \max_{\mathbf{x}\in\mathbb{Z}^d} M(i,\mathbf{x})$，与上一节3.2相似，由定理3.6可知，$g(d)$ 的值决定了有界性问题的结果。

下面说明 $g(d)$ 的值会被某个双指数限制住。一个简单的想法是，如果一条路径只有指数长，那么其格局的大小也最多指数大。另一方面如果反过来思考的话，如果一个 i 有界、自覆盖的格局串里最多只有指数大的格局但是路径长度却远远超过指数，那么里面一定由很多规则组合起来起到了 $\mathbf{0}$ 的效果。但因为这是一条自覆盖的串，因此即使去掉一些这些规则，也可以重新安排这个路径使其依旧保持 i 有界、自覆盖的特性，从而得到了一条更短的 i 有界、自覆盖的格局串。

具体来说，令 $T_i(\mathbf{w})$ 表示将由 $\mathbf{w}$ 前 i 维组成的 i 维向量，对 V 上的一条路径 $\pi = \mathbf{a}_1 \dots \mathbf{a}_n$，如果 $T_i(\Delta(\pi)) = \mathbf{0}_i$，则称 π 是 i-零和圈 (loop)，特别的如果对于任何一个连续的子路径 $\pi' = \pi[j,k]$ 都有 $T_i(\Delta(\pi')) \neq \mathbf{0}_i$，则称 π 是一个简单 i-零和圈 (simple loop)。对于一个 $\mathbb{Z}^d$ 上的格局 $\mathbf{x}$ 和一个路径 $\pi = \mathbf{a}_1\mathbf{a}_2 \dots \mathbf{a}_n$，令 $p = \mathbf{w}_0\mathbf{w}_1 \dots \mathbf{w}_n$，其中 $\mathbf{w}_j \stackrel{\text{def}}{=} \mathbf{x} + \Delta(\pi[1,j])$，显然 p 是一个 $\mathbb{Z}^d$ 上的格局串，如果 p 是 (i,r) 有界的，则称 π 对于 $\mathbf{x}$ 来说是 (i,r)-合法的 (valid)。Rackoff 给出了如下的引理，其证明了如果存在一条 (i,r)-有界的自覆盖的格局串，那么就存在一条长度不超过 r^{n^c} 的 (i,r)-有界的自覆盖的格局串，这里 c 是一个常数。

引理 3.3 (Rackoff[93])　给定一个 $\mathbb{Z}^d$ 上的格局 $\mathbf{x}$，如果从 $\mathbf{x}$ 出发存在一条 (i,r)-有界、自覆盖的格局串，那么存在一条由 $\mathbf{x}$ 出发长度不超过 r^{n^c} 的 (i,r)-有界、自覆盖的格局串，这里 $c \in \mathbb{N}$ 是一个与 $n,r,\mathbf{x}$ 无关的常数。

证明　令 $p = \mathbf{v}_1 \dots \mathbf{v}_n\mathbf{w}_0 \dots \mathbf{w}_m$ 是一个最短的的由 $\mathbf{x}$ 出发的 (i,r)-有界、自覆盖的格局串，其中 $\mathbf{w}_0 < \mathbf{w}_m$, 由定义我们有 $T_i(p[k]) \in \mathbb{N}^i$，并且由 p 的最短性我们有对于 $j,k \in [n]$ 满足 $T_i(\mathbf{v}_j) \neq T_i(\mathbf{v}_k)$，否则可以去除这一段获得一个更短的格局串 p，而 p' 也是 (i,r)-有界、自覆盖的，与假设矛盾，因此 $n \leq r^{d+1}$。

令格局 $p' = \mathbf{w}_0 \dots \mathbf{w}_m$，则路径 $\pi = \mathbf{a_1} \dots \mathbf{a}_m$ 是格局串 p 对应的路径串，由定义 $\Delta(\pi) > \mathbf{0}_d$。这里的想法是如果 p' 很长，那么由于其是 (i,r) 有界的，那么这段路径上一定存在很多的 i-零和圈，我们证明如果 m 太大，便可以删除以些 i-零和圈使其仍然是 (i,r)-有界、自覆盖的。令 C_π 表示在路径 π 里的所有简单 i-零和圈的变化，即

$$C_\pi = \{\Delta(\pi') | \pi' = \pi[j,k] \wedge \pi'\text{是一个简单}i\text{-零和圈}\}$$

构造如下的序列 $\mathbf{x}_0, \pi_0, \mathbf{x}_1, \pi_1, \ldots$ 满足：

- $\mathbf{x}_0 = \mathbf{0}_d$, $\pi_0 = \pi$。
- π_k 在 $\mathbf{x}_k$ 出发的格局串和 π_0 在 $\mathbf{0}_d$ 出发的格局串对前 i 的投影所构成的集合是相同的，并且 π_k 对于 $\mathbf{x}_k$ 来说是 (i,r)-合法的。
- $\mathbf{x}_k \in \mathbb{Z}^d$ 并且有 $\mathbf{x}_k + \Delta(\pi_k) = \Delta(\pi_0)$。
- $\mathbf{x}_k$ 可以由 $C_{\pi_{k-1}}$ 中的非负线性组合表示。

现在来阐述具体的构造过程。假设已经有了 $\mathbf{x}_k, \pi_k$，如果 $|\pi_k| < (r^d+1)^2$，那么构造停止。否则令 $\pi_k = \mathbf{a}_{k1}\mathbf{a}_{k2}\ldots\mathbf{a}_{km_k}$, $m_k \geq (r^d+1)^2$，其对应的格局串为 $\mathbf{y}_0\mathbf{y}_1\ldots\mathbf{y}_{m_k}$。考虑其格局串的前 $(r^d+1)^2$ 项，将其分成 r^d+1 段，每段包含 r^d+1 个格局，如下所示：

$$\mathbf{y}_{k1}\cdots\mathbf{y}_{k(r^d+1)}, \mathbf{y}_{k(r^d+2)}\cdots\mathbf{y}_{k(2r^d+2)}, \cdots, \mathbf{y}_{k(r^d(r^d+1)+1)}\cdots\mathbf{y}_{k((r^d+1)^2)}, \cdots\mathbf{y}_{km_k} \tag{3–11}$$

考察这些格局在前 i 维的投影，由于其是 (i,r)-有界的，即对于任意的 $\mathbf{y}_j$，有 $T_i(\mathbf{y}_j)[l] < r$, $l \in [i]$，因此在每一段的 r^d+1 个格局中一定有两个格局的前 i 维是相同的，并且存在一段其中所有格局的前 i 维投影都不是第一次出现在这个格局串中。令这一段为 $q = \mathbf{y}_{k(j(r^d+1)+1}\cdots\mathbf{y}_{k((j+1)(r^d+1))}$，$q$ 中一定存在简单 i-零和圈，假设为 $\pi'_k = \mathbf{a}_{k(j_1+1)}\mathbf{a}_{k(j_1+2)}\ldots\mathbf{a}_{k(j_2)}$，显然去掉这一段依旧满足上述条件，即令 $\mathbf{x}_{k+1} = \mathbf{x}_k + \Delta(\pi'_k)$, $\pi_{k+1} = \mathbf{a}_{k1}\ldots\mathbf{a}_{k(j_1)}\mathbf{a}_{k(j_2+1)}\ldots\mathbf{a}_{km_k}$，则我们构造了一组满足条件的 $\mathbf{x}_{k+1}$, π_{k+1}，并且 $|\pi_{k+1}| < |\pi_k|$。

令构造终止在 $\mathbf{x}_l, \pi_l$，则有 $|\pi_l| < (r^d+1)^2$。接下来说明可以将其作适当的修改接在格局串 $p_0 = \mathbf{v}_1\ldots\mathbf{v}_n$ 的后面从而获得一个较短的 (i,r)-有界、自覆盖的格局串。由构造 $\mathbf{x}_l$ 是由一些简单 i-零和圈的线性组合而成的，令 L 为这些简单 i-零和圈的集合。再令 A 为这样的矩阵，其每列都是 L 中的一个不重复的元素，则 A 是一个 $d \times h$ 的矩阵，其中 h 是简单 i-零和圈的个数，即 $h = |L|$。注意到 $\mathbf{x}_l + \Delta(\pi_l) = \Delta(\pi_0) > \mathbf{0}_d$，令 $\mathbf{b} = -\Delta(\pi_0)$，则不等式组 $\mathrm{A}\mathbf{x} > \mathbf{b}$ 存在一个非平凡的非负整数解 $\mathbf{x}_0$。由推论2.2，其存在一个解 $\mathbf{x}_1$ 满足:

$$||\mathbf{x}_1||_1 \leq (1 + h \cdot ||\mathrm{A}||_\infty + h \cdot ||\mathbf{b}||_\infty)^{h+1} \tag{3–12}$$

并且 $\mathrm{A}\mathbf{x}_1$ 可以被写成

$$\mathrm{A}\mathbf{x}_1 = \sum_{\mathbf{c}_i \in L} n_i \mathbf{c}_i \tag{3–13}$$

其中 $n_i \geq 0$，$\Sigma_{i\in[h]} n_i = ||x||_1$。考察由 $\mathrm{A}\mathbf{x}_1$ 出发路径为 π_l 的格局串 $\mathbf{s}_1, \mathbf{s}_2\ldots\mathbf{s}_t$，其满足 $\mathrm{A}\mathbf{x}_1 + \Delta(\pi) > \mathbf{0}_d$。考虑出现在等式3–13中的 $\mathbf{c}_i$，并且这个格局串是 (i,r) 有界

的。由之前的构造能在路径 π_l 中找到一个相同的简单 i-零和圈，将 $\mathbf{c}_i$ 添加进其对应的位置，则得到了一条由 $\mathbf{A}\mathbf{x}_1 - \mathbf{c}_i$ 出发的格局串，并且该格局串还是 (i,r) 有界的。重复以上过程，最终能得到一条 $\mathbf{0}_d$ 出发的 (i,r) 有界的格局串 p_f，其对应的路径为 π_f。则格局串 $p'' = \mathbf{v_1} \dots \mathbf{v}_n, \mathbf{w}'_0 \dots \mathbf{w}'_{m'}$ 是一个 (i,r) 有界、自覆盖的格局串，这里 $\mathbf{w}'_k$ 满足 $\mathbf{w}'_k = \mathbf{w_0} + \Delta(\pi_f[1,k])$，其对应的路径记为 π''。

最后计算路径 π'' 的长度。一个简单 i-零和圈的长度最多为 r^d，所以对于 $\mathbf{c}_i \in L$ 来说，有 $||\mathbf{c}_i||_\infty < 2^n r^d$，这里 n 是向量加法系统 V 的大小，所以 $h = |L| < (2 \cdot 2^n r^d + 1)^d < r^{3n^2}$，因此 $||\mathbf{x}_1||_1 \leq r^{n^{c_1}}$ 对于某个无关 $n, \mathbf{x}$ 的常数 $c_1 \in \mathbb{N}$。因此最后获得格局串 p_f 的过程最多需要 $r^{n^{c_1}}$ 次操作，每次操作至多将路径变长 r^d 次，因此路径 π_f 的长度至多为:

$$|\pi_f| < (r^d + 1)^2 + r^d \cdot r^{n^{c_1}} < r^{2n^{c_1}} \tag{3–14}$$

因此路径 π'' 的长度不超过 $r^d + r^{2n^{c_1}} < r^{n^{c_2}}$，这里 $c_2 \in \mathbb{N}$ 也是一个无关 $n, \mathbf{x}$ 的常数，引理得证。 □

引理3.3说明如果一条 i-有界、自覆盖的格局串里的格局是被 r 限制住的，那么就可以找到一条长度不超过 r^{n^c} 的 (i,r)-有界、自覆盖的格局串。由此，类比于在可覆盖性问题中对 f 的估计，Rackoff 进一步得出了关于 $g(i)$ 的估计:

引理 3.4 (Rackoff[93])　存在常数 $c_1, c_2 \in \mathbb{N}$ 使得函数 g 满足下列性质:

1. $g(0) \leq 2^{n^{c_1}}$。
2. $g(i+1) \leq (2^n g(i))^{n^{c_1}}$。
3. $g(d) \leq 2^{2^{c_2 n \log n}}$。

证明　第一点是显然的，因为假设存在自覆盖的格局串，那么就存在一条平凡的 $(0,2)$-有界、自覆盖的格局串，由引理3.3，在 V 中存在一条长度不超过 $2^{n^{c_1}}$ 的 $(0,2)$-有界、自覆盖的格局串，这里 $c_1 \in \mathbb{N}$ 是一个常数，因此 $g(0) \leq 2^{n^{c_1}}$。

接下来证明第二点。假设从 $\mathbf{u}$ 出发存在一条 $(i+1)$-有界、自覆盖的格局串。分两种情况讨论。

情况一:　存在一条由 $\mathbf{u}$ 出发的 $(i+1, 2^n g(i))$-有界、自覆盖的格局串，则由引理3.3，存在一条长度不超过 $(2^n g(i))^{n^{c_1}}$ 的 $(i+1, 2^n g(i))$-有界、自覆盖的格局串，因此 $g(i+1) \leq (2^n g(i))^{n^{c_1}}$。

情况二:　不存在由 $\mathbf{u}$ 出发的 $(i+1, 2^n g(i))$-有界、自覆盖的格局串。假设由 $\mathbf{u}$ 出发的 $(i+1)$-有界、自覆盖的格局串为 $p = \mathbf{w}_0 \mathbf{w}_1 \dots \mathbf{w}_m$，其对应的路径为 $\pi = \mathbf{a}_1 \dots \mathbf{a}_m$，

并且不妨假设 $j \in [m]_0$ 满足 $\mathbf{w}_j < \mathbf{w}_m$。令 $l \in [m]_0$ 使得其是最小的下标满足 $\mathbf{w}_l$ 不是 $(i+1, 2^n g(i))$-有界的，并且不妨假设 $\mathbf{w}_l[i+1] \geq 2^n g(i)$。

考虑一条最短的由 $\mathbf{u}$ 出发的 $(i+1, 2^n g(i))$-有界格局串 $p_1 = \mathbf{v}_0\mathbf{v}_1 \ldots \mathbf{v}_{m'}$ 满足 $T_{i+1}(\mathbf{v}_{m'}) = T_{i+1}(\mathbf{w}_l)$，其对应路径为 $\pi_1 = \mathbf{b}_1 \ldots \mathbf{b}_{m'}$。则由 $\mathbf{v}_{m'}$ 出发的对应路径为 $\pi[l+1, m]\pi[j+1, m]$ 的格局串 $p_2 = \mathbf{v}_{m'}\mathbf{x}_0\mathbf{x}_1 \ldots \mathbf{x}_{m-l}\mathbf{y}_0\mathbf{y}_1\mathbf{y}_{m-j}$ 是 $(i+1)$-有界的、自覆盖的，这是由 $T_{i+1}(\mathbf{v}_{m'}) = T_{i+1}(\mathbf{w}_l)$ 以及 $\Delta(\pi[j+1, m]) > \mathbf{0}_d$ 所保证的。因此存在一条由定义 $\mathbf{v}'_m$ 出发长度不超过 $g(i)$ 的 i-有界、自覆盖的格局串 $\mathbf{v}_{m'}\mathbf{v}'_1 \ldots \mathbf{v}'_h$，其对应路径为 $\pi_2 = \mathbf{c}_1 \ldots \mathbf{c}_h$。接下来证明由 $\mathbf{u}$ 出发对应路径为 $\pi_3 = \pi_1\pi_2$ 的格局串 p_3 是 $(i+1)$-有界、自覆盖的。由构造只需要考察第 $i+1$ 维是否大于等于 0，注意到 $(\mathbf{u} + \Delta(\pi_1))[i+1] = \mathbf{w}_l[i+1] > 2^n g(i)$，而任何一步规则的改变量最多为 2^n，因此对于任意的 $h \in [|\pi_3|]$ 有：

$$(\mathbf{u} + \Delta(\pi_3[1, h]))[i+1] > 2^n g(i) - 2^n g(i) = 0 \tag{3–15}$$

下面计算一下 p_3 的长度，由于 p_1 是最短的，因此不存在 $h, k \in [m']_0$ 满足 $T_{i+1}(\mathbf{v}_h) = T_{i+1}(\mathbf{v}_k)$，从而 $m' \leq (2^n g(i))^{i+1}$，因此 p_3 的长度至多为 $g(i) + m' \leq (2^n g(i))^{i+1} + g(i) \leq (2^n g(i))^{n^{c_1}}$，第 2 点得证。

最后估计 $g(d)$，事实上由第 1 点和第 2 点不难得到：

$$g(d) \leq (2^n)^{n^{c_1 d+1}} \cdot g(0)^{n^{c_1 d}} \leq 2^{n^{2c_1 d + c_1 + 2}} \leq 2^{2^{c_2 n \log n}} \tag{3–16}$$

这里 $c_2 \in \mathbb{N}$ 是一个常数，引理得证。 □

引理3.4直接给出了有界性问题的指数空间算法。

定理 3.7 (Rackoff [93]) 向量加法系统有界性问题是在指数空间内可解决的。

证明 由引理 3.4，可以构造一个非确定性的图灵机让其猜一条长度小于 $2^{2^{cn \log n}}$ 的路径并验证其是否是 d-有界、自覆盖的，图灵机接收当且仅当不存在这样一条满足要求的路径。显然这是一个指数空间的算法。 □

3.4 本章小结

本章介绍了可覆盖性问题和有界性问题的基本结论。我们介绍了可覆盖性问题和有界性问题的指数空间算法，而关于其 **EXPSPACE**-难的证明，将在章节六中作介绍。此外，关于其固定维度的可覆盖性问题和有界性问题，当维度 $d \geq 2$ 时在 [77, 95-96] 中其证明了是 **PSPACE**-完备的，而维度 $d = 1$ 时在 [97] 中被证明是 **NP**-完备的。

这两个问题在向量加法系统的研究中有着重要的意义，首先从方法上看，所用到的 Karp-Miller 树对解决无穷状态但满足某种良序的验证问题起到了重要的作用；其次相比较同样处于核心问题的可达性问题，这两个问题的复杂性远远低于可达性问题，并且这两个性质都可以作为某种安全性的验证，在实际中起着重大的作用。因此关于这两个问题的研究，在向量加法系统功能方面更着重的在于如何在实际中设计一些更高效的算法来完成一些验证目标；而对于向量加法系统的衍生模型，则可以先通过对这两个问题的研究去探索可达性问题，比如在下推向量加法系统模型中，其可达性问题和可覆盖性问题就是可以互相规约的，额外开销仅仅是多一个维度[42]。

第四章　可达性算法-KLMST 分解

本章将介绍向量加法系统可达性问题的上界算法-KLMST 算法。可达性问题一直是向量加法系统研究中的重点，迄今已有 50 多年的历史。自 Mayr，Kosaraju，Lambert，Sacerdote，Tenney 在上世纪八十至九十年代提出其可判定的 KLMST 分解算法[69-72, 142] 以来，人们对其的研究一直没有停歇；这其中有着 KLMST 分解算法本身非常难理解的原因，也有着该算法并没有获得具体的上界还有进一步的空间。另一方面，KLMST 分解算法也对其他的问题起到了很重要的作用，比如其可以证明向量加法系统所定义的语言的向下闭包 (downward closure) 是可以有效计算的[143-144]。Leroux 也据此在 [73, 145] 提出了可达性问题可判定性的一个简单证明，该证明将在下一章做介绍。而在 15 年 Leroux 和 Schmitz 在 [83-84] 提出了对 KLMST 分解算法的一个简化并且容易理解的版本，即用理想来分解，并根据秩函数获得了可达性算法的第一个具体上界 $\mathbf{F}_{\omega^3}$，随后这一上界迅速的在 [85] 中被提升到了 $\mathbf{F}_{\omega^2}$。随后在 19 年，Leroux 通过在他人对该问题的研究[146-148] 得到启发最终证明其算法是在 $\mathbf{F}_\omega$ 中的[79]，由于下界方面的最新结论[80-82] 因此我们知道这一上界是紧的；本章就将介绍这个证明，给出 KLMST 分解算法一个简单直观明了的解释。

本章的安排如下：第一节将介绍 KLMST 分解算法的核心序列——KLM 序列；第二节将介绍可泵性的概念，我们将说明满足一类性质的向量加法系统有着简单的可达性证据，第三节将具体的介绍 KLMST 分解算法的内容以及通过秩函数获取其最终的 $\mathbf{F}_\omega$ 上界。此外注意到本章所提到的算法需要对向量加法系统进行分解，因此本章默认向量加法系统是带状态的，令本章讨论的 d 维带状态的向量加法系统为 $V = (Q, T, A)$，初始与目标格局为 $\mathbf{m} = p(\mathbf{u})$ 与 $\mathbf{n} = q(\mathbf{v})$。

4.1　KLM 序列

本节将介绍 KLM 序列。KLM 序列最早由 Kosaraju[71] 提出，可以视作是一般化的向量加法系统。严格来说，定义如下：

定义 4.1　一个 KLM 序列 $\xi = (\mathbf{x}_0 V_0 \mathbf{y}_0)\mathbf{a}_1(\mathbf{x}_1 V_1 \mathbf{y}_1)\mathbf{a}_2 \dots \mathbf{a}_n(\mathbf{x}_n V_n \mathbf{y}_n)$ 满足：

- $V_i = (Q_i, T_i, A_i, p_i, q_i)$ 是一个有起点 p_0 和终点 q_0 的 d 维向量加法系统满足 $A_i \subseteq A$。
- 对于 $i \in \{0, 1, \dots, n-1, n\}$ 有 $\mathbf{x}_i, \mathbf{y}_i \in \mathbb{N}^d_\omega$。
- 对于 $i \in [n]$ 有 $\mathbf{a}_i \in A$。

ξ 的大小 $|\xi|$ 定义为：

$$|\xi| = 2(d+1)^{(d+1)}[n + \sum_{i=1}^{n} ||\mathbf{a}_i||_1 + \sum_{i=0}^{n}(||\mathbf{x}_i||_1 + |V_i| + ||\mathbf{y}_i||_1)] \tag{4–1}$$

其中 V_i 用一进制编码的大小。我们称 $\pi = \sigma_0\mathbf{a}_0\sigma_1 \ldots \mathbf{a}_n\sigma_n$ 为 KLM 序列 ξ 上的一条路径，其中 σ_i 是 V_i 中 $\mathbf{x}_i$ 到 $\mathbf{y}_i$ 在 $\mathbb{Z}_\omega^d$ 上的一条路径。如果 π 满足存在 $\mathbf{m}_0, \mathbf{n}_0 \ldots \mathbf{m}_n, \mathbf{n}_n \in \mathbb{N}^d$ 使得：

- $\mathbf{m}_i \sqsubseteq \mathbf{x}_i,\ \mathbf{n}_i \sqsubseteq \mathbf{y}_i$。
- 对 $i \in [n]_0$ 有 $p_i(\mathbf{m}_i) \xrightarrow{\sigma_i}_{\mathbb{N}^d} q_i(\mathbf{n}_i)$。
- 对 $i \in [n-1]_0$ 有 $\mathbf{m}_{i+1} = \mathbf{n}_i + \mathbf{a}_i$。

则称 π 是完全的，记 L_ξ 为 ξ 上的所有完全的路径的集合。

事实上，如果在 $\mathbb{Z}^d$ 上考虑向量加法系统的可达性问题，其可达性问题几乎等同于解一个线性方程组，即是否存在一组非负整数解 $x_\mathbf{a}$ 满足:$\mathbf{u} + \sum_{\mathbf{a}\in A} x_\mathbf{a}\mathbf{a} = \mathbf{v}$。这一方法不适用于可达性问题原因在于其可能在中间出现负数的格局，但这一方程依旧有助于对可达性问题进行研究，比如如果该方程无解，那么原来的目标格局一定不可达。因此下面我们来定义关于一个 KLM 序列的特征方程。对于一个 KLM 序列 $\xi = (\mathbf{x}_0V_0\mathbf{y}_0)\mathbf{a}_1(\mathbf{x}_1V_1\mathbf{y}_1)\mathbf{a}_2 \ldots \mathbf{a}_n(\mathbf{x}_nV_n\mathbf{y}_n)$，定义其特征方程 E_ξ 如下：

$$\mathbf{n}_i = \mathbf{m}_i + \sum_{\mathbf{t}=(p,\mathbf{a},q)} \phi_i(\mathbf{t}) \cdot \mathbf{a},\ i \in [n]_0 \tag{4–2}$$

$$\mathbf{m}_{i+1} = \mathbf{n}_i + \mathbf{a}_i,\ i \in [n-1]_0 \tag{4–3}$$

$$\mathbf{1}^i_{q_0} - \mathbf{1}^i_{p_0} = \sum_{\mathbf{t}=(p,\mathbf{a},q)\in T_i} \phi_i(\mathbf{t})(\mathbf{1}^i_q - \mathbf{1}^i_p),\ i \in [n]_0 \tag{4–4}$$

$$\mathbf{m}_i \sqsubseteq \mathbf{x}_i \quad , \quad \mathbf{n}_i \sqsubseteq \mathbf{y}_i,\ i \in [n]_0 \tag{4–5}$$

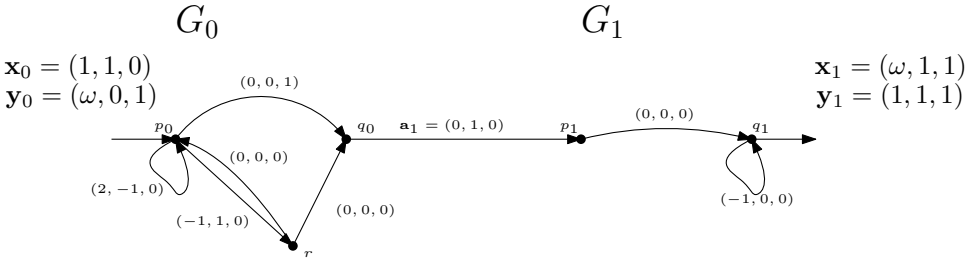
其中 $\mathbf{1}^i_p \in \mathbb{N}^{|Q_i|}$ 表示 V_i 中状态 p 的指示向量，即只有 p 这一维为 1 其余都为 0，$\phi \in \mathbb{N}^T$ 则是 $T_i \to \mathbb{N}$ 的一个映射，在后面也会将等式4–2中的 $\sum_{\mathbf{t}=(p,\mathbf{a},q)} \phi_i(\mathbf{t}) \cdot \mathbf{a}$ 记为 $\Delta(\phi_i)$，方程4–4即是基尔霍夫方程 (kirchhoff system)。我们称满足上述方程的序列 $h = (\mathbf{m}_0, \phi_0, \mathbf{n}_0) \ldots (\mathbf{m}_n, \phi_n, \mathbf{n}_n)$ 为 ξ 的特征序列，并且记 $h(i) = (\mathbf{m}_i, \phi_i, \mathbf{n}_i)$，令 $h(i)[\_]$ 表示其中的 $\_$ 项，记 $|h| = \sum_{i=1}^{n}(||\mathbf{m}_0||_1 + ||\mathbf{n}_i||_1 + \sum_{t\in T_i} \phi_i(t))$ 为 h 的大小，对于 ξ 如果存在一个特征序列，则称 E_ξ 是可满足的 (satisfiable)，也称 ξ 是可满足的，否则称 ξ 是不可满足的。下面的引理很自然的说明了不可满足的 KLM 序列是不可达的。

引理 4.1　不可满足的 KLM 序列不存在完全的路径。

证明 假设 KLM 序列 $\xi = (\mathbf{x}_0 V_0 \mathbf{y}_0)\mathbf{a}_1(\mathbf{x}_1 V_1 \mathbf{y}_1)\mathbf{a}_2 \ldots \mathbf{a}_n(\mathbf{x}_n V_n \mathbf{y}_n)$ 存在一条完全的路径 $\pi = \sigma_0 \mathbf{a}_0 \sigma_1 \ldots \mathbf{a}_n \sigma_n$，则存在 $\mathbf{m}_0, \mathbf{n}_0 \ldots \mathbf{m}_n, \mathbf{m}_n$ 满足：

$$\mathbf{m}_0 \xrightarrow{\sigma_0} \mathbf{n}_0 \xrightarrow{\mathbf{a}_1} \mathbf{m}_1 \xrightarrow{\sigma_1} \cdots \xrightarrow{\mathbf{a}_n} \mathbf{m}_n \xrightarrow{\sigma_n} \mathbf{n}_n$$

并且由定义有 $\mathbf{m}_i \sqsubseteq \mathbf{x}_i$，$\mathbf{n}_i \sqsubseteq \mathbf{y}_i$。令 ϕ_i 为 σ_i 的 Parikh 像，不难验证 $h = (\mathbf{m}_0, \phi_0, \mathbf{n}_0) \ldots (\mathbf{m}_n, \phi_n, \mathbf{n}_n)$ 是其的一个特征序列，即 ξ 是可以满足的，引理得证。□



$G_0 = (Q_0, T_0, p_0, q_0)$

$Q_0 = \{p_0, q_0, r\}$

$T_0 = \{t_1, t_2, t_3, t_4, t_5\}$

$t_1 = (p_0, (2,-1,0), p_0),\ t_2 = (p_0, (-1,1,0), r)$

$t_3 = (p_0, (0,0,1), q_0),\ t_4 = (r, (0,0,0), p_0)$

$t_5 = (r, (0,0,0), q_0)$

$G_1 = (Q_1, T_1, p_1, q_1)$

$Q_1 = \{p_1, q_1\}$

$T_1 = \{t_6, t_7\}$

$t_6 = (p_1, (0,0,0), q_1),\ t_7 = (q_1, (-1,0,0), q_1)$

图 4–1 KLM 序列例子

Figure 4–1 An example of KLM sequence

例 4.1 接下来考虑一个 KLM 序列的例子来理解上述概念。考虑图4–1中的 KLM 序列 $\xi = (\mathbf{x}_0 G_0 \mathbf{y}_1)\mathbf{a}_1(\mathbf{x}_1 G_1 \mathbf{y}_1)$，其特征方程为：

- $\mathbf{m}_0 \sqsubseteq (1,1,0),\ \mathbf{n}_0 \sqsubseteq (\omega,0,1)$。
- $\mathbf{m}_1 \sqsubseteq (\omega,1,1),\ \mathbf{n}_0 \sqsubseteq (1,1,1)$。
- $\mathbf{m}_1 = \mathbf{n}_0 + \mathbf{a}_1$。
- $\mathbf{n}_0 = \mathbf{m}_0 + \phi_0(t_1)(2,-1,0) + \phi_0(t_2)(-1,1,0) + \phi_0(t_3)(0,0,1) + \phi_0(t_4)(0,0,0) + \phi_0(t_5)(0,0,0)$。
- $\mathbf{n}_1 = \mathbf{m}_1 + \phi_1(t_6)(0,0,0) + \phi_1(t_7)(-1,0,0)$。
- $(0,1,0) - (1,0,0) = \phi_0(t_1)((1,0,0) - (1,0,0)) + \phi_0(t_2)((0,0,1) - (1,0,0)) + \phi_0(t_3)((0,1,0) - (1,0,0)) + \phi_0(t_4)((1,0,0) - (0,0,1)) + \phi_0(t_5)((0,1,0) - (0,0,1))$。
- $(0,1) - (1,0) = \phi_1(t_6)((0,1) - (1,0)) + \phi_1(t_7)((0,1) - (0,1))$。

该方程是可满足的，下述序列 $h = (\mathbf{m}_0 \phi_0 \mathbf{n}_0)(\mathbf{m}_1 \phi_1 \mathbf{n}_1)$ 便是其的一个特征序列：

- $\mathbf{m}_0 = (1,1,0),\ \mathbf{n}_0 = (3,0,1),\ \mathbf{m}_1 = (3,1,1),\ \mathbf{n}_1 = (1,1,1)$。

- $\phi_0(t_1) = \phi_0(t_3) = 1,\ \phi_0(t_2) = \phi_0(t_4) = \phi_0(t_5) = 0$。
- $\phi_1(t_6) = 1,\ \phi_1(t_7) = 2$。

特别的该序列蕴含了一条完全的路径，即：

$$p_0(1,1,0) \xrightarrow{t_1 t_3} p_1(3,0,1) \xrightarrow{\mathbf{a}_1} p_1(3,1,1) \xrightarrow{t_6 t_7 t_7} q_1(1,1,1)$$

这并不一定能成立，比如令 ξ 中的 $\mathbf{x}_0$ 替换成 $\mathbf{x}'_0 = (0,0,0)$，其依旧存在特征序列 $h' = (\mathbf{m}'_0\phi'_0\mathbf{n}'_0)(\mathbf{m}'_1\phi'_1\mathbf{n}'_1)$:

- $\mathbf{m}_0 = (0,0,0),\ \mathbf{n}_0 = (1,0,1),\ \mathbf{m}_1 = (1,1,1),\ \mathbf{n}_1 = (1,1,1)$。
- $\phi_0(t_1) = \phi_0(t_3) = \phi_0(t_2) = \phi_0(t_4) = 1,\ \phi_0(t_5) = 0$。
- $\phi_1(t_6) = 1,\ \phi_1(t_7) = 0$。

但很显然，其并不蕴含一条完全的路径。此外，如果令 $\mathbf{x}_1 = (\omega, 1, 2)$，则很显然新得到的 KLM 序列 ξ'' 是不可满足的，因为其特征方程无解，也自然不存在完全的路径。

接下来定义 ξ 上的齐次特征方程 E^0_ξ(homogeneous characteristic system)，其具体如下：

$$\mathbf{n}_i = \mathbf{m}_i + \sum_{\mathbf{t}=(p,\mathbf{a},q)} \phi_i(\mathbf{t}) \cdot \mathbf{a},\ i \in [n]_0 \tag{4–6}$$

$$\mathbf{0} = \sum_{\mathbf{t}=(p,\mathbf{a},q)\in T_i} \phi_i(\mathbf{t})(\mathbf{1}^i_q - \mathbf{1}^i_p),\ i \in [n]_0 \tag{4–7}$$

$$\mathbf{m}_i[j] = 0,\ \text{if } \mathbf{x}_i[j] \neq \omega,\ i \in [n]_0,\ j \in [d] \tag{4–8}$$

$$\mathbf{n}_i[j] = 0,\ \text{if } \mathbf{y}_i[j] \neq \omega,\ i \in [n]_0,\ j \in [d] \tag{4–9}$$

满足上述方程的序列 $h^0 = (\mathbf{m}^0_0, \phi^0_0, \mathbf{n}^0_0) \ldots (\mathbf{m}^0_n, \phi^0_n, \mathbf{n}^0_n)$ 为 ξ 的齐次特征序列，并且记 $h^0(i) = (\mathbf{m}^0_i, \phi^0_i, \mathbf{n}^0_i)$，令 $h^0(i)[\_]$ 表示其中的 _ 项，我们记 $|h_0| = \sum_{i=1}^n (||\mathbf{m}^0_0||_1 + ||\mathbf{n}^0_i||_1 + \sum_{t\in T_i} \phi^0_i(t))$ 为 h_0 的大小，由章节2.4的内容可以得出下列引理：

引理 4.2 (Leroux[79])　给定一个可满足的 KLM 序列 $\xi = (\mathbf{x}_0 G_0 \mathbf{y}_1)\mathbf{a}_1(\mathbf{x}_1 G_1 \mathbf{y}_1)$，则有如下结论：

- 对于任意的 $j \in [d]$，集合 $\{h(i)[\mathbf{m}_i[j]] | h$是一个特征序列$\}$ 是无穷的当且仅当存在一个齐次特征序列 h_0 满足 $h^0(i)[\mathbf{m}_i[j]] > 0$。
- 对于任意的 $j \in [d]$，集合 $\{h(i)[\mathbf{n}_i[j]] | h$是一个特征序列$\}$ 是无穷的当且仅当存在一个齐次特征序列 h_0 满足 $h^0(i)[\mathbf{n}_i[j]] > 0$。
- 对于任意的 $t \in T_i$，集合 $\{h(i)[\phi_i(t)] | h$是一个特征序列$\}$ 是无穷的当且仅当存在一个齐次特征序列 h_0 满足 $h^0(i)[\phi^0_i(t)] > 0$。

特别的 $\{h(i)[\_]\}$ 如果是有限的，其和不会超过 $|\xi|^{|\xi|-1}$。

证明 只要注意到齐次特征序列是可加的，即令 h 是 ξ 的一个特征序列，h_0 是 ξ 的一个齐次特征序列，则对于任意 $n \in \mathbb{N}$，$h + nh_0$ 都是 ξ 的一个特征序列，再由定理2.4和推论2.1,2.2可知任何一个特征序列都可以分解成一个大小不超过 $|\xi|^{|\xi|-1}$ 的特征序列和一组大小不超过 $|\xi|^{|\xi|-3}$ 的齐次特征序列的和即可得到结论。 □

接下来定义一些 KLM 序列 ξ 上的一些性质，而在本节的最后将证明如果 ξ 满足一定的性质，则其如果存在一条完全的路径，那么便可以找到一条不太长的完全的路径。

如果向量加法系统 V_i 对应的有向图是强连通的，则称 V_i 是强连通的。给定 $\xi = (\mathbf{x}_0 V_0 \mathbf{y}_0)\mathbf{a}_1(\mathbf{x}_1 V_1 \mathbf{y}_1)\mathbf{a}_2 \dots \mathbf{a}_n(\mathbf{x}_n V_n \mathbf{y}_n)$，如果 ξ 满足每个 V_i 都是**强连通**的 (strongly connected)，则称 ξ 是**强连通**的。如果 ξ 满足：

- $\mathbf{x}_i[j] = \omega$ 蕴含集合 $\{h(i)[\mathbf{m}_i[j]]\}$ 是无限的。
- $\mathbf{y}_i[j] = \omega$ 蕴含集合 $\{h(i)[\mathbf{n}_i[j]]\}$ 是无限的。

则称 ξ 是**可容忍**的 (saturated)。如果对于 $t \in T_i$，集合 $\{h(i)[\phi_i(t)]\}$ 是无限的，则称 t 是无限的，如果对任意 $t \in T_i$ 都是无限的，则称 ξ 是**无界**的 (unbounded)。

对于一个有起点和终点的的 d 维向量加法系统 $V = (Q, T, p, q)$，考虑其中特殊的一类维度 $I \subseteq [d]$，对于任何一个 $i \in I$，其存在一个函数 $g_i : Q \to \mathbb{N}$ 满足：对任意 $t = (p, \mathbf{a}, q) \in T$ 有 $g_i(q) = g_i(p) + \mathbf{a}[i]$，称 I 里的维度都是**固定**的 (fixed)，事实上 g_i 函数表明一个格局在 V 运行到不同状态时第 i 维的值完全由初始格局决定，因此可以理解为这一维可以被编码进状态里，因此称 g_i 是关于 i 的偏移函数。将此想法扩展到 KLM 序列里，称 $\mathbf{x}V\mathbf{y}$ 是**固定**的，如果对于 V 里任何一个固定的维度 i，存在函数 $g_i : Q \to \mathbb{N}$ 满足：

- 对任意 $t = (p, \mathbf{a}, q) \in T$ 有 $g_i(q) = g_i(p) + \mathbf{a}[i]$。
- $g_i(p) \sqsubseteq \mathbf{x}[i]$, $g_i(q) \sqsubseteq \mathbf{y}[i]$

如果对于 ξ 里每一个 $\mathbf{x}_i V_i \mathbf{y}_i$ 其都是**固定**的，则称 ξ 是**固定**的。下述引理给出了一个简单的判断强连通的 KLM 序列是否是固定的方法。

引理 4.3 给定一个强连通的 KLM 序列 $\xi = \mathbf{x}V\mathbf{y}$，其中 $V = (Q, T, p, q)$，令 $I \subseteq [d]$ 是 V 固定的维度，ξ 是固定的当且仅当对任意 $i \in I$ 其偏移函数 f_i 满足对任意的 $r \in Q$：

- 如果 $\mathbf{x}[i], \mathbf{y}[i] \in \mathbb{N}$，则 $\mathbf{x}[i] - f_i(p) = \mathbf{y}[i] - f_i(q)$。
- 如果 $\mathbf{x}[i] = \omega, \mathbf{y}[i] \in \mathbb{N}$，则 $\mathbf{x}[i] - f_i(p) + f_i(r) \geq 0$。
- 如果 $\mathbf{x}[i] \in \mathbb{N}$, $\mathbf{y}[i] = \omega$，则 $\mathbf{y}[i] - f_i(q) + f_i(r) \geq 0$。

证明　假设 ξ 是固定的，即对于 $i \in I$ 存在函数 $g_i : Q \to \mathbb{N}$ 满足对于任意的 $(p, \mathbf{a}, q) \in T$ 有 $g_i(q) = g_i(p) + \mathbf{a}[i]$，则存在 $z_i \in \mathbb{Z}$ 使得对任意的 $r \in Q$ 有 $z_i + f_i(p) = g_i(p)$，结合 $g_i(p) \sqsubseteq \mathbf{x}[i]$, $g_i(q) \sqsubseteq \mathbf{y}[i]$ 可以得到上述结论。

反之，如果 f_i 满足上述三点，我们可以如下的构造 g_i 函数：

- 如果 $\mathbf{x}[i] = \mathbf{y}[i] = \omega$，则令 $g_i = f_i$。
- 如果 $\mathbf{x}[i] \in \mathbb{N}$，则令 $g_i(r) = \mathbf{x}[i] - f_i(p) + f_i(r)$。
- 如果 $\mathbf{y}[i] \in \mathbb{N}$，则令 $g_i(r) = \mathbf{y}[i] - f_i(q) + f_i(r)$。

第一种情况显然 ξ 是固定的，接下来说明第二种情况下 ξ 也是固定的，而第三种则是对称的。注意到 $g_i(q') = g_i(p') + \mathbf{a}[i]$ 对于任意的 $(p', \mathbf{a}, q') \in T$，只需说明 $g_i(q) \sqsubseteq y[i]$。事实上，如果 $y[i] = \omega$ 则是显然的，反之由第一个条件有 $\mathbf{x}[i] - f_i(p) = \mathbf{y}[i] - f_i(q)$，因此 $g_i(q) = y[i] \sqsubseteq y[i]$ 从而 ξ 是固定的，引理得证。□

最后来介绍 KLM 序列里最特殊的一个性质-可泵性。回顾之前的讨论，一个 KLM 序列 ξ 的特征方程有解不代表其有一条完全的路径，这是因为其可能在路径的中间某一维度小于 0，但如果 ξ 可以做到在一开始的时候将这一维度增大的足够大，而在最后又任意的降回来，那么这一问题便不复存在，比如在图4–1中的 KLM 序列里，G_0 可以通过 $t_1t_2t_3$ 的循环将第一维任意增大，而在 G_1 中通过 t_7 可以将第一维任意的减少，那么就不用担心在运行的过程中第一维会发生小于 0 的情况。我们将这样的性质称之为可泵性，具体定义如下：对于 $\mathbf{x}V\mathbf{y}$，其中 $V = (Q, T, p, q)$，令 $I \subseteq [d]$ 是 V 中固定维度的集合，如果对于任何的 $i \notin I$ 存在 $\mathbf{x}' \in \mathbb{N}_\omega^d$ 满足：$p(\mathbf{x}) \xrightarrow{V_i} p(\mathbf{x}')$ 并且有 $\mathbf{x}' \geq \mathbf{x}$, $\mathbf{x}'[i] > \mathbf{x}[i]$ 或者 $\mathbf{x}[i] = \omega$，则称其是**前可泵**的 (forward pumpable)；如果对于任何的 $i \notin I$ 存在 $\mathbf{y}' \in \mathbb{N}_\omega^d$ 满足：$q(\mathbf{y}') \xrightarrow{V_i} q(\mathbf{y})$ 并且有 $\mathbf{y}' \geq \mathbf{y}$, $\mathbf{y}'[i] > \mathbf{y}[i]$ 或者 $\mathbf{y}[i] = \omega$，则称其是**后可泵**的 (backward pumpable)，如果即是前可泵又是后可泵的，则称其是**可泵**的，特别的如果对于 $i \notin I$ 不满足，则称其关于 i 是不前可泵 (后可泵) 的。对于一个 KLM 序列 $\xi = (\mathbf{x}_0V_0\mathbf{y}_0)\mathbf{a}_1(\mathbf{x}_1V_1\mathbf{y}_1)\mathbf{a}_2 \ldots \mathbf{a}_n(\mathbf{x}_nV_n\mathbf{y}_n)$，如果其每个部分 $\mathbf{x}_iV_i\mathbf{y}_i$ 都是**可泵**的，则称 ξ 是**可泵**的。

一个 KLM 序列如果是可满足的，强连通的，可容忍的，无界的，固定的，则称其是**标准**的 (standard)；如果一个标准 KLM 序列是可泵的，则称其是**正则**的 (normal)。在下一节将证明一个正则的 KLM 序列一定存在一条长度不太长的完全的路径，即如下定理：

定理 4.1 (Leroux[79])　对于一个正则的 KLM 序列 ξ，存在一条长度不超过 $|\xi|^{3|\xi|}$ 的完全的路径。

4.2 正则 KLM 序列

本节将对正则的 KLM 序列作进一步的探讨，首先考虑可泵性这一性质，其实际上是某种可覆盖性体现。事实上根据第三章 Rackoff 关于可覆盖性在 **EXPSPACE** 空间的结果，考虑 $\mathbf{x}V\mathbf{y}$，其中 $V = (Q, T, p, q)$，要判断 $i \in [d]$ 是否是前可泵的，只需要调用可覆盖性算法判断 $p(\mathbf{x} + \mathbf{1}_i)$ 对于 $p(\mathbf{x})$ 是否是可覆盖的即可，对于后可覆盖将 V 反向即可。因此至多需要 $2d$ 次调用可覆盖性的算法便可以判定 $\mathbf{x}V\mathbf{y}$ 是否是可泵的，因此可以在 **EXPSPACE** 空间内判定一个 KLM 序列是否是可泵的。

但兴趣不止于此。我们不仅希望知道如何判断一个 KLM 序列是可泵的，还希望知道不可泵的界，即如果 $\mathbf{x}V\mathbf{y}$ 对于 i 是不前可泵的，则有 $\mathbf{x}[i] < \omega$，此时我们希望知道 $\mathbf{x}[i]$ 的大小的界 B，因为如果有界，则可以将这一维编码进状态里使得这一维变成固定的从而不用再去研究。在 [33] 中通过使用 Karp-Miller 树能计算出 B 有一个 **Ackermann** 的界，但考虑在章节三中介绍的对可覆盖性的讨论，如果每一维都能变得很大，则不可能不能实现可覆盖性，由此可以想象如果不可泵，一定存在某一维上的取值严格受限。遵循这个思路，Leroux 给出了下列引理，其告诉我们可以在指数空间内判断是否可泵，并且给出了关于 B 一个指数上的界。

引理 4.4 (Leroux[79]) 令 V 是一个 d 维带状态向量加法系统，令 $C > |V|$，假设其中存在一条路径：

$$q_0(\mathbf{c}_0) \xrightarrow{\mathbf{a}_1} q_1(\mathbf{c}_1) \xrightarrow{\mathbf{a}_2} \cdots \xrightarrow{\mathbf{a}_k} q_k(\mathbf{c}_k) \tag{4–10}$$

满足 $\mathbf{c}_i \in \mathbb{N}_\omega^d$，并且对于任意的 $i \in [d]$ 都存在 $j \in [k]_0$ 使得 $\mathbf{c}_j[i] > C^{1+n^n}$，其中 $n = |I| = |\{i|\mathbf{c}_0[i] \in \mathbb{N}\}|$，则存在一条路径 $|\pi| < C^{(n+1)^{(n+1)}}$ 满足：

$$q_0(\mathbf{c}_0) \xrightarrow{\pi} q_k(\mathbf{c}) \tag{4–11}$$

其中 $\mathbf{c}$ 满足对任意 $i \in [d]$ 都有 $\mathbf{c}[i] > C - |V|$。

证明 对 n 做归纳，当 $n = 0$ 时命题显然成立，只需要找一条 q_0 到 q_k 的路径即可。假设对 $< n$ 的命题均成立，考虑 $= n$ 的情况，由假设存在一条路径 $\mathbf{a}_1 \dots \mathbf{a}_k$ 满足：

$$q_0(\mathbf{c}_0) \xrightarrow{\mathbf{a}_1} q_1(\mathbf{c}_1) \xrightarrow{\mathbf{a}_2} \cdots \xrightarrow{\mathbf{a}_k} q_k(\mathbf{c}_k) \tag{4–12}$$

令 k_i 表示最小的下标使得 $\mathbf{c}_{k_i}[i] > C^{1+n^n}$，记 $\hat{k} = \min_{i \in I} k_i$，则对于任意的 $i \in I$，$j \in [\hat{k} - 1]$ 都有 $\mathbf{c}_j[i] < C^{1+n^n}$，从而 $\hat{k} < |Q| \cdot (C^{1+n^n})^n < C^{n+1+n^{n+1}}$。

记 $I' = \{i | k_i > \hat{k}\}$，很显然有 $|I'| < |I| = n$，对 $j \geq \hat{k}$ 我们定义 $\mathbf{x}_j = \mathbf{c}_j|_{I'}$，显然有 $\mathbf{x}_j \sqsubseteq \mathbf{c}_j$，因此我们有：

$$q_{\hat{k}}(\mathbf{x}_{\hat{k}}) \xrightarrow{\mathbf{a}_{\hat{k}+1}} \cdots \xrightarrow{\mathbf{a}_k} q_k(\mathbf{x}_k) \tag{4–13}$$

注意到运行满足:

- $n' \overset{\text{def}}{=} |\{i | \mathbf{x}_{\hat{k}}[i] \in \mathbb{N}\}| = |I'|$。
- 对于 $j \in [d] \setminus I'$ 有 $\mathbf{x}_{\hat{k}}[j] = \omega > C^{1+n'^{n'}}$。
- 对于 $j \in I'$ 有 $\mathbf{x}_{k_j}[j] > C^{1+n^n} > C^{1+n'^{n'}}$。

因此有归纳假设存在路径 $|\boldsymbol{\pi}'| < C^{(n'+1)^{(n'+1)}}$ 使得:

$$q_{\hat{k}}(\mathbf{x}_{\hat{k}}) \xrightarrow{\pi'} q_k(\mathbf{c}) \tag{4–14}$$

其中 $\mathbf{c}$ 满足对于 $j \in [d]$ 有 $\mathbf{c}[j] > C - |V|$。接下来证明 $\boldsymbol{\pi}'' = \mathbf{a}_1 \ldots \mathbf{a}_{\hat{k}} \boldsymbol{\pi}'$ 就是所求的路径 $\boldsymbol{\pi}$。首先考虑 $\boldsymbol{\pi}''$ 的长度：

$$|\boldsymbol{\pi}''| = \hat{k} + |\boldsymbol{\pi}'| < C^{n+1+n^{n+1}} + C^{(n'+1)^{(n'+1)}} \leq C^{n+1+n^{n+1}} + C^{n^n} < C^{(n+1)^{n+1}} \tag{4–15}$$

最后来证明 $q_0(\mathbf{c}_0) \xrightarrow{\pi''} q_k(\mathbf{c})$。只需要注意到在路径的后半段 $\boldsymbol{\pi}'$，其对一个维度的影响最多为 $|V| \cdot C^{n^n}$，因此对于满足 $k_i = \hat{k}$ 的维度 i 由归纳假设有 $\mathbf{c}_{\hat{k}}[i] > C^{n^n+1}$，注意到:

$$C^{1+n^n} - |V| \cdot C^{n^n} = (C - |V|) \cdot C^{n^n} > C - |V| \tag{4–16}$$

从而 $\boldsymbol{\pi}''$ 是一条合理的路径，引理得证。 □

由上述引理立马可以获得下述关于正则的 KLM 序列上的性质。

引理 4.5　给定一个正则的 KLM 序列 $\xi = (\mathbf{x}_0 V_0 \mathbf{y}_0)\mathbf{a}_1(\mathbf{x}_1 V_1 \mathbf{y}_1)\mathbf{a}_2 \ldots \mathbf{a}_n(\mathbf{x}_n V_n \mathbf{y}_n)$，则存在一组路径序列 $\{\pi_i, \sigma_i\}$ 以及一组向量序列 $\{\mathbf{x}'_i, \mathbf{y}'_i\}$ 满足：

- 对任意的 $i \in [n]_0$ 有 $|\pi_i|, |\sigma_i| < |\xi|^{(d+1)^{d+1}}$。
- 对任意的 $i \in [n]_0$ 有 $\mathbf{x}'_i \geq \mathbf{x}_i$，且对于 $j \in [d]$ 有 $\mathbf{x}_i[j] \neq \omega$ 蕴含 $\mathbf{x}'_i[j] \geq \mathbf{x}_i[j]$。
- 对任意的 $i \in [n]_0$ 有 $\mathbf{y}'_i \geq \mathbf{y}_i$，且对于 $j \in [d]$ 有 $\mathbf{y}_i[j] \neq \omega$ 蕴含 $\mathbf{y}'_i[j] \geq \mathbf{y}_i[j]$。
- 对任意的 $i \in [n]_0$ 有 $p_i(\mathbf{x}_i) \xrightarrow{\pi_i}_{V_i} p_i(\mathbf{x}'_i)$，$q_i(\mathbf{y}'_i) \xrightarrow{\sigma_i}_{V_i} q_i(\mathbf{y}_i)$。

证明　只需注意到一个正则的 KLM 序列既是可泵的也是强连通的，因此利用引理4.4寻找到增长的路径后再利用强连通性回到起点或者终点即可。 □

最后回到定理4.1的证明。

证明 (定理4.1的证明) 令正则 KLM 序列 $\xi = (\mathbf{x}_0 V_0 \mathbf{y}_0)\mathbf{a}_1(\mathbf{x}_1 V_1 \mathbf{y}_1)\mathbf{a}_2 \dots \mathbf{a}_n(\mathbf{x}_n V_n \mathbf{y}_n)$，注意到其是标准的，因此存在一个特征序列 $h = (\mathbf{m}_0, \phi_0, \mathbf{n}_0) \dots (\mathbf{m}_n, \phi_n, \mathbf{n}_n)$ 和齐次特征序列 $h^0 = (\mathbf{m}_0^0, \phi_0^0, \mathbf{n}_0^0) \dots (\mathbf{m}_n^0, \phi_n^0, \mathbf{n}_n^0)$ 满足：

- $|h| \le 2|\xi|^{|\xi|-1}$, $|h_0| \le |\xi|^{|\xi|-2}$。
- 对于任何 $t \in T_i$ 有 $h(i)[\phi_i(t)] > 0$, $h^0(i)[\phi_i^0(t)] > 0$。
- 对于任何 $j \in [n]_0$ 有 $\mathbf{x}_i[j] = \omega$ 蕴含 $h^0(i)[\mathbf{m}_i^0[j]] > 0$，$\mathbf{y}_i[j] = \omega$ 蕴含 $h^0(i)[\mathbf{n}_i^0[j]] > 0$。

由引理4.5存在一组路径序列 $\{\pi_i, \sigma_i\}$，记 ϕ_{π_i}, ϕ_{σ_i} 是其对应的在 V_i 上的 Parikh 像，由上述定义存在 $r \in \mathbb{N}$ 满足对任意 $t \in T_i$：

$$\varphi_i(t) \overset{\text{def}}{=} (r\phi_i^0 - \phi_{\pi_i} - \phi_{\sigma_i})(t) > 0 \tag{4–17}$$

这里 $r \overset{\text{def}}{=} 2|\xi|^{1+(d+1)^{d+1}}$ 即可。注意到 φ_i 满足 E_ξ^0 中的基尔霍夫方程4–7，并且注意到 V_i 是强连通的，因此由欧拉引理存在一个在 p_i 上的圈 π_{φ_i} 满足其 Parikh 像是 φ_i，记 I_j 是 V_i 中固定的维度的集合，令：

- $\mathbf{m}_i^1 = r\mathbf{m}_i^0 + \Delta(\phi_{\pi_i})$。
- $\mathbf{n}_i^1 = r\mathbf{n}_i^0 - \Delta(\phi_{\sigma_i})$。

我们先说明 $\mathbf{m}_i^1, \mathbf{n}_i^1 \in \mathbb{N}^d$ 并且满足对于 $j \in I_i$ 有 $\mathbf{m}_i^1[j] = \mathbf{n}_i^1[j] = 0$，对于 $j \notin I_i$ 则有 $\mathbf{m}_i^1[j], \mathbf{n}_i^1[j] > 0$。由对称性只用证明 $\mathbf{m}_i^1$ 的性质，并且由定义只需考虑 $j \notin I_i$，这里分两种情况讨论：

- $\mathbf{x}_i[j] \in \mathbb{N}$。由于 ξ 是可泵的，因此由引理4.5，有 $\Delta(\phi_{\pi_i})[j] > 0$，从而有 $\mathbf{m}_i^1[j] > 0$。
- $\mathbf{x}_i[j] = \omega$。同样由引理4.5有 $|\pi_i| < |\xi|^{(d+1)^{d+1}}$，因此有:

$$\mathbf{m}_i^1[j] > r + \Delta(\phi_{\pi_i})[j] > r + |\xi| \cdot |\pi_i| > 0 \tag{4–18}$$

因此由上述讨论，存在如下的运行：

$$p_i(\mathbf{m}_i + r\mathbf{m}_i^0) \xrightarrow{\pi_i} p_i(\mathbf{m}_i + \mathbf{m}_i^1),\ q_i(\mathbf{n}_i + \mathbf{n}_i^1) \xrightarrow{\sigma_i} q_i(\mathbf{n}_i + r\mathbf{n}_i^1) \tag{4–19}$$

由于 $\mathbf{m}_i^1, \mathbf{n}_i^1 \ge \mathbf{0}$，因此对于任意的 $s \in \mathbb{N}$ 有如下的运行：

$$p_i(\mathbf{m}_i + sr\mathbf{m}_i^0) \xrightarrow{(\pi_i)^s} p_i(\mathbf{m}_i + s\mathbf{m}_i^1),\ q_i(\mathbf{n}_i + s\mathbf{n}_i^1) \xrightarrow{(\sigma_i)^s} q_i(\mathbf{n}_i + sr\mathbf{n}_i^1) \tag{4–20}$$

注意到 $\mathbf{n}_i^1 = \mathbf{m}_i^1 + \Delta(\varphi_i)$ 以及 π_{φ_i} 是 p_i 上的一个圈，因此有：

$$p_i(\mathbf{m}_i + s\mathbf{m}_i^1) \xrightarrow{(\pi_{\varphi_i})^s} p_i(\mathbf{m}_i + s\mathbf{n}_i^1) \tag{4–21}$$

最后注意到 V_i 是强连通的，因此存在一条 p_i 到 q_i 的路径 θ_i 满足其 Parikh 像为 ϕ_i，并且由于 $|\phi_i| \leq 2|\xi|^{|\xi|-1}$，只要令 $s = r|\xi|^{|\xi|-2}$ 既可以保证 $p_i(\mathbf{m}_i + s\mathbf{n}_i^1)$ 在运行路径 θ_i 的时候其每一维都不会小于 0，因此有如下运行：

$$\begin{aligned} p_i(\mathbf{m}_i + sr\mathbf{m}_i^1) &\xrightarrow{(\pi_i)^s} p_i(\mathbf{m}_i + s\mathbf{m}_i^1) \xrightarrow{(\pi_{\varphi_i})^s} p_i(\mathbf{m}_i + s\mathbf{n}_i^1) \\ &\xrightarrow{\theta_i} q_i(\mathbf{n}_i + s\mathbf{n}_i^1) \xrightarrow{(\sigma_i)^s} q_i(\mathbf{n}_i + sr\mathbf{n}_i^0) \end{aligned}$$

因此路径 $\pi = (\pi_0)^s(\pi_{\varphi_0})^s\theta_0(\sigma_0)^s\mathbf{a}_1 \ldots \mathbf{a}_\mathbf{n}(\pi_n)^s(\pi_{\varphi_n})^s\theta_n(\sigma_n)^s$ 是一条完全的路径，最后来计算 π 的长度：

$$|\pi| < n + 2|\xi|^{|\xi|-1} + s \cdot r \cdot n \cdot |\xi|^{|\xi|-2} < |\xi|^{3|\xi|} \tag{4–22}$$

因此定理得证，即正则 KLM 序列 ξ 存在一条长度不超过 $|\xi|^{3|\xi|}$ 的完全的路径。□

定理4.1告诉我们一旦 KLM 序列是正则的，便可以在指数空间内找寻到其可达的一个证据，注意到对于向量加法系统 V 问 $q(\mathbf{v})$ 是否对于 $p(\mathbf{u})$ 是可达的，即相当于问 KLM 序列 $\mathbf{u}V\mathbf{v}$ 是否存在一条完全的路径，其中 V 的起点定义为 p，终点定义为 q。因此不难想象 KLMST 算法就是通过将不是正则的 KLM 序列分解至其是若干个正则的 KLM 序列，最终在通过定理4.1获取其可达的路径，而在下一节中我们将仔细介绍这个分解算法的流程。

4.3　分解算法

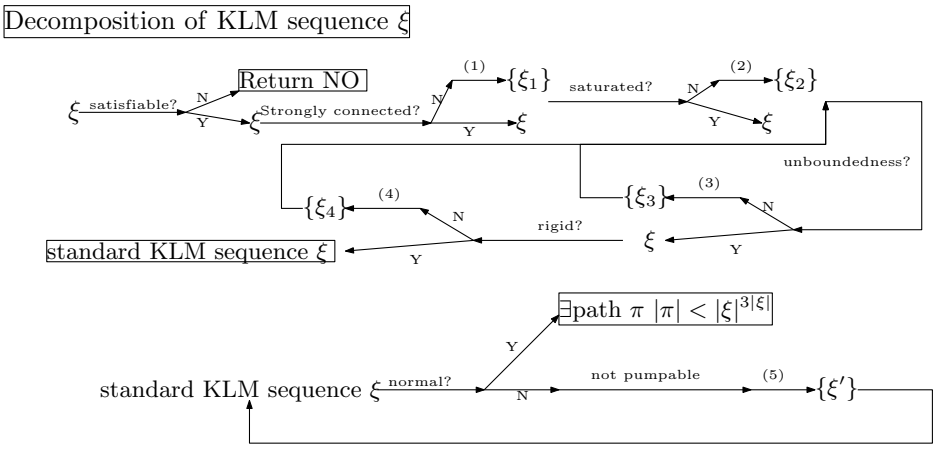


图 4–2 KLM 序列分解

Figure 4–2 Decomposition of KLM sequence

本节将介绍整个分解算法的流程。图4–2展示了 KLM 序列的分解流程，事实上我们可以将任何一个 KLM 序列首先转化为标准的 KLM 序列，然后通过不停的判断其是否正则，最终将其转化为一个是正则的 KLM 序列。

如图4–2所示，对一个序列的分解可能会破坏其他性质，因此为了证明该分解是可终止的，我们首先来定义关于 KLM 序列的秩 (rank) 的概念。

对于一个向量加法系统 $V = (Q, T, A)$，定义对于 $t \in T$ 的向量空间 $\mathbb{V}_V(t) \subseteq \mathbb{Q}^d$，其是所有在 V 里包含 t 的圈的增量所张成的空间，令其所有的圈的增量张成的向量空间为 $\mathbb{V} \subseteq \mathbb{Q}^d$，则显然 $\mathbb{V}_V(t)$ 是 $\mathbb{V}$ 的子空间，而下面说明强连通性和无界性都能使其跟 $\mathbb{V}$ 相同。

引理 4.6 (Leroux[79]) 给定一个强连通的向量加法系统 $V = (Q, T, A)$，对于任意的 $t \in T$ 有 $\mathbb{V}_V(t) = \mathbb{V}$。

证明 $\mathbb{V}_V(t) \subseteq \mathbb{V}$ 是显然的，考虑另一方向由于 V 是强连通的，因此存在 V 上的一个圈 π 满足其经过了 V 上所有的边，从而 $\Delta(\pi) \in \mathbb{V}_V(t)$。令 $\pi_1, \dots \pi_k$ 生成 V，则由 π 的构造 $\pi + \pi_j$ 都可表示为 V 上的某个包含 t 的圈，从而对于任意 $j \in [k]$ 有 $\Delta(\pi + \pi_j) \in \mathbb{V}_V(t)$，因此 $\Delta(\pi_j) \in \mathbb{V}_V(t)$，即 $\mathbb{V} \subseteq \mathbb{V}_V(t)$，引理得证。 □

引理 4.7 (Leroux[79]) 给定一个强连通的 KLM 序列 $\mathbf{x}V\mathbf{y}$，令其无限的边的集合为 T'，将只包含 T' 里边的圈的增量长成的空间记为 $\mathbb{V}'$，则 $\mathbb{V} = \mathbb{V}'$ 蕴含 $T = T'$。

证明 令 $h_0 = \mathbf{m}_0\phi_0\mathbf{n}_0$ 是其的一个齐次特征序列满足对于任意 $t \in T$ 有 $\phi_0(t) > 0$。由于 V 是强连通的可知存在一个包含了所有 $t \in T$ 的圈 π，记其 Parikh 像为 ϕ_π，由条件 $\mathbb{V} = \mathbb{V}'$，则存在由 T' 中组成的圈 $\theta_1 \dots \theta_k$ 以及 $c_1, \dots c_k$ 满足：

$$\Delta(\pi) = \sum_{i=1}^{k} c_i \Delta(\theta_i) \tag{4–23}$$

记圈 θ_i 的 Parikh 像为 ϕ_{θ_i}，显然该 Parikh 像满足对任意的 $t \in T \setminus T'$ 有 $\phi_{\theta_i}(t) = 0$。令 $M = \max_{i \in [k],\, t \in \theta_i} c_i \phi_{\theta_i}(t)$，则 $M\phi_0$ 满足对于任何 $i \in [k]$ 有:

$$\forall t \in T',\ M\phi_0(t) - c_i\phi_{\theta_i}(t) > 0 \tag{4–24}$$

令 $\phi_0' = kM\phi_0 - \sum_{i=1}^{k} c_i\phi_{\theta_i} + \phi_\pi$，注意到其是可以满足基尔霍夫方程的，从而 $h_0' = (kM\mathbf{m}_0)\phi_0'(kM\mathbf{n}_0)$ 也是一个齐次特征序列，因此有 $T \subseteq T'$，从而 $T = T'$，引理得证。 □

由上述边生成的向量空间可以定义一个向量加法系统的秩，令其秩 $\mathrm{rank}(V) = (r_d, \ldots, r_0) \in \mathbb{N}^{d+1}$ 是一个 $d+1$ 维的自然数向量定义为：

$$r_i = |\{t|\dim(\mathbb{V}_V(t) = i)\}|,\ i \in [d]_0 \tag{4–25}$$

而对于一个 KLM 序列 $\xi = (\mathbf{x}_0 V_0 \mathbf{y}_0)\mathbf{a}_1(\mathbf{x}_1 V_1 \mathbf{y}_1)\mathbf{a}_2 \ldots \mathbf{a}_n(\mathbf{x}_n V_n \mathbf{y}_n)$，其秩定义为 $\mathrm{rank}(\xi) = \sum_{i=1}^{n} \mathrm{rank}(V_i)$，秩之间使用 $<_{lex}$ 比较关系，注意到 $(\mathbb{N}^{d+1}, <_{lex})$ 是一个良序，其序数类型为 ω^ω。

注　该秩是 Leroux19 年在 [79] 所提出来的，其通过表示边产生的向量空间的维度的向量找出了一个新的终止关系，并且其序数类型为 ω^ω，最终使得该算法能落到 **Ackermann** 里。而在此之前对于 KLM 序列的秩则是定义为由 $\mathbf{r}_i = (n_{i_1}, n_{i_2}, n_{i_3})$ 组成的多重集，其中 $\mathbf{r}_i$ 是 KLM 序列中第 i 部分 $\mathbf{x}_i V_i \mathbf{y}_i$ 的秩，其定义为：

- $n_{i_1} = d - \max\{|\{j|\mathbf{x}_i[j] \in \mathbb{N}\}|, |\{j|\mathbf{y}_i[j] \in \mathbb{N}\}|\}$。
- $n_{i_2} = |T_i|$。
- $n_{i_3} = 2d - |\{j|\mathbf{x}_i[j] \in \mathbb{N}\}| - |\{j|\mathbf{y}_i[j] \in \mathbb{N}\}|$

其三元组用的是字典序 $<_{lex}$，该序关系则是定义在多重集上的[149]，从而其序数类型为 ω^{ω^3}，最终获得了 $\mathbf{F}_{\omega^3}$ 的界，而在 [85] 中进一步获得了序数类型为 $\omega^{\omega(d+1)}$ 的秩，将算法上界提升到了 $\mathbf{F}_{\omega^2}$。

4.3.1　分解成标准 KLM 序列

下面先来介绍转换为标准 KLM 序列的步骤。转换成强连通的 KLM 序列是简单的，如图4–3只需要将其分解成若干个强连通的子图 V_1, V_2, V_3，然后将其不能强连通的部分用外在的规则连接起来即可。注意到这些子图可能之间不止一条路径，因此最后分解成的是若干个强连通的 KLM 序列，即 ξ 最终被分解成了三个强连通的 KLM 序列 ξ_1, ξ_2, ξ_3，这也就是图4–2中的 (1)，具体请看下列引理。

引理 4.8　给定一个不是强连通的 KLM 序列 ξ，可以在 $2^{O(|\xi|)}$ 的时间内构造出一组有限的强连通序列集合 $\Xi = \{\xi'\}$ 满足 $L_\xi = \bigcup_{\xi' \in \Xi} L_{\xi'}$，$|\xi'| \le |\xi|$，$\mathrm{rank}(\xi') <_{lex} \mathrm{rank}(\xi)$。

证明　令 $\zeta = \mathbf{x}_i V_i \mathbf{y}_i$ 是 ξ 不是强连通的部分，则由有向图的定义存在 $k \le |Q_i| \in \mathbb{N}$ 使得 V_i 可以分解成 $V_{i_1} \ldots V_{i_k}$ 这 k 个强连通的部分，满足 p_i 是 V_{i_1} 的起点，q_i 是 V_{i_k} 的终点，并且这 k 个部分形成一个起点是 V_{i_1} 终点是 V_{i_k} 的有向无环图 G_i。

则对于 G_i 上一条从 V_{i_1} 到 V_{i_k} 的路径 $V_{j_0} t_1 \ldots t_l V_{j_l}$，其中 $t_i = (p^i, \mathbf{a}^i, q^i)$ 满足 $p^i \in V_{j_{i-1}}$，$q^i \in V_{j_i}$，构造如下新的 KLM 序列 ζ'：

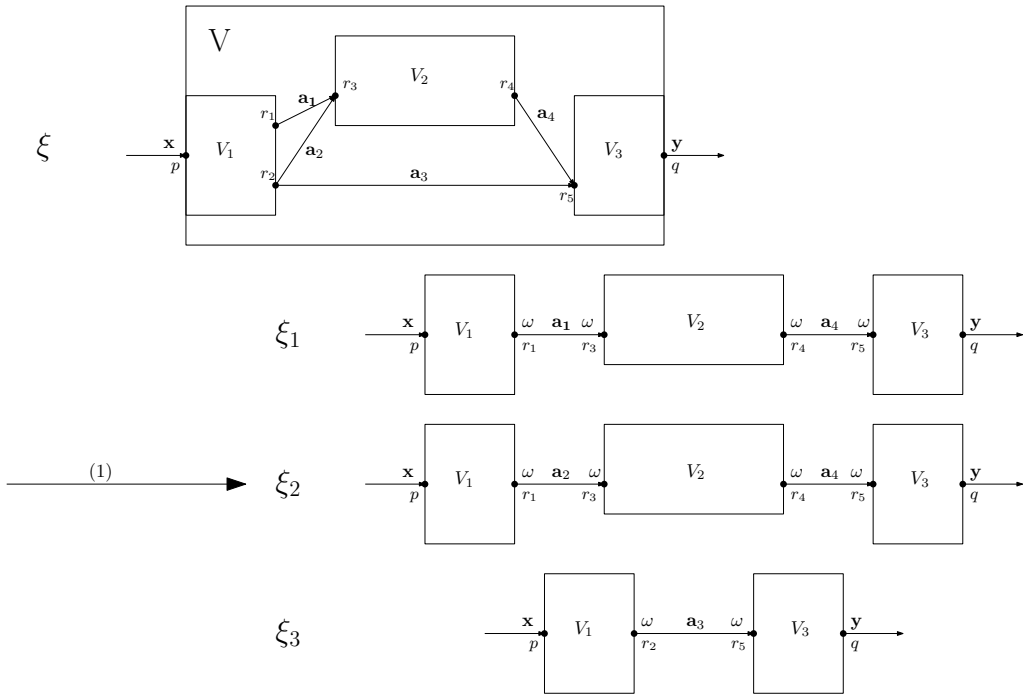


图 4–3 KLM 序列分解-(1)

Figure 4–3 Decomposition of KLM sequence(1)

- $\zeta' = (\mathbf{x}_i V_{j_0} \omega)\mathbf{a}^1(\omega V_{j_1}\omega)\ldots\mathbf{a}^l(\omega V_{j_l}\mathbf{y}_i)$。
- 记 $p^0 = p_i$, $q^{l+1} = q_i$，则 V_{j_i} 的起点为 p^i 终点为 q^{i+1}。

显然有 $L_{\zeta'} \subseteq L_\zeta$，并且有 $|\zeta'| = 2(d+1)^{(d+1)}(n + \sum_{i=1}^{n}||\mathbf{a}^i||_1 + \sum_{i=0}^{l}|V_{j_i}|) \leq |\zeta|$。并且由于 G_i 是有限的，因此上述路径也是有限的；同时注意到任何一条 ζ 上的完全的路径都能从 G_i 上找寻到一条对应的路径，从而将 $|\xi|$ 中所有不强连通的部分都按上述方式分解后，其可以分解成一个有限的强连通 KLM 序列集合 $\{\xi'\}$ 满足 $L_\xi = \bigcup_{\xi\in\Xi} L_\xi$, $|\xi'| \leq |\xi|$，其时间显然在 $2^{O(|\xi|)}$ 内，由定义其秩 $\text{rank}(\xi') <_{lex} \text{rank}(\xi)$，引理得证。 □

接下来我们介绍如何把一个强连通的 KLM 序列变成可容忍的，即图4–2中的(2)。事实上由引理4.2只要根据其对应的特征方程与齐次方程的解的情况，将中间的 ω 作适当的代换即可。图4–4做了个简单的解释，不难验证 $\mathbf{y}_0, \mathbf{x}_1$ 的第二维的取值是有限的，因此其可以分解成 ξ_1, ξ_2 两个序列，这里省略了那些不可满足的 KLM 序列，具体可见下面引理。

引理 4.9 给定一个强连通的 KLM 序列 ξ，可以在 $2^{O(|\xi|^{|\xi|})}$ 的时间内构造出一组有限的可容忍的强连通序列集合 $\Xi = \{\xi'\}$ 满足 $L_\xi = \bigcup_{\xi'\in\Xi} L_{\xi'}$, $|\xi'| \leq |\xi|^{|\xi|}$, $\text{rank}(\xi') <_{lex} \text{rank}(\xi)$。

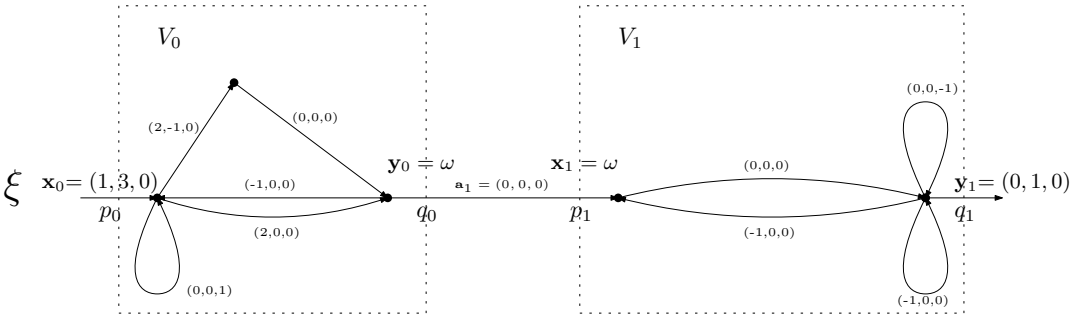


图 4–4　KLM 序列分解-(2)

Figure 4–4　Decomposition of KLM sequence(2)

证明　令 $\xi = (\mathbf{x}_0 V_0 \mathbf{y}_0)\mathbf{a}_1(\mathbf{x}_1 V_1 \mathbf{y}_1)\mathbf{a}_2 \dots \mathbf{a}_n(\mathbf{x}_n V_n \mathbf{y}_n)$，考虑其特征序列 h 中那些取值有限的 $h(i)[[\mathbf{m}]_i[\mathbf{j}]]$，$h(i)[\mathbf{n}_i[j]]$，由引理4.2这些值 $\leq |\xi|^{|\xi|-1}$，记将 ξ 里 $\mathbf{x}_i, \mathbf{y}_i$ 对应为 ω 的位置换成 $\leq |\xi|^{|\xi|-1}$ 的一个值的集合为 Ξ，显然 Ξ 是有限的，并且对于任何一个 $\xi' \in \Xi$，ξ' 是强连通的并且有 $|\xi'| \leq |\xi|^{|\xi|}$，并且由定义其秩 $\mathrm{rank}(\xi') <_{lex} \mathrm{rank}(\xi)$，从而引理成立。 □

接下来我们继续进行 KLM 序列的分解，下一个考虑的性质是无界性，注意到由引理4.2此只需要判断其齐次特征序列 h_0 中是否存在 $h_0(i)[\phi_i^0(t)] = 0$ 即可判断，因此可以在 **NP** 的时间内判断 ξ 是不是无界的。下述引理说明如果 ξ 不是无界的，则可以将其分解成秩更小的 KLM 序列的集合。直觉上来说我们只需要将其中使用次数有限的边全部拆出来，而由引理4.7该操作不会使得秩变大。图4–5给了个简单的例子作直观的解释，其省略了再使用引理4.9将其变得可容忍的步骤。

引理 4.10　给定一个可满足的强连通的 KLM 序列 ξ，如果其不是无界的，则可以在 $2^{O(|\xi|^{|\xi|}|\xi|)}$ 的时间内构造出一组可容忍可满足的强连通序列集合 $\Xi = \{\xi'\}$ 满足 $L_\xi = \bigcup_{\xi' \in \Xi} L_{\xi'}$，$|\xi'| \leq |\xi|^{|\xi|^{|\xi|+1}}$，$\mathrm{rank}(\xi') <_{lex} \mathrm{rank}(\xi)$。

证明　令 $\xi = (\mathbf{x}_0 V_0 \mathbf{y}_0)\mathbf{a}_1(\mathbf{x}_1 V_1 \mathbf{y}_1)\mathbf{a}_2 \dots \mathbf{a}_n(\mathbf{x}_n V_n \mathbf{y}_n)$，假设 $\mathbf{x}_i V_i \mathbf{y}_i$ 不是无界的，令 T_i' 表示其无限的边，考虑一个边序列 $\zeta = t^1 t^2 \dots t^k$，其中 $t^j = (p^j, \mathbf{a}^j, q^j) \in T_i \setminus T_i'$，$k \leq |\xi|^{|\xi|}$，我构造如下的 KLM 序列 $\xi_i = (\mathbf{x}^0 V^0 \mathbf{y}^0)\mathbf{a}^1 \dots \mathbf{a}^k(\mathbf{x}^k V^k \mathbf{y}^k)$:

- $\mathbf{x}^0 = \mathbf{x}_i$，$\mathbf{y}^k = \mathbf{y}_i$。
- 对 $j \in [k], l \in [k-1]_0$ 有 $\mathbf{x}^j = \mathbf{y}^l = \omega$。

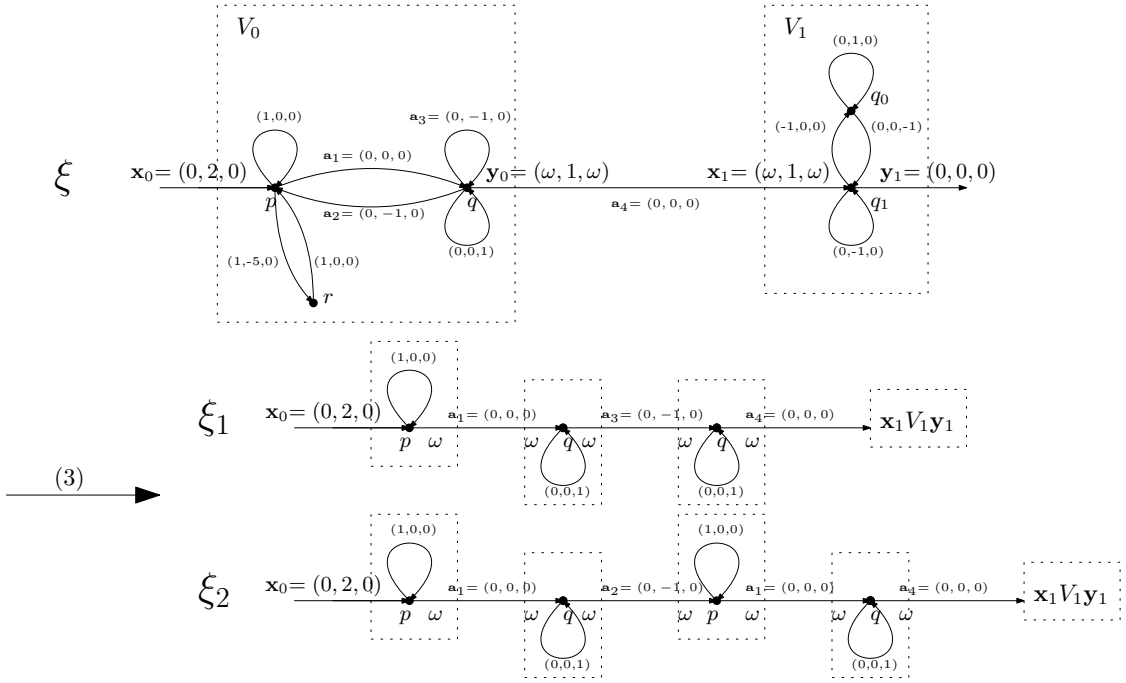


图 4–5 KLM 序列分解-(3)

Figure 4–5 Decomposition of KLM sequence(3)

- V^0 的起点是 p_i，终点是 p^1;V^k 的起点是 q^k，终点是 q_i。对于 $j \in [k-1]$，V^j 的起点是 q^j，终点是 p^{j+1}。
- 对于 $j \in [k]_0$，V^j 的构造是由起点出发用 T_i' 中所有能用的边构造成的 V_i 的子向量加法系统，如果其不能到达终点，则放弃这个边序列。

将 ξ_i 替换 $\mathbf{x}_iV_i\mathbf{y}_i$ 后可以得到新的 KLM 序列，类似的将所有不无界的部分都用这种方式替换，我们得到一个无界的 KLM 序列，将通过这种方法构造的新的 KLM 序列的集合定义为 Ξ。注意到 ζ 的个数是有限的，而则由引理4.2，任何一个特征序列 h 里 $T_i \setminus T_i'$ 的边的使用之和不超过 $|\xi|^{|\xi|}$，因此 h 一定也是其中某一个新 KLM 序列的特征序列，从而有 $L_\xi = \bigcup_{\xi' \in \Xi} L_{\xi'}$，并且有 $\forall\, \xi' \in \Xi$, $|\xi'| \le |\xi|^{|\xi|}$，由引理4.7该构造不会产生能构造更大的向量空间的 V，从而有 $\forall\, \xi' \in \Xi$, $\mathrm{rank}(\xi') <_{lex} \mathrm{rank}(\xi)$。

最后注意到 Ξ 里的 KLM 序列已经是强连通的了，忽略到其中不满足的 KLM 序列，由引理4.9，我们可以构造出有限集 $\Xi' = \bigcup_{\xi \in \Xi} \Xi_\xi$，其中的 KLM 序列都是可容忍可满足强连通的，并且有 $\forall\, \xi' \in \Xi'$, $|\xi'| \le |\xi|^{|\xi|^{|\xi|+1}}$，引理得证。 □

最后我们来说明如何将不固定的 KLM 序列分解成固定的 KLM 序列。事实上由引理4.3可知，可以在多项式时间内判断一个 KLM 序列是否是固定的，而转换成固定的 KLM 序列的方法也很简单，如果起点和终点不满足条件，则说明该序

列是不可满足的，另一方面对于其他的节点如果不满足该要求，其实是说明一条完全的路径不会经过这个节点，因此只要删除掉那个点即可。图4–6给了个直观的例子表示，下述引理则阐述了这一操作。

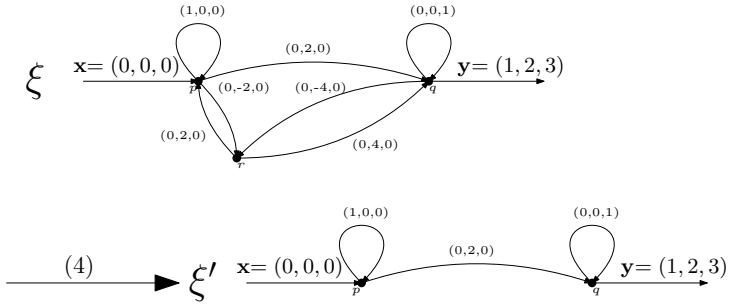


图 4–6 KLM 序列分解-(4)

Figure 4–6 Decomposition of KLM sequence(4)

引理 4.11 给定一个可满足的强连通的 KLM 序列 ξ，如果其不是固定的，则可以在 $2^{O(|\xi|^{|\xi|})}$ 的时间内构造出一组可容忍可满足的强连通序列集合 $\Xi = \{\xi'\}$ 满足 $L_\xi = \bigcup_{\xi' \in \Xi} L_{\xi'}$, $|\xi|^{|\xi|}$, $\mathrm{rank}(\xi') <_{lex} \mathrm{rank}(\xi)$。

证明 令 $\xi = (\mathbf{x}_0 V_0 \mathbf{y}_0)\mathbf{a}_1(\mathbf{x}_1 V_1 \mathbf{y}_1)\mathbf{a}_2 \ldots \mathbf{a}_n(\mathbf{x}_n V_n \mathbf{y}_n)$，假设 $\mathbf{x}_i V_i \mathbf{y}_i$ 不是固定的，令 $I_i \subseteq [d]$ 是 V_i 固定的维度，则由引理4.3，存在 $j \in I_i$ 使得其对应的偏移函数 f_j 存在下述三种状况之一：

- $\mathbf{x}_i[j], \mathbf{y}_i[j] \in \mathbb{N}$ 但是 $\mathbf{x}_i[j] - f_j(p_i) \neq \mathbf{y}_i[j] - f_j(q_i)$。
- $\mathbf{x}_i[j] \in \mathbb{N}$ 并存在 $r \in Q_i$ 使得 $\mathbf{x}_i[j] - f_j(p_i) + f_j(r) < 0$。
- $\mathbf{y}_i[j] \in \mathbb{N}$ 并存在 $r \in Q_i$ 使得 $\mathbf{y}_i[j] - f_j(q_i) + f_j(r) < 0$。

事实上若是第一种情况，则 $\mathbf{x}_i V_i \mathbf{y}_i$ 是不可满足的。考虑后面两种情况，显然 $r \neq \{p_i, q_i\}$，我们构造新的 $\mathbf{x}_i V_i' \mathbf{y}_i$ 来代替原来的部分，其中 V_i' 是将 V_i 中 r 和与 r 有关的所有的规则去掉以后得到的新的向量加法系统，其起点和终点保持不变，令新得到的 KLM 序列为 ξ'，显然我们有 $|\xi'| \leq |\xi|$, $\mathrm{rank}(\xi') <_{lex} \mathrm{rank}(\xi)$。下面我们证明 $L_{\xi'} = L_\xi$，只需说明 ξ 的任何一个完全的路径都不会经过 r 即可。事实上，令:

$$p^0(\mathbf{c}_0) \xrightarrow{\mathbf{a}^1} p^1(\mathbf{c}_1) \xrightarrow{\mathbf{a}^2} \cdots \xrightarrow{\mathbf{a}^k} p^k(\mathbf{c}_k) \tag{4–26}$$

是 $\mathbf{x}_i V_i \mathbf{y}_i$ 的一个成功的路径，其中 $p^0 = p_i$, $p^k = q_i$, $\mathbf{c}_0 \sqsubseteq \mathbf{x}_i$, $\mathbf{c}_k \sqsubseteq \mathbf{y}_i$。如果 $\mathbf{x}_i[j] \in \mathbb{N}$，则我们有 $\mathbf{c}_0[j] = \mathbf{x}_i[j]$，并有归纳可知对任意的 $l \in [k]$，有:

$$\mathbf{c}_l[j] = \mathbf{c}_0[j] - f_j(p_i) + f(p^k) = \mathbf{x}_i[j] - f_j(p_i) + f(p^k) \geq 0 \tag{4–27}$$

从而 $r \notin \{p^i | i \in [k]_0\}$，同理可证 $\mathbf{y}_i[j] \in \mathbb{N}$ 的情况，注意到上书可以在多项式时间内完成，我们再将 ξ' 转换成可容忍可满足的强连通序列，由引理4.8和引理4.9可知，可以在 $2^{O(|\xi|^{|\xi|})}$ 的时间内构造出一组可容忍可满足的强连通序列集合 $\Xi = \{\xi'\}$ 满足 $L_\xi = \bigcup_{\xi' \in \Xi} L_{\xi'}$，$|\xi|^{|\xi|}$，$\mathrm{rank}(\xi') <_{lex} \mathrm{rank}(\xi)$，引理得证。 □

至此我们已经处理完了所有标准 KLM 序列所具备的性质，由引理4.8至引理4.11可知，在执行图4–2中的 (1) – (4) 步时，每一次分解都是降低其的秩，因此其过程是可以终止的，即最终会获得一组标准 KLM 序列的集合 Ξ 满足 $L_\xi = \bigcup_{\xi' \in \Xi} L_\Xi$，将其分解视作一棵树，则存在一个初等函数 h 使得对于该分解树上父子节点的两个 KLM 序列的 ξ, ξ' 满足 $|\xi'| \leq h(|\xi|)$，即下述定理：

定理 4.2 令 $h(x) = x^{x^{x+1}}$，给定一个 KLM 序列 ξ，可以计算出一个有限的标准 KLM 序列集合 Ξ 满足 $L_\xi = \bigcup_{\xi' \in \Xi} L_\xi$，并且对于每个 $\xi' \in \Xi$，都存在一个分解序列 $\xi_1, \xi_2, \ldots, \xi_k$ 满足：

- $\xi_1 = \xi$，$\xi_k = \xi'$。
- 对于 $j \in [k-1]$ 有 $\mathrm{rank}(\xi_j) <_{lex} \mathrm{rank}(\xi_{j+1})$，$|\xi_{j+1}| \leq h(|\xi_j|)$。

4.3.2 分解成正则 KLM 序列

本节将介绍如何分解成一个正则 KLM 序列。定理4.2说明了如何将一个 KLM 序列 ξ 分解成标准 KLM 序列，因此其距离正则 KLM 序列只欠一个可泵性的性质。由引理4.4可知，如果其不是可泵的，则一定存在某一维有上界。因此对于不可泵的标准 KLM 序列，其想法核心是降维，即把有上界的那一维编码进状态里，即将那一维变成向量加法系统中固定的一维，从而消除了原先不可泵的维数。注意到其会导致标准性被破坏，因此需要继续使用定理4.2将其分解为标准 KLM 序列后再进行可泵性的判定。

下面来具体的描述该过程。令 $\xi = (\mathbf{x}_0 V_0 \mathbf{y}_0)\mathbf{a}_1(\mathbf{x}_1 V_1 \mathbf{y}_1)\mathbf{a}_2 \ldots \mathbf{a}_n(\mathbf{x}_n V_n \mathbf{y}_n)$ 是一个标准 KLM 序列，如果其不是可泵的，则存在一个部分 $\mathbf{x}_i V_i \mathbf{y}_i$ 不是可泵的，则由引理4.4可知，存在不固定的 $r \in [d]$ 使得其关于 r 是不可泵的，不妨令其是前不可泵的，则有 $\mathbf{x}_i[r] = A < B$，这里 $B = (2|\xi|)^{1+d^d}$。令 (B) 表示 $[B] \cup \{0, \omega\}$，下面定义向量加法系统 $V_i^r[b] = (Q_i^r, T_i^r, p_i^A, q_i^b)$：

- $Q_i^r = \{(p, n) | p \in Q_i,\ n \in (B)\}$。
- $T_i^r = \{((p, m), \mathbf{a}, (q, n)) | (p, \mathbf{a}, q) \in T_i,\ n = m + \mathbf{a}[r] \vee m = n = \omega \vee m + \mathbf{a}[r] > B, n = \omega,\ m = \omega \Rightarrow p \neq \omega\}$。
- $p_i^A = (p_i, A)$，$q_i^b = (q_i, b)$。

令 $\xi_i = \mathbf{x}_i V_i \mathbf{y}_i$，构造集合 Ξ_i^r:

$$\Xi_i \stackrel{\text{def}}{=} \{\xi_i^{r,b} | \xi_i^{r,b} = \mathbf{x}_i V_i^r[b] \mathbf{y}_i \wedge b \in (B)\}$$

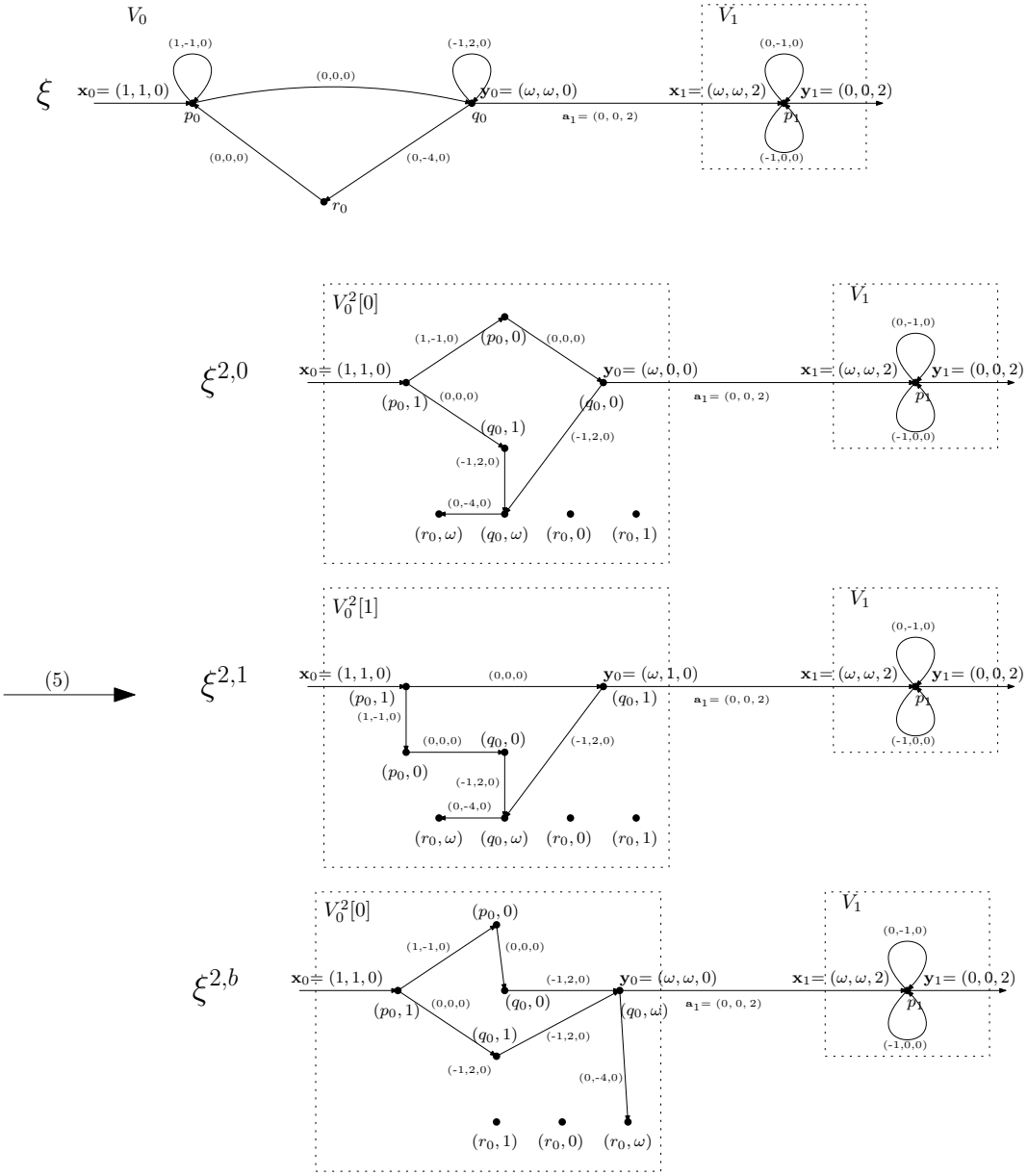


图 4–7 KLM 序列分解-(5)

Figure 4–7 Decomposition of KLM sequence(5)

图4–7给了个直观的例子解释，显然其中标准 KLM 序列 ξ 里 V_0 对于 $\{1,2\}$ 是前不可泵的，对其第二维作降维之后便分解成了 $\xi^{2,0}$, $\xi^{2,1}$, $\xi^{2,\omega}$ 这三条新的 KLM 序列。下面证明 Ξ_i^r 会是 ξ_i 的一个分解。

引理 4.12 对于任意 $b \in (B)$，有 $\mathrm{rank}(\xi_i^{r,b}) <_{lex} \mathrm{rank}(\xi_i)$。

证明 令 $\mathbb{V}_i$ 表示 V_i 中所有圈的增量张成的向量空间，令 $\dim(\mathbb{V}_i) = k$。注意到 V_i 是强连通的，因此对于 $t \in T_i$ 有 $\mathbb{V}_{V_i}(t) \subseteq \mathbb{V}_i$。并且由于 r 不是 V_i 中固定的，因此一定存在 $\mathbf{v} \in \mathbb{V}_i$ 满足 $\mathbf{v}[r] > 0$，并且由引理4.6有 $\mathrm{rank}(\xi_i)[d+1-k] = |T|$。

对于 $t' = ((p,m), \mathbf{a}, (q,n)) \in V_r^i[b]$，分三种情况讨论：

- $m,\ n \neq \omega$。注意到这种边形成的圈 θ 一定满足 $\Delta(\theta)[r] = 0$，因此由其构成的向量空间 $\mathbb{V}_{V_i^r[b]}(t')$ 一定满足：$\mathbb{V}_{V_i^r[b]}(t') \subsetneq \mathbb{V}_i$。
- $m \neq \omega,\ n = \omega$。注意到这样子的边不会在 $V_i^r[b]$ 有圈，因此 $\mathbb{V}_{V_i^r[b]}(t') = \{\mathbf{0}\}$。
- $m = \omega$，由定义 $n = \omega$。注意到此时 $p \neq p_i$，因此最多有 $|T|-1$ 条这样的边，从而 $\mathrm{rank}(\xi_i^{r,b})[d+1-k] \leq |T|-1$。

综上所述，有 $\mathrm{rank}(\xi_i^{r,b}) <_{lex} \mathrm{rank}(\xi_i)$，引理得证。 □

引理 4.13 令 R 是 ξ_i 所有不前可泵的维度的集合，则存在 $r \in R$ 使得 $L_{\xi_i} = \bigcup_{\xi' \in \Xi_i} L_{\xi_i}$。

证明 $\supseteq$ 方向是显然的，下面来证 $\subseteq$。令 $\pi = \mathbf{a}^1 \dots \mathbf{a}^k$ 是 ξ_i 中一个成功的路径，则存在 $\mathbf{m}_0, \dots \mathbf{m}_k \in \mathbb{N}^d$ 满足：

$$p^0(\mathbf{m}_0) \xrightarrow{\mathbf{a}^1} \cdots \xrightarrow{\mathbf{a}^k} p^k(\mathbf{m}_k) \tag{4–28}$$

其中 $p^0 = p_i,\ p^k = q_i$。如果存在 $r \in R$ 使得对于任意的 $l \in [k]$ 有 $\mathbf{m}_l[r] \leq B$，则 $\pi \in L_{\xi_i^r[\mathbf{m}_k[r]]}$。反之，对任意的 $r \in R$ 存在最小 $l_r \in [k]$ 有 $\mathbf{m}_{l_r}[r] > B$，令 $l = \max_{r \in R} l_r$，则存在 $r \in R$ 使得，对于任意的 $j \in [l-1]$ 有 $\mathbf{m}_j[r] \leq B$。接下来分两种情况讨论：

- 如果对于任意的 $j \in \{l, \dots, k\}$ 有 $p^j \neq p_i$，则有 $\pi \in L_{\xi_i^r[\omega]}$。
- 如果存在 $j \in \{l, \dots, k\},\ p^j = p_i$，则说明存在一条 p^j 到 p_i 的路径，令其最短的路径为 θ，显然有 $|\theta| < |Q_i|$。令 $\mathbf{c}_i = \mathbf{m}_i|_R$，则有：

  $$p^0(\mathbf{c}_0) \xrightarrow{\mathbf{a}^1} \cdots \xrightarrow{\mathbf{a}^l} p^l(\mathbf{c}_l) \tag{4–29}$$

  由引理4.4存在一条路径 σ 和 $\mathbf{c}$ 满足 $p^0(\mathbf{c}_0) \xrightarrow{\sigma} p^l(\mathbf{c})$，并且有对于任意 $j \in [d]$ 有 $\mathbf{c}[j] > 2|\xi| - |V_i|$。从而有：

  $$p^0(\mathbf{c}_0) \xrightarrow{\sigma\theta} p^0(\mathbf{c}') \tag{4–30}$$

  其中对于任意的 $r \in R$ 有 $\mathbf{c}'[r] \in \mathbb{N}$，$|\sigma\theta| < B^{d^3}$。注意到固定的维度并不需要考虑，而对于其余的维度，其都是前可泵的，由定义存在路径 ζ 满足：

  $$p^0(\mathbf{x}_i) \xrightarrow{\zeta} p^0(\mathbf{x}') \tag{4–31}$$

其中 $\mathbf{x}' \leq \mathbf{x}_i$，并且对于前可泵的 j 有 $\mathbf{x}'[j] > \mathbf{x}_i[j]$。因此由单调性，有：

$$p^0(\mathbf{x}_i) \xrightarrow{\zeta^{|\xi|\cdot B d^3}\sigma\theta} p^0(\mathbf{x}'') \tag{4–32}$$

其中 $\mathbf{x}'' \geq \mathbf{x}$，并且对于 $r \in R$ 有 $\mathbf{x}''[r] > \mathbf{x}[r]$，与 R 的定义矛盾，从而引理得证。 □

由上述两个引理不难得到对于非正则的标准 KLM 序列的分解方法，即下述定理：

定理 4.3　令 $g(x) = (2x)^{2+d^d}$，给定一个非正则的标准 KLM 序列 ξ，可以在 $2^{O(g(|\xi|))}$ 的时间内分解成一组有限的 KLM 序列集合 $\Xi = \{\xi'\}$ 满足 $L_\xi = \bigcup_{\xi' \in \Xi} L_\xi$，$|\xi'| \leq g(|\xi|)$，$\mathrm{rank}(\xi') <_{lex} \mathrm{rank}(\xi)$。

注　Leroux 在 [79] 里对该算法的阐述与这不尽相同，其将可满足可容忍强连通的 KLM 序列称之为纯净的 KLM 序列 (clean KLM sequence)，原因在于引理4.8和4.9不会对别的性质造成破坏，因此只需一个确定的步骤就能将 KLM 序列分解成纯净的 KLM 序列，而对于不固定、不无界、不可泵的 KLM 序列的分解，其可能会破坏其他的性质，比如强连通性等。因此其实对于后三个性质而言，保障其终止性的原因是秩的不断减少，这也是 Leroux 做纯净 KLM 序列和正则 KLM 序列分类的原因，在其描述中分解是两步进行的。

这里本文将固定性和无界性都放在前面考虑 (文章称之为标准 KLM 序列) 的原因是在于我们认为这几个步骤是对向量加法系统在做分解，而后面对于不可泵的处理其实是一种降维操作。做出这一改变的初衷是希望通过这一细分去获取向量加法系统更好的上界，但最新的结果表明 $\mathbf{F}_\omega$ 上界已经是最好的结果。不过，这一想法仍有助于研究固定维的向量加法系统的可达性问题，原因在于其实该想法给出了相邻维度向量加法系统可达性关系可能的一层联系。

4.3.3　一个例子

下面用 Leroux 在 [79] 介绍该算法时的例子来完整的运行一遍该算法。图2–1中的 V 就是一个三维带状态的向量加法系统，我们来考察是否存在一条路径从 $p(0,0,2)$ 到 $q(1,1,0)$，不难验证，所有符合的路径集合 L_V 为：

$$L_V = \{\pi | \pi = \mathbf{a}_1^{2+3n}\mathbf{a}_3\mathbf{a}_6^{1+4n}\mathbf{a}_7\mathbf{a}_8^{1+2n}\mathbf{a}_9, n \in \mathbb{N} \vee \pi = \mathbf{a}_1^{2+3n}\mathbf{a}_3\mathbf{a}_6^{4n}\mathbf{a}_7\mathbf{a}_8^{1+2n}\mathbf{a}_9\mathbf{a}_6, n \in \mathbb{N}\}$$

其等价于求 KLM 序列 $\xi = (0,0,2)V(1,1,0)$ 的完全的路径，即 $L_\xi = L_V$。下面依据上面的算法来对 ξ 进行分解。

注意到 V 不是强连通的，因此由引理4.8, 其可以分解成如下两个 KLM 序列：

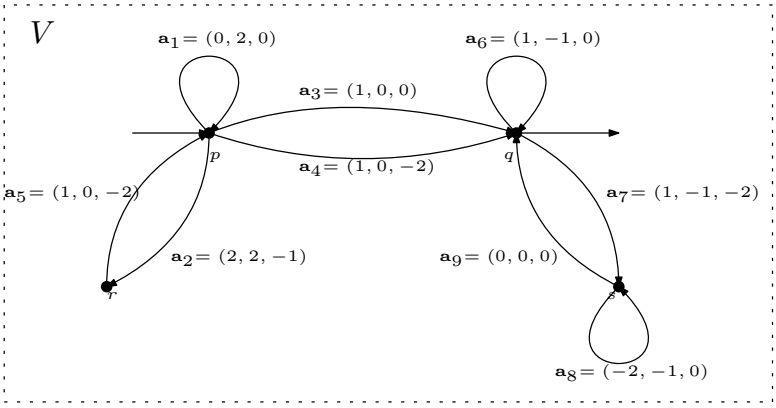


图 4–8 向量加法系统 V

Figure 4–8 Vector Addition System with State V

- $\xi_1' = (0,0,2)V_1(\omega,\omega,\omega)\mathbf{a}_3(\omega,\omega,\omega)V_2(1,1,0)$。
- $\xi_2' = (0,0,2)V_1(\omega,\omega,\omega)\mathbf{a}_4(\omega,\omega,\omega)V_2(1,1,0)$。

其中 V_1, V_2 如图4–9所示。由引理4.9，上述两个强连通的 KLM 序列可以转换成如下可容忍的强连通 KLM 序列：

- $\xi_1 = (0,0,2)V_1(0,\omega,2)\mathbf{a}_3(1,\omega,2)V_2(1,1,0)$。
- $\xi_2 = (0,0,2)V_1(0,\omega,2)\mathbf{a}_4(0,\omega,0)V_2(1,1,0)$。

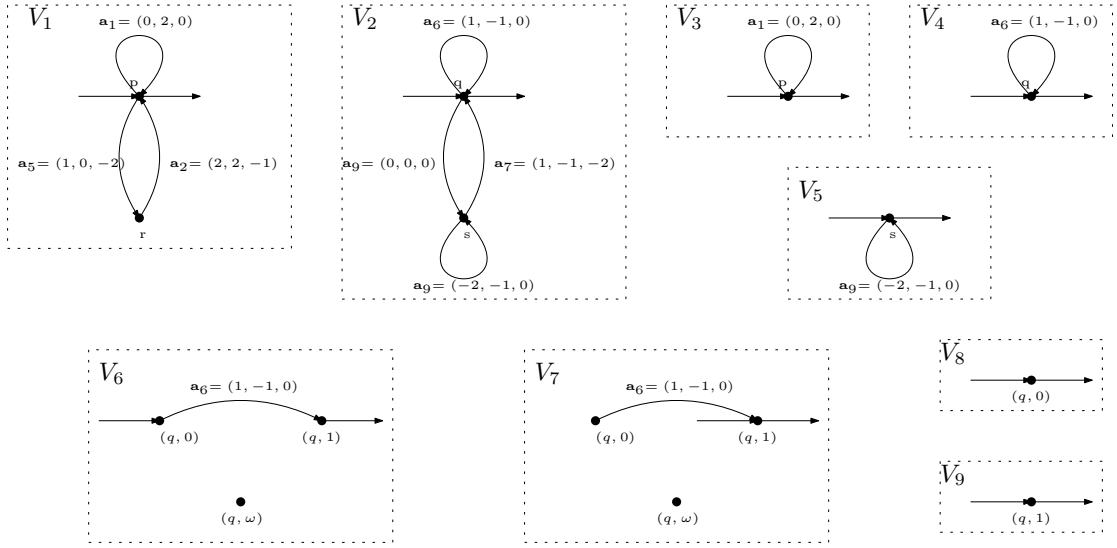


图 4–9 KLM 序列 ξ 的分解

Figure 4–9 Decomposition of KLM sequence ξ

注意到尽管可以看出 ξ_2 中不存在完全的路径，但此时其还是可满足的。注意到无论在 ξ_1', ξ_2' 中边 $\mathbf{a}_7$, $\mathbf{a}_9$ 的使用次数都是有界的，因此由引理4.10可以将其分解

成如下的 KLM 序列：

- $\xi_3 = (0,0,2)V_1(0,\omega,2)\mathbf{a}_3(1,\omega,2)V_4(\omega,\omega,\omega)\mathbf{a}_7(\omega,\omega,\omega)V_5(\omega,\omega,\omega)\mathbf{a}_9(\omega,\omega,\omega)V_4(1,1,0)$。
- $\xi_4 = (0,0,2)V_1(0,\omega,2)\mathbf{a}_4(0,\omega,0)V_4(\omega,\omega,\omega)\mathbf{a}_7(\omega,\omega,\omega)V_5(\omega,\omega,\omega)\mathbf{a}_9(\omega,\omega,\omega)V_4(1,1,0)$。

此时 ξ_4 是不可满足的，因此可以忽略。将 ξ_3 变成可容忍的 KLM 序列：

- $\xi_3^1 = (0,0,2)V_1(0,\omega,2)\mathbf{a}_3(1,\omega,2)V_4(\omega,\omega,2)\mathbf{a}_7(\omega,\omega,0)V_5(0,2,0)\mathbf{a}_9(0,2,0)V_4(1,1,0)$。
- $\xi_3^2 = (0,0,2)V_1(0,\omega,2)\mathbf{a}_3(1,\omega,2)V_4(\omega,\omega,2)\mathbf{a}_7(\omega,\omega,0)V_5(1,1,0)\mathbf{a}_9(1,1,0)V_4(1,1,0)$。

考虑 $(0,2,0)V_4(1,1,0)$ 和 $(1,1,0)V_4(1,1,0)$，其前两维都是不固定的，并且都是不可泵的，将其第二维编入状态得到：

- $\xi_5 = (0,0,2)V_1(0,\omega,2)\mathbf{a}_3(1,\omega,2)V_4(\omega,\omega,2)\mathbf{a}_7(\omega,\omega,0)V_5(0,2,0)\mathbf{a}_9(0,2,0)V_6(1,1,0)$。
- $\xi_6 = (0,0,2)V_1(0,\omega,2)\mathbf{a}_3(1,\omega,2)V_4(\omega,\omega,2)\mathbf{a}_7(\omega,\omega,0)V_5(1,1,0)\mathbf{a}_9(1,1,0)V_7(1,1,0)$。

最后对 $(0,2,0)V_6(1,1,0)$ 和 $(1,1,0)V_7(1,1,0)$ 做强连通性的处理后得到了两条正则的 KLM 序列：

- $\xi_7 = (0,0,2)V_1(0,\omega,2)\mathbf{a}_3(1,\omega,2)V_4(\omega,\omega,2)\mathbf{a}_7(\omega,\omega,0)V_5(0,2,0)\mathbf{a}_9(0,2,0)V_8(1,1,0)\mathbf{a}_6(1,1,0)V_9(1,1,0)$。
- $\xi_8 = (0,0,2)V_1(0,\omega,2)\mathbf{a}_3(1,\omega,2)V_4(\omega,\omega,2)\mathbf{a}_7(\omega,\omega,0)V_5(1,1,0)\mathbf{a}_9(1,1,0)V_9(1,1,0)$。

由定理4.1可知如果 ξ_7, ξ_8 存在完全的路径，则会存在着长度不超过 x^{3x} 的完全的路径，这里 $x = \max\{|\xi_7|, |\xi_8|\}$，分解完成。

4.3.4　算法分析

现在来分析一下上述算法的复杂性。由上述分解算法，对于 KLM 序列 ξ 的分解，我们可以获得一个秩不停减小的 KLM 序列串 $\xi_1\xi_2\ldots$，由定理4.2和定理4.3可得，该序列满足：存在一个函数 $H(x) = \max\{x^{x^{x+1}}, (2x)^{2+d^d}\}$ 使得 $|\xi_j| \leq H(|\xi_{j-1}|)$，注意到 $(\mathbb{N}^{d+1}, <_{lex})$ 是一个良序，因此由长度控制定理2.3，可知该序列的长度 $L(\xi)$ 至多为 $H_{\omega^{d+1}}(n)$，从而最终的 KLM 序列 ξ' 的长度为：

$$|\xi'| \leq H^{L(\xi)}(|\xi|) \leq H^{\omega^{d+1}}(|\xi|) \tag{4–33}$$

因此由定理4.1可知，ξ 存在一条长度不超过 $l(H^{\omega^{d+1}}(|\xi|))$ 的完全的路径，其中 $l(x) = x^{3x}$，从而有：

定理 4.4 向量加法系统的可达性问题是在 $\mathbf{F}_\omega$ 的，特别的对于 d 维的可达性问题，其是在 $\mathbf{F}_{d+4}$ 的。

证明 给定 d 维向量加法系统 V 和两个格局 $p(\mathbf{u})$ 和 $q(\mathbf{v})$，记 $n = |V| + ||\mathbf{u}||_1 + ||\mathbf{v}||_1$，由上述讨论可知，如果存在一条 $p(\mathbf{u})$ 到 $q(\mathbf{v})$ 的路径，则存在一条长度不超过 $l(H^{\omega^{d+1}}(n))$ 的路径，因此该问题在 $\mathbf{F}_{H,\omega^{d+1}}$ 中，因为由定理2.1我们只需要遍历所有长度不超过 $l(H^{\omega^{d+1}}(n))$ 的路径即可，同时注意到 $H \in \mathscr{F}_{<3}$，再由定理2.2可知，该问题在 $\mathbf{F}_{d+4}$，定理得证。 □

4.4 本章小结

本章讨论了可达性问题最重要的算法-KLMST 分解算法。Leroux 对于该算法的研究让大家能够更容易的理解该算法的运作方式，其也最终证明了最好的上界。本文则给出了一个该算法新的理解方式，即分解-降维的循环，我们认为不可泵性其实就是一种可降维的体现，因为这相当于可以通过枚举来将这一维编码进状态，这说明了高维的可达性问题某种意义下是可以转化成低维的可达性问题，因此我们相信其对于固定维度的可达性问题的上界有着一定的帮助。

第五章　可达性算法-Presburger 不变量

本章将介绍 Leroux 在[73, 145, 150-151] 中提出的一种跟 KLMST 分解算法不同的可达性算法-Presburger 不变量算法。其想法是，本身可达性问题是一个半可判定的算法，即如果一个点是可达的，那么就可以通过不断枚举的方式找到这样一条路径。因此只需要寻找到一个点不可达的证据 (witness)。

如果一个向量加法系统满足了一些特定的条件，那么就可以比较容易的寻找到这样的不可达证据。半线性 (semi-linear)[152], 也就是 Presburger 可定义的集合是人们希望可达集合能满足的性质，研究者们也在满足可达集是半线性的向量加法系统里获得了很多的进展。Hauschildt 在 [153] 证明了判定可达集是否是 Presburger 可定义的是可判定的。在一些低维的向量加法系统中，由于其可达集是 Presburger 可定义的，因此产生了许多新的结论[76-77]，具体会在章节七进一步阐述。

半线性的一个好处是可以找到一个所谓的 Presuburger 分离对，即如果一个格局是不可达的，则可以找到两个 Presburger 集合 X, Y，使得 X 包含了所有初始格局可以到达的格局，而 Y 包含了所有可以到达目标格局的格局，而这两个集合是不相交的，这样就完成了对不可达的证据的一个寻找。但是 Hopcroft 在 [40] 证明了 5 维以上的向量加法系统的可达集合不一定是半线性的。幸运的是，Leroux 发现了一个关于半线性的拓展性质[150]，称为几乎半线性 (almost semi-linear)，满足该性质的集合也同样有上述的性质，并且其进一步证明了向量加法系统的可达集是几乎半线性的[73, 150]，最终给出了一个完全不依赖于 KLMST 分解的新的可判定算法。

本章下面将介绍这个可判定算法。第一节将介绍 Presburger 不变量技术，我们将说明如果一个关系是几乎半线性的，那么就会存在一个 Presburger 分离对能够将不在这个关系里的两个元素分离开来，第二节则将证明向量加法系统的可达集是几乎半线性的，从而完成其可达性问题的可判定算法。第三节则是本章小结。

5.1　Presburger 不变量

本节将介绍 Presburger 不变量这一技术，首先来介绍一些集合上的几何性质。

5.1.1　锥集

我们先来介绍一下锥集的相关概念以及其的一些性质。一个锥集 (conic set) 简单来说是在 $\mathbb{Q}^d$ 上的一个子集并且满足加法是封闭的，其严格定义如下：

定义 5.1 (锥集) 一个集合 $C \subseteq \mathbb{Q}^d$ 如果满足：1. $\mathbf{0}_d \in C$, 2. 对于 $\mathbf{c}_1, \mathbf{c}_2 \in C$ 有 $\mathbf{c}_1 + \mathbf{c}_2 \in C$, 3. 对于 $q \in \mathbb{Q}_{\geq 0}, \mathbf{c} \in C$ 有 $q\mathbf{c} \in C$，则称 C 是一个锥集。

注 锥集的第二个性质写也可以改写成 $C + C \subseteq C$，而第三个性质可以改写写成 $\mathbb{Q}_{\geq 0}C \subseteq C$。

对于一个锥集 C，如果存在 $\mathbf{c}_1, \mathbf{c}_2 \ldots \mathbf{c}_k \in C$ 使得对于任何一个 $\mathbf{c} \in C$，都存在 $q_1, \ldots q_k \in \mathbb{Q}_{\geq 0}$ 满足 $\mathbf{c} = \sum_{i=1}^{k} q_i \mathbf{c}_i$，则称 C 是有限生成的 (finitely generated)，如果 C 是可以由 $FO(\mathbb{Q}, +, \leq, 0)$ 可定义的，则称 C 是可定义的 (definable)。显然一个有限生成的锥集也会是一个可定义的锥集。下面给出几个例子来帮助理解这些定义。

例 5.1 考虑如下三个锥集：

- $C_1 = \mathbb{Q}_{\geq 0}(1, 1) + \mathbb{Q}_{\geq 0}(1, 0)$.
- $C_2 = \{(q_1, q_2) \in \mathbb{Q}^2_{>0} \mid q_1 \geq q_2\}$.
- $C_3 = \{(q_1, q_2) \in \mathbb{Q}^2_{>0} \mid q_1 \geq \sqrt{2} q_2\}$.

不难验证，C_1 是有限生成的，C_2 不是有限生成的但是是可定义的，C_3 都不是可定义的。

接下来介绍有关有限生成的锥集和可定义的锥集所具有的性质。为此，首先补充一下向量空间 (vector space) 和拓扑闭包 (topological closure) 的概念。

一个向量空间 $V \subseteq \mathbb{Q}^d$ 满足 $\mathbf{0}_d \in V$, $V + V \subseteq V$, $\mathbb{Q}V \subseteq V$。对一个集合 $X \subseteq \mathbb{Q}^d$，如果 V 满足对于其中任何一个元素 $\mathbf{v}$, 都存在 $\mathbf{x}_1, \ldots \mathbf{x}_k \in X$ 和 $\lambda_1, \ldots, \lambda_k \in \mathbb{Q}$ 满足：

$$\mathbf{v} = \lambda_1 \mathbf{x}_1 + \ldots + \lambda_k \mathbf{x}_k \tag{5–1}$$

则称向量空间 V 是由 X 生成的，特别的如果 X 是一个锥集，则有 $V = X - X$。显然任何一个向量空间 $V \subseteq \mathbb{Q}^d$ 可由最多 d 个向量生成。定义 V 的秩 $rank(V)$ 为生成 V 所需要的最少的向量个数，即 $rank(V) \stackrel{\text{def}}{=} \min |X|$。

给定一个集合 $X \subseteq \mathbb{Q}^d$，其拓扑闭包 $\overline{X}$ 定义为

$$\overline{X} \stackrel{\text{def}}{=} \{\mathbf{r} | \forall \epsilon > 0, \exists \mathbf{x} \in X, ||\mathbf{r} - \mathbf{x}||_\infty < \epsilon\}$$

如果 $X = \overline{X}$，则称 X 是闭的 (closed)，也称 X 是闭集。自然的，向量空间 V 是一个闭集，如果 C 是一个锥集，则 $\overline{C}$ 也是一个锥集。如果 $\mathbf{c} \in C$ 满足存在一个 $\epsilon \in \mathbb{Q}_{>0}$ 对所有的 $\mathbf{v} \in C - C$, $||\mathbf{v}||_\infty < \epsilon$ 都有 $\mathbf{c} + \mathbf{v} \in C$，则称 $\mathbf{c}$ 是一个内点 (interior)，用 $\text{int}(C)$ 表示 C 中所有内点的集合。

下述引理介绍了有限生成的锥集的对偶性质。

引理 5.1 (duality[133])　令 $V \subseteq \mathbb{Q}^d$ 是一个向量空间。一个锥集 $C \subseteq V$ 是有限生成的当且仅当存在 $\mathbf{h}_1, \ldots \mathbf{h}_k \in V \setminus \{\mathbf{0}_d\}$ 满足：

$$C = \bigcap_{j=1}^{k} \{\mathbf{v} \in V | \sum_{i=1}^{d} \mathbf{h}_j[i]\mathbf{v}[i] \geq 0\} \tag{5–2}$$

并且如下等式成立当且仅当 V 可以由 C 生成：

$$\text{int}(C) = \bigcap_{j=1}^{k} \{\mathbf{v} \in V | \sum_{i=1}^{d} \mathbf{h}_j[i]\mathbf{v}[i] > 0\} \tag{5–3}$$

例 5.2　下面用一个例子来解释一下对偶性质。考虑如下两个在 $\mathbb{Q}^2$ 内的锥集

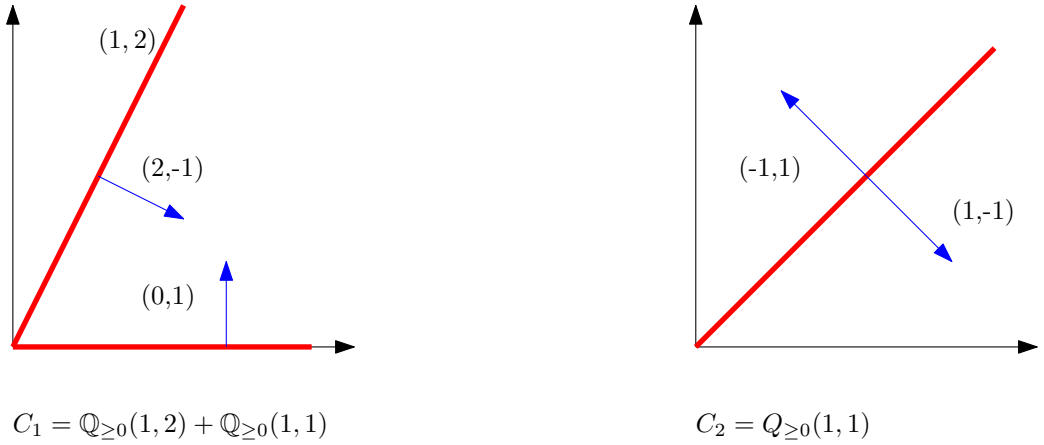


图 5–1　一个对偶性质的例子

Figure 5–1　An example of duality property

C_1, C_2。令 $H_1 = \{(2,-1),(0,1)\}$, $H_2 = \{(-1,1),(1,-1)\}$，不难发现：

- $C_1 = \{\mathbf{v} \in \mathbb{Q}^2 | 2 \cdot \mathbf{v}[1] + (-1) \cdot \mathbf{v}[2] \geq 0,\ 0 \cdot \mathbf{v}[1] + 1 \cdot \mathbf{v}[2] \geq 0\}$.
- $C_2 = \{\mathbf{v} \in \mathbb{Q}^2 | 1 \cdot \mathbf{v}[1] + (-1) \cdot \mathbf{v}[2] \geq 0,\ -1 \cdot \mathbf{v}[1] + 1 \cdot \mathbf{v}[2] \geq 0\}$

另一方面，注意到 C_1 可以生成 $\mathbb{Q}^2$ 而 C_2 不能，相对应的也有：

- $\text{int}(C_1) = \{\mathbf{v} \in \mathbb{Q}^2 | 2 \cdot \mathbf{v}[1] + (-1) \cdot \mathbf{v}[2] > 0,\ 0 \cdot \mathbf{v}[1] + 1 \cdot \mathbf{v}[2] > 0\}$.
- $\text{int}(C_2) \neq \{\mathbf{v} \in \mathbb{Q}^2 | 1 \cdot \mathbf{v}[1] + (-1) \cdot \mathbf{v}[2] > 0,\ -1 \cdot \mathbf{v}[1] + 1 \cdot \mathbf{v}[2] > 0\}$.

接下来介绍可定义的锥集的一些性质。显然有限生成的锥集都是可定义的，但不是所有可定义的锥集都是有限生成的，比如例子5.1中的 C_2。但是下述两个引理将说明可定义的锥集是可以用有限生成的锥集所来刻画的。作为帮助，下面首先阐述可定义集合的闭包是一堆有限生成的集合的并集这一性质。

引理 5.2 (Leroux[73]) 给定锥集 $X \subseteq \mathbb{Q}^d$，如果其是由 $FO(\mathbb{Q}, +, \leq, 0)$ 可定义的，则存在有限生成的集合 $X_1, \ldots X_k$ 满足：$\overline{X} = \bigcup_{i=1}^{k} X_k$。

证明 通过量词消去，存在 $A_1, \ldots A_k \subseteq \mathbb{Q}^d \times \{\geq, >\}$，使得 X 可以表示为：

$$X = \bigcup_{j=1}^{k} (\bigcap_{(\mathbf{h},\#) \in A_j} \{\mathbf{v} \in \mathbb{Q}^d | \sum_{i=1}^{d} \mathbf{v}[i] \cdot \mathbf{h}[i] \ \# \ 0\}) \tag{5–4}$$

记 $X_j = \bigcap_{(\mathbf{h},\#) \in A_j} \{\mathbf{v} \in \mathbb{Q}^d | \sum_{i=1}^{d} \mathbf{v}[i] \cdot \mathbf{h}[i] \ \# \ 0\}$，令：

$$R_j = \bigcap_{(\mathbf{h},\#) \in A_j} \{\mathbf{v} \in \mathbb{Q}^d | \sum_{i=1}^{d} \mathbf{v}[i] \cdot \mathbf{h}[i] \ \geq \ 0\} \tag{5–5}$$

记 $R = \bigcup_{j=1}^{k} R_j$，由引理5.1 R_j 是有限生成的，并且 R 是闭集。接下来只需说明 $\overline{X} = R$。$\overline{X} \subseteq R$ 是比较简单的，注意到 $X_j \subseteq R_j$ 以及 R 是闭集即可。另一方面，设 $\mathbf{r} \in R$，令 $\mathbf{x}_j \in X_j$，则有 $\mathbf{r} + \mathbb{Q}_{\geq 0}\mathbf{x}_j \in X_j$，因此由定义 $\mathbf{r} \in \overline{X}$，即 $R \subseteq \overline{X}$，命题得证。 □

由引理5.2，很容易猜想是不是可以用其闭包是有限生成的这一定义去代替可定义。遗憾的是，这一猜想仅仅在 ≤ 2 维的时候成立。考察如下的一个在三维空间里的锥集 C：

$$C = \{(q_1, q_2, q_3) \in \mathbb{Q}_{\geq 0}^2 \times \mathbb{Q}_{>0} | q_3 = 0 \Rightarrow q_2 \leq \sqrt{2} q_1\} \tag{5–6}$$

显然 $\overline{C} = \mathbb{Q}_{\geq 0}^3$ 是一个有限生成的锥集，但是稍后将会看到 C 不是可定义的。

接下来刻画可定义锥集。下面的引理将说明，一个锥集是可定义的，当且仅当其和任何一个向量空间的交的闭包都是有限生成的。

引理 5.3 (Leroux[73]) 一个锥集 $C \subseteq \mathbb{Q}^d$ 是可定义的当且仅当对于任何一个向量空间 V，与其的交集的闭包 $\overline{C \cap V}$ 都是有限生成的。

证明 先证 $\Rightarrow$ 方向。令 $X = C \cap V$，下面证明 $\overline{X}$ 是有限生成的。由引理5.2有 $\overline{X} = \bigcup_{j=1}^{k} C_j$，其中 C_j 是有限生成的。注意到 $\overline{X}$ 是闭的，因此 $\sum_{j=1}^{k} C_j \subseteq \overline{\sum_{j=1}^{k} C_j} = \overline{X}$；另一方面由于 $\mathbf{0}_d \in C_j$，因此对于 $j \in [k]$ 均有 $C_j \subseteq \sum_{j=1}^{k} C_j$，即 $\overline{X} \subseteq \sum_{j=1}^{k} C_j$，所以有 $\overline{X} = \sum_{j=1}^{k} C_j$，从而 $\overline{X}$ 是有限生成的。

再来证 $\Leftarrow$ 方向。记 $r(C) = rank(C - C)$，对 $r(C)$ 做归纳，$r = 0$ 时 $C = \{\mathbf{0}_d\}$ 命题显然成立。假设命题对于 $< r$ 成立，即对于满足 $r(C) < r$ 的锥集 C，如果对于任何一个向量空间 V，其交集的闭包都是有限生成的，则 C 是可定义的。现在考察 $r(C) = r$ 的情况，由条件令 $V = \mathbb{Q}^d$，则有 $\overline{C}$ 是有限生成的。令 $W = C - C$，由引理5.1存在 $\mathbf{w}_1, \ldots \mathbf{w}_k \in W \setminus \{\mathbf{0}_d\}$ 满足：

- $C = \cap_{j=1}^{k}\{v \in W\} \mid \sum_{i=1}^{d} \mathbf{w}_j[i]\mathbf{v}[i] \geq 0\}$.
- $\text{int}(C) = \cap_{j=1}^{k}\{v \in W\} \mid \sum_{i=1}^{d} \mathbf{w}_j[i]\mathbf{v}[i] > 0\}$.

令 $W_j = \{w \in V \mid \sum_{i=1}^{d} \mathbf{w}[i]\mathbf{w}_j[i] = 0\}$，注意到 $\mathbf{w}_j \notin W_j$ 因此有 $rank(W_j) < rank(W) = r$。C 可以被分解成如下形式：

$$C = \text{int}(C) \cup (\bigcup_{j=1}^{k} C \cap W_j) \tag{5–7}$$

考虑 $C_j = C \cup W_j$，有 $r(C_j) < r$，并且对于任何一个向量空间 V，$C_j \cap V$ 的闭包都是有限生成的，因此由归纳假设 $C \cap W_j$ 是可定义的，从而 C 是可定义的，归纳假设成立，命题得证。 □

再回过头来看下在5–6里所定义的锥集 C，令 $V = \{(q_1, q_2, q_3) \in \mathbb{Q}^3 \mid q_3 = 0\}$，则有:

$$C' = C \cap V = \{(q_1, q_2, q_3) \in \mathbb{Q}_{\geq 0}^2 \times \mathbb{Q}_{>0} \mid q_3 = 0,\ q_2 \leq \sqrt{2}q_1\}$$

显然 C 不是有限生成的，因而根据引理5.3 C 不是可定义的。事实上两个可定义的锥集的交依旧是可定义的，然而两个闭包是有限生成的锥集的交的闭包却不一定有有限生成的性质，这也是提出可定义这一概念的原因。

5.1.2　Presburger 集

上一节介绍了在 $\mathbb{Q}^d$ 上锥集的一些特征。这一节将介绍在如果在 $\mathbb{Z}^d$ 上的子集满足相似的条件（这样的集合为周期集 (periodic set)），周期集上会有怎么样的性质。

定义 5.2 (周期集)　一个集合 $P \subseteq \mathbb{Z}^d$ 如果满足：1. $\mathbf{0}_d \in P$, 2. 对于 $\mathbf{p}_1, \mathbf{p}_2 \in P$ 有 $\mathbf{p}_1 + \mathbf{p}_2 \in P$, 3. 对于 $n \in \mathbb{N}_{\geq 0}, \mathbf{p} \in P$ 有 $n\mathbf{p} \in P$，则称 P 是一个周期集。

考虑到一个定义在 $\mathbb{Z}^d$ 上的关系可以视作一个 $\mathbb{Z}^{2d}$ 的集合，因此也可以同样的定义周期关系。不难发现，其有如下性质：

引理 5.4　令 $P_1, P_2 \subseteq \mathbb{Z}^d$ 是一个周期集，R_1, R_2 是 Z^d 上的周期关系，则有：

- $\mathbb{Q}_{\geq 0}(P_1 \cap P_2) = \mathbb{Q}_{\geq 0}P_1 \cap \mathbb{Q}_{\geq 0}P_2$。
- $\mathbb{Q}_{\geq 0}(R_1 \circ R_2) = \mathbb{Q}_{\geq 0}R_1 \circ \mathbb{Q}_{\geq 0}R_2$。

证明 直接由定义验证即可，这里只证第一个。令 $\mathbf{p} \in \mathbb{Q}_{\geq 0}(P_1 \cap P_2)$，则存在 $n \in \mathbb{N}$ 满足 $n\mathbf{p} \in P_1 \cap P_2$，因而 $\mathbf{p} \in \mathbb{Q}_{\geq 0}P_1$，$\mathbf{p} \in \mathbb{Q}_{\geq 0}P_2$，从而 $\mathbb{Q}_{\geq 0}(P_1 \cap P_2) \subseteq \mathbb{Q}_{\geq 0}P_1 \cap \mathbb{Q}_{\geq 0}P_2$。另一方面，令 $\mathbf{p} \in \mathbb{Q}_{\geq 0}P_1 \cap \mathbb{Q}_{\geq 0}P_2$，则存在 $n_1, n_2 \in \mathbb{N}$ 满足：$n_1\mathbf{p}_1 \in P_1$，$n_2\mathbf{p}_2 \in P_2$，从而 $n_1n_2\mathbf{p} \in P_1 \cap P_2$，即 $\mathbf{p} \in \mathbb{Q}_{\geq 0}(P_1 \cap P_2)$，因而有 $\mathbb{Q}_{\geq 0}P_1 \cap \mathbb{Q}_{\geq 0}P_2 \subseteq \mathbb{Q}_{\geq 0}(P_1 \cap P_2)$，命题成立。 □

类似于锥集，如果存在有限的 $\mathbf{p}_1, \ldots, \mathbf{p}_k \in P$ 满足：$P = \sum_{i=1}^{k} \mathbb{N}\mathbf{p}_i$，则称 P 是有限生成的。一个集合 $X \subseteq \mathbb{Z}^d$ 如果满足 $X = \bigcup_{j=1}^{k}(\mathbf{b}_j + P_j)$，其中 $\mathbf{b}_j \in \mathbb{Z}^d$，$P_j$ 是一个有限生成的周期集，则 X 是一个半线性集。集合 $X \subseteq \mathbb{Z}^d$ 如果可以在 Presburger 算术中定义，则称 X 是一个 Presburger 集。在 [152] 有：

命题 5.1 一个集合 $X \subseteq \mathbb{Z}^d$ 是一个 Presburger 集当且仅当其是半线性的。

下面介绍一些半线性集的性质。

引理 5.5 给定一个半线性周期集 P，则 $\overline{\mathbb{Q}_{\geq 0}P}$ 是有限生成的。

证明 由定义 $P = \bigcup_{j=1}^{k} \mathbf{b}_j + P_j$，其中 $\mathbf{b}_j \in \mathbb{Z}^d, P_j$ 是一个周期集。令 $C = \sum_{j=1}^{k}(\mathbb{Q}_{\geq 0}\mathbf{b}_j + C_j)$，其中 $C_j = \mathbb{Q}_{\geq 0}P_j$，下面证明 $\overline{\mathbb{Q}_{\geq 0}P} = C$。显然 $\overline{\mathbb{Q}_{\geq 0}P} \subseteq C$，因此只需要证明 $C \subseteq \overline{\mathbb{Q}_{\geq 0}P}$。令 $\mathbf{p} \in P_j$，对于任意的 $n \in \mathbb{N}$ 有 $\mathbf{b}_j + n\mathbf{p} \in P$，从而 $\frac{1}{n}\mathbf{b}_j + \mathbf{p} \in \mathbb{Q}_{\geq 0}P$，即 $\mathbf{p} \in \overline{\mathbb{Q}_{\geq 0}P}$，因此有 $C_j = \mathbb{Q}_{\geq 0}P_j \subseteq \overline{\mathbb{Q}_{\geq 0}P}$，另一方面，显然 $\mathbb{Q}_{\geq 0}\mathbf{b}_j \in \overline{\mathbb{Q}_{\geq 0}P}$，从而 $C \subseteq \overline{\mathbb{Q}_{\geq 0}P}$，命题得证。 □

注意到对于任何一个周期集 $P \subseteq \mathbb{Z}^d$，$\mathbb{Q}_{\geq 0}P$ 是一个锥集。因此可以在周期集上定义类似可定义的概念，具体如下：

定义 5.3 一个周期集 $X \subseteq \mathbb{Z}^d$ 如果满足 $\mathbb{Q}_{\geq 0}P$ 是可定义的，则称 X 是渐近可定义 (asymptotically definable) 的。

回想引理5.3，一个可定义的锥集和任何一个向量空间的并的闭包都是有限生成的，因此定义如下的线性化操作，将一个渐近可定义的周期集与一个有限生成的锥集联系起来。给定一个渐近可定义周期集，定义：

$$\mathrm{lin}(P) = (P - P) \cap \overline{\mathbb{Q}_{\geq 0}P} \tag{5–8}$$

我们称 $\mathrm{lin}(P)$ 是 P 的线性化 (linearization)。下面说明 $\mathrm{lin}(P)$ 是有限生成的。

引理 5.6 令 $P \subseteq Z^d$ 是渐近可定义的，则 $\mathrm{lin}(P)$ 是有限生成的。

证明　由定义及引理5.3，$C = \overline{\mathbb{Q}_{\geq 0}P}$ 是有限生成的，因此可以令 $C = \sum_{i=1}^{k} \mathbb{Q}_{\geq 0}\mathbf{c}_i$，特别的不妨假设 $\mathbf{c}_i \in P - P$，这是因为总可以令其乘一个足够大的系数 $n \in \mathbb{N}$ 使其满足在 $P - P$ 里。定义如下集合 R:

$$R = \{\mathbf{c}_1, \ldots, \mathbf{c_k}\} \cup \{\mathbf{c} | \mathbf{c} = \sum_{i=1}^{k} \lambda_i \mathbf{c}_i, 0 \leq \lambda_i < 1 \wedge \lambda_i \in \mathbb{Q}, \mathbf{c} \in P - P\} \tag{5–9}$$

注意到 $|\lambda_i| < 1$，因此存在 $B \in \mathbb{N}$，对于任意的 $\mathbf{c} \in R$ 满足 $||\mathbf{c}||_\infty < B$，从而 R 是有限的。令 L 是由 R 生成的周期集，下面证明 $\mathrm{lin}(P) = L$。注意到 $R \subseteq \mathrm{lin}(P)$，因而有 $L \subseteq \mathrm{lin}(P)$。对于另一个方向，设 $\mathbf{p} \in \mathrm{lin}(P)$，则存在 $\mu_1, \ldots \mu_k \in \mathbb{Q}_{\geq 0}$ 使得 $\mathbf{p} = \sum_{i=1}^{k} \mu_i \mathbf{c}_i$。令 $[\mu_i]$ 表示不超过 μ_i 的最大整数，则有:

$$\mathbf{p} = \sum_{i=1}^{k} [\mu_i]\mathbf{c}_i + (\sum_{i=1}^{k} (\mu_i - [\mu_i])\mathbf{c}_i) \tag{5–10}$$

注意到 $\mathbf{r} = \sum_{i=1}^{k} (\mu_i - [\mu_i])\mathbf{c}_i \in R$，因此有 $\mathbf{p} \in L$，从而 $L = \mathrm{lin}(P)$，即 $\mathrm{lin}(P)$ 是有限生成的，命题得证。 □

引理 5.7　令 $P \subseteq \mathbb{Z}^d$ 是一个渐近可定义周期集，令 $\mathbf{b} \in \mathrm{lin}(P), \mathbf{c} \in int(\mathbb{Q}_{\geq 0}P)$，则存在 $k \in \mathbb{N}$ 满足 $\mathbf{b} + k\mathbf{c} \in P$。

证明　由 $\mathrm{lin}(P)$ 定义，存在 $\mathbf{p}_1, \mathbf{p}_2 \in P$ 满足 $\mathbf{p}_1 - \mathbf{p}_2 = \mathbf{b}$；由 $\mathbf{c}$ 是 $\mathbb{Q}_{\geq 0}P$ 的内点，因而存在 $k \in \mathbb{N}$ 满足 $\mathbf{c} + \frac{1}{k}\mathbf{p}_2 \in \mathbb{Q}_{\geq 0}P$ 并且存在 $k_2 \in \mathbb{N}$ 使得 $k_1\mathbf{c} \in P$，这里不妨令 k 是 k_1 的倍数并且令 $k = k_1 \cdot k_2$，则有:

$$\mathbf{b} + k\mathbf{c} = \mathbf{p}_1 - \mathbf{p}_2 + k_1 \cdot (k_2\mathbf{c}) + \mathbf{p}_2 = \mathbf{p}_1 + k_1 \cdot (k_2\mathbf{c}) \in P \tag{5–11}$$

□

可以看到，线性化操作可以将一个不太好表示的渐近可定义的周期集转换成一个好表达的有限生成的锥集。Leroux 还发现，线性化另一个特点是，尽管本来两个渐近可定义的周期集可以是不相交的，但是其线性化后的交集可能是非空的，但是其交集在某种‘维度’上会变的更小，这也是 Presburger 分离中最重要的一个特征，下面来叙述这个性质。首先来描述如何定义任何一个 $X \subseteq \mathbb{Z}^d$ 的‘维度’。

定义 5.4 (dimension)　给定一个集合 $X \subseteq \mathbb{Z}^d$，其维度 $\dim(X)$ 定义为最小的 $r \in [d]$，使得存在满足一组 $rank(V_i) \leq r$ 的向量空间 $V_1, \ldots V_k$ 和一组 $\mathbf{b}_i \in \mathbb{Z}^d$ 满足 $X \subseteq \bigcup_{i=1}^{k} \mathbf{b}_j + V_i$。

注　考虑到锥集和周期集具有相似的性质但不在同一个定义域，我们可以扩展维度的概念到一个锥集 $C \subseteq \mathbb{Q}^d$ 上，定义其维度为 $\dim(C) = \dim(C \cap \mathbb{Z}^d)$。

下面的引理说明一个周期集的维度大小就是其对应的向量空间的秩。

引理 5.8 给定一个周期集 $X \subseteq \mathbb{Z}^d$，令 V 是由 X 生成的向量空间，则有 $\dim(X) = rank(V)$。

证明 注意到 $X \subseteq V$，因此只需要证明 $rank(V) \leq \dim(X)$。令 $X \subseteq \sum_{i=1}^{k} \mathbf{b}_i + V_i$，下面证明一定存在 V_j 满足：$X \subseteq \mathbf{b}_j + V_j$。反设结论不成立，则对于某个 $j \in [k]$ 令 $\mathbf{p}_0 \in X \backslash \mathbf{b}_j + V_j$，则对于任意的 $\mathbf{p} \in X$，存在 $h \in [k], n_1, n_2 \in \mathbb{N}$ 满足：$n_1\mathbf{p}_0 + \mathbf{p}, n_2\mathbf{p}_0 + \mathbf{p} \in \mathbf{b}_h + V_h$，令 $n_i\mathbf{p}_0 + \mathbf{p} = \mathbf{b}_h + \mathbf{v}_i$，其中 $i = 1, 2$，从而有：

$$\mathbf{p}_0 = \frac{\mathbf{v}_2 - \mathbf{v_1}}{n_2 - n_1} \in V_h \tag{5–12}$$

$$\mathbf{p} = \mathbf{b}_h + (\mathbf{v_1} - n_1\mathbf{p}_0) \in \mathbf{b}_h + V_h \tag{5–13}$$

由式子5–13令 $\mathbf{p} = \mathbf{b}_h + \mathbf{v}$，其中 $\mathbf{v} \in V$，则有：

$$\mathbf{b}_h = \frac{1}{2}(\mathbf{v}_1 - n_1\mathbf{p}_0 - \mathbf{v}) \in V_h \tag{5–14}$$

从而有 $X \subseteq \mathbf{b}_h + V_h \subseteq V_h$ 与假设矛盾。因此一定存在一个 j 满足 $X \subseteq \mathbf{b}_j + V_j \subseteq V_j$，从而 $rank(V) \leq rank(V_j) \leq \dim(X)$，命题得证。 □

接下来的线性化定理，则是本节索要叙述的最重要的定理，也是后面 Presburger 分离对技术能成立的核心所在，其说明对于两个本来不相交的带偏移的渐近可定义的周期集，即使线性化后他们相交非空，其交集的维度会严格的减少，从而使得这样相交非空的次数是有限的。

定理 5.1 (Leroux[73]) 令 $\mathbf{b}_1, \mathbf{b}_2 \in \mathbb{Z}^d$，$P_1, P_2$ 是 $\mathbb{Z}^d$ 上的两个渐近可定义的周期集，如果其满足 $\mathbf{b}_1 + P_1 \cap \mathbf{b}_2 + P_2 = \emptyset$，则令 $X = \mathbf{b}_1 + \text{lin}(P_1) \cap \mathbf{b}_2 + \text{lin}(P_2)$ 有：

$$\dim(X) < \max\{\dim(\mathbf{b}_1 + P_1), \dim(\mathrm{b}_2 + P_2)\} \tag{5–15}$$

在正式证明该定理之前，下面先来用 Leroux 在 [73] 中提出的一个例子直观的解释一下，考虑如下两个集合：

- $C_1 = \mathbf{b}_1 + P_1$，其中 $\mathbf{b}_1 = (0, 0)$，$P_1 = \{(p_1, p_2) \in \mathbb{N}^2 | p_2 \leq p_1 \leq 2^{p_2} - 1\}$。
- $C_2 = \mathbf{b}_2 + P_2$，其中 $\mathbf{b}_2 = (8, 3)$，$P_2 = \mathbb{N}(1, 0) + \mathbb{N}(3, -1)$

如图5–2，左图两条蓝色线中间部分的区域 1 中所有整数点即是 C_1，而两条红色线中间的区域 2 中的所有整数点即是 C_2，显然有 $C_1 \cap C_2 = \emptyset$。另一方面，记 $C_i' = \mathbf{b}_i + \text{lin}(P_i)$，则有 C_1' 是右图中区域 1′ 两条蓝色线中间所有整点的集合，而 C_2'

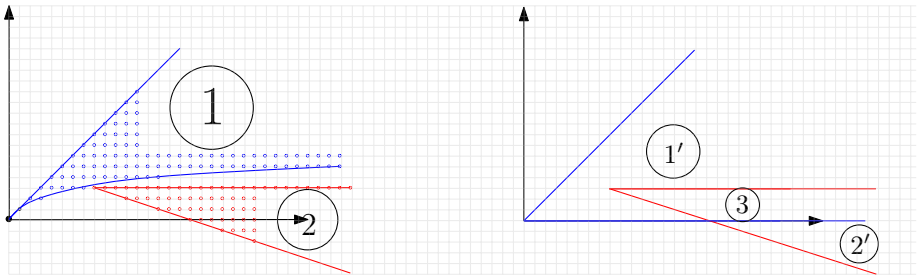


图 5–2　一个定理5.1的例子

Figure 5–2　An example of theroem5.1

则是区域 $2'$ 两条红色线中间所有整点的集合，很显然其的交集不空 $C_1' \cap C_2' \neq \emptyset$，也就是右图中的区域 3，令该集合为 X 有：

$$X \subseteq (8,3) + \mathbb{N}(1,0) \bigcup (11,2) + \mathbb{N}(1,0) \bigcup (14,1) + \mathbb{N}(1,0) \bigcup (17,0) + \mathbb{N}(1,0)$$

即 $\dim(X) = 1$，而显然 $\dim C_1 = \dim(C_2) = 2$。这也就是定理5.1想要说明的，如果 C_1, C_2 本身是不相交的，那么即使其线性化后的 C_1', C_2' 后的交集非空，其交集 X 的维度也会严格的减少，这为后面的 Presburger 分离提供了强有力的保障。

现在回到定理5.1的证明。其核心思想在于，线性化后产生的交集是由原来集合的边界所引起的，从而导致最后 X 的维度会变小。在 Leroux 的证明中，其用到了如下引理：

引理 5.9　令 $C_{\geq}, C_{\leq} \subseteq \mathbb{Q}^d$ 是两个生成同一向量空间 V 的有限生成的锥集并且满足 $C_{\geq} \cap C_{\leq}$ 生成的向量空间 $V_{\cap} \subsetneq V$，则存在一个向量 $\mathbf{h} \in V \setminus \{\mathbf{0}\}$ 满足，对于 $\# \in \{\geq, \leq\}$ 有：

$$C_{\#} \subseteq \{\mathbf{v} \in V \mid \sum_{i=1}^{d} \mathbf{v}[i] \cdot \mathbf{h}[i] \ \# \ 0\} \tag{5–16}$$

该引理的证明需要使用 Farkas 引理，并且之后的证明也较为复杂。下面提出另一种证明方式，该证明由郑扬珞提出。一个关键的发现点在于，如果两个锥集的内点组成的集合的交集是空的，其交集的维度便会严格减少，而另一方面，定理5.1的条件其实也蕴含了 $int(\text{lin}(P)_1) \cap int(\text{lin}(P)_1) = \emptyset$。事实上，这一发现非常直观；首先锥集的内点如果不相交的话，其交集都会是在边界，而边界的维度显然是要比整个维度低的，其次在定理5.1中的条件里，如果 $\text{lin}(P_1), \text{lin}(P)_2$ 的内点有交集，则很容易通过乘以适当的系数放大导致 $\mathbf{b}_1 + P_1$ 与 $\mathbf{b}_2 + P_2$ 的交集非空。下面严格的来叙述这个证明。

引理 5.10 令 $C_1, C_2 \subseteq \mathbb{Q}^d$ 是两个生成同一向量空间 V 的有限生成的锥集，并且满足：$int(C_1) \cap int(C_2) = \emptyset$，则有：

$$\dim(C_1 \cap C_2) < rank(V) \tag{5–17}$$

证明 由引理5.1，对于 C_i，存在一组向量 $H_i \subseteq V \setminus \{\mathbf{0}_d\}$ 满足：

$$C_i = \bigcap_{\mathbf{h} \in H_i} \{\mathbf{v} | V | \sum_{j=1}^{d} \mathbf{h}[j] \cdot \mathbf{v}[j] \geq 0\} \tag{5–18}$$

$$int(C_i) = \bigcap_{\mathbf{h} \in H_i} \{\mathbf{v} | V | \sum_{j=1}^{d} \mathbf{h}[j] \cdot \mathbf{v}[j] > 0\} \tag{5–19}$$

因此，令 $H = H_1 \cup H_2$ 有：

$$C_1 \cap C_2 = \bigcap_{\mathbf{h} \in H} \{\mathbf{v} | V | \sum_{j=1}^{d} \mathbf{h}[j] \cdot \mathbf{v}[j] \geq 0\} \tag{5–20}$$

$$int(C_1) \cap int(C_2) = \bigcap_{\mathbf{h} \in H} \{\mathbf{v} | V | \sum_{j=1}^{d} \mathbf{h}[j] \cdot \mathbf{v}[j] > 0\} \tag{5–21}$$

对于 $\mathbf{h} \in H$，定义 $W_{\mathbf{h}} = \{\mathbf{v} | V | \sum_{j=1}^{d} \mathbf{h}[j] \cdot \mathbf{v}[j] = 0\}$，注意到 $\mathbf{h} \notin W_{\mathbf{h}}$，所以有 $rank(W_{\mathbf{h}}) < rank(V)$。由条件 $int(C_1) \cap int(C_2) = \emptyset$，因而有:

$$C_1 \cap C_2 \subseteq \bigcup_{\mathbf{h} \in H} C_1 \cap C_2 \cap W_{\mathbf{h}} \subseteq \bigcup_{\mathbf{h} \in H} W_{\mathbf{h}} \tag{5–22}$$

从而 $\dim(C_1 \cap C_2) \leq \max_{\mathbf{h} \in H} rank(W_{\mathbf{h}}) < rank(V)$，引理得证。 □

现在来开始证明定理5.1。

证明 (定理5.1的证明) 如果 X 是空集则定理成立。下面假设 X 非空，由引理5.6可知 X 是一个半线性集，因此令 $X = \bigcup_{i=1}^{k} \mathbf{c}_i + X_i$，其中 X_i 是有限生成的周期集。

首先证明对任意的 $i \in [k]$ 有 $X_i \subseteq \overline{\mathbb{Q}_{\geq 0} P_1} \cap \overline{\mathbb{Q}_{\geq 0} P_2}$。由 $\mathbf{c}_i + X_i \subseteq X$ 可知对于 $j \in \{1, 2\}$ 和任意的 $n \in \mathbb{N}$ 有：

$$\mathbf{c}_i - \mathbf{b}_j + nX_i \subseteq \text{lin}(P_j) \subseteq \overline{\mathbb{Q}_{\geq 0} P_j} \tag{5–23}$$

从而 $X_i + \frac{1}{n}(\mathbf{c}_i - \mathbf{b}_j) \subseteq \overline{\mathbb{Q}_{\geq 0} P_j}$，即 $X_i \subseteq \overline{\overline{\mathbb{Q}_{\geq 0} P_j}} = \overline{\mathbb{Q}_{\geq 0} P_j}$ 对任意的 $j = 1, 2$ 都成立。

令 V_1, V_2 是 P_1, P_2 对应生成的向量空间，显然有 $X_i \subseteq V_1 \cap V_2$，如果 $V_1 \neq V_2$，则由引理5.8：

$$\begin{aligned} \dim(X) \leq \max_{i\in[k]} \dim(X_i) &\leq rank(V_1 \cap V_2) \\ &< \max\{rank(V_1), rank(V_2)\} \\ &= \max\{\dim(\mathbf{b}_1 + P_1), \dim(\mathbf{b}_2 + P_2)\} \end{aligned}$$

下面令 $V_1 = V_2 = V$，接下来说明此时 $\mathbb{Q}_{\geq 0}P_1, \mathbb{Q}_{\geq 0}P_2$ 的内点的交集为空，即 $int(\mathbb{Q}_{\geq 0}P_1) \cap int(\mathbb{Q}_{\geq 0}P_2) = \emptyset$。事实上，反设结论不成立，即存在 $\mathbf{c} \in int(\mathbb{Q}_{\geq 0}P_1) \cap int(\mathbb{Q}_{\geq 0}P_2)$，令 $\mathbf{b} \in X$，即 $\mathbf{b} \in \mathbf{b}_j + \lin(P_j)$，则有 $\mathbf{b} - \mathbf{b}_j \in \lin(P_j)$, $\mathbf{c} \in int(\mathbb{Q}_{\geq 0}P_j)$，由引理5.7，存在 $k_j \in \mathbb{N}$ 满足:

$$\mathbf{b} - \mathbf{b}_j + k_j\mathbf{c} \in P_j \tag{5–24}$$

从而有 $\mathbf{b} + (k_1 + k_2)\mathbf{c} \in \mathbf{b}_1 + P_1 \cap \mathbf{b}_2 + P_2$，与条件矛盾。由于 P_j 和 $\mathbb{Q}_{\geq 0}P_j$ 会生成相同的向量空间 V_j，因此由引理5.10，$dim(\mathbb{Q}_{\geq 0}P_1 \cap \mathbb{Q}_{\geq 0}P_2) < rank(V)$，从而：

$$\begin{aligned} \dim(X) \leq \max_{i\in[k]} \dim(X_i) &\leq dim(\mathbb{Q}_{\geq 0}P_1 \cap \mathbb{Q}_{\geq 0}P_2) && (5\text{–}25) \\ &< rank(V) = \{\dim(\mathbf{b}_1 + P_1), \dim(\mathbf{b}_2 + P_2)\} && (5\text{–}26) \end{aligned}$$

□

5.1.3 Presburger 分离对

这一节将介绍使用 Presburger 分离对的技术，即如果一个关系是几乎半线性的，则对于不在其克莱尼星闭包关系的一对 (X, Y)，能找到一对 Presburger 分离对将其严格的分离开来，从而证明其不在该关系的克莱尼星闭包。下面先来定义几乎半线性 (almost semi-linear) 的概念。

定义 5.5　给定一个集合 $X \subseteq \mathbb{Z}^d$，如果 X 满足对于任意的 Presburger 集 $S \subseteq \mathbb{Z}^d$ 都有 $X \cap S = \bigcup_{i=1}^{k} \mathbf{b}_i + P_i$，其中 $\mathbf{b}_i \in \mathbb{Z}^d$，$P_i \subseteq \mathbb{Z}^d$ 是一个渐近可定义的周期集，则称 X 是几乎半线性的。

半线性集是几乎半线性集的一种特例。而在下一节中我们将证明向量加法系统的可达关系是几乎半线性的，结合在[40] 中证明的其可达关系不是半线性的，自然能得到几乎半线性集是严格包含半线性集的。首先阐述一个关于几乎半线性的性质，其表述为如果一个关系是几乎半线性的，则它能将一个 Presburger 集映射成一个几乎半线性的集合。为了叙述方便，本节采用如下一些记号。令 R 是 $\mathbb{Z}^d$ 上的一个关系，$X \subseteq \mathbb{Z}^d$ 是 $\mathbb{Z}^d$ 上的一个集合，定义 X 在 T 上的前像 (forward image)$Pre_R(X)$ 和后像 (backward image)$Post_R(X)$ 分别为：

- $Pre_R(X) = \{y|(y,x) \in R,\ x \in X\}$.
- $Post_R(X) = \{y|(x,y) \in R,\ x \in X\}$.

特别的，如果前像满足 $Pre_R(X) \subseteq X$，则称 X 是关于 R 的前不变量 (forward invariant)；如果后像满足 $Post_R(X) \subseteq X$，则称 X 是关于 R 的后不变量 (backward invariant)。

引理 5.11 令 R 是 $\mathbb{Z}^d$ 上的一个几乎半线性的关系，$X \subseteq \mathbb{Z}^d$ 是一个 Presburger 集，则有 $Pre_R(X), Post_R(X)$ 都是几乎半线性的。

证明 注意到令 $R^{-1} = \{(y,x)|(x,y)\}$，显然 R^{-1} 也是几乎半线性的，并且有 $Pre_R(X) = Post_{R^{-1}}(X)$，因此只需要证 $Post_R(X)$ 是几乎半线性的。

令 $S \subseteq \mathbb{Z}^d$ 是一个 Presburger 集，则 $X \times S$ 是一个 Presburger 关系，则由定义:

$$R \cap (X \times S) = \bigcup_{i=1}^{k} (\mathbf{a}_i, \mathbf{b}_i) + R_i \tag{5–27}$$

其中 R_i 是一个渐进可定义的周期关系。令 $P_i = \{y|(x,y) \in R_i\}$，则有 $Post_R(X) \cap S = \bigcup_{i=1}^{k} \mathbf{b}_i + P_i$。下面说明 P_i 是个渐近可定义的周期集，首先由于 R_i 是周期的，因此 P_i 也是一个周期集。其次 R_i 是一个渐近可定义的关系，从而 $\mathbb{Q}_{\geq 0} R_i$ 是可定义的。令 $C_i = \{y|(x,y) \in \mathbb{Q}_{\geq 0} R_i\}$，下面证明 $C_i = \mathbb{Q}_{\geq 0} P_i$。$\mathbb{Q}_{\geq 0} P_i \subseteq C_i$ 是显然的，只用证明 $C_i \subseteq \mathbb{Q}_{\geq 0} P_i$。令 $\mathbf{c} \neq \mathbf{0}_d \in C_i$，则存在 $\mathbf{b} \in \mathbb{Q}^d$ 满足 $(\mathbf{b}, \mathbf{c}) \in \mathbb{Q}_{\geq 0} R_i$，则由定义存在 $q \in \mathbb{Q}_{>0}$ 满足 $(q\mathbf{b}, q\mathbf{c}) \in R_i$，即有 $q\mathbf{c} \in P_i, \mathbf{c} = \frac{1}{q} \cdot q\mathbf{c} \in \mathbb{Q}_{\geq 0} P_i$，从而 $\mathbb{Q}_{\geq 0} P_i$ 是可定义的，即 $Post_R(X)$ 是几乎半线性的。 □

接下来介绍分离对的概念。给定两个 Presburger 集 $X, Y \subseteq \mathbb{Z}^d$ 和一个 $\mathbb{Z}^d$ 上的自反传递的几乎半线性关系 R^*，如果其满足 $R^* \cap X \times Y = \emptyset$，则称 (X,Y) 是一个分离对。令 $D = \mathbb{Z}^d \setminus (X \cup Y)$，称 D 是分离对 (X,Y) 的遗留域 (domain)，如果 $D = \emptyset$，则分离对 (X,Y) 就是 $\mathbb{Z}^d$ 的一个划分 (partition)，此时不难得到 X, Y 便是关于 R 的一个不变量。下述引理说明对于任一个分离对 (X,Y)，总能将其变为一个划分。

引理 5.12 (Leroux[73]) 给定一个 $\mathbb{Z}^d$ 上的自反传递的几乎半线性关系 R^* 和其上的一个分离对 (X,Y)，如果其遗留域 $D \neq \emptyset$，则存在一个分离对 (X', Y') 满足 $X \subseteq X'$, $Y \subseteq Y'$，并且其遗留域 D' 满足 $\dim(D') < \dim(D)$。

证明 D 是一个 Presburger 集，因此由引理5.11，$Post_{R^*}(X) \cap D = \bigcup_{i=1}^{k_1} \mathbf{b}_i + P_i$，其中 $\mathbf{b}_i \in \mathbb{Z}^d$，$P_i$ 是一个渐近可定义的周期集。定义如下的 Presburger 集:

$$S = \bigcup_{i=1}^{k_1} \mathbf{b}_i + \mathrm{lin}(P_i) \tag{5–28}$$

令 $Y' = Y \cup (D \setminus S)$, 注意到 $Post_{R^*}(X) \cap D \subseteq S$, 因此有 $Post_{R^*}(X) \cap Y' = \emptyset$, 即 (X, Y') 也是一个分离对。通过相同的方法可以构造出相应的 X'，其中令 $Pre_{R^*}(Y) \cap D = \bigcup_{i=1}^{k_2} \mathbf{c}_i + Q_i$，下面证明 (X', Y') 则是满足要求的分离对，只需要考虑其遗留域 D'。定义集合：

$$Z_{j,i} \stackrel{\text{def}}{=} \mathbf{b}_j + \text{lin}(P_j) \ \cap \ \mathbf{c}_i + \text{lin}(Q_i) \tag{5–29}$$

□

则有 $D' = D \cap (\bigcup_{\substack{1 \le i \le k_1 \\ 1 \le j \le k_2}} Z_{j,i})$。注意到 $Post_{R^*}(X) \cap Pre_{R^*} = \emptyset$, 因此对于 $i \in [k_1]$, $j \in [k_2]$ 有 $(\mathbf{b}_j + P_j) \cap (\mathbf{c}_i + Q_i) = \emptyset$，从而由定理5.1可以得到 $\dim(Z_{j,i}) < \max\{\dim(\mathbf{b}_j + P_j), \dim(\mathbf{c}_i + Q_i)\}$，从而有：

$$\dim(D') \le \max \dim(Z_{j,i}) < \max\{\dim(\mathbf{b}_j + P_j), \dim(\mathbf{c}_i + Q_i)\} \le \dim(D)$$

由引理5.12可以直接得到 Presburger 的分离对定理。

定理 5.2 (leroux[73])　令 R^* 是一个 $\mathbb{Z}^d$ 上的自反传递的几乎半线性关系，$X, Y \subseteq \mathbb{Z}^d$ 是 Presburger 集, 如果其满足 $R^* \cap (X \times Y) = \emptyset$, 则存在一个 $\mathbb{Z}^d$ 上的划分 (X', Y') 满足 X' 是一个后不变量 Y' 是一个前部变量，并且有 $X \subseteq X', Y \subseteq Y'$。

定理5.2提供了一个证明 $(X \times Y) \cap R^* = \emptyset$ 的方法, 即可以枚举所有的 Presburger 不变量, 如果其交集真的为空, 那么一定存在一个 Presburger 不变量 S 满足 $(S, \mathbb{Z}^d \setminus S)$ 可以把 X, Y 给分离开来。以向量加法系统的可达性问题为例，假设其可达关系是自反传递几乎半线性的，如果 $\mathbf{v}$ 对于 $\mathbf{u}$ 是不可达的，令 $X = \{\mathbf{u}\}$, $Y = \{\mathbf{v}\}$，那么由定理5.2，一定存在一个 Presburger 不变量 $(S, \mathbb{Z}^d \setminus S)$ 将 X, Y 分别包裹起来。另一方面，如果一个格局是可达的，必然可以通过枚举所有的路径来找到这样一条可达路径，从而获得了可达性问题的可判定性结论。

定理 5.3 (leroux[73])　向量加法系统的可达性问题是可判定的。

因此为了证明其可判定性，只需要证明向量加法系统的可达性关系是几乎半线性的，这将在下一节证明该结论。

5.2　可达集的几何性质

本节将证明向量加法系统的可达性关系是几乎半线性的。为了证明该结论需要证明，对于任何一个 Presburger 集 $X = (\mathbf{m}, \mathbf{n}) + P$, $\stackrel{*}{\rightarrow} \cap X$ 都可以被分解成若干个带偏移的渐近可定义的周期关系，注意到后者其实相当于在某个偏移基础上的可达关系，因此本节首先将证明这样一种相对可达关系是渐近可定义的周期关系，从而最终证明可达性关系是几乎半线性的。在本节固定一个 d 维向量加法系统 $\mathbf{V} = \{\mathbf{A}\}$。

5.2.1 可达集相对可达关系

首先定义相对可达关系 (production relation)。给定一个格局 $\mathbf{m} \in \mathbb{N}^d$，定义如下关系 $\rightarrow_{\mathbf{m}}$:

$$\rightarrow_{\mathbf{m}} \stackrel{\text{def}}{=} \{(\mathbf{x}, \mathbf{y}) \mid \mathbf{m} + \mathbf{x} \stackrel{*}{\rightarrow} \mathbf{m} + \mathbf{y},\ \mathbf{x}, \mathbf{y} \in \mathbb{N}^d\}. \tag{5–30}$$

对于一个格局串 $\rho = \mathbf{m}_1 \dots \mathbf{m}_k$，可以通过复合的方式来定义关于 ρ 的相对可达关系，即 $\rightarrow_\rho \stackrel{\text{def}}{=} \rightarrow_{\mathbf{m}_1} \circ \cdots \circ \rightarrow_{\mathbf{m}_k}$。下面的引理说明相对可达关系是一个周期关系，并且加上偏移之后其便是可达关系的一个子集。

引理 5.13 令 $\mathbf{m} \in \mathbb{N}^d$, $\rho \in (\mathbb{N}^d)^*$ 分别是一个格局和格局串，则有:

- $\rightarrow_{\mathbf{m}}$ 是一个周期关系。
- $(src(\rho), tgt(\rho)) + \rightarrow_\rho \subseteq \stackrel{*}{\rightarrow}$。

证明 先证第一点。令 $(\mathbf{u}_1, \mathbf{v}_1), (\mathbf{u}_2, \mathbf{v}_2) \in \rightarrow_{\mathbf{m}}$，由于 $\mathbf{u}_1, \mathbf{v}_1 \in \mathbf{N}^d$，因此: $\mathbf{u}_1 + \mathbf{u}_2 \rightarrow_{\mathbf{m}} \mathbf{v}_1 + \mathbf{u}_2,\ \mathbf{u}_2 + \mathbf{v}_1 \rightarrow_{\mathbf{m}} \mathbf{v}_2 + \mathbf{v}_1$，即:

$$\mathbf{u}_1 + \mathbf{u}_2 \rightarrow_{\mathbf{m}} \mathbf{v}_1 + \mathbf{u}_2 \rightarrow_{\mathbf{m}} \mathbf{v}_1 + \mathbf{v}_2 \tag{5–31}$$

从而 $(\mathbf{u}_1 + \mathbf{u}_2, \mathbf{v}_1 + \mathbf{v}_2) \in \rightarrow_{\mathbf{m}}$。下面证第二点。令 $\rho = \mathbf{m}_1 \dots \mathbf{m}_n$，则 $src(\rho) = \mathbf{m}_1$, $tgt(\rho) = \mathbf{m}_n$ 并且令 $\mathbf{m}_{i+1} = \mathbf{m}_i + \mathbf{a}_i$，其中 $\mathbf{a}_i \in \mathbb{Z}^d$。假设 $(\mathbf{u}, \mathbf{v}) \in \stackrel{\rho}{\rightarrow}$，则存在 $\mathbf{x}_0, \dots \mathbf{x}_n$ 满足:

- $\mathbf{x}_0 = \mathbf{u},\ \mathbf{x}_n = \mathbf{v}$。
- 对于 $i \in [n]$ 有 $\mathbf{x}_{i-1} \rightarrow_{\mathbf{m}_i} \mathbf{x}_i$，即 $\mathbf{x}_{i-1} + \mathbf{m}_i \stackrel{*}{\rightarrow} \mathbf{x}_i + \mathbf{m}_i$。

从而有:

$$\mathbf{m}_1 + \mathbf{u} = \mathbf{m}_1 + \mathbf{x}_0 \stackrel{*}{\rightarrow} \mathbf{m}_1 + \mathbf{x}_1 \stackrel{\mathbf{a}_1}{\rightarrow} \mathbf{m}_2 + \mathbf{x}_1 \stackrel{*}{\rightarrow} \cdots \stackrel{*}{\rightarrow} \mathbf{m}_n + \mathbf{x}_n = \mathbf{m}_n + \mathbf{v}$$

即 $(src(\rho) + \mathbf{x}_0, tgt(\rho) + \mathbf{x}_n) \in \stackrel{*}{\rightarrow}$。 □

下面证明相对可达关系是渐近可定义的周期关系。由引理5.4，只需要证明 $\rightarrow_{\mathbf{m}}$ 是渐近可定义的，即 $\mathbb{Q}_{\geq 0} \rightarrow_{\mathbf{m}}$ 是可定义的。由引理5.3，只要证明对于任一向量空间 $V \subseteq \mathbb{N}^d$，$\overline{\mathbb{Q}_{\geq 0} \rightarrow_{\mathbf{m}} \cap V}$ 是有限生成的，为了叙述方便，下面用 $\rightarrow_{\mathbf{m},V}$ 表示该关系。

我们只需要研究那些起点和终点都在 $\{(\mathbf{m}, \mathbf{m})\} + \mathbb{N}^d \times \mathbb{N}^d \cap V$ 的运行，将所有这样的运行的集合记作 $\Omega_{\mathbf{m},V}$。记 $Q_{\mathbf{m},V}$ 表示在 $\Omega_{\mathbf{m},V}$ 中所有出现过的格局，$I_{\mathbf{m},V}$ 是 $Q_{\mathbf{m},V}$ 中上界为无穷的那些维度的集合，即 $I_{\mathbf{m},V} = \{i \mid \sup_{\mathbf{q} \in Q_{\mathbf{m},V}} \mathbf{q}[i] = \infty\}$。记

$J_{\mathbf{m},V} = [d] \setminus I_{\mathbf{m},V}$，则 $Q_{\mathbf{m},V}$ 中的格局在 $J_{\mathbf{m},V}$ 上的投影只有有限多个，并且可以构成一张有限图 $G_{\mathbf{m},V}$。一个思路是希望说明 $\rightarrow_{\mathbf{m}\cap V}$ 中的运行，都可以用 $G_{\mathbf{m},V}$ 中的某条路径近似的表示。为此，先说明对于 $I_{\mathbf{m},V}$ 中的维度，任何一个 $\Omega_{\mathbf{m},V}$ 中的运行，都可以将这些维度上的值抬上任意的高度。下面来介绍自生泵 (intraproduction) 的概念。

定义 5.6　给定 $\mathbf{m} \in \mathbb{N}^d$ 和向量空间 V，一个关于 $(\mathbf{m}, V)$ 的自生泵是一个三元组：$(\mathbf{r}, \mathbf{x}, \mathbf{s})$ 满足：

- $\mathbf{x} \in \mathbf{N}^d$，$(\mathbf{r}, \mathbf{s}) \in (\mathbf{N}^d \times \mathbf{N}^d) \cap V$。
- $\mathbf{r} \rightarrow_{\mathbf{m}} \mathbf{x} \rightarrow_{\mathbf{m}} \mathbf{s}$。

特别的，如果对于所有的 $i \in I_{\mathbf{m},V}$ 都有 $\mathbf{x}[i] > 0$ 则称其是全自生泵 (total intraproduction)。

由于 $\rightarrow_{\mathbf{m},V}$ 是周期的，因此如果 $(\mathbf{r}, \mathbf{x}, \mathbf{s})$ 是一个自生泵，则对于任意的 $i \in I_{\mathbf{m},V}$ 有 $(n\mathbf{r}, n\mathbf{x}, n\mathbf{s})$ 也是一个自生泵，这说明 $\mathbf{m}+n\mathbf{x}$ 将会出现在 $Q_{\mathbf{m},V}$ 中，即如果 $i \in J_{\mathbf{m},V}$ 则有 $\mathbf{x}[i] = 0$。此外，下面的引理还说明，对于任意的 $\mathbf{m}, V$，其存在一个全自生泵。

引理 5.14 (Leroux[73])　对于任何一个 $\mathbf{m}, V$，都存在一个全自生泵。

证明　自生泵是具有可加性的，因此只需要证明对于每个 $i \in I_{\mathbf{m},V}$ 都存在一个自生泵 $(\mathbf{r}, \mathbf{x}, \mathbf{s})$ 满足 $\mathbf{x}[i] > 0$ 即可。由于 $sup_{\mathbf{q}\in Q_{\mathbf{m},V}} \mathbf{q}[i] = \infty$，因此可以找到一个 $\mathbf{q}[i]$ 严格增长的无限序列，结合 $(\mathbb{N}^d, \leq)$ 是一个良序，其中存在 $\mathbf{q}, \mathbf{q}'$ 满足：

- $\mathbf{q} \leq \mathbf{q}'$，$\mathbf{q}[i] < \mathbf{q}'[i]$。
- 存在 $(\mathbf{r}, \mathbf{s}) \in \mathbb{N}^d \times \mathbb{N}^d \cap V$ 满足：$\mathbf{m} + \mathbf{r} \xrightarrow{*} \mathbf{q} \xrightarrow{*} \mathbf{m} + \mathbf{s}$。
- 存在 $\mathbf{r}', \mathbf{s}' \in \mathbb{N}^d \times \mathbb{N}^d \cap V$ 满足：$\mathbf{m} + \mathbf{r}' \xrightarrow{*} \mathbf{q}' \xrightarrow{*} \mathbf{m} + \mathbf{s}'$。

从而有：

$$\mathbf{m} + \mathbf{r} + \mathbf{r}' \xrightarrow{*} \mathbf{q}' + \mathbf{r} = \mathbf{q}' - \mathbf{q} + \mathbf{q} + \mathbf{r} \xrightarrow{*} \mathbf{q}' - \mathbf{q} + \mathbf{m} + \mathbf{s} + \mathbf{r} \xrightarrow{*} \mathbf{q}' + \mathbf{s} \xrightarrow{*} \mathbf{m} + \mathbf{s} + \mathbf{s}'$$

令 $\mathbf{x} = \mathbf{s}+(\mathbf{q}'-\mathbf{q})+\mathbf{r}$ 上述运行说明 $(\mathbf{r}+\mathbf{r}', \mathbf{x}, \mathbf{s}+\mathbf{s}')$ 也是一个自生泵，结合 $\mathbf{q}[i] < \mathbf{q}'[i]$ 有 $\mathbf{x}[i] > 0$，引理得证。 □

现在回来证明 $\overline{\mathbb{Q}_{\geq 0} \rightarrow_{\mathbf{m},V}}$ 是有限生成的，即如下定理：

定理 5.4 (Leroux[73])　向量加法系统的可达关系是几乎半线性的。

在证明之前，先来严格定义一下上面所说到的图 $G_{\mathbf{m},V}$，为了方便叙述，令 $I = I_{\mathbf{m},V}$，$J = J_{\mathbf{m},V}$，则定义 $G_{\mathbf{m},V} = (Q, E)$，其中 $Q = \{\mathbf{q}_J | \mathbf{q} \in Q_{\mathbf{m},V}\}$，$E = \{(\mathbf{p}_J, \mathbf{a}, \mathbf{q}_J) | \mathbf{q} = \mathbf{p} + \mathbf{a},\ \mathbf{a} \in \mathbf{A},\ \mathbf{p}, \mathbf{q} \in Q_{\mathbf{m},V}\}$。在 $G_{\mathbf{m},V}$ 上定义关系 $R_{\mathbf{m},V} \subseteq (\mathbb{N}^d \times \mathbb{N}^d) \cap V$，如果 $(\mathbf{x}, \mathbf{y})$ 满足：

- $(\mathbf{x},\mathbf{y}) \in (\mathbb{N}^d \times \mathbb{N}^d) \cap V$，并且对于 $i \in J$ 由 $\mathbf{x}[i] = \mathbf{y}[i] = 0$。
- $G_{\mathbf{m},V}$ 中的点 $\mathbf{m}_J$ 上存在一个圈，其上面的标记为 $\mathbf{a}_1 \ldots \mathbf{a}_n$ 满足 $\mathbf{y} = \mathbf{x} + \sum_{i=1}^{n} \mathbf{a}_i$。

则有 $(\mathbf{x},\mathbf{y}) \in R_{\mathbf{m},V}$。下面的引理说明 $\rightarrow_{\mathbf{m},V}$ 和上述关系 $R_{\mathbf{m},V}$ 的关系。

引理 5.15 $\overline{\mathbb{Q}_{\geq 0} R_{\mathbf{m},V}} = \overline{\mathbb{Q}_{\geq 0} \rightarrow_{\mathbf{m},V}}$。

证明 先证 $\supseteq$。令 $(\mathbf{u},\mathbf{v}) \in \rightarrow_{\mathbf{m},V}$，则有 $\mathbf{m} + n\mathbf{u} \xrightarrow{*} \mathbf{m} + n\mathbf{v}$ 对任意的 $n \in \mathbb{N}$。由此若 $\mathbf{u}[i] > 0$ 或者 $\mathbf{v}[i] > 0$，则 $i \in I$。记 $\mathbf{m}+\mathbf{u}$ 到 $\mathbf{m}+\mathbf{v}$ 的路径为 π，则 π 也是 $G_{\mathbf{m},V}$ 里 $\mathbf{m}_J$ 上的一个圈，从而有 $(\mathbf{u},\mathbf{v}) \in R_{\mathbf{m},V}$，因此有 $\overline{\mathbb{Q}_{\geq 0} \rightarrow_{\mathbf{m},V}} \subseteq \overline{\mathbb{Q}_{\geq 0} R_{\mathbf{m},V}}$。

再证 $\subseteq$。令 $(\mathbf{u},\mathbf{v}) \in R_{\mathbf{m},V}$，则对于任意的 $i \in I$ 有 $\mathbf{u}[i] = \mathbf{v}[i] = 0$；并且存在 $\pi = \mathbf{a}_1 \ldots \mathbf{a}_n$ 满足：$\mathbf{m} + \mathbf{u} + \sum_{i=1}^{n} \mathbf{a}_i = \mathbf{m} + \mathbf{v}$。如果对于任意 $j \in [n]$ 有 $\mathbf{m} + \mathbf{u} + \sum_{i=1}^{j} \mathbf{a}_i \geq \mathbf{0}_d$，则命题成立。否则可以通过全自生泵将其抬上去成为一个合法的运行，令 $(\mathbf{r},\mathbf{x},\mathbf{s})$ 是 $(\mathbf{m},V)$ 上的一个全自生泵，则存在 $p \in \mathbb{N}$ 满足对任意 $j \in [n]$ 有：$\mathbf{m} + p\mathbf{x} + \mathbf{u} + \sum_{i=1}^{j} \mathbf{a}_i \in \mathbb{N}^d$。因此对于任意的 $k \in \mathbb{N}$ 可以构造下列运行：

$$\mathbf{m} + p\mathbf{r} + k\mathbf{u} \xrightarrow{*} \mathbf{m} + p\mathbf{x} + k\mathbf{u} \xrightarrow{\pi^k} \mathbf{m} + p\mathbf{x} + k\mathbf{v} \xrightarrow{*} \mathbf{m} + p\mathbf{s} + k\mathbf{v}$$

因此有 $p(\mathbf{r},\mathbf{s}) + k(\mathbf{u},\mathbf{v}) \in \rightarrow_{\mathbf{m},V}$，令 $k = K \cdot p$ 从而 $\frac{1}{K}(\mathbf{r},\mathbf{s}) + (\mathbf{u},\mathbf{v}) \in \mathbb{Q}_{\mathbf{m},V} \rightarrow_{\mathbf{m},V}$，即 $(\mathbf{u},\mathbf{v}) \in \overline{\mathbb{Q}_{\mathbf{m},V} \rightarrow_{\mathbf{m},V}}$，从而 $\overline{\mathbb{Q}_{\geq 0} R_{\mathbf{m},V}} \subseteq \overline{\mathbb{Q}_{\geq 0} \rightarrow_{\mathbf{m},V}}$，引理得证。 □

由于正则语言的 Parikh 像是 Presburger 集，所以 $R_{\mathbf{m},V}$ 是一个 Presburger 集，由引理5.5有 $\overline{\mathbb{Q}_{\geq 0} R_{\mathbf{m},V}}$ 是有限生成的，结合引理5.15和本节开始的讨论，便得到了相对可达关系的性质：

定理 5.5 向量加法系统相对可达关系 $\rightarrow_\rho$ 是一个渐近可定义的周期关系。

5.2.2 可达集是几乎半线性集合

本节将证明可达关系是几乎半线性的，在上节已经证明了相对可达关系是渐近可定义的，不难想象，为了证明可达关系的几乎半线性性，现在需要挑选一些合适的偏移，即分解中的 $\mathbf{b}_i$，为此首先来定义运行上的关系 $\preceq$[154]。定义 $\rho \preceq \rho'$，如果其满足：

$$(src(\rho') + tgt(\rho'))+ \rightarrow_{\rho'} \subseteq (src(\rho) + tgt(\rho))+ \rightarrow_{\rho} \tag{5–32}$$

该序是想刻画 ρ 表示是比 ρ 更加细分的一类相对可达关系。先说明 $\preceq$ 是一个良序。给定一个运行 $\rho = \mathbf{m}_0 \ldots \mathbf{m}_n$ 和其对应的路径 $\pi = \mathbf{a}_1 \ldots \mathbf{a}_n$，定义 $\mathbf{A} \times \mathbb{N}^d$ 上的序关系 $\sqsubseteq$，$(\mathbf{a},\mathbf{m}) \sqsubseteq (\mathbf{a}',\mathbf{m}')$ 当且仅当 $\mathbf{a} = \mathbf{a}'$，$\mathbf{a} \leq \mathbf{a}'$，并且记 $w(\rho) \sqsubseteq w(\rho')$，如果对任意的 $i \in [|\rho|] = [|\rho'|]$ 都有 $(\mathbf{a}_i,\mathbf{m}_i) \sqsubseteq (\mathbf{a}'_i,\mathbf{m}'_i)$。由 Higman 引理 $(\mathbf{A} \times \mathbb{N}^d, \sqsubseteq^*)$ 是一个良序。定义运行上的序关系 $\triangleright$，如果两个运行 ρ,ρ' 满足：

- $src(\rho) \leq src(\rho')$, $tgt(\rho) \leq tgt(\rho')$。
- $w(\rho) \sqsubseteq^* w(\rho')$。

则称 $\rho \rhd \rho'$，显然 $((\mathbf{A} \times \mathbb{N}^d)^*, \rhd)$ 也是一个良序。Leroux 证明了 $\rhd$ 能够蕴含 $\preceq$，从而证明了 $\preceq$ 是一个良序。

定理 5.6 (Leroux[73])　给定 $\mathbf{V}$ 上的两个运行 ρ, ρ'，$\rho \rhd \rho'$ 蕴含了 $\rho \preceq \rho$，从而 $((\mathbf{N}^d)^*, \preceq)$ 是一个良序。

证明　令 $\rho = \mathbf{m}_0 \dots \mathbf{m}_k$。先来说明，存在 $\mathbf{v}_0, \dots \mathbf{v}_{k+1} \in \mathbb{N}^d$ 使得 ρ' 能分解成 $\rho' = \rho_0 \dots \rho_k$ 并且满足 $\mathbf{m}_j + \mathbf{v}_j \xrightarrow{\rho_j} \mathbf{m}_j + \mathbf{v}_{j+1}$。事实上，由定义 $w(\rho') = (\pi_0, \rho_0)(\mathbf{a}_1, \mathbf{m}'_1)(\pi_1, \rho_1) \dots (\mathbf{a}_k, \mathbf{m}'_k)(\pi_k, \rho_k)$，并且对于任意的 $j \in [k]$ 有 $\mathbf{m}'_j \geq \mathbf{m}_j$。从而令 $i \in [k]$, $\mathbf{v}_i = \mathbf{m}'_j - \mathbf{m}_j, \mathbf{v}_0 = src(\rho') - src(\rho), \mathbf{v}_{k+1} = tgt(\rho') - tgt(\rho)$，自然对于 $j \in [k]$ 有 $\mathbf{m}_j + \mathbf{v}_j \xrightarrow{\rho_j} \mathbf{m}_j + \mathbf{v}_{j+1}$。

由引理5.13有 $(\mathbf{v}_j, \mathbf{v}_{j+1})+ \rightarrow_{\rho_j} \subseteq \rightarrow_{\mathbf{m}_j}$，因此 $(\mathbf{v}_0, \mathbf{v_k})+ \rightarrow_{\rho'} \subseteq \rightarrow_\rho$，即 $(src(\rho') + tgt(\rho'))+ \rightarrow_{\rho'} \subseteq (src(\rho) + tgt(\rho))+ \rightarrow_\rho$，从而 $\rho \preceq \rho'$，$((\mathbf{N}^d)^*, \preceq)$ 是一个良序。　□

最后回过头来证明本节的主要结论定理5.4。我们关心的是 $\xrightarrow{*} \cap (\mathbf{m}, \mathbf{n}) + P$ 的性质，这里 $\mathbf{m}, \mathbf{n} \in \mathbb{N}^d$，$P \subseteq \mathbb{N}^d \times \mathbb{N}^d$ 是一个有限生成的周期集。为此定义在 P 上的序关系 $\leq_P$，如果 p, p' 满足：$p' \in p + P$，则称 $p \leq_P p'$，由 Dickson 引理和 P 的有限生成性，$(P, \leq_P)$ 是一个良序。再定义 $\Omega_{\mathbf{m},\mathbf{n},P}$ 是所有起点和终点都在 $(\mathbf{m}, \mathbf{n}) + P$ 的运行的集合，定义其上面的序 $\leq_P$：如果 ρ, ρ' 满足：$\rho \preceq \rho'$，$(src(\rho), tgt(\rho)) - (\mathbf{m}, \mathbf{n}) \leq_P (src(\rho'), tgt(\rho')) - (\mathbf{m}, \mathbf{n})$，则称 $\rho \leq_P \rho'$。显然 $\min_{\leq_P} \Omega_{\mathbf{m},\mathbf{n},P}$ 是有限集，从而有下面引理：

引理 5.16

$$\xrightarrow{*} \cap (\mathbf{m}, \mathbf{n}) + P = \bigcup_{\rho \in \min_{\leq_P} \Omega_{\mathbf{m},\mathbf{n},P}} (src(\rho), tgt(\rho)) + (\rightarrow_\rho \cap P)$$

证明　$\supseteq$ 的方向是显然的。由引理5.13我们有 $(src(\rho), tgt(rho))+ \rightarrow_\rho \subseteq \xrightarrow{*}$，由 $\rho \in \min_{\leq_P} \Omega_{\mathbf{m},\mathbf{n},P}$ 有 $(src(\rho), tgt(\rho)) \in (\mathbf{m}, \mathbf{n}) + P$，从而 $(src(\rho), tgt(\rho)) + (\rightarrow_\rho \cap P) \subseteq (\mathbf{m}, \mathbf{n}) + P$，$\supseteq$ 方向成立。

再来证 $\subseteq$ 方向。令 $(\mathbf{u}, \mathbf{v}) \in \xrightarrow{*} \cap (\mathbf{m}, \mathbf{n}) + P$，则存在 $\rho \in \Omega_{\mathbf{m},\mathbf{n},P}$ 满足 $src(\rho) = \mathbf{u}$, $tgt(\rho) = \mathbf{v}$。由定义存在 $\rho' \in \min_{\leq_P} \Omega_{\mathbf{m},\mathbf{n},P}$ 使得 $\rho' \leq_P \rho$，则不难得出 $(\mathbf{u}, \mathbf{v}) \in (src(\rho'), tgt(\rho'))+ \rightarrow_{\rho'}$，从而 $\subseteq$ 方向成立，引理得证。　□

由引理5.16直接可以推出定理5.4的成立，即可达关系是几乎半线性的，定理得证。

5.3 本章小结

本章介绍了不同于 KLMST 算法的另一个关于可达性算法–Presburger 分离算法。严格来说它由两个半可判定算法组成，其核心是说明无论是否可达，都存在一个证据来说明这件事，即如果可达则存在一条路径，如果不可达则存在一个 Presburger 公式将这两个点区分开来。相比于传统的 KLMST 算法而言，它不用对向量加法系统去做过多的分解，而这一方法也在研究其他问题上起了重要的作用。Bonnet 在 [46] 中用同样的方法证明了带一个测 0 的向量加法系统的可达性问题也是可判定的。

另一方面，不同于 KLMST 算法对任意维给出了 **Ackermann** 的上界和对固定的 d 维给出了 $\mathbf{F}_{d+4}$ 的上界，该方法目前只给出了其判定性的结论，对于其复杂性的结论我们还只有欠缺的认识。Czwre'nski 在 [155] 中证明了某些特定的向量加法系统的不可达格局的分离对需要非常大的周期向量，从而说明该算法也不一定能获得比较好的上界。但是研究其公式的长度还是十分有意义的，目前尽管对任意维的可达性问题已经获得了圆满的结论，但是对于 3 维以上的固定维可达性问题的复杂性还是有相当大的空间，因此分离算法可能能对这一方面的结论做进一步的改进。

第六章　可达性问题的复杂性

向量加法系统[20] 自被提出以来，可达性问题就一直是研究者最为关注的核心问题。在章节四和章节五介绍了可达性问题的两种算法，而这一章将介绍可达性问题的困难性。

关于可达性问题的下界研究也有着非常长的历史。对于一般的向量加法系统，Lipton 在 [89] 所证明的 **EXPSPACE**-难下界曾经是近 40 年内的最好结果，在此期间很多人一直猜想可达性问题是 **EXPSPACE** 完备的，比如 Bouziane 在 [156] 宣称找到了一个双指数空间的算法，但很快被 Jančar 找到了错误[157]。直到 Czerwiński 等人 在 [74, 90] 证明了可达性问题是 **TOWER**-难的，这一结果终结了长久以来对可达性问题是指数空间完备的这一猜想，说明了可覆盖性问题和可达性问题这两者是有着本质上的区别的，同时也说明了可达性问题比可覆盖性问题要难得多。同时在 [74] 中其提出了证明可达性问题下界非常有效的一个技术手段，因而仅仅在两年后，Czerwiński、Lasota、Leroux 就各自提出了新的改进手段[80-82] 最终证明了可达性问题是 **Ackermann** 完备的。

本章将介绍可达性问题下界的证明，其中第一节将介绍下界的证明所用到的核心工具-计数程序，第二节将介绍 Lipton 经典的 **EXPSPACE** 难下界的证明，第三节将介绍 19 年 Czerwiński 所提供的新技术，以及 Czerwiński、Lasota、Leroux 等人最终是如何证明可达性问题是 **Ackermann**-完备的。第四节则是本章小结。

6.1　计数程序介绍

本节将介绍计数程序，这是用来证明可达性下界的核心工具。本节所用到的记号定义的计数程序参考了 [74, 89, 116] 的方式。一个计数器 (counter) c，指的是一个存放了自然数的寄存器，我们用 $v(c)$ 来表示其存放的值。对于一个计数器 c 我们可以有进行如下命令：

- $'+1'$ 命令 $c++$，即将 c 里的值 +1。
- $'-1'$ 命令 $c--$，即将 c 里的值 −1。
- $'?0'$ 命令 **test** $c?$，即测试 c 的值是否为 0，如果为 0 返回 **True**，否则返回 **False**。

注　由头两条命令我们可以简单的定义对任何常数 D 的加减操作，只需执行 D 次 $'+1'$ 或者 $'-1'$ 命令即可，后面会用 $c+=D$ 和 $c-=D$ 来表示这种操作。

一个 k-计数程序则是一个二元组 $M = (C, P)$，其中 C 是所用到的计数器的集合并且 $k = |C|$，P 则是一个程序，一个程序是一段连续的命令，这里的命令除了上述控制计数器相关的命令，还有如下的控制语句：

- 'goto' 命令 **goto** L or L'，其中 L, L' 是程序中的某两条命令的位置，指程序运行完该句命令以后会不确定的从 $P(L)$ 或者 $P(L')$ 开始执行。

并且 P 最后以如下的命令收尾：

- 'halt' 命令 **halt if** $v(c_1), \ldots v(c_n) = 0$，即如果计数器 $c_1, \ldots c_n$ 的值为 0，则程序终止。需要注意的是，'halt' 命令可以没有计数器的判断。

显然可以令 'goto' 命令中 L 和 L' 相等，此时简写为 **goto** L，并且不妨假设程序最后的 'halt' 命令中相关的计数器都至少在前面的一条命令出现过，对于 M 令其大小为程序命令的数目，即 $|M| = |P|$。

注 有的时候为了叙述方便，我们会将一个程序 M 用若干个程序 $M_1, \ldots M_k$ 来表示，这时候引入符号 $M <$ k $>$ 表示程序 M 的第 k 行命令，这样 **goto** $M_i <$ j $>$ 表示程序将跳转到 M_i 的第 j 行命令开始执行。

M 的一个格局是一个二元组 $\mathbf{m} = (\mathbf{v}, i)$，其中 $\mathbf{v}$ 是一个 $|C|$ 维向量记录计数器上的值，用 $\mathbf{v}[c]$ 来表示计数器 c 上的值 $v(c)$，显然 $\mathbf{v} \in \mathbb{N}^k$，特别的用 $\mathbf{1}_c$ 表示 $\mathbf{1}_c[c] = 1$ 其余都为 0 的特殊向量；i 则表示当前运行的命令位置，即接下来 M 将运行命令 $P[i]$，用 # 表示命令已经运行完，特别的如果 $j > |P|$ 也视作 #。下面定义 M 上格局的运行关系 $\xrightarrow{M}$，即如果有下列之一条件满足：

1. $P(i) = c{+}{+}$，则 $j = i + 1$, $\mathbf{v}_2 = \mathbf{v}_1 + \mathbf{1}_c$。
2. $P(i) = c{-}{-}$，则 $j = i + 1$, $\mathbf{v}_2 = \mathbf{v}_1 - \mathbf{1}_c$。
3. $P(i) =$ **test** $c?$，则 $j = i + 1$, $\mathbf{v}_1[c] = 0$, $\mathbf{v}_2 = \mathbf{v}_1$。
4. $P(i) =$ **goto** L or L，则 $j = L$ *or* L', $\mathbf{v}_2 = \mathbf{v}_1$。
5. $P(i) =$ **halt if** $v(c_1), \ldots v(c_n) = 0$，则 $\forall i \in [n]$, $\mathbf{v}_1[c_i] = 0$, $j = \#$, $\mathbf{v}_2 = \mathbf{v}_1$。

我们称 $(\mathbf{v}_1, i) \xrightarrow{M} (\mathbf{v}_2, j)$，$M$ 上的一个运行是一串格局串 $\mathbf{m}_1 \ldots \mathbf{m}_n$ 满足对于 $i \in [n-1]$ 有 $\mathbf{m}_i \xrightarrow{M} \mathbf{m}_{i+1}$，也记作 $\mathbf{m}_1 \xrightarrow{M*} \mathbf{m}_n$，称 $\mathbf{m}_1$ 能运行 M 到 $\mathbf{m}_2$。对于一个格局 $\mathbf{m} = (\mathbf{v}, i)$，如果 $i = \#$ 称其是终止格局，如果不存在任何一个格局 $\mathbf{m}'$ 使其满足 $\mathbf{m} \xrightarrow{M} \mathbf{m}'$ 并且 $\mathbf{m}$ 也不是终止格局，则称格局 $\mathbf{m}$ 是无效的。一个运行如果最后是一个终止格局则称这是一个成功的运行。如果 M 存在一个成功的运行，则称 M 是可以终止的。给定两个程序 M_1, M_2 用 $M_1; M_2$ 表示其顺序连接，即先执行 M_1 的程序再执行 M_2 的程序，而计数器的集合则取并集。设 $M = (C, P)$ 和 D 分别是一个计数程序和一个计数器的集合，则对于 $c \in C$ 令 $M(c \to D)$ 表示将 M 程序中的

所有相关计数器 c 的命令换成 D 中计数器相同的命令，比如对于一条命令 $c++$，我们对每个 $d \in D$ 都在相同位置添加一条 $d++$ 的命令来替换这条命令。

注　关于计数程序 M 还补充以下两点：

- 在所使用的计数器明确的情况下，称一个计数程序 M 只是指其里面的程序 P 会忽略计数器。
- 有的时候会用多个分段程序来表示一个程序，这时候称一个分段程序 P 的成功运行指的是其跳出了这个分段程序 P 进入了其他的程序。

例 6.1　下面是一个简单的计数程序的例子。

计数程序 6–1 一个简单的计数程序 M

1 x_1++;
2 **goto** 1 or 3;
3 x_2++, x_1--;
4 **goto** 3 or 5;
5 **halt if** $v(x_1) = 0$;

我们会将某些命令写在一行，如 M 中的第三行这并不影响阅读。可以看到 M 的作用先非确定给 x_1 赋一个非零的值，然后再将其交换给 x_2，并且 M 终止当且仅当 x_1 的值全部转移给了 x_2。

在计数程序6–1中我们可以看到，其中的两条 'goto' 命令起了一个循环的作用，其分别令第 1 行和第 3 行命令进行了非确定的循环。因此可以定义一条循环命令 'loop'，其效果是对 < body > 部分的命令进行非确定次数的循环，即

1. **Loop**
2. < body >

表示如下的命令:

1. **goto** 2 or 3
2. < body >
3. **goto** 2 or 4

特别的，'loop' 命令是可以不执行里面的循环的，即循环次数可以为 0。

一个初始格局 $\mathbf{m}_0$ 则是将所有计数器都设为 0 并且从程序的第一条命令开始执行，即 $\mathbf{m}_0 = (\mathbf{0}, 1)$，计数程序的停机问题 (halting problem) 则是问从初始格局出发是否存在一个成功的运行。Miner 在 [87] 证明了在这样的计数程序中即使只有两个计数器该模型也是图灵等价的，即：

定理 6.1 (Miner[87]) 2-计数程序的停机问题是不可判定的。

但是当计数程序中的计数器所能达到的值有限制时，关于停机问题则不仅可以获得可判定的结论，还可以获得复杂性完备的结论。具体来说，如果一个计数程序的任何一个格局的每个计数器大小都不会超过 B，则称该计数程序为 B-有界的计数程序。如果该计数程序只有一个计数器，则有：

定理 6.2 (Fearnley[88]) B-有界的 1-计数程序 M 的停机问题是 **PSPACE**-完备的。

如果该计数程序有超过 2 个计数器，则

定理 6.3 (schmitz[86]) 令 $\alpha \geq 3$，$F_\alpha(|M|)$-有界的 2-计数程序 M 的停机问题是 $\mathbf{F}_\alpha$ 完备的。

该结果是证明向量加法系统可达性问题下界的重要手段，在后面可以看到，可达性下界的证明，其实是将 B-有界的计数程序的停机问题规约到了可达性问题上，而思考一下向量加法系统就可以知道，其实向量加法系统的每一维度就是一个没有测 0 指令的计数器，所以问题的关键在于如何去模拟测 0，这也是可达性问题下界研究中的关键之处。

而为了之后能叙述的更加清楚，下面再对计数程序这一概念作一些拓展，将其作更为细致的划分，提出两类特殊的计数器：

- B-有界测 0 计数器 c，c 可以进行测 0 操作，但规定 c 储存的值不能超过 B。
- 无测 0 计数器 c，c 的值可以到达任意大，但是 c 仅可以在程序的 $'halt'$ 命令中进行测 0，即只可以在最后测一次 0。

对于 B-有界测 0 计数器 c，定义一条额外的指令：

- $'?\max'$ 命令 **max** c?，即测试 c 的值是否为 B，如果为 B 返回 **True**，否则返回 **False**。

需要注意的是，这条命令并没有增加了计数器的能力，事实上对于一个 B-有界计数器 c，我们可以添加一个同样的 B-有界计数器 $\hat{c}$，并且保持 $v(c)+v(c')=B$，这样对 c 进行 $'?\max'$ 命令就是对 c' 进行 $'?0'$ 命令，因此该命令的定义只能算是一个语法糖 (syntactic sugar)。

一个 B-有界测 0 计数程序是一个三元组 $\mathbb{M}=(C,T,P)$，其中 P 与计数程序相同是一个程序，C 表示无测 0 计数器的集合，T 表示 B-有界测 0 计数器的集合，这里 C 和 T 表示两个不相交的集合。令 $k=|C\cup T|$ 表示 $\mathbb{M}$ 用到的计数器数量，则 $\mathbb{M}$ 的一个格局是一个表示着当前计数器的值的 k-维向量和当前程序所在位置的标号组成的两元组 $\mathbf{m}=(\mathbf{v},i)$，而关于运行关系，运行等其他概念与计数程序相同。

可以看到，如果对 B-有界测 0 计数程序 $\mathbb{M}$ 中的 C 和 T 进行一些限制，那么 $\mathbb{M}$ 会转换成我们之前所介绍的问题，比如：

- 令 $C = \emptyset$，那么 $\mathbb{M}$ 就是一个 B-有界的计数程序。
- 令 $T = \emptyset$，那么 $\mathbb{M}$ 可以视作带状态的一个向量加法系统。特别的，本章我们后面不作特别的叙述的话向量加法系统指的都是带状态的向量加法系统。

显然由定理6.3，对于 B-有界测 0 计数程序 $\mathbb{M}$ 有如下推论：

推论 6.1　令 $\alpha \geq 3$，对于 $F_\alpha(|\mathbb{M}|)$-有界测 0 计数程序 $\mathbb{M} = (C, T, P)$，如果 $|T| \geq 3$，则其的停机问题是 $\mathbf{F}_\alpha$-难的。

因此如果可以用无测 0 的计算器有效的去模拟一个 B-有界的测 0 计数器，则可以获得相应的下界结论，即推论6.1可以进一步转换为：

推论 6.2　令 $\alpha \geq 3$，对于 $F_\alpha(|\mathbb{M}|)$-有界测 0 计数程序 $\mathbb{M} = (C, T, P)$，如果 C 中的计数器能在 $\mathcal{F}_{<\alpha}(|\mathbb{M}|)$ 内模拟 T 中的计数器，则其的停机问题是 $\mathbf{F}_\alpha$-难的，特别的由此向量加法系统的可达性问题也是 $\mathbf{F}_\alpha$-难的。

由此看到，其证明的关键是在于如何用不能测 0 的计数器去模拟一个能测 0 但是有上界 B 的计数器，在接下来的几节中我们将逐步介绍现有的模拟测 0 的技术。

6.2　EXPSPACE 下界的证明

本节将介绍 Lipton 在 [89] 的成果，他通过成功模拟 2^{2^n}-有界测 0 计数器获得了 **EXPSPACE**-难这一下界。Esparza 也在 [116] 叙述了该成果。

Lipton 提出了一个模拟测 0 的核心想法，即如果要模拟一个 B-有界测 0 计数器 d，其想法是找到一组无测 0 计数器 c, $\hat{c}$ 去保持 $v(c) + v(\hat{c}) = B$，因此可以通过对 $\hat{c}$ 进行 2^{2^n} 次减法来对 c 进行测 0，并且若 $\hat{c}$ 进行不了那么多次减法，则令其进入一个无效格局，这样就完成了一次模拟 d 的测 0 操作。

接下来说明可以用 $8k + 2$ 个无测 0 计数器来模拟一个 2^{2^k}-有界测 0 计数器 c 的操作。首先定义模拟一个 B-有界的计数器 c：

定义 6.1　给定 1 个 B-有界测 0 计数器 c，称 k 个无测 0 计数器 $c_1 \ldots c_k$ 可以在 T 时间内模拟 c 当且仅当存在一函数 $f : \mathbb{N}^k \to \mathbb{N}$，存在四段大小为 $O(T)$ 的程序 M_{inc}, M_{dec}, M_{zero}, M_{max} 满足：

- M_{inc} 对于格局 $(\mathbf{v}, 1)$ 的成功运行的终止格局 $(\mathbf{v}', \#)$ 满足 $f(\mathbf{v}') = f(\mathbf{v}) + 1$。

- M_{dec} 对于格局 $(\mathbf{v},1)$ 的成功运行的终止格局 $(\mathbf{v}',\#)$ 满足 $f(\mathbf{v}')=f(\mathbf{v})-1$。
- M_{zero} 对于格局 $(\mathbf{v},1)$ 有成功的运行当且仅当 $f(\mathbf{v})=0$，并且终止格局 $(\mathbf{v}',\#)$ 满足 $f(\mathbf{v}')=f(\mathbf{v})$。
- M_{max} 对于格局 $(\mathbf{v},1)$ 有成功的运行当且仅当 $f(\mathbf{v})=B$，并且终止格局 $(\mathbf{v}',\#)$ 满足 $f(\mathbf{v}')=f(\mathbf{v})$。
- 程序 M_{inc}，M_{dec}，M_{zero}，M_{max} 的大小不超过 $O(T)$。

例 6.2 下面说明可以用 2 个计数器 $x_r,\hat{x_r}$ 构造一个大小至多为 $O(B)$ 的计数程序来平凡的模拟一个 B-有界计数器 r。事实上，我们可以先添加 B 条命令使 $\hat{x_r}$ 的值变为 B，然后随时令 x_r 的值与 r 相同，并且保持 $v(x_r)+v(\hat{x_r})=B$，则 M_{inc} 与 M_{dec} 来说是显然的，对于 M_{zero} 只需要添加 B 条 $\hat{x_r}--$ 命令去测试 $\hat{x_r}$ 是否能做 B 次减法，再添加 B 条 $\hat{x_r}++$ 令其恢复原状即可，M_{max} 同理。

由例子6.2可以看到，如何有效的去模拟是非常关键的一点。事实上，M_{inc}，M_{dec} 的构造总是简单的，因为 +1 和 −1 是无测 0 计数器也有的功能，而 ′?max′ 命令可以用 ′?0′ 命令来定义，因此可以模拟的关键实际上关于 M_{zero} 的构造。下述引理说明可以在多项式时间内用 $8k+2$ 个无测 0 计数器模拟一个 2^{2^k}-有界测 0 计数器。

引理 6.1 (Liption[89]) 给定 $8k+2$ 个计数器 $x_j,\hat{x_j},y_j,\hat{y_j},s_j,\hat{s_j},b_j,c_j,j\in[k]$ 和 $r,\hat{r}$ 和一个多项式 P，令 $\mathbf{v}\in\mathbb{N}^{8k+2}$ 表示其计数器存的值，则存在一个大小不超过 $P(k)$ 的程序 P 对满足如下要求的格局 $(\mathbf{v},1)$：

- $\mathbf{v}[r]=0,\ \mathbf{v}[\hat{r}]=2^{2^{k+1}}$。
- 对所有 $j\in[k]$ 有 $\mathbf{v}[b_i]=0,\ \mathbf{v}[c_i]=0$。
- 对所有 $j\in[k]$ 有 $\mathbf{v}[x_i]=\mathbf{v}[y_i]=\mathbf{v}[s_i]=0$ 和 $\mathbf{v}[\hat{x_i}]=\mathbf{v}[\hat{y_i}]=\mathbf{v}[\hat{s_i}]=2^{2^i}$。

开始的任何一个成功的运行，其最终格局 $(\mathbf{v}',\#)$ 满足仅有计数器 r 和 $\hat{r}$ 交换所存储的值。

证明 令 C_k 表示这 $8k+2$ 个计数器的集合，无论何时我们令 $\mathbf{v}$ 是满足引理叙述要求的对计数器的赋值。将形如 $*$ 与 $\hat{*}$ 这样的两个计数器视为一组，这里的核心想法是保持每组计数器的值之和不发生变化，比如维持 $v(r)+v(\hat{r})=2^{2^{k+1}}$。接下来对 k 作归纳。$k=0$ 时是简单的，如下所示的程序 $\mathbb{M}_1$ 便可以满足，即对于格局 $(\mathbf{v},1)$ 存在一个成功的运行使得 r 与 $\hat{r}$ 对换其存储的值。

接下来假设 $\leq k-1$ 的情况都成立，考虑 k 的情况，记 $\mathbb{M}_j(t)$ 表示由归纳假设的对 $k=j$ 的程序，并且将其中的计数器 $r,\hat{r}$ 换成了 $t,\hat{t}$，特别的令 $\hat{\hat{*}}=*$。考虑如下的程序 $\mathbb{M}_k$:

计数程序 6–2 $\mathbb{M}_1$

```
1 goto 1 or 3;
2 r − −, r + +;
3 r = r + 4, r̂ = r − 4;
```

计数程序 6–3 $\mathbb{M}_k$

```
1 goto 1 or 3;
2 r − −, r + +;
3 Loop;
4    x_k + +, x̂_k − −;
5    Loop;
6       y_k + +, ŷ_k − −;
7       r + +, r̂ − −;
8    M_{k−1}(ŷ_k);
9 M_{k−1}(x̂_k);
```

考察从引理叙述的格局 $(\mathbf{v}, 1)$ 出发，注意到 $v(\hat{x_k}) + v(x_k) = v(\hat{y_k}) + v(y_k) = 2^{2^k}$，由归纳假设 $\mathbb{M}_{k-1}(\hat{y_k})$ 能有一次成功的运行当且仅当进入时 $v(\hat{y_k}) = 0,\ v(y_k) = 2^{2^k}$，并且结束时 $v(y_k) = 0,\ v(\hat{y_k}) = 2^{2^k}$，同理 $\mathbb{M}_{k-1}(\hat{x_k})$。因此对于 $\mathbb{M}_k$ 的一个成功运行，相当于在 3 − 9 行对 r 做了次 2 层循环其中每层循环次数为 2^{2^k} 的减法，因此对从 $(\mathbf{v}, 1)$ 出发的成功运行的终止格局 $(\mathbf{v}', \#)$ 一定有：

- $\mathbf{v}'[r] = \mathbf{v}[r] - 2^{2^{k+1}} = 0,\ \mathbf{v}'[\hat{r}] = \mathbf{v}[\hat{r}] + 2^{2^{k+1}} = 2^{2^{k+1}}$。
- 对任意的 $c \in C_k$ 有 $\mathbf{v}'[c] = \mathbf{v}[c]$。

因此 $\mathbb{M}_k$ 从效果上是一个符合要求的程序 P，但是需要注意的是，由于 $\mathbb{M}_j$ 每次调用了两次 $\mathbb{M}_{j-1}$，因此计数程序6–3的长度会有 $O(2^k)$ 那么长，下面通过一个小的技巧来使最终 $\mathbb{M}'_k$ 依旧是多项式长的。

注意到在 $\mathbb{M}_k$ 中由于分别要对 $\hat{x_k}$, $\hat{y_k}$ 测 0，因此 $\mathbb{M}_{k-1}$ 在 $\mathbb{M}_k$ 中被复制了两次，如果将其转移到同一个计数器上测 0，便可以只用一次 $\mathbb{M}_{k-1}$，从而避免了 $\mathbb{M}_k$ 的指数长度增长。具体来说将 $\mathbb{M}_{k-1}(\hat{y_k})$ 改写成如下的程序 $\mathbb{M}_{k-1}[y]$。

其中 $\mathbb{M}_{k-1}[s]$ 的定义如计数程序6–5所示。$\mathbb{M}_{k-1}[x]$ 的定义也是类似的，仅仅需要将 $\mathbb{M}_{k-1}[y]$ 中将 $x_k, \hat{x_k}, b_k$ 与 $y_k, \hat{y_k}, c_k$ 互换，并且将 $\mathbb{M}_{k-1}[y]$ 中的第 7 行命令改写成 **goto** $\mathbb{M}_{k-1}[s] < 3 >$ 即可。最终修改后的程序 $\mathbb{M}'_k$ 如计数程序6–6所示。

由于 $\mathbb{M}_{k-1}[x]$, $\mathbb{M}_{k-1[y]}$ 都是常数长的，而 $\mathbb{M}_{k-1}[s]$ 只调用了一次 $\mathbb{M}_{k-1}$，因此 $\mathbb{M}'_k$ 对于 k 来说是多项式长的，更严格的说其大小关于 k 是线性的。接下来说明

计数程序 6–4 $\mathbb{M}_{k-1}[y]$

1 **goto** 1 or 3;
2 $\hat{y_k}--,\ y_k++;$
3 **Loop**;
4 $\quad \hat{y_k}++,\ y_k--;$
5 $\quad s_k++,\ \hat{s}_k--;$
6 $b_k++;$
7 **goto** $\mathbb{M}_{k-1}[s] < 1 >;$
8 $c_k--;$

计数程序 6–5 $\mathbb{M}_{k-1}[s]$

1 $b_k--;$
2 **goto** 4;
3 $c_k--;$
4 $\mathbb{M}_{k-1}(\hat{s_k});$
5 **goto** 6 or 8;
6 $c_k++;$
7 **goto** $\mathbb{M}_{k-1}[y] < 8 >;$
8 $b_k++;$
9 **goto** $\mathbb{M}_{k-1}[x] < 8 >;$

$\mathbb{M}_{k-1}[y]$ 能起到与 $\mathbb{M}_{k-1}(\hat{y_k})$ 相同的效果。$\mathbb{M}_{k-1}[x]$ 和 $\mathbb{M}_{k-1}(\hat{x_k})$ 也是类似的。事实上只需说明从 $(\mathbf{v}, 1)$ 出发运行 $\mathbb{M}'_k$ 到第 9 行时，其计数器必须满足 $v(\hat{y_k}) = 0$ 才能进入 $\mathbf{M}_{k-1}[y]$ 后存在一个成功的运行，且运行后的格局必须满足只有 $\hat{y_k}$ 与 y_k 的值发生交换，其余寄存器的值并不发生改变。事实上，进入 $\mathbb{M}_{k-1}$ 时由条件 $v(y_k) = 2^{2^k}$ 并且 $s_k, \hat{s_k}, b_k, c_k$ 寄存器里的值不会发生变化，即此时 $v(\hat{y_k}) = v(\hat{s_k}) = v(b_k) = v(c_k) = 0$。设此时在 $\mathbb{M}_{k-1}[y]$ 的格局为 $\mathbf{m} = (\mathbf{v}, 1)$，则格局 $\mathbf{m}$ 在 $\mathbb{M}_{k-1}$ 要有一个成功的运行当且仅当满足以下两件事：

- 能成功运行过第 7 行命令，即运行到格局 $(\mathbf{v}_1, 7)$ 时有归纳假设我们有 $v(\hat{s_k}) = 0$。
- 格局在子程序 $\mathbb{M}_{k-1}[s]$ 中执行第 7 行命令跳回 $\mathbb{M}_{k-1}$。

可以看到在 $\mathbb{M}_{k-1}[y]$ 中 $\hat{s_k}$ 与 $\hat{y_k}$ 的变化量是一样的，因此第 1 点要能满足当且仅当格局 $\mathbf{m}$ 满足 $\mathbf{v}[y_k] = 2^{2^k}$，也即 $\mathbf{v}[\hat{y_k}] = 0$。关于第 2 点则可以由 b_k，c_k 的值保证，因为如果 $\mathbb{M}_{k-1}[s]$ 在第 5 行选择了错误的跳转，则格局会因为 b_k，c_k 的值降到 0 以下而无法运行下去。因此 $\mathbb{M}'_k$ 是个满足要求的程序，引理得证。 □

计数程序 6–6 $\mathbb{M}'_k$

```
1  goto 3 or 4;
2  M_{k-1}[s];
3  r − −, r + +;
4  Loop;
5     x_k + +, x̂_k − −;
6     Loop;
7        y_k + +, ŷ_k − −;
8        r + +, r̂ − −;
9     M_{k-1}[y];
10 M_{k-1}[x];
```

引理6–3证明了如果计数器的初始值满足一定的条件则可以对一个 2^{2^k}-有界的计数器进行测 0，但是我们还需要证明可以将计数器设定到这样的初始值，这就是如下引理所要阐述的。

引理 6.2 (Lipton[89])　给定 $8k+2$ 个计数器 $x_j, \hat{x_j}, y_j, \hat{y_j}, s_j, \hat{s_j}, b_j, c_j, j \in [k]$ 和 $r, \hat{r}$ 和一个多项式 P，令存在一个大小不超过 $P(k)$ 的程序 P 对初始格局 $(\mathbf{v}, 1)$ 的一个成功的运行，其最终格局 $(\mathbf{v}', \#)$ 满足如下条件：

- 对于所有 $i \in [k]$ 有 $v(x_i) = v(y_i) = v(s_i) = v(b_i) = v(c_i) = 0$。
- $v(r) = 0,\ v(\hat{r}) = 2^{2^{k+1}}$。
- $v(\hat{x_i}) = v(\hat{y_i}) = v(\hat{s_i}) = 2^{2^i}$。

证明　依旧是对 k 做归纳，$k = 0$ 的时候是显然的。假设 $\leq k-1$ 的时候引理成立，即存在一个满足要求的程序 $\mathbb{N}_{k-1}$ 可以将计数器预设到上述值，为了构造 $\mathbb{N}_k$，还需要在 $\mathbb{N}_{k-1}$ 的基础上完成如下两点：

- 将 $\hat{x_k}$, $\hat{y_k}$, $\hat{s_k}$ 的值设置成 2^{2^k}。
- 将 $\hat{r}$ 的值设置成 $2^{2^{k+1}}$。

接下来说明这两点。由引理6.1，如果进入 $\mathbb{M}_k$ 的格局 $(\mathbf{v}, \mathbb{M}_k < 1 >)$ 满足：

- $\mathbf{v}[r] = 0,\ \mathbf{v}[\hat{r}] = 2^{2^{k+1}}$。
- 对所有 $j \in [k]$ 有 $\mathbf{v}[b_i] = 0,\ \mathbf{v}[c_i] = 0$。
- 对所有 $j \in [k]$ 有 $\mathbf{v}[x_i] = \mathbf{v}[y_i] = \mathbf{v}[s_i] = 0$ 和 $\mathbf{v}[\hat{x_i}] = \mathbf{v}[\hat{y_i}] = \mathbf{v}[\hat{s_i}] = 2^{2^i}$。

则其是一个可以用计数器 r 模拟 2^{2^k} 有界测 0 计数器的程序，因此构造程序如下：

这里 $\mathbb{M}_{k-1}(\hat{x_k})$, $\mathbb{M}_{k-1}(\hat{y_k})$ 与引理6.1定义相同。$\mathbb{N}_k$ 的长度是关于 k 的多项式，因为在递归构造的时候只引用了一次 $\mathbb{N}_{k-1}$ 并且由引理6.1 $\mathbb{M}_{k-1}(\hat{x_k})$, $\mathbb{M}_{k-1}(\hat{y_k})$ 的长度也是关于 k 多项式长的。

计数程序 6–7 $\mathbb{N}_k$

```
1 ℕ_{k-1}[r → {x_k, y_k}, r̂ → {x̂_k, ŷ_k}];
2 Loop;
3     x_k ++, x̂_k − −;
4     Loop;
5         y_k ++, ŷ_k − −;
6         r̂ ++;
7     𝕄_{k-1}(ŷ_k);
8 𝕄_{k-1}(x̂_k);
```

最后说明 $\mathbb{N}_k$ 是符合要求的程序。由归纳假设 $\mathbb{N}_k$ 的一个由初始格局出发成功运行第一行后有 $v(x_k) = v(y_k) = 0$, $v(\hat{x_k}) = v(\hat{y_k}) = 2^{2^k}$。因此由引理6.1 $\mathbb{M}_{k-1}(\hat{x_k})$, $\mathbb{M}_{k-1}(\hat{y_k})$ 是一个可以模拟 x_k 和 y_k 测 0 的程序，因此最后格局 $(\mathbf{v}, \#)$ 必须满足 $\mathbf{v}[r] = 2^{2^k} \cdot 2^{2^k} = 2^{2^{k+1}}$, $\mathbf{v}[\hat{x}_k] = \mathbf{v}[\hat{y}_k] = 2^{2^k}, \mathbf{v}[x_k] = \mathbf{v}[y_k] = 0$，引理得证。 □

由引理6.1和引理6.2可以获得可达性问题的指数空间下界，即如下定理：

定理 6.4 (Lipton[89]) 向量加法系统的可达性问题是 **EXPSPACE**-难的。

证明 由引理6.1和引理6.2可以用 $8k + 2$ 个无测 0 计数器模拟 $2^{2^{k+1}}$-有界测 0 计数器，并且其中 $8k$ 个无测 0 计数器可以复用。因此由推论6.2可知向量加法系统的可达性问题是 **EXPSPACE**-难的。 □

注意到构造的模拟测 0 的程序是有界的，即从初始格局出发所能到达的格局是有限的，我们不难得到可覆盖性问题和有界性问题的复杂性的精确结论。

推论 6.3 (Rackoff[93],Lipton[89]) 向量加法系统的可覆盖性问题和有界性问题都是 **EXPSPACE**-完备的。

证明 由定理3.7和定理3.5只需要证明其是 **EXPSPACE**-难的。由前面的构造可知，可覆盖性的 **EXPSPACE**-难的结论是直接明显的，关于有界性问题，由于上述构造的模拟程序从初始格局出发的格局是有界的，所以可以构造出一个无界的向量加法系统使得判断其是否有界必须要用到指数大小的空间，因此有界性问题也是 **EXPSPACE**-难的。 □

6.3 Ackermann 下界的证明

上一节介绍了 Lipton 关于可达性问题的下界成果，其最重要的思想是用不能测 0 的计数器在多项式时间内模拟了一个 2^{2^k}-有界测 0 计数器 c。其用来模拟测 0 的方法是维持一组计数器满足 $v(c)+v(\hat{c})=2^{2^k}$，则对 $v(c)=0$ 当且仅当 $\hat{c}$ 能做 2^{2^k} 次减法，换句话说如果 c 不是 0，程序会因为不能对 $\hat{c}$ 做 2^{2^k} 次减法而终止。

一个很自然的思路是能不能构建一个更大的 B，即在维持 $v(c)+v(\hat{c})=B$ 的情况下如果能精确的控制对 $\hat{c}$ 做 B 次减法，那么就能模拟 B-有界测 0 的计数器，但是从 Lipton 的方法出发会出现如下的问题：

- 注意到 $F_{i+1}(n)=f_i^{n+1}(n)$，而 Lipton 的构造出的模拟程序已经是关于 k 多项式的，因此想以此构造出 $F_3(n)$ 之上的模拟程序长度也会爆炸增长到 $F_3(n)$ 之上。
- Lipton 的构造方法对可覆盖性问题也是同样适用的，而可覆盖性问题已经在 [93] 中被证明是 **EXPSPACE**-完备的，因此如果存在一个更大的 B 使其能做精确的 B 次减法便会与可覆盖性问题现有的结论矛盾。

前者其实是一个如何精确计算 B 的问题，后者则是一个如何对 B-有界的计数器测 0 的问题。Czerwiński 等人在 2019 年提出了一个新的测 0 想法[74]，简单来说其并不要求每次测 0 的时候一定要精确的 B 次减法，但如果没有做足 B 次减法，那么在运行中便会留下一些无法在后面去除的副作用，从而在最终的时候运行不了 'halt' 命令。下面会借助一个例子再来帮助理解该想法。

例 6.3　考察如下的计数程序 $\mathbb{P}_1$，其中 $\mathbb{P}$ 是一个不用到计数器 x 的程序，并且不妨假设开始的时候计数器 x 里的存的值为 $B>0$。则可以看到尽管 $\mathbb{P}_1$ 可以任意的选择执行第二行命令的次数，但如果其没有精准的执行 B 次，那么 x 里的值就不能变为 0，因此程序最后也无法终止。

计数程序 6–8 $\mathbb{P}_1$

```
1 Loop;
2    x − −;
3 ℙ;
4 halt if v(x) = 0;
```

关于这个技术将在章节6.3.2中做更为细致详细的介绍。

另一方面，关于如何精准的计算出一个更大的 B，比如 $B=2^{\cdot^{\cdot^{\cdot^{2}}}\}x}$ 也是下界证明需要解决的一个方面。事实上考虑 Lipton 的结果，其实际上是用常数个计数

器构建了一个常数大小的精确计算乘法的程序，并以此作迭代获得了一个能精确计算 2^{2^B} 的程序。Czerwiński 等人在 [74] 利用如下的事实：$\Pi_{i=1}^{k-1}\frac{i+1}{i}=k$ 构造了一个常数大小的可以精确计算阶乘的程序再进行 n 次迭代，从而获得了一个能精确计算 $O(B=2^{\cdot^{\cdot^{\cdot^{2}}}}\}x)$ 的程序，并且其将这种方法称为扩大器 (ampilifer)，我们将在下一节6.3.1中作详细的介绍。

6.3.1 计数程序中的放大器

本节将介绍 Czerwiński 等人在 [74] 介绍的一个帮助模拟测 0 的技术-放大器。首先来介绍一种新的测 0 方法，假设需要模拟一个 B-有界测 0 计数器的测 0，考察如下的程序 $\mathbb{P}_2=(C,T,P)$，其中 $C=\{x,\hat{x},b,c,d\}$, $T=\emptyset$:

计数程序 6–9 $\mathbb{P}_2$

```
1 Loop;
2   x++, x̂--;
3   d--;
4 c--;
```

考虑其一个格局 $\mathbf{m}=(\mathbf{v},1)$ 满足:

1. $\mathbf{v}[x]+\mathbf{v}[\hat{x}]=B$。
2. $\mathbf{v}[b]=B$, $\mathbf{v}[d]=\mathbf{v}[b]\cdot\mathbf{v}[c]$。

从 $\mathbf{m}$ 出发到终止格局的一个运行满足对其中的任何一个格局 $\mathbf{m}'(\mathbf{v},i)$ 都有:

- $\mathbf{v}'[x]+\mathbf{v}'[\hat{x}]=B$。
- $\mathbf{v}'[d]-\mathbf{v}[d]\leq B$。

因此，对于 $\mathbf{m}$ 出发的终止格局 $\mathbf{m}_f=(\mathbf{v}_f,\#)$ 依旧满足 $\mathbf{v}_f[d]=\mathbf{v}_f[b]\cdot\mathbf{v}_f[c]$ 当且仅当第 2 行命令执行了 B 次，也即 $\mathbf{v}_f[x]=0$, $\mathbf{v}_f[\hat{x}]=B$。

这段程序可以视作对 x 的一次测 0，令 $\mathbb{P}_2[\hat{x}]$ 表示将其中的 x 与 $\hat{x}$ 对换，我们可以用这个思路来完成对一个 B-有界测 0 计数器 r 的模拟。简单来说，对于一个包含 r 的计数程序，首先用一组无测 0 计数器 $x,\hat{x}$ 来模拟其运行，即保持其和不变并且 x 里储存的值恰好为 r 里储存的值。然后预先猜测该程序的一个成功的运行所需要对 r 测 0 的次数 R，假设还存在额外的三个无测 0 计数器 b,c,d，令其初始的赋值满足 $v(b)=B$, $v(c)=2t$, $v(d)=B\cdot R$，最后将程序中对 r 测 0 的命令全部换成 $\mathbb{P}'=\mathbb{P}_2;\mathbb{P}[\hat{x}]$，根据上面的分析，如果运行到最后 d 中的值为 0，则因为 $\mathbb{P}'$ 根据假设运行了 R 次，因此最后 $v(d)=0$ 当且仅当每次 $\mathbb{P}'$ 中 d 的值被减了 $2B$ 次，也即 x 里的值开始是 0 结束运行 $\mathbb{P}'$ 的时候也是 0，因此对 r 的 R 次测 0 成功

当且仅当终止格局里 $v(d) = 0$，即可以用最后 d 里的值来确保前面对 r 的测 0 全部成功。

注意到相比于 Lipton 的方法，该方法并不需要每次测 0 的时候去进行精确的 B 次减法，即其允许没减够 B 次，但是一旦没减足，其会通不过最后的终止命令，即会在程序最后受到惩罚。通过这样的一个松弛的想法，Czerwiński 等人得到了一个更好的下界结果。接下来本节将详细的介绍这个思路。首先介绍在 [74] 提出的放大器的概念。

定义 6.2 (放大器, Czerwiński[74])　给定两个自然数 B, R，一个 (B, R)-放大器是一个 B-有界测 0 计数程序满足考虑其中的 3 个无测 0 计数器 b, c, d，对于其任何一个成功的运行的最终格局 $\mathbf{m} = (\mathbf{u}, \#)$ 都满足:

$$\mathbf{u}(b) = R,\ \mathbf{u}(c) > 0,\ \mathbf{u}(d) = \mathbf{u}(b) \cdot \mathbf{u}(c) \tag{6–1}$$

并且 $\mathbf{u}[c]$ 的取值范围是无限的。

例 6.4　考察程序 $\mathbb{P}_3$，令 R 是一个常数，显然对其的任何一个成功运行都满足条件6–1，并且由于其没有使用任何测 0 的计数器，因此对于任意的 $B \in \mathbb{N}$，程序 $\mathbb{P}_3$ 是一个 (B, R)-放大器。

计数程序 6–10 $\mathbb{P}_3$

```
1 b+ = R;
2 d+ = R, c++;
3 Loop;
4     d+ = R, c++;
5 halt;
```

注　特别的称 $\mathbb{P}_3$ 这种不用到测 0 计数器的放大器为 R-生成器。给定一个 $(B, f(B))$-放大器，如果将其的 B-有界测 0 计数器换成 B_1-有界测 0 计数器以后所得的 B_1-有界测 0 计数程序依旧是 $(B_1, f(B_1))$ 放大器，则称该程序是 f-放大器。

接下来说明放大器的作用。假设 $\mathbb{A} = (C_A, T_A, P_A)$ 是一个 (B, R)-放大器，$\mathbb{P} = (C_P, T_P, P_P)$ 是一个 R-有界测 0 程序，则可以按如下的方式定义一个 B-有界测 0 程序 $\mathbb{A} \triangleright \mathbb{P} = (C, T, P)$:

- 将计数器重命名使 $\mathbb{P}$ 和 $\mathbb{A}$ 里的计数器互不相同，并且令 $C = C_A$。

- 令 $T = T_A \cup T_P \cup T_C$，其中 $T_C = \{x_r, \hat{x_r} | r \in C_P\}$，即对于 $\mathbb{P}$ 中每一个 R-有界测 0 计数器 r，都用一组计数器 $x_r, \hat{x_r}$ 去代替。
- $P = A'_P; P'_P$，其中 $A'_P; P'_P$ 是 A_P, P_P 根据如下方式作修改而来：
  - A'_P 是 A_P 去除最后一行 'halt' 命令所遗留的程序。
  - 令 $T_C = \{x_i, \hat{x_i} | i \in [k]\}$，则在 P_P 最前面添加如下初始化的命令：

    Loop;

    $\hat{x_1}++, \ldots \hat{x_k}++$;

    $b--,\ d--$;

    $c--$。
  - 对于 $r \in C_P$，P_P 中每一条 $r++$ 的命令在 P_P 中都被改写成如下的命令：

    $x_r++,\ \hat{x_r}--$;
  - 对于 $r \in C_P$，P_P 中每一条 $r--$ 的命令在 P_P 中都被改写成如下的命令：

    $x_r--,\ \hat{x_r}++$;
  - 对于 $r \in C_P$，P_P 中每一条 **test** r? 的命令在 P_P 中都被改写成如下的程序 $\mathbb{P}_{zero}[x_r]$：

    Loop;

    $x_r++,\ \hat{x_r}--$;

    $d--$;

    $c--$;

    Loop;

    $\hat{x_r}++,\ x_r--$;

    $d--$;

    $c--$;
  - 对于 $r \in C_P$，P_P 中每一条 **max** r? 的命令在 P'_P 中都被改写成 $\mathbb{P}_{zero}[\hat{x_r}]$，即将其中的 x_r 与 $\hat{x_r}$ 对换。
  - 令 $y_1, \ldots y_m$ 是 A_P 最后一行终止所需测 0 的计数器，$z_1, \ldots z_n$ 是 P_P 最后一行终止所需测 0 的计数器，则 P'_P 将 P_P 的最后一行 'halt' 命令改写成：

    halt if $v(d), v(y_1), \ldots v(y_m), v(z_1), \ldots v(z_n) = 0$.

接下来的引理说明，构造出的 B-有界计数程序 $\mathbb{A} \triangleright \mathbb{P}$ 与 R-有界计数程序 $\mathbb{P}$ 效果是一样的。为了方便叙述，不妨假设 T_C 中的 x_r 与替代的 r 重名。令 $\cong$ 和 $\mathbf{v}$ 分别代表某两个计数器集合 C_1 和 C_2 的计数器的值的向量，定义 $\mathbf{u} \overset{\mathrm{v}}{=} \mathbf{v}$ 表示 $\mathbf{u}$ 和 $\mathbf{v}$ 在同一个计数器上的值相同，即对于任何一个 $c \in C_1 \cap C_2$ 都有 $\mathbf{u}[c] = \mathbf{v}[c]$。

引理 6.3 (Czerwiński[74])　对于 $C_P \cup T_P$ 中的计数器的一个存储值向量 v，其可能出现在 $\mathbb{P}$ 的一个成功运行的最终格局上当且仅当其也会出现在 $\mathbb{A} \triangleright \mathbb{P}$ 的一个成功运行的最终格局上，即存在一个 $\mathbb{P}$ 的成功运行的最终格局 $\mathbf{m}_1 = (\mathbf{u}_1, \#)$ 满足 $\mathbf{u}_1 \overset{\mathrm{v}}{=} \mathbf{v}$ 当且仅当存在一个 $\mathbb{A} \triangleright \mathbb{P}$ 的成功运行的最终格局 $\mathbf{m}_2 = (\mathbf{u}_2, \#)$ 满足 $\mathbf{u}_2 \overset{\mathrm{v}}{=} \mathbf{v}$。

证明　首先证明 $\Rightarrow$ 方向，对于 $\mathbb{P}$ 上的一个成功运行的最终格局 $\mathbf{m}_1 = (\mathbf{u}, \#)$，不妨假设在运行过程中一共进行了 r 次 $'?0'$ 和 $'?max'$ 命令，则可以按下面的方式构建一个 $\mathbb{A} \triangleright \mathbb{P}$ 的运行：

- $\mathbb{A} \triangleright \mathbb{P}$ 先运行 $\mathbb{A}$ 的部分将 b, c, d 里的寄存器存储的值设为 $v(b) = R,\ v(c) = 2r + 1,\ v(d) = (2r + 1)R$。
- 运行到 $\mathbb{A} \triangleright \mathbb{P}$ 中 $\mathbb{P}$ 的部分时，在一开始的初始化，令整个循环一共走了 R 次减法而终止。
- 每次运行到原来 $\mathbb{P}$ 中对应 $'?0'$ 和 $?max$ 的命令的时候，令对应的 $\mathbb{P}_{zero}[x_r]$ 与 $\mathbb{P}_{zero}[\hat{x_r}]$ 中两个循环都做满，具体来说是每个循环都做了 R 次。
- 其余的运行与 $\mathbb{P}$ 上的成功运行保持一致。

下面说明，这样的一个运行是 $\mathbb{A} \triangleright \mathbb{P}$ 上的一个成功运行，并且其最终格局 $\mathbf{m}_2 = (\mathbf{u}_2, \#)$ 满足 $\mathbf{u}_1 \overset{\mathrm{v}}{=} \mathbf{u}_2$。为此只需要说明两件事：

- 在 $\mathbb{P}_{zero}[x_r]$ 与 $\mathbb{P}_{zero}[\hat{x_r}]$ 的一次运行相当于对 r 进行了一次 $'?0'$ 和 $'?\mathrm{max}'$ 命令。
- 该运行能通过最后的 $'\mathrm{halt}'$ 命令。

第一点是非常显然的，由于令 $\mathbb{P}$ 部分开始初始化的时候整个循环一共执行了 R 次，因此对于任意的 $c \in C_P$ 初始化后都有 $v(x_r) + v(\hat{x_r}) = R$，从而 $\mathbb{P}_{zero}[x_r]$ 每个循环能做满 R 次当且仅当在进入 $\mathbb{P}_{zero}[r]$ 的时候有 $v(x_r) = 0$，并且执行完整段 $\mathbb{P}_{zero}[x_r]$ 后依旧保持 $v(x_r) = 0$，唯一有的变化是 d 里的值减少了 $2R$，而 c 里的值减少了 2。因此第一点成立，关于第二点由构造方法只需要验证 d 是否是 0 即可。注意到在初始化的时候 d 会被减去 R，而在每次在 $\mathbb{P}_{zero}[x_r]$ 与 $\mathbb{P}_{zero}[\hat{x_r}]$ 中都被减去 $2R$ 里，而一共有 r 次 $\mathbb{P}_{zero}[x_r]$ 与 $\mathbb{P}_{zero}[\hat{x_r}]$ 的运行，因此在最后 d 的值一共被减去了 $R + 2rR = (2r + 1)R$ 次，即 $v(d) = 0$，该运行会是一个成功的运行，$\Rightarrow$ 方向得证。

接下来证明 $\Leftarrow$ 方向，注意到对于 $\mathbb{A} \triangleright \mathbb{P}$ 的一个成功运行，其最后必须满足 $y_1 \ldots y_m$ 里的值为 0，因此在运行完 $\mathbb{A}$ 的部分后，b, c, d 里的值一定满足：

$$v(b) = R,\ v(c) > 0,\ v(d) = v(b) \cdot v(c) \tag{6–2}$$

因此在经过 $\mathbb{P}$ 的初始化之后，对于任意一个 $r \in C_P$ 有：

$$v(x_r) + v(\hat{x_r}) \leq R,\ v(d) \geq v(c) \cdot R \tag{6–3}$$

我们说明在运行的任何时刻如果出现 $v(d) > v(c) \cdot R$，则该运行就不可能是一个成功的运行。事实上，除了 $\mathbb{P}$ 开头的初始化，其余会影响 c, d 的值的变化的命令仅在 $\mathbb{P}_{zero}[x_r]$ 或 $\mathbb{P}_{zero}[\hat{x_r}]$ 出现过，而每执行完一次 $\mathbb{P}_{zero}[x_r]$ 或 $\mathbb{P}_{zero}[\hat{x_r}]$，$c$ 的值会减少 2，而 d 的值最多会减少 $2R$，因此 $v(d) - v(c) \cdot R$ 的值在运行过程中只会增加不会减少，所以一旦 $v(d) - v(c) \cdot R > 0$，便会导致在最终有 $v(d) > 0$。

因此在 $\mathbb{A} \triangleright \mathbb{P}$ 的任何一个成功的运行中，在其 $\mathbb{P}$ 初始化的部分以及 $\mathbb{P}_{zero}[x_r]$ 与 $\mathbb{P}_{zero}[\hat{x_r}]$ 的运行每次循环都是做满 R 次的，从而每次 $\mathbb{P}_{zero}[x_r]$ 与 $\mathbb{P}_{zero}[\hat{x_r}]$ 都可分别视作对 r 的一次 ′?0′ 与 ′?max′ 命令，假设其最终格局为 $\mathbf{m}_1 = (\mathbf{u}_1, \#)$，则会存在 $\mathbb{P}$ 上的一个成功运行，其最终格局 $\mathbf{m}_2 = (\mathbf{u}_2, \#)$ 满足：$\mathbf{u}_1 \overset{\mathrm{v}}{=} \mathbf{u}_2$，引理得证。□

引理6.3介绍了模拟测 0 的一个新方法，事实上假设存在一个 B-放大器 $\mathbb{P}$，则对于任何一个 B-有界测 0 程序 T，由引理6.3程序 $\mathbb{P} \triangleright \mathbb{T}$ 与 $\mathbb{T}$ 有着一致的成功运行，即可以用 $\mathbb{P}$ 来模拟 B-有界测 0 计数器的行为。另一方面，给我们一个 $(B, f(B))$-放大器，根据引理6.3也可以用 $\triangleright$ 构造一个更大的放大器，由上可得如下定理：

定理 6.5 如果存在一个常数大小的 f-放大器 $\mathbb{P}$，则对于一个多项式 P，向量加法系统可以在多项式时间内模拟 $f^{P(n)}(B)$-有界测 0 计数器的命令。

证明 证明借助例子6.4里的程序 $\mathbb{P}_3$，并且不放这里混淆 $\mathbb{P}_3$ 里的 R 与 B，考察如下的程序 $\mathbb{T}$：

$$\mathbb{T} \overset{\text{def}}{=} \underbrace{\mathbb{P}_3 \triangleright \mathbb{P} \triangleright \cdots \triangleright \mathbb{P}}_{P(n)} \tag{6–4}$$

由 $\triangleright$ 的构造定义我们可知 $\mathbb{T}$ 是多项式大小的，并且由引理6.3可知，对于常数 B，$\mathbb{T}$ 是一个 $f^{P(n)}(B)$-生成器，即 $\mathbb{T}$ 没有用到任何测 0 的计数，但是其存在三个计数器 b, c, d 对于任何其一个成功的运行，这三个计数器的值满足：

$$v(b) = f^{P(n)}(B),\ v(c) > 0,\ v(d) = v(b) \cdot v(c) \tag{6–5}$$

因此对于任何一个 $f^{P(n)}(B)$-有界测 0 程序 $\mathbb{M}$，由引理6.3可以用 $\mathbb{T} \triangleright \mathbb{M}$ 这样一个没有测 0 计数器的计数程序去模拟，因此向量加法系统可以在多项式时间内模拟 $f^{P(n)}(B)$-有界测 0 计数器的命令，引理得证。□

定理6.5提供了一种新的测 0 方法，也提供了一个新的证明下界的手段，而接下来的例子就给出了用这种新方法所得出的一种有别于 Lipton 的证明的 **EXPSPACE** 下界证明。

例 6.5　考察如下的 (B, B^2)-有界放大器 $\mathbb{P}_{exp}$，其用到 2 个 B-有界计数器 x, y，其余的 b, c, d 都是无测 0 计数器：

计数程序 6–11 $\mathbb{P}_{exp}$

```
1  Loop;
2    x++, y++, b++;
3  max x?;
4  Loop;
5    Loop;
6      x−−;
7      Loop;
8        y−−, d++;
9      test y?, y ← B;
10   test x?, x ← B, c++;
11 halt;
```

这里 $x \leftarrow B$ 表示将 x 的值赋值成 B，可用下面的命令来表示：

- **Loop**;
  x++;
  max x?;

则由定理6.5，向量加法系统可以模拟 B^{2^n}-有界测 0 计数器的命令，即其可达性问题是 **EXPSPACE**-难的。特别的，计数程序 $\mathbb{T} = \underbrace{\mathbb{P}_{exp} \rhd \ldots \rhd \mathbb{P}_{exp}}_{n}$ 是一个 (B, B^{2^n})-放大器，因此计数程序 $\mathbb{P}_3 \rhd \mathbb{T}$ 是一个 B^{2^n}-放大器。

最后来解释一下为什么需要的是一个常数大小的放大器，如果存在一个关于 B 多项式大小的 $(B, f(B))$-放大器 $\mathbb{P}$，那么 $(f^k(B), f^{k+1}(B))$-放大器 $\mathbb{P}$ 的大小就是关于 $f^k(B)$ 的多项式大小，因此考虑程序 $\underbrace{\mathbb{P} \rhd \ldots \rhd \mathbb{P}}_{n}$，其程序大小将会是关于 $f^n(B)$ 的多项式大小，便失去了其模拟的意义。

至此本节已经介绍了放大器的概念，其是一种新的帮助测 0 的手段，而在后面将介绍 Czerwiński、Lasota、Leroux 等人是如何通过这种新方法来构造出有效模

拟测 $F_3(n)$-有界测 0 计数器乃至 $F_n(n)$-有界测 0 计数器的手段并最终提升了向量加法系统的可达性下界。

6.3.2 非初等下界的证明

本节将介绍可达性问题的非初等下界证明[74]。由前一节证明可知，如果有一个常数大小的 $(B, f(B))$-放大器，其中 $f(x) = \Omega(2^x)$，那么根据定理6.5，就可以获得一个 $F_3(n)$-生成器从而获得关于可达性问题是非初等难的下界结论。

接下来将介绍 Czerwiński 等人在 [74] 给出的 $(k, k!)$-放大器的构造。为了后面叙述方便先介绍几个新命令，令 $x, \hat{x}$ 是一个计数器，i, $\hat{i}$ 是 B-有界测 0 计数器，定义如下的三条命令：

- $x- = i$ 表示将计数器 x 里的值减少计数器 i 里的值，在执行这条命令时假设 $v(\hat{i}) = 0$，可用下面这段命令来表示：

  Loop;
  $i--,\ \hat{i}++;$
  test? i
  Loop;
  $x--,\ \hat{i}--,\ i++;$
  test? $\hat{i}$;

- $x+ = i + 1$ 与 $x- = i$ 相似，表示将计数器 x 里的值加上计数器 i 里的值以后再额外加 1，同样在执行这条命令时假设 $v(\hat{i}) = 0$，可用下面这段命令来表示：

  $x++;$
  Loop;
  $i--,\ \hat{i}++;$
  test? i
  Loop;
  $x++,\ \hat{i}--,\ i++;$
  test? $\hat{i}$;

- **Loop at most** x **times** < body > 这条命令指的是给循环次数设置了个最大次数，即计数器 x 里存的值，执行这条命令时假设 $v(\hat{x}) = 0$，可以用下面这段命令表示：

  Loop;
  $x--,\ \hat{x}++;$

Loop;

$x++,\ \hat{x}--,\ x++$;

< body >;

Czerwiński 的构造关键在于如何做乘法，比如可以用六个计数器 $x, \hat{x}, y, z, i, \hat{i}$，实现一个无测 0 计数器和一个 B-有界测 0 计数器的乘法，其中 $i, \hat{i}$ 是一个 B-有界测 0 计数器，则要令 x 里存的值变成原来的 $v(i)$ 倍可用如下一段程序 $\mathbb{P}_{mult_1}$ 实现，这里假设初始除了 x 与 i 外其余的计数器的值都为 0：

计数程序 6–12 $\mathbb{P}_{mult_1}$

```
1 z+ = i;
2 Loop;
3     x − −, x̂+ = i;
4 Loop;
5     y + +, x + +, x̂ − −;
6 Loop;
7     y− = i, z − −;
8 halt if v(z) = 0;
```

令 x 初始的值为 v_x、i 初始的值为 v_i，则对于任何一个运行到第六行命令的格局 $\mathbf{m} = (\mathbf{u}, 6)$，有 $\mathbf{u}[y] \le \mathbf{u}[x]$, $\mathbf{u}[x] \le v_x \cdot v_i$, $\mathbf{z} = v_i$。从而最终 $v(z) = 0$ 当且仅当此时 $\mathbf{u}[x] = \mathbf{u}[y] = v_x \cdot v_i$，即完成了两个计数器之间的一个乘法。但是在这段程序中一个乘法的成功计算是通过一次对不能测 0 的计数器进行测 0 即在最终的 'halt' 命令测 0 所保证的，因此如果做多个乘法的话需要用多个不同的计数器在最终测 0，会使其大小变得与乘法次数相关。

为了减少在这里的开销，Czerwiński 提出了一种新的关于一个无测 0 计数器和一个 B-有界测 0 计数器的乘法，其只关注于对 $\times B$ 的操作，其核心思想是 $\times B$ 可以由如下 $B-1$ 个乘法复合而得：

$$\frac{2}{1} \cdot \frac{3}{2} \cdots \frac{B}{B-1} = B. \tag{6–6}$$

考虑计数程序 $\mathbb{P}_{mult_2}$。与 $\mathbb{P}_{mult_1}$ 类似，只有 $i, \hat{i}$ 是 B-有界测 0 计数器，令初始的时候 x, y, z 这三个计数器里的值均为 v_0，这里 v_0 是一个 $B!$ 的倍数，而其余计数器里的值都为 0。可以看到其会存在一个到达终止格局的运行，必须在运行到格局 $\mathbf{m} = (\mathbf{u}, 9)$ 的时候满足 $\mathbf{u}[y] = \mathbf{u}[x] = v_0 \cdot B$，而对于任何一个格局 $\mathbf{m}_1 = (\mathbf{u}_1, 3)$，其第一次运行到形如 $\mathbf{m}_2 = (\mathbf{u}_2, 7)$ 这样的格局时一定满足 $\mathbf{u}_2[x] \le \mathbf{u}_1[x] \cdot \frac{\mathbf{u}_2[i]}{\mathbf{u}_2[i]-1}$。因

计数程序 6–13 $\mathbb{P}_{mult_2}$

```
1  i++;
2  Loop;
3     Loop;
4        x−=i, y−=i, x̂+=i;
5     Loop;
6        x̂−−, x++, y++;
7     i++;
8  max i?;
9  Loop;
10    y−=i, z−−;
11 halt if v(z)=0;
```

此由6–6，该程序能有一个到达终止格局的运行当且仅当之前每次循环次数都做满了。

$\mathbb{P}_{mult_2}$ 是能带来重大启发的，我们可以据此构造出一个倍数差距 $k!$ 的一组数，这里 $k!$ 就是正常的 $k\cdot(k-1)\cdots 1$ 的阶乘定义。具体来说可以假设一开始三个计数器的值全是 a，然后令第一个作上述的乘法最后变成值为 $a\cdot k$，第二个则不断的做除法变成 $\frac{a}{(k-1)!}$，最后只需要验证第三个计数器和第一个计数器里的值是 k 倍的关系，就获取了一组倍数差距为 $k!$ 的值 $(a\cdot k, \frac{a}{(k-1)!})$，而这只需要两个 k-有界测 0 计数器即可。据此构造了程序 $\mathbb{P}_{factorial}$，其只用到了两个 k-有界测 0 计数器 $i,\hat{i}$，其余则都是无测 0 计数器，下述引理说明了 $\mathbb{P}_{factorial}$ 的正确性。

引理 6.4 (Czerwiński[74]) $\mathbb{P}_{factorial}$ 是一个 $(k,k!)$-放大器。

证明 考察 $\mathbb{P}_{factorial}$ 的任何一个成功的运行，令 $\mathbf{m}=(\mathbf{u},4)$ 表示运行中第一次执行第 4 行命令的格局，$\mathbf{n}=(\mathbf{v},19)$ 表示运行中第一次执行第 19 行命令的格局，我们说明这两个格局必须满足如下条件：

- $\mathbf{u}[c]=\mathbf{u}[d]=\mathbf{u}[x]=\mathbf{y}=a$, $\mathbf{u}[b]=\mathbf{u}[i]=1$，这里 a 是某个自然数。
- $\mathbf{v}[v]=\mathbf{v}[x]=a\cdot k$, $\mathbf{v}[y]=a$, $\mathbf{v}[i]=k$, $\mathbf{v}[b]=k!$, $\mathbf{v}[c]=\frac{a}{(k-1)!}$。

关于 $\mathbf{m}$ 是比较容易说明的，因为程序可以运行第二行的循环任意多次，接下来说明，在格局 $\mathbf{m}$ 确定后，格局 $\mathbf{n}$ 里的寄存器的值必须满足上述条件。$\mathbf{v}[x]=\mathbf{v}[d]=k\cdot a$ 是显然的，否则其不能通过最后的 'halt' 命令，下面说明 b,c 的值也如上所示。

注意到第 18 行对于 i 做的 '?max' 测试因此第 4 行开始到第 17 行的大循环一共会精确的执行 $k-1$ 次，令第 i 次开始执行这个大循环的格局为 $\mathbf{m}_i=(\mathbf{u}_i,4)$，显然 $\mathbf{m}_1=\mathbf{m}$。接下来证明对于任何一个 $j\in[k-2]$，$\mathbf{u}_j$ 与 $\mathbf{u}_{j+1}$ 有如下的关系：

计数程序 6–14 $\mathbb{P}_{factorial}$

```
1  i++, b++, c++, d++, x++, y++;
2  Loop;
3     c++, d++, x++, y++;
4  Loop;
5     Loop;
6        c−=i, ĉ+=1;
7           Loop at most b times;
8              d−=i, x−=i, d̂+=i+1;
9     Loop;
10       b−−, b̂+=i+1;
11    Loop;
12       b̂−−, b++;
13    Loop;
14       ĉ−−, c++;
15       Loop at most b times;
16             d̂−−, d++, x++;
17    i++;
18 max i?;
19 Loop;
20    x−=i, y−−;
21 halt if v(y)=0;
```

1. $\mathbf{u}_j[d] = \mathbf{u}_j[x]$.
2. $\mathbf{u}_{j+1}[i] = \mathbf{u}_j[i] + 1$.
3. $\mathbf{u}_{j+1}[d] + \mathbf{u}_{j+1}[\hat{d}] \le \frac{j+1}{j} \cdot (\mathbf{u}_j[d] + \mathbf{u}_j[\hat{d}])$.
4. $\mathbf{u}_{j+1}[c] + \mathbf{u}_{j+1}[\hat{c}] \ge \frac{1}{j} \cdot (\mathbf{u}_j[c] + \mathbf{u}_j[\hat{c}])$.
5. $\mathbf{u}_{j+1}[b] + \mathbf{u}_{j+1}[\hat{b}] \le (j+1) \cdot (\mathbf{u}_j[b] + \mathbf{u}_j[\hat{b}])$.

第 1, 2 点是显然的，而 3, 4, 5 的不等号也容易判断，以第 4 点为例，只要注意到每次会更改 $v(c)+v(\hat{c})$ 的值的命令只有在第 6 行将 c 里的值减去当前 i 里的值并且将 $\hat{c}$ 里的值加 1，因此每次循环过后其和是必然大于原来的 $\frac{1}{v(i)}$ 的，$v(b)+v(\hat{b})$ 与 $v(d)+v(\hat{d})$ 同理。

更需要关注的是等号的成立，下面说明第三个等号成立必须满足下述条件：

- 对于任何的 $l \in [j]$ 有 $\mathbf{u}_l[\hat{b}] = \mathbf{u}_l[\hat{c}] = \mathbf{u}_l[\hat{d}] = 0$。
- 对于任何的 $l \in [j]$ 有 $\mathbf{u}_l[d] = \mathbf{u}_l[b] \cdot \mathbf{u}_l[c]$。

固定 $j \in [k]$，考察第 8 行命令的执行次数，显然其最多能执行 $\min\{\frac{\mathbf{u}_j[d]}{j}, \frac{\mathbf{u}_j[c]}{j} \cdot \mathbf{u}_j[b]\}$，因此仅有在 $\mathbf{u}_j[d] = \mathbf{u}_j[b] \cdot \mathbf{u}_j[c]$ 的情况下，在执行完第 5 行至第 8 行的循环后，$v(d) + v(\hat{d})$ 的值会变成原来的 $\frac{j+1}{j}$ 倍，并且为了确保循环次数必须有 $\mathbf{u}_j[\hat{b}] = \mathbf{u}_j[\hat{c}] = \mathbf{u}_j[\hat{d}] = 0$。因此关于 c, b 的两个关系式的等号也会成立。

最后为了满足格局 n 中 $\mathbf{v}[d] = a \cdot k$ 必须有：

$$\mathbf{v}[d] = \frac{k}{k-1} \cdot \mathbf{u}_{k-1}[d] = \frac{k}{k-1} \cdot \frac{k-1}{k-2} \cdot \mathbf{u}_{k-2}[d] = \cdots = \frac{k}{k-1} \cdots \frac{2}{1} \cdot \mathbf{u}[d] = k \cdot a$$

因此 b, c 的值也都必须满足：

- $\mathbf{v}[b] = k \cdot \mathbf{u}_{k-1}[b] = \cdots = k!$.
- $\mathbf{v}[c] = \frac{1}{k-1} \cdot \mathbf{u}_{k-1}[c] = \cdots = \frac{a}{(k-1)!}$.

即 a 必须是 $(k-1)!$ 的倍数，并且 $\mathbb{P}_{factorial}$ 是一个 $(k, k!)$-放大器，引理得证。□

注意到 $f(k) = k! > 2^k$，因此有 $f^n(n) > 2^{\overset{2}{\cdots}\}n}$，所以由引理6.4和定理6.5可以得出关于向量加法系统可达性问题的下界结论。

定理 6.6 (Czerwiński[74]) 向量加法系统的可达性问题是 $\mathbf{F}_3$-难的。

该方法也可以得出关于可达性问题以维度为参数的参数复杂性结论，只需要注意到 $f^k(n) > 2^{\overset{2^n}{\cdots}\}k}$，因此由定理6.5有如下结论：

定理 6.7 (Czerwiński[74]) 对于任意的 $h \in \mathbb{N}$，$(h+13)$ 维带状态的向量加法系统的可达性问题是 h-**EXPSPACE**-难的。

证明 可以使用如下的 $\mathbb{P}_h = \mathbb{P}_3 \triangleright \underbrace{\mathbb{P}_{factorial} \triangleright \cdot \triangleright \mathbb{P}_{factorial}}_{h}$ 的放大器来完成证明，并且将其简化讲一些可以复用的计数器合并为一个即可获得结论。□

这里我们不给出定理6.7的具体证明，因为在下一节中可以看到通过进一步的优化获得更好的参数复杂性下界，而参数复杂性下界只是由定理6.5衍生出来的直接推论。

6.3.3 进一步优化

本节将完成对可达性问题 **Ackermann**-完备的介绍。随着放大器概念的提出，可达性问题的下界被进一步提升，Czerwiński，Lasota，Leroux 也分别在 [80-82] 提出了对该方法的进一步优化，本节将介绍这部分的结论。

6.3.3.1　$6k$ 维向量加法系统的 $\mathbf{F}_k$ 下界证明

这一节将介绍 Czerwiński 在 [80] 带来的改进。可以看到放大器的提出，实际上是将一些模拟测 0 的操作如果没有真实的模拟则将其产生的副作用到最后去进行惩罚，换句话说，其可以理解成将若干次测 0 合并成一次测 0，但这个合并测 0 的操作是有前提的，即需要测 0 的那个计数器其界都是一致的，不能说有一个 B_1-有界测 0 的计数器和一个 B_2-有界测 0 的计数器，我们用放大器的方法将对其两个计数器的测 0 操作合并起来用一次测 0 来保证。

而 Czerwiński 的改进，就是基于此，他提出在满足一定的条件，即预先知道需要多少次测 0 的情况下，可以将界不同的计数器一起来模拟测 0。为了描述清楚此想法，下面来用个例子做介绍：

计数程序 6–15 $\mathbb{P}_{2dim}$

```
1 x++;
2 Loop;
3   Loop;
4     x−−, y++;
5   Loop;
6     x+=2, y−−;
7   i++;
8 max i?;
9 halt;
```

例 6.6　考察程序 $\mathbb{P}_{2dim}$，这里令 i 是一个 B-有界测 0 计数器，其余都是无测 0 计数器，事实上对于该程序的一个成功运行，如果确保中间两个循环次数都能做满的话，最后会得到 x 里存储的值为 2^B，但这需要在执行完第 3 行和第 5 行两个循环之后分别对 x 和 y 进行测 0。这显然是用放大器的方法无法做到的，因为 x, y 一直在变大。

但另一方面，为了确保最后 x 的值是 2^B，只需要分别对 x, y 各进行 B 次测 0 即可。下述描述了一种可以通过一次测 0 就可以保证 x, y 这 B 次需要测 0 时都为 0 的方法。为了描述清楚先仅考虑 x。在 $\mathbb{P}_{2dim}$ 的一次成功运行中，一共会运行到第 5 行 B 次，令这 B 个格局分别为 $\mathbf{m}_1 = (\mathbf{u}_1, 5), \ldots, \mathbf{m}_B = (\mathbf{u}_B, 5)$，由上述讨论可知，其最终格局 $\mathbf{m} = (\mathbf{u}, \#)$ 要满足 $\mathbf{u}[x] = 2^B$ 则必须有对 $j \in [B]$ 都有 $\mathbf{u}_j[x] = 0$。

注意到对于任意 $j \in [B]$ 有 $\mathbf{u}_j[x] \geq 0$，因此上述条件满足当且仅当 $\sum_{j=1}^{B} \mathbf{u}_j[x] = 0$。仔细观察发现，上述和可以如此重新计算：将在运行到格局 $\mathbf{m}_1$ 时

对 x 存储值的变化记为 v_1，对于格局 $\mathbf{m}_j$ 运行到 $\mathbf{m}_{j+1}$ 这一过程对 x 的存储值的变化记为 v_{j+1}，则有如下关系：

$$\sum_{j=1}^{B} \mathbf{u}_j[x] = B \cdot v_1 + (B-1) \cdot v_2 + \cdots + v_B \tag{6–7}$$

如果能用一个计数器 c 来储存右边这个值，那么在最后就能通过对 c 测 0 来保证在 $\mathbf{m}_j,\ j \in [B]$ 这 B 个格局时 x 存储的值均为 0。该想法可以用下述程序 $\mathbb{P}_{2dim'}$ 来表示：

计数程序 6–16 $\mathbb{P}_{2dim'}$

```
1 x++, c+=B;
2 Loop;
3   Loop;
4     x--, y++, c-=B+1-i;
5   Loop;
6     x+=2, y--, c+=2(B-i);
7   i++;
8 max i?;
9 halt if v(c)=0;
```

这里先不关注 $+=B,\ -=B+1-i,\ +=2(B-i)$ 是如何实现的，但是如果这些命令可以实现，那么依据上述的讨论，就可以通过最后 $v(c)=0$ 来确保第 3 行至第 4 行的循环每次结束时 $v(x)=0$。特别的，可以用同样的方法保证 y 的测 0，并且令其使用一个计数器，即如下程序 $\mathbb{P}_{2dim''}$：

计数程序 6–17 $\mathbb{P}_{2dim''}$

```
1 x++, c+=B;
2 Loop;
3   Loop;
4     x--, y++, c-=B+1-i, c+=B+1-i;
5   Loop;
6     x+=2, y--, c+=2(B-i), c-=B+1-i;
7   i++;
8 max i?;
9 halt if v(c)=0;
```

则 $\mathbb{P}_{2dim''}$ 的一个成功的运行必然满足其终止格局 $\mathbf{m}=(\mathbf{u},\#)$ 有 $\mathbf{u}[x]=2^B$。

接下来形式化的来介绍这个方法[80]。首先补充一些定义。给定一个计数程序 $\mathbb{M}$ 和其的一个运行 ρ，令其对应的格局为 $\mathbf{m}_1 = (\mathbf{u}_1, 1), \ldots, \mathbf{m}_n$，定义 $\delta_\rho[i,j]$ 表示从格局 $\mathbf{m}_i$ 运行到 $\mathbf{m}_j$ 计数器存储值的变化量，即 $\delta_\rho[i,j] = \mathbf{u}_j - \mathbf{u}_i$。下面的引理说明了上述测 0 方法的正确性。

引理 6.5 (Czerwiński[80])　给定一个计数程序 $\mathbb{M}$ 和其的一个运行 ρ，令其对应的格局为 $\mathbf{m}_1 = (\mathbf{u}_1, 1), \ldots, \mathbf{m}_n$。令 $c, c_1, \ldots c_k$ 是 $\mathbb{M}$ 中的 $k+1$ 个计数器，$S_i \subseteq [n]$, $i \in [k]$ 是 k 个下标集合，并且对于每个 S_i，定义函数 $N_i(j) = |\{x | x > j,\ x \in S_i\}|$，如果下述条件成立：

- $\mathbf{u}_1[c] = \sum_{i=1}^{k} |S_i| \cdot \mathbf{u}_1[c_i]$。
- 对于任意的 $i, j \in [n]$ 有 $\delta_\rho[i,j][c] = \sum_{i=1}^{k} \sum_{h=i}^{j-1} N_i(h) \cdot \delta_\rho[h, h+1][c_i]$。
- $\mathbf{u}_n[c] = 0$。

则对任意的 $i \in [k] j \in S_i$ 有 $\mathbf{u}_j[c_i] = 0$。

证明　只要证明下列事实：

$$\mathbf{u}_n[c] = \sum_{i=1}^{k} \sum_{j \in S_i} \mathbf{u}_j[c_i]. \tag{6–8}$$

则由于对于任意的 $j \in [n], i \in [k]$ 有 $\mathbf{u}_j[c_i] \geq 0$，因此引理成立。事实上注意到对 $i \in [k]$ 和 $l \in [n]$ 有:

$$\mathbf{u}_l[c_i] = \mathbf{u}_1[c_i] + \sum_{h=1}^{l-1} \delta[h, h+1][c_i]. \tag{6–9}$$

□

因此重新计算 $\mathbf{u}_n[c]$ 可得：

$$\begin{aligned}
\mathbf{u}_n[c] &= \mathbf{u}_1[c] + \delta[1, n-1][c] \\
&= \sum_{i=1}^{k} |S_i| \cdot \mathbf{u}_1[c_i] + \sum_{i=1}^{k} \sum_{h=1}^{n-1} N_i(h) \cdot \delta_\rho[h, h+1][c_i] \\
&= \sum_{i=1}^{k} (N_i(0) \cdot \mathbf{u}_1[c_i] + \sum_{h=1}^{n-1} N_i(h) \delta_\rho[h, h+1][c_i]) \\
&= \sum_{i=1}^{k} \sum_{j \in S_i} (\mathbf{u}_1[c_i] + \sum_{h=1}^{j-1} \delta_\rho[h, h+1][c_j]) \\
&= \sum_{i=1}^{k} \sum_{j \in S_i} \mathbf{u}_j[c_i].
\end{aligned}$$

接下来证明可达性问题是 **Ackermann**-完备的。令函数 $G_1, G_2, \ldots$ 定义如下：

- $G_1(n) = 2n$.
- $G_k(n) = \underbrace{G_{k-1} \circ \cdots \circ G_{k-1}(1)}_{n}$。

G_i 与 F_i 的定义类似，比如 $G_2(n) = 2^n$，$G_3(n) = 2^{\cdot^{\cdot^{\cdot^{2}}}\}n}$，$G_n(n)$ 也是著名的 **Ackermann** 函数等。同样有如下的结论：

定理 6.8 (schmitz[86]) $G_\alpha(|M|)$-有界的计数程序 M 的停机问题是 $\mathbf{F}_\alpha$ 完备的。

计数程序 6–18 $\mathbb{G}_{init}[b,c,d]$

```
1 b+ = n;
2 d+ = n, c + +;
3 Loop;
4     d+ = n, c + +;
```

下面证明，可以使用引理6.5来构造一个 G_k-放大器。

引理 6.6 (Czerwiński[80]) 给定 k，$n \in \mathbb{N}$ 可以在 $O(k+n)$ 时间内构造一个 $G_k(n)$-生成器。

证明 首先回顾例子6.4中的程序 $\mathbb{P}_3$，我们可以构造一个同样的程序 $\mathbb{G}_{init}[b,c,d]$，使其的成功运行满足 $v(b) = n, v(c) = a, v(d) = n \cdot a$。这里需要注意的是 $\mathbb{G}_{init}$ 的长度是 $O(n)$ 的，因为 $'+ = n'$ 命令实际上是由 n 条 $'+1'$ 命令组成。因此在后续的证明的中可以假定有一组计数器初始输入是 $(n, a, n \cdot a)$，其中 n 是一个给定的自然数，而 a 则是任意的一个自然数。

接下来归纳证明这个结论，$k = 1$ 的时候是显然的，程序 $\mathbb{G}_1$ 便是一个 $G_1(n)$-生成器。计数器 (b,c,d) 将会输出 $(G_1(n), t, G_1(n)\cdot t)$，这里 $t, a \in \mathbb{N}$ 满足 t 可以取遍所有的自然数。为了叙述方便将 $\mathbb{G}_1$ 记为 $\mathbb{G}_1(b_0,c_0,d_0)[b,c,d]$ 表示将 (b_0,c_0,d_0) 理解为一组初始为 $(n, a, n\cdot a)$ 的输入，而 $[b,c,d]$ 理解为对应的输出 $(G_1(n), t, G_1(n)\cdot t)$。

假设 $\leq k-1$ 的情况成立，考察 k 的情况。由归纳假设存在一个 $6k-6$ 维的 $G_{k-1}(n)$-生成器，考虑程序 $\mathbb{G}_k$，其使用如下 $6k$ 个计数器：

- 一组初始输入为 $(n, x, n \cdot a)$ 的计数器 (i, i_1, i_2)。
- 两个额外辅助的计数器 a_1, a_2 和一个控制测 0 的计数器 C_k。
- $\mathbb{G}_{k-1}$ 的 $6k-6$ 个计数器，并令 (b_0,c_0,d_0) 为输入计数器，b,c,d 为输出计数器。

计数程序 6–19 $\mathbb{G}_1$

```
1  G_init[b_0, c_0, d_0];
2  Loop;
3      Loop;
4          c_0 − −, c + +, d_0 − −, d+ = 2;
5      Loop;
6          c_0 + +, c − −, d_0 − −, d+ = 2;
7  Loop;
8      c_0 − −, c + +, d_0− = 2, d+ = 4;
9  b_0 − −, b+ = 4;
10 halt if v(d_0) = 0;
```

计数程序 6–20 $\mathbb{G}_k$

```
1  b_0 + +, C_k+ = i;
2  Loop;
3      c_0 + +, d_0 + +, C_k+ = i, C_k+ = i;
4  Loop;
5      G^i_{k−1}(b_0, c_0, d_0)[b, c, d];
6      Loop;
7          b − −, b_0 + +, C_k − −;
8      Loop;
9          c − −, c_0 + +, C_k − −;
10     Loop;
11         d − −, d_0 + +, C_k − −;
12     i − −, a_2 + +;
13 halt if v(i), v(C_k) = 0;
```

我们维护 $v(i)+v(a_1)+v(a_2)=n$，显然初始满足条件。下面来对 $\mathbb{G}_k$ 中出现的一些命令作一些解释。

- 首先介绍对 i,a_1 的测 0 命令。注意到 a_1,i 的取值都在 $[0,n]$ 之间，所以与上一节相同，其实是在用 (n,x,nx) 的三元组来进行测 0，**test**? i 如下所示，**test**? a_1 则是将其中的 i 与 a 交换即可。

  $i_1-=2$;
  Loop;
  $a_1--,\ i++,\ i_2--$;
  Loop;
  $a_2--,\ a_1++,\ i_2--$;
  Loop;
  $a_1--,\ a_2++,\ i_2--$;
  Loop;
  $i--,\ a_1++,\ i_2--$;

  这里需要注意的是，因为程序是维持 $v(i)+v(a_1)+v(a_2)=n$，因此对 i 测 0 是相当于把 a_1 里的值放进 i 里，再将 a_2 里的值放进 a_1 里，这一操作能执行 n 次 -1 操作当且仅当 $v(i)=0$。完成以后还需要复位，因此还要反着再执行一次。
- $+=i$ 与 $-=i$ 的命令与上一节几乎相同，只不过需要将 $\hat{i}$ 换成 a_1 即可。
- $\mathbb{G}_{k-1}^{i}(b_0,c_0,d_0)[b,c,d]$ 表示将 $\mathbb{G}_{k-1}$ 做如下改动：
  - 去除其最后的 'halt' 命令。
  - 对于 $\mathbb{G}_{k-1}$ 中每一个需要最后测 0 的计数器和输出计算器 x，将其中的每一个 $x++$ 都多加一行命令 $C_k+=i$，对于每一个 $x--$ 我们都多加一行命令 $C_k-=i$，注意到这里的计数器包括 b_0,c_0,d_0,b,c,d 以及一些其他需要测 0 的计数器。

接下来说明 $\mathbb{G}_k$ 是一个 $G_k(n)$-生成器。首先说明程序可以在最终计数器 b,c,d 上输出 $(G_k(n),t,G_k(n)\cdot t)$。先忽略 C_k 的命令，事实上，第 4 行至第 12 行的循环一共可以执行 n 次，可以构造如下的运行：

1. 在第 i 次循环开始时，假设 b_0,c_0,d_0 存储的值是 $(G_{k-1}^{i-1}(1),t_{i-1},G_{k-1}^{i-1}(1)\cdot t_{i-1})$，由归纳假设在执行完第五行以后，$b_0,c_0,d_0$ 里的值变为 0，b,c,d 的值则变为 $(G_{k-1}^{i}(1),t_i,G_{k-1}^{i}(1)\cdot t_i)$。
2. 执行完 6 到 10 行的命令后，程序将 b,c,d 里的值全部转移至 b_0,c_0,d_0，即在第 i 次循环后 b_0,c_0,d_0 的值为 $(G_{k-1}^{i}(1),t_i,G_{k-1}^{i}(1)\cdot t_i)$，而 b,c,d 的值全

为 0。

3. 在第 n 次循环结束第 5 行的运行后直接跳过后面的循环，直接进入最后一行的 'halt' 命令。

可以看到这种运行最后是满足要求的。接下来说明任何一个成功的运行都满足最终 b, c, d 输出 $(G_k(n), t, G_k(n) \cdot t)$。考察上述运行需要有的保证：

- 令 $\mathbb{G}_{k-1}$ 最后需要测 0 的非输入计数器的计数器集合为 C，在每次执行完 $\mathbb{G}_{k-1}^i(b_0, c_0, d_0)[b, c, d]$ 时，C 中的计数器需要是 0。
- 在每次执行完 $\mathbb{G}_{k-1}^i(b_0, c_0, d_0)[b, c, d]$ 的时候 b_0, c_0, d_0 的值需要是 0。
- 在每次执行完第 4 行至第 12 行的大循环的时候 b, c, d 里的值需要为 0。

事实上 b, c, d, b_0, c_0, d_0 以及 C 里的计数器在整个运行中恰好都需要进行 n 次测 0，因此可以使用引理6.5来构造一个一起测 0 的计数器 C_k。令上述计数器在进行其第 j 次测 0 前发生 $'+1'$ 或者 $'-1'$ 时，C_k 相应的发生 $'+(n+1-j)'$ 或者 $'-(n+1-j)'$ 次操作，而这一个值是被计数器 i 维护的，因此只要对 C_k 做 $+=i$ 或者 $-=i$ 操作便可满足。接下来说明 G_k 是满足上述条件的：

- 在第 1 行和第 3 行中，因为 b_0, c_0, d_0 都还没有测 0，因此其计数器里的值每加 1，C_k 的值都需要加 n，而此时 i 里的值为 n。
- 在第 j 次执行第 4 行至第 12 行的大循环的 $\mathbb{G}_{k-1}^i(b_0, c_0, d_0)[b, c, d]$ 时候，此时 $v(i) = n+1-j$，需要测 0 的计数器都执行了 $j-1$ 次测 0，因此每次当对这些计数器进行 $'+1'$ 或者 $'-1'$ 操作时 C_k 需要对应的加上或者减去 $n+1-j$，即 $'+=i'$ 或者 $'-=i'$。
- 在第 j 次执行第 4 行至第 12 行的大循环的第 6 行至第 11 行时，由于此时 b_0, c_0, d_0 已经进行完 j 次测 0，因此其每次做 $'+1'$ 操作 C_k 需要增加 $n-j$，而 b, c, d 还只完成了 $j-1$ 次测 0，因此其每次完成 $'-1'$ 操作时 C_k 需要减少 $n-j+1$，综合起来，每次 C_k 需要 $'-1'$ 来完成目标。

综上，由上述讨论以及引理6.5，b, c, d, b_0, c_0, d_0 以及 C 里的计数器能够完成这 n 次测 0 当且仅当在执行完第 n 次循环的 $\mathbb{G}_{k-1}^i(b_0, c_0, d_0)[b, c, d]$ 时 C_k 的值为 0，也当且仅当最后 $v(i) = v(C_k) = 0$。因此对于任何一个 $\mathbb{G}_k$ 的成功运行，其在 b, c, d 上的输出为 $(G_{k-1}^n(1), t, G_{k-1}^n(1) \cdot t)$，即 $(G_k(n), t, G_k(n) \cdot t)$，且 t 的取值可以是任意的，引理得证。 □

因此由定理6.5和引理6.6可以直接得到可达性问题的下界结果。

定理 6.9 (Czerwiński[80])　向量加法系统的可达性问题是 **Ackermann**-难的，特别的令 $k \geq 3$，则 $6k$ 维带状态的向量加法系统的可达性问题是 $\mathbf{F}_k$-难的。

6.3.3.2 3k+2 维向量加法系统的 $\mathbf{F}_k$ 下界证明

本节将介绍 Lasota 在 [81] 上的关于可达性问题的下界结果，其在同样得到 **Ackermann**-难的下界的基础上，对固定维的下界结果进行了进一步的提升。

Lasota 对于测 0 方法的改进关键在于对于计数器 (b, c, d) 的不同运用，在之前的章节中 b 里所存储的值 B 视作的是有界测 0 计数器的上界，但在 [81] 中 Lasota 将其视作为测 0 次数的上界，具体来说注意到如下的结果，一个测 0 计数器如果没有上界限制，则称为无界测 0 计数器：

定理 6.10 令 $k \geq 3$，给定一个无界测 0 计数程序 $\mathbb{M} = (C, T, P)$，其中 $|T| = 2$，则判定其是否存在一个测 0 次数不超过 $G_k(|\mathbb{M}|)$ 的成功的运行是 $\mathbf{F}_k$-难的。

这里函数 G_k 的定义在上一节。令 $\mathbb{N}_4$ 表示所有能被 4 整除的自然数，为了本节叙述的方便，定义一个与 $\{G_k\}$ 类似的一族函数 $\{H_k\}$，H_k 是一个从 $\mathbb{N}_4$ 映射到 $\mathbb{N}_4$ 的函数，其定义如下所示：

$$H_1(n) = 2n,\ H_{i+1}(n) = \underbrace{H_i \cdot H_i \cdot \cdots \cdot H_i}_{\frac{n}{4}}. \tag{6–10}$$

不难验证，对任意的 $i, n \in \mathbb{N}$，$H_i(4 \cdot n) = 4 \cdot G_i(n)$，因此有：

定理 6.11 给定一个无界测 0 计数程序 $\mathbb{M} = (C, T, P)$，其中 $|T| = 2$，则判定其是否存在一个测 0 次数不超过 $H_k(|\mathbb{M}|)$ 的成功的运行是 $\mathbf{F}_k$-难的。

接下来说明，如果存在一组组计数器 (b, c, d) 满足 $v(b) = 2m$，$v(d) = 2m \cdot v(c)$，这里 m 是一个给定的值，$v(c)$ 可以取任意多的值，则可以使用这三个计数器去代替只能测 0 至多 m 次的无界测 0 计数程序中的测 0 操作，即用一个无测 0 计数程序来模拟该无界测 0 计数程序。

引理 6.7 给定一个无界测 0 计数程序 $\mathbb{P}$，其实用到两个能测 0 的计数器 x, y，存在一个无测 0 计数程序 $\mathbb{P}^*$，其用到额外的三个计数器 b, c, d，$\mathbb{P}$ 存在一个使用刚好进行 m 次测 0 的成功运行，当且仅当 $\mathbb{P}^*$ 存在一个成功的运行，并且存在 $x \in \mathbb{N}$，使得 b, c, d 的初始输入满足：

$$v(b) = 2m,\ v(c) = x,\ v(d) = 2m \cdot B,\ m, B \in \mathbb{N} \tag{6–11}$$

证明 证明将给出 $\mathbb{P}^*$ 的构造，其核心想法是维持 $v(c) + v(x) + v(y)$ 的值不变，即由于开始 x, y 里的值为 0，因此让其和保持初始在 c 里的值，不妨令其值为 B。$\mathbb{P}^*$ 相比 $\mathbb{P}$ 做如下改动：

- 对于 x, y 的每一次 $'+1'$ 或者 $'-1'$ 操作，对 c 进行相应的操作来保持 $v(x)+v(y)+v(c)$ 不变，即：
  - $'x++'$ 变为 $'x++,\ c--'$；$'x--'$ 变为 $'x--,\ c++'$。
  - $'y++'$ 变为 $'y++,\ c--'$；$'y--'$ 变为 $'y--,\ c++'$。
- 将 $\mathbb{P}$ 中每一个对 x 的测 0 操作 **test** x? 改写为如下程序 zero$[x]$：

  Loop;
  $y--,\ x++,\ d--$;
  Loop;
  $c--,\ y++,\ d--$;
  Loop;
  $c++,\ y--,\ d--$;
  Loop;
  $y++,\ x--,\ d--$;
  $b-=2$;

  对 $\mathbb{P}$ 中每一个对 y 的测 0 操作也改写成类似上述的程序 zero$[y]$，仅仅需要将其中的 x 和 y 对换即可。
- 在 $\mathbb{P}$ 最后的 'halt' 命令中添加要求检测 $v(d)$ 是否为 0 的命令。

接下来先说明在 zero$[x]$ 这段程序里，b, c, d 能保持 $v(d)=v(c)\cdot v(b)$ 乘积的关系当且仅当其真的模拟了一次成功的测 0。给定一个格局 $\mathbf{m}_0=(\mathbf{u}_0, n_0)$，其通过 zero$[x]$ 的格局为 $\mathbf{m}_1=(\mathbf{u}_1, n_1)$，则有:

$$\mathbf{u}_1[d]-\mathbf{u}_0[d]\leq 2\cdot B \tag{6–12}$$

事实上，由于 $v(c)+v(x)+v(y)=B$ 是 $\mathbb{P}^*$ 一直在保持的性质，因此在 zero$[x]$ 的第一第二个循环中 d 的值最多被减少 $\mathbf{u}_0[y]+\mathbf{u}_0[c]\leq B$ 次，在第三第四个循环中 d 的值最多被减少 $\mathbf{u}_1[y]+\mathbf{u}_1[c]\leq B$ 次，且两个等号成立当且仅当 $\mathbf{u}_0[c]=\mathbf{u}_1[c]=0$ 并且满足 B 大于等于计数器 x, y 在运行中出现的任何值，因此不等式6–12成立。对于 zero$[y]$ 也是同理。

接下来证明引理6.7成立。事实上由不等式6–12可知用 zero$[x]$ 或 zero$[y]$ 对 x, y 进行一次成功的测 0 当且仅当 d 的值被减少了 $2B$。因此 $\mathbb{P}$ 有一个 m 次测 0 的成功运行 ρ，当且仅当 $\mathbb{P}^*$ 存在一个成功的运行并且满足 b, c, d 的初始值为 $v(b)=2m,\ v(c)=B,\ v(d)=2m\cdot B$，其中 B 是一个大于在 ρ 中 x, y 里出现过的任何值的数。 □

我们可以用这个性质来证明向量加法系统的下界。简单来说如果存在一个

$H_k(n)$-生成器，那么就可以用向量加法系统来模拟一个不超过 $H_k(n)$ 次测 0 计数程序的运行。Lasota 在 [81] 证明了如下定理。

定理 6.12 (Lasota[81]) 给定一个 m_k 维的 $H_k(n)$-生成器，则 m_k+2 维带状态的向量加法系统可以模拟一个不超过 $\frac{H_k(n)}{2}$ 次测 0 的无界测 0 计数程序。特别的，m_k+2 维带状态的向量加法系统的可达性问题是 $\mathbf{F}_k$-难的。

证明 令 $\mathbb{M}$ 是一个 $H_k(n)$-生成器，其输出的计数器为 b, c, d；$\mathbb{P}$ 是一个不超过 $\frac{H_k(n)}{2}$ 次测 0 的无界测 0 计数程序，并且其使用的会测 0 的计数器为 x, y，构造如下程序：$\mathbb{S} = \mathbb{M}'; \mathbb{P}^*; \mathbb{L}$:

- $\mathbb{M}'$ 将最后一句 'halt' 删除即可。
- $\mathbb{P}^*$ 的构造如同引理6.7，并且也需要将最后一句 'halt' 删除。
- $\mathbb{L}$ 定义如下，这里 $v(\mathbb{M})$ 表示 $\mathbb{M}$ 运行到最后在最后的 'halt' 命令时需要测 0 的计数器，$v(\mathbb{P})$ 同理：

  Loop;
  $x--,\ c++$;
  Loop;
  $zero[x]$;
  halt if $v(\mathbb{M}), v(\mathbb{P}), d = 0$;

  $\mathbb{P}$ 有一个 n_1 次测 0 的成功运行，其中 $n_1 \le \frac{H_k(n)}{2}$，当且仅当 $\mathbb{P}^*$ 模拟了 n_1 次测 0 运行，$\mathbb{L}$ 模拟进行了 $\frac{H_k(n)}{2} - n_1$ 次测 0 运行，由引理6.7$\mathbb{P}$ 有一个不超过 $\frac{H_k(n)}{2}$ 次测 0 的运行当且仅当 $\mathbb{S}$ 有一个成功的运行，且 $\mathbb{S}$ 用到的计数器数量为 m_k+2，引理得证。 □

注 如果 $\mathbb{M}$ 里的计数器由冗余的话，还可以用其中的计数器来代替 x, y，这样结论可以进一步被改进到 m_k 维的向量加法系统的可达性问题是 $\mathbf{F}_k$-难的。

最后只需要证明存在一个 $3k+2$ 维的 $H_k(n)$-生成器即可，其同样也用到了上述提到的测 0 方法。

定理 6.13 (Lasota[81]) 给定 k 和 n，可以在 $O(n+k)$ 时间内构造一个 $H_k(n)$-生成器，其用到了 $3k+2$ 个计数器。

证明 证明对 k 作归纳给出 $H_k(n)$-生成器的构造，并且其用到 $3k+3$ 个计数器。$k=1$ 时，$H_1(n) = 2n$，在引理6.6中的 $\mathbb{G}_1$ 即符合要求，并且我们令其中的 b, c, d

计数器重命名为 b_1, c_1, d_1。注意到 $H_k(n)$ 的定义，下面只考虑在 $\mathbb{N}_4$ 上的放大，即假设 n 是可以被 4 整除的。

假设已经存在了一个 $H_{k-1}(n)$-生成器 $\mathbb{H}_{k-1}$，接下来构造 $\mathbb{H}_k$。由归纳假设 $\mathbb{H}_{k-1}$ 可以视作一个有一组输入计数器 b_0, c_0, d_0 的计数程序，其输出计数器为 $b_{k-1}, c_{k-1}, d_{k-1}$，$\mathbb{H}_k$ 在此基础上多使用 3 个计数器 b_k, c_k, d_k。首先介绍如下在 $\mathbb{H}_k$ 要用到的一些指令：

- 构造会使用 zero$[x]$ 这些子程序，该定义与引理6.7的证明里所定义的相同，只是其中的计数器 b, c, d 被重命名为了 b_k, c_k, d_k。
- 构造使用一个子程序 $\mathbb{L}$，其可以视作将输入计数器 $b_{k-1}, c_{k-1}, d_{k-1}$ 里储存的值转移到输出计数器 b_0, c_0, d_0 里。不难验证，令格局 $\mathbf{m} = (\mathbf{u}, 1)$ 满足 $\mathbf{u} = [B, x, Bx, 0, 0, 0]$，则 $\mathbb{L}$ 从其出发的一个成功的运行满足最终格局 $\mathbf{m}' = (\mathbf{u}', \#)$ 有 $\mathbf{u}' = [0, 0, 0, B, x, Bx]$。

计数程序 6–21 $\mathbb{L}$

```
1  Loop;
2    Loop;
3      c_{k-1} − −, c_0 + +, d_{k-1} − −, d_0 + +;
4    Loop;
5      c_{k-1} + +, c_0 − −, d_{k-1} − −, d_0 + +;
6    b_{k-1} − = 2, b_0 + = 2;
7  Loop;
8    c_{k-1} − −, c_0 + +, d_{k-1} − −, d_0 + +;
9  b_{k-1} − = 2, b_0 + = 2;
10 halt if v(d_{k-1}) = 0;
```

- 构造还将使用 SetZero$[c_k]$ 子程序，其定义如下：

  Loop;
  $c_k − −$, $d_k − = 4$;
  $zero[c_k]$;
  $zero[c_k]$;

- 对于一个计数程序 $\mathbb{P}$，用 $\overline{\mathbb{P}}$ 表示经如下修改的程序：
  - 对每个 d_0, d_{k-1} 做加减的操作，添加上对 c_k 做减加的命令，以此来保证 $v(c_k) + v(d_0) + v(d_{k-1})$ 的值不变。
  - 去除此最后的 'halt' 命令。

接下来给出 $\mathbb{H}_k$ 的定义。$\mathbb{H}_k$ 的定义如下：

计数程序 6–22 $\mathbb{H}_k$

```
1  b_k+ = 4(n+1), d_k+ = 4(n+1), c_k ++;
2  Loop;
3      d_k+ = 4(n+1), c_k ++;
4  b_0+ = 8, d_0+ = 8, c_0 ++, c_k− = 8;
5  Loop;
6      d_0+ = 8, c_0 ++, c_k− = 8;
7  Loop;
8      ℙ̄;
9      zero[d_0];
10     𝕃̄;
11     zero[d_{k−1}];
12 SetZero[c_k];
13 halt if v(d_k) = 0;
```

先说明 $\mathbb{H}_k$ 可以最终在计数器 b_0, c_0, d_0 上输出 $(H_k(n), x, H_k(n)\cdot x)$。注意到每次执行 zero$[d_0]$ 或者 zero$[d_{k-1}]$ 计数器 b_k 的值都会减少 2，执行 SetZero$[c_k]$ 时 b_k 的值也会减少 4，因此第 7 行开始到第 11 行的大循环至多执行 n 次。假设存在一次执行了 n 次的运行，并且每次执行 zero$[d_0]$ 与 zero$[d_{k-1}]$ 时 d_0, d_{k-1} 的值都分别为 0，令 $\mathbf{u}_i$ 表示第 i 次开始执行第 7 行命令时计数器存储的值，特别的令 $\mathbf{u}_{n+1}$ 表示第 n 次循环结束后计数器的值，则由归纳假设有：

- $\mathbf{u}_1[b_0] = 8,\ \mathbf{u}_1[d_0] = 8\cdot\mathbf{u}_1[c_0]$.
- 对于 $j \geq 2$，我们有 $\mathbf{u}_j[b_0] = H_{k-1}(\mathbf{u}_{j-1}[b_0]),\ \mathbf{u}_j[d_0] = \mathbf{u}_j[b_0]\cdot\mathbf{u}_j[c_0]$。

注意到 $H_k(n) = \underbrace{H_{k-1}\circ\cdots\circ H_{k-1}(8)}_{\frac{n}{4}-1}$，因此令 $m = 4(n+1)$，由归纳假设有：

$$\mathbf{u}_{n+1}[b_0] = H_{k-1}^n(8) = H_k(m),\ \mathbf{u}_{n+1}[d_0] = H_k(m)\cdot\mathbf{u}_{n+1}[c_0] \tag{6–13}$$

然后说明，$\mathbb{H}_k$ 的任何一次成功运行当且仅当满足 b_0, c_0, d_0 最后输出 $(H_k(n), x, H_k(n)\cdot x)$。事实上只要注意到，每次在执行 zero$[x]$ 的命令时，由引理6.7b_k, c_k, d_k 的变化量 $\Delta(b_k), \Delta(c_k), \Delta(d_k)$ 满足：

$$\Delta(b_k) = 2,\ \Delta(d_k) \leq 2\cdot\Delta(c_k) \tag{6–14}$$

因此令执行到第四行时 c_k 里的值为 B，则最终 d_k 的值为 0 当且仅当执行到第 12 行命令时其计数器的值 $\mathbf{u}$ 满足：

$$\mathbf{u}_n[b] = 4,\ \mathbf{u}_n[d] = 4\cdot\mathbf{u}_n[c] \tag{6–15}$$

因此由关系6–14，等式6–15能满足当且仅当之前每一次模拟测 0 都如预想般执行，即 $\mathbb{H}_k$ 是一个 $H_k(n)$-生成器。显然 H_k 是在 $O(n+k)$ 时间可以完成的，最后注意到计数器 b_k 的大小最多为 $4n+4$，因此可以将其编码至程序里而省去一个计数器，所以我们在 $O(n+k)$ 的时间内构造出了一个用 $3k+2$ 个计数器的 $H_k(n)$-生成器，引理得证。 □

6.3.3.3　2k+4 维向量加法系统的 $\mathbf{F}_k$ 下界证明

本节将介绍 Leroux 在 [82] 提出的对向量加法系统可达性问题的下界改进，其通过一种不同于 Czerwiński 与 Lasota 对于 [74] 中对 Ackermann 函数的计算方法以及对测 0 方法的改进，获得了目前已知的最好以维数为参数的参数复杂性下界。

首先回顾 $F_k(x)$ 的定义，有 $F_0(x) = x+1$, $F_k(x) = F_{k-1}^{x+1}(x)$，并且 Ackermann 函数 $A(n) = F_\omega(n)$。由此可以定义一个在向量上的函数 $F_{\mathbf{v}}$，令 $\mathbf{v}$ 是一个 $\mathbb{N}$ 上的 d 维向量，则 $F_{\mathbf{v}}$ 定义为：

$$F_{\mathbf{v}}(n) = F_d^{\mathbf{v}[d]} \circ F_{d-1}^{\mathbf{v}[d-1]} \circ \cdots \circ F_1^{\mathbf{v}[1]}(n) \tag{6–16}$$

我们使用一些记号，令 $\mathbf{k}_{d,i}$ 表示一个仅在第 i 维为 k 其余为 0 的 d 维向量，则显然有 $\mathbf{F}_{\mathbf{1}_{d,d}}(n) = F_d(n)$, $F_{(\mathbf{n+1})_{d,d}}(n) = F_{d+1}(n)$。接下来介绍计算 Ackermann 函数的一个新方法，令 $\mathrm{Eval}_d(\mathbf{u}, n) : \mathbb{N}^d \times \mathbb{N} \to \mathbb{N}^d \times \mathbb{N}$ 是一个 $d+1$ 维上的函数，$p \stackrel{\text{def}}{=} \min\{i \mid \mathbf{u}[i] > 0\}$，则 Eval_d 定义如下：

$$\mathrm{Eval}_d(\mathbf{u}, n) \stackrel{\text{def}}{=} \begin{cases} (\mathbf{u} - \mathbf{1}_{d,p}, 2n+1), & \text{if } p = 1, \\ (\mathbf{u} - \mathbf{1}_{d,p} + (\mathbf{n+1})_{d,p-1}, n), & \text{if } p > 1 \end{cases}$$

显然对于确定的 $\mathbf{u}$ 和 n，Eval_d 的迭代次数是有限的，记其迭代次数最多的函数为 $E(\mathbf{u}, n)$，即：$E(\mathbf{u}, n) = \mathrm{Eval}_d^{max}(\mathbf{u}, n)$。在 [158] 中关于 E 函数有如下结论：

引理 6.8 ([158])　给定 $\mathbf{u} \in \mathbb{N}^d$ 和 $n \in \mathbb{N}$，有：

$$E(\mathbf{u}, n) = (\mathbf{0}_d, F_{\mathbf{u}}(n))$$

特别的有 $E((\mathbf{n+1})_d, n) = (\mathbf{0}_d, F_{d+1}(n))$。

有了 E 函数，便可以通过只做乘法和加法的方式计算出 $A(n)$ 的值，即对一个 n 维向量和一个值 n 不停的调用 Eval 函数。为了进一步减少模拟测 0 的计数器使用，Leroux 在 [82] 提出了一种 K-预生成器 (K-preamplifier)，其是对定义6.2的一种扩展。

定义 6.3 (Leroux[82]) 一个 K-预生成器是一个无测 0 计数程序 $\mathbb{M} = (C, P)$，考察其中的 3 个计数器 b, c, d，其满足：

- 对于一个成功运行的最终格局 $\mathbf{m} = (\mathbf{u}, \#)$ 都满足对于计数器 $i \in C \setminus \{b, c, d\}$ 有 $\mathbf{u}[i] = 0$，并且有 $\mathbf{u}[d] \geq \mathbf{u}[b] \cdot \mathbf{u}[c]$，等号成立当且仅当 $\mathbf{u}[b] = K$。
- 对于任意的 $l \in \mathbb{N}$ 都存在一个成功运行的最终格局 $\mathbf{m} = (\mathbf{u}, \#)$ 满足：$\mathbf{u}[c] = l$, $\mathbf{u}[d] = \mathbf{u}[b] \cdot \mathbf{u}[c]$。

例 6.7 计数程序 $\mathbb{Q}_1$ 就是一个非常简单的 R-预生成器。

计数程序 6–23 $\mathbb{Q}_1$

```
1 b+ = R;
2 d+ = R, c++;
3 Loop;
4     d+ = R, c++;
5 Loop;
6     b−−;
7 halt;
```

给定一个 K-预生成器 $\mathbb{Q}$，可以通过添加 4 个计数器的方式生成一个 K-生成器 $\mathbb{Q}'$。简单来说，令 b_1, c_1, d_1 这 3 个额外的计数器在 $\mathbb{Q}$ 赋值 b, c, d 的操作，使其保持 $v(b) = v(b_1)$, $v(c) = v(c_1)$, $v(d) = v(d_1)$，再令一个额外的计数器 $\hat{b}_1$ 保持 $v(\hat{b}_1) = 0$。最后在 $\mathbb{Q}$ 的最后添加如下命令：

Loop;
    $b_1 - -$, $\hat{b_1}++$, $d_1 - -$;
$c_1 - -$;
Loop;
    b_1++, $\hat{b_1} - -$, $d_1 - -$;
$c_1 - -$;
halt if $v(d_1) = 0$;

不难发现最后 d_1 里的值为 0 当且仅当进入这段命令时的格局 $\mathbf{m} = (\mathbf{u}, \_)$ 满足：$\mathbf{u}[d_1] = \mathbf{u}[b_1] \cdot \mathbf{u}[c_1]$，即 $\mathbf{u}[d] = \mathbf{u}[b] \cdot \mathbf{u}[c]$，从而由定义6.3我们有 $\mathbf{u}[b] = K$，因此 $\mathbb{Q}'$ 是一个 K-生成器。

但 Leroux 发现仅仅需要一个 K-预生成器 $\mathbb{Q}$，便能获得与生成器几乎同样的效果。如果一个计数程序在任何时候所有计数器里的值加起来都不超过 K，则称

其是 K-限界的。类似于引理6.3和定理6.5，对于一个 K-限界的计数程序 $\mathbb{P}$，如果存在一个 K-预生成器 $\mathbb{Q}$，我们则可以构造一个无测 0 计数程序 $\mathbb{Q} \trianglerighteq \mathbb{P}$ 来模拟 $\mathbb{P}$ 的行为。

$\mathbb{Q} \trianglerighteq \mathbb{P}$ 的构造是这样的。假设 $\mathbb{Q}$ 和 $\mathbb{P}$ 使用的计数器名字是互不相同的，否则可以重命名。令 $\mathbf{P}$ 中需要测 0 的计数器为 $b_1, \ldots b_n$，$\mathbb{Q}$ 的输出计数器为 b, c, d，$\mathbb{Q} \trianglerighteq \mathbb{P}$ 的核心想法就是令 $v(b) + v(b_1) + \cdots + v(b_n)$ 一直保持不变，即一直为 K。则类似于引理6.7，对 $B = \{b_1, \ldots b_n\}$ 的测 0 命令用如下程序 $\mathrm{zero}_B[b_j]$ 来模拟，令需要测 0 的计数器为 b_j，将这些计数器和计数器 b 重新编号为 $b_{i_0}, \ldots b_{i_n}$，并且令 $b_{i_0} = b_j,\ b_{i_1} = b$：

For **k** = **1** to **n**;
  Loop;
    $b_{i_k}--,\ b_{i_{k-1}}++,\ d--$;
For **k** = **n** to **1**;
  Loop;
    $b_{i_{k-1}}--,\ b_{i_k}++,\ d--$;
$c-=2$;

这里 For **k** = **1** to **n** < body > 是一种简写，表示将 < body > 的命令按 $k = 1$ 到 n 按顺序写 n 遍。其能模拟测 0 的核心想法是一样的，在每次执行过程中 d 的值最多减少 $2K$，并且刚好能减少 $2K$ 当且仅当被测 0 的计数器 b_j 开始和结束的值都为 0，所以只要维护 $v(d) = K \cdot v(c)$ 就可以在最后对 c, d 测 0 来确保模拟测 0 的正确性。因此 $\mathbb{T} = \mathbb{Q} \trianglerighteq \mathbb{P} = \mathbb{Q}'; \mathbb{P}'; \mathbb{S}$ 定义如下：

- $\mathbb{Q}'$ 减少了最后一条终止命令。
- $\mathbb{P}'$ 将 $\mathbb{P}$ 中每一个对 $b_j \in B$ 的测 0 命令都换成 $\mathrm{zero}_B[b_j]$，并且对于 b_j++ 或者 b_j--，都相应添加 $b--$ 或者 $b++$，并且也删除最后一条终止命令。
- $\mathbb{S}$ 定义如下：

  Loop;
    $b--$;
  halt if $v(\mathbb{Q}),\ v(\mathbb{P}),\ v(b),\ v(c),\ v(d) = 0$;

与之前的定理相同，K-限界计数程序 $\mathbb{P}$ 有一个成功的运行当且仅当 $\mathbb{T}$ 有一个成功的运行，而 $\mathbb{T}$ 的大小是 $O(n(|\mathbb{P}| + |\mathbb{Q}|))$ 的，因此有如下定理：

定理 6.14 (Leroux[82])　给定一个 K-预生成器 $\mathbb{Q}$，对于任何一个 K-限界计数程序 $\mathbb{P}$ 可以构造一个大小为 $O(n(|\mathbb{P}| + |\mathbb{Q}|))$ 的向量加法系统来模拟，特别的如果 $\mathbb{Q}$ 的大小是在 $\mathcal{F}_{<3}$ 的，则 VAS 的可达性问题是 **Ackermann**-难的。

有了定理6.14，接下来只需要构造一个有效的 $A(n)$-预生成器。由引理6.8，首先需要构造一个能够计算 Eval_k 的无测 0 计数程序，Leroux 在 [82] 使用了 $2k+6$ 个计数器来计算 Eval_k，接下来介绍这个程序。首先介绍计数程序 $\mathbb{U}_{k,p}$，其目的是

计数程序 6–24 $\mathbb{U}_{k,p}$

```
1  y_p − −, y_{p−1} + +;
2  Loop;
3     c_1 − −, ... c_{k+1} − −, d + +;
4  Loop at most c times;
5     d − −, c_p + +, c_1 + +, ... c_{k+1} + +;
6  Loop at most y_0 times;
7     y_{p−1} + +;
8     Loop at most c times;
9        ĉ − −;
10       Loop at most b times;
11          d − −, c_p + = 2, c_{p+1} + +, ..., c_{k+1} + +;
12 updateB_{k,p};
```

为了对于一个特定的 $p \in [k]$ 来计算 Eval_k，则 $\mathbb{U}_k$ 可以通过一个非确定的选择调用 $\mathbb{U}_{k,p}$ 来完成对 Eval_k 的运算。

$\mathbf{U}_{k,p}$ 使用 $2k+4$ 个计数器，分别为 $c, \hat{c}, c_1, \ldots c_{k+1}, d, b, \hat{b}, y_0, \hat{y}_0, y_1, \ldots, y_k$。在程序 $\mathbb{U}_{k,p}$ 中还使用了一些特别的命令，介绍如下：

- **Loop at most** c **times**< body > 该命令与章节6.3.2中相同，简单来说就是令 < body > 内的指令至多执行 $v(c)+v(\hat{c})$ 次，再次回顾下其具体定义：

  Loop;
  $c - -,\ \hat{c} + +;$
  Loop;
  $c + +,\ \hat{c} - -,$ < body >;

- **updateB**$_{\mathbf{k,p}}$ 是一段子程序，当 $p > 1$ 的时候其为空，而当 $p = 1$ 的时候，其定义如下：

  $y_0 + +;$
  Loop at most y_0 **times**;
  $\hat{y_0} - -,\ y_0 + +;$
  Loop at most b **times**;
  $b + +;$

y_0 − −;

关于 $\mathbf{updateB_{k,1}}$ 有如下性质。

引理 6.9　令 $\mathbf{updateB_{k,1}}$ 的一个运行是从格局 $\mathbf{m}_1 = (\mathbf{u}_1, i_1)$ 至格局 $\mathbf{m}_2 = (\mathbf{u}_2, i_2)$，则有：

- $\mathbf{u}_1[y_0] + \mathbf{u}_1[\hat{y_0}] = \mathbf{u}_2[y_0] + \mathbf{u}_2[\hat{y_0}]$。
- $\mathbf{u}_2[b] + \mathbf{u}_2[\hat{b}] \leq 2^{\frac{\mathbf{u}_1[y_0]+1}{2}}(\mathbf{u}_1[b] + \mathbf{u}_1[\hat{b}])$，并且当 $\mathbf{u}_1[y_0] + \mathbf{u}_1[\hat{y_0}]$ 是奇数时等号可以取到。

证明　注意到在 $\mathbf{updateB_{k,1}}$ 中第 2 行至第 5 行对 $y_0, \hat{y_0}$ 的操作都是互补的，即每当对 y_0 做加减操作时，会立即对 $\hat{y_0}$ 做相反的操作，因此 $v(y_0) + v(\hat{y_0})$ 的值能保持不变，第一点成立。

对于第二点只需要注意，第 4 行到第 5 行的 **Loop at most** b **times**< b + + > 命令执行一次最多将 b 里存储的值变成原来的两倍，另一方面第 2 行的循环里由于每次 $\hat{y_0}$ 多减了一次，因此最多执行 $\frac{\mathbf{u}_1[y_0]+\mathbf{u}_1[\hat{y_0}]+1}{2}$ 次，从而：

$$\mathbf{u}_2[b] + \mathbf{u}_2[\hat{b}] \leq 2^{\frac{\mathbf{u}_1[y_0]+1}{2}}(\mathbf{u}_1[b] + \mathbf{u}_1[\hat{b}]) \tag{6–17}$$

等号成立当且仅当 $\mathbf{u}_1[y_0] + \mathbf{u}_1[\hat{y_0}] + 1$ 能被 2 整除，即 $\mathbf{u}_1[y_0] + \mathbf{u}_1[\hat{y_0}]$ 是奇数，并且有上述讨论有 $\mathbf{u}_2[\hat{y_0}] = \mathbf{u}_2[\hat{b}] = 0$，其余计数器的值保持不变。　□

将 $y_0, \ldots y_k$ 视作一个输入输出的编码，即令其编码 $(\mathbf{y}, n) \in \mathbb{N}^k \cdot \mathbb{N}$，其中 $\mathbf{y} = [v(y_1), \ldots, v(y_k)]$，$v(y_0) = n$。对于一个格局 $\mathbf{m} = (\mathbf{u}, i)$，令其编码 $\mathbf{f}[\mathbf{m}] = (\mathbf{u}[y_1, \ldots, y_k], \mathbf{u}[y_0])$。

称一个格局 $\mathbf{m} = (\mathbf{u}, i)$ 是好的，如果其满足下列条件：

- $\mathbf{u}[d] = \mathbf{u}[\hat{c}] = \mathbf{u}[\hat{b}] = \mathbf{u}[\hat{y_0}] = 0$。
- $\mathbf{u}[c] > 0$, $\mathbf{u}[b] = 2^{\mathbf{u}[y_0]}$, $\mathbf{u}[c_1] = 2^{\mathbf{u}[y_0]} \cdot \mathbf{u}[c]$。
- 对于 $j \in [k]$，有 $\mathbf{u}[c_{j+1}] = 2^{\mathbf{u}[y_j]} \cdot \mathbf{u}[c_j]$。

而对于 $j \geq 2$，称一个格局 $\mathbf{m} = (\mathbf{u}, i)$ 是 j-坏的，如果其满足：

$$\mathbf{u}[c] + \mathbf{u}[\hat{c}] > 0,\ \mathbf{u}[c_j] + \mathbf{u}[d] > 2^{\mathbf{u}[y_{j-1}]}(\mathbf{u}[c_{j-1}] + \mathbf{u}[d])$$

并且对于 $l > j$ 有 $\mathbf{u}[c_l] + \mathbf{u}[d] = 2^{\mathbf{u}[y_{l-1}]}(\mathbf{u}[c_{l-1}] + \mathbf{u}[d])$。如果格局满足 $\mathbf{u}[\hat{c}] = \mathbf{u}[d] = 0$, $\mathbf{u}[b] + \mathbf{u}[\hat{b}] < 2^{\mathbf{u}[y_0]+\mathbf{u}[\hat{y_0}]}$, $\mathbf{u}[c_1] = 2^{\mathbf{u}[y_0]+\mathbf{u}[\hat{y_0}]}\mathbf{u}[c]$ 并且对于 $l > 1$ 有 $\mathbf{u}[c_l] = 2^{\mathbf{u}[y_{l-1}]} \cdot \mathbf{u}[c_{l-1}]$ 则称该格局是 1-坏的。为了方便叙述，称一个好的格局是 0-坏的。下面说明对于 $\mathbb{U}_{k,p}$，如果其存在由一个 j_1-坏的格局到 j_2-坏的格局，则必须会有 $j_2 \geq j_1$，即好的格局只能从好的格局出发运行到，并且一旦变成了 j-坏的格局 $(j > 0)$，则不可能再到达一个好的格局。

引理 6.10 (Leroux[82]) 令 $\mathbf{m}_1 = (\mathbf{u}_1, i_1)$ 是一个 j_1-坏的格局，如果 $\mathbb{U}_{k,p}$ 存在一个由 $\mathbf{m}_1$ 到 $\mathbf{m}_2$ 的运行，则 $\mathbf{m}_2 = (\mathbf{u}_2, i_2)$ 是一个 j_2-坏的格局并且 $j_2 \geq j_1$，特别的，如果 $\mathbf{m}_1$ 是一个好的格局，并且满足 $\mathbf{u}_1[c]$ 是 $\mathbf{u}_1[b]$ 的倍数并且 p 是最小的下标满足 $\mathbf{u}_1[y_p] > 0$，则存在一个运行使其终止格局 $\mathbf{m}_3 = (\mathbf{u}_3, i_3)$ 也是一个好的格局，并且满足：

- $\mathbf{u}_3[c_{k+1}] = \mathbf{u}_1[c_{k+1}]$。
- 令 $\mathbf{m}_1, \mathbf{m}_3$ 两个格局的编码分别为 f, g，则有 $g = \mathrm{Eval}_{k,p}(f)$。

证明 先来证明第二点。第二点是容易的，令 $\mathbf{m}_1 = (\mathbf{u}_1, i_1)$ 是一个好的格局，下面构造一个运行，使其最后的格局 $\mathbf{m}_3 = (\mathbf{u}_3, i_3)$ 满足要求。事实上只需要给出各个循环的执行次数即可，该运行按如下的方式进行：

- 第 2 行开始的循环执行 $\mathbf{u}_1[c_1]$ 次。
- 第 4 行开始的循环执行 $\mathbf{u}_1[c]$ 次。
- 第 6 行开始的循环执行满，一共 C 次。
- 第 8 行开始的循环每次都执行满 $\mathbf{u}_1[c]$ 次，记总共执行的次数为 C_1 次。
- 第 10 行开始的循环也每次都执行满 $\mathbf{u}_1[b]$ 次，记总共执行的次数为 C_2 次。

先考虑 $p > 1$ 的情况，则 $C = \mathbf{u}_1[y_0]$。注意到因为 m 是一个好的格局，所以 $\mathbf{u}_1[c_1] = 2^{\mathbf{u}_1[y_0]} \cdot \mathbf{u}_1[c]$，因此第 8 行一共执行的次数为：

$$C_1 = \sum_{j=1}^{C} \frac{\mathbf{u}_1[c]}{2^j} = \sum_{j=1}^{C} 2^{-j} \cdot \mathbf{u}_1[c] = (1 - 2^{-C}))\mathbf{u}_1[c] \tag{6–18}$$

因此第 10 行一共执行的次数为：

$$C_2 = C_1 \cdot \mathbf{u}_1[b] = (2^C - 1)\mathbf{u}_1[c] \tag{6–19}$$

这些次数是可以执行完的，因为在执行到第 6 行开始时，计数器里 d 的值为：

$$v(d) = \mathbf{u}_1[c_1] - \mathbf{u}[c] = (2^C - 1)\mathbf{u}[c] \tag{6–20}$$

因此第 11 行里我们可以对 d 做 C_2 次减法，即第 10 行开始的循环可以做 C_2 次。所以对最终格局 $\mathbf{m}_3$ 有：

- $\mathbf{u}_3[\hat{b}] = \mathbf{u}_3[\hat{c}] = \mathbf{u}_3[\hat{y_0}] = 0$, $\mathbf{u}_3[d] = 2^C\mathbf{u}_1[c] - \mathbf{u}_1[c] - C_2 = 0$。
- 对于 y_i 的值的变化情况有：

$$\mathbf{u}_3[y_j] = \begin{cases} \mathbf{u}_1[y_j], & \text{if } j \neq p-1, p, \\ \mathbf{u}_1[y_j] + \mathbf{u}_1[y_0] + 1, & \text{if } j = p, \\ \mathbf{u}_1[y_j] - 1, & \text{if } j > p \end{cases}$$

从而对于 $\mathbf{u}_1, \mathbf{u}_3$ 的两个编码 f, g 有 $g = \mathrm{Eval}_{k,p}(f)$。

- $\mathbf{u}_3[c] = \frac{\mathbf{u}_1[c]}{2^{\mathbf{u}_1[y_0]}} = \frac{\mathbf{u}_1[c]}{\mathbf{u}_1[b]}$。
- 对于 c_j 的值的变化情况有：

$$\mathbf{u}_3[c_j] = \begin{cases} \mathbf{u}_1[c_j] - \mathbf{u}_1[c_1] + \mathbf{u}_1[c], & \text{if } j < p, \\ \mathbf{u}_1[c_j] - \mathbf{u}_1[c_1] + 2(\mathbf{u}_1[c] + C_2), & \text{if } j = p, \\ \mathbf{u}_1[c_j] - \mathbf{u}_1[c_1] + \mathbf{u}_1[c] + C_2, & \text{if } j > p \end{cases}$$

从而对于 $j \in [k]$ 有：

$$\mathbf{u}_3[c_{j+1}] = 2^{\mathbf{u}_3[y_j]} \cdot \mathbf{u}_3[c_j] \tag{6–21}$$

即 $\mathbf{m}_3$ 依旧是一个好的格局。$p = 1$ 时类似，因为并不改变循环做的次数，并且由引理6.9不难同样验证 $\mathbf{m}_3$ 是一个好的格局，并且编码了一次 $\mathrm{Eval}_{k,p}$ 的运算。

然后来证明第一点。先考虑 $j_1 \geq 2$ 的情况，令 $\mathbf{m}_1$ 是一个 j_1-坏的格局，则有：

- $\mathbf{u}_1[c] + \mathbf{u}_1[\hat{c}] > 0$。
- 对于 $l > j_1$ 有 $\mathbf{u}_1[c_l] + \mathbf{u}_1[d] = 2^{\mathbf{u}_1[y_{l-1}]}(\mathbf{u}_1[c_{l-1}] + \mathbf{u}_1[d])$。
- 对于 j_1 有 $\mathbf{u}_1[c_{j_1}] + \mathbf{u}_1[d] > 2^{\mathbf{u}_1[y_{j_1-1}]}(\mathbf{u}_1[c_{j_1-1}] + \mathbf{u}_1[d])$。

证明的关键还是考虑其中循环的执行次数，令五个循环的执行次数分别为 L_1, L_2, L_3, L_4, L_5，则不难发现执行后的格局 $\mathbf{m}_2$ 满足：

- 对于 c_j 的值的变化来说有：

$$\mathbf{u}_2[c_j] = \begin{cases} \mathbf{u}_1[c_j] - L_1 + L_2, & \text{if } j < p, \\ \mathbf{u}_1[c_j] - L_1 + 2(L_2 + L_5), & \text{if } j = p, \\ \mathbf{u}_1[c_j] - L_1 + L_2 + L_5, & \text{if } j > p \end{cases}$$

- 对于 y_j 的值的变化来说有：

$$\mathbf{u}_2[y_j] = \begin{cases} \mathbf{u}_1[y_j], & \text{if } j \neq p-1, p, \\ \mathbf{u}_1[y_j] + L_3 + 1, & \text{if } j = p-1, \\ \mathbf{u}_1[y_j] - 1, & \text{if } j = p \end{cases}$$

- $\mathbf{u}_2[c] + \mathbf{u}_2[\hat{c}] = \mathbf{u}_1[c] + \mathbf{u}_1[\hat{c}] - L_4$。
- $\mathbf{u}_2[d] = \mathbf{u}_1[d] + L_1 - L_2 - L_5$。

下面观察 $\mathbf{u}_1[c_j] + \mathbf{u}_1[d]$ 和 $2^{\mathbf{u}_1[y_{j-1}]}(\mathbf{u}_1[c_{j-1}] + \mathbf{u}_1[d])$ 的变化。为了方便叙述，记第一个的变化值为 Δ_1，第二个的变化值为 Δ_2。对 j 做分类讨论：

- $j < p$，则有 $\Delta_1 = -L_5$, $\Delta_2 = -L_5 \cdot 2^{\mathbf{u}_1[y_{j-1}]}$。
- $j = p$，则有 $\Delta_1 = L_2 + L_5$, $\Delta_2 = 2^{\mathbf{u}_1[y_{p-1}]+L_3+1}(\mathbf{u}_1[c_{p-1}] + \mathbf{u}_1[d] - L_5) - 2^{\mathbf{u}_1[y_{p-1}]}(\mathbf{u}_1[c_{p-1}] + \mathbf{u}_1[d])$。
- $j = p+1$，则有 $\Delta_1 = 0$, $\Delta_2 = 2^{\mathbf{u}_1[y_p]-1}(L_2 + L_5 - \mathbf{u}_1[c_p] - \mathbf{u}_1[d])$。
- $j > p+1$，则有 $\Delta_1 = \Delta_2 = 0$。

因此只需要考虑 $j = p, p+1$ 的情况。$p+1$ 的情况是简单的，注意到 $\mathbf{u}_2[d] \geq 0$, $\mathbf{u}_1[c_p] \geq L_1$，因此有 $L_2 + L_5 \leq \mathbf{u}_1[d] + L_1 \leq \mathbf{u}_1[c_p] + \mathbf{u}_1[d]$，即此时 $\Delta_2 \leq 0$。

再来考虑 $j = p$ 的情况，只需要证明 $j_1 \leq p$ 的时候成立即可。注意到此时有 $\mathbf{u}_2[c_{p+1}] + \mathbf{u}_2[d] = 2^{\mathbf{u}_1[y_p]}(\mathbf{u}_1[p] + \mathbf{u}_1[d])$，因此有上述讨论有 $\mathbf{u}_1[c_p] = L_1$。但此时由定义有 $\mathbf{u}_1[c_{j_1}] < \mathbf{u}_1[c_p]$，因此:

$$\mathbf{u}_1[c_p] = L_1 = \min\{\mathbf{u}_1[c_j] | j \in [k+1]\} \leq \mathbf{u}_1[c_{j_1}] < \mathbf{u}_1[c_p] = L_1 \tag{6–22}$$

这是矛盾的，因此不可能发生。再来考虑 $\mathbf{m}_1$ 是 1-坏的情况，只需证明此时 $\mathbf{m}_2$ 不可能是一个好的格局。反设 $\mathbf{m}_2$ 是一个好的格局，注意到 $\mathbf{u}_1[b] + \mathbf{u}_1[\hat{b}] < 2^{\mathbf{u}_1[y_0]+\mathbf{u}_1[\hat{y_0}]}$，并且 $\mathbf{u_2}[b] = 2^{\mathbf{u}_2[y_0]}$，因此 $\mathbf{u}_2[\hat{y_0}] > 0$，这已经与 $\mathbf{m}_2$ 是好的格局相矛盾。

对于 $p = 1$ 的情况，由引理6.9和上述的讨论同理可证，因此结论成立，命题得证。 □

有了引理6.10，便可以获得一个 $F_d(n)$-预生成器，见如下定理。

定理 6.15 (Leroux[82]) 存在一个使用 $2k+4$ 个计数器，大小为 $O(nk4^n)$ 的 $2^{F_{d+1}(n)}$-预生成器。

证明 先来证明 $\mathbb{A}_{k,n}$ 是一个 $2^{F_{d+1}(n)}$-预生成器。首先说明存在一个从初始格局到一个格局 $\mathbf{m} = (\mathbf{u}, \#)$ 的成功运行满足 $\mathbf{u}[b] = 2^{F_{d+1}(n)}$, $\mathbf{u}[d] = 2^{F_{d+1}(n)} \cdot C$, $\mathbf{u}[c] = C$，其中 C 是一个大于 1 的自然数。构造运行如下:

- 第 3 行的循环一共执行 $2^{F_{d+1}(n)-2n-1} \cdot C - 1$ 次。
- 第 5 行的循环按引理6.10好的格局的运行执行尽可能多次。
- 后面两个循环能执行多少次就执行多少次。

令第五行的运行一共执行了 M 次，令第 j 次开始的格局为 $\mathbf{m}_{j-1} = (\mathbf{u}_j, 5)$，第一次运行到第 7 行的格局为 $\mathbf{m}_M$。注意到对于 $\mathbf{m}_0$ 来说有 $f[\mathbf{m}_0] = ((\mathbf{0}_{k-1}, n), n+1)$, $\mathbf{u}_1[c_{k+1}] = 2^{F_{d+1}(n)} \cdot C$ 次。则根据引理6.10对于 $j \in \{2, \ldots M\}$ 有 $f[\mathbf{m}_j] = \mathrm{Eval}_k(f[\mathbf{m}_{j-1}])$，因此由引理6.8我们有 $f[\mathbf{m}_M] = E(f[\mathbf{m}_1]) = E((\mathbf{n}+\mathbf{1})_{k,k}, n) = (\mathbf{0}_k, F_{d+1}(n))$。

计数程序 6–25 $\mathbb{A}_{k,n}$

1 $y_k+=n,\ y_0+=(n+1)$;
2 $c++,\ b+=2^{n+1},\ c_1+=2^n,\ldots,\ c_k+=2^n,\ c_{k+1}+=2^{2n+1}$;
3 **Loop**;
4 $\quad c++, c_1+=2^n,\ldots,\ c_k+=2^n,\ c_{k+1}+=2^{2n+1}$;
5 **Loop**;
6 $\quad \mathbb{U}_k$;
7 **Loop**;
8 $\quad d++,\ c_1--,\ldots,\ c_{k+1}--$;
9 **Loop**;
10 $\quad y_0--$;
11 **halt if** $v(c_{k+1}), v(\hat{b}), v(\hat{y_0}), v(y_0),\ldots v(y_k)=0$;

注意到 $\mathbf{m}_M$ 是一个好的格局，又因为 $\mathbf{u}_M[y_0]=F_{d+1}(n), \mathbf{u}_M[y_j]=0(j\geq 1)$，因此有 $\mathbf{u}_M[c_j]=\mathbf{u}_M[c_1]=2^{F_{d+1}(n)}\cdot C(j\in[k+1])$，并且有 $\mathbf{u}_M[b]=2^{F_{d+1}(n)}$。因此其最终格局 $\mathbf{m}$ 满足：

$$\mathbf{u}[b]=2^{F_{d+1}(n)},\ \mathbf{u}[c]=C>0,\ \mathbf{u}[d]=2^{F_{d+1}(n)}\cdot C \tag{6–23}$$

并且对于任何一个计数器 $i\notin\{b,c,d\}$ 有 $\mathbf{u}[i]=0$。

最后说明任何一个成功的运行其最后格局 $\mathbf{m}$ 都满足 $\mathbf{u}'[d]\geq\mathbf{u}'[b]\cdot\mathbf{u}[c]$，并且等号成立当且仅当 $\mathbf{u}'[b]=2^{F_{d+1}(n)}$。沿用上面的记号，考察 $\mathbf{m}_0$ 和 $\mathbf{m}_M$ 两个格局，由引理6.10，$\mathbf{m}_M$ 是个 i-坏的格局。下面分情况讨论。

如果 $\mathbf{m}_M$ 是个 i-坏的格局，并且 $i\geq 2$，则由定义有 $\mathbf{u}_M[c_i]+\mathbf{u}_M[d]>2^{\mathbf{u}_M[y_{i-1}]}(\mathbf{u}_M[c_{i-1}]+\mathbf{u}_M[d])$，并且有 $\mathbf{u}_M[c_j]+\mathbf{u}_M[d]=2^{\mathbf{u}_M[y_{j-1}]}(\mathbf{u}_M[c_{i-1}]+\mathbf{u}_M[d])$ 对于所有的 $j>i$，不难推出此时有 $\mathbf{u}_M[c_{k+1}]>\mathbf{u}_M[c_{i-1}]$，因此其不能通过第 7 行的循环将其全部重制为 0，即通不过最后的测 0 命令，矛盾。

如果 $\mathbf{m}_M$ 是一个 1-坏的格局，则有 $\mathbf{u}_M[\hat{b}]=\mathbf{u}_M[y_0]=\cdots=\mathbf{u}_M[y_k]=0$，并且 $\mathbf{u}_M[c_{k+1}]=\cdots=\mathbf{u}_M[c_1]$。因此最后的格局中满足 $\mathbf{u}'[d]=\mathbf{u}_M[c_{k+1}]$，即 $\mathbf{u}'[d]>\mathbf{u}'[c]\cdot\mathbf{u}'[b]$。

如果 $\mathbf{m}_M$ 是一个好的格局，由之前的讨论可知，在格局 $\mathbf{m}_M$ 中有 $\mathbf{u}_M[b]=2^{F_{d+1}}(n),\ \mathbf{u}_M[c_j]=\mathbf{u}_M[c_1]=2^{F_{d+1}(n)}\cdot C$，因此其最终格局同样也满足关系式6–23，从而 $\mathbb{A}_{k,n}$ 是一个 $2^{F_{d+1}(n)}$-预生成器。

最后注意到 $\mathbb{A}_{k,n}$ 使用了 $2k+6$ 个计数器，其大小是 $O(k\cdot 4^n)$，注意到其中的 $y_k\leq n$，因此可以用复制 n 遍 $\mathbb{A}_{k,n}$ 的方式将 y_k 编码进程序里，此外，注意到 $v(c_k)+v(d)$ 的值在运行过程中只会增加不会减少，并且对于一个好的格局可以将

c_k 的值通过其他的计数器来计算出来，因此最后可以构造一个只用 $2k+4$ 个计数器的 $2^{F_{d+1}(n)}$-预生成器 $\mathbb{A}'_{k,n}$，其大小为 $O(nk \cdot 4^n)$，定理得证。 □

由定理6.14和定理6.15，不难得出向量加法系统的可达性问题的下界。

定理 6.16 (Leroux[82]) 令 $k \geq 3$，$(2k+4)$ 维带状态的向量加法系统的可达性问题是 $\mathbf{F}_k$-难的。

6.4 本章小结

本章介绍了向量加法系统可达性问题下界的证明，下界的证明在沉寂了几十年之后在最近两三年取得了突飞猛进的成果，其关键核心还是使用了一组满足 $v(d) = v(c) \cdot v(b)$ 计数器 b, c, d 对一个不超过 $v(b)$ 的计数器进行测 0 的想法，这使得对可达性问题与可覆盖性问题和有界性问题的理解产生了本质上的区别，也验证了之前 Lipton 在 [89] 中提出的下界证明方法只能做到 **EXPSPACE**-难的下界，因为其方法同样对可覆盖性问题和有界性问题下界有效。

另一方面，结合在章节四中介绍的 Leroux 在 [79] 中的上界成果，本章介绍的下界证明[80-82] 已经证明了一般维的向量加法系统可达性问题是一个 **Ackermann**-完备的，这也说明了不仅一个向量加法系统的可达集可以有 **Ackermann** 那么大[142]，其可达性问题的判定也是相当难的。但是在固定维的向量加法系统的可达性问题上还是有很多疑问没有解决的。比如在下一章中将介绍 2 维带状态的向量加法系统的可达性问题是 **PSPACE**-完备的，但对于 3 维向量加法系统的可达性问题我们却没有丝毫的认识。而在现有的结论当中，目前只有 d 维向量加法系统的可达性问题是在 $F_{\frac{d-4}{2}}$ 到 F_{d+3} 之间的，这还是一个巨大的鸿沟，但是从下界的方法来分析，目前我认为还有两个可以思考的方向：

- 首先考虑到目前的测 0 方法相当于复杂度每从 F_d 升成 F_{d+1} 需要多用两个计数器来实现，一个很自然的想法是能不能将这个开销进一步缩减，即说明存在一个常数 M，使得 $d+M$ 维向量加法系统的可达性问题是 F_d-难的，这是目前提升该结果最为直接的方式，并且这也与目前的上界结果是最为匹配的一个结论。
- 向量加法系统有很多变种，比如带栈的，带一个可以测 0 的计数器的，又或是分支的向量加法系统，在这上面可以尝试是否可以运用新的测 0 技术对其的验证问题获取新的结论，比如关于带栈的向量加法系统，在 [74] 中的技术提出来之前，Lazic 在 [45] 通过对栈的一个简单运用就获得了比当时向量加法系统可达性问题下界更好的 F_3 下界，而现在显然已经是一个平凡的下界了。

第七章　固定维度下的可达性问题

本章将讨论固定维度下向量加法系统的可达性问题。对于固定维的可达问题，是否带状态变成了十分关键的因素，目前的研究成果基本集中于 1, 2 维上带状态的向量加法系统。Hopcroft 在 [40] 证明了 5 维以下的向量加法系统的可达集都是半线性的，从而证明了其是可判定的。当然注意到 2 维带状态的向量加法系统是 5 维向量加法系统的一个子集，从而也获得了其的可判定性。而随后在 [159] 里 Howell 等人证明了一些复杂性上的结论，他们给出 2 维带状态的向量加法系统的可达性问题是在 2-**EXPTIME** 内的，而下界结论则跟编码有关，在一进制的编码下该问题是 **NL**-难的，而在二进制的编码下则是 **NP**-难的。

2004 年 Leroux 在 [76] 证明了低维带状态的向量加法系统有一个很好的性质-平坦性 (flatness)。简单来说，即 2 维以下带状态的向量加法系统的路径都可以转换成没有嵌套圈的结构-线性路径策略 (linear path scheme)。随后低维的复杂性结果被进一步优化。Hasse 在 [75] 证明了 1 维带状态的可达性问题是 **NP**-完备的。Blondin 等人也在 Leroux 的基础上在 [77] 中证明了 2 维带状态的可达性问题是 **PSPACE**-完备的，Czwreński 也在 [78] 中提出了一个关于 2 维带状态的可达性问题是 **PSPACE**-完备的新证明。

遗憾的是，关于更高的维度 (≥ 3) 目前只有在章节四中介绍的 KLMST 算法中和章节六中下界证明的结论，即 d 维带状态的向量加法系统的可达性问题是在 $\mathbf{F}_{d+4}$ 的，以及 $2d+4$ 维带状态的向量加法系统的可达性问题是 $\mathbf{F}_d$-难的。原因主要在于，之前对于 1 维和 2 维的研究里非常依赖其可达集是半线性的这一性质，包括所用到的平坦性，但是这些性质在 3 维以上统统不成立，因此对其的研究，研究者们一直不能找到特殊的性质来获得更为细致的结论。而近些年随着可达性研究的进展，研究者们逐渐发现高维的可达性问题可能确实比较困难，比如针对章节五中提到的 Presburger 不变量算法，Czwreński 在 [74] 中提出了一些比较长的运行会产生比较大的分离算子，导致通过该想法不太可能得到固定维很快的算法。

本章将介绍目前固定维可达性问题上的一些结论。第一节将介绍一维带状态的向量加法系统可达性问题是 **NP**-完备的[75]，第二节则将介绍固定维目前最好的结果-二维带状态的向量加法系统的可达性问题是 **PSPACE**-完备的[77]，第三节将介绍一些研究者对高维可达性问题的研究结论[74]，第四节则是本章小结。

7.1 1 维带状态向量加法系统可达性的讨论

本节将介绍一维带状态向量加法系统的可达性问题，首先将介绍 **NP**-完备的结论[75]，其中下界可以通过子集和问题来规约完成，而上界则将通过分析路径的结构来获取。然后本节将介绍对于一类子类问题-一元更新的一维带状态的向量加法系统，其中一元更新指的是每条规则对维度上的值的更改范围为 $\{-1, 0, 1\}$，其可达路径可以转换成更为漂亮的结构，即只包含一个圈的线性路径策略。

7.1.1 1 维带状态的向量加法系统的可达性问题是 NP-完备的

7.1.1.1 **NP**-难的证明

我们将子集和问题规约到 1 维带状态的向量加法系统的可达性问题来完成下界的证明。子集合问题定义如下：

问题 7.1 (子集合问题) 给定一个由非负整数组成的集合 $S = \{a_1, \ldots, a_n\}$ 和目标值 C，问是否存在一个子集 $S' \subseteq S$ 满足 S' 里所有元素的和为 C?

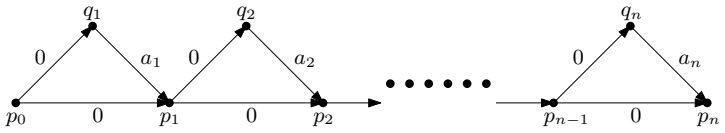


图 7–1 用来解子集合问题的 1-VASS V_1

Figure 7–1 1-VASS V_1 for subset sum problem

该问题是一个经典的 **NP**-完备问题[160]，可以通过构造一个对应的 1-VASS 的可达性问题来解决。如图7–1所示，我们将说明存在子集 S' 当且仅当 V_1 中存在 $p_0(0)$ 到 $p_n(C)$ 的一条路径。事实上，如果存在 $S' = \{a_{i_1}, \ldots a_{i_k}\}$，则 V_1 中存在如下的一条路径：

$$p_0(0) \to \cdots p_{i_1-1}(c_1) \xrightarrow{0} q_{i_1}(c_1) \xrightarrow{a_{i_1}} p_{i_1}(c_2) \to \cdots p_j(c_i) \xrightarrow{0} p_{j+1}(c_i) \to \cdots p_n(C)$$

其中如果 $a_l \in S'$ 则该路径经过 q_l，否则就不经过 q_l，不难验证该是一条可达的路径。反之如果存在一条可达的路径，则可以根据其经过的 $\{q_i\}$ 点构造一个子集 S' 满足其和是 C。注意到该构造显然是多项式时间的，因此有了如下的引理，即本节的主要结论：

引理 7.1 (Haase[75]) 1 维带状态的向量加法系统的可达性问题是 **NP**-难的。

7.1.1.2　在 **NP** 的算法

本节将说明 1 维带状态的向量加法系统的可达性问题是在 **NP** 的。给定 1-VASS$V = (Q, T, A)$ 和两个格局 $p(u), q(v)$，$n = |V|$ 表示 V 的大小。令 l 是 V 中的一个圈，如果 $\Delta(l) < 0$ 我们则称该圈是负数圈，反之则称为正数圈。令 π 是 $p(u)$ 到 $q(v)$ 的一条路径，则其对应的 Parikh 像 ϕ_π 满足如下两个方程：

$$\mathbf{1}_q - \mathbf{1}_p = \sum_{\mathbf{t}=(p,a,q)\in T} \phi(\mathbf{t})(\mathbf{1}_q - \mathbf{1}_p) \tag{7–1}$$

$$\Delta(\phi) = v - u \tag{7–2}$$

这里 $\mathbf{1}_p$ 与章节四相同，表示状态 p 的指示向量，显然可以在多项式时间内验证一个 Parikh 像 ϕ 是否是其的解。如果 ϕ 中存在在 $\phi \le \phi$ 使得 ϕ' 是某个圈 l 的 Parikh 像，则称 ϕ 中存在圈 l。尽管 ϕ 并不能对应一条 V 里的运行，但是下述引理说明只要满足一定的条件，便可以通过 ϕ 在多项式时间内判断是否存在一个运行。

引理 7.2　给定一个满足方程7–1和7–2的 Parikh 像 ϕ，如果 ϕ 满足以下之一条件：

- ϕ 中不存在正数圈。
- ϕ 中不存在负数圈。

则可以在 **NP** 内判断其是否对应 $p(u)$ 到 $q(v)$ 的一条路径 π。特别的，如果 π 存在，则 $|\pi| \le n|v - u|$。

证明　只用证 ϕ 不存在正数圈的情况，令 $q_0 = p, q_1, \dots q_k = q \in Q$ 表示 ϕ 中所有用到的边出现过的状态的集合。下面证明其对应 $p(u)$ 到 $q(v)$ 的一条路径 π，当且仅当存在满足 ϕ 可被分解成 $\phi_1, \phi_2, \dots \phi_k$ 满足：

- $\sum_{i=1}^{k} \Delta(\phi_i) = \Delta(\phi) = v - u$。
- 对于 $j \in [k]$ 有 $\sum_{i=1}^{j} \Delta(\phi_i) > -u$。
- 对于 $j \in [k]$，ϕ_j 满足下列方程：

$$\mathbf{1}_{q_j} - \mathbf{1}_{q_{j-1}} = \sum_{\mathbf{t}=(p,a,q)\in T} \phi(\mathbf{t})(\mathbf{1}_q - \mathbf{1}_p)$$

$$\phi_j(t) = 0,\ t = (p, a, q) \in T,\ q \in \{q_1, \dots q_{j-1}\}$$

注意到任何对应 ϕ 的路径 π 也不会存在正数圈，因此只要保证对于 Q 中的每个点，只要保证最后一次运行到该点时，路径的增量还是超过 $\ge -c$ 的即可。严格来说，假设 $p(u) \xrightarrow{\pi} q(v)$，令该路径分解为：

$$p(u) \xrightarrow{\pi_1} q_1(u_1) \to \cdots \xrightarrow{\pi_k} p_k(u_k) \tag{7–3}$$

其中 π_i 表示最后一次运行到状态 p_i 的路径，显然 $p_k(u_k) = q(v)$。令 π_j 对应的 Parikh 像为 ϕ_j，显然 $\{\phi_j\}_{j\in[k]}$ 满足上述要求。反之注意到如果存在 ϕ_j 不对应 $q_{j-1}(u_{j-1})$ 到 $q_j(u_j)$ 的一条边，则一定存在 $q_l \in \{q_j, \dots q_k\}$ 使得 $q_{j-1}(u_{j-1}) \xrightarrow{\pi'_j} q_l(c)$ 满足 $c < 0, \phi_{\pi'_j} \leq \phi_j$。注意到 $\sum_{i=1}^{l} \Delta(\phi_i) > -u$，因此存在 q_l 上的一个圈 l 满足 $\Delta(l) = \sum_{i=j}^{l} \Delta(\phi_i) - \Delta(\phi_{\pi'_j}) > 0$，与条件矛盾。

注意到可以猜测一个 ϕ 的分解然后验证是否满足上述条件，该猜测是多项式大小的，因此可以在 **NP** 内判断其是否对应一条路径，引理得证。 □

引理 7.3 如果以下条件满足：

- 存在 p 上的正数圈 l_1 和 q 上的负数圈 l_2 满足：$p(u) \xrightarrow{l_1} p(u')$, $q(v') \xrightarrow{l_2} q(v)$。
- 存在满足方程7–1和7–2的 Parikh 像 ϕ。

则存在 $p(u)$ 到 $q(v)$ 的一条路径 π，并且 $|\pi| \leq 2^{P_1(n)} \cdot P_2(u+v)$，其中 P_1, P_2 是两个多项式，并且验证上述条件可以在 **NP** 内完成。

证明 由定理2.4存在 ϕ 满足:$|\phi| \leq 2^{P_1(n)} \cdot P_2(u+v)$，其中 P_1, P_2 是两个多项式，令：

$$M = \sum_{t=(p,a,q)\in T, a<0} \phi(t) \cdot a \tag{7–4}$$

由欧拉定理 ϕ 对应一条 p 到 q 的路径 π，令 $L_1 = \Delta(l_1)$, $L_2 = \Delta(l_2)$，则存在如下运行：

$$p(u) \xrightarrow{l_1^{M_1 L_2}} p(u + M_1 L_1 L_2) \xrightarrow{\pi} q(v + M_1 L_1 L_2) \xrightarrow{l_2^{M_1 L_1}} q(v) \tag{7–5}$$

注意到正数圈和负数圈可以通过 Bellman-Ford 算法验证，因此可以在 **NP** 内验证上述条件，引理得证。 □

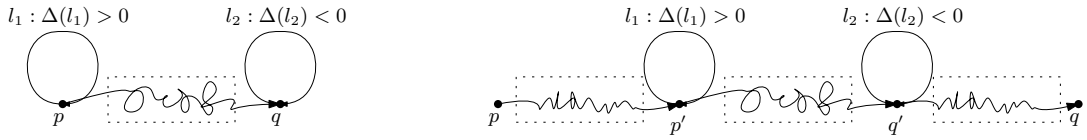


图 7–2 1-VASS 的路径形状

Figure 7–2 the path of 1-VASS

引理7.3说明了一个很简单的充分性性质，即只要 p 存在一个可执行的正数圈 q 存在一个可执行的负数圈，则只要方程7–1和7–2有解，那么就存在一条对应的路径，如图7–2的左半部分。下面将这个想法扩展到所有路径上，可以说明如果 $p(u)$ 到

$q(v)$ 有一条路径，那么其便会存在一条比较有规律的路径，如图7–2的右半部分。首先注意到对于一条路径 π，其没有负数圈和其对应的 Parikh 像 ϕ_π 没有负数圈是不一样的，因此有：

引理 7.4　令 π 是 $p(u)$ 到 $q(v)$ 的一条路径，ϕ_π 是对应的 Parikh 像。如果 π 中没有正数圈，则下列之一成立：

- ϕ_π 中没有正数圈。
- 存在一条 $p(u)$ 到 $q(v)$ 的路径 $\theta = \theta_1\theta_2\theta_3$ 满足 $|\theta_1| < |\pi|$，θ_2 则是一个正数圈。

如果 π 中没有负数圈，则下列之一成立：

- ϕ_π 中没有负数圈。
- 存在一条 $p(u)$ 到 $q(v)$ 的路径 $\theta = \theta_1\theta_2\theta_3$ 满足 $|\theta_3| < |\pi|$，θ_2 则是一个负数圈。

证明　只用证明没有正数圈的情况。反设 ϕ_π 存在正数圈 l，令 $p(u) \xrightarrow{\pi'} r(w)$ 是该运行中第一次运行到 l 上的状态。如果 $r(w) \xrightarrow{l} r(w')$ 则引理已经成立，反之令 $r \xrightarrow{l_1} r' \xrightarrow{l_2} r$，其中 $l = l_1l_2$，r' 是使得 $\Delta(l_1)$ 最小的状态，很显然有 $w+\Delta(l_1) < 0$。考察 π 中第一次运行到 r' 的时刻，由假设 r' 会后于 r 碰到，即 $p(u) \xrightarrow{\pi'} r(w) \xrightarrow{l_3} r'(W)$，令 $l' = l_3l_2$，则有：

- $\Delta(l) = \Delta(l_3) + \Delta(l_2) > \Delta(l_1) + \Delta(l_2) > 0$。
- $\Delta(l_3) > -u$。

因此 $r(w) \xrightarrow{l'} r(w'')$。从而令 $\theta = \pi' l' \theta_3$，$\theta_3$ 是 ϕ_π 剩下的 Parikh 像组成的路径，由欧拉定理存在 r 到 q 的路径，从而引理得证。　□

引理 7.5　如果 $q(v)$ 对于 $p(u)$ 是可达的，则存在一条路径 $\pi = \pi_1\pi_2\pi_3$ 满足：

- π_1 的 Parikh 像 ϕ_1 不存在正数圈。
- π_3 的 Parikh 像 ϕ_3 不存在负数圈。
- 令 $p(u) \xrightarrow{\pi_1} p'(u')$，$q'(v') \xrightarrow{\pi_3} q(v)$，则 p' 上存在一个可执行的正数圈，q' 上存在一个可执行的负数圈。

证明　由条件存在 $\pi = t_1 \dots t_k$ 满足：

$$p(u) \xrightarrow{t_1} p_1(u_1) \xrightarrow{t_2} \cdots \xrightarrow{t_k} p_k(u_k)(= q(v)) \tag{7–6}$$

对 π 分情况讨论：

情况一：π 中没有正数圈。如果其对应的 Parikh 像 ϕ_π 也没有正数圈，则令 $\pi_1 = \pi, \pi_2 = \pi_3 = \emptyset$，引理成立。反之由引理7.4存在一条 $p(u)$ 到 $q(v)$ 的路径

$\theta = \theta_1\theta_2\theta_3$ 满足：θ_1 对应的 Parikh 像没有正数圈，θ_2 是一个正数圈。如果 θ_3 对应的 Parikh 像没有负数圈，则令 $\pi_1 = \theta_1, \pi_2 = \emptyset, \pi_3 = \theta_2\theta_3$ 引理成立。否则令 $p(u) \xrightarrow{\theta_1\theta_2} p'(u')$ 由引理7.4存在一条 $p'(u')$ 到 $q(v)$ 的路径 $\theta' = \theta'_1\theta'_2\theta'_3$ 满足：θ'_3 对应的 Parikh 像没有负数圈，θ_2 是一个负数圈，则令 $\pi_1 = \theta_1$，$\pi_2 = \theta_2\theta'_1\theta'_2$，$\pi_3 = \theta'_3$ 即可。

下面令 π 中有正数圈 l_1，并且 $p_i(u_i)$ 是第一个在圈 l_1 上的状态。

情况二：p_i 到 p_k 之间没有负数圈。则由引理7.4和情况一的讨论可知，存在一条路径 $\pi_1\pi_2\pi_3$ 满足上述要求。

情况三：p_i 到 p_k 之间有负数圈 l_2，令 $p_j(u_j)$ 是运行中最后一个在圈 l_2 上的格局。考虑路径 $t_{i+1} \dots t_j$，由条件 p_i 上有个可执行的正数圈 l_1，p_j 上存在一个可执行的负数圈 l_2，因此由情况一的讨论，存在一条路径 $\pi_1\pi_2\pi_3$ 满足上述要求，从而引理得证。 □

注意到在引理7.5中由引理7.2和7.3 ϕ_1, ϕ_2, ϕ_3 都是多项式大小的，因此其可以在 **NP** 里判定，结合上一节的结论有：

定理 7.1 [75] 1 维带状态的向量加法系统的可达性问题是 **NP**-完备的。

7.1.2 一元更新的 1 维带状态的向量加法系统的路径结构

本节将讨论一类特殊的 1 维带状态的向量加法系统 $V = (Q, T, A)$，其中 $A = \{-1, 0, 1\}$，称满足这样的向量加法系统为一元更新 (unary update) 的。这一节将证明在这类特殊的向量加法系统里，大部分可达的路径可以转换成中间只有一个圈的路径，而在下一节则可以看到，该结构就是最简单的线性路径策略，具体如下引理：

引理 7.6 (Valiant[161]) 令 V 是一元更新 1 维带状态的向量加法系统，如果存在 $p(u)$ 到 $q(v)$ 的一条路径并且 $|v-u| > 2|Q|^2$，则其存在一条可达的路径 $\pi = \alpha_1\beta^i\alpha_2$，其中：

- β 是一个圈并且有 $|\beta| < |Q|$，$i \in \mathbb{N}$。
- $|\alpha_1\alpha_2| < |Q|^2 + |Q|^3$。

证明 只用考虑 $u - v > 2|Q|^2$ 的情况，不妨假设 $v \geq |Q|^2$。对于每个状态 $r \in Q$，定义 $s(r) = \max_\gamma -\Delta(\gamma)/|\gamma|$，其中 γ 是一个包含 r 的简单圈，显然 $|\gamma| \leq |Q|$，直观上来说 $s(r)$ 表示了该状态将数值下降的最快速度，由定义 $s(r) \leq 1$。令 π 是 $p(u)$ 到 $q(v)$ 的一条最短的路径，考虑其中使得 $s(\_)$ 最大的状态 r 和其对应的下降圈

γ_r，令 $|\Delta(\gamma_r)| = D$。现在考虑 π 中所有的简单圈，若有圈 $\gamma_1, \ldots, \gamma_k$ 满足其增量和是 $-D$ 的倍数，则删去对应的圈，则最后剩下的路径 π' 中，最多只有 $D-1$ 个简单圈，否则依旧可以找出若干个圈的增量是 D 的倍数，从而 $|\pi'| < (D-1)|Q| + |Q| < |Q|^2$。将 π' 以 r 为界写成 $\alpha_1\alpha_2$，则有 $p(u) \xrightarrow{\alpha_1\alpha 2} q(v + kD)$，其中 $k \in \mathbb{N}$，注意到 $u - v > |Q|^2$, $v \geq |Q|^2$，因此对于任何的 $k' \leq k$ 都有:

$$p(u) \xrightarrow{\alpha_1\gamma_r^{k'}\alpha_2} q(v + (k-k')D) \tag{7–7}$$

若 $v < |Q|^2$，则令 $p'(u')$ 为 $p(u) \xrightarrow{\pi} q(v)$ 中最后一个满足 $u' \geq |Q|^2$ 的格局，则由前面讨论存在 π_1 满足:

- $p(u) \xrightarrow{\pi_1} p'(u')$。
- $\pi_1 = \alpha_1\beta^i\alpha_2$ 并且满足 $|\alpha_1\alpha_2| < |Q|^2, |\beta| < |Q|$。

令 $p'(u') \xrightarrow{\pi_2} q(v)$，由 $p'(u')$ 的选择可知 $|\pi_2| < |Q|^3$，因此令 $\alpha_2' = \alpha_2\pi_2$，则有:

$$p(u) \xrightarrow{\alpha_1\beta^i\alpha_2} q(v) \tag{7–8}$$

从而引理成立。 □

由上述引理马上可得:

定理 7.2　令 V 是一元更新的 1 维带状态的向量加法系统，如果存在 $p(u)$ 到 $q(v)$ 的一条路径，则存在一条可达的路径 π 满足：$|\pi| < 10|Q|^4 + |v-u| \cdot |Q|$。

证明　先考虑 $|v-u| > 2|Q|^2$ 的情况，由引理7.6存在一条形如 $\pi = \alpha_1\beta^i\alpha_2$ 的路径，其中 $|\alpha_1\alpha_2| < |Q|^2 + |Q|^3$, $|\beta| < |Q|$，注意到 $i < |v-u| + |\alpha_1\alpha_2|$ 从而我们有

$$|\pi| < |Q|^2 + |Q|^3 + i|\beta| < 10|Q|^4 + |v-u| \cdot |Q| \tag{7–9}$$

再考虑 $|v-u| \leq 2|Q|^2$ 的情况。令 π 是 $p(u)$ 到 $q(v)$ 的最短路径，则下列之一必定满足:

- 对于运行中任意的格局 $p'(u')$ 都有 $|u'-v| < 2|Q|^2$。
- 存在中间格局 $p'(u')$ 满足 $|u'-v| = 2|Q|^2$。

如果是前者，则有 $|\pi| < 4|Q|^3$。如果是后者，则由三角不等式有:

$$|u'-u| \leq |u'-v| + |u-v| \leq 4|Q|^2 \tag{7–10}$$

从而 $p(u)$ 到 $q(v)$ 的最短路径可以被分解为 $p(u)$ 到 $p'(u)'$ 的最短路径 π_1 与 $p'(u')$ 到 $q(v)$ 的最短路径 π_2 之和，即:

$$|\pi| < 8|Q|^3 + |Q|^2 + |Q|^3 + 2|Q|^3 < 10|Q|^4 \tag{7–11}$$

因此，定理得证。 □

注 注意到该定理并不能表示一元更新的 1-VASS 的可达性问题是 **NP** 里的，原因在于如果 u, v 是以二进制作为输入，其刻画的路径长度相对于输入是指数大小的，但其依旧给了一个路径的好的刻画。事实上，一般情况下的 1-VASS 的可达路径也可以有类似的处理，因为我们可以将 (p, a, q) 拆分成 a 条规则将其转换成一个一元更新的 1-VASS 看待。

7.2 2 维带状态向量加法系统可达性的讨论

本节将介绍 2 维带状态的向量加法系统的可达性问题，首先将介绍线性路径策略 [76-77]，并证明 2-VASS 的可达路径都可以用该策略替代，然后将介绍其可达性问题是 **PSPACE**-完备的[77]，其下界通过有界的 1-计数程序的可达性问题[88]规约获得，而上界则是通过分析线性路径策略获得可达路径的长度[77] 获得。

7.2.1 线性路径策略

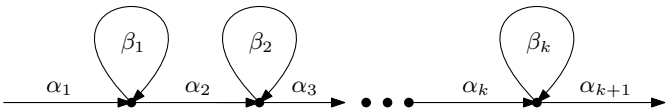


图 7–3 线性路径策略

Figure 7–3 linear path scheme

如图7–3所示，一个线性路径策略 $\rho = \alpha_1\beta_1^* \ldots \alpha_k\beta_k^*\alpha_{k+1}$ 包含了所有可以写成 $\alpha_1\beta_1^* \ldots \alpha_k\beta_k^*\alpha_{k+1}$ 的路径，其中对于任意的 $i \in [k]$，β_i 是一个圈，即 $p(u) \xrightarrow{\rho} q(v)$ 表示存在 $l_1, \ldots l_k$ 使得 $p(u) \xrightarrow{\alpha_1\beta_1^{l_1} \ldots \alpha_k\beta_k^{l_k}\alpha_{k+1}} q(v)$，令 ρ 的长度为 $|\rho| \overset{\text{def}}{=} |\alpha_1\beta_1 \ldots \alpha_k\beta_k\alpha_{k+1}|$，用 $\delta(\rho)$ 表示符合 ρ 的所有的路径的增量集合，即 $\delta(\rho) = \{\Delta(\pi) | \pi = \alpha_1\beta_1^{l_1} \ldots \alpha_k\beta_k^{l_k}\alpha_{k+1}, l_1 \ldots l_k \in \mathbb{N}\}$。

事实上，任何一条路径都可以看成某个线性路径策略，因为 β_i 可以是一个非常复杂的嵌套圈。但下面将说明，有些特定情况下可以用非常简单的线性路径策略来刻画所有的可达路径。

7.2.1.1 d-VASS 在 $\mathbb{Z}^d$ 上的可达路径

首先如果允许格局走到 < 0 的分量，即考虑在 $\mathbb{Z}$ 上的可达性，那么对于任意维向量加法系统，其可达路径都可以用较短的线性路径策略来刻画。直观上来说在 $\mathbb{Z}^d$ 上可以任意掉换规则的顺序，只要其满足在对应的有向图的可达关系即可，假设 $p(\mathbf{u})$ 到 $q(\mathbf{v})$ 存在一条路径 π，则先可以考虑一条 p 到 q 的路径，其经过了 π

上所有的状态，则 π 剩下用到的规则必然是若干圈的组合，将其拆分成简单圈插进上面的路径即可。因此得到了 d 维带状态向量加法系统在 $\mathbb{Z}^d$ 上的可达路径的刻画：

定理 7.3 (Blondin[77]) 令 $V = (Q, T, A)$ 是一个 d 维带状态的向量加法系统，$p(\mathbf{u}), q(\mathbf{v})$ 是两个格局。令 S 是一个线性路径策略的集合，满足任意 $\rho \in S$，$|\rho| \leq 2|Q| \cdot |T|$，并且 ρ 至多有 $|T|$ 个圈，则有：

$$p(\mathbf{u}) \xrightarrow{*}_{\mathbb{Z}^d} q(\mathbf{v}) \iff p(\mathbf{u}) \xrightarrow{S}_{\mathbb{Z}^d} q(\mathbf{v}) \tag{7–12}$$

证明 只需要考虑 $\Rightarrow$ 的方向。令 $p(\mathbf{u})$ 到 $q(\mathbf{v})$ 存在一条路径 π，则其 Parikh 像 ϕ 满足如下两个方程：

$$\mathbf{1}_q - \mathbf{1}_p = \sum_{\mathbf{t}=(p,a,q)\in T} \phi(\mathbf{t})(\mathbf{1}_q - \mathbf{1}_p) \tag{7–13}$$

$$\Delta(\phi) = \mathbf{v} - \mathbf{u} \tag{7–14}$$

由 Euler 定理，其存在一条最短的 p 到 q 的路径 ρ_0 满足其经过了 π 中所有经过的状态，并且使用了所有 π 用到的边，从而 $|\rho_0| \leq |Q||T|$。令其对应的 Parikh 像为 ϕ_0，记 $\phi_1 = \phi - \phi_0$，则 ϕ_1 满足下列方程：

$$0 = \sum_{\mathbf{t}=(p,\mathbf{a},q)\in T} \phi(\mathbf{t})(\mathbf{1}_q - \mathbf{1}_p) \tag{7–15}$$

因此 ϕ_1 可以视作若干个简单圈的集合。按如下方法将这些简单圈加入 π_0 中：

- 令 $\mathbf{t}$ 是满足 $\phi_1(\mathbf{t}) > 0$ 中使用次数最少的。
- 取出 ϕ_1 中一个包含 $\mathbf{t}$ 的简单圈 β_1。
- 令 $\rho = \alpha_1 t \alpha_2$，将 $\phi_1(\mathbf{t})$ 个圈加入到 ρ 中 α_1 后的位置，得到新的路径即 $\rho_1 = \alpha_1 \beta_1^{\phi_1(\mathbf{t})} t \alpha_2$。

记 ρ_1 对应的 Parikh 像为 ϕ_1，令 $\phi_2 = \phi - \phi_1$，重复上述过程直到 $\phi = \phi_k$，得到了如下的一条路径：

$$\rho_k = \alpha_1 \beta_{i_1}^{\phi_{i_1}(\mathbf{t}_{i_1})} \alpha_2 \ldots \beta_{i_k}^{\phi_{i_k}(\mathbf{t}_{i_k})} \alpha_{k+1} \tag{7–16}$$

其中 $i_1, i_2 \ldots i_k$ 是 $[k]$ 的一个重排列，令线性路径策略 $\rho = \alpha_1 \beta_{i_1}^* \ldots \alpha_k \beta_{i_k}^* \alpha_{k+1}$，注意到上述方法至多执行 $|T|$ 轮，因此有：

- $|\rho| \leq |\rho_0| + k \cdot \max_{j\in[k]} |\beta_j| \leq 2|Q| \cdot |T|$。
- $k \leq |T|$，$|\beta_i| \leq |Q|$。

从而 $p(\mathbf{u}) \xrightarrow{\rho}_{\mathbb{Z}^d} q(\mathbf{v})$，即 $p(\mathbf{u}) \xrightarrow{S}_{\mathbb{Z}^d} q(\mathbf{v})$，定理得证。 □

7.2.1.2 2-VASS 的可达路径

本节将介绍 Blondin 在 [77] 中的关键结论，即在 2 维带状态的向量加法系统的可达路径也可以用简单的线性路径策略来刻画：

定理 7.4 (Blondin[77]) 令 $V = (Q, T, A)$ 是一个 2 维带状态的向量加法系统，$p(\mathbf{u}), q(\mathbf{v})$ 是两个格局。令 S 是一个线性路径策略的集合，满足任意 $\rho \in S$，$|\rho| \leq 2^{17}(|Q| \cdot |T| \cdot ||T||)^{15}$，并且 ρ 至多有 $6|Q|^2$ 个圈，则有：

$$p(\mathbf{u}) \xrightarrow{*} q(\mathbf{v}) \iff p(\mathbf{u}) \xrightarrow{S} q(\mathbf{v}) \tag{7–17}$$

假设存在一条路径 π 使得 $p(\mathbf{u}) \xrightarrow{\pi} q(\mathbf{v})$。首先先考虑一个简单的情况，即 $p = q$。注意到在这种情况下，任何一条路径都能拆解成若干的圈的组合，并且其先后顺序不受到严格的限制。同时注意到在两维的情况下，至多两个向量是线性无关的，因此在 u, v 足够大的情况下，可以将其中的圈转换成两个增量线性无关的圈，这一性质在 [76] 被称为终极平坦性 (ultimately flat)。具体来说，令 $D = 2^{17} \cdot (|Q| \cdot |T| \cdot ||T||)^{14}$ 以及 $\mathbb{D} = [D, +\infty]$，则有如下引理：

引理 7.7 令 $V = (Q, T, A)$ 是一个 2 维带状态的向量加法系统，$p(\mathbf{u}), p(\mathbf{v})$ 是两个格局满足 $u, v \in \mathbb{D}^2$。令 S 是一个线性路径策略的集合，满足任意 $\rho \in S$，$|\rho| \leq 2^{17}(|Q| \cdot |T| \cdot ||T||)^{13}$，并且 ρ 至多有 2 个圈，则有：

$$p(\mathbf{u}) \xrightarrow{*} p(\mathbf{v}) \iff p(\mathbf{u}) \xrightarrow{S} p(\mathbf{v}) \tag{7–18}$$

注 这一想法在两维能成立而更高维度不能成立的原因在于这两个圈可能数量是不平衡的，但只有两个圈的时候一个用完还剩一个就接着用；但出现三个圈以上的时候一个用完可能还剩两个，此时它们之间的顺序还是需要值得考虑的，从而会产生某种嵌套的复杂关系。

其次再回顾一下上面关于 d 维带状态的向量加法系统在 $\mathbb{Z}^d$ 上的路径结构的证明，其很重要的一点是我们几乎可以任意的调换规则的顺序，这对于一般的可达路径来说是不可能的；但是在如果该路径上每个格局的向量都非常大的情况下，调换顺序不会使得格局的值比 0 小，从而可以利用类似上面的做法。具体来说，本节将利用引理7.7证明在 $\mathbb{D} \times \mathbb{D}$ 上的可达路径有简单的线性路径策略，即如下引理：

引理 7.8 令 $V = (Q, T, A)$ 是一个 2 维带状态的向量加法系统，$p(\mathbf{u}), q(\mathbf{v})$ 是两个格局满足 $u, v \in \mathbb{D}^2$。令 S 是一个线性路径策略的集合，满足任意 $\rho \in S$，$|\rho| \leq 2^{17}(|Q| \cdot |T| \cdot ||T||)^{14}$，并且 ρ 至多有 $2|Q|$ 个圈，则有：

$$p(\mathbf{u}) \xrightarrow{*}_{\mathbb{D}^2} q(\mathbf{v}) \iff p(\mathbf{u}) \xrightarrow{S} q(\mathbf{v}) \tag{7–19}$$

最后，当 u, v 满足某一维有界的时候，其可以被视作降维。因此利用上一节 1 维带状态向量加法系统的结论即引理7.6和定理7.2，令 $\mathbb{L} = [0, D] \times \mathbb{N} \cup \mathbb{N} \times [0, D]$，可以证明在 $\mathbb{L}$ 上的可达路径也有简单的线性路径策略：

引理 7.9　令 $V = (Q, T, A)$ 是一个 2 维带状态的向量加法系统，$p(\mathbf{u}), q(\mathbf{v})$ 是两个格局满足 $\mathbf{u}, \mathbf{v} \in \mathbb{L}$。令 S 是一个线性路径策略的集合，满足任意 $\rho \in S$，$|\rho| \leq P_3$，并且 ρ 至多有 2 个圈，则有：

$$p(\mathbf{u}) \xrightarrow{*}_{\mathbb{L}} q(\mathbf{v}) \Longleftrightarrow p(\mathbf{u}) \xrightarrow{S} q(\mathbf{v}) \tag{7–20}$$

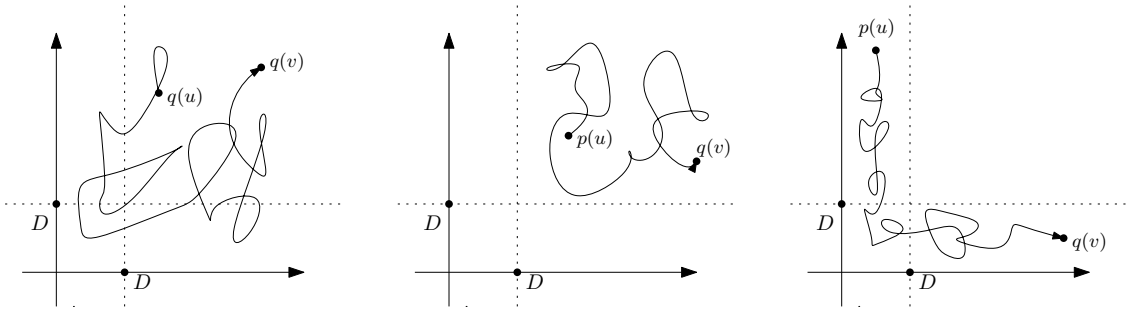


图 7–4 2-VASS 的三类路径

Figure 7–4 three kinds of path in 2-VASS

上述三个引理刻画了 2 维带状态向量加法系统中三种不同的可达路径，如图所示。接下来说明任何 2-VASS 里的一条可达路径，都可以拆分成若干种这些路径，从而由引理7.7，7.8，7.9来完成对定理7.4的证明。

证明 (定理7.4的证明)　令 $L = D$，定义如下两个区域：

- $\mathbb{L}_1 = [L, +\infty] \times [L, +\infty]$。
- $\mathbb{L}_2 = [0, L + ||T||] \times \mathbb{N} \cup \mathbb{N} \times [0, L + ||T||]$。

并记 $\mathbb{L} = \mathbb{L}_1 \cap \mathbb{L}_2$。令 $p(u)$ 到 $q(v)$ 的一条路径 $\boldsymbol{\pi} = \mathbf{t}_1 \ldots \mathbf{t}_k$ 对应的运行为：

$$p(\mathbf{u}) = q_0(\mathbf{v}_0) \xrightarrow{\mathbf{t}_1} q_1(\mathbf{v}_1) \xrightarrow{\mathbf{t}_2} \cdots \xrightarrow{\mathbf{t}_k} q_k(\mathbf{t}_k) = q(\mathbf{v}) \tag{7–21}$$

考察其中所有使得 $\mathbf{v}_j$ 在 $\mathbb{L}$ 里的下标，令其为 $i_1 < i_2 < \ldots < i_l$, $l < k$，记 $l(i_j) \stackrel{\text{def}}{=} \max_{q_{i_j} = q_{i_h}} i_h$，令 $h^0, h_1, h^1, h_2, h^2, \ldots h_m, h^m, h_{m+1}$ 表示如下下标：

- $h^0 = 0,\ h_{m+1} = k,\ h_1 = i_1,\ h^m = i_k$。
- 记 $h^j = i_l$，则 $h_{j+1} = i_{l+1}$。
- $h^j = i_{l(h_j)}$，即 $q_{h_j} = q_{h^j}$。

则可以将上述运行分解成如下段：

$$q_0(\mathbf{v}_0) \xrightarrow{\pi_1} q_{h_1}(\mathbf{v}_{h_1}) \xrightarrow{\pi^1} q_{h^1}(\mathbf{v}_{h^1}) \xrightarrow{\pi_2} \cdots \xrightarrow{\pi^m} q_{h^m}(\mathbf{v}_{h^m}) \xrightarrow{\pi_{m+1}} q_k(\mathbf{v}_k)$$

该分解满足：

- $m \le |Q|$。
- 对于 $j \in [m]$ 运行 $q_{h_j}(\mathbf{v}_{h_j}) \xrightarrow{\pi^j} q_{h^j}(\mathbf{v}_{h^j})$ 满足：$\mathbf{v}_{h_j}, \mathbf{v}_{h^j} \in \mathbb{L}_1$。
- 对于 $j \in [m+1]$ 运行 $q_{h^{j-1}}(\mathbf{v}_{h^{j-1}}) \xrightarrow{\pi^j} q_{h_j}(\mathbf{v}_{h_j})$ 满足：$q_{h^{j-1}}(\mathbf{v}_{h^{j-1}}) \xrightarrow{\pi^j}_{\mathbb{L}_1} q_{h_j}(\mathbf{v}_{h_j})$ 或者 $q_{h^{j-1}}(\mathbf{v}_{h^{j-1}}) \xrightarrow{\pi^j}_{\mathbb{L}_2} q_{h_j}(\mathbf{v}_{h_j})$。

令 $P_1 = \max\{D, P_3(D + ||T||)\}, P_2 = 2^{17}(|Q| \cdot |T| \cdot ||T||)^{14}$，定义如下三个线性路径策略集合 S_1, S_2, S：

- 对任意 $\rho \in S_1$，ρ 至多有 2 个圈，并且 $|\rho| \le P_1$。
- 对任意 $\rho \in S_2$，ρ 至多有 $2|Q|$ 个圈，并且 $|\rho| \le P_2$。
- 对任意 $\rho \in S$，ρ 至多有 $6|Q|^2$ 个圈，并且 $|\rho| \le (1 + |Q|)P_2 + |Q|P_1$。

从而由引理7.7存在 $\rho_j \in S_1$ 满足 $q_{h_j}(\mathbf{v}_{h_j}) \xrightarrow{\rho_j} q_{h^j}(\mathbf{v}_{h^j})$，由引理7.8和引理7.9存在 $\rho^j \in S_1 \cup S_2$ 满足：$q_{h^{j-1}}(\mathbf{v}_{h^{j-1}}) \xrightarrow{\rho^j} q_{h_j}(\mathbf{v}_{h_j})$，从而存在 $p(u)$ 到 $q(v)$ 的一个线性路径策略 ρ' 满足：

- $|\rho'| \le (1 + |Q|)P_2 + |Q|P_1 < 2^{17}(|Q| \cdot |T| \cdot ||T||)^{15}$。
- ρ' 中圈的个数有 $2|Q|(|Q| + 1) + 2 \cdot |Q| < 6|Q|^2$。

即存在 $\rho \in S$ 满足 $p(\mathbf{u}) \xrightarrow{\rho} q(\mathbf{v})$，定理得证。 □

接下来只需要完成三个引理的证明。

7.2.1.3 引理7.7和7.8的证明

现在来完成引理7.7和7.8的证明。引理7.8的证明是简单的，事实上，对于 $p(\mathbf{u})$ 到 $q(\mathbf{v})$ 在 $\mathbb{D}^2$ 上的一条路径 $\boldsymbol{\pi}$ 可以分解为：

$$p(\mathbf{u}) \xrightarrow{\alpha_1}_{\mathbb{D}^2} q_1(\mathbf{v}_1) \xrightarrow{\beta_1}_{\mathbb{D}^2} q_1(\mathbf{v}'_1) \xrightarrow{\alpha_2}_{\mathbb{D}^2} \cdots \xrightarrow{\alpha_k}_{\mathbb{D}^2} q_k(\mathbf{v}_k) \xrightarrow{\beta_k}_{\mathbb{D}^2} q_k(\mathbf{v}'_k) \xrightarrow{\alpha_{k+1}}_{\mathbb{D}^2} q(\mathbf{v})$$

这里 β_i 的选择是满足 $q_i(\mathbf{v}'_i)$ 是运行中最后到达 q_i 状态的格局，$|\alpha_i| \le |Q|$，因此有 $|\alpha_1 \dots \alpha_{k+1}| \le |Q|^2$。由引理7.7，每个 β_i 可以被一个不超过两个圈，长度不超过 P_1 的线性路径策略所替代，因此令线性路径策略集合 S 满足，对任意的 $\rho \in S$，ρ 至多有 $2|Q|$ 个圈，并且 $\rho \le |Q|^2 + |Q| \cdot 2^{17}(|Q| \cdot |T| \cdot ||T||)^{13} \le 2^{17}(|Q| \cdot |T| \cdot ||T||)^{14}$，则有 $p(\mathbf{u}) \xrightarrow{*}_{\mathbb{D}^2} q(\mathbf{v})$ 当且仅当 $p(\mathbf{u}) \xrightarrow{S} q(\mathbf{v})$。

接下来完成引理7.7的证明。正如之前所说，我们先证明在 $\mathbf{u}, \mathbf{v}$ 都充分大的情况下，$p(\mathbf{u})$ 到 $p(\mathbf{v})$ 的路径可以视作只有两个等效的圈在起作用。事实上，考虑一个线

性路径策略，其所表示的所有路径的增量集合可以看作一个带偏移的锥集。下面说明，这样一个带偏移的锥集与某个象限相交后的交集实际上是一堆带偏移的锥集的并集满足，这些锥集的生成元素都是同向的并且不超过其个数不超过 2。下面先补充一些记号，记 $\mathbb{K}$ 表示某个象限，即 $\mathbb{K} \in \{\mathbb{N}\times\mathbb{N}, (-\mathbb{N})\times\mathbb{N}, (-\mathbb{N})\times(-\mathbb{N}), \mathbb{N}\times(-\mathbb{N})\}$，设 $e \in \mathbb{N}$ 是个自然数，令 $P = \{\mathbf{p}_1, \ldots \mathbf{p}_k,\ \mathbf{p}_i \in \mathbb{Z}^2\}$，定义 P 的大小为 $||P|| = |P| \cdot \max_i ||p_i||_1$，并且定义 $cone_e(P)$ 是由 P 生成的各系数在 $\mathbb{Q} \cap [0, e]$ 之间的元素的集合，即:$cone_e(P) \stackrel{\text{def}}{=} \{\sum_{i=1}^{k} \lambda_i \mathbf{p}_i,\ \lambda_i \in \mathbb{Q} \cap [0, e]\}$，特别的 $cone(P)$ 表示由 P 生成的锥集，则有下列引理:

引理 7.10 (Blondin[77])　令 $P = \{\mathbf{p}_1, \ldots \mathbf{p}_k\} \subseteq \mathbb{Z}^2$，$\mathbf{b} \in P$，$\mathbb{K}$ 是一个象限，则有:$(\mathbf{b} + cone(P)) \cap \mathbb{K} = \bigcup_{i=1}^{n} \mathbf{c}_i + cone(P_i)$，其中:

- $|P_i| \leq 2$，并且 $P_i \subseteq (P \cup \mathbf{b} + cone(P)) \cap \mathbb{K}$。
- 存在 $e \leq 4||P||^{12}$ 使得 $\{\mathbf{c}_i\} \cup (P_i \cap \mathbf{b} + cone(P)) \subseteq \mathbf{b} + cone_e(P)$。

证明　不妨假设 $\mathbb{K} = \mathbb{N}^2$，并且令 P 中元素是两两线性无关的，记 $B = ||P||$，则对于任意的 $i, j, l \in [k]$ 方程 $\lambda_l \mathbf{p}_l = \lambda_i \mathbf{p}_i + \lambda_j \mathbf{p}_j$ 总存在一组解 $\lambda_i, \lambda_j, \lambda_l$ 满足：$\lambda_l \in [1, B^2]$，$\lambda_i, \lambda_j \in [-B^2, B^2]$。从而对于任意的 $\mathbf{p} = \sum_{i=1}^{k} \lambda_i \mathbf{p}_i \in cone(P)$，总能存在 $\mathbf{u}, \mathbf{v} \in P$ 使得 $\mathbf{p}$ 被重写为：$\mathbf{p} = \lambda_u \mathbf{u} + \lambda_v \mathbf{v} + \sum_{\mathbf{p}_i \neq \mathbf{u},\mathbf{v}} \lambda'_i \mathbf{p}_i$ 满足：$\lambda'_i \leq B^2$。因此有:

$$\mathbf{b} + cone(P) = \bigcup_{\mathbf{b}' \in \mathbf{b} + cone_{B^2}(P)} \bigcup_{|P'| \leq 2, P' \subseteq P} \mathbf{b}' + cone(P') \tag{7–22}$$

注意到令 $e = B^4$，则 $\mathbf{b}' \in \mathbf{b} + cone_e(P)$。令 $P' = \{\mathbf{u}, \mathbf{v}\}$，若 $\mathbf{u}, \mathbf{v} \in \mathbb{N}^2$ 则引理已经成立，否则令 $\mathbf{u} = (u_1, u_2), \mathbf{v} = (v_1, v_2)$，接下来分情况讨论:

情况一：有一个元素不属于 $\mathbb{N}^2$，不妨假设 $\mathbf{u} \in \mathbb{N}^2$，并且令 $\mathbf{u}$ 到 $\mathbf{v}$ 的逆时针夹角小于 180°。下面说明，存在 $\mathbf{w} = (0, w)$ 满足以下两点:

- $\mathbf{w} \in \mathbf{b} + cone(P)$。
- 存在 $\zeta_1, \zeta_2 \in \mathbb{N}$ 使得 $\mathbf{w} = \zeta_1 \mathbf{u} + \zeta_2 \mathbf{v}$。

注意到由 $\mathbf{u}, \mathbf{v}$ 的假设，存在 $\mathbf{w}_0 = \eta_1 \mathbf{u} + \eta_2 \mathbf{v}$，其中 $w_0 \in \mathbb{N}, \eta_1, \eta_2 \in [1, B^2] \cap \mathbb{N}$，并且由上述讨论，存在 $\gamma_0 \in [1, B^2], \gamma_1, \gamma_2 \in [-B^2, B^2]$ 使得 $\gamma_0 \mathbf{b} = \gamma_1 \mathbf{u} + \gamma_2 \mathbf{v}$，因而有:

$$\begin{aligned} B^2 \mathbf{w}_0 &= B^2 \eta_1 \mathbf{u} + B^2 \eta_2 \mathbf{v} \\ &= \gamma_0 \mathbf{b} + (B^2 \eta_1 - \gamma_1)\mathbf{u} + (B^2 \eta_2 - \gamma_2)\mathbf{v} \in \mathbf{b} + cone(P) \end{aligned}$$

令 $\mathbf{w} = B^2 \mathbf{w}_0 = \zeta_1 \mathbf{u} + \zeta_2 \mathbf{v}$，其中 $\zeta_i = B^2 \eta_i$，则有 $0 \leq \zeta_i \leq B^4$，从而 $\mathbf{w} \in \mathbf{b} + cone_e(P)$。

现在令 $\mathbf{z} = \mathbf{b}' + \lambda_1 \mathbf{u} + \lambda_2 \mathbf{v} \in \mathbb{N}$，$P'' = \{\mathbf{u}, \mathbf{w}\}$，则有:

- 若 $\lambda_1 \geq \zeta_1 \cdot \lfloor \frac{\lambda_2}{\zeta_2} \rfloor$，令 $\mathbf{b}'' = \mathbf{b}' + (\lambda_2 - \zeta_2 \cdot \lfloor \frac{\lambda_2}{\zeta_2} \rfloor)\mathbf{v}$，则有：

$$\mathbf{z} = \mathbf{b}' + (\lambda_2 - \zeta_2 \cdot \lfloor \frac{\lambda_2}{\zeta_2} \rfloor)\mathbf{v} + (\lambda_1 - \zeta_1 \cdot \lfloor \frac{\lambda_2}{\zeta_2} \rfloor)\mathbf{u} + \lfloor \frac{\lambda_2}{\zeta_2} \rfloor \mathbf{w} \in \mathbf{b}'' + cone(P'')$$

- 若 $\lambda_1 < \zeta_1 \cdot \lfloor \frac{\lambda_2}{\zeta_2} \rfloor$，则有：$\mathbf{z} = \mathbf{b}' + (\lambda_1 - \zeta_1 \cdot \lfloor \frac{\lambda_1}{\zeta_1} \rfloor)\mathbf{u} + (\lambda_2 - \zeta_2 \cdot \lfloor \frac{\lambda_1}{\zeta_1} \rfloor)\mathbf{v} + \lfloor \frac{\lambda_1}{\zeta_1} \rfloor \mathbf{w}$，注意到有：

$$\mathbf{z}[1] \leq \mathbf{b}'[1] + (\lambda_1 - \zeta_1 \cdot \lfloor \frac{\lambda_1}{\zeta_1} \rfloor)\mathbf{u}[1] \leq kB^3 + B^5$$

从而 $(\lambda_2 - \zeta_2 \cdot \lfloor \frac{\lambda_1}{\zeta_1} \rfloor) \leq \frac{kB^3+B^5}{\mathbf{v}[2]}$。

因此对于任何一个 $j \in [\frac{kB^3+B^5}{\mathbf{v}[2]}]$，记 $\mathbf{b}_j = \mathbf{b}' + j\mathbf{v} + u_j\mathbf{u}$，其中 u_j 是使得 $\mathbf{b}_j \in \mathbb{N}^2$ 的最小系数，则：

$$(\mathbf{b}' + cone(P')) \cap \mathbb{N}^2 = \bigcup_{j \in [\frac{kB^3+B^5}{\mathbf{v}[2]}]} \mathbf{b}_j + cone(P'') \tag{7–23}$$

并且令 $e = 4B^{12}$，从而 $\mathbf{b_j}, \mathbf{u}, \mathbf{w} \in +cone_e(P)$。

情况二：$\mathbf{u}, \mathbf{v} \notin \mathbb{N}^2$。此时不妨假设 $\mathbf{u}$ 在第二象限，$\mathbf{v}$ 在第四象限，并且 $cone(\{\mathbf{u}, \mathbf{v}\}) \cap \mathbb{N}^2 \neq \emptyset$，否则其他情况都是有限的。类似于情况一：

- $\mathbf{w}_1 = (w_1, 0) = \eta_1\mathbf{u} + \eta_2\mathbf{v} \in \mathbf{b} + cone(P),\ 0 \leq \eta_1, \eta_2 \leq B^4$。
- $\mathbf{w}_2 = (0, w_2) = \zeta_1\mathbf{u} + \zeta_2\mathbf{v} \in \mathbf{b} + cone(P),\ 0 \leq \zeta_1, \zeta_2 \leq B^4$。

令 $\mathbf{z} = \mathbf{b}' + \lambda_1\mathbf{u} + \lambda_2\mathbf{v} \in \mathbb{N}^2$，$P'' = \{\mathbf{w}_1, \mathbf{w}_2\}$，我们的目标是将此情况下的 $\mathbf{z}$ 都改写成 $\mathbf{b}'' + cone(P'')$ 的形式。考察 $\mathbf{y} = \lambda_1\mathbf{u} + \lambda_2\mathbf{v}$：

- 如果 $\mathbf{y} \in \mathbb{N}^2$，从而令：

$$\mathbf{y} = (\mathbf{y}[1] - w_1 \cdot \lfloor \frac{\mathbf{y}[1]}{w_1} \rfloor, \mathbf{y}[2] - w_2 \cdot \lfloor \frac{\mathbf{y}[2]}{w_2} \rfloor) + \lfloor \frac{\mathbf{y}[1]}{w_1} \rfloor \mathbf{w}_1 + \lfloor \frac{\mathbf{y}[2]}{w_2} \rfloor \mathbf{w}_2$$

注意到 $\mathbf{y} \in \mathbb{N}^2$ 时必有 $\lambda_1, \lambda_2 > 0$，因此有 $(\mathbf{y}[1] - w_1 \cdot \lfloor \frac{\mathbf{y}[1]}{w_1} \rfloor, \mathbf{y}[2] - w_2 \cdot \lfloor \frac{\mathbf{y}[2]}{w_2} \rfloor) = \lambda_1'\mathbf{u} + \lambda_2'\mathbf{v}, \lambda_1', \lambda_2' \in [0, B^4]$。

- 如果 $\mathbf{y} \notin \mathbb{N}^2$，不妨令 $\mathbf{y}[1] > 0, \mathbf{y}[2] \leq 0$，显然 $\mathbf{y}_2 \geq -\mathbf{b}'[2]$。记 $K = \min\{\lfloor \frac{\lambda_1}{\eta_1} \rfloor, \lfloor \frac{\lambda_2}{\eta_2} \rfloor\}$，则 $\mathbf{y}$ 可以写成：

$$\mathbf{y} = K\mathbf{w}_1 + (\lambda_1 - K\eta_1)\mathbf{u} + (\lambda_2 - K\eta_2)\mathbf{v}$$

注意到 $\mathbf{y}[2] = (\lambda_1 - K\eta_1)\mathbf{u}[2] + (\lambda_2 - K\eta_2)\mathbf{v}[2]$，因此有 $\lambda_1 - K\eta_1, \lambda_2 - K\eta_2 \leq B^6$，否则会使得 $\mathbf{y}[2] \geq 0$ 或者 $\mathbf{y}[1] \leq -||\mathbf{b}'||[2]$。

因此对于任何 $i,j\in[0,B^6]$，令 $\mathbf{w}_{i,j}=i\mathbf{u}+j\mathbf{v}$, $\mathbf{w}'_{i,j}=\mathbf{w}_{i,j}+j_i\mathbf{u}+i_j\mathbf{v}$, $\mathbf{w}''_{i,j}=\mathbf{w}_{i,j}+j'_i\mathbf{u}+i'_j\mathbf{v}$，其中 $i_j,j_i\in\mathbb{N}$ 是满足最小的系数使得 $\mathbf{w}'_{i,j}\in\mathbb{N}\times[-\mathbf{b}'[2],+\infty]$，$i'_j,j'_i\in\mathbb{N}$ 是满足最小的系数使得 $\mathbf{w}''_{i,j}\in[-\mathbf{b}'[1],+\infty]\times\mathbb{N}$，显然 $i_j,i'_j,j'_i,j_i\le B^{11}$，从而：

$$\begin{aligned}(\mathbf{b}'+cone(P'))\cap\mathbb{N}^2=&\bigcup_{\mathbf{w}_{i,j}\in\mathbb{N}^2}(\mathbf{b}'+\mathbf{w}_{i,j})+cone(P'')\\ \cup&\bigcup_{i,j}(\mathbf{b}'+\mathbf{w}'_{i,j})+cone(\{\mathbf{w}_1\})\\ \cup&\bigcup_{i,j}(\mathbf{b}'+\mathbf{w}''_{i,j})+cone(\{\mathbf{w}_2\})\end{aligned}$$

引理得证。 □

现在回到引理7.7的证明。不妨假设 $\mathbf{u}\le\mathbf{v}$，注意到 $\mathbf{u},\mathbf{v}\in\mathbb{D}^2$，因此可以适当调换顺序使得 $p(\mathbf{u})$ 到 $p(\mathbf{v})$ 被某些线性路径策略所刻画，从而使用引理7.10将其转换成一个只有两个圈的线性路径策略。

证明 (引理7.7的证明) 下面先来说明对于 p 到 p 自身的一个线性路径策略 $\rho=\alpha_1\beta_1\dots\beta_k\alpha_{k+1}$，可以用一组有限的只有两个圈的线性路径策略集合 R_ρ 所刻画，并且满足：

- $\forall\,\sigma\in R_\rho$ 有 $|\sigma|\le 9(|\rho|\cdot||T||)^{13}$。
- $\forall\,\sigma\in R_\rho$，令其的两个圈为 β_1,β_2，则 $\Delta(\beta_1),\Delta(\beta_2)$ 在同一个象限。

不妨假设 $\alpha_1\dots\alpha_{k+1}$ 也是 ρ 的一个圈，否则可以用 $\rho(\alpha_1\dots\alpha_{k+1})^*$ 来代替。事实上，令 $\mathbf{b}=\Delta(\alpha_1\dots\alpha_{k+1}),P=\{\Delta(\beta_1),\dots\Delta(\beta_k)\}$，则 $||P||\le|\rho|\cdot||T||$ 并且有 $\delta(\rho)=\mathbf{b}+cone(P)$。令 $\mathbb{K}$ 是一个象限，则有：$\delta_{\mathbb{K}}(\rho)=\bigcup_{\mathbb{K}}\delta(\rho)\cap\mathbb{K}$，从而由引理7.10：

$$\delta_{\mathbb{K}}(\rho)=\bigcup_{\mathbb{K}}(\bigcup_{i=1}^{n}\mathbf{c_i}+cone(P_i)) \tag{7–24}$$

其中：

- $|P_i|\le 2$，并且 $P_i\subseteq(P\cup\mathbf{b}+cone(P))\cap\mathbb{K}$。
- 存在 $e\le 4(|\rho|\cdot||T||)^{12}$ 使得 $\{\mathbf{c}_i\}\cup(P_i\cap\mathbf{b}+cone(P))\subseteq\mathbf{b}+cone_e(P)$。

对于任意的 $\mathbf{c}_i$，存在 $e_1\dots e_k\le 4(|\rho|\cdot||T||)^{12}$ 使得 $\mathbf{c}_i=\Delta(\alpha_1\beta_1^{e_1}\dots\beta_k^{e_k}\alpha_{k+1})$，记该条路径为 $\pi_{\mathbf{c}_i}$；对于 P_i 中的元素 $\mathbf{p}$，若 $\mathbf{p}\in\mathbf{b}+cone(P)$，则类似 c_i 存在一条路径 $\pi_{\mathbf{p}}$ 满足：$\mathbf{p}=\Delta(\pi_{\mathbf{p}})$。考虑 P_i 中所有属于 P 的元素，定义如下线性路径策略 σ_{P_i}：

$$\sigma_{\mathbf{p}}=\alpha_1\beta_1^{e_1}\theta_1\dots\beta_k\theta_k\alpha_{k+1}$$

其中 θ_j 满足若 $\Delta(\beta_j)\in P_i$ 则 $\theta_j=\beta_j^*$，反之为空。定义线性路径策略 $\rho_i=\sigma_{P_i}\Pi_{\mathbf{p}\in P_i\setminus P}(\pi_{\mathbf{p}})^*$，则有：$|\rho_i|\le|\rho|+8(|\rho|\cdot||T||)^{13}\le 9(|\rho|\cdot||T||)^{13}$，并且 $\delta_{\rho_i}=\mathbf{c}_i+cone(P_i)$。因此，令 $R_\rho=\{\rho_i\}$，则有 $\delta_(\rho)=\bigcup_{\rho\in R_\rho}\delta_(\rho)$。

回到引理7.10，由定理7.3 $p(\mathbf{u})$ 到 $q(\mathbf{v})$ 在 $\mathbb{Z}^2$ 上的路径可以被一个长度不超过 $2|Q||T|$ 的线性路径策略集合 S 所刻画，定义 $R = \cup_{\rho\in S} R_\rho$，则对于任意的 $\sigma \in R$ 有 $|\sigma| < 2^{17}(|Q|\cdot|T|\cdot||T||)^{13}$，令 $\sigma = \alpha_1\beta_1^*\alpha_2\beta_2^*\alpha_3$，由前面的假设存在 e_1, e_2 使得 $p(\mathbf{u}) \xrightarrow{\alpha_1\beta_1^{e_1}\alpha_2\beta_2^{e_2}\alpha_3}_{\mathbb{Z}^2} p(\mathbf{v})$。注意到 $\mathbf{v}-\mathbf{u}$ 和 $\Delta(\beta_1), \Delta(\beta_2)$ 是同一象限，从而令 $\pi_\sigma = \alpha_1\beta_1\alpha_2\beta_2\alpha_3$ 对任意的 $i \in |\pi_\sigma|$ 有：

$$\mathbf{u} + \Delta(\sigma_\pi[1,j]) \geq (D,D) - (|\sigma|\cdot||T||, |\sigma|\cdot||T||) > 0 \tag{7–25}$$

从而 $p(\mathbf{u}) \xrightarrow{\alpha_1\beta_1^{e_1}\alpha_2\beta_2^{e_2}\alpha_3}_{\mathbb{N}^2} p(\mathbf{v})$，引理得证。 □

7.2.1.4 引理7.9的证明

下面来证明引理7.9，从而完成对定理7.4的全部证明。注意到这一部分的路径上的格局都是有一维受限的，因此直观上来说可以将其想象成一个一维带状态的向量加法系统，从而可以利用上一节的结论来完成上述引理的证明。

首先由引理7.6和定理7.2可知，对于一元更新的 1 维带状态的向量加法系统，可以找到一个线性路径策略集合 S 满足：

- 对于任意的 $\rho \in S$，$|\rho| \leq 10|Q|^4$ 并且 ρ 至多有 1 个圈。

使得 $p(u) \xrightarrow{*} q(v)$ 当且仅当 $p(u) \xrightarrow{S} q(v)$。而对于一般的 1 维带状态的向量加法系统 $V = (Q,T,A)$，我们也有类似的结论，事实上只需要构造一个对应的一元更新的一维带向量的向量加法系统 $V' = (Q',T',\{-1,0,1\})$ 即可。对于 V 中的一个规则 $t = (p,a,q)$，其中不妨令 $a > 0$，构造如下的 $a+2$ 条规则以及添加新的 a 个状态：

$$p \xrightarrow{t_1=(p,0,(t,0))} (t,0) \xrightarrow{t_2=((t,0),1,(t,1))} (t,1)\cdots \xrightarrow{t_{a+1}=((t,a-1),1,(t,a))} (t,a) \xrightarrow{t_{a+2}=((t,a),0,q)} q$$

不难看出 $|T'| \leq |T|\cdot(||T||+2)$, $|Q'| \leq |Q| + |T|\cdot||T||$，并且我们有 $p(u) \xrightarrow{t} q(v)$ 当且仅当 $p(u) \xrightarrow{t_1\ldots t_{a+2}} q(v)$，从而利用上面的结论可以得到关于一般的一元带状态的向量加法系统的路径结构，即：

定理 7.5 令 $V = (Q,T,A)$ 是一个 1 维带状态的向量加法系统，$p(u), q(v)$ 是两个格局。令 S 是一个线性路径策略的集合，满足任意 $\rho \in S$，$|\rho| \leq 10(|Q|+|T|\cdot||T||)^4$，并且 ρ 至多有 1 个圈，则有：

$$p(u) \xrightarrow{*} q(v) \iff p(u) \xrightarrow{S} q(v) \tag{7–26}$$

现在回到引理7.9的证明，证明的关键之处是在于在这种情况下总有一维是永远受限的，所以直觉上来说其可以视作 1 维带状态的向量加法系统，这里需要解决

的问题是 $\mathbb{L}$ 是一个"**L**" 型区域，虽然总有 1 维受限，但是这一维却可以发生变化，所以并不能真正的将其视作一个 1 维带状态的向量加法系统进行考虑。解决方案也很简单，将其拆分成两个区域考虑即可，即 $\mathbb{L}_1 = \mathbb{N} \times [0, D]$ 和 $\mathbb{L}_2 = [0, D] \times \mathbb{N}$；下面将证明 $p(\mathbf{u})$ 到 $q(\mathbf{v})$ 在 $\mathbb{L}$ 上的运行，能够转换为至多只需要考虑两个在 $\mathbb{L}_1$ 或者 $\mathbb{L}_2$ 上的运行。

证明 (引理7.9的证明)　先证明，令 $\mathbb{B} \in \{\mathbb{L}_1, \mathbb{L}_2\}$，则存在一个线性路径策略集合 S 满足：对于任意的 $\rho \in S$ 有 $|S| \leq$ 并且 ρ 至多有一个圈。

事实上不妨令 $\mathbb{B} = \mathbb{L}_1$，定义如下的 1 维带状态的向量加法系统 $V' = (Q', T', A')$ 其中：

- $Q' = \{(q, i) | q \in Q, i \in [D]\}$。
- $T' = \{((p, i), a, (q, j)) | t = (p, (a, j - i), q) \in T\}$。
- $A' = \{i | (i, j) \in A\}$。

对于 V' 中的一条规则 $t = ((p, i), a, (q, j))$，定义 $T(t) \stackrel{\text{def}}{=} (p, (a, j - i), q)$，对于一条路径 $\pi = t_1 \dots t_k$，则 $T(\pi) \stackrel{\text{def}}{=} T(t_1) \dots T(t_k)$。不难验证，在 V 中 $p(\mathbf{u}) \xrightarrow{\pi}_{\mathbb{B}} q(\mathbf{v})$ 当且仅当在 V' 中 $(p, \mathbf{u}[2]) \xrightarrow{T(\pi)} (q, \mathbf{v}[2])$，注意到 $|Q'| = D|Q|$, $|T|' \leq D|T|, ||T'|| \leq |T|$，则由定理7.5可知存在一个线性路径策略 ρ 满足：$|\rho| \leq 10D^4 \cdot (|Q| + |T||T||)^4$ 并且 ρ 至多只有一个圈使得 $p(\mathbf{u}) \xrightarrow{\rho} q(\mathbf{v})$。

回到引理7.9中。令 π 是 $p(\mathbf{u})$ 到 $q(\mathbf{v})$ 在 $\mathbb{L}$ 的一条路径，对其作如下分解：

$$p(\mathbf{u}) \xrightarrow{\pi_1} p_1(\mathbf{u}_1) \xrightarrow{\pi_2} p_2(\mathbf{u}_2) \xrightarrow{\pi_3} \cdots \xrightarrow{\pi_k} p_k(\mathbf{u}_k) \xrightarrow{\pi_{k+1}} q(\mathbf{v}) \tag{7–27}$$

其中满足：

- $\mathbf{u}_i \in [0, D] \times [0, D]$。
- 对于 $i \in [k + 1]$，$p_{i-1}(\mathbf{u}_{i-1}) \xrightarrow{\pi_i} p_i(\mathbf{u}_i)$ 是在 $\mathbb{L}_1$ 或者 $\mathbb{L}_2$ 上的运行。

注意到 $k \leq (D + 1)^2 \cdot |Q|$，并且有：

- 对于 $j \in \{2, \dots, k\}$ 存在一条长度不超过 $10D^4 \cdot |Q|^4$ 的路径 π'_j 使得 $p_{j-1}(\mathbf{u}_{j-1}) \xrightarrow{\pi'_j} p_j(\mathbf{u}_j)$。
- 对于 $j = 1, k + 1$ 存在一个至多只有一个圈并且长度不超过 $10D^4 \cdot (|Q| + |T||T||)^4$ 的线性路径策略 ρ_j 使得 $p_{j-1}(\mathbf{u}_{j-1}) \xrightarrow{\rho_j} p_j(\mathbf{u}_j)$

定义线性路径策略集合 S，S 满足：

- 对于任意的 $\rho \in S$，ρ 至多有两个圈。
- $|\rho| \leq (D+1)^2 \cdot |Q| \cdot 10D^4 \cdot |Q|^4 + 2 \cdot 10D^4 \cdot (|Q| + |T||T||)^4 \leq 20D^6 \cdot (|Q| + |T||T||)^4$。

则有：$p(\mathbf{u}) \xrightarrow{*}_{\mathbb{L}} q(\mathbf{v})$ 当且仅当 $p(\mathbf{u}) \xrightarrow{S} q(\mathbf{v})$，引理得证。 □

至此，我们完成了对定理7.4的证明。

7.2.2 2 维带状态向量加法系统的可达性问题是 PSPACE-完备的

本节将完成 2 维带状态向量加法系统的可达性问题是 **PSPACE**-完备的证明。

PSPACE-难的证明是容易的，回顾章节六中提到的定理6.2，即 B-有界的 1-计数程序 M 的停机问题是 **PSPACE**-完备的，我们可以构造一个 2 为带状态的向量加法系统 V 去模拟该问题；事实上，假设 M 的程序有 n 行，则令 V 有 n 个状态 $\{q_1, \ldots, q_n\}$ 对应每一行程序；对于 M 中的第 k 条命令，若其改变了计数器的值并且改变量为 c，则可以在 V 中添加规则：$t = (q_k, (c, -c), q_{k+1})$；若其是测 0 命令，则添加一个中间状态 q'_k 并添加如下两条规则：$t_1 = (q_k, (0, -B), q'_k)$, $t_2 = (q'_k, (0, B), q_k)$；若其是'goto'命令，则添加对应状态的转移规则，不难验证：

$$(0,1) \xrightarrow{*}_M (0,n) \quad \Longleftrightarrow \quad q_1((0,b)) \xrightarrow{*}_V q_n((0,b))$$

从而有：

引理 7.11 2 维带状态的向量加法系统的可达性问题是 **PSPACE**-难的。

上界的证明则需要用到上一节所证明的 2 维带状态的向量加法系统的可达性路径都能转换成线性路径策略这一结论。事实上，一个关键的发现在于如果可达路径可以用线性路径策略来表示，则不管是几维的向量加法系统，都可以将其是否可达转换成一个线性方程组是否有解；因此直观上来说只需要验证第一个圈和最后一个圈可以运行即可，下面先通过一个简化的版本来理解这一发现。

引理 7.12 (Blondin[77]) 令 $V = (Q, T)$ 是一个 d 维带状态的向量加法系统，$\beta = t_1 \ldots t_m$ 是 V 里 p 上的一个圈，则对于格局 $p(\mathbf{u})$，存在一个线性方程组 $\Xi : \mathbf{a}x \geq \mathbf{c}$ 满足：

- $p(\mathbf{u}) \xrightarrow{*} p(\mathbf{u} + e\Delta(\beta))$ 当且仅当 e 是 Ξ 的一个解。
- $\mathbf{a}, \mathbf{c} \in \mathbb{Z}^{d+1}$ 并且满足：$||a|| \leq |\beta| \cdot ||T||$, $||c|| \leq 2|\beta| \cdot ||T|| + ||u||$。

证明 事实上，$p(\mathbf{u}) \xrightarrow{*} p(\mathbf{u} + e\Delta(\beta))$ 等价于如下方程成立：

$$\forall j \in [m],\ \mathbf{u} + \Delta(\beta[1,j]) \ \geq \ \mathbf{0} \tag{7–28}$$

$$\forall j \in [m],\ \mathbf{u} + (e-1)\Delta(\beta) + \Delta(\beta[1,j]) \ \geq \ \mathbf{0} \tag{7–29}$$

前者保证了第一圈可以运行，后者保证了最后一圈可以运行，中间的圈可以运行是由于可达关系的单调性成立的。上述关系可以转换成一个线性方程组，令 $\mathbf{c}_j = \Delta(\beta[1,j]) + \Delta(\beta) - \mathbf{u}$，则构造 $\Xi : \mathbf{a}x \geq \mathbf{c}$，其中：

- $\mathbf{a} = (0, \Delta(\beta))$。

- $\mathbf{c} = (c_0, c_1, \ldots, c_d)$，其中 $c_0 = -\min_{j\in[m],i\in[d]}(\mathbf{u} + \Delta(\beta[1,j]))[i]$，$c_i = \max_{j\in[m]} \mathbf{c}_j[i]$。

显然 $||a|| \leq |\beta| \cdot ||T||$，$||c|| \leq 2|\beta| \cdot ||T|| + ||u||$，并且 Ξ 有解 e 当且仅当 $p(\mathbf{u}) \xrightarrow{*} p(\mathbf{u} + e\Delta(\beta))$，引理成立。 □

对于一个线性路径策略，我们也只需要检测每个圈运行的第一次和最后一次是否满足要求即可，从而结合定理7.4和推论2.2，不难得到如下引理：

引理 7.13　令 $V = (Q, T)$ 是一个 2 维带状态的向量加法系统，$\rho = \alpha_1\beta_1^* \ldots \alpha_k\beta_k^*\alpha_{k+1}$ 是一个线性路径策略，$p(\mathbf{u})$ 和 $q(\mathbf{v})$ 是 V 上的两个格局，则若 $p(u) \xrightarrow{\rho} q(\mathbf{v})$，则存在 $e_1, \ldots e_k \leq (1 + 2k^2(||v-u|| + |\rho| \cdot ||T||))^{k+1}$ 使得：

$$p(\mathbf{u}) \xrightarrow{\alpha_1\beta_1^{e_1}\ldots\alpha_k\beta_k^{e_k}\alpha_{k+1}}_{\mathbb{N}^2} q(\mathbf{v})$$

证明　我们的目标是构造一个线性方程组 Ξ 满足 $p(\mathbf{u}) \xrightarrow{\alpha_1\beta_1^{e_1}\ldots\alpha_k\beta_k^{e_k}\alpha_{k+1}}_{\mathbb{N}^2} q(\mathbf{v})$ 当且仅当 $\mathbf{e} = (e_1, \ldots e_k)$ 是 Ξ 的一个解。令 $\mathbf{x} = (x_1, \ldots, x_k)$，记 $\mathbf{u}_i = \mathbf{u} + \sum_{j\leq i}\Delta(\alpha_i) + \sum_{j<i} x_j\Delta(\beta_j)$，$\mathbf{v}_i = \mathbf{u}_i + x_i\Delta(\beta_i)$，$p(\mathbf{u})$ 走完 $\alpha_1\beta_1 \ldots \alpha_j$ 的状态为 p_j，则由引理7.12，$p_j(\mathbf{u}_j) \xrightarrow{\beta_j^{x_j}} p_j(\mathbf{v}_j)$ 当且仅当：

$$\begin{aligned} \forall i \in [|\beta_j|],\ \mathbf{u}_j + \Delta(\beta_j[1,i]) &\geq \mathbf{0} \\ \forall i \in [|\beta_j|],\ \mathbf{u}_j + (x_j - 1)\Delta(\beta_j) + \Delta(\beta_j[1,i]) &\geq \mathbf{0} \end{aligned}$$

因此对于 $j \in [k]$，定义对应的线性方程组：$A_j\mathbf{x} \geq \mathbf{c}_j$

- A_j 是一个 $d \times k$ 的矩阵，满足 $A_j = [\Delta(\beta_1), \ldots, \Delta(\beta_j), \mathbf{0}, \ldots, \mathbf{0}]$。
- $\mathbf{c}_j = (c_j^1, \ldots c_j^d)$ 满足：

$$c_j^k = \max_{i\in[|\beta_j|]_0} (\Delta(\beta_j[i+1, |\beta_j|]) - \mathbf{u} - \sum_{k\leq i}\Delta(\alpha_k))[k]$$

再定义矩阵 $A_0 = [\Delta(\beta_1), \ldots, \Delta(\beta_k)]$ 和线性方程组 $\Xi : A\mathbf{x} = \mathbf{c}$，其中：

$$A = \begin{bmatrix} A_0 \\ A_1 \\ \vdots \\ A_k \end{bmatrix}, \qquad \mathbf{c} = \begin{bmatrix} \mathbf{v} - \mathbf{u} - \sum_{i=1}^{k+1}\Delta(\alpha_i) \\ \mathbf{c}_1 \\ \vdots \\ \mathbf{c}_k \end{bmatrix} \tag{7–30}$$

则由引理7.12可知，Ξ 有一组解当且仅当 $p(\mathbf{u}) \xrightarrow{\rho} q(\mathbf{v})$。注意到，$||A|| \leq k \cdot |\rho| \cdot ||T||, ||\mathbf{c}|| \leq ||v-u|| + 2|\rho| \cdot ||T||$，从而由推论2.2，存在一组解 $\mathbf{e}$，满足：$e_i \leq (1 + 2k^2(||v-u|| + |\rho| \cdot ||T||))^{k+1}$，引理得证。 □

由引理7.11和引理7.13不难得到本节的主要结论：

定理 7.6 2 维带状态的向量加法系统的可达性问题是 **PSPACE**-完备的。

证明 由引理7.11只需要给出 **PSPACE** 的算法，给定 $p(\mathbf{u})$ 和 $q(\mathbf{v})$，由定理7.4可知若其存在一条路径，则存在一条满足某个长度不超过 $|\rho| \leq 2^{17}(|Q| \cdot |T| \cdot ||T||)^{15}$ 并且至多有 $6|Q|^2$ 个圈的线性路径策略 ρ 的的路径 π，由引理7.13可知 π 的长度至多是关于 V 指数大的，并且可以在多项式空间内描述，因此只要枚举所有这样的路径在验证即可，从而该算法是在 **PSPACE** 内的。 □

注 若 2 维带状态的向量加法系统是以一进制存储的，则上述猜测过程可以在多项式时间内完成，因此此时的可达性问题是在 **NP** 里的。

7.3 高维向量加法系统可达性的复杂原因探讨

本节对高维 (≥ 3) 的带状态的向量加法系统作一定的讨论。目前来说，高维方面的结论非常的少，首先从低维的角度出发，当维数升上 3 之后，Hopcroft 在 [40] 就给出了一个可达集不是半线性集，如下图所示。在 V_3 中考虑由格局 $p(0,0,1)$ 出发的可达格局，显然对于其所达的格局 $q(\mathbf{v})$ 有着 $\mathbf{v}[3] \leq 2^{\mathbf{v}[1]}$，其不是半线性的，尽管在章节五知道该可达集几乎半线性的，但这一结论也使得对于使用线性路径策略去刻画 3 维以上的带状态的向量加法系统成为了不太现实的方案。

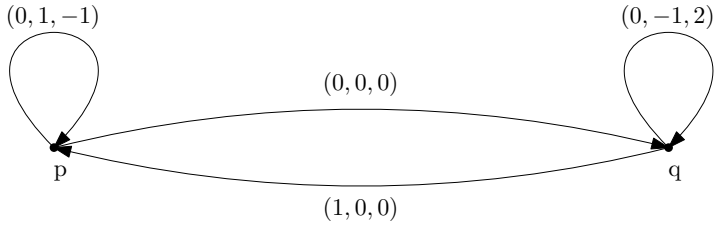


图 7–5 不满足可达集是半线性的 3-VASS V_3

Figure 7–5 An example of 3-VASS V_3 that its reachable set isn't semi-linear

再从另一个角度来讨论一下线性路径策略会遇到的困难。事实上仔细理解上面一节关于 2 维带状态向量加法系统的可达性的证明可以发现，其非常依赖引理7.10，而该引理表述的是对于一个带偏移的锥集和某个象限相交之后的集合可以用有限个只用两个生成向量组成的锥集来表示。做个简单的类比，这种性质更像是向量空间的性质，即 n 维向量空间至多只有 n 个线性无关的向量，也就是其中任何一个向量都可以用特定的 n 个向量表示。

但这一引理在 ≥ 3 的情况下就不成立了。直观上来说，总能拼出一个在对应卦限下的向量，但是当其是 2 维的时候相当于只剩下了一个方向的向量，因此此时判别是简单的，要么该向量不能任意使用只有有限次，要么能任意使用基本上可以作为另一个生成向量；但是当维度升到 3 甚至之上的时候，此时还剩下至少两个方向的向量，而这两个向量不仅会相互制约，还会受到分解出来的向量的影响，从而导致引理的不成立。

我们可以用图7–5作为例子再来说明一下。考虑 V_3 在 $\mathbb{Z}^d$ 上 p 到 p 的会经过 q 的路径，其可以表示为 $L \overset{\text{def}}{=} (1,0,0) + cone(P)$，其中 $P = \{(1,0,0),(0,1,-1),(0,-1,2)\}$。考虑 L 和 $\mathbb{N}^3$ 的交集，自然可以分离出 $(1,0,0)$ 第一个向量，但是由于 $(0,1,0),(0,0,1) \notin L$，因此并不能将其表示为由 3 个在 $\mathbb{N}^3$ 上生成向量张成的锥集，从而引理7.10的分解并不能被扩展，我们不能通过此来获取 3 维带状态的向量加法系统可达路径的信息。

除此之外，近两年在对固定维向量加法系统的研究中，一些研究者也发现了很多低维的向量加法系统有着很长的运行。Czwre'nski 在 [74] 中证明了存在一个 4 维的带状态的向量加法系统，其上面的任何一个运行都有双指数长，下面介绍该例子。

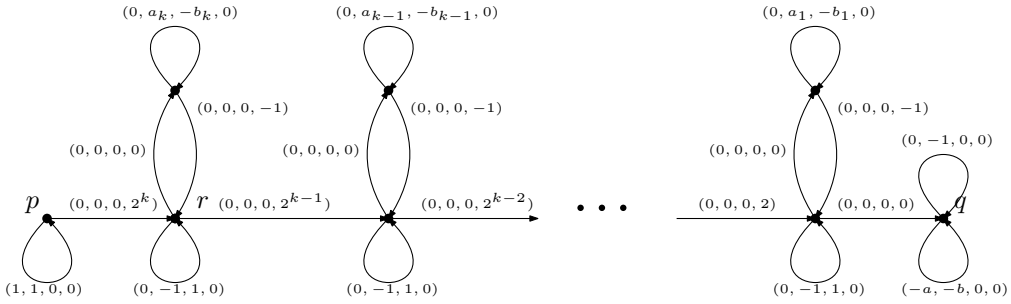


图 7–6 具有双指数路径的 4-VASS V_4

Figure 7–6 An example of 4-VASS V_4 that any halting run has length of doubly exponential

其中 a_k, b_k 按如下定义：

- $a_i = 4^{k(k-i)}(4^k + 2^{k-i})$, $b_i = \Pi_{j=0}^{k-i+1}(4^k + 2^j)$。
- $a = \Pi_{j=0}^{k-1}(4^k + 2^j)$, $b = 4^{2k^2}$。

注意到如下事实：

$$(\frac{a_k}{b_k})^{2^k} \cdot (\frac{a_{k-1}}{b_{k-1}})^{2^{k-1}} \cdot \cdots \cdot (\frac{a_1}{b_1})^{2^1} = \frac{a}{b} \tag{7–31}$$

因此任何从 $p(\mathbf{0})$ 到 $q(\mathbf{0})$ 的一个运行在离开状态 p 时的格局中的向量 $(N,N,0,0)$ 必须满足：$(b_k)^{2^k} | N$，但是 V_4 的大小仅仅是多项式大的，从而该运行至少有一个

双指数长的长度。

而对于章节五所提到的 Presburger 不变量方法，近些年也有悲观的结论。Czwre'nski 在 [155] 中证明一些很长的可达路径也可以导致很大的分离对，即如下引理：

引理 7.14 (Czwre'nski[155]) 令 V 是一个 d 维带状态的向量加法系统，$p(\mathbf{u})$ 和 $q(\mathbf{v})$ 是两个 V 上的格局，令 r 是 V 上的一条状态，$l = \mathbf{a} + \mathbb{N}\Delta$ 是 $\mathbb{N}^d$ 上的一条线满足：

- $\Delta \in \mathbb{N}^2 \times \{0\}^{d-2}$。
- $p(\mathbf{u})$ 和 $q(\mathbf{v})$ 之间不存在路径。
- 对 l 上的任一个点 $\mathbf{c}$ 有 $p(\mathbf{u}) \xrightarrow{*} r(\mathbf{c})$。
- 对于 l 上的任一个点 $\mathbf{c}$ 和任意正整数的 n 有 $r(\mathbf{c} + n\mathbf{e}_2) \xrightarrow{*} q(\mathbf{v})$。

则对于任意一个分离 $p(\mathbf{u})$ 和 $q(\mathbf{v})$ 的分离对都包含周期向量 $k\Delta$，其中 $k \in \mathbb{Q}$。

证明 考虑任何一个 $p(\mathbf{u})$ 和 $q(\mathbf{v})$ 的分离对 $(S, \mathbb{N}^d \setminus S)$，记 $S_r = \{\mathbf{x} | q(\mathbf{x}) \in S\}$，则由定义 S_r 是有限个线性集的并，即 $S_r = \bigcup_i L_i$。由条件存在 L_i 满足：$l \subseteq L_i$，令 $L = \mathbf{b} + cone(\{\mathbf{p}_1, \ldots, \mathbf{p}_n\})$，则对于任意的 $n \in \mathbb{N}$ 存在 $m_1 \ldots m_n$ 满足：

$$\mathbf{a} + n\Delta = \mathbf{b} + \sum_{i=1}^{n} m_i \mathbf{p}_i$$

通过排列 $\mathbf{p}_1 \ldots \mathbf{p}_n$ 的顺序，可以找到 $k \in [n]$ 满足：对任意 $j > k$ 和对任意的 $i \geq 3$ 有 $\mathbf{p}_j[i] > 0$，而对任意 $j \leq k$ 和对任意的 $i \geq 3$ 有 $\mathbf{p}_j[i] = 0$，则可以重写上式为：

$$\mathbf{a} - \mathbf{b} + n\Delta - \sum_{i=k+1}^{n} m_i \mathbf{p}_i = \sum_{i=1}^{k} m_i \mathbf{p}_i \tag{7–32}$$

注意到 $\Delta \in \mathbb{N}^2 \times \{0\}^{d-2}$，从而 $m_{k+1}, \ldots m_n$ 被 $\mathbf{a} - \mathbf{b}$ 的值所限制，并且与 n 无关。因此令 $\mathbf{v} = \mathbf{a} - \mathbf{b} - \sum_{i=k+1}^{n} m_i \mathbf{p}_i$，接下来证明必有一个 $\mathbf{p}_i, i \in [k]$ 满足 $\Delta = k\mathbf{p}_i$，其中 $k \in \mathbb{Q}$。

只需要考虑前两维。如果存在 $\mathbf{p}_i, i \in [k]$ 使得 $\Delta[1]\mathbf{p}_i[2] > \Delta[2]\mathbf{p}_i[1]$, 则考虑 $\Delta[1]\mathbf{p}_i$, 其满足：$\Delta[1]\mathbf{p}_i = k_1\Delta + k_2\mathbf{e}_2$，从而与第四个条件矛盾。因此对于任意的 $\mathbf{p}_i, i \in [k]$ 有：$\Delta[1]\mathbf{p}_i[2] \leq \Delta[2]\mathbf{p}_i[1]$。

如果所有的等号都不成立，则令等式7–32左右两边分别为 M_1, M_2 有：

$$\lim_{n\to\infty} \frac{M_1[2]}{M_1[1]} = \frac{\Delta[2]}{\Delta[1]} > \max_{i\in[k]} \frac{\mathbf{p}_i[2]}{\mathbf{p}_i[1]} = \lim_{n\to\infty} \frac{M_2[2]}{M_2[1]}$$

矛盾！从而必有 $\mathbf{p}_i$ 满足：$\Delta[1]\mathbf{p}_i[2] = \Delta[2]\mathbf{p}_i[1]$，从而引理成立。 □

引理7.14给出了一个分量在分离对中的一个充分条件，我们最后介绍将该引理用到上面所举的 4 维带状态的向量加法系统的例子中，说明很长的运行可以产生很大的分离对。在 V_4 中考虑 $p((0,0,0,0))$ 和 $q((0,1,0,0))$，显然两个格局之间是不可达的，因为对于从 $p((0,0,0,0))$ 出发的到达 $q(\mathbf{v})$ 的任何一个格局中必须满足：$\mathbf{v}[2] \leq \frac{b}{a} \cdot \mathbf{v}[1]$，令 $\Delta = (N, N \cdot (\frac{a_k}{b_k})^{2^k}, 0, 0)$，其中 $N = \Pi_{i=1}^{k} b_i^{2^i}$，则可以验证：

- 对任意 $n \in \mathbb{N}$，$p((0,0,0,0)) \xrightarrow{*} r(n\Delta)$。
- 对任意 $n_1, n_2 \in \mathbb{N}$，$r(n_1\Delta + n_2\mathbf{e}_2) \xrightarrow{*} q((0,1,0,0))$。

从而由引理7.14可知，任何一个分离对都将包含 $r\Delta$ 这样的一个类型，然而描述 Δ 是需要双指数大的空间，从而用 Presburger 不变量方法证明 $p((0,0,0,0))$ 和 $q((0,1,0,0))$ 是不可达的需要考虑一个双指数大的分离对。

7.4　本章小结

本章介绍了固定维向量加法系统可达性问题的一些成果。可以看到在低维的时候由于其可达集有良好的半线性集性质，因此可以分析出其路径的长短获得了一些完备的结论。但是维度 ≥ 3 的时候目前还没有很好的研究手段。

近些年随着对一般维向量加法系统的新的上界算法和新的下界研究成果进行考虑，研究者们找到了高维向量加法系统里存在一些长度很长的运行，这一点在最后一节有所介绍，但值得注意的是尽管找出了这样一些特定的向量加法系统，但并不能说明 4 维带状态的向量加法系统的可达性问题就是 2-**EXPSPACE**-难的，因此对于 ≥ 3 维带状态的向量加法系统的可达性问题，目前还是没有很好的认识。

一个可以考虑的方向是不同维向量加法系统可达性问题之间的联系。从目前的结果上来看，从下界出发似乎是两维可以将复杂性从 $\mathbf{F}_k$ 升至 $\mathbf{F}_{k+1}$；而从上界来看对于 d 维一个在 $\mathbf{F}_t$ 的算法可以对应构造一个 $d+1$ 维在 $\mathbf{F}_{t+1}$ 的算法，两者目前还是有比较大的鸿沟。而在章节四提到的对 Leroux 等人在 [79] 提出的最新的上界算法的另一种理解希望可以对这里产生帮助。

第八章　非确定计算计数

本章从计数的角度来研究进程中的一些问题。从计数的角度上出发，我们可以认识非确定计算 (nondeterministic computation) 的复杂性；比如可以询问：n 个状态的允许与三个信道 a, b, c 交互的进程一共有多少个？这些进程的数量在某种程度上刻画了非确定计算的复杂性。

为了方便研究，本章考虑了进程中的非确定计算对象 (nondeterministic computational object)。简单来说，即是不能与其他进程交互的那些进程；直观上来理解，就是那些仅仅承担计算任务的进程，其能接收一个输入，最后返回一个输出。这类对象能够很好的反映分叉带来的计算复杂性。此外本章仅仅考虑 CCS 上的一个子模型 $\mathbb{CCS}^{\mu}$，即其只允许 τ 动作的存在。Fu 在 [114] 中提出这上面的非确定计算对象有一个好的表示：C-图 (C-graph)，而本章将介绍对其计数方面的结论，即在分支情况下最坏的复杂性结果。

本章的安排如下：第一节将介绍进程的一些基本概念以及介绍 C-图，第二节将介绍本章的核心内容：C-图计数，我们将从高度这个参数给出 C-图个数的公式结论，并且给出一个高度和 C-图顶点个数之间的关系；第三节则将介绍一类特殊的 C-图：正则 C-图；第四节则是本章小结。

8.1　C-图介绍

C-图是 $\mathbb{CCS}^{\mu}$ 上有限状态计算对象的形式化表示，首先来介绍其另一个形式化表示，即使用传统的进程演算的记号。在 $\mathbb{CCS}^{\mu}$ 上，一个有限状态项 (finite state term) 是由下列的 BNF 所生成的：

$$\begin{aligned} T &:= X \mid S \mid \Delta(T) \mid \mu X.T, \\ S &:= \mathbf{0} \mid \tau.T \mid S + S. \end{aligned}$$

其中 X 是一个项变量 (term variable)，S 是一个非确定项 (nondeterministic term)，$\mu X.T$ 是一个递归项 (recursive term)，其中 X 是约束的 (bound)，$\Delta(T)$ 则是一个延迟操作子。一个有限状态计算对象 (finite state computational object) 则是一个不包含自由变量的项，也简称为计算对象。下面给出了 $\mathbb{CCS}^{\mu}$ 的语义：

$$\frac{}{\tau.T \xrightarrow{\tau} T} \qquad \frac{S_i \xrightarrow{\tau} S_i' \quad i \in \{0,1\}}{S_0 + S_1 \xrightarrow{\tau} S_i'}$$

$$\frac{}{\Delta(T) \xrightarrow{\tau} \Delta(T)} \qquad \frac{T \xrightarrow{\tau} T'}{\Delta(T) \xrightarrow{\tau} T'} \qquad \frac{T\{\mu X.T/X\} \xrightarrow{\tau} T'}{\mu X.T \xrightarrow{\tau} T'}$$

其中 $\Delta(T) \xrightarrow{\tau} \Delta(T)$ 称为自循环 (self loop)，$\mathbf{0}$ 是最简单的终止 (terminating) 的计算对象，记 $\Omega = \Delta(\mathbf{0})$，则 Ω 是最简单的发散的 (divergent) 计算对象。特别的，用 $\Rightarrow$ 表示 $\xrightarrow{\tau}$ 的传递闭包。

下面介绍两个计算对象相等的概念。互模拟等价 (bismulation)[62, 162] 是进程演算中一个基本的等价关系。本文采用 [163] 所提出来的关于弱互模拟等价 (weak bismulation) 的一个改良版本-分支互模拟等价关系 (branching bismulation)。

定义 8.1　二元关系 $\mathcal{R}$ 是一个互模拟关系如果下列条件满足：

- 如果 $P\mathcal{R}Q \xrightarrow{\tau} Q'$，则以下之一成立：
  - 存在 $P \Rightarrow P'$，并且有 $P'\mathcal{R}Q, P'\mathcal{R}Q'$。
  - 存在 $P \Rightarrow P''\mathcal{R}Q$ 并且有 $P'' \xrightarrow{\tau} P'\mathcal{R}Q'$。
- 如果 $Q\mathcal{R}^{-1}P \xrightarrow{\tau} P'$，则以下之一成立：
  - 存在 $Q \Rightarrow Q'$，并且有 $Q'\mathcal{R}^{-1}P, Q'\mathcal{R}^{-1}P'$。
  - 存在 $Q \Rightarrow Q''\mathcal{R}^{-1}P$ 并且有 $Q'' \xrightarrow{\tau} Q'\mathcal{R}^{-1}P'$。

上述关系描述了这样一种等价关系：一个进程可以通过作一些不改变自己状态的 τ 动作后去模拟另一个进程，另一个进程也是如此。但是这种等价关系依旧有些缺陷，比如其不能很好的区分一个发散的进程和一个终止的进程。比如可以证明 Ω 和 $\mathbf{0}$ 在上述定义下是互模拟等价的，但这两个项显然应该是不相等的，从而需要定义一种关于发散的关系来区分这种情况。研究者们为此给出了很多的方法[66-67, 164-168]，本文采用 [169] 提出的想法。

定义 8.2　二元关系 R 是共发散的 (codivergent) 如果其满足以下的条件：

- 如果 $P_0\,\mathcal{R}\,Q_0 \xrightarrow{\tau} Q_1 \xrightarrow{\tau} \ldots \xrightarrow{\tau} Q_n \xrightarrow{\tau} \ldots$，则存在 P_1 和 $j > 0$ 使得 $P_0 \overset{\tau}{\Longrightarrow} P_1\,\mathcal{R}\,Q_j$。
- 如果 $P_0\,\mathcal{R}^{-1}\,Q_0 \xrightarrow{\tau} Q_1 \xrightarrow{\tau} \ldots \xrightarrow{\tau} Q_n \xrightarrow{\tau} \ldots$，则存在 P_1 和 $j > 0$ 使得 $P_0 \overset{\tau}{\Longrightarrow} P_1\,\mathcal{R}^{-1}\,Q_j$。

有了上述两个定义，我们可以定义两个计算对象 P, Q 的等价关系 $\simeq$：如果 (P, Q) 是一个共发散的互模拟关系，则称 P 和 Q 是相等的。注意到在 $\mathbb{CCS}^{\mu}$ 中不相等的计算对象是有无穷多个的，Fu 在 [114] 中便构造了无穷多个互不相等的计算对象，如下所示：

$$\begin{aligned} \Upsilon_{2i+1} &= \Delta(\tau.\mathbf{0} + \tau.\Upsilon_{2i}), \\ \Upsilon_{2i+2} &= \tau.\mathbf{0} + \tau.\Omega + \tau.\Upsilon_{2i+1}. \end{aligned}$$

不难验证，对于任意 $i \neq j$，有 $\Upsilon_i \neq \Upsilon_j$。

注 注意到使用弱互模拟关系定义计算对象的相等性是没有意义的，因为在忽略共发散性的情况下在 $\mathbb{CCS}^{\mu}$ 里的所有计算对象都是弱互模拟的。

接下来从另一个角度-图的角度来形式化的介绍计算对象这一概念。将每个计算对象视作一个顶点，则对于一步计算 $P \xrightarrow{\tau} P'$，可以视作 P 到 P' 的一条边，因此关于一个计算对象 P，可以用一个有根节点的有向图去描述，Fu 在 [114] 提出了 D-图 (D-graph) 的概念。

定义 8.3 给定一个有向图 $G = (V, E)$，如果其满足：

- 存在一个特殊的点 c，称为根节点，该节点没有入边。
- G 是简单图，即顶点与顶点之间最多只有一条有向边。
- 对于根节点来说，图中任何一个顶点都是可达的。

则称 G 是一个 D-图 $\mathcal{G} = (G, c)$。

这里会用图论中一些经典的记号。一个自循环 (self-loop) 指的是顶点自己到自己的一条边，一个顶点的出度 (out-degree) 指的是该顶点的出边数量。对于两个顶点 p, q，如果 p 到 q 有一条边，则称 q 是 p 的孩子；如果 p 到 q 有一条路径，则称 p 是 q 的祖先。

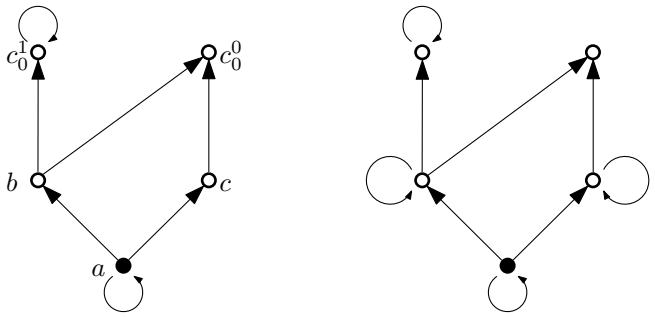


图 8–1 两个 D-图的例子

Figure 8–1 Examples of D-graph

有时我们会用根节点来表示某些 D-图。考虑仅有一个顶点的 D-图，一共有两种，前者只有一个点没有边，后者则有一个自循环的边。为了方便起见，称前者为 c_0^0，后者为 c_0^1。任何一个 D-图都会有至少其中一个上述的导出子图 (induced subgraph)。一个内点 (internal node) 则是指少一条出边到不同顶点的顶点。

图8–1给了两个有关 D-图的例子，其中黑色实心的点分别表示了两张 D-图的根节点。事实上，一个 D-图也是一个有限状态计算对象的表示。在图中，左面

那张 D-图可以理解为 $\Delta(\tau.\tau+\tau.(\tau.\Omega+\tau)))$，因为可以将上面的每一个点进行标记：$a$ 表示为 $\Delta(\tau.\tau+\tau.(\tau.\Omega+\tau)))$，$b$ 表示为 $\tau.\Omega+\tau$，c 表示为 τ，c_0^0 和 c_0^1 则分别为 $\mathbf{0}$ 和 Ω。同样右面那张图可以理解为 $\Delta(\tau.\Delta(\tau)+\tau.\Delta(\Omega+\tau))$。因此可以将上述定义的等价关系运用在 D-图里，比如在左边的 D-图里 c 和 c_0^0 便是相等的，因为 $\{(a,a),(b,b),(c_0^1,c_0^1),(c_0^0,c_0^0),(c,c),(c,c_0^0),(c_0^0,c)\}$ 是一个共发散的互模拟关系。考虑 D-图中那些没有两个顶点相等的图，称为 C-图。

定义 8.4　如果一个 D-图 $\mathcal{G}=(G,c)$ 满足没有任何两个顶点相等，则称 $\mathcal{G}$ 是一个 C-图。

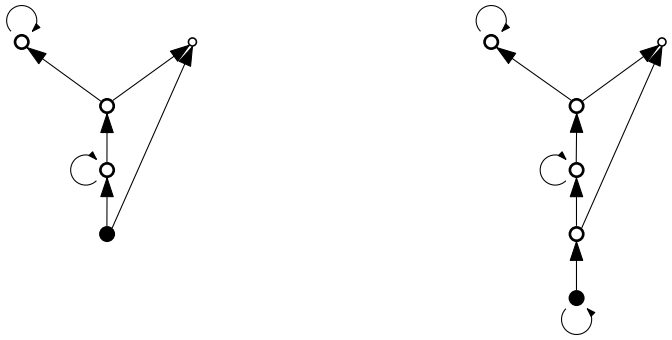


图 8–2　两个 C-图的例子

Figure 8–2　Examples of C-graph

显然对于不同的 C-图，其根节点是不相等的，因此可以用根节点来表示对应的 C-图。在图8–1由上述讨论可知，左面的不是 C-图，而右面的是 C-图。两个单顶点的 D-图都是 C-图，并称 c_0^0 和 c_0^1 为平凡 C-图。事实上，几乎所有的 C-图都包含这两个顶点，这一点可以再通过图8–2来理解。左右两个 C-图的根节点是不相等的，因为左边可以直接到达 c_0^0，但是有边不行，其需要先改变自己到达一个不相等的节点以后才能到达 c_0^0。

注　图8–2其实就是如何用 n 个点构造 $n-2$ 高度的 C-图的例子。

8.2　C-图计数

本节将介绍 C-图的计数。C-图可以表示计算对象，因此 C-图里的一条路径可以想象成一条计算所花费的步数，这里忽略自循环，也即计算所需经历的不同状态数。为此定义给定一个 C-图，对于其中两个不同的顶点 u,v，$dist(u,v)$ 表示 u

到 v 最长的不包含自循环的路径长度，如果不存在路径，则定义为 -1。将任何一个顶点到两个平凡顶点的距离的较大者定义为其的高度，即：

$$\mathfrak{h}(u) \stackrel{\text{def}}{=} \max\{dist(u, c_0^0), dist(u, c_0^1)\} \tag{8–1}$$

定义一个 C-图 $\mathcal{G}$ 的高度 $\mathfrak{h}(\mathcal{G})$ 为其根节点的高度，显然 c_0^0 和 c_0^1 的两个平凡 C-图的高度为 0，图8–3给出了高度为 $0, 1, 2$ 的所有 C-图的例子，这里用 c_i^j 来表示某个 C-图，其中 j 表示该图的高度，i 表示某个排列中的序号。

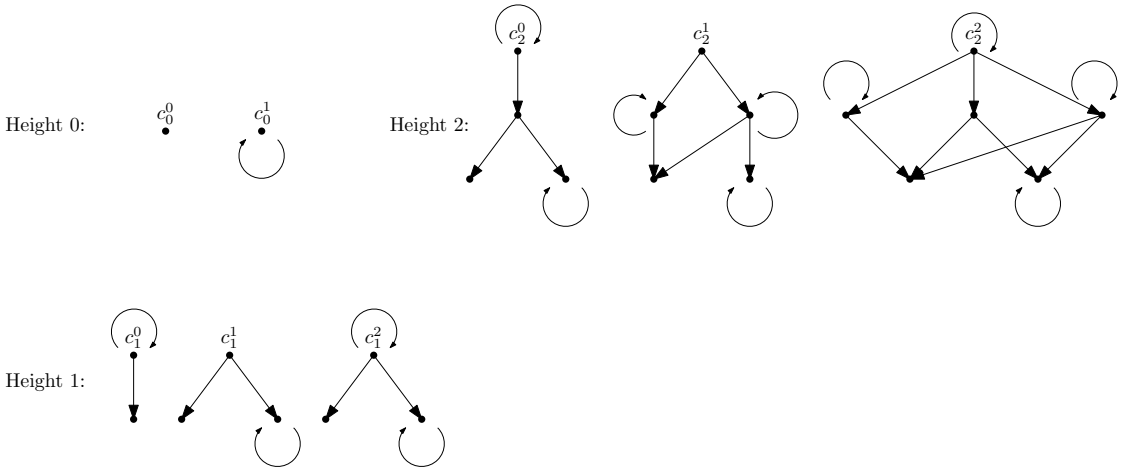


图 8–3 高度为 0，1，2 的 C-图

Figure 8–3 C-graphs of height 0，1，2

从高度这一概念出发，可以将所有的 C-图汇聚成一张无限大的图来进行理解。事实上对于任何一个高度为 $i+1$ 的 C-图来说，其所有的儿子的高度都不会超过 i，并且一定至少有一个儿子高度为 i。从而可以将所有的 C-图都汇聚在一张无限大的按层分类的无限图中，如图8–4所示。图中第 i 层表示所有高度为 i 的节点，可以看到图8–3中的三种高度的 C-图的根节点分别在 $0, 1, 2$ 层。而对于其中任何一个顶点 c_n^k，将其所有可达的点和对应的边组成的图便是以其为根节点的 C-图。因此图8–4中任何一个点都对应了一个不同的 C-图。

进一步观察图8–4，还可以得到如下结论：

- 同一层级的两个点之间不会有边。
- 每个在层级 n 的点至少有一条连接某个 $n-1$ 层级的点。
- 令 a 表示在层级 n 上的一个点，$C(a)$ 表示其所有的儿子，$D(a) \in \{0, 1\}$ 表示其是否有自循环。令 b 是层级 n 上的另一个点，则有 $C(a) \neq C(b)$ 或者 $D(a) \neq D(b)$。

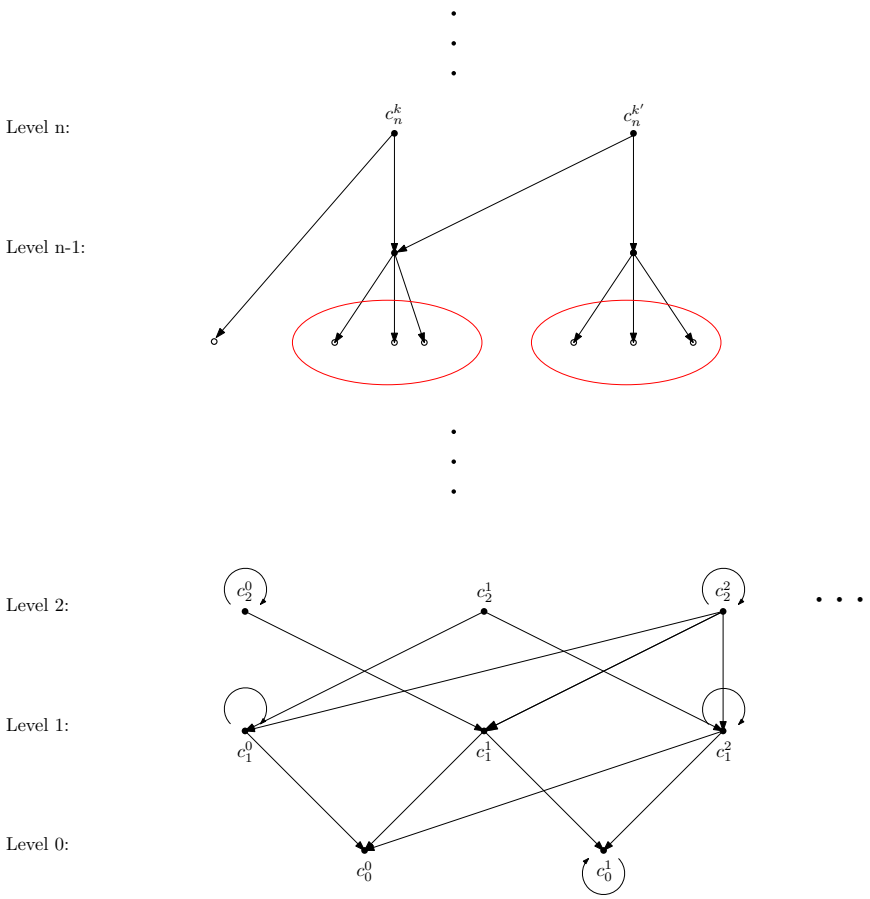


图 8–4　C-图在层级的表示

Figure 8–4　Finite C-graphs Arranged in Levels

- 令 a 是层级 n 上的一个点，a' 是层级 $n-1$ 上的一个点，则有 $C(a) \not\subseteq C(a') \cup \{a'\}$ 或者 $D(a) = 1, D(a') = 0$。

接下来再介绍一些记号。令 $\mathcal{L}_n$ 表示所有在层级 n 上的 C-图的集合，$L_n = |\mathcal{L}_n|$ 表示其大小。令 $\mathcal{L}_n^j$ 表示层级 n 里所有出度为 j 的根节点的 C-图的集合，L_n^j 表示其大小。根据其根节点是否有自循环，再将 $\mathcal{L}_n^j$ 分成了两类 C-图，前者有自循环后者没有，分别记作：$\mathcal{L}_{n,\circlearrowleft}^j$ 和 $\mathcal{L}_{n,\bullet}^j$，用 $L_{n,\circlearrowleft}^j$ 和 $L_{n,\bullet}^j$ 表示其大小。最后令 $\mathcal{S}_n = \bigcup_{i=1}^n \mathcal{L}_i$ 表示所有在层级 n 或者以下的 C-图的集合，并用 S_n 表示其它大小，显然有：

$$\begin{aligned} L_n^j &= L_{n,\bullet}^j + L_{n,\circlearrowleft}^j, \\ L_n &= \sum_{j=1}^{S_{n-1}+1} L_n^j. \end{aligned}$$

注　由定义不难发现，$L_{n,\circlearrowleft}^{S_{n-1}+1} = 1, L_{n,\bullet}^{S_{n-1}+1} = 0, L_{n,\bullet}^{S_{n-1}} = 1.$

8.2.1 按高度计数

本节将完成按高度对 C-图进行计数这一工作，目标是证明如下定理：

定理 8.1 对于 $n \geq 3$，我们有如下等式：

$$S_n = \left(\sum_{k=2}^{n}(-1)^k \frac{k!}{k-1} \cdot k^{S_{n-(k-1)}}\right) - (-1)^n(n-1)! \cdot (n^3+2n^2+n+1).$$

首先回顾一些组合的基本记号，$\binom{n}{k}$ 表示从 n 个东西里取出 k 个东西的不同取法个数，显然：

$$\binom{n}{k} = \begin{cases} \frac{n!}{(n-k)!\cdot k!}, & \text{if } n \geq k, \\ 0, & \text{if } n < k. \end{cases}$$

下面列出了一些基本的在后续证明中会使用到的关于组合的公式：

$$\binom{K}{k-1} + \binom{K}{k} = \binom{K+1}{k}, \tag{8–2}$$

$$\sum_{i=0}^{k}\binom{K-K'}{i}\binom{K'}{k-i} = \binom{K}{k}, \tag{8–3}$$

$$\sum_{i=0}^{k} d^i \binom{k}{i} = (d+1)^k. \tag{8–4}$$

接下来会到定理8.1的证明。考察在 n 层的一个 C-图，其儿子会有两种情况：

- 其有两个或者以上在层级 $n-1$ 的儿子。
- 其只有一个在层级 $n-1$ 的儿子。

本文分别用 $\mathcal{B}_{\_,\_}$ 和 $\mathcal{A}_{\_,\_}$ 表示上述对应的集合，用 $B_{\_,\_}$ 和 $A_{\_,\_}$ 表示其对应的大小，比如 $\mathcal{A}^j_{n,\circlearrowleft}$ 便表示集合 $\mathcal{L}^j_{n,\circlearrowleft}$ 中根节点仅有一个在层级 $n-1$ 的儿子的 C-图的集合，而 $A^j_{n,\circlearrowleft} = |\mathcal{A}^j_{n,\circlearrowleft}|$ 则表示其大小。

现在分别来计算其大小。第一种情况是简单的，因为对于一个在层级 n 有两个以上层级 $n-1$ 的儿子的图来说，该点一定是跟这两个儿子不相等的，因此对于任何一个在层级 n 有 j 个儿子的点来说，只要其随意的选择 $t(\geq 2)$ 在层级 $n-1$ 的儿子和 $j-t-1$ 或者 $j-t$ 个在其他更低层级的儿子即可，从而我们有：

$$B^j_{n,\circlearrowleft} = \sum_{t=2}^{j-1}\binom{L_{n-1}}{t}\binom{S_{n-2}}{j-t-1},\ B^j_{n,\bullet} = \sum_{t=2}^{j}\binom{L_{n-1}}{t}\binom{S_{n-2}}{j-t}. \tag{8–5}$$

经过计算可得：

$$\begin{aligned} B^j_{n,\cup} + B^j_{n,\bullet} &= \sum_{t=2}^{j-1} \binom{L_{n-1}}{t}\binom{S_{n-2}}{j-t-1} + \sum_{t=2}^{j} \binom{L_{n-1}}{t}\binom{S_{n-2}}{j-t} \\ &\overset{(8--3)}{=} \binom{S_{n-1}}{j-1} - L_{n-1}\binom{S_{n-2}}{j-2} - \binom{S_{n-2}}{j-1} + \binom{S_{n-1}}{j} - L_{n-1}\binom{S_{n-2}}{j-1} - \binom{S_{n-2}}{j} \\ &\overset{(8--2)}{=} \binom{S_{n-1}+1}{j} - \binom{S_{n-2}+1}{j} - L_{n-1}\cdot\binom{S_{n-2}+1}{j-1}. \end{aligned} \tag{8–6}$$

注　上述等式8–6也有一个组合的解释，事实上考虑一个出度为 j 的点，其 j 条边可以 $\mathcal{S}_{n-1}$ 和自循环中进行选择，所以一共会有 $\binom{S_{n-1}+1}{j}$ 种不同的情况，此外还需要去除其中不在 $\mathcal{B}^j_n$ 的情形，一共有两种情况：

- 这 j 个儿子没有一个在层级 $n-1$ 中，这种情况一共有 $\binom{S_{n-2}+1}{j}$ 个。
- 这 j 个儿子只有一个在层级 $n-1$ 中，这种情况一共有 $L_{n-1}\cdot\binom{S_{n-2}+1}{j-1}$ 个。

回到第二种情况，有如下引理：

引理 8.1　如下关系成立：

1. $A^j_{n,\cup} = \sum_{i=1}^{S_{n-2}+1}\left(L^i_{n-1,\bullet}\cdot\binom{S_{n-2}}{j-2} + \sum_{t=1}^{j-2} L^i_{n-1,\cup}\cdot\binom{S_{n-2}-i+1}{t}\cdot\binom{i-1}{j-2-t}\right)$.
2. $A^j_{n,\bullet} = \sum_{i=1}^{S_{n-2}+1}\sum_{t=1}^{j-1}\left(L^i_{n-1,\bullet}\cdot\binom{S_{n-2}-i}{t}\cdot\binom{i}{j-1-t} + L^i_{n-1,\cup}\cdot\binom{S_{n-2}-i+1}{t}\cdot\binom{i-1}{j-1-t}\right)$.

证明　只证明第一个等式，第二个的证明是类似的。固定 j,i，考虑在 $\mathcal{A}^j_{n,\cup}$ 中的一个 C-图的根节点 c_n，其有两种情况：

- c_n 连接了某个在 $\mathcal{L}^i_{n-1,\bullet}$ 的点 c_{n-1}，注意到此时 c_n 是有自循环而 c_{n-1} 没有，因此 $c_n \neq c_{n-1}$，因此只要从 $\mathcal{S}_{n-2}$ 里再任选 $j-2$ 个节点作为 c_n 的儿子即可，这里一共有：$L^i_{n-1,\bullet}\cdot\binom{S_{n-2}}{j-2}$ 个。
- c_n 连接了某个在 $\mathcal{L}^i_{n-1,\cup}$ 的点 c_{n-1}，令 c_{n-1} 有 i 个儿子，为了使得 $c_n \neq c_{n-1}$，c_n 必须在 $\mathcal{S}_{n-2}$ 找到一个不属于 c_{n-1} 的儿子，这里还需要忽略自循环这条边，从而一共有：$\sum_{t=1}^{j-2} L^i_{n-1,\cup}\binom{S_{n-2}-i+1}{t}\binom{i-1}{j-2-t}$ 个。 □

有了引理8.1，便可以计算 $\mathcal{A}_n$ 的大小：

$$\begin{aligned} A^j_{n,\cup} + A^j_{n,\bullet} &= \sum_{i=1}^{S_{n-2}+1}\left(L^i_{n-1,\bullet}\cdot\binom{S_{n-2}}{j-2} + \sum_{t=1}^{j-2} L^i_{n-1,\cup}\cdot\binom{S_{n-2}-i+1}{t}\cdot\binom{i-1}{j-2-t}\right) + \\ &\quad \sum_{i=1}^{S_{n-2}+1}\sum_{t=1}^{j-1}\left(L^i_{n-1,\bullet}\cdot\binom{S_{n-2}-i}{t}\cdot\binom{i}{j-1-t} + L^i_{n-1,\cup}\cdot\binom{S_{n-2}-i+1}{t}\cdot\binom{i-1}{j-1-t}\right). \end{aligned}$$

考虑其中 $L^i_{n-1,\cup}$ 的系数：

$$\binom{S_{n-2}}{j-2}+\sum_{t=1}^{j-1}\binom{S_{n-2}-i}{t}\binom{i}{j-1-t} \overset{(8-3)}{=} \binom{S_{n-2}}{j-2}+\binom{S_{n-2}}{j-1}-\binom{i}{j-1} \overset{(8-2)}{=} \binom{S_{n-2}+1}{j-1}-\binom{i}{j-1},$$

考虑其中 $L^i_{n-1,\bullet}$ 的系数：

$$\begin{aligned}
&\sum_{t=1}^{j-2}\binom{S_{n-2}-i+1}{t}\binom{i-1}{j-2-t}+\sum_{t=1}^{j-1}\binom{S_{n-2}-i+1}{t}\binom{i-1}{j-1-t}\\
\overset{(8-3)}{=}\ &\binom{S_{n-2}}{j-2}-\binom{i-1}{j-2}+\binom{S_{n-2}}{j-1}-\binom{i-1}{j-1}\\
\overset{(8-2)}{=}\ &\binom{S_{n-2}+1}{j-1}-\binom{i}{j-1}.
\end{aligned}$$

这两者的系数相等，从而可以得到：

$$A^j_n=\sum_{i=1}^{S_{n-2}+1}L^i_{n-1}\cdot\left(\binom{S_{n-2}+1}{j-1}-\binom{i}{j-1}\right)=L_{n-1}\cdot\binom{S_{n-2}+1}{j-1}-\sum_{i=1}^{S_{n-2}+1}L^i_{n-1}\cdot\binom{i}{j-1}$$

注 上述等式同样有组合的解释，注意到对于每个在层级 $n-1$ 的点 c_{n-1}，其可以生成 $\binom{S_{n-2}}{j-2}+\binom{S_{n-2}}{j-1}=\binom{S_{n-2}+1}{j-1}$ 不同的高度为 n 的 D-图，因此只要去除其中不是 C-图的点即可，这里同样有两种情况：

- 如果 c_{n-1} 不存在自循环并且其出度为 i，则新产生的节点 c_n 满足 $c_n \simeq c_{n-1}$ 当且仅当 c_n 的其他 $j-1$ 个儿子也都是 c_{n-1} 的儿子，这里一共有：$\sum_{i=1}^{S_{n-2}+1}L^i_{n-1,\bullet}\cdot\binom{i}{j-1}$ 种。
- 如果 c_{n-1} 存在自循环并且其出度为 i 则新产生的节点 c_n 满足 $c_n \simeq c_{n-1}$ 当且仅当如下之一情况发生：
  - c_n 有自循环并且剩下的 $j-2$ 个儿子也是 c_{n-1} 的儿子，这里一共有 $\sum_{i=1}^{S_{n-2}+1}L^i_{n-1,\cup}\cdot\binom{i-1}{j-2}$ 种。
  - c_n 没有循环并且剩下的 $j-1$ 个儿子也是 c_{n-1} 的儿子，这里一共有 $\sum_{i=1}^{S_{n-2}+1}L^i_{n-1,\cup}\cdot\binom{i-1}{j-1}$ 种。

  因此这种情况下一共有：$\sum_{i=1}^{S_{n-2}+1}L^i_{n-1,\cup}\cdot\binom{i}{j-1}$ 种。

回到定理8.1的证明，由上述讨论，可以计算出 L^j_n 的大小，即：

$$\begin{aligned}
L^j_n &= L^j_{n,\cup}+L^j_{n,\bullet}\\
&= A^j_{n,\cup}+B^j_{n,\cup}+A^j_{n,\bullet}+B^j_{n,\bullet}\\
&= \binom{S_{n-1}+1}{j}-\binom{S_{n-2}+1}{j}-\sum_{i=1}^{S_{n-2}+1}L^i_{n-1}\cdot\binom{i}{j-1}. \qquad (8\text{–}7)
\end{aligned}$$

对 j 求和，则可以计算 L_n：

$$\begin{aligned} L_n &= \sum_{j=1}^{S_{n-1}+1} L_n^j \qquad (8\text{–}8)\\ &= \sum_{j=1}^{S_{n-1}+1}\left(\binom{S_{n-1}+1}{j}-\binom{S_{n-2}+1}{j}-\sum_{i=1}^{S_{n-2}+1} L_{n-1}^i\cdot\binom{i}{j-1}\right)\\ &= \sum_{j=0}^{S_{n-1}+1}\binom{S_{n-1}+1}{j}-\sum_{j=0}^{S_{n-2}+1}\binom{S_{n-2}+1}{j}-\sum_{j=1}^{S_{n-2}+1}\sum_{i=1}^{S_{n-2}+1} L_{n-1}^i\cdot\binom{i}{j-1}\\ &\overset{(8\text{–}4)}{=} 2^{S_{n-1}+1}-2^{S_{n-2}+1}-\sum_{i=1}^{S_{n-2}+1}\left(L_{n-1}^i\cdot\sum_{j=1}^{S_{n-2}+1}\binom{i}{j-1}\right)\\ &\overset{(8\text{–}4)}{=} 2^{S_{n-1}+1}-2^{S_{n-2}+1}-\sum_{i=1}^{S_{n-2}+1} 2^i L_{n-1}^i. \qquad (8\text{–}9) \end{aligned}$$

注意到重复利用等式8–7，可以得到：

$$\begin{aligned} \sum_{i=1}^{S_{m-1}+1} k^i L_m^i &\overset{(8\text{–}7)}{=} \sum_{i=1}^{S_{m-1}+1} k^i\left(\binom{S_{m-1}+1}{i}-\binom{S_{m-2}+1}{i}-\sum_{t=1}^{S_{m-2}+1} L_{m-1}^t\binom{t}{i-1}\right)\\ &= \sum_{i=0}^{S_{m-1}+1} k^i\binom{S_{m-1}+1}{i}-\sum_{i=0}^{S_{m-2}+1} k^i\binom{S_{m-2}+1}{i}-\sum_{i=1}^{S_{m-1}+1} k^i\sum_{t=1}^{S_{m-2}+1} L_{m-1}^t\binom{t}{i-1}\\ &\overset{(8\text{–}4)}{=} (k+1)^{S_{m-1}+1}-(k+1)^{S_{m-2}+1}-k\sum_{t=1}^{S_{m-2}+1} L_{m-1}^t\sum_{i=1}^{S_{m-1}+1} k^{i-1}\binom{t}{i-1}\\ &= (k+1)^{S_{m-1}+1}-(k+1)^{S_{m-2}+1}-k\sum_{t=1}^{S_{m-2}+1} L_{m-1}^t\sum_{i=0}^{t} k^i\binom{t}{i}\\ &\overset{(8\text{–}4)}{=} (k+1)^{S_{m-1}+1}-(k+1)^{S_{m-2}+1}-k\sum_{t=1}^{S_{m-2}+1} L_{m-1}^t(k+1)^t. \end{aligned}$$

上式两边乘以 $(k-1)!$ 可以得到：

$$(k-1)!\sum_{i=1}^{S_{m-1}+1} k^i L_m^i=\frac{(k+1)!}{k}((k+1)^{S_{m-1}}-(k+1)^{S_{m-2}})-k!\sum_{t=1}^{S_{m-2}+1}(k+1)^t L_{m-1}^t. \qquad (8\text{–}10)$$

注意到 $L_0=2, L_1=3, L_1^1=0, L_1^2=2, L_1^3=1$，可以计算出：

$$\begin{aligned} L_n &= 2^{S_{n-1}+1}-2^{S_{n-2}+1}-\sum_{j=1}^{S_{n-2}+1} 2^j L_{n-1}^j.\\ &= 2^{S_{n-1}+1}-2^{S_{n-2}+1}-\left(3^{S_{n-2}+1}-3^{S_{n-3}+1}-2\sum_{j=1}^{S_{n-3}+1} 3^j L_{n-2}^j\right) \end{aligned}$$

$$
\begin{aligned}
&= \cdots \\
&= \frac{2!}{1}(2^{S_{n-1}} - 2^{S_{n-2}}) - \frac{3!}{2}(3^{S_{n-2}} - 3^{S_{n-3}}) + \cdots + (-1)^n(n-2)!\left(n^{S_1+1} - n^{S_0+1} - (n-1)\sum_{t=1}^{S_0+1} n^t L_1^t\right) \\
&= \sum_{k=2}^{n}\left((-1)^k \frac{k!}{k-1}(k^{S_{n+1-k}} - k^{S_{n-k}})\right) + (-1)^{n+1}(n-1)!\sum_{t=1}^{3} n^t L_1^t \\
&= \sum_{k=2}^{n}\left((-1)^k \frac{k!}{k-1}(k^{S_{n+1-k}} - k^{S_{n-k}})\right) - (-1)^n n!(n^2+2n).
\end{aligned}
$$

从而对 n 求和，可以计算出 S_n：

$$
S_n = \sum_{i=0}^{n} L_i = L_0 + L_1 + \sum_{i=2}^{n} L_i = 5 + \sum_{i=2}^{n}\left(\sum_{k=2}^{i}(-1)^k \frac{k!}{k-1}(k^{S_{i+1-k}} - k^{S_{i-k}}) + (-1)^{i+1}(i-1)!(2i^2+i^3)\right).
$$

注意到：

$$
\sum_{i=2}^{n}\sum_{k=2}^{i}(-1)^k \frac{k!}{k-1}(k^{S_{i+1-k}} - k^{S_{i-k}}) = \sum_{k=2}^{n}(-1)^k \frac{k!}{k-1}\sum_{i=k}^{n}(k^{S_{i+1-k}} - k^{S_{i-k}}) = \sum_{k=2}^{n}(-1)^k \frac{k!}{k-1}(k^{S_{n+1-k}} - k^{S_0}).
$$

因此有：

$$
\begin{aligned}
S_n &= 5 + \sum_{k=2}^{n}(-1)^k \frac{k!}{k-1}(k^{S_{n+1-k}} - k^{S_0}) + \sum_{i=2}^{n}(-1)^{i+1}(i-1)!(2i^2+i^3) \\
&= 5 + \sum_{k=2}^{n}(-1)^k \frac{k!}{k-1}k^{S_{n+1-k}} - \sum_{k=2}^{n}(-1)^k \frac{k!}{k-1}k^2 + \sum_{k=2}^{n}(-1)^{k+1}(k-1)!(2k^2+k^3) \\
&= 5 + \sum_{k=2}^{n}(-1)^k \frac{k!}{k-1}k^{S_{n+1-k}} + \sum_{k=2}^{n}\left((-1)^{k+1}(k-1)!(2k^2+k^3) - (-1)^k \frac{k!}{k-1}k^2\right) \\
&= \sum_{k=2}^{n}(-1)^k \frac{k!}{k-1}\cdot k^{S_{n-(k-1)}} + \left(\sum_{k=2}^{n}(-1)^{k-1}(k-2)!(k^4+2k^3-2k^2)\right) + 5. \quad (8\text{–}11)
\end{aligned}
$$

最后来化简8–11中的第二个求和符号。定义函数：

$$
\mathsf{af}(n) = \sum_{k=1}^{n}(-1)^{n-k}k!.
$$

，注意到上述函数满足：$\mathsf{af}(n) = n! - \mathsf{af}(n-1)$，下面用 $\mathsf{af}(n)$ 来化简 $\sum_{k=2}^{n}(-1)^{k-1}(k-$

$2)!(k^4+2k^3-2k^2$，记该式为 A，则有：

$$\begin{aligned}
A &= \sum_{k=2}^{n}(-1)^{k-1}(k+2)! - \sum_{k=2}^{n}(-1)^{k-1}k! + \sum_{k=2}^{n}(-1)^{k-1}(k-1)! + \sum_{k=2}^{n}(-1)^{k-1}(k-2)! \\
&= \sum_{k=1}^{n}(-1)^{k-1}(k+2)! + \sum_{k=1}^{n}(-1)^{k}k! + \sum_{k=1}^{n-1}(-1)^{k}k! - \sum_{k=1}^{n-2}(-1)^{k}k! - 6 \\
&= (-1)^{n+1}\sum_{k=1}^{n+2}(-1)^{n+2-k}k! + (-1)^{n}\sum_{k=1}^{n}(-1)^{n-k}k! + (-1)^{n-1}(n-1)! - 5 \\
&= (-1)^{n+1}\left(\mathsf{af}(n+2) - \mathsf{af}(n) + (n-1)!\right) - 5 \\
&= (-1)^{n+1}\left((n+2)! - (n+1)! + (n-1)!\right) - 5 \\
&= (-1)^{n+1}(n-1)!(n^3+2n^2+n+1) - 5.
\end{aligned}$$

将上式代回等式8–11可得：

$$S_n = \left(\sum_{k=2}^{n}(-1)^k\frac{k!}{k-1}\cdot k^{S_{n-(k-1)}}\right) - (-1)^n(n-1)!\cdot(n^3+2n^2+n+1).$$

即完成了对定理8.1的证明。

8.2.2　高度和顶点个数的关系

上一节给出了 S_n 和 L_n 的递推表达式，而在这一节将讨论 L_n 之间的关系，以及高度和顶点个数之间的关系。

首先来关注 L_n 之间的关系。显然 L_n 至少有 $\mathcal{L}_{n-1}$ 中大小 ≥ 2 的子集数量那么多，即 $L_n \geq 2^{L_{n-1}} - L_{n-1} - 1$，因此 L_n 的大小是非初等的，而下面的定理作了更精确的刻画，其说明 L_n 可以被 $2^{L_{n-1}}L_{n-1}$ 刻画，即 $L_n = \Theta(2^{L_{n-1}}L_{n-1})$。

定理 8.2　对于 $n \geq 2$，L_n 和 L_{n-1} 满足下列不等式：

$$2^{L_{n-1}}L_{n-1} < L_n < 2^{L_{n-1}}\left(L_{n-1} + 4\log^2(L_{n-1})\right).$$

证明　$n = 2$ 时直接验证可知不等式成立，下面考虑 $n \geq 3$ 的情况，由不等式 $L_n < 2^{S_{n-1}+1}$ 及等式 $S_{n-1} - S_{n-2} = L_{n-1}$ 可知：

$$\begin{aligned}
L_n &\overset{(8\text{--}11)}{=} 2^{S_{n-1}+1} - 2^{S_{n-2}+1} - \sum_{i=1}^{S_{n-2}+1}2^i L_{n-1}^i \;>\; 2^{S_{n-1}+1} - 2^{S_{n-2}+1} - L_{n-1}2^{S_{n-2}+1} \\
&= 2^{S_{n-2}+1}(2^{L_{n-1}} - L_{n-1} - 1) \;\geq\; (L_{n-1}+1)(2^{L_{n-1}} - L_{n-1} - 1) \\
&\geq L_{n-1}2^{L_{n-1}}.
\end{aligned}$$

为了证明不等式的另一端，考虑 L_n 与 L_{n-1} 的商有：

$$\begin{aligned}\frac{L_n}{L_{n-1}} &= \frac{2^{S_{n-1}+1}-2^{S_{n-2}+1}-\sum_{i=1}^{S_{n-2}+1}2^i L_{n-1}^i}{2^{S_{n-2}+1}-2^{S_{n-3}+1}-\sum_{i=1}^{S_{n-3}+1}2^i L_{n-2}^i}\\ &< \frac{2^{S_{n-1}+1}}{2^{S_{n-2}+1}-2^{S_{n-3}+1}-L_{n-2}2^{S_{n-3}+1}}\\ &= 2^{L_{n-1}}\left(1+\frac{L_{n-2}+1}{2^{L_{n-2}}-L_{n-2}-1}\right)\\ &\le 2^{L_{n-1}}\left(1+\frac{2L_{n-2}+2}{2^{L_{n-2}}}\right).\end{aligned}$$

注意到 $2L_{n-2} > 2^{S_{n-3}+1}$，所以有：$L_{n-1} \le 2^{S_{n-2}+1} = 2^{S_{n-3}+1}\cdot 2^{L_{n-2}} \le 2L_{n-2}2^{L_{n-2}}$，从而：

$$\frac{L_n}{2^{L_{n-1}}L_{n-1}} \le 1+\frac{2L_{n-2}+2}{\frac{L_{n-1}}{2L_{n-2}}} = 1+\frac{4L_{n-2}(L_{n-2}+1)}{L_{n-1}} \le 1+\frac{4\log^2(L_{n-1})}{L_{n-1}}$$

定理得证。 □

接下来讨论一些高度与节点个数的关系。一个很自然的问题是，给定一个固定高度的 C-图，其最多可以有多少个节点，最少又必须要有多少个节点？也可以反过来问，给定一个固定大小的 C-图，其高度最多是多少，最少又是多少？回顾一下图8–2，其构造实际上给出了上述问题的一些直观，即 n 个节点的 C-图可以有 $n-2$ 的高度。而下述定理，严格的给出来关于高度和大小之间的关系，其将指出尽管有些 C-图的高度和其大小相差比较小，但是也有 C-图可以有很大的大小但是却只有很小的高度。为了方便叙述，用 $\mathfrak{s}(\mathcal{G})$ 来表示一个 C-图的顶点个数，并且令 $\log^*(n)$ 表示最小的 k 使其满足：$n \le 2^{\left.\cdots^{2}\right\}k}$。

定理 8.3 下述命题成立：

1. 对于 $n \ge 3$，给定一个 n 个点的 C-图 $\mathcal{G}_n$ 有：

$$\log^*(n)-1 \le \mathfrak{h}(\mathcal{G}_n) \le n-2. \tag{8–12}$$

2. 对于 $h \ge 1$，给定一个高度为 h 的 C-图 $\mathcal{G}_h$ 有：

$$h+2 \le \mathfrak{s}(\mathcal{G}_h) \le 1+S_{h-1}. \tag{8–13}$$

证明 $\mathfrak{h}(G) \le n-2$ 是显然的，因为对于 $n \ge 3$，$\mathcal{G}_n$ 必然包含两个高度为 0 的平凡 C-图，同时注意到图8–2中给出了高度为 $n-2$ 的 $\mathcal{G}_n$ 构造，因此不等式8–12的右边和不等式8–13的左边都是成立的，并且等号可以取到。

不等式8–13的右边显然也是成立的，因为一个高度为 h 的节点可以将所有高度不超过 $h-1$ 的节点作为自己的儿子，从而有 $\mathfrak{s}(\mathcal{G}_h) \leq 1 + S_{h-1}$。

最后来完成对 $\log^*(n) - 1 \leq \mathfrak{h}(\mathcal{G}_n)$ 的证明。$n = 3$ 时直接验证可知，等号也是可以成立的从而时紧的。考虑 $n \geq 4$ 的情况，注意到如下两个事实：

- 如果 $n \geq 2 + S_{h-2}$ 则 $\mathcal{G}_n$ 的高度至少为 h。
- 如果 $\mathcal{G}_n$ 的高度为 $h-1$，则 n 至多为 $1 + S_{h-2}$。

下面证明：

$$2^{S_{h-1}} < L_h < 2 \cdot 2^{S_{h-1}}. \tag{8–14}$$

事实上不等式的右边由等式 8–11 立马可得。对于第一个不等号，注意到

$$2^{S_{h-2}+1} + \sum_{j=1}^{S_{h-2}+1} 2^j L_{h-1}^j \leq 2^{S_{h-2}+1} + 2^{S_{h-2}+1} \sum_{j=1}^{S_{h-2}+1} L_{h-1}^j \leq 2^{S_{h-2}}(2 + 2L_{h-1}) \leq 2^{S_{h-2}} \cdot 2^{L_{h-1}} = 2^{S_{h-1}}$$

再次使用等式 8–11 可得 $L_h > 2^{S_{h-1}+1} - 2^{S_{h-1}} = 2^{S_{h-1}}$。接下来只要证明当 $h = \log^*(n) - 1$ 时有 $2^{n-2} \geq L_{h-1}$ 即可。这是因为由8–14有：$2^{n-2} \geq L_{h-1} \geq 2^{S_{h-2}}$，从而 $n - 2 \geq S_{h-2}$。

回到不等式 $2^{n-2} \geq L_{h-1}$，下面证明当 $h \geq 2$ 时由如下不等式成立：

$$2^{\left.\cdots^{2}\right\}h} \leq L_h \leq \frac{1}{4} \cdot 2^{\left.\cdots^{2}\right\}h+2}. \tag{8–15}$$

左边的不等式时显然的，考虑右边的不等式，$h = 2$ 时，有 $L_2 = 40 < \frac{1}{4}2^{2^{2^2}}$。由定理8.2通过归纳可知：

$$\begin{aligned} L_h &\leq 2^{L_{h-1}}(L_{h-1} + 4\log^2(L_{h-1})) \\ &\leq 2^{2L_{h-1}} \\ &\leq 2^{\frac{1}{2} \cdot 2^{\left.\cdots^{2}\right\}h+1}} \\ &\leq \frac{1}{4} \cdot 2^{\left.\cdots^{2}\right\}h+2}, \end{aligned}$$

这里最后一步成立是由于当 $x \geq 6$ 时有 $2^{\frac{1}{2}x} \leq \frac{1}{4}2^x$ 并且 $L_2 = 40 > 6$，从而不等式 $\log^*(n) - 1 \leq \mathfrak{h}(\mathcal{G}_n)$ 成立，定理得证。 □

8.3　正则 C-图

本节介绍一类特殊的 C-图，正则 C-图。考虑一个计算对象，如果其每步的选择是有限的，这样的计算对象显然更加方便研究，比如在定义非确定性图灵机时，可以令每一步其只有两个选择。将这个想法形式化到 C-图中，则有如下的定义：

定义 8.5 如果一个 C-图 $\mathcal{G}$ 满足其所有的内点的出度最多是 k，则称 $\mathcal{G}$ 是 k-正则的。

在 k-正则的 C-图中，2-正则的 C-图是最简单的正则 C-图。下面先对 2-正则的 C-图的个数作一定的探讨。令 $\mathcal{K}_n$ 表示 $\mathcal{L}_n$ 中所有的 2-正则 C-图，$K_n = |\mathcal{K}_n|$ 表示其大小，令 $\mathcal{T}_n$ 表示所有不超过层级 n 的 2-正则 C-图，即 $\mathcal{T}_n = \bigcup_{i=0}^{n} \mathcal{K}_i$，并且令 T_n 来表示其大小。由8–7可知，当 $n \geq 2$ 时有：

$$K_n = \binom{T_{n-1}+1}{2} - \binom{T_{n-2}+1}{2} - 2K_{n-1}.$$

注意到等式右边可以重新写成如下的形式：

$$\begin{aligned}\binom{T_{n-1}+1}{2} - \binom{T_{n-2}+1}{2} - 2K_{n-1} &= \frac{1}{2}\left((T_{n-1}+1)T_{n-1}\right) - \frac{1}{2}\left((T_{n-2}+1)T_{n-2}\right) - 2K_{n-1} \\ &= \frac{1}{2}(T_{n-1}+T_{n-2}+1)(T_{n-1}-T_{n-2}) - 2K_{n-1} \\ &= \frac{1}{2}(T_{n-1}+T_{n-2}+1)K_{n-1} - 2K_{n-1}.\end{aligned}$$

因此有：$\frac{2K_n}{K_{n-1}} + 3 = T_{n-1} + T_{n-2}$ 和 $\frac{2K_{n-1}}{K_{n-2}} + 3 = T_{n-2} + T_{n-3}$。整理可得：$\frac{2K_n}{K_{n-1}} = K_{n-1} + K_{n-2} + \frac{2K_{n-1}}{K_{n-2}}$，从而获得了如下的关于 K_n 的递推关系式：

引理 8.2 对于 $n \geq 3$，有如下等式：

$$K_n = \frac{K_{n-1}^2}{2} + \frac{K_{n-1}K_{n-2}}{2} + \frac{K_{n-1}^2}{K_{n-2}}.$$

接下来估计 K_n 的增长速度，下面将证明 K_n 关于 n 是指数大的。注意到 $K_0 = K_1 = 2, K_2 = 3, K_3 = 12$，从而对于 $n \geq 3$ 有：$\frac{K_{n-1}^2}{K_{n-2}} \leq \frac{K_{n-1}^2}{3}$。另一方面当 $n \geq 4$ 时有：$K_n > \frac{K_{n-1}^2}{2} > \frac{12K_{n-1}}{2} > 3K_{n-1}$，结合 $K_3 > 3K_2$ 可得当 $n \geq 4$ 时有 $3K_{n-1}K_{n-2} \leq K_{n-1}^2$，从而有 $\frac{K_{n-1}K_{n-2}}{2} + \frac{K_{n-1}^2}{K_{n-2}} \leq \frac{K_{n-1}^2}{2}$，即如下引理：

引理 8.3 当 $n \geq 4$ 时 K_n 和 K_{n-1} 有如下关系：

$$\frac{K_{n-1}^2}{2} \leq K_n \leq K_{n-1}^2$$

回过头来考虑一般的 k-正则 C-图。令 $\mathcal{K}_n^k$ 表示 $\mathcal{L}_n$ 中所有 k-正则 C-图，K_n^k 表示其大小，令 $\mathcal{K}_n^{k(i)}$ 表示 $\mathcal{K}_n^k$ 中所有根节点出度为 i 的 C-图的集合，$K_n^{k(i)}$ 表示其大小，则有如下定理：

定理 8.4　对于给定的 k 和 $n \geq 2$，令 $\mathcal{T}_n = \bigcup_{i=1}^{n} \mathcal{K}_i^k$ 表示不超过层级 n 的所有 k-正则 C-图，T_n^k 表示其大小，则有：

$$c_1 \left(T_{n-1}^k\right)^k \leq T_n^k \leq c_2 \left(T_{n-1}^k\right)^k$$

其中 c_1, c_2 是两个与 k 有关的常数。

证明　类似在 2-正则图的推导，由8–7可知当 $n \geq 2$ 时有：

$$K_n^k = \sum_{j=1}^{k} \binom{T_{n-1}^k + 1}{j} - \sum_{j=1}^{k} \binom{T_{n-2}^k + 1}{j} - \sum_{i=1}^{k} 2^i K_n^{k(i)} \tag{8–16}$$

注意到：

$$\sum_{i=1}^{k} 2^i K_n^{k(i)} \leq 2^k \sum_{i=1}^{k} K_n^{k(i)} = 2^k K_n^k. \tag{8–17}$$

从而由8–16和8–17有：

$$K_n^k \geq \sum_{j=1}^{k} \binom{T_{n-1}^k + 1}{j} - \sum_{j=1}^{k} \binom{T_{n-2}^k + 1}{j} - 2^k K_n^k. \tag{8–18}$$

因此由8–16和8–18立马可得：

$$\sum_{j=1}^{k} \binom{T_{n-1}^k + 1}{j} - \sum_{j=1}^{k} \binom{T_{n-2}^k + 1}{j} \leq (2^k + 1) K_n^k \leq (2^k + 1) \left(\sum_{j=1}^{k} \binom{T_{n-1}^k + 1}{j} - \sum_{j=1}^{k} \binom{T_{n-2}^k + 1}{j} \right).$$

在上述不等式中对 n 求和，注意到 $T_1^k = 5$，则有：

$$\sum_{j=1}^{k} \binom{T_{n-1}^k + 1}{j} - 64 \leq (2^k + 1) T_n^k \leq (2^k + 1) \sum_{j=1}^{k} \binom{T_{n-1}^k + 1}{j}. \tag{8–19}$$

注意到 $\left(\frac{n}{k}\right)^k \leq \binom{n}{k} \leq \left(\frac{en}{k}\right)^k$，上式可进一步推导成：

$$\frac{1}{2^k + 1} \sum_{j=1}^{k} \left(\frac{T_{n-1}^k + 1}{j} \right)^j - \frac{64}{2^k + 1} \leq T_n^k \leq e^k \sum_{j=1}^{k} \left(\frac{T_{n-1}^k + 1}{j} \right)^j. \tag{8–20}$$

从而定理成立，得证。 □

定理8.4叙述了一个很有意思的事实，即当限定每个节点的出边个数时，C-图随着高度的数量会瞬间从非初等降到了双指数，也就是说在非确定的计算中，每一步的分支数量是否受限，其实是会影响到其最终复杂性的结果。

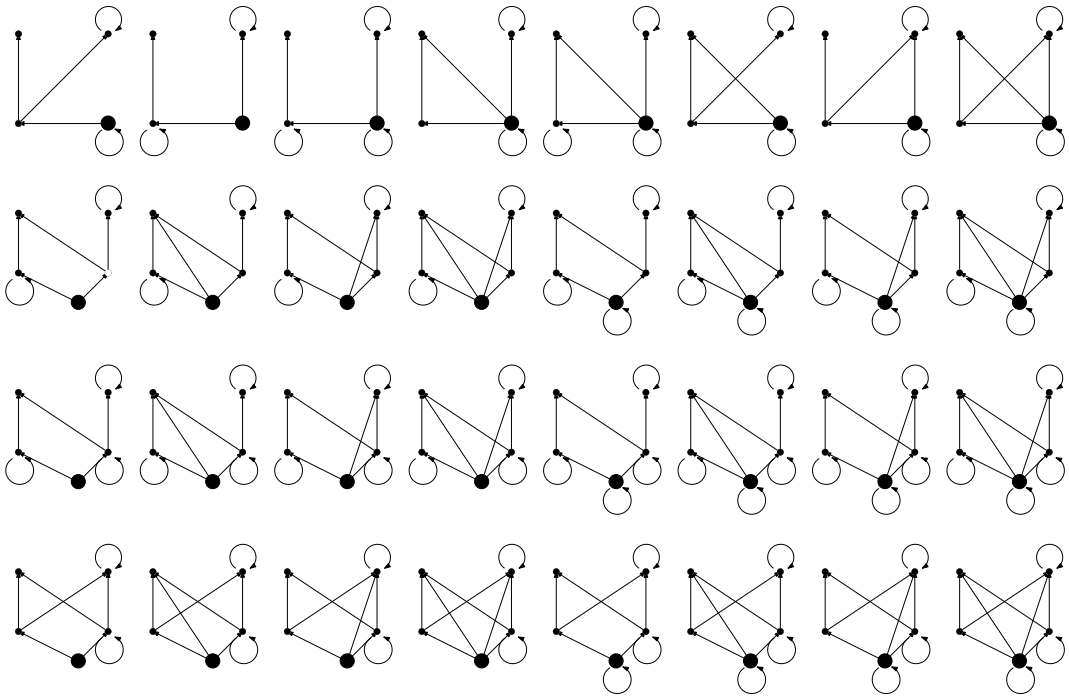


图 8–5 所有高度为 2 的 C-图

Figure 8–5 C-Graphs of Height 2

8.4 本章小结

本章从计数的方式讨论了计算对象的一些特性。通过使用了 C-图这一概念，本文用计算 C-图的个数来代替对计算对象复杂性的研究，并且对按高度分类给出了 C-图数量的递推式，同时也介绍了一类特殊的 C-图，k-正则 C-图。

从定理8.1可以看出一些关于非确定性的复杂之处。尽管 $L_0 = 2, L_1 = 3, L_2 = 40$ 看起来不是很大，图8–5也列出了所有高度为 2 的 C-图，但是 $L_3 = 2^{46} - 676$ 就变成了很夸张的数字，这其中有只有 6 个点的 C-图，也有指数大的 C-图，从而反映尽管所有的计算步骤都只经过很少的不同状态，其整个计算对象也可能十分复杂。

未来还有这两方面的方向可以拓展，首先目前对于固定大小的 C-图，目前只有一个算法可以算出其数量，而是否也如高度一样存在着简单的递推式却不得而知；其次 C-图的提出是搭了一个考察非确定计算复杂性的一个框架，其反映出了非确定计算的一些特性，但能否找到一些实际的应用使其能够真正发挥作用也是一个值得探究的工作。

第九章　总结

9.1　全文总结

本文首先介绍了向量加法系统上的一些验证问题的结论和方法，分别有可覆盖性问题和有界性问题 (章节三), 以及可达性问题 (章节四至七)。在可达性问题中首先介绍了两种不同的上界算法 (章节四,五)，然后介绍了其下界的证明 (章节六)，最后介绍了可达性问题在固定维度下的向量加法系统的结论 (章节七)。其中，针对可达性问题使用到的上界算法-KLMST 算法，本文提出了一个有别于 Leroux 等人在 [79] 的解释，将其分解和维度联系了起来，相信这会对研究固定维度下的向量加法系统的可达性问题有所帮助。而对于下界问题，本文整理了目前所有的下界方法，即用不能测 0 计数器去模拟一个有界的计数器测 0，这将对其他相关模型相关问题的下界证明起着指导的作用。

其次，本文从计数的角度研究了非确定计算的复杂性。Fu 在中在一个只有 τ 动作的进程模型 $\mathbb{CCS}^{\mu}$ 上定义了有限状态计算对象的形式化表示，并且给出其一个等价的图表示：C-图。本文在此基础上首先定义了 C-图的高度，即可表示为一个非确定的计算对象中一条计算路径经过的最多的不同的状态，并在此之上通过组合的方式，计算给出了给定高度的不同的 C-图的个数的递推式，从而反映出对应的非确定计算的复杂性。这是第一篇通过该角度来研究非确定计算复杂性的研究工作，为这方面的工作奠定了基础。

9.2　未来展望

未来有如下的方向继续研究：

- 进一步研究高维的向量加法系统的可达性问题。任意维下的向量加法系统的可达性问题在近些年已经获得了解决，即 **Ackermann**-完备[79-82]，但对于固定维的向量加法系统，目前只有 1 维和 2 维有相应的完备结论[75, 77]，而对于 3 维以上没有好的结论。而本文在章节四对 KLMST 算法的重新解释则在一定程度上联系了维度之间的关系，从而应该能对高维向量加法系统的可达性问题提供帮助。
- 研究一些相关模型的相关验证问题。向量加法系统随着在 60 年代的推出，也衍生了许多的相关模型，比如下推向量加法系统 (pushdown vector addition system)，带一维可测 0 的向量加法系统 (vector addition system with zero test)，

分支向量加法系统 (branching vector addition system) 等。向量加法系统上的一些技巧方法也可以运用到上面模型的验证问题研究当中，比如 bonnet 在 [46] 中使用了 Presburger 不变量方法证明了带一维可测 0 的向量加法系统的可达性问题是可判定的等。而近些年对着 KLMST 算法，测 0 模拟的新理解是否可以对这些模型的相关验证问题产生帮助也是值得思考的方向。

- 进一步用计数的方式研究非确定计算的复杂性。在本文中，我们用组合的方式计算出了给定高度不同 C-图的个数。一个很自然的问题是给定大小问不同 C-图的个数。目前我们的工作能够给出该问题的一个算法，但是是否能够给出其的递推式我们目前没有好的想法。此外将其运用到一些更具体的问题上也是日后的一个研究目标。

参考文献

[1] PETRI C A. Kommunikation mit automaten[J]., 1962.

[2] BALL T, CHAKI S, RAJAMANI S K. Parameterized verification of multithreaded software libraries[C]//International Conference on Tools and Algorithms for the Construction and Analysis of Systems. [S.l. : s.n.], 2001: 158-173.

[3] VAN DER AALST W M. The application of Petri nets to workflow management[J]. Journal of circuits, systems, and computers, 1998, 8(01): 21-66.

[4] HEINER M, GILBERT D, DONALDSON R. Petri nets for systems and synthetic biology[C]//International school on formal methods for the design of computer, communication and software systems. [S.l. : s.n.], 2008: 215-264.

[5] LEI L, LIN C, ZHONG Z. Stochastic Petri nets for wireless networks[J]., 2019.

[6] FAN L J, WANG Y Z, LI J Y, et al. Privacy Petri net and privacy leak software[J]. Journal of Computer Science and Technology, 2015, 30(6): 1318-1343.

[7] YU W, YAN C, DING Z, et al. Modeling and validating e-commerce business process based on Petri nets[J]. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 2013, 44(3): 327-341.

[8] 林闯. 随机 Petri 网和系统性能评价[M]. [出版地不详]: 清华大学出版社有限公司, 2005.

[9] 蒋屹新, 林闯, 曲扬, 等. 基于 Petri 网的模型检测研究[J]. Journal of Software, 2004, 15(9).

[10] BURNS F, KOELMANS A, YAKOVLEV A. Wcet analysis of superscalar processors using simulation with coloured petri nets[J]. Real-Time Systems, 2000, 18(2): 275-288.

[11] LEROUX H, ANDREU D, GODARY-DEJEAN K. Handling exceptions in Petri net-based digital architecture: From formalism to implementation on FPGAs[J]. IEEE Transactions on Industrial Informatics, 2015, 11(4): 897-906.

[12] ANGELI D, DE LEENHEER P, SONTAG E D. Persistence results for chemical reaction networks with time-dependent kinetics and no global conservation laws[J]. SIAM Journal on Applied Mathematics, 2011, 71(1): 128-146.

[13] BALDAN P, COCCO N, MARIN A, et al. Petri nets for modelling metabolic pathways: a survey[J]. Natural Computing, 2010, 9(4): 955-989.

[14] PELEG M, RUBIN D, ALTMAN R B. Using Petri net tools to study properties and dynamics of biological systems[J]. Journal of the American Medical Informatics Association, 2005, 12(2): 181-199.

[15] LI Y, DEUTSCH A, VIANU V. VERIFAS: a practical verifier for artifact systems[J]. ArXiv preprint arXiv:1705.10007, 2017.

[16] VAN DER AALST W M. Business process management as the "Killer App"for Petri nets[J]. Software & Systems Modeling, 2015, 14(2): 685-691.

[17] GERMAN S M, SISTLA A P. Reasoning about systems with many processes[J]. Journal of the ACM (JACM), 1992, 39(3): 675-735.

[18] BOUAJJANI A, EMMI M. Analysis of recursively parallel programs[J]. ACM Sigplan Notices, 2012, 47(1): 203-214.

[19] KAISER A, KROENING D, WAHL T. A widening approach to multithreaded program verification[J]. ACM Transactions on Programming Languages and Systems (TOPLAS), 2014, 36(4): 1-29.

[20] KARP R M, MILLER R E. Parallel program schemata[J]. Journal of Computer and system Sciences, 1969, 3(2): 147-195.

[21] HACK M. The recursive equivalence of the reachability problem and the liveness problem for Petri nets and vector addition systems[C]//15th Annual Symposium on Switching and Automata Theory (swat 1974). [S.l. : s.n.], 1974: 156-164.

[22] HACK M H T. Decidability questions for Petri Nets.[D]. Massachusetts Institute of Technology, 1976.

[23] KELLER R M. A fundamental theorem of asynchronous parallel computation[C]//Sagamore Computer Conference. [S.l. : s.n.], 1974: 102-112.

[24] BOJAŃCZYK M, DAVID C, MUSCHOLL A, et al. Two-variable logic on data words[J]. ACM Transactions on Computational Logic (TOCL), 2011, 12(4): 1-26.

[25] COLCOMBET T, MANUEL A. Generalized Data Automata and Fixpoint Logic[C]//LIPIcs: 34th International Conference on Foundation of Software Technology and Theoretical Computer Science, FSTTCS 2014, December 15-17, 2014, New Delhi, India: vol. 29. [S.l.]: Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2014: 267-278.

[26] DEMRI S, FIGUEIRA D, PRAVEEN M. Reasoning about data repetitions with counter systems[C]//2013 28th Annual ACM/IEEE Symposium on Logic in Computer Science. [S.l. : s.n.], 2013: 33-42.

[27] GANTY P, MAJUMDAR R. Algorithmic verification of asynchronous programs[J]. ACM Transactions on Programming Languages and Systems (TOPLAS), 2012, 34(1): 1-48.

[28] CRESPI-REGHIZZI S, MANDRIOLI D. Petri nets and Szilard languages[J]. Information and Control, 1977, 33(2): 177-192.

[29] ESPARZA J, GANTY P, LEROUX J, et al. Verification of population protocols[J]. Acta Informatica, 2017, 54(2): 191-215.

[30] MEYER R. A theory of structural stationarity in the π-calculus[J]. Acta Informatica, 2009, 46(2): 87-137.

[31] DECKER N, HABERMEHL P, LEUCKER M, et al. Ordered navigation on multi-attributed data words[C]//International Conference on Concurrency Theory. [S.l. : s.n.], 2014: 497-511.

[32] KANOVICH M I. Petri nets, Horn programs, linear logic and vector games[J]. Annals of Pure and Applied Logic, 1995, 75(1-2): 107-135.

[33] CARDOZA E, LIPTON R, MEYER A R. Exponential space complete problems for Petri nets and commutative semigroups (preliminary report)[C]// Proceedings of the eighth annual ACM symposium on Theory of computing. [S.l. : s.n.], 1976: 50-54.

[34] MAYR E W, MEYER A R. The complexity of the word problems for commutative semigroups and polynomial ideals[J]. Advances in mathematics, 1982, 46(3): 305-329.

[35] JONES N D, LANDWEBER L H, LIEN Y E. Complexity of some problems in Petri nets[J]. Theoretical Computer Science, 1977, 4(3): 277-299.

[36] CHENG A, ESPARZA J, PALSBERG J. Complexity results for 1-safe nets[J]. Theoretical Computer Science, 1995, 147(1-2): 117-136.

[37] STEWART I A. On the reachability problem for some classes of Petri nets[M]. [S.l.]: University of Newcastle upon Tyne, Computing Laboratory, 1991.

[38] HUYNH D T. Commutative grammars: The complexity of uniform word problems[J]. Information and Control, 1984, 57(1): 21-39.

[39] HOWELL R R, JANCAR P, ROSIER L E. Completeness results for single-path Petri nets[J]. Information and Computation, 1993, 106(2): 253-265.

[40] HOPCROFT J, PANSIOT J J. On the reachability problem for 5-dimensional vector addition systems[J]. Theoretical Computer Science, 1979, 8(2): 135-159.

[41] LEROUX J, PRAVEEN M, SUTRE G. Hyper-Ackermannian bounds for pushdown vector addition systems[C]//Proceedings of the Joint Meeting of the Twenty-Third EACSL Annual Conference on Computer Science Logic (CSL) and the Twenty-Ninth Annual ACM/IEEE Symposium on Logic in Computer Science (LICS). [S.l. : s.n.], 2014: 1-10.

[42] LEROUX J, SUTRE G, TOTZKE P. On the coverability problem for pushdown vector addition systems in one dimension[C]//International Colloquium on Automata, Languages, and Programming. [S.l. : s.n.], 2015: 324-336.

[43] LEROUX J, SUTRE G, TOTZKE P. On boundedness problems for pushdown vector addition systems[C]//International Workshop on Reachability Problems. [S.l. : s.n.], 2015: 101-113.

[44] FINKEL A, LEROUX J. Recent and simple algorithms for Petri nets[J]. Software & Systems Modeling, 2015, 14(2): 719-725.

[45] LAZIC R. The reachability problem for vector addition systems with a stack is not elementary[J]. ArXiv preprint arXiv:1310.1767, 2013.

[46] BONNET R. The reachability problem for vector addition system with one zero-test[C]//International Symposium on Mathematical Foundations of Computer Science. [S.l. : s.n.], 2011: 145-157.

[47] LEROUX J, SUTRE G. Reachability in two-dimensional vector addition systems with states: One test is for free[J]. ArXiv preprint arXiv:2007.09096, 2020.

[48] BONNET R, FINKEL A, LEROUX J, et al. Model checking vector addition systems with one zero-test[J]. ArXiv preprint arXiv:1205.4458, 2012.

[49] BONNET R, FINKEL A, LEROUX J, et al. Place-boundedness for vector addition systems with one zero-test[C]//FSTTCS 2010. [S.l. : s.n.], 2010: 192-203.

[50] LINCOLN P, MITCHELL J, SCEDROV A, et al. Decision problems for propositional linear logic[J]. Annals of pure and Applied Logic, 1992, 56(1-3): 239-311.

[51] COURTOIS J B, SCHMITZ S. Alternating vector addition systems with states[C]//International Symposium on Mathematical Foundations of Computer Science. [S.l. : s.n.], 2014: 220-231.

[52] VERMA K N, GOUBAULT-LARRECQ J. Karp-Miller trees for a branching extension of VASS.[J]. Discrete Mathematics and Theoretical Computer Science. DMTCS [electronic only], 2005, 7(1): 217-239.

[53] LAZIĆ R. The reachability problem for branching vector addition systems requires doubly-exponential space[J]. Information processing letters, 2010, 110(17): 740-745.

[54] DEMRI S, JURDZIŃSKI M, LACHISH O, et al. The covering and boundedness problems for branching vector addition systems[J]. Journal of Computer and System Sciences, 2012, 79(1): 23-38.

[55] LAZIĆ R, SCHMITZ S. Nonelementary complexities for branching VASS, MELL, and extensions[J]. ACM Transactions on Computational Logic (TOCL), 2015, 16(3): 1-30.

[56] GÖLLER S, HAASE C, LAZIĆ R, et al. A polynomial-time algorithm for reachability in branching VASS in dimension one[J]. ArXiv preprint arXiv:1602.05547, 2016.

[57] MAZOWIECKI F, PILIPCZUK M. Reachability for Bounded Branching VASS[C]//30th International Conference on Concurrency Theory (CONCUR 2019). [S.l. : s.n.], 2019.

[58] FIGUEIRA D, LAZIĆ R, LEROUX J, et al. Polynomial-space completeness of reachability for succinct branching VASS in dimension one[C]//International Colloquium on Automata, Languages, and Programming (ICALP). [S.l. : s.n.], 2017: 119.

[59] TURING A M, et al. On computable numbers, with an application to the Entscheidungsproblem[J]. J. of Math, 1936, 58(345-363): 5.

[60] BERNAYS P. Alonzo Church. An unsolvable problem of elementary number theory. American journal of mathematics, vol. 58 (1936), pp. 345–363.[J]. The Journal of Symbolic Logic, 1936, 1(2): 73-74.

[61] GÖDEL K. Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme I[J]. Monatshefte für mathematik und physik, 1931, 38(1): 173-198.

[62] MILNER R. Communication and concurrency[M]. [S.l.]: Prentice hall Englewood Cliffs, 1989.

[63] HOARE C A R. Communicating sequential processes[J]. Communications of the ACM, 1978, 21(8): 666-677.

[64] MILNER R, SANGIORGI D. Barbed bisimulation[C]//International Colloquium on Automata, Languages, and Programming. [S.l. : s.n.], 1992: 685-695.

[65] PALAMIDESSI C. Comparing the expressive power of the synchronous and asynchronous-calculi[J]. Mathematical Structures in Computer Science, 2003, 13(5): 685-719.

[66] FU Y, LU H. On the expressiveness of interaction[J]. Theoretical Computer Science, 2010, 411(11-13): 1387-1451.

[67] FU Y. Theory of interaction[J]. Theoretical Computer Science, 2016, 611: 1-49.

[68] LEEUWEN J V. A partial solution to the reachability-problem for vector-addition systems[C]//Proceedings of the sixth annual ACM symposium on Theory of computing. [S.l. : s.n.], 1974: 303-309.

[69] SACERDOTE G S, TENNEY R L. The decidability of the reachability problem for vector addition systems (preliminary version)[C]//Proceedings of the ninth annual ACM symposium on Theory of computing. [S.l. : s.n.], 1977: 61-76.

[70] MAYR E W. An algorithm for the general Petri net reachability problem[J]. SIAM Journal on computing, 1984, 13(3): 441-460.

[71] KOSARAJU S R. Decidability of reachability in vector addition systems (preliminary version)[C]//Proceedings of the fourteenth annual ACM symposium on Theory of computing. [S.l. : s.n.], 1982: 267-281.

[72] LAMBERT J L. A structure to decide reachability in Petri nets[J]. Theoretical Computer Science, 1992, 99(1): 79-104.

[73] LEROUX J. Vector addition system reachability problem: a short self-contained proof[C]//International Conference on Language and Automata Theory and Applications. [S.l. : s.n.], 2011: 41-64.

[74] CZERWIŃSKI W, LASOTA S, LAZIĆ R, et al. The reachability problem for Petri nets is not elementary[J]. Journal of the ACM (JACM), 2020, 68(1): 1-28.

[75] HAASE C, KREUTZER S, OUAKNINE J, et al. Reachability in succinct and parametric one-counter automata[C]//International Conference on Concurrency Theory. [S.l. : s.n.], 2009: 369-383.

[76] LEROUX J, SUTRE G. On flatness for 2-dimensional vector addition systems with states[C]//International Conference on Concurrency Theory. [S.l. : s.n.], 2004: 402-416.

[77] BLONDIN M, FINKEL A, GÖLLER S, et al. Reachability in two-dimensional vector addition systems with states is PSPACE-complete[C]//2015 30th Annual ACM/IEEE Symposium on Logic in Computer Science. [S.l. : s.n.], 2015: 32-43.

[78] CZERWIŃSKI W, LASOTA S, LÖDING C, et al. New pumping technique for 2-dimensional VASS[J]. ArXiv preprint arXiv:1906.10494, 2019.

[79] LEROUX J, SCHMITZ S. Reachability in vector addition systems is primitive-recursive in fixed dimension[C]//2019 34th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS). [S.l. : s.n.], 2019: 1-13.

[80] CZERWIŃSKI W, ORLIKOWSKI Ł. Reachability in Vector Addition Systems is Ackermann-complete[J]. ArXiv preprint arXiv:2104.13866, 2021.

[81] LASOTA S. Improved Ackermannian lower bound for the VASS reachability problem[J]. ArXiv preprint arXiv:2105.08551, 2021.

[82] LEROUX J. The reachability problem for petri nets is not primitive recursive[J]. ArXiv preprint arXiv:2104.12695, 2021.

[83] LEROUX J, SCHMITZ S. Demystifying reachability in vector addition systems[C]//2015 30th Annual ACM/IEEE Symposium on Logic in Computer Science. [S.l. : s.n.], 2015: 56-67.

[84] LEROUX J, SCHMITZ S. Ideal decompositions for vector addition systems (invited talk)[C]//33rd Symposium on Theoretical Aspects of Computer Science (STACS 2016). [S.l. : s.n.], 2016.

[85] SCHMITZ S. Algorithmic complexity of well-quasi-orders[D]. École normale supérieure Paris-Saclay, 2017.

[86] SCHMITZ S. Complexity hierarchies beyond elementary[J]. ACM Transactions on Computation Theory (TOCT), 2016, 8(1): 1-36.

[87] MINSKY M L. Computation[M]. [S.l.]: Prentice-Hall Englewood Cliffs, 1967.

[88] FEARNLEY J, JURDZIŃSKI M. Reachability in two-clock timed automata is PSPACE-complete[J]. Information and Computation, 2015, 243: 26-36.

[89] LIPTON R. The reachability problem requires exponential space[J]. Department of Computer Science. Yale University, 1976, 62.

[90] CZERWIŃSKI W, LASOTA S, ORLIKOWSKI Ł. Improved Lower Bounds for Reachability in Vector Addition Systems[C]//48th International Colloquium on Automata, Languages, and Programming (ICALP 2021). [S.l. : s.n.], 2021.

[91] FINKEL A, GOUBAULT-LARRECQ J. Forward analysis for WSTS, part I: Completions[J]. ArXiv preprint arXiv:0902.1587, 2009.

[92] FINKEL A, GOUBAULT-LARRECQ J. Forward analysis for WSTS, part II: Complete WSTS[C]//International Colloquium on Automata, Languages, and Programming. [S.l. : s.n.], 2009: 188-199.

[93] RACKOFF C. The covering and boundedness problems for vector addition systems[J]. Theoretical Computer Science, 1978, 6(2): 223-231.

[94] GEFFROY T, LEROUX J, SUTRE G. Occam's razor applied to the Petri net coverability problem[C]//International Workshop on Reachability Problems. [S.l. : s.n.], 2016: 77-89.

[95] ROSIER L E, YEN H C. A multiparameter analysis of the boundedness problem for vector addition systems[J]. Journal of Computer and System Sciences, 1986, 32(1): 105-135.

[96] CHAN T H. The boundedness problem for three-dimensional vector addition systems with states[J]. Information processing letters, 1988, 26(6): 287-289.

[97] HAASE C. On the complexity of model checking counter automata[D]. Citeseer, 2012.

[98] BAKER B S, BOOK R V. Reversal-bounded multipushdown machines[J]. Journal of Computer and System Sciences, 1974, 8(3): 315-332.

[99] IBARRA O H. Reversal-bounded multicounter machines and their decision problems[J]. Journal of the ACM (JACM), 1978, 25(1): 116-133.

[100] DANG Z, IBARRA O H, SAN PIETRO P. Liveness verification of reversal-bounded multicounter machines with a free counter[C]//International Conference on Foundations of Software Technology and Theoretical Computer Science. [S.l. : s.n.], 2001: 132-143.

[101] DEMRI S. On selective unboundedness of VASS[J]. Journal of Computer and System Sciences, 2013, 79(5): 689-713.

[102] HOFMAN P, TOTZKE P. Trace inclusion for one-counter nets revisited[C]// International Workshop on Reachability Problems. [S.l. : s.n.], 2014: 151-162.

[103] JANČAR P, ESPARZA J, MOLLER F. Petri nets and regular processes[J]. Journal of Computer and System Sciences, 1999, 59(3): 476-503.

[104] CZERWIŃSKI W, FIGUEIRA D, HOFMAN P. Universality Problem for Unambiguous VASS[J]. ArXiv preprint arXiv:2007.10907, 2020.

[105] HACK M. The equality problem for vector addition systems is undecidable[J]. Theoretical Computer Science, 1976, 2(1): 77-95.

[106] HILBERT D. Mathematische probleme[G]//Dritter Band: Analysis · Grundlagen der Mathematik · Physik Verschiedenes. [S.l.]: Springer, 1935: 290-329.

[107] JANČAR P. Decidability questions for bisimilarity of Petri nets and some related problems[C]//Annual Symposium on Theoretical Aspects of Computer Science. [S.l. : s.n.], 1994: 581-592.

[108] JANČAR P. Undecidability of bisimilarity for Petri nets and some related problems[J]. Theoretical Computer Science, 1995, 148(2): 281-301.

[109] REINHARDT K. Reachability in Petri nets with inhibitor arcs[J]. Rapport technique WSI-96-30, Wilhelm-Schickard Institut für Informatik, Universität Tübingen, 1996.

[110] HENNESSY M. Algebraic theory of processes[M]. [S.l.]: MIT press, 1988.

[111] MILNER R, PARROW J, WALKER D. A calculus of mobile processes, i[J]. Information and computation, 1992, 100(1): 1-40.

[112] NESTMANN U. Welcome to the jungle: A subjective guide to mobile process calculi[C]//International Conference on Concurrency Theory. [S.l. : s.n.], 2006: 52-63.

[113] SANGIORGI D, WALKER D. The pi-calculus: a Theory of Mobile Processes[M]. [S.l.]: Cambridge university press, 2003.

[114] FU Y. Non-deterministic structures of computation[J]. Mathematical Structures in Computer Science, 2015, 25(6): 1295-1338.

[115] ESPARZA J, NIELSEN M. Decidability issues for Petri nets[J]. Petri nets newsletter, 1994, 94: 5-23.

[116] ESPARZA J. Decidability and complexity of Petri net problems—an introduction[C]//Advanced Course on Petri Nets. [S.l. : s.n.], 1996: 374-428.

[117] BRAUER W, REISIG W, ROZENBERG G. Petri nets: central models and their properties: advances in petri nets 1986, part I proceedings of an advanced course bad honnef, 8.–19. September 1986[M]. [S.l.]: Springer, 2006.

[118] 张文博, 龙环. 向量加法系统验证问题研究综述[J]. 软件学报, 2017, 29(6): 1566-1581.

[119] PETRI C. Kommunikation mit Automaten", Bonn: Institut fur Instrumentelle Mathematik, Schriften des IIM Nr. 3, 1962, also, English translation," Communication with Automata[J]. Tech. Rep. RADC-TR-65-377, 1966.

[120] DICKSON L E. Finiteness of the odd perfect and primitive abundant numbers with n distinct prime factors[J]. American Journal of Mathematics, 1913, 35(4): 413-422.

[121] HIGMAN G. Ordering by divisibility in abstract algebras[J]. Proceedings of the London Mathematical Society, 1952, 3(1): 326-336.

[122] CANTOR G. Ueber unendliche, lineare Punktmannichfaltigkeiten[J]. Mathematische Annalen, 1879, 15(1): 1-7.

[123] VON NEUMANN J. On the introduction of transfinite numbers[J]. Reprinted in van Heijenoort (1967), 1923: 346-354.

[124] GRZEGORCZYK A. Some classes of recursive functions[J]. Journal of Symbolic Logic, 1955, 20(1).

[125] LÖB M H, WAINER S S. Hierarchies of number-theoretic functions. I[J]. Archiv für mathematische Logik und Grundlagenforschung, 1970, 13(1-2): 39-51.

[126] ACKERMANN W. Zum hilbertschen aufbau der reellen zahlen[J]. Mathematische Annalen, 1928, 99(1): 118-133.

[127] WAINER S S. A classification of the ordinal recursive functions[J]. Archiv für mathematische Logik und Grundlagenforschung, 1970, 13(3-4): 136-153.

[128] FAIRTLOUGH M, WAINER S S. Hierarchies of provably recursive functions[J]. Handbook of proof theory, 1998, 137: 149-207.

[129] SCHMITZ S. Complexity bounds for ordinal-based termination[C]// International Workshop on Reachability Problems. [S.l. : s.n.], 2014: 1-19.

[130] CICHON E A, BITTAR E T. Ordinal recursive bounds for Higman's theorem[J]. Theoretical Computer Science, 1998, 201(1-2): 63-84.

[131] SCHWICHTENBERG H, WAINER S S. Proofs and computations[M]. [S.l.]: Cambridge University Press, 2011.

[132] SCHMITZ S, SCHNOEBELEN P. Algorithmic aspects of WQO theory[J]., 2012.

[133] SCHRIJVER A. Theory of linear and integer programming[M]. [S.l.]: John Wiley & Sons, 1998.

[134] GILES F R, PULLEYBLANK W R. Total dual integrality and integer polyhedra[J]. Linear algebra and its applications, 1979, 25: 191-196.

[135] POTTIER L. Minimal solutions of linear diophantine systems: bounds and algorithms[C]//International Conference on Rewriting Techniques and Applications. [S.l. : s.n.], 1991: 162-173.

[136] BOUAJJANI A, DEREVENETC E, MEYER R. Checking and enforcing robustness against TSO[C]//European Symposium on Programming. [S.l. : s.n.], 2013: 533-553.

[137] EMERSON E A, NAMJOSHI K S. On model checking for non-deterministic infinite-state systems[C]//Proceedings. Thirteenth Annual IEEE Symposium on Logic in Computer Science (Cat. No. 98CB36226). [S.l. : s.n.], 1998: 70-80.

[138] CANAL C, POIZAT P, SALAÜN G. Model-based adaptation of behavioral mismatching components[J]. IEEE Transactions on Software Engineering, 2008, 34(4): 546-563.

[139] FINKEL A, HADDAD S, KHMELNITSKY I. Minimal coverability tree construction made complete and efficient[C]//International Conference on Foundations of Software Science and Computation Structures. [S.l. : s.n.], 2020: 237-256.

[140] KAISER A, KROENING D, WAHL T. Dynamic cutoff detection in parameterized concurrent programs[C]//International Conference on Computer Aided Verification. [S.l. : s.n.], 2010: 645-659.

[141] KONIG D. Theorie der endlichen und unendlichen Graphen[M]. [S.l.]: American Mathematical Soc., 2001.

[142] MAYR E W, MEYER A R. The complexity of the finite containment problem for Petri nets[J]. Journal of the ACM (JACM), 1981, 28(3): 561-576.

[143] GOUBAULT-LARRECQ J, HALFON S, KARANDIKAR P, et al. The ideal approach to computing closed subsets in well-quasi-orderings[G]//Well-Quasi Orders in Computation, Logic, Language and Reasoning. [S.l.]: Springer, 2020: 55-105.

[144] ZETZSCHE G. An approach to computing downward closures[C]//International Colloquium on Automata, Languages, and Programming. [S.l. : s.n.], 2015: 440-451.

[145] LEROUX J. Vector addition systems reachability problem (a simpler solution)[C]//EPiC: vol. 10. [S.l. : s.n.], 2012: 214-228.

[146] MÜLLER H. The reachability problem for VAS[G]//Advances in Petri nets 1984. [S.l.]: Springer, 1985: 376-391.

[147] REUTENAUER C. The mathematics of Petri nets[M]. [S.l.]: Prentice-Hall, Inc., 1990.

[148] LASOTA S. VASS reachability in three steps[J]. ArXiv preprint arXiv:1812.11966, 2018.

[149] DERSHOWITZ N, MANNA Z. Proving termination with multiset orderings[J]. Communications of the ACM, 1979, 22(8): 465-476.

[150] LEROUX J. The general vector addition system reachability problem by Presburger inductive invariants[C]//2009 24th Annual IEEE Symposium on Logic In Computer Science. [S.l. : s.n.], 2009: 4-13.

[151] LEROUX J. Presburger vector addition systems[C]//2013 28th Annual ACM/IEEE Symposium on Logic in Computer Science. [S.l. : s.n.], 2013: 23-32.

[152] GINSBURG S, SPANIER E. Semigroups, Presburger formulas, and languages[J]. Pacific journal of Mathematics, 1966, 16(2): 285-296.

[153] HAUSCHILDT D. Semilinearity of the reachability set is decidable for Petri nets[D]. University of Hamburg, Germany, 1990.

[154] JANČAR P. Decidability of a temporal logic problem for Petri nets[J]. Theoretical Computer Science, 1990, 74(1): 71-93.

[155] CZERWIŃSKI W, JĘDRYCH A. Long Runs Imply Big Separators in Vector Addition Systems[J]. ArXiv preprint arXiv:2105.00052, 2021.

[156] BOUZIANE Z. A primitive recursive algorithm for the general Petri net reachability problem[C]//Proceedings 39th Annual Symposium on Foundations of Computer Science (Cat. No. 98CB36280). [S.l. : s.n.], 1998: 130-136.

[157] JANČAR P. Bouziane's transformation of the Petri net reachability problem and incorrectness of the related algorithm[J]. Information and Computation, 2008, 206(11): 1259-1263.

[158] SCHNOEBELEN P. Revisiting Ackermann-hardness for lossy counter machines and reset Petri nets[C]//International Symposium on Mathematical Foundations of Computer Science. [S.l. : s.n.], 2010: 616-628.

[159] HOWELL R R, ROSIER L E, HUYNH D T, et al. Some complexity bounds for problems concerning finite and 2-dimensional vector addition systems with states[J]. Theoretical Computer Science, 1986, 46: 107-140.

[160] GAREY M R, JOHNSON D S. Computers and intractability[M]. [S.l.]: freeman San Francisco, 1979.

[161] VALIANT L G, PATERSON M S. Deterministic one-counter automata[J]. Journal of Computer and System Sciences, 1975, 10(3): 340-350.

[162] PARK D. Concurrency and automata on infinite sequences[G]//Theoretical computer science. [S.l.]: Springer, 1981: 167-183.

[163] VAN GLABBEEK R J, WEIJLAND W P. Branching time and abstraction in bisimulation semantics[J]. Journal of the ACM (JACM), 1996, 43(3): 555-600.

[164] ACETO L, HENNESSY M. Termination, deadlock, and divergence[J]. Journal of the ACM (JACM), 1992, 39(1): 147-187.

[165] WALKER D J. Bisimulation and divergence[J]. Information and Computation, 1990, 85(2): 202-241.

[166] LOHREY M, D'ARGENIO P R, HERMANNS H. Axiomatising divergence[J]. Information and computation, 2005, 203(2): 115-144.

[167] VAN GLABBEEK R, LUTTIK B, TRČKA N. Branching bisimilarity with explicit divergence[J]. Fundamenta Informaticae, 2009, 93(4): 371-392.

[168] Van GLABBEEK R, LUTTIK B, SPANINKS L. Rooted divergence-preserving branching bisimilarity is a congruence[J]. ArXiv preprint arXiv:1801.01180, 2018.

[169] PRIESE L. Concept of Simulation in Asynchronous, Concurrent Systems.[J]., 1980.

致　谢

时光荏苒，在交大已经不知不觉度过了 10 个年头。回想 2012 年初入交大之时，我绝不会想到我会在这边一路完成本科、博士阶段的学习，并且有幸加入了 BASICS 实验室这一个友爱欢乐的大集体。我的博士阶段留有遗憾，但我始终确信，加入 BASICS 是我这些年里最值得的选择之一。而在此论文完成之际，我有太多的感谢要向这一路上遇到的帮助过我的人所诉说。

首先我要感谢我的导师傅育熙教授。初次与您相见还是 2012 年本科刚入学时在新生仪式上聆听您作为院长给我们的寄语，当时已经被您的气度所折服，没想到在四年之后非常有幸能作为您的学生开展博士阶段的学习。傅老师用勤奋、求实、严谨的治学态度教导我带领我开展研究，时常给予我很大的启迪。同时傅老师对于所教授课程的细心负责，也深深的令我敬佩。此外在生活中傅老师也对我们十分关心，经常与我们一起踏青聚餐打桥牌，分享他生活中的经验，尽力帮助解决我们碰到的困难。在我心中，傅老师就是我理想中的教授风范，也是我一直以来的学习榜样和目标。

其次我要感谢李国强博士。感谢李老师一直以来在学业上关心支持和鼓励我。李老师在异步程序分析上提出了很多可以创新的研究内容让我印象深刻，对我的研究也起了重要的影响。

然后我感谢龙环博士。龙老师非常关心我们的学习生活，愿意分享他的经验，也非常乐于为我们排忧解难，同时龙老师也在科研和求职上给了我很多建议，很庆幸在实验室能遇到这么一位是老师但更像是一个大姐姐的存在。

接着我感谢 BASICS 实验室的其他老师。感谢陈翌佳教授。陈老师的风趣幽默为我们带来了十足的乐趣，同时陈老师细致清晰的授课也深深的令我钦佩。感谢符鸿飞博士，符老师令我领略到了并发程序线性性相关方面的研究。感谢蔡小娟博士，是蔡老师引领我进入了 BASICS 实验室这一个大家庭。感谢尹强博士，无论是我刚来作为师兄的你还是现在作为老师的你，都给予了我很多关心和帮助。我还要感谢邓玉欣教授，张民教授，董笑菊博士，张弛豪博士，感谢你们对于我学习生活上的帮助。

我还要感谢 BASICS 实验室的各位小伙伴们。感谢黄明璋博士，陶秀挺博士，张文博博士，王培新博士，我们一起在 327 里度过了非常愉快的时光。感谢何超栋博士、徐贤博士、薛建新博士、汪洋博士，非常开心能与他们一起交流。感谢现在我的学弟学妹们，他们分别是：吴昊，郑扬珞，陈薇骏，邱国良，刘国航，王

玉林等，与你们相处也总是能带给我欢乐和动力。

最后我要感谢我的父母，谢谢你们的包容让我可以自由的在我选择的道路驰骋前行，也感谢你们长久以来对我的关心和支持。我还要感谢亲爱的曹靓滢，感谢你一直默默的支持我给我力量。

攻读学位期间发表（或录用）的学术论文

[1] Yang, Qizhe, and Yuxi Fu. ”Counting nondeterministic computations.” Theoretical Computer Science. 897, 49-63, 2022.

[2] 杨启哲, 李国强. 基于通信 Petri 网的异步通信程序验证模型. 软件学报. 28(4), 804-818，2017.

攻读学位期间参与的项目

[1] 国家自然科学基金，面上项目，61672340，“异步通讯程序的程序分析理论与方法”，2017.1-2017.12。

[2] 国家自然科学基金，面上项目，61772336，“无穷状态系统等价性验证”，2018.1-2021.12。

[3] 国家自然科学基金，面上项目，62072299，“VASS 可达性的算法研究”，2021.1-2024.12。

[4] 国家自然科学基金，面上项目，61872232，“基于基本并发进程的异步通讯程序的验证模型与高效算法”，2019.1-2022.12。

[5] 国家自然科学基金，青年基金，61802254，“概率程序验证的理论研究”，2019.1-2021.12。

[6] 国家自然科学基金，面上项目，62172271，“概率程序断言违背概率的形式化方法理论”，2022.1-2025.12。