arXiv:2301.08969v1 [cs.FL] 21 Jan 2023

无限字上的 Parikh 自动机

Mario Grobler

University of Bremen, Bremen, Germany

Leif Sabellek

University of Bremen, Bremen, Germany

Sebastian Siebertz

University of Bremen, Bremen, Germany

摘要

有限字上的 Parikh 自动机最早由 Klaedtke 和 Rueß 在 [《自动机、语言与编程》 2003] 中引入。本文介绍了无限字上的 Parikh 自动机的若干变体，并研究其表达能力。我们证明，我们的一种新模型等价于 Fernau 和 Stiebe 在 [Fundamenta Informaticae 2008] 中引入的同步盲计数器机。所有这些模型都支持 ε-消除；据我们所知，对于盲计数器自动机而言，这仍是一个尚未解决的问题。随后，我们研究这些新自动机模型中的经典判定问题。

关键词：自动机理论、Parikh 自动机、无限词、ε-消除

目录

1 引言 2

2 预备知识 4

2.1 有限词与无限词.4 2.2 正则和 ω-正则语言.4 2.3 半线性集.4 2.4 Parikh 可识别语言.5

3 无限词上的 Parikh 自动机 5

3.1 Parikh-B¨uchi 自动机的定义.5 3.2 简单观察.7 3.3 ε-WPBA 与 ε-SPBA 的等价性.9

Email addresses: grobler@uni-bremen.de (Mario Grobler), sabellek@uni-bremen.de (Leif Sabellek), siebertz@uni-bremen.de (Sebastian Siebertz)

4 Schnepsilon 消除 104.1 ε-消除，用于 PPBA.10 4.2 ε-消除，用于 SPBA.21**5 PPBA 与盲计数器机的等价性及其闭包性质 26 6 PPBA 和 SPBA 的表达能力 28 7 判定问题 30 8 结论与展望 34**

1. 引言

有限字上的 Parikh 自动机（PA）最初由 Klaedtke 和 Rueß 于 [9],提出，是一种增配了计数器的有限自动机。PA 是一种非确定有限自动机，此外还配备了一个半线性集 C。此外，每个转移都配有一个 d元非负整数元组；每次使用转移时，计数器都会相应地按该元组中的数值递增。若 PA 终止于某个终态，且计数器中最终得到的 d元组属于 C，则该输入字被接受。PA 所识别的语言类包含所有正则语言，也包含许多其他语言，甚至包括非上下文无关语言，例如语言 $\{a^n b^n c^n \mid n \in \mathbb{N}\}$。另一方面，回文语言是上下文无关的，但无法由 PA 识别。

对于有限字，Parikh 自动机已得到广泛研究。如 [9],所示，PA 所识别的语言类恰好由扩展了线性基数约束的、带一个后继关系的弱存在单子二阶逻辑刻画。Cadilhac、Finkel 和 McKenzie 引入了仿射 Parikh 自动机这一变体 [2] ，并利用 Parikh 自动机刻画了有界语言 [3]。文献 [4] 研究了双向 PA，文献 [7]则研究了带可见下推栈的 PA。

本文开启了对无限字上的 Parikh 自动机的研究。Klaedtke 和 Rueß 在其工作 [9] 的结论中提出了这一研究方向。无限字上的自动机在逻辑和形式化验证中发挥着重要作用，参见例如教材 [11]。例如，一个需要将资源平均分配给三位消费者的非终止系统，可以用语言 $\{a^n b^n c^n \mid n \in \mathbb{N}\}^\omega$ 建模；该语言既不是 ω正则语言，甚至也不是 ω上下文无关语言。根据 B¨uchi 定理，每个正则ω语言都可以表示为 $U_i V_i^\omega$ 的有限并，其中 U_i, V_i 是正则语言。一个自然的推广是语言类 $\mathcal{L}^\omega_{\mathsf{PA}}$ ，其中的语言可以表示为 $U_i V_i^\omega$ 的并，其中 U_i, V_i 是 Parikh 可识别语言。Fernau 和 Stiebe 在 [5] 中也提到了这一类，并称之为 $\mathcal{K}_*$ 。

我们提出三种不同的定义，用于将带有 Parikh 条件的 B¨uchi 自动机 [1] 推广到我们称为$Parikh\text{-}B\ddot{u}chi$ 自动机 （PBA）的模型。所有模型在语法上都与概率自动机相同，区别仅在于语义。我们考虑带前缀接受条件的 *PBA* （PPBA）、带强重置接受条件的 *PBA* （SPBA），以及带弱重置接受条件的 *PBA* （WPBA）。此外，我们还考虑它们带有 ε-转移的变体，以及每个接受状态都配有各自半线性集的变体。

如果存在一条运行，使得该字的无穷多个前缀都满足接受条件，即自动机处于接受状态且向量当前总和属于该自动机的半线性集的情况会无穷多次发生，那么 PPBA 就接受其输入。该模型由 Klaedtke 和 Rueß 提出 [9]。我们用 $\mathcal{L}_{\mathsf{Prefix}}$ 表示 PPBA可识别语言这一类。我们证明，PPBA 等价于 Fernau 和 Stiebe 引入的同步盲计数器机 [5]。这一结果的主要难点在于消除 PPBA 中的 ε-转移，这项工作出乎意料地具有挑战性。根据这一等价性，PPBA可识别语言对并集封闭，但对交集或补集不封闭。此外，如果 L 是 Parikh语言，那么 L^ω 不一定是 PPBA可识别的，例如，语言 $\{a^n b^n \mid n \in \mathbb{N}\}^\omega$ 就不是 PPBA可识别的。

这种最后的缺陷促使我们定义了具有重置所有计数器值能力的 PBA 接受方式，由此引出了 SPBA 和 WPBA 的定义。SPBA 的定义与 PPBA 类似，但每当访问接受状态时，向量的当前总和必须 属于半线性集。访问接受状态后，所有计数器都重置为零。类似地，WPBA 并不强制必须 在接受状态重置，但可以重置。两种自动机都在无限次重置时接受。我们证明这两种模型具有相同的表达能力，并用 $\mathcal{L}_{\mathsf{Reset}}$ 表示 SPBA/WPBA 可识别的ω语言类。我们的一个主要结果是，通过证明 $\mathcal{L}^{\omega}_{\mathsf{PA}} \subsetneq \mathcal{L}_{\mathsf{Reset}}$，我们得出带重置条件的 PBA 的表达能力严格强于 PPBA。结合盲计数器自动机的已知结果，我们得到

$$\mathcal{L}_{\mathsf{Prefix}} \subsetneq \mathcal{L}^{\omega}_{\mathsf{PA}} \subsetneq \mathcal{L}_{\mathsf{Reset}}.$$

如果 L 是 Parikh语言，那么 L^ω 可由 SPBA/WPBA 识别，因此它们没有上述 PPBA 的缺陷。

最后，我们研究 PPBA 和 SPBA/WPBA 的经典判定问题。我们证明，对于所有模型，空性是 coNP 完全的，而普遍性（因而包含关系和等价性）是不可判定的。

组织结构。 在第 2 节介绍必要的背景知识后，我们将在第 3 节给出无限字上的 Parikh 自动机的定义，并进行一些简单观察。在第 4 节，我们将说明如何消除 ε-转移。在第 5 节，我们将证明 PPBA 与盲计数器自动机的等价性，并讨论一些推论。在第 6 节，我们将证明 $\mathcal{L}^{\omega}_{\mathsf{PA}} \subsetneq \mathcal{L}_{\mathsf{Reset}}$。在第 7 节，我们将研究 PPBA 和 SPBA/WPBA 的判定问题。最后，我们将在第 8 节总结全文。

2. 预备知识

2.1. 有限词与无限词

我们用 $\mathbb{N}$ 表示包含 0 在内的非负整数集合，用 $\mathbb{Z}$ 表示所有整数的集合。设 Σ 为一个字母表，即一个有限非空集合，并设 Σ^* 为 Σ上的所有有限词的集合。对于一个字 $w \in \Sigma^*$，我们用 $|w|$ 表示 w 的长度，用 $|w|_a$ 表示字母 $a \in \Sigma$ 在 w 中出现的次数。我们用 ε 表示长度为 0 的空词，用 $\Sigma^+ = \Sigma^* \setminus \{\varepsilon\}$ 表示 Σ上的所有非空有限词的集合。

字母表 Σ 上的无限词 是一个函数 $\alpha : \mathbb{N} \setminus \{0\} \to \Sigma$。我们通常用 α_i代替 $\alpha(i)$。因此，我们可以将无限词理解为符号 $\alpha = \alpha_1\alpha_2\alpha_3\ldots$构成的无限序列。对于 $m \le n$，我们用 $\alpha[m,n]$简写有限中间子串 $\alpha_m\ldots\alpha_n$。我们用 Σ^ω 表示字母表 Σ上的所有无限词组成的集合。我们称子集 $L \subseteq \Sigma^\omega$ 为一个 ω-语言。此外，对于 $L \subseteq \Sigma^*$，我们定义 $L^\omega = \{w_1w_2\cdots \mid w_i \in L \setminus \{\varepsilon\}\} \subseteq \Sigma^\omega$。

2.2. 正则语言与 ω-正则语言

非确定有限自动机（NFA）是一个元组 $\mathcal{A} = (Q, \Sigma, q_0, \Delta, F)$，其中 Q是状态的有限集，Σ 是输入字母表，$q_0 \in Q$ 是初始状态，$\Delta \subseteq Q \times \Sigma \times Q$是转移集，$F \subseteq Q$ 是接受状态集。自动机 $\mathcal{A}$ 在字 $w = w_1\cdots w_n \in \Sigma^*$ 上的运行是一个（可以为空的）转移序列 $r = r_1\ldots r_n$，其中 $r_i = (p_{i-1}, w_i, p_i) \in \Delta$，且满足 $p_0 = q_0$。如果 $p_n \in F$，我们就称 r 为接受运行。当且仅当 $q_0 \in F$时，ε 上的空运行是接受运行。我们将 $\mathcal{A}$ 所识别的语言定义为 $L(\mathcal{A}) = \{w \in \Sigma^* \mid$ 存在自动机 $\mathcal{A}$ 在 $w\}$上的接受运行。如果语言 L 由某个非确定有限自动机 $\mathcal{A}$识别，我们就称 L 为正则语言。

B ¨uchi 自动机（BA）是一个 NFA $\mathcal{A} = (Q, \Sigma, q_0, \Delta, F)$，以无限词作为输入。自动机 $\mathcal{A}$ 在无限词 $\alpha_1\alpha_2\alpha_3\ldots$上的一次运行，是一个无限转移序列$r = r_1r_2r_3\ldots$，其中 $r_i = (p_{i-1}, \alpha_i, p_i) \in \Delta$ 满足 $p_0 = q_0$。如果有无穷多个 i 满足 $p_i \in F$，我们就称 r 为接受运行。我们将由 $\mathcal{A}$ 识别的 ω-语言定义为 $L_\omega(\mathcal{A}) = \{\alpha \in \Sigma^\omega \mid$ 存在一条自动机 $\mathcal{A}$ 在 $\alpha\}$上的接受运行。如果某个 BA $\mathcal{A}$识别了一个 ω-语言 L，我们就称该语言为 L ω-正则。B ¨uchi 定理确立了正则语言与 ω-正则语言之间的重要联系：

> **定理 2.1（B ¨uchi）。** 语言 $L \subseteq \Sigma^\omega$ 是 ω-正则的，当且仅当存在正则语言 U_1、V_1、.....、U_n，$V_n \subseteq \Sigma^*$ 对于某个 $n \ge 1$，使得 $L = U_1V_1^\omega \cup \cdots \cup U_nV_n^\omega$。

2.3. 半线性集

维数为 d 的 $d \ge 1$ 线性集是形如 $\{b_0 + b_1z_1 + \cdots + b_\ell z_\ell \mid z_1, \ldots, z_\ell \in \mathbb{N}\} \subseteq \mathbb{N}^d$ 的集合，其中 $b_0, \ldots, b_\ell \in \mathbb{N}^d$。半线性集是有限个线性集的并集。对于向量 $\mathbf{u} = (u_1, \ldots, u_c) \in \mathbb{N}^c$和 $\mathbf{v} = (v_1, \ldots, v_d) \in \mathbb{N}^d$，我们用 $\mathbf{u} \cdot \mathbf{v} = (u_1, \ldots, u_c, v_1, \ldots, v_d) \in \mathbb{N}^{c+d}$ 表示 $\mathbf{u}$ 和 $\mathbf{v}$ 的拼接。我们将此定义扩展到向量集。设 $U \subseteq \mathbb{N}^c$ 且 $V \subseteq \mathbb{N}^d$。则 $U \cdot V = \{\mathbf{u} \cdot \mathbf{v} \mid \mathbf{u} \in U, \mathbf{v} \in V\} \subseteq \mathbb{N}^{c+d}$。我们用 $\mathbf{0}^d$（如果 d 在上下文中明确，也简写为 $\mathbf{0}$）表示全零向量；用 $\mathbf{e}_i^d$（或简写为 $\mathbf{e}_i$）表示 d维向量，其中第 i个分量为 1，其余分量均为 0。

2.4. Parikh 可识别语言

概率自动机（PA）是一个元组 $\mathcal{A}=(Q,\Sigma,q_0,\Delta,F,C)$，其中 Q、 Σ、 q_0和 F的定义与非确定有限自动机（NFA）相同， $\Delta\subseteq Q\times\Sigma\times\mathbb{N}^d\times Q$ 是带标签的转移集合， $C\subseteq\mathbb{N}^d$ 是一个半线性集。我们称 d为 $\mathcal{A}$ 的维数，并将转移$(p,a,\mathbf{v}\ q)$中向量 $\mathbf{v}$ 的各分量称为计数器。与 NFA 类似， $\mathcal{A}$ 在字 $w=x_1\dots x_n$ 上的运行是带标签转移 $r=r_1\dots r_n$ 的一个（可能为空的）序列，其中 $r_i=(p_{i-1},x_i,\mathbf{v}_i,p_i)\in\Delta$，且满足 $p_0=q_0$。我们将运行 r 的扩展 Parikh 像定义为 $\rho(r)=\sum_{i\le n}\mathbf{v}_i$ （约定空和等于 0 ）。如果 $p_n\in F$ 且 $\rho(r)\in C$，则称 r 为接受的；后一个条件称为 Parikh 条件。我们将 $\mathcal{A}$ 识别的语言定义为 $L(\mathcal{A})=\{w\in\Sigma^*\mid$ 存在 $\mathcal{A}$ 在 $w\}$上的接受运行。如果某个 PA 识别语言$L\subseteq\Sigma^*$，则称 L 为 Parikh 可识别。我们用 $\mathcal{L}_{\mathsf{PA}}$ 表示所有 Parikh 可识别语言构成的类。

3. 无限字上的 Parikh 自动机

本节将介绍我们提出的无限字上的 Parikh 自动机模型，并作一些简单观察。

3.1. Parikh-B ¨uchi 自动机的定义

我们先给出无限字上的 Parikh 自动机的定义。

定义3.1. *Parikh-B ¨uchi* 自动机(PBA) 是一个 PA $\mathcal{A}=(Q,\Sigma,q_0,\Delta,F,C)$。 $\mathcal{A}$ 在无限词 $\alpha_1\alpha_2\alpha_3\dots$ 上的一次运行，是由带标签的转移构成的无限序列$r=r_1r_2r_3\dots$，其中 $r_i=(p_{i-1},\alpha_i,\mathbf{v}_i,p_i)\in\Delta$ 满足 $p_0=q_0$。我们考虑三种接受条件。

1. 具有前缀接受条件的 *PBA* (PPBA)：如果存在无穷多个 $i\ge1$，使得 $p_i\in F$且 $\rho(r_1\dots r_i)\in C$，我们就称 r 满足前缀接受条件（简称接受）。对于这些位置 i，我们称在 r_i 中存在一次接受命中。我们将 PPBA $\mathcal{A}$ 识别的 ω-语言定义为 $P_\omega(\mathcal{A})=\{\alpha\in\Sigma^\omega\mid\mathcal{A}$ 接受 $\alpha\}$。2. 具有弱重置接受条件的 *PBA* (WPBA)：如果存在无穷多个重置位置 $0=k_0<k_1<k_2\dots$，使得 $p_{k_i}\in F$且对于所有 $i\ge1$， $\rho(r_{k_{i-1}+1}\dots r_{k_i})\in C$，我们就称 r 满足弱重置接受条件（简称接受）。我们将 WPBA $\mathcal{A}$ 识别的 ω-语言定义为 $W_\omega(\mathcal{A})=\{\alpha\in\Sigma^\omega\mid\mathcal{A}$ 接受 $\alpha\}$。

3. 具有强重置接受条件的 *PBA* (SPBA)：令 $k_0=0$，并用k_1， k_2，... 表示 r中所有接受状态的位置，即对于所有 $i\ge1$， $p_{k_i}\in F$。我们称 r 满足强重置接受条件 （简称接受 ），如果k_1， k_2，... 是一个无限序列，并且对于所有 $i\ge1$， $\rho(r_{k_{i-1}+1}\dots r_{k_i})\in C$。我们将 SPBA $\mathcal{A}$ 识别的 ω- 语言定义为： $S_\omega(\mathcal{A})=\{\alpha\in\Sigma^\omega\mid\mathcal{A}$ 接受 $\alpha\}$。

直观地说，SPBA $\mathcal{A}$ 接受所有无限词 α，其方式是：每当 $\mathcal{A}$ 在 α 上的运行访问到一个接受状态时，就必须 满足 Parikh 条件。之后，计数器会被重置。相比之下，WPBA可以 在访问接受状态时重置计数器，但并非必须如此。只要它们无限次重置计数器，两者就都接受各自的

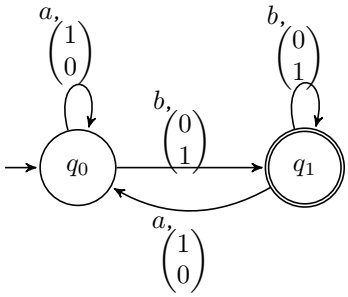

图 1：来自例 1 的 PBA $\mathcal{A}$，其具有 $C = \{(z,\ z) \mid z \in \mathbb{N}\}$。

计数器。PPBA 完全不会重置其计数器。若存在一条运行，其中有无限多个满足 Parikh 条件的接受（有限）前缀，它就会接受。由于所有自动机在语法上都是相同的对象，为了明确区分不同的接受条件，我们用 P_ω、W_ω 和 S_ω 表示其接受的 ω-语言。出于同样的原因，在 SPBA/WPBA 的语境下，我们通常会说重置状态，而不是接受状态。

例 1。设 $\mathcal{A}$，其具有 $C = \{(z,\ z) \mid z \in \mathbb{N}\}$，是图 1 所示的 PBA。则$W_\omega(\mathcal{A}) = \{w \in \{a,b\}^* \cdot \{b\} \mid |w|_a = |w|_b\}^\omega$。请注意，该语言不是 ω-正则的。第一个计数器记录已经读取的 as 的数量，第二个计数器记录已经读取的 bs 的数量。半线性集 C 实质上表示这两个值最终需要相同。由于 WPBA 在访问接受状态时可能会也可能不会重置其计数器，因此 $\mathcal{A}$ 可以在其两个状态之间“自由移动”。请注意，$P_\omega(\mathcal{A}) = W_\omega(\mathcal{A})$。

考虑强重置接受条件，我们有 $S_\omega(\mathcal{A}) = \{ab\}^\omega$，该自动机只接受一个无限词。假设 $\mathcal{A}$ 尝试接受一个无限词 α，其中有一个中间子串 a^n 满足 $n > 1$。也就是说，$\mathcal{A}$ 在 q_0 中循环 n 次，因此第一个计数器的值为 n，第二个计数器的值为 0。增加第二个计数器的唯一机会是转移到接受状态 q_1。由于此时两个计数器的值并不相同，α 将被拒绝。如果 $\mathcal{A}$ 尝试接受一个以 b 开头的无限词，或者其中有一个中间子串 b^n 满足 $n > 1$，则出于类似原因，输入将被拒绝。

现在我们介绍一种 PBA，它允许所有接受状态分别对应不同的半线性集，我们称之为 Multi-PBA。

定义 3.2. 一个*Multi-PBA* 是一个元组 $\mathcal{A} = (Q, \Sigma, q_0, \Delta, F, C)$，其中 $C\colon F \to \mathcal{P}(\mathbb{N}^d)$ 是一个函数，为每个 $q \in F$ 指定一个半线性集。我们将 Multi-PPBA (MPBA)、MWPBA 和 MSPBA 分别定义为 SPBA、WPBA 和 SPBA，只是将各自定义中的条件 $\rho(r_i, \dots, r_j) \in C$ 替换为 $\rho(r_i, \dots, r_j) \in C(p_j)$。

最后，我们给出允许 ε-转移的 PBA 的定义。

定义 3.3。ε-PBA 是一个元组 $\mathcal{A} = (Q, \Sigma, q_0, \Delta, \mathcal{E}, F, C)$，其中 $\mathcal{E} \subseteq Q \times \{\varepsilon\} \times \mathbb{N}^d \times Q$ 是带标记的 ε-转移集合，其他所有项的定义与 PBA 相同。在无限词 $\alpha_1\alpha_2\alpha_3\dots$ 上，$\mathcal{A}$ 的一次运行是一个无限转移序列 $r \in (\mathcal{E}^*\Delta)^\omega$，记作 $r = r_1r_2r_3\dots$，其中 $r_i = (p_{i-1}, \gamma_i, \mathbf{v}_i, p_i)$，满足 $p_0 = q_0$；若 $r_i \in \mathcal{E}$，则 $\gamma_i = \varepsilon$；若 $r_i \in \Delta$ 是 r 中第 j 次出现的（非 ε）转移，则 $\gamma_i = \alpha_j$。

前缀接受条件、弱重置接受条件和强重置条件与定义 3.1 中的相同，并以自然的方式扩展到 Multi-PBA。我们分别使用术语 ε-PPBA、ε-WPBA、ε-SPBA、ε-MPPBA、ε-MWPBA 和 ε-MSPBA。我们将形如 $(p,\ \varepsilon,\ \mathbf{v}\ p)$ 的转移称为 $\in \mathcal{E}$ 一个 ε-环。

注意，我们可以将每个 PBA 都视为一个 ε-PBA，也就是说，PBA $\mathcal{A}=(Q,\ \Sigma,\ q_0,\ \Delta,\ F,\ C)$ 等价于 ε-PBA $\mathcal{A}'=(Q,\ \Sigma,\ q_0,\ \Delta,\ \varnothing,\ F,\ C)$，其含义是 $P_\omega(\mathcal{A}) = P_\omega(\mathcal{A}')$、$W_\omega(\mathcal{A}) = W_\omega(\mathcal{A}')$，且 $S_\omega(\mathcal{A}) = S_\omega(\mathcal{A}')$。

3.2. 简单观察。

我们先给出几个简单观察。首先，我们证明 ε-PPBA $\mathcal{A}$ 的接受状态 F与其线性集 C 在某种意义上是相互独立的，具体如下引理所述。

引理 3.1。设 $\mathcal{A}=(Q,\ \Sigma,\ q_0,\ \Delta,\ \mathcal{E},\ F,\ C)$ 是一个 ε-PPBA，其中 $C=C_1\cup\cdots\cup C_\ell$是线性集 $C_j, 1\le j\le \ell$。那么 $P_\omega(\mathcal{A}) = \bigcup_{f\in F}\bigcup_{1\le j\le \ell} P_\omega(Q,\ \Sigma,\ q_0,\ \Delta,\ \mathcal{E},\ \{f\},\ C_j)$。

证明。从右到左的方向显然成立，因此我们证明从左到右的方向。设$\alpha\in P_\omega(\mathcal{A})$有一个接受运行 $r=r_1r_2r_3\ldots$，其中 $r_i=(p_{i-1},\ \gamma_i,\ \mathbf{v}_i,\ p_i)$，即，存在无限多个 $i\ge 1$ 使得 $p_i\in F$且 $\rho(r_1\ldots r_i)\in C$。根据无限鸽巢原理，存在一个状态 $f\in F$ 使得 $p_{i'}=f$，并且存在 j，使得对于无限多个 $i'\ge 1$，都有 $\rho(r_1\ldots r_{i'})\in C_j$。因此，$\alpha\in P_\omega(Q,\ \Sigma,\ q_0,\ \Delta,\ \mathcal{E},\ \{f\},\ C_j)$。 □

请注意，这个引理通常不适用于 ε-WPBA 和 ε-SPBA。

下面的引理展示了一种简单却重要的组合方法：使用额外的计数器和一个经过调整的半线性集来存储有关运行的信息。

引理 3.2. 对于每个 ε-MPPBA $\mathcal{A}$，都存在一个具有单个接受状态的 ε-PPBA $\mathcal{A}'$，使得 $P_\omega(\mathcal{A}) = P_\omega(\mathcal{A}')$。如果 $\mathcal{A}$ 没有非自环的 ε-转移，那么 $\mathcal{A}'$ 也没有非自环的 ε-转移；并且如果 $\mathcal{A}$ 是一个 MPPBA，那么 $\mathcal{A}'$ 就是一个 PPBA。

证明。设 $\mathcal{A}=(Q,\ \Sigma,\ q_0,\ \Delta,\ \mathcal{E},\ F,\ C)$ 是一个维数为 d的 ε-MPPBA，且令 $F=\{q_1,\ldots,q_k\}$。由于 $\mathcal{A}$是一个 ε-MPPBA，$C:\ F\to\mathcal{P}(\mathbb{N}^d)$ 是一个将每个 $q\in F$映射到一个半线性集的函数。我们构造一个维数为 $d+k$ 且具有单个接受状态的 ε-PPBA $\mathcal{A}'$。我们引入一个新的接受状态 q_f。一种简单的做法是通过 ε-转移将所有 q_i 连接到 q_f，反之亦然。然而，这种做法行不通，因为 q_f无法获知某次运行是从哪个接受状态进入的，因此也不知道哪些出边有效，从而无法避免产生无效的捷径。因此，我们利用非确定性，猜测一个会被无限次访问的接受状态（例如 q_i），以满足前缀接受条件。此外，我们为每个接受状态引入一个新的计数器，以确保我们只使用与 q_i相关的转移。

形式上，令 $\mathcal{A}'=(Q\cup\{q_f\}、\ \Sigma、\ \Delta'、\ \mathcal{E}'、\ \{q_f\}、\ C)$，其中

$$\begin{aligned}\Delta' = &\{(q,a,\mathbf{v}\cdot\mathbf{0}^k,q') \mid (q,a,\mathbf{v},q')\in\Delta\}\\ &\cup\{(q,a,\mathbf{v}\cdot\mathbf{e}_i^k,q_f) \mid (q,a,\mathbf{v},q_i)\in\Delta, i\le k\}\\ &\cup\{(q_f,a,\mathbf{v}\cdot\mathbf{e}_i^k,q) \mid (q_i,a,\mathbf{v},q)\in\Delta, i\le k\}\\ &\cup\{(q_f,a,\mathbf{v}\cdot\mathbf{e}_i^k,q_f) \mid (q_i,a,\mathbf{v},q_i)\in\Delta, i\le k\}\end{aligned}$$

and

$$\begin{aligned}\mathcal{E}' = &\{(q, \varepsilon, \mathbf{v} \cdot \mathbf{0}^k, q') \mid (q, \varepsilon, \mathbf{v}, q') \in \mathcal{E}\} \\ &\cup \{(q, \varepsilon, \mathbf{v} \cdot \mathbf{e}_i^k, q_f) \mid (q, \varepsilon, \mathbf{v}, q_i) \in \mathcal{E}, i \le k, q \ne q_i\} \\ &\cup \{(q_f, \varepsilon, \mathbf{v} \cdot \mathbf{e}_i^k, q) \mid (q_i, \varepsilon, \mathbf{v}, q) \in \mathcal{E}, i \le k, q \ne q_i\} \\ &\cup \{(q_f, \varepsilon, \mathbf{v} \cdot \mathbf{e}_i^k, q_f) \mid (q_i, \varepsilon, \mathbf{v}, q_i) \in \mathcal{E}, i \le k\}.\end{aligned}$$

定义 $C_i = C(q_i) \cdot \{z\mathbf{e}_i^k \mid z \in \mathbb{N}\}$，并令 $C = \bigcup_{i \le k} C_i$。我们声称 $P_\omega(\mathcal{A}) = P_\omega(\mathcal{A}')$。

为了证明 $P_\omega(\mathcal{A}) \subseteq P_\omega(\mathcal{A}')$，设 $\alpha \in P_\omega(\mathcal{A})$ 有一个接受运行 $r = r_1 r_2 r_3 \ldots$，它是 $\mathcal{A}$ 的运行，其中 $r_j = (p_{j-1}, \gamma_j, \mathbf{v}_j, p_j)$。根据无限鸽巢原理，存在一个接受状态 q_i，它无限多次满足前缀接受条件，即存在无限多个 j 使得 $p_j = q_i$，且 $\rho(r_1 \ldots r_j) \in C(q_i)$。由 r，我们通过替换每个读取完 α_1 后形如 $(p_{j-1}, \gamma_j, \mathbf{v}_j, q_i)$ 的转移为 $(p_{j-1}, \gamma_j, \mathbf{v}_j \cdot \mathbf{e}_i^k, q_f)$，将每个 $(q_i, \gamma_j, \mathbf{v}_j, q_i)$ 替换为 $(q_f, \gamma_j, \mathbf{v}_j \cdot \mathbf{e}_i^k, q_f)$，并将每个 $(q_i, \gamma_j, \mathbf{v}_j, p_j)$ 替换为 $(q_f, \gamma_j, \mathbf{v}_j \cdot \mathbf{e}_i^k, p_j)$，从而构造出 $\mathcal{A}'$ 的一个接受运行 $r' = r'_1 r'_2 r'_3 \ldots$。最后，将其他每个转移 $(p_{j-1}, \gamma_j, \mathbf{v}_j, p_j)$ 替换为 $(p_{j-1}, \gamma_j, \mathbf{v}_j \cdot \mathbf{0}^k, p_j)$。根据 Δ' 和 $\mathcal{E}'$ 的选取，运行 r' 确实是 $\mathcal{A}'$ 在 α 上的运行。此外，r' 是接受的，因为在 q_i 中，r 里读取完 α_1 后的每个接受命中（读取 α 的第一个符号之前，我们可能无法使用捷径进入 q_f）都会对应到 r' 中的一个接受命中：两个运行中的前 d 个计数器值在每个位置都相等，并且根据构造，所附加的 k 维向量在 i 分量中只有一个非零值。因此，只要有 $\rho(r_1 \ldots r_j) \in C(q_i)$，就有 $\rho(r'_1 \ldots r'_j) \in C_i \subseteq C$。

为了证明 $P_\omega(\mathcal{A}) \supseteq P_\omega(\mathcal{A}')$，设 $\alpha \in P_\omega(\mathcal{A}')$ 有接受运行 $r' = r'_1 r'_2 r'_3 \ldots$，其中 $\mathcal{A}'$满足 $r'_j = (p_{j-1}, \gamma_j, \mathbf{v}_j, p_j)$。根据 C的构造，存在唯一的 C_i，使得$\rho(r'_1 \ldots r'_j) \in C_i$对无穷多个 j成立。与上文类似，我们可以将 r' 中每个 q_f 都替换为 q_i，并舍弃附加计数器，从而得到 $\mathcal{A}$ 在 α上的一个有效运行 r。该运行是接受的，因为 r' 中 q_f 上的每个接受命中，都对应于 r中 q_i 上的一个接受命中。

最后请注意，如果 $\mathcal{A}$ 只有非循环的ε-转移，那么 $\mathcal{A}'$ 也只有非循环的 ε-转移；因为在这种情况下，$\mathcal{E}'$ 定义的第一行和第四行只引入ε-循环，而第二行和第三行所定义的集合为空。如果 $\mathcal{A}$ 是一个 MPPBA，我们可以将其视为一个带有 $\mathcal{E}' = \varnothing$的 ε-MPPBA，因此在这种情况下，$\mathcal{A}'$ 可以看作一个 PPBA。 □

注意，我们可以使用类似的技巧，将 (ε-)MSPBA 和 (ε-)MWPBA 转换为等价的 (ε-)SPBA 和 (ε-)WPBA。不过，一般而言，单个接受状态已不再足够。

> 引理 3.3. 对于每个 ε-MSPBA（ε-MWPBA）$\mathcal{A}$，都存在一个 ε-SPBA（ε- WPBA）$\mathcal{A}'$，使得 $S_\omega(\mathcal{A}) = S_\omega(\mathcal{A}')$（$W_\omega(\mathcal{A}) = W_\omega(\mathcal{A}')$）。如果 $\mathcal{A}$ 没有非自环ε-转移，那么 $\mathcal{A}'$ 也没有非自环 ε-转移；并且如果 $\mathcal{A}$ 是一个 MSPBA（MWPBA），那么 $\mathcal{A}'$ 是一个 SPBA（WPBA）。

证明概要。令 $\mathcal{A} = (Q, \Sigma, q_0, \Delta, \mathcal{E}, F, C)$ 为一个维数为 d 的（ε-）MSPBA，并令 $F = \{q_1, \ldots, q_k\}$。同样，$C\colon F \to \mathcal{P}(\mathbb{N}^d)$ 是一个将半线性集赋给每个 $q \in F$ 的函数。我们构造一个等价的、维数为 $d + k$ 且只有一个半线性集 $C' \subseteq \mathbb{N}^{d+k}$ 的（ε-）SPBA $\mathcal{A}'$。

其思路是利用附加计数器标记我们进入了哪个重置状态：也就是说，每个不转移到重置状态的转移都只需补上 0；而每个进入重置状态 q_i 的转移，则恰好在第 $d+i$ 个位置上标记为 1。半线性集 C' 定义为 $\bigcup_{1\le i\le k} C(q_i)\cdot\{\mathbf{e}_i^d\}$。

当 $\mathcal{A}$ 是一个（ε）-MWPBA 时，我们还会为每条进入重置状态的转移添加一条补以 0 的转移。这样，得到的自动机就能非确定性地决定是否要重置。

最后请注意，如果 $\mathcal{A}$只有非自环的 ε-转移，那么得到的自动机 $\mathcal{A}'$ 也只有非自环的 ε-转移。如果 $\mathcal{A}$ 是一个 MPPBA，我们可以将其视为一个满足 $\mathcal{E}'=\varnothing$ 的 ε-MPPBA，因此，在这种情况下，$\mathcal{A}'$ 可以看作一个 PPBA。 □

3.3. ε-WPBA 与 ε-SPBA 的等价性

现在我们证明，重置模型 ε-SPBA 和 ε-WPBA 定义了相同的 ε-语言类。此外，我们可以高效地将 ε-SPBA 转换为 ε-WPBA，反之亦然。

> 引理 3.4。 每个 ε-SPBA $\mathcal{A}$ 都等价于一个 ε-WPBA $\mathcal{A}'$，二者具有相同数量的状态，且后者多使用一个计数器。如果 $\mathcal{A}$ 是一个 SPBA，那么 $\mathcal{A}'$ 是一个 WPBA。

证明。 令 $\mathcal{A}=(Q,\ \Sigma,\ q_0,\ \Delta,\ \mathcal{E},\ F,\ C)$ 为一个 ε-SPBA。我们构造一个与之等价的 ε-WPBA $\mathcal{A}'$ 来模拟 $\mathcal{A}$，并确保任何运行都不会在未重置的情况下访问接受状态。为此，我们添加一个额外的计数器，用于跟踪访问接受状态的次数（期间未进行重置）。此外，我们修改 C，使得每次访问接受状态时，这个新计数器都必须设为 1，从而禁止在未重置的情况下经过此类状态。

我们选择 $\mathcal{A}'=(Q,\ \Sigma,\ q_0,\ \Delta',\ \mathcal{E}'F,\ C')$，其中

$$\begin{aligned}\Delta' = &\{(q,x,\mathbf{v}\cdot 0,q') \mid (q,x,\mathbf{v},q')\in\Delta, q'\notin F\}\\ &\cup\{(q,x,\mathbf{v}\cdot 1,q_f) \mid (q,x,\mathbf{v},q_f)\in\Delta, q_f\in F\},\end{aligned}$$

类似地，

$$\begin{aligned}\mathcal{E}' = &\{(q,\varepsilon,\mathbf{v}\cdot 0,q') \mid (q,\varepsilon,\mathbf{v},q')\in\mathcal{E}, q'\notin F\}\\ &\cup\{(q,\varepsilon,\mathbf{v}\cdot 1,q_f) \mid (q,\varepsilon,\mathbf{v},q_f)\in\mathcal{E}, q_f\in F\}.\end{aligned}$$

最后，令 $C'=C\cdot\{1\}$。我们声称 $\mathcal{A}'$ 是一个与 $\mathcal{A}$ 等价的 ε-WPBA。

令 $\alpha\in S_\omega(\mathcal{A})$。由于 $\mathcal{A}$ 每次访问接受状态时都会重置，$\mathcal{A}'$ 可以使用相同的状态和重置位置来模拟一次接受运行。特别地，每次访问接受状态时，新计数器的值都为 1。因此，C' 的选择意味着 $\alpha\in W_\omega(\mathcal{A}')$。

现在，令 $\alpha\in W_\omega(\mathcal{A}')$。由于 C' 的选择，$\mathcal{A}$ 每次访问接受状态时都必须重置计数器。否则，用于跟踪访问接受状态次数的新计数器将大于 1，从而违反弱重置接受条件。因此 $\alpha\in S_\omega(\mathcal{A})$。 □

> 引理 3.5. 每个 ε-WPBA $\mathcal{A}$ 都等价于一个 ε-SPBA $\mathcal{A}'$，其状态数至多是原来的两倍，且计数器数量相同。如果 $\mathcal{A}$ 是一个SPBA，那么 $\mathcal{A}'$ 也是一个WPBA。

证明。 设 $\mathcal{A} = (Q, \Sigma, q_0, \Delta, \mathcal{E}, F, C)$是一个 ε-WPBA。我们构造一个与之等价的 ε-SPBA $\mathcal{A}'$，通过赋予其任意长时间“避开”接受状态的选项来模拟 $\mathcal{A}$。为此，我们创建 F的一个非接受副本。因此， $\mathcal{A}'$ 可以利用非确定性决定继续还是重置一次部分运行。

我们选择 $\mathcal{A}' = (Q \cup \{\hat{q}_f \mid q_f \in F\},\ \Sigma,\ q_0,\ \Delta',\ \mathcal{E}',\ F,\ C)$，其中

$$\begin{aligned}\Delta' = \Delta &\cup \{(q, x, \mathbf{v}, \hat{q}_f) \mid (q, x, \mathbf{v}, q_f) \in \Delta, q_f \in F\} \\ &\cup \{(\hat{q}_f, x, \mathbf{v}, q) \mid (q_f, x, \mathbf{v}, q) \in \Delta, q_f \in F\} \\ &\cup \{(\hat{q}_f, x, \mathbf{v}, \hat{q}'_f) \mid (q_f, x, \mathbf{v}, q'_f) \in \Delta, q_f, q'_f \in F\},\end{aligned}$$

类似地,

$$\begin{aligned}\mathcal{E}' = \mathcal{E} &\cup \{(q, \varepsilon, \mathbf{v}, \hat{q}_f) \mid (q, \varepsilon, \mathbf{v}, q_f) \in \mathcal{E}, q_f \in F\} \\ &\cup \{(\hat{q}_f, \varepsilon, \mathbf{v}, q) \mid (q_f, \varepsilon, \mathbf{v}, q) \in \mathcal{E}, q_f \in F\} \\ &\cup \{(\hat{q}_f, \varepsilon, \mathbf{v}, \hat{q}'_f) \mid (q_f, \varepsilon, \mathbf{v}, q'_f) \in \mathcal{E}, q_f, q'_f \in F\}.\end{aligned}$$

W我们声称 $\mathcal{A}'$ 是一个与 ε 等价的 ε-SPBA $\mathcal{A}$.

$\Rightarrow$ 我们首先证明 $W_\omega(\mathcal{A}) \subseteq S_\omega(\mathcal{A}')$。设 $\alpha \in W_\omega(\mathcal{A})$，且 $r = r_1 r_2 r_3 \ldots$是 $\mathcal{A}$在 α上的一个接受运行，其重置位置为 $k_0 < k_1 < k_2 \ldots$。现在， $\mathcal{A}'$能够通过为每个状态 $q_i \in F$ 选择状态 $\hat{}q_i$ 来模拟 r，其中 $i \neq k_j$, $j \geq 1$。因此， $\mathcal{A}'$ 访问接受状态，当且仅当 $\mathcal{A}$ 在同一位置重置其计数器。于是， $\alpha \in S_\omega(\mathcal{A}')$。

$\Leftarrow$为了证明 $S_\omega(\mathcal{A}') \subseteq W_\omega(\mathcal{A})$，设 $\alpha \in S_\omega(\mathcal{A}')$且 $r' = r'_1 r'_2 r'_3 \ldots$，其中 $r'_i = (p'_{i-1}, \gamma_i, \mathbf{v}_i, p'_i)$，对所有 $i \geq 0$， $p'_i \in \{p_i, \hat{}p_i\}$，是 $\mathcal{A}'$在 α上的一个满足强重置接受条件的运行。那么， $r = r_1 r_2 r_3 \ldots$及 $r_i = (p_{i-1}, \gamma_i, \mathbf{v}_i, p_i)$ 是 $\mathcal{A}$在 α上的一个运行。此外， r 满足弱重置接受条件：设 $k_1, k_2, \ldots$表示 r'中所有 p'_{f_i}为接受状态的位置。特别地， $k_1, k_2, \ldots$是无限多个（可能的）重置位置，因此满足弱重置接受条件。因此，$\alpha \in W_\omega(\mathcal{A})$。

□

因此，如果存在一个 SPBA $\mathcal{A}$，使得 $S_\omega(\mathcal{A}) = L$，我们就称 ω-语言 L 重置可识别。我们用 $\mathcal{L}_{\mathsf{Reset}}$ 表示所有重置可识别语言构成的类。类似地，如果存在一个 PPBA $\mathcal{A}$，使得 $P_\omega(\mathcal{A}) = L$，我们就称 L 前缀可识别，并用 $\mathcal{L}_{\mathsf{Prefix}}$ 表示前缀可识别语言构成的类。

4. Schnepsilon 消除

4.1. 面向 *PPBA* 的 ε-消除

现在我们证明，可以消除 ε-PPBA 中的 ε-转移，也就是说，每个 ε-PPBA $\mathcal{A}$ 都等价于一个 PPBA $\mathcal{A}'$。我们的证明分两步进行。第一步，我们证明可以将每个 ε-PPBA 转换为一个等价的 ε-PPBA，其中所有出现的 ε-转移都是 ε-循环。第二步，我们说明如何移除所有 ε-循环，从而得到一个不含 ε-转移的 PPBA。

令 $\Sigma = \{\sigma_1, \ldots, \sigma_k\}$。有限字 $w \in \Sigma^*$ 的*Parikh* 像 是向量$p(w) = (|w|_{\sigma_1}, \ldots\ldots, |w|_{\sigma_k}) \in \mathbb{N}^k$。定义自然地扩展到语言 $L \subseteq \Sigma^*$： $p(L) = \{p(w) \mid w \in L\}$。我们证明中的一个重要要素是 Parikh 定理 [10],（其第一个陈述），内容如下。

定理 4.1（Parikh）。对于每个正则语言 L，集合 $p(L)$ 都是半线性的。

设 $\Gamma \subseteq \mathbb{N}^k$ 为有限集。对于一个字 $w = w_1 \dots w_\ell \in \Gamma^*$，记 $\sum w$ 为 $\sum_{1 \le i \le k} w_i \in \mathbb{N}^k$。对于语言 $L \subseteq \Gamma^*$，定义 $\sum L = \{\sum w \mid w \in L\}$。设 $\Sigma = \{\sigma_1, \dots, \sigma_k\}$ 为字母表。令 $\mathcal{X}_\Sigma = \{\mathbf{e}_i^k \mid 1 \le i \le k\} \subseteq \mathbb{N}^k$ 为 k 维单位向量的集合。于是，Parikh 定理蕴含：对于字母表 $\mathcal{X}_\Sigma$ 上的每个正则语言 L，集合 $\sum L$ 都是半线性的。Klaedtke 和 Ruess 证明了，对于任意有限字母表 $\Gamma \subseteq \mathbb{N}^k$，这一结论同样成立。

引理 4.1（Klaedkte、Ruess，对 [9]，中引理 5 的改写）。设 $\Gamma \subseteq \mathbb{N}^k$ 为有限字母表。则对于字母表 Γ 上的每个正则语言 L，集合 $\sum L$ 都是半线性的。

对于一个 ε-PPBA $\mathcal{A} = (Q, \Sigma, q_0, \Delta, \mathcal{E}, F, C)$，我们将 $\mathcal{A}^\varepsilon$ 定义为 $\mathcal{A}$，其中移除了所有非-ε-转移，即 $\mathcal{A}^\varepsilon = (Q, \Sigma, q_0, \varnothing, \mathcal{E}, F, C)$。此外，对于$p$，$q \in Q$，令 $\mathcal{A}_{p,q}$ 定义为 $\mathcal{A}$，其初始状态为 p，且有一个单个接受状态 q，即$\mathcal{A}_{p,q} = (Q, \Sigma, p, \Delta, \mathcal{E}, \{q\}, C)$。因此 $\mathcal{A}^\varepsilon_{p,q} = (Q, \Sigma, p, \varnothing, \mathcal{E}, \{q\}, C)$。令 $\mathcal{B}_{p,q}$ 为字母表 $\Gamma \subseteq \mathbb{N}^k$ 上的非确定有限自动机，它由 $\mathcal{A}^\varepsilon_{p,q}$ 得到：将每个带标签的 ε-转移替换为仅保留向量的转移，即 $\mathcal{B}_{p,q} = (Q, \Gamma, \{p\}, \Delta', \{q\})$，其中$\Delta' = \{(q_1, \mathbf{v}\ q_2) \mid (q_1, \varepsilon, \mathbf{v}\ q_2) \in \mathcal{E}\}$。于是，以下结论可由引理 4.1 直接得出。

推论 4.1。对于任意 ε-PPBA $\mathcal{A} = (Q, \Sigma, q_0, \Delta, \mathcal{E}, F, C)$ 以及任意 $p, q \in Q$，集合$\sum L(\mathcal{B}_{p,q})$ 都是半线性的。

我们首先证明，可以假设初始状态没有非循环的 ε-转移。

引理 4.2。对于任意 ε-PPBA $\mathcal{A} = (Q, \Sigma, q_0, \Delta, \mathcal{E}, F, C)$，存在一个等价的 ε-PPBA $\mathcal{A}' = (Q \cup \{q_0'\}, \Sigma, q_0', \Delta', \mathcal{E}', F, C')$，使得 q_0' 的所有出向 ε-转移都是 ε-循环。

证明。我们引入一个新的初始状态 q_0'。证明思路如下：对于每个可由一系列 ε-转移后接一个标记为符号 a 的单次转移从 q_0 到达的$q \in Q$，我们引入一个标记为 a 的捷径。为弥补缺失的ε-转移，我们在 q_0'上引入 ε-循环，使得直到 q 的运行可以通过 q_0' 上的 ε-转移，再接上通往 q 的 a-捷径来模拟。q_0' 上新 ε-转移的标记由自动机 $\mathcal{B}_{q_0,q}$ 给出。此外，我们将使用附加计数器，确保不会混淆不同状态 q, q' 对应的转移。下面给出形式化细节。

对于每个状态 $q \in Q$，我们考虑集合 $\sum L(\mathcal{B}_{q_0,q})$，根据推论 4.1，该集合是半线性集。注意，如果无法通过 ε-转移从 q_0 到达 q，则 $\sum L(\mathcal{B}_{q_0,q})$ 为空。否则，$\sum L(\mathcal{B}_{q_0,q})$ 可以写成线性集的有限并，记作 $C_1^{(q)} \cup \dots \cup C_{\ell_q}^{(q)}$，其中 $C_i^{(q)} = \{b_{i,0}^{(q)} + b_{i,1}^{(q)} z_1 + \dots + b_{i,k_{i,q}}^{(q)} z_{k_{i,q}} \mid z_1, \dots, z_{k_{i,q}} \in \mathbb{N}\}$。我们用 $k_{i,q}$ 表示 $C_i^{(q)}$ 的周期向量数量，用 ℓ_q 表示半线性集 $\sum L(\mathcal{B}_{q_0,q})$ 中线性集的数量，并定义 $M = \sum_{q \in Q} \ell_q$。我们固定 π，使其为 $C_i^{(q)}$ 与 $\{1, \dots, M\}$ 之间的任意一个双射。

现在我们介绍 q_0'上新的捷径和 ε-循环。非正式地说，每当 $\mathcal{A}$ 的一次运行以 ε-转移序列 $r_1 \dots r_m r_{m+1}$开始，且 $r_1 \dots r_m \in \mathcal{E}^*$ and$r_{m+1} = (p_m, \alpha_1, \mathbf{v}m, p_{m+1}) \in \Delta$时，自动机 $\mathcal{A}'$ 会猜测该 ε-转移序列的最后一个状态 p_m，并从 q_0' 到 p_{m+1}取一条捷径。由于 $\rho(r_1 \dots r_m) \in C^{(p_m)}i$

对于某个 $i \le \ell_{p_m}$，自动机 $\mathcal{A}'$ 还会猜测集合 $C^{(p_m)}{}_i$，并使用 q_0' 上标记为 $C_i^{(q)}$ 的周期向量与一个维数为 M、仅有一个分量为 1 的 0-1 向量拼接而成的 ε-循环。这些循环取代了所有离开 q_0 的 ε-转移序列，而新的计数器确保我们不会混淆不同线性集的周期向量。

形式化地，我们定义如下。

$$\begin{aligned}\Delta' = &\{(p, a, \mathbf{v}\cdot\mathbf{0}^M, q) \mid (p, a, \mathbf{v}, q) \in \Delta\}\\ &\cup \{(q_0', a, \mathbf{v}\cdot\mathbf{0}^M, q) \mid (q_0, a, \mathbf{v}, q) \in \Delta\}\\ &\cup \{(q_0', a, (\mathbf{v} + b_{i,0}^{(p)})\cdot \mathbf{e}^M_{\pi(C_i^{(p)})}, q) \mid (p, a, \mathbf{v}, q) \in \Delta, 1 \le i \le \ell_p\},\end{aligned}$$

$$\begin{aligned}\mathcal{E}' = &\{(p, \varepsilon, \mathbf{v}\cdot\mathbf{0}^M, q) \mid (p, \varepsilon, \mathbf{v}, q) \in \mathcal{E}, p \ne q_0\}\\ &\cup \{(q_0', \varepsilon, b_{i,j}^{(p)}\cdot \mathbf{e}^M_{\pi(C_i^{(p)})}, q_0') \mid p \in Q, 1 \le i \le \ell_p, 1 \le j \le k_{i,q}\},\end{aligned}$$

and

$$C' = C \cdot \{z\mathbf{e}_i^M \mid z \in \mathbb{N}, 1 \le i \le M\}.$$

我们证明 $\mathcal{A}'$ 与 $\mathcal{A}$ 等价。

$\Rightarrow$ 为了证明 $P_\omega(\mathcal{A}) \subseteq P_\omega(\mathcal{A}')$，取 $\alpha \in P_\omega(\mathcal{A})$，其接受运行为 $r = r_1 r_2 r_3 \ldots$，其中 $r_i = (p_{i-1}, \gamma_i, \mathbf{v}_i, p_i)$。我们分两种情况讨论。

第一种情况，假设 $r_1 = (q_0, \alpha_1, \mathbf{v}\ p_1) \in \Delta$。此时，只需将 r_1 替换为 $r_1' = (q_0', \alpha_1, \mathbf{v}\cdot\mathbf{0}, p_1)$，然后按照 $\mathcal{A}$ 中的方式继续运行，并用零填充所有向量。也就是说，对于 $i \ge 2$，令 $r_i' = (p_{i-1}, \gamma_i, \mathbf{v}_i\cdot\mathbf{0}^M, p_i)$。那么，$r' = r_1' r_2' r_3'$ 就是 $\mathcal{A}'$ 在 α 上的一次运行。我们证明 r' 也是接受的。由于 r 是接受的，存在无穷多个 i，使得 $\rho(r_1 \ldots r_i) \in C$。由于 r' 与 r 基本相等（除了填充的零），因此存在无穷多个 i，使得 $\rho(r_1' \ldots r_i') \in C \cdot \{\mathbf{0}^M\} \subseteq C'$，故 r' 是接受的。

对于第二种情况，假设 $r_1 \in \mathcal{E}$，也就是说，$\mathcal{A}$ 的第一条转移是一个 ε-转移。令 r_m 为初始 ε- 序列的最后一次出现，即 $r_1 \ldots r_m \in \mathcal{E}^*$ 和 $r_{m+1} = (p_m, \alpha_1, \mathbf{v}_m, p_{m+1}) \in \Delta$。注意，对于某个 $i \le \ell_{p_m}$，$\rho(r_1 \ldots r_m) \in C^{(p_m)}{}_i$，因此 $\rho(r_1 \ldots r_m) = b^{(p_m)}{}_{i,0} + b_i^{(p_m)}{}_{,1} z_1 + \cdots + b_{i\,k_i}^{(p_m)}{}_{,p_m} z_{k_i,p_m}$。由于 q_0' 配有标记为 $b^{(p_m)}{}_{i,j} \cdot \mathbf{e}^M_\pi(C_i^{(p_m)})$ q_0' 连接）的 ε-循环，$\mathcal{A}'$ 可以通过将转移 $(q_0', \varepsilon, b^{(p_m)}{}_{i,1} \cdot \mathbf{e}^M_\pi(C_i^{(p_m)})\, q_0')$ 执行 z_1 次、将转移 $(q_0', \varepsilon, b^{(p_m)}{}_{i,2} \cdot \mathbf{e}^M_\pi(C_i^{(p_m)})\, q_0')$ 执行 z_2 次，依此类推，从而用附加计数器将 $\rho(r_1 \ldots r_m) - b^{(p_m)}{}_{i,0}$ 累加起来。最后，我们执行捷径转移 $\hat{r}_n = (q_0', a, (\mathbf{v} + b^{(p_m)}{}_{i,0}) \cdot \mathbf{e}^M_\pi(C_i^{(p_m)})\, p_{m+1})$，以补上缺失的基向量 $b^{(p_m)}{}_{i,0}$。接下来，我们可以像 $\mathcal{A}'$ 中那样继续运行，再次使用补零转移。这得到 $\mathcal{A}'$ 在 α 上的运行 $r' = \hat{r}_1 \ldots \hat{r}_{n-1} \hat{r}_n r'_{m+1} r'_{m+2} \cdots$，其中 $\hat{r}_1 \ldots \hat{r}_{n-1}$ 是一串 ε-循环，$\hat{r}_n$ 是捷径转移，而 $i \ge m+1$ 对应的 r_i' 定义与第一种情况相同。接下来只需证明 r' 是接受的。首先注意，$\rho(\hat{r}_1 \ldots \hat{r}_n) \in \{\rho(r_1 \ldots r_{m+1})\} \cdot \{z\mathbf{e}^M_\pi(C_i^{(p)}) \mid z \in \mathbb{N}\}$，且对所有 $j \ge m+1$，都有 $\rho(r'_{m+1} \cdot r'_j) = \mathbf{v}\cdot\mathbf{0}^M$，其中 $\mathbf{v}$ 为某个值。因此，r 在 r_m 之后发生的所有（无限多个）接受命中都会对应为 r' 的接受命中，所以 r' 是接受的。」

$\Leftarrow$ 反之，为了证明 $P_\omega(\mathcal{A}') \subseteq P_\omega(\mathcal{A})$，令 $\alpha \in P_\omega(\mathcal{A}')$，其接受运行为 $r' = r'_1 r'_2 r'_3 \ldots$，其中 $r'_i = (p_{i-1},\ \gamma_i,\ \mathbf{v}_i \cdot \mathbf{u}_i,\ p_i)$ 对于 $\mathbf{u}_i \in \{z\mathbf{e}_i^M \mid z \in \{0,1\},\ 1 \le i \le M\}$。同样，我们分两种情况讨论。

在第一种情况下，假设 r'_1 的形式为 $(q'_0,\ a,\ \mathbf{v} \cdot \mathbf{0}^M,\ p_1)$。根据构造，我们有 $p_1 \neq q'_0$。特别地，对于所有 $i \ge 1$，都有 $p_i \neq q'_0$。因此，我们可以将 r'_1 替换为$r_1 = (q_0,\ a,\ \mathbf{v}\ p_1)$，并将所有 r'_i 替换为 $r_i = (p_{i-1},\ a,\ \mathbf{v}_i,\ p_i)$，其中 $i \ge 2$。那么，运行 $r_1 r_2 r_3 \ldots$ 是 $\mathcal{A}$ 在 α 上的一个接受运行。

在第二种情况下，假设 r'_1 不是上述形式。那么存在一个（唯一的） $m \ge 1$，使得对于某个 $j \le M$，r'_m 的形式为 $(q'_0,\ \alpha_1,\ \mathbf{v} \cdot \mathbf{e}_j^M,\ p_m)$。令 $C_i^{(q)}$ 为满足 $\pi(C_i^{(q)}) = j$ 的集合（由于 π 是一个双射，该集合是唯一确定的）。特别地，对于所有 $k \ge 1$，都有 $\rho(r'_1 \ldots r'_k) = \mathbf{u} \cdot z\mathbf{e}_j^M$，其中 $z \in \mathbb{N}$ 由 C' 的选取确定。因此，我们可以将 $r'_1 \ldots r'_m$ 替换为 $\mathcal{A}$ 中由 ε 个转移组成的序列 $r_1 \ldots r_n$，使得 $r_1 = (q_0,\ \varepsilon,\ \mathbf{v}1,\ p_1)$、$r_n = (p_{n-1},\ \varepsilon,\ \mathbf{v}_n,\ q)$，且 $\rho(r_1 \ldots r_n) \in C_i^{(q)}$。令$r_{n+1} = (q,\ \alpha_1,\ \mathbf{v}\ p_m)$，其中 $\mathbf{v}$ 是 r'_m 上的向量。注意，根据构造，$r_{n+1} \in \Delta$。那么，$r_1 \ldots r_n r_{n+1} r_{m+1} r_{m+2} \ldots$ 是 $\mathcal{A}$ 在 α 上的一个合法运行，其中对于所有 $i \ge m+1$，都有 $r_i = (p_{i-1},\ \gamma_k,\ \mathbf{v}_i,\ p_i)$。此外，如果对于某个 $i \ge m+1$，$\rho(r'_1, \ldots, r'_i) = \mathbf{u} \cdot z\mathbf{e}_j^M$，那么$\rho(r_1, \ldots, r_{n+1} r_{m+1}, \ldots, r_i) = \mathbf{u}$。由于 r' 是接受的，我们可以得出结论： r 是接受的。 □

> 引理 4.3。设 $\mathcal{A} = (Q、\ \Sigma、\ q_0、\ \Delta、\ \mathcal{E}、\ F、\ C)$ 是一个维数为 d 的 ε-PPBA。存在一个等价的 ε-PPBA $\mathcal{A}'$，其中所有 ε-转移都是 ε-循环。

证明。 根据引理 3.2，不妨设 $\mathcal{A}$ 只有一个接受状态 f，即 $F = \{f\}$；根据引理 4.2，不妨设初始状态的所有出射 ε-转移都是 ε-循环。

开始时，$\mathcal{A}' = (Q'、\ \Sigma、\ q'_0、\ \Delta'、\ \mathcal{E}'、\ F'、\ C')$ 是 $\mathcal{A}$ 的一个副本，该副本将逐步进行修改。首先，我们从 $\mathcal{A}'$ 中移除所有非循环的 ε-转移。

该构造的直观理解如下。我们引入新状态，使得在 $\mathcal{A}$ 在一个无限字上的每次运行中，都可以将在 ε-转移上的每个极大部分运行 $r_i \ldots r_j$（其中 $r_{i-1} = (p_{i-2}, \alpha_{z-1}, \mathbf{v}_{i-1}, p_{i-1})$ $r_i = (p_{i-1}, \varepsilon, \mathbf{v}_i, p_i)$ $r_j = (p_{j-1}, \varepsilon, \mathbf{v}j, p_j)$，以及 $r_{j+1} = (p_j, \alpha_z, \mathbf{v}j{+}1, p_{j+1})$）替换为新状态上依赖于 p_{i-1}、p_j 和 $\sum L(\mathcal{B}_{p_{i-1},p_j})$ 的一系列 ε-循环。有限的部分运行 $r_i \ldots r_j$ 将转化为 $\mathcal{B}_{p_{i-1}、\ p_j}$ 中的一个接受运行，因此 $\rho(r_i \ldots r_j) \in \sum L(\mathcal{B}_{p_{i-1},p_j})$。根据推论 4.1，集合 $\sum L(\mathcal{B}_{p_{i-1},p_j})$ 是半线性的，因而是有限个线性集的并集；每个线性集由一个基向量和一组周期向量定义。我们可以将周期向量编码为 ε-循环，并为基向量设置捷径。如果我们在部分运行 $r_i \ldots r_j$ 中访问接受状态 f，则需要格外谨慎。

让我们继续进行形式化构造。我们遍历状态的所有配对 p, q。如果 $L(\mathcal{B}_{p,q}) = \varnothing$，则不做任何操作，继续处理下一对状态。否则，状态 q 在 $\mathcal{B}_{p,q}$ 中可从 p 到达，并且 $\sum L(\mathcal{B}_{p,q})$ 非空，可以表示为有限个线性集的并集。为便于阅读，我们假设 $\sum L(\mathcal{B}_{p,q}) = C_{p,q}$ 是一个线性集。对于一般情形，我们对并集中的每个线性集分别应用以下构造。下文中，对于任意一对状态 s, t，我们用$b_{s,t}$ 表示 $C_{s,t}$的基向量。

我们向新状态集 Q' 中添加一个新状态 (p, q)（该新状态是状态 p 和 q 组成的元组）。如果 $C_{p,q} = \{b_{p,q} + b_1 z_1 + \cdots + b_k z_k \mid z_1, \ldots, z_k \in \mathbb{N}\}$，我们就向状态 (p, q) 添加带有 $C_{p,q}$ 的周期向量 $b_1, \ldots, b_k$ 标记的 ε-循环。最后，我们向 (p, q) 添加捷径，也就是说，对于每个形如 $(s, a, \mathbf{v}\ p) \in \Delta$ 的转移，我们向 Δ' 添加转移 $(s, a, \mathbf{v} + b_{p,q}, (p, q))$（注意，我们将 $C_{p,q}$ 的基向量 $b_{p,q}$ 加到 $\mathbf{v}$ 上）。同样，对于每个形如 $(q, a, \mathbf{v}\ s) \in \Delta$ 的转移，我们向 Δ' 添加转移 $((p, q)\ a, \mathbf{v}\ s)$。最后，我们通过进一步添加转移 $((p, q)\ a, \mathbf{v} + b_{s,t}, (s, t))$ 来连接新状态：对于每个转移 $(q, a, \mathbf{v}\ s) \in \Delta$ 和 $t \in Q$，只要 $(s, t) \in Q'$ 即可（仍然假设 $C_{s,t}$ 是线性的；否则，我们会有多个 (s, t) 副本，每个副本都通过如上的转移连接）。

彻底应用这一过程后，$\mathcal{A}'$ 可以模拟 $\mathcal{A}$ 的每一次运行，也就是说，$\mathcal{A}$ 中出现的所有 ε-转移都已被化简为若干个配有 ε-循环的新状态。然而，如果接受状态 f 在 $\mathcal{B}_{p中q}$ 可以从 p 到达，我们必须考虑到，上述过程可能跳过了该接受状态，从而导致遗漏接受命中。为避免这种情况，我们需要添加额外的新状态。

我们再次遍历所有状态对 $p, q \in Q$。不过，这次我们考虑自动机 $\mathcal{A}^{\varepsilon}_{p,f}$ 和 $\mathcal{A}^{\varepsilon}_{f,q}$。与上文一样，令 $\mathcal{B}_{p,f}$ 和 $\mathcal{B}_{f,q}$ 分别为这样两个非确定有限自动机：它们的字母表是 $\mathbb{N}^d$ 的子集，分别由 $\mathcal{A}^{\varepsilon}_{p,f}$ 和 $\mathcal{A}^{\varepsilon}_{f,q}$ 仅保留 ε-转移的向量后得到。根据推论 4.1，集合 $\sum L(\mathcal{B}_{p,f})$ 和 $\sum L(\mathcal{B}_{f,q})$ 是半线性的。

如果这两个集合都非空，我们可以将它们分别写成线性集的有限并。同样，为了便于阅读，我们假设 $\sum L(\mathcal{B}_{p,f}) = C_{p,f}$ 和 $\sum L(\mathcal{B}_{f,q}) = C_{f,q}$ 是线性集。对于一般情况，我们对 $L(\mathcal{B}_{p,f})$ 的每个线性集与 $L(\mathcal{B}_{f,q})$ 的每个线性集的组合，分别应用以下构造。

设 $C_{p,f} = \{b_{p,f} + b_1 z_1 + \cdots + b_k z_k \mid z_1, \ldots, z_k \in \mathbb{N}\}$ 和 $C_{f,q} = \{b_{f,q} + c_1 z_1 + \cdots + c_\ell z_\ell \mid z_1, \ldots, z_\ell \in \mathbb{N}\}$。我们向 Q' 中添加一个新的接受状态 (p, f, q)，并为其配备标记为 $C_{p,f}$ 的周期向量 $b_1, \ldots, b_k$ 所对应的 ε 个循环。此外，对于每个形如 $(r, a, \mathbf{v}\ p) \in \Delta$ 的转移，我们向 Δ' 添加一条捷径 $(r, a, \mathbf{v} + b_{p,f}, (p, f, q))$。

现在让我们将 (p, q)-状态与刚刚引入的状态连接起来。对于每个转移 $(q, a, \mathbf{v}\ s) \in \Delta$，我们添加入向转移 $((p, q)\ a, \mathbf{v} + b_{s,f}, (s, f, t))$，其中所有 $t \in Q$ 都满足 Δ'。

下一步，我们引入 (p, f, q)-状态的出边。不幸的是，情况更加复杂，因此我们还必须引入更多状态 (p, f, q, s) 和 (p, f, q, s, t)。其思路是，这些状态分别充当 s 和 (s, t) 的副本，但还额外配有标记为 $C_{f,q}$ 的周期向量的 ε-循环。我们不能简单地将这些循环添加到 (p, f, q)，因为这样可能会导致 $\mathcal{A}'$ 中出现接受命中，而在 $\mathcal{A}$ 中使用这些循环（它们对应于离开 f 的 ε-序列的向量）是不可能的。我们也不能直接忽略它们，因为要适当地模拟 $\mathcal{A}$ 的运行，就必须用到这些循环。因此需要这些副本，这样我们就能使用这些循环，同时避免产生伪接受命中。

形式上，对于所有 $p, q, s \in Q$，我们插入一个新状态 (p, f, q, s)；对于所有 $p, q \in Q$ 和 $(s, t) \in Q'$，我们插入一个新状态 (p, f, q, s, t)。对于每个转移 $(q, a, \mathbf{v}\ s) \in \Delta$，我们向 Δ' 添加以下转移：$((p, f, q)\ a, \mathbf{v} + b_{f,q}, (p, f, q, s))$，以及 $((p, f, q)\ a, \mathbf{v} + b_{f,q} + b_{s,t}, (p, f, q, s, t))$，其中 $t \in Q$ 满足存在 $(s, t) \in Q'$。

最后，我们按如下方式连接新状态 (p, f, q, s) 和 (p, f, q, s, t)。首先，(p, f, q, s) 和 (p, f, q, s, t) 配备了标记为 $C_{f,q}$ 的周期向量的 ε-循环。此外，(p, f, q, s) 拥有 s 的所有出边；也就是说，对于每个转移 $(s, a, \mathbf{v}\ t) \in \Delta'$，我们还向 Δ' 添加一个转移 $((p, f, q, s)\ a, \mathbf{v}\ t)$。类似地，$(p, f, q, s, t)$ 获得了 Δ' 中 (s, t) 的所有出边。此外，(p, f, q, s, t) 获得了 (s, t) 的所有 ε-循环。

现在，$\mathcal{A}'$ 中所有剩余的 ε-转移都是循环，我们也完成了 $\mathcal{A}'$的构造。」

我们证明 $\mathcal{A}'$ 与 $\mathcal{A}$等价。下文中，ε-序列指包含至少一个非循环 ε-转移的极大 ε-转移序列。

$\Rightarrow$ 为证明 $P_\omega(\mathcal{A}) \subseteq P_\omega(\mathcal{A}')$，设 $\alpha \in P_\omega(\mathcal{A})$，其接受运行是 $r = r_1 r_2 r_3 \ldots$。如果 r 中不存在 ε-序列，那么证明即告完成（因为 r 也是 $\mathcal{A}'$ 在 α上的一个运行）。

否则，我们通过逐步替换 ε-序列，构造 $\mathcal{A}'$ 在 α 上的一个接受运行 r'。令 i 为满足 $r_i \ldots r_j$是一个 ε-序列的最小值。注意，由于 q_0没有非循环 ε-转移，因此有 $i > 1$。令 $r_{i-1} = (p_{i-2}, \alpha_{z-1}, \mathbf{v}_{i-1}, p_{i-1})$、$r_i = (p_{i-1}, \varepsilon, \mathbf{v}_i, p_i)$ $r_j = (p_{j-1}, \varepsilon, \mathbf{v}j, p_j)$，以及 $r_{j+1} = (p_j, \alpha_z, \mathbf{v}^{j+1}, p_{j+1})$。类似地，令 $k > j$为满足 $r_k \ldots r_\ell$是一个 ε-序列的最小值，也就是说，$r_k \ldots r_\ell$ 是 r 中的第二个 ε-序列（这样的 k 可能不存在，我们将在下文明确处理这种情况）。注意，我们有 $\rho(r_i \ldots r_j) \in \sum L(\mathcal{B}_{p_{i-1}, p_j})$，且 $\rho(r_k \ldots r_\ell) \in \sum L(\mathcal{B}_{p_{k-1}, p_\ell})$。

我们区分以下几种情形（及其组合）。

- 存在 f_1，满足 $i \le f_1 \le j$，使得我们在 r_{f_1} 中有一个接受命中（F），或者没有接受命中（N）。回顾一下，我们只有一个接受状态 f，此处 f_1 表示接受命中的位置。

- 我们有 $k = j + 2$，也就是说，在两个序列 $r_i \ldots r_j$ 和 $r_k \ldots r_\ell$ 之间只有一次非 ε 转移（1）；或者 $k > j + 2$，也就是说，在这两个序列之间至少有两次非 ε 转移（2）。

- 存在 f_2，满足 $k \le f_2 \le \ell$，使得我们在 r_{f_2} 中有一个接受命中（F），或者没有接受命中；或者 k 甚至不存在（N）。

因此，我们总共考虑八种情形。

- 情形 (F1N)。这意味着我们需要处理第一个ε-序列中的接受命中（位于位置 f_1），但第二个 ε-序列中没有接受命中（此处我们假设第二个序列存在；另一种情形在 (F2N) 中处理），并且这两个序列之间只有一次非 ε 转移。

我们用 $\hat{r}_{i-1} = (p_{i-2}, \alpha_{z-1}, \mathbf{v}_{i-1} + b_{p_{i-1}, f}, (p_{i-1}, f, p_j)) \in \Delta'$替换 r_{i-1}。由于位置 f_1处有一个接受命中，因此我们有 $\rho(r_1 \ldots r_{f_1}) = \rho(r_1 \ldots r_{i-1}) + \rho(r_i \ldots r_{f_1}) \in C$。特别地，我们有 $\rho(r_i \ldots r_{f_1}) \in C_{p_{i-1}, p_{f_i}}$。根据构造，在 (p_{i-1}, f, p_j) 上存在 ε-循环，其标记为 $C_{p_{i-1}, p_{f_i}}$的周期向量。因此，我们可以用一系列 ε-循环 $\hat{r}_i \ldots \hat{r}_m$替换 $r_i \ldots r_{f_1}$，这些循环位于 (p_{i-1}, f, p_j) 上，并满足 $\rho(\hat{r}_i \ldots \hat{r}_m) = \rho(r_i \ldots r_{f_1}) - b_{p_{i-1}, f}$。由于 $b_{p_{i-1}, f}$ 已添加到 $\hat{r}_{i-1}$中，序列 $\hat{r} = r_1 \ldots r_{i-2} \hat{r}_{i-1} \hat{r}_i \ldots \hat{r}_m$是 $\mathcal{A}'$ 在 α 上的一次运行的前缀，且满足 $\rho(\hat{r}) \in C$。

我们描述如何从此处继续运行。我们取转移 $\hat{}r_{m+1} = ((p_{i-1}, f, p_j)\ \alpha_z, \mathbf{v}_{j+1} + b_{f,p_j} + b_{p_{j+1},p_\ell}, (p_{i-1}, f, p_j, p_{j+1}, p_\ell)) \in \Delta$（回忆一下，$p_{j+1} = p_{k-1}$）。由于 $(p_{i-1}, f, p_j, p_{j+1}, p_\ell)$ 配有 ε-循环，其标记为 C_{f,p_j}的周期向量，也配有 $C_{j+1,\ell}$的周期向量，因此我们可以用一系列 ε-循环 $\hat{}r_{m+2} \dots \hat{}r_n$ 替换部分运行 $r_{m+1} \dots r_j$；这些循环位于 $(p_{i-1}, f, p_j, p_{j+1}, p_\ell)$ 上，并满足 $\rho(\hat{}r_{f_1+1} \dots \hat{}r_j) = \rho(r_{f_1+1} \dots r_j) - b_{f,p_j}$。我们还可以用一系列 ε-循环 $\hat{}r_{n+1} \dots \hat{}r_o$替换部分运行 $r_{j+2} \dots r_\ell$；这些循环位于 $(p_{i-1}, f, p_j, p_{j+1}, p_\ell)$ 上，并满足 $\rho(\hat{}r_{j+2} \dots \hat{}r_\ell) = \rho(r_{j+2} \dots r_\ell) - b_{p_{j+1},p_\ell}$。由于 b_{p_{j+1},p_ℓ}和 b_{f,p_j} 已添加到 $\hat{}r_{m+1}$中，部分运行 $\hat{}r_1 \dots \hat{}r_m \dots \hat{}r_n \dots \hat{}r_o$与$r_i \dots r_\ell$等价，具体而言，两次运行都至少有一次接受命中，读取 $\alpha_{z-1}\alpha_z$，具有相同的扩展 Parikh 像，并且都“适配”整体运行 r；这是因为 $(p_{i-1}, f, p_j, p_{j+1}, p_l)$ 与 p_l具有相同的出边（包括可能存在的捷径）。

现在继续处理 r中的下一个 ε-序列。

- 情形（F1F）。我们采用与（F1N）完全相同的替换。因此，第二个序列中会失去一个接受命中，不过这并不成问题，因为第一个序列中仍有一个接受命中，而且之后还会有无限多个接受命中。

- 情形（F2N）。这与（F1N）类似，但我们选择$\hat{}r_{m+1} = ((p_{i-1}, f, p_j)\ \alpha_z, \mathbf{v}_{j+1} + b_{f,p_j}, (p_{i-1}, f, p_j, p_{j+1}))$。由于此时无需考虑集合$C_{k,\ell}$，我们只需将 $r_{f_1+1} \dots r_j$ 替换为由 ε-循环组成的匹配序列 $r_{m+2} \dots r_n$，其所在对象为 $(p_{i-1}, f, p_j, p_{j+1})$。

  此时，我们继续进行，仿佛身处 p_{j+1}中。请注意，如果第二个序列不存在，我们就已经完成了。

- 情形（F2F）与（F2N）相同。

- 情形（N1F）。在这里，我们用 $r_{i-1}\hat{}r_{i-1} = (p_{i-2}, \alpha_{z-1}, \mathbf{v}_i + b_{p_{i-1},p_j}, (p_{i-1}, p_j)) \in \Delta'$ 替换。由于 $\rho(r_i \dots r_j) \in C_{p_{i-1},p_j}$，我们可以用一串 ε-循环 $\hat{}r_i \dots \hat{}r_m$ 替换这段部分运行，它们位于 (p_{i-1}, p_j) 上，且满足 $\rho(\hat{}r_i \dots \hat{}r_m) = \rho(r_i \dots r_j) - b_{p_{i-1},p_j}$。由于 b_{p_{i-1},p_j} 已经加入 $\hat{}r_{i-1}$，序列 $\hat{}r_{i-1}\hat{}r_i \dots \hat{}r_m$ 与 $r_{i-1}r_i \dots r_j$ 等价，因为这两种运行都读取 α_{z-1}，具有相同的 Parikh 像，并且都适合嵌入整体运行 r；这是因为我们可以完全按照 p_j 中的方式，从 (p_i, p_j) 继续运行，因此接下来处理下一个 ε- 序列。

- 剩余的情形（N1N）、（N2F）和（N2N）与（N1F）相同。

所有不在 ε-序列中的接受命中都一一对应。这就完成了正向方向的证明。」

$\Leftarrow$为证明 $P_\omega(\mathcal{A}') \subseteq P_\omega(\mathcal{A})$，设 $\alpha \in P_\omega(\mathcal{A}')$ 有一个接受运行 $r' = r'_1 r'_2 r'_3 \dots$。如果出现在 r 中的所有状态都属于 $\mathcal{A}$的状态集 Q，则结论成立，因为该运行也是 $\mathcal{A}$的一个接受运行。

否则，我们逐步构造 $\mathcal{A}$ 在 α 上的一个接受运行 r。设 i 是满足以下条件的最小值：$r'_i = (p'_{i-1},\ \alpha_z,\ \mathbf{v}'_i,\ p'_i)$ 包含一个不属于 Q的状态。由于 $\mathcal{A}'$ 的所有（接受）运行都从 q_0 开始（它属于 Q），因此有 $p'_i \notin Q$。我们区分两种情形。

如果对于某个 $s, t \in Q$，有 $p'_i = (s, t)$，那么根据 Δ' 的选择，我们有 $\mathbf{v}'_i = \mathbf{v}_i + b_{s,t}$。令 $j \geq i$ 为满足对所有 $i \leq k \leq j$，r'_k 都是 (s, t) 上的一个 ε-环的最大值，也就是说，它具有如下形式：$((s, t)\ \varepsilon, \mathbf{v}'_k, (s, t))$。

根据 (s,t) 的语义，在 $\mathcal{A}$ 中存在一个从 s 开始并在 t 结束的 ε-序列 $r_i \dots r_n$。具体而言，我们有 $r_i = (s, \varepsilon, \mathbf{v}_i, p_i)$ 和 $r_n = (p_{n-1}, \varepsilon, \mathbf{v}_n, t)$，并且存在一个转移 $r_{i-1} = (p_{i-2}, \alpha_z, \mathbf{v}_i, s) \in \Delta$，使得 $\mathbf{v}_i = \mathbf{v}'_i$，因此 $\rho(r_{i-1} r_i \dots r_n) = \rho(r'_i \dots r'_j)$。此外，注意到 $p_{i-2} = p'_{i-1}$，因此 $r_{i-1} r_i \dots r_n$ 与 $r'_i \dots r'_j$ 等价：两次运行都读取 α_z，具有相同的扩展 Parikh 像，并且都适用于整体运行 r'，因为我们可以从 t 开始，完全按照 (s,t) 中的方式继续运行。

若对于某些 $s,t \in Q$，有 $p'_i = (s,f,t)$，则根据 Δ' 的选择，我们有 $\mathbf{v}'_i = \mathbf{v}_i + b_{s,f}$。令 $j \geq i$ 为满足以下条件的最大值：对于所有 $i \leq k \leq j$，r'_k 都是 (s,f,t) 上的一个 ε-循环，也就是说，r'_k 是形如 $((s,f,t)\ \varepsilon, \mathbf{v}'_k, (s,f,t))$ 的转移，并且在 r'_j 中存在一个接受命中（若不存在接受命中，则令 j 为该序列中最后一个 ε-循环）。

根据 (s,f,t) 的语义，$\mathcal{A}$ 中存在一个 ε-序列 $r_i \dots r_m \dots r_n$，它从 s 开始，访问 f，并在 t 结束。具体而言，我们有 $r_i = (s, \varepsilon, \mathbf{v}_i, p_i)$、$r_m = (p_{m-1}, \varepsilon, \mathbf{v}_m, f)$、$r_n = (p_{n-1}, \varepsilon, \mathbf{v}_n, t)$，并且存在一个转移 $r_{i-1} = (p_{i-2}, \alpha_z, \mathbf{v}_i, s) \in \Delta$，使得 $\mathbf{v}_i = \mathbf{v}'_i$。特别地，若 r'_j 中存在一个接受命中，我们可以选择 m，使得 $\rho(r_{i-1} r_i \dots r_m) = \rho(r'_i \dots r'_j)$，因此 r_m 中也存在一个接受命中。根据 Δ' 的选择，r'_j 后面跟着一个（可能为空的）由 (s,f,t) 上的 ε-循环组成的序列，之后跟着一个形如 $\delta_1 = ((s,f,t)\ \alpha_{z+1}, \mathbf{v}+b_{f,t}, (s,f,t,p))$ 或 $\delta_2 = ((s,f,t)\ \alpha_{z+1}, \mathbf{v}+b_{f,t}+b_{p,q}, (s,f,t,p,q))$ 的转移，其中 $p,q \in Q$ 为某些值。这个由 (s,f,t) 上的 ε-循环组成的序列，与 $\mathcal{A}$ 中一个从 f 开始并在此结束、且具有相同扩展 Parikh 像的 ε-转移序列相对应，因此我们据此将其替换。

现在考虑下一个转移。在第一种情况下，我们假设它的形式为 δ_1。回想一下，(s,f,t,p) 配备了若干标记为 $C_{f,t}$ 的周期向量的 ε-循环，以及可能存在的 p 在 $\mathcal{A}$中的 ε-循环。这意味着该转移后面可能跟着一个（也可能为空的）序列，其中包含在 (s,f,t,p) 上的 ε-循环。不失一般性，我们假设这些循环按如下方式排序：首先出现所有标记为周期向量的 ε-循环，例如 $\hat{}r_1 \dots \hat{}r_o$，随后是可能存在的 p的 ε-循环（这没有问题，因为我们只需交换 ε-循环；并且由于 (s,f,t,p) 非接受，因此无需顾及任何接受命中）。根据构造，我们有 $\rho(\hat{}r_1 \dots \hat{}r_o) = \rho(r_{m+1} \dots r_n) - b_{f,t}$。由于 $b_{f,t}$ 已经加入 δ_1，我们将 $\delta_1\ \hat{}r_1 \dots \hat{}r_o$ 替换为 $r_{m+1} \dots r_n r_{n+1}$，其中 $r_{n+1} = (t, \alpha_{z+1}, \mathbf{v}\ p)$。由于 p 配备了 (s,f,t,p) 的所有出边（包括可能存在的 ε-循环，但不包括其他 ε-转移），我们可以在 $\mathcal{A}$ 中以与在 $\mathcal{A}'$ 中相同的方式继续运行。

最后，我们假设下一步转移的形式为 δ_2。这种情况类似，唯一的区别是，$(s,\ f,\ t,\ p,\ q)$“表现”得像$(p,\ q)$，也就是说，$(s,\ f,\ t,\ p,\ q)$拥有$(p,\ q)$的所有出边。因此，我们从这里继续，按照第一种情况的做法处理这种形式的状态。

同样，所有使用 Q 中状态的接受命中都一一对应。这就完成了反向方向的证明。□

现在我们继续消除剩余的 ε-循环。我们需要以下引理来对有限字自动机进行 ε-消除。

引理 4.4. [Klaedtke 和 Ruess 的定理 22 [9], 的改写] 对于任意一个维数为 d、定义在有限词上的 PA$\mathcal{A} = (Q, \Sigma, q_0, \Delta, F, C)$（带有 ε-转移），都存在一个等价的、无 ε 的 PA $\mathcal{A} = (Q, \Sigma, q_0, \Delta, F, C)$，其定义在相同的状态集上，维数为$d + |Q| - 1$。

引理 4.5. 设 $\mathcal{A} = (Q, \Sigma, q_0, \Delta, \mathcal{E}, F, C)$ 是一个维数为 d 的 ε-PPBA，其中所有出现的 ε-转移都是 ε-循环。存在一个等价的 PPBA $\mathcal{A}'$。

证明。 我们先概述证明思路。直观地说，我们将把自动机分成两部分。我们将猜测会被无限多次访问的状态集合 S 。在自动机的第一部分中，我们将处理只被有限次访问的状态集合。为此，我们应用引理 4.4 的构造，并将所有接受状态 $\hat{}$改为非接受状态，从而得到自动机 $\mathcal{A}$。我们将以非确定性方式切换到第二部分，在那里验证对于某个集合 $S \subseteq Q$ ，恰好集合 S 中的状态会被无限多次访问。我们沿用 Klaedtke 和 Ruess [9] 针对有限词的思路：由于向量的添加时机和顺序无关紧要，我们可以通过适当修改半线性集来模拟 ε-循环。直观而言，我们可以从 C中“减去”与该循环对应的半线性集，而不是在某个状态上进行 ε-循环。形式上，对于每一种可能的猜测 S ，我们都将构造一个自动机 $\mathcal{A}_S$ ，并从 $\mathcal{A}$适当地使用捷径 $\hat{}$。我们将为每个 $\mathcal{A}_S$使用一个半线性集，因此形式上构造的是一个 MPPBA。最后，我们应用引理 3.2，将这个自动机转换为等价的 PPBA。现在进入正式证明。

根据引理 3.2，我们可以假设 $\mathcal{A}$ 只有一个接受状态，记作 $F = \{f\}$。我们首先将 $\mathcal{A}$ 视为一个作用于有限词上的概率自动机，并将根据引理 4.4 对每个向量填充 d 个零后得到的无 ε 自动机记作 $\mathcal{A}$（也就是说， $\mathcal{A}$ 的维数为 $2d + |Q| - 1$，而不是 $d + |Q| - 1$）。我们将 $\mathcal{A}$ 的半线性集记作 C。

现在，对于每个至少包含 f 的非空状态子集 $S \subseteq Q$ ，我们构造一个维数为 d 的 MPPBA $\mathcal{A}_S$ ，如下所示。任意排列 S 的状态，记作 $S = \{s_1, \ldots, s_m\}$。 S 是一个候选集，表示恰好由那些将被无限次访问的状态组成的集合。我们通过将 $\mathcal{A}$ 中所有指向 f 的转移捷径化至 $\mathcal{A}_S$ 的初始状态，从而将 $\mathcal{A}$ 与所有 $\mathcal{A}_S$ 相连。这样，我们便可以非确定性地从 $\mathcal{A}$ 切换到某个 $\mathcal{A}_S$。

自动机 $\mathcal{A}_S$ 由 $m + 1$ 个 $\mathcal{A}$ 副本以及一个新状态 q_S组成； q_S是 $\mathcal{A}_S$的初始状态，也是 $\mathcal{A}_S$唯一的接受状态。下文中，我们将 $\mathcal{A}_S$ 的第 i 个副本称为 $\mathcal{A}_S$的第 i 个 层 。通过规定只有访问状态 s_i后才能从第 i 层切换到第 $(i + 1)$ 层，这 m 层确保我们会无限次访问所有状态 s_1、.....、 s_m。从最后一层，我们可以通过捷径到达新状态 q_S，从而“等待”下一次转移将我们带到 f 且计数器的值落入半线性集的时刻。此外， q_S 的出边基本上与接受状态 f的出边相同，但它们会通向 s_1对应的层。

形式化地，令 $\mathcal{A}_S = ((S \times \{1, \ldots, m + 1\}) \cup \{q_S\},\ \Sigma,\ q_S,\ \Delta_S,\ \{q_S\},\ C_S)$，其中

$$
\begin{aligned}
\Delta_S = {} & \{((s_i, k), a, \mathbf{0}^{d+|Q|-1} \cdot \mathbf{v}, (s_j, k)) \mid (s_i, a, \mathbf{v}, s_j) \in \Delta, k \leq m + 1\} \\
& \cup \{((s_i, i), a, \mathbf{0}^{d+|Q|-1} \cdot \mathbf{v}, (s_j, i + 1) \mid (s_i, a, \mathbf{v}, s_j) \in \Delta, i \leq m\} \\
& \cup \{((s_i, m + 1), a, \mathbf{0}^{d+|Q|-1} \cdot \mathbf{v}, q_S) \mid (s_i, a, \mathbf{v}, f) \in \Delta\} \\
& \cup \{(q_S, a, \mathbf{0}^{d+|Q|-1} \cdot \mathbf{v}, (s_i, 1) \mid (f, a, \mathbf{v}, s_i) \in \Delta\}.
\end{aligned}
$$

对于两个维数为 d 的半线性集 C_1，C_2，令 $C_1 - C_2 = \{\mathbf{u} \in \mathbb{N}^d \mid \mathbf{u} + \mathbf{v} \in C_1$，其中某个 $\mathbf{v} \in C_2\}$。正如 Klaedtke 和 Ruess 所示，集合 $C_1 - C_2$ 仍然是半线性集（注意，它可以用 Presburger 算术定义）。对于 $q \in Q$，令 $C_q = \sum L(B_{q,\ q})$ 并选择 $C_S(q_S)$ $= \{\mathbf{0}^{d+|Q|-1}\} \cdot (\vec{C} - C_{s_1} - \cdots - C_{s_m})$，其中 $\vec{C}$的定义与 C相同，但不包含任何基向量。

最后，令 $\mathcal{A}' = (Q', \Sigma, q_0, \Delta', \{q_S \mid S \subseteq Q\}, C')$ 为 $\hat{}\mathcal{A}$ 与所有 $\mathcal{A}_S\hat{}$ 的（不相交）并集，并添加转移 $(q, a, \mathbf{v}\ q_S)$，其中每个 $(q, a, \mathbf{v}\ f) \in \Delta$ 且 $C'(q_S) = C + C_S(q_S)$。我们声称 $\mathcal{A}'$ 与 $\mathcal{A}$ 等价。

$\Rightarrow$我们首先证明 $P_\omega(\mathcal{A}) \subseteq P_\omega(\mathcal{A}')$。设 $\alpha \in P_\omega(\mathcal{A})$ 有一个接受运行 $r = r_1 r_2 r_3 \ldots$，其中 $r_i = (p_{i-1}, \gamma_i, \mathbf{v}_i, p_i)$。令 S 为在 r 中无限次出现的状态集合，并令 i 为满足以下条件的最小集合：r_i 中存在一次接受命中，且对于所有 $j \geq i$，都有 $p_j \in S$。令 $u = \alpha_1 \ldots \alpha_z \in \Sigma^*$ 为访问 p_i 时已读入的 α 的前缀，并令 $\beta = \alpha_{z+1}\alpha_{z+2} \cdots \in \Sigma^\omega$，即 $\alpha = u\beta$。

首先注意，根据定义，$u \in L(\mathcal{A})$作为 $r_1 \ldots r_i$ 是 $\mathcal{A}$ 在 u 上的接受运行。根据引理 4.4，自动机 $\hat{}\mathcal{A}$ 与 $\mathcal{A}$ 在有限词上的概率自动机意义下等价，因此 $\mathcal{A}$ 在 u上也存在一条接受运行，记作 $\hat{}r = \hat{}r_1 \ldots \hat{}r_z$。注意，$\hat{}r_z = (p_{z-1},\ \alpha_z,\ \mathbf{v}_z,\ f)$，因此 $\mathcal{A}'$有一个形如 $\delta = (p_{z-1},\ \alpha_z,\ \mathbf{v}_z,\ q_S)$的转移，所以我们可以用 $\hat{}r_1 \ldots \hat{}r_{z-1}\delta$来模拟 $\mathcal{A}'$ 的这段部分运行。

到目前为止，我们只使用了前 $d+|Q|-1$ 个计数器，现在将只使用最后 d 个计数器。根据 $C(q_S)$的定义，接下来只需证明 $\beta \in P_\omega(\mathcal{A}_S)$（注意，我们已经有了一次接受命中，因此已从 $C(q_S)$中移除了基向量）。注意，在 r 中可以安全地删除任何 ε-转移，而不会破坏运行，因为所有 ε-转移都是循环。令 $j_0 > i$ 为满足 $r_{j_0} \in \Delta$的最小值，例如 $r_{j_0} = (f,\ \alpha_{z+1},\ \mathbf{v}_{j_0},\ p_{j_0})$。根据构造，$\mathcal{A}'$中有一个形如$(q_S,\ \alpha_{z+1},\ \mathbf{0}^{d+|Q|-1} \cdot \mathbf{v}_{j_0},\ (p_{j_0},\ 1))$的转移，我们利用它继续在 $\mathcal{A}'$ 中的运行（即忽略 f上的 ε-循环）。

现在令 $j_{m+1} > j_0$ 为最小值，使得存在 $j_0 < j_1 < j_2 < \cdots < j_m < j_{m+1}$ 满足：对于所有 $1 \leq k \leq m$，都有 $p_{j_k} = s_k$，且 $r_{j_{m+1}}$ 是一次接受命中。对于所有 $j_{k-1} < \ell < j_k$，$1 \leq k \leq m+1$，我们考虑转移 r_ℓ。如果 $r_\ell \in \mathcal{E}$，我们就直接将其忽略。否则，将 r_ℓ 替换为 $r'_\ell = ((p_{\ell-1}, k)\ \gamma_\ell, \mathbf{0} \cdot \mathbf{v}_\ell, ((p_\ell, k))$。此外，对于 $1 \leq k \leq m$，我们将 r_{j_k} 替换为 $r'_{j_k} = ((p_{j_k-1}, k)\ \gamma_{j_k}, \mathbf{0} \cdot \mathbf{v}_{j_k}, (p_{j_k}, k+1))$，最后将 $r_{j_{m+1}}$ 替换为 $r'_{j_{m+1}} = ((p_{j-1}, m+1)\ \mathbf{0} \cdot \mathbf{v}_j, q_S)$。

注意，对于部分运行 $r' = r'_{j_0} \ldots r'_{j_{m+1}}$，我们有 $\rho(r') \in C(q_S)$，因为 $\rho(r_{i+1} \cdots r_{j_{m+1}}) \in C$；这是由于我们只删除了 ε-转移（这些状态在构造中都已见过），且 $C(q_S)$ 是据此定义的。

现在我们可以迭代这一构造，从而得到 $\mathcal{A}_S$ 在 β 上的一个接受运行。至此，正向部分得证。

$\Leftarrow$现在我们证明 $P_\omega(\mathcal{A}') \subseteq P_\omega(\mathcal{A})$。设 $\alpha \in P_\omega(\mathcal{A}')$，其接受运行为 $r' = r'_1 r'_2 r'_3 \ldots$，其中 $r'_i = (p'_{i-1},\ \alpha_i,\ \mathbf{v}'_i,\ p'_i)$。设 $j_1 < j_2 < \ldots$ 为满足以下条件的位置：对于所有 $i \geq 1$，r'_{j_i} 都是一个接受命中。我们将通过证明一系列断言来继续。

断言 1：$\vec{\mathbf{0}} \in C \subseteq C - C_{s_1} - \cdots - C_{s_m}$。这直接源于 C没有基向量这一事实。

断言 2：对于所有 $k < j_1$，我们都有 $p'_k \in Q$（回想一下，Q 是属于 $\mathcal{A}$ 的 $\mathcal{A}'\hat{}$ 的状态集的子集）。因此 $\rho(r'_1 \ldots r'_{j_1}) \in C \cdot \{\mathbf{0}\}$。根据以下事实，该断言显然成立：作为 $\mathcal{A}'$ 的子自动机，$\hat{}\mathcal{A}$ 没有接受状态；而从 $\hat{}\mathcal{A}$ 到某个 $\mathcal{A}_S$ 的转移会到达接受状态 q_S，也就是说 $r'_{j_1} = (p'_{j_1-1},\ \alpha_{j_1},\ \mathbf{v}'_{j_1},\ q_S)$，其中 $p'_{j_1-1} \in Q$。

断言 3：对于每个 $j \geq j_1 + 1$，$\rho(r'_{j_1+1} \dots r'_j) \in \{\mathbf{0}\} \cdot \mathbb{N}^d$。因此，$\mathbf{v}'_j$ 可以写成 $\mathbf{0} \cdot \mathbf{v}_j$，其中 $\mathbf{v}_j \in \mathbb{N}^d$。此外，每个 p'_j 要么是 q_S，要么具有 (s_k, ℓ) 的形式，其中 $k \leq m$ 和 $\ell \leq m+1$。根据 $\mathcal{A}'$ 的构造，这显然成立，因为在第一次接受命中之后，我们已经切换到某个 $\mathcal{A}_S$。若 $p'_j = q_S$，定义 $p_j = f$；若 $p'_j = (s_k, \ell)$，则定义 $p_j = s_k$。类似地，令 $r_j = (p_{j-1}, \alpha_j, \mathbf{v}_j, p_j)$。

断言 4：在每两个连续的 $j_\ell, j_{\ell+1}$ 之间，都会访问 S 的所有状态。这是因为要再次访问 q_S 中的 $\mathcal{A}_S$，就必须遍历 $\mathcal{A}_S$ 的所有层。

断言 5：$\rho(r'_{j_\ell+1} \dots r'_{j_{\ell+1}}) \in \{\mathbf{0}\} \cdot (\vec{C} - C_{s_1} - \cdots - C_{s_m})$，设 $\rho(r'_{j_\ell+1} \dots r'_{j_{\ell+1}}) = \mathbf{v}_{j_\ell} - \mathbf{v}_{s_1} - \cdots - \mathbf{v}_{s_m}$，其中 $\mathbf{v}_{s_i} \in C_{s_i}$。根据断言 3 和 $\mathcal{A}_S$ 的构造，这一点成立，因为 ε-循环已被移除。令 $j_\ell < k_1 < k_2 < \cdots < k_m < j_{\ell+1}$ 为部分运行中改变层的位置，即 $r'_{k_i} = ((s_i, i)\ \alpha_{k_i}, \mathbf{v}'_i, (p_{k_i}, i+1))$。根据断言 4，这些位置确实存在。

断言 6：对于每个 $\mathbf{v}_{s_i} \in C_{s_i}$，都存在一个由 ε-循环 λ_i 组成的序列，它们位于 s_i 中、在 $\mathcal{A}$ 中，且满足 $\rho(\lambda_i) = \mathbf{v}_{s_i}$。这由 C_{s_i} 的选择直接得到。

断言 7：$\mathcal{A}$ 的运行 $\bar{r}_\ell = r_{j_\ell+1} \dots \lambda_1 r_{k_1} \dots \lambda_2 r_{k_2} \dots \lambda_m r_{k_m} \dots r_{j_{\ell+1}}$ 与 $\mathcal{A}'$ 中 $\bar{r}'_\ell = r'_{j_\ell} \dots r'_{j_{\ell+1}}$ 等价，具体而言，$\rho(\bar{r}_\ell) = \rho(\bar{r}'_\ell) - \mathbf{v}_{s_1} - \cdots - \mathbf{v}_{s_m} \in C - C_{s_1} - \cdots - C_{s_m}$，因此 $\rho(\bar{r}_\ell) \in \vec{C}$。该断言由断言 5 和断言 6 推出。

我们已准备好完成证明。根据引理 4.4，存在 $\mathcal{A}$ 中的 $\hat{}\hat{}\alpha_1 \dots \alpha_{j_1}$ 的一个接受运行 $\hat{}r$（将其视为有限词上的概率自动机）。根据构造，我们有 $\rho(\hat{}r) \in C \cdot \{\mathbf{0}\}$。我们按如下方式构造 $\mathcal{A}$ 的运行 r。在 r' 中，我们用 $\hat{}r$ 替换 $r'_1 \dots r'_{j_1}$。现在，对于 $\ell \geq 1$，我们用根据断言 7 构造的 $\bar{r}_\ell$ 替换 $r'_{j_\ell+1} \dots r'_{j_{\ell+1}}$。注意，这是 $\mathcal{A}$ 的一个有效运行。此外，根据断言 7，对于所有 $\ell \geq 1$，$\mathcal{A}$ 在位置 $(|\hat{}r| - j_1) + j_\ell$ 处都有接受命中。因此，r 在 $\mathcal{A}$ 中是接受的。」

我们已经证明，MPPBA $\mathcal{A}'$ 与 $\mathcal{A}$等价。结合引理 3.2，我们完成该引理的证明。

□

结合前面的引理，我们得出本节的主要定理（见a另见图 2）。

定理 4.2. 可由 ε-PPBA 识别的 ω-语言类与可由 PPBA 识别的 ω-语言类相同。换言之，ε-PPBA 可以进行 ε-消除。

定理 4.3. 设 L 是一个 ω-语言。那么，以下陈述等价。

1. L 可由 PPBA 识别。2. L 可由 MPPBA 识别。3. L可由 ε-PPBA 识别。4. L 可由 ε-MPPBA 识别

.

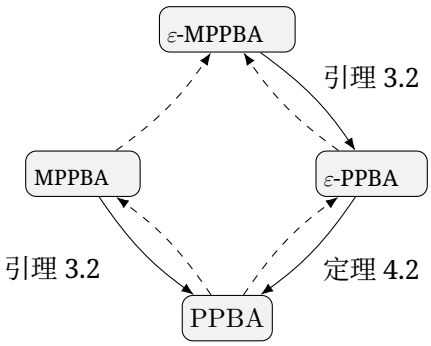

图 2：前缀模型的等价性。虚线表示显然的包含关系。

4.2. SPBA 的 ε-消除

最后，我们证明重置模型允许进行 ε-消除。本节中，由于我们讨论的是 SPBA，其中每次访问接受状态都会触发重置，因此我们称其为重置状态，而不是接受状态。

引理 4.6. 对于每个 ε-SPBA $\mathcal{A}$，都存在一个与之等价的 SPBA。换言之，SPBA 支持 ε-消除。

证明。设 $\mathcal{A} = (Q, \Sigma, q_0, \Delta, \mathcal{E}, F, C)$ 是一个维数为 d 的 ε-SPBA。我们不失一般性地假设 q_0 没有入向转移（这可以通过引入 q_0 的一个新副本来实现）。此外，我们假设 $F \neq \varnothing$ （否则 $S_\omega(\mathcal{A}) = \varnothing$）。任意排列 Q 的状态，记作 $Q = \{q_0, \dots, q_{n-1}\}$。我们构造一个与之等价、维数为 $d+n$ 的 SPBA $\mathcal{A}' = (Q', \Sigma, q_0, \Delta', F', C')$。起初，$\mathcal{A}'$ 是 $\mathcal{A}$ 的一个副本（暂时保留 ε-转移），随后逐步对其进行修改。新计数器的作用是跟踪已访问的状态（自上次重置以来）。因此，最初我们按如下方式修改转移：对于每个转移 $(q_i, \gamma, \mathbf{v}\ q_j) \in \Delta \cup \mathcal{E}$，我们将 $\mathbf{v}$ 替换为 $\mathbf{v} \cdot \mathbf{e}_j^n$。

˜令 $p, q \in Q$。假设存在一个转移序列 $\lambda = r_1 \dots r_j \dots r_k \in \mathcal{E}^* \Delta \mathcal{E}^*; 1 \leq j \leq k \leq 2n+1$，其中

- $r_j = (p_{j-1}, a, \mathbf{v}j, p_j) \in \Delta$，并且
- $r_i = (p_{i-1}, \varepsilon, \mathbf{v}_i, p_i) \in \mathcal{E}$ 对所有 $i \neq j, i \leq k$，
- 使得 $p_0 = p, p_k = q$，并且 $p_i \neq p_\ell$ 对于 $i, \ell \leq j$ 以及 $p_i \neq p_\ell$ 对于 $i, \ell \geq j$，并且
- 所有内部状态均为非重置状态，即 $p_i \notin F$ 对所有 $0 < i < k$。

˜ ˜接着，我们引入捷径 $(p, a, \rho(\lambda)\ q)$，其中 $\rho(\lambda)$ 已针对新的计数器计算得出，这些计数器用于跟踪 λ 中的 p_i 是否已被访问过；也就是说，该序列中对应于 p_i 的计数器值非零。

令 $p, q \in Q$。如果一个（可能为空的）序列 $\lambda = r_1 \dots r_k \in \mathcal{E}^*$; $k \geq 0$ 满足$r_i = (p_{i-1}, \varepsilon, \mathtt{v}_i, p_i)$ 且 $p_0 = p, p_k = q$，并且所有内部状态均为非重置状态，即对所有 $0 < i < k$ 都有 $p_i \notin F$，则称其为从 p 到 q 的无重置 ε-序列。无重置 ε-路径是满足对 $i \neq j$ 有 $p_i \neq p_j$ 的无重置序列。注意，从 p 到 q 的无重置 ε- 路径集合是有限的，因为每条路径的长度均不超过 $n-1$。如果存在从 p 到 q 的无重置 ε-序列 r，使得 $\rho(r) \in C$，其中 $\rho(r)$ 在 $\mathcal{A}$ 中计算得出，则称二元组 (p, q) 为一个 C-对。

令 $S = (f_1, \dots, f_\ell)$ 为一个非空的、两两不同的重置状态序列（注意，这意味着 $\ell \leq n$）。若每个 (f_i, f_{i+1}) 都是一个 C-对，则称 S 为 C-序列 。

对于所有满足以下条件的 $p, q \in F$和 C-序列 S：若 $p \in F$则 $p = f_1$，且若 $q \in F$则 $q = f_\ell$，我们引入一个新状态 (p, S, q)。我们将 (p, S, q) 加入 F'，也就是说，将这些新状态设为重置状态。状态 (p, S, q) 表示自动机的一段部分运行：仅包含从 p 开始、按该顺序访问 S 的重置状态并最终到达 q 的 ε-转移。

注意，以下步骤中，根据上述观察，我们只会引入有限多个转移，因此不再在每一步重复说明这一点。令 $p, q \in Q$ 和 $S = (f_1, \dots, f_\ell)$ 为一个 C-序列。对于每个形如 $(s, a, \mathtt{v}\ p) \in \Delta$ 的转移，我们插入新转移 $\{(s, a, \mathtt{v} + \rho(\lambda), (p, S, q))$；$\mid \lambda$ 是一条从 p 到 $f_1\}$ 再到 Δ' 的无重置 ε-路径。类似地，对于每个形如 $(q, a, \mathtt{v}\ t) \in \Delta$ 的转移，我们插入新转移 $\{((p, S, q)\ a, \mathtt{v} + \rho(\lambda)\ t)$；$\mid \lambda$ 是一条从 f_ℓ 到 $q\}$ 再到 Δ' 的无重置 ε-路径。同样，这个集合是有限的。此外，令 $p', q' \in Q$ 和 $S' = (f'_1, \dots, f'_k)$ 为一个 C-序列。对于每个序列 $\lambda = \lambda\delta\lambda'$，其中 λ 是一条从 f_ℓ 到 q 的无重置 ε-路径，$\delta = (q, a, \mathtt{v}\ p')$，且 λ' 是一条从 p' 到 f'_1 的无重置 ε-路径，我们将捷径 $\big((p, S, q)\ a、\ \rho(\tilde{\lambda})、(p', S', q')\big)$ 加入 Δ'。

最后，我们以类似的方式连接初始状态 q_0（请回想，我们假设 q_0 没有入向转移，特别是没有循环）。对于每个转移 $(p, a, \mathtt{v}\ q) \in \Delta$，以及每个满足以下性质的 C-序列 $S = (f_1, \dots, f_l)$：(q_0, f_1) 是一个 C-对，且存在一条从 f_ℓ 到 p 的无重置 ε-路径 λ，我们针对每条这样的路径 λ 引入转移 $(q_0, a, \rho(\lambda) + \mathtt{v}\ q)$。此外，对于每个满足以下条件的 C-序列 $S' = (f'_1, \dots f'_k)$：存在一条从 q 到 f'_1 的无重置 ε-路径 λ'，我们针对所有这样的路径 λ, λ' 和 $t \in Q$ 引入转移 $\big(q_0, a, \rho(\lambda) + \mathtt{v} + \rho(\lambda'), (q, S', t)\big)\hat{}$。进一步，对于每条从 q_0 到 p 的无重置 ε-路径 λ，我们针对所有 $t \in Q$ 引入转移 $(q_0, a, \rho(\lambda) + \mathtt{v} + \rho(\lambda'), (q, S', t))\hat{}$。担心此时我们可能引入过多转移的读者应当记住，如果不存在从 f'_k 到 t 的无重置 ε-路径，(q, S', t) 就没有出向转移。最后，我们删除所有 ε-转移。

我们按照 Klaedtke 和 Ruess 的构造 [9]，定义 C'。对于每个 $q \in Q$，我们定义 $C_q = \sum \hat{}\mathcal{B}_{q,q}$，其中 $\hat{}\mathcal{B}_{q,q}$ 定义为 $\mathcal{B}_{q,q}$，但不包含任何接受状态，也就是说，对于合适的字母表 $\Gamma \subseteq \mathbb{N}^d$，有 $\hat{}\mathcal{B}_{q,q} = (Q \setminus F, \Gamma, q, \{(p, \mathtt{v}\ p') \mid (p, \varepsilon, \mathtt{v}\ p') \in \mathcal{E}, p, p' \notin F\}, \{q\})$。然后 $C' = \{\mathtt{v} \cdot (x_0, \dots, x_{n-1}) \mid \mathtt{v} + \mathtt{u} \in C,\ \mathtt{u} \in \sum_{x_i \geq 1} C_{q_i}\}$。这样，如果计数器的值大于或等于 1，也就是说，该状态已被访问，我们就减去 C_{q_i} 对应于 q_i 的部分。至此，构造完成。

现在我们证明 $\mathcal{A}'$ 与 $\mathcal{A}$ 等价。一个方向上，我们使用适当的捷径来压缩运行；另一个方向上，则相应地将其展开。

$\Rightarrow$为了证明 $S_\omega(\mathcal{A}) \subseteq S_\omega(\mathcal{A}')$，令 $\alpha \in S_\omega(\mathcal{A})$，其接受运行是 $r = r_1 r_2 r_3 \dots$。如果 r 中没有 ε-转移，我们就完成了（因为 r 也是 $\mathcal{A}'$ 在 α 上的接受运行）。

否则，我们通过逐步替换 r 中的极大 ε-序列，构造 $\mathcal{A}'$ 在 α 上的接受运行 r'。令 i 为使得 $r_i \dots r_j$ 是极大 ε-序列的最小值。令 $r_i = (p_{i-1}, \varepsilon, \mathrm{v}_i, p_i)$ $r_j = (p_{j-1}, \varepsilon, \mathrm{v}j, p_j)$，且 $r_{j+1} = (p_j, \alpha_z, \mathrm{v}j{+}1, p_{j+1})$。可能有 $i = 1$，即运行 r 以离开 q_0 的一条 ε-转移开始。否则，$i > 1$，且我们可以写成 $r_{i-1} = (p_{i-2}, \alpha_{z-1}, \mathrm{v}_{i-1}, p_{i-1})$。通过允许空序列，我们可以假设总是存在第二个（可能为空的）极大 ε-序列 $r_{j+2} \dots r_k$，它紧接在 r_{j+1} 之后开始。我们区分以下几种（组合）情况。

- $r_i \dots r_j$ 中至少有一个状态是重置状态，即存在一个位置 $i - 1 \le \ell \le j$，使得 $p_\ell \in F$（F）或非（N）。
- $r_{j+2} \dots r_k$ 中至少有一个状态是重置状态，即存在一个位置 $j + 1 \le \ell' \le k$，使得 $p_{\ell'} \in F$（F）或非（N）。如果 $r_{j+2} \dots r_k$ 为空，则属于情形（N）。

因此，我们总共考虑四种情况。

• 情形（NN）。也就是说，$r_i \dots r_k$ 中没有重置状态。注意，ε-序列 $r_i \dots r_j$ 可以分解为一个 ε-路径和若干 ε-循环，如下所示。如果对于所有 $i \le i_1 < j_1 \le j$，都有 $p_{i_1} \neq p_{j_1}$，则问题已解决，因为 $r_i \dots r_j$ 已经是一个 ε-路径。否则，令 $i_1 \ge i$ 为满足存在 $j_1 > i_1$ 使得 $p_{i_1} = p_{j_1}$ 的最小值；也就是说，$r_{i_1+1} \dots r_{j_1}$ 是一个 ε-循环。如果 $r_i \dots r_{i_1} r_{j_1+1} \dots r_j$ 是一个 ε-路径，则问题已解决。否则，令 $i_2 > j_1$ 为满足存在 $j_2 > i_2$ 使得 $p_{i_2} = p_{j_2}$ 的最小值；也就是说，$r_{i_2+1} \dots r_{j_2}$ 是一个 ε-循环。接着，如果 $r_i \dots r_{i_1} r_{j_1+1} \dots r_{i_2} r_{j_2+1} \dots r_j$ 是一个 ε-路径，则问题已解决。否则，我们可以重复这一论证，得到一组 ε-循环 $r_{i_1+1} \dots r_{j_1}, \dots, r_{i_m+1} \dots r_{j_m}$（其中某个 m），以及一个将 $r_i \dots r_j$ 划分开的 ε-路径 $\hat{r}_{i,j} = r_i \dots r_{i_1} r_{j_1+1} \dots r_{i_m} r_{j_m+1} \dots r_j$。现在注意，$\rho(r_{i_1+1} \dots r_{j_1}) + \dots + \rho(r_{i_m+1} \dots r_{j_m}) \in C_{p_{i_1}} + \dots + C_{p_{i_m}}$。我们可以对 ε-序列 $r_{j+2} \dots r_k$ 进行同样的分解，将其分解为一组 ε-循环和一个 ε-路径 $\hat{r}_{j+2,k}$。根据 Δ' 的构造，存在一条捷径 $\delta = (p_{i-1}, \alpha_z, (\rho(\hat{r}_{i,j}) + \mathrm{v}_{j+1} + \rho(\hat{r}_{j+2,k})) \cdot \hat{\mathrm{e}}^n, p_k)$，其中 $\hat{\mathrm{e}}^n$ 是一个 n 维向量，用于统计出现在 $\hat{r}_{i,j}$ 和 $\hat{r}_{j+2,k}$ 中的状态以及状态 p_{j+1}。根据 Δ' 和 C' 的构造，我们可以减去在 $r_i \dots r_k$ 中经过的所有 ε-循环，因此可以用 δ 替换 $r_i \dots r_k$，从而精确模拟 $\mathcal{A}$ 的行为。

• 情况（NF）。也就是说，$r_i \dots r_j$ 中不存在重置状态，但 $r_{i+2} \dots r_k$ 中至少存在一个重置状态（特别地，该序列不为空）。令 ℓ_1、……、ℓ_m 表示 $r_{i+2} \dots r_k$ 中重置状态的位置，并令 $\ell_0 < \ell_1$ 为满足 ℓ_0 是重置状态的最大值（该位置早于 r_i；如果不存在这样的 ℓ_0，则令 $\ell_0 = 0$），也就是说，ℓ_0 是位置 ℓ_1 处重置之前的最后一次重置的位置。由于 r 是一个接受运行，序列 $S = (\ell_1, \dots, \ell_m)$ 是一个 C-序列（我们可以假设 S 中的所有状态两两不同，否则就存在一个重置环，而该重置环可以忽略）。与前一种情况相同，我们可以将 ε-序列 $r_i \dots r_j$ 划分为一个 ε-路径 $\hat{r}_{i,j}$ 以及一组 ε-环，并从 C 中减去这些环。同样地，我们可以将序列 $r_{j+2} \dots r_{\ell_1}$ 划分为一个 ε-路径 $\hat{r}_{j+2,\ell_1}$ 以及具有相同性质的 ε-环。根据 Δ' 的构造，存在一条捷径 $(p_{i-1}, a\rho(\hat{r}_{i,j}) + \mathrm{v}j{+}1, p_{j+1})$，因此存在一个转移 $\delta = (p_{i-1}, a, \rho(\hat{r}_{i,j}) + \mathrm{v}_{j+1} + \rho(\hat{r}_{j+2,\ell_1}), (p_{j+1}, S, p_k))$（注意，当 $i = 1$ 时同样如此）。因此，我们用 δ 替换 $r_i \dots r_k$。特别地，$\rho(r_{\ell_0+1} \dots r_{i-1}\delta)$ 可以通过从 $\rho(r_{\ell_0+1} \dots r_{\ell_1})$ 中减去在该部分运行中访问过的所有 ε-环来获得

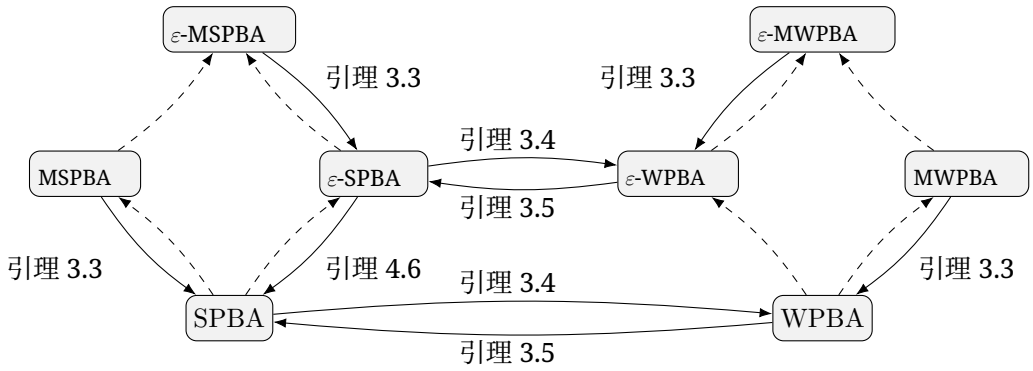

图 3：重置模型的等价性。虚线表示显然的包含关系。

运行。此外，注意，$\rho(r_{\ell_1+1}\dots r_{\ell_2}) \in C$、.....、$\rho(r_{\ell_{m-1}+1}\dots r_{\ell_m}) \in C$只取决于自动机，而不取决于输入字。由于计数器在$r_{\ell_m}$中被重置，我们可以从 δ 继续运行，其方式与在r_k中相同，使用来自 Δ'的适当转移来添加向量 $\rho(\hat{}r_{\ell_m+1}, k)$，从而满足接受条件。

• 情形（FN）。与（NF）类似，但这次我们将 $r_{i-1}r_i\dots r_j$ 替换为一个适当的转移，进入形如 $(p_{i-2}, \alpha_z, \mathbf{v}, (p_{i-1}, S, p_j))$ 的状态，其中 C-序列 S和向量 $\mathbf{v}$ 适当，随后通过一条捷径到达 p_k。如果 $i = 0$（即在读取第一个符号之前进入一个 C-序列），我们会使用专门为 q_0引入的转移。

• 情形（FF）。与（FN）和（NF）类似，但我们从形如 (p_{i-1}, S, p_j) 的状态转移到形如 (p_{j+1}, S', p_k) 的状态，其中 C-序列 S, S' 适当，并且再次遵循情形 $i = 0$。」

$\Leftarrow$ 为证明 $S_\omega(\mathcal{A}') \subseteq S_\omega(\mathcal{A})$，我们展开构造中引入的捷径和 (p, S, q)-状态。设 $\alpha \in S_\omega(\mathcal{A}')$ 的接受运行为 $r' = r'_1 r'_2 r'_3 \dots$。我们将每个转移 $r'_i \in \Delta' \setminus \Delta$（即未出现在 $\mathcal{A}$ 中的转移）替换为 $\mathcal{A}$ 中相应的转移序列。设 $i \geq 1$ 是满足 r'_i 为 $\Delta' \setminus \Delta$ 中一个转移的最小值。

我们区分 r'_i 的形式，并证明可能的形式与正向中的各个情形一一对应。

• 情形 (NN)。$r'_i = (p, a, \rho(\lambda)\ q)$ 为捷径，即 $\tilde{\lambda} \in \mathcal{E}^*\Delta\mathcal{E}^*$，对应于情形 (NN)。特别地，$r$ 中没有接受状态。设 $k < i$ 是 r'_i之前最后一次重置的位置，k' 是 r'_i之后第一次重置的位置；如果 r'_i 转移到一个重置状态，则 $k' = i$。由接受条件可知，对于某个基于计数器值的集合 $Q' \subseteq Q$，有 $\rho(r'_{k+1}\dots r'_{k'}) \in C - (\sum_{q\in Q'} C_q)$。因此，我们可以用部分运行 λ 替换 r'_i，并在 Q' 中的一些状态上填入可能的 ε-循环。

• 情况（NF）。$r'_i = (s, a, \mathbf{v} + \rho(\lambda), (p, S, q))$ 的情况中，若 $S = (f_1, \dots f_\ell)$ 是一个 C-序列，则存在转移 $\delta = (s, a, \mathbf{v}\ p) \in \Delta$，且 λ 是一条从 p 到 f_1 的无重置 ε-路径，这对应于情况（NF）。根据 C-序列的定义，存在一系列 ε 转移 r_{f_1,f_ℓ}，它们位于 $\mathcal{A}$ 中，从 f_1 开始，到 f_ℓ 结束，并按此顺序经过重置状态 f_1 到 f_ℓ，使得重置接受条件

在每次访问某个接受状态时都成立。于是我们可以将 r'_i 替换为 $\delta\lambda r_{f_1,\, f_\ell}$，并且可以像前一种情况那样，根据 λ 的状态计数器，再用一些 ε-循环填充。请注意，此时我们还没有展开从 f_ℓ 到 q 的路径，因为这取决于运行 r' 如何继续（下一两种情况将对此进行处理）。

- 情形（FN）。若 $r'_i = ((p, S, q)\ a, \mathrm{v}\ +\rho(\lambda)\ t)$满足 $S = (f_1, \ldots f_\ell)$是一个C-序列，且存在转移 $\delta = (q, a, \mathrm{v}\ t)$，$\in \Delta$并且 λ 是从 f_ℓ到 q的一条无重置 ε-路径，则对应于情形（FN）。与前一种情形类似，我们可以将 r'_i替换为 $\lambda\delta$，必要时再根据 λ的状态计数器添加一些 ε-环。如果 $i = 1$，该转移也可能具有如下形式：$r'_1 = (q_0, \alpha_1, \rho(\lambda)\ +\mathrm{v}\ t)$，其中 S 是一个 C-序列，且具有如下性质：(q_0, f_1)是一个 C-对。于是，存在如上所述的一系列 ε-转移$r_{q_0,\, f_\ell}$，位于 $\mathcal{A}$ 中。因此，我们改为用$r_{q_0, f_\ell}\lambda\delta$ （可能带有 ε-环）替换 r'_1。

- 情形（FF）。情形 $r'_i = ((p, S, q)\ a, \rho(\lambda), (p', \tilde{S'}, q')$，满足 $S = (f_1, \ldots f_\ell)$ 和 $S' = (f'_1, \ldots, f'_k)$ 是 C-序列，存在转移 $\delta = (q, a, \mathrm{v}\ p') \in \tilde{\Delta}$和 $\lambda = \lambda\delta\lambda'$，其中 λ 是从 f_ℓ 到 q 的无重置 ε-路径，λ' 是从 p' 到 f'_1的无重置 ε-路径，对应于情形（FF）。此情形基本上是前述情形的并集。与情形（RF）一样，在 $\mathcal{A}$中存在一个由 ε-转移组成的序列$r_{f'_1, f'_k}$。因此，我们用 $\lambda r_{f'_1, f'_k}$替换 r'_i（可能包含 ε-循环）。如果 $i = 1$，该转移也可能具有如下形式：$r'_1 = (q_0, \alpha_1, \rho(\lambda)\ +\mathrm{v}\ +\rho(\lambda'), (p', S', q'))$，满足 (q_0, f_1)是一个 C-对。此时，如上所述，在 $\mathcal{A}$ 中存在一个由 ε-转移组成的序列 r_{q_0, f_ℓ}，我们用 $r_{q_0, f_\ell}\lambda r_{f'_1, f'_k}$ 替换 r'_1 （可能包含 ε-循环）。」

这就完成了引理的证明。 □

作为本节的总结，我们证明了所有重置模型的等价性，具体如以下定理所述。示意见图 3。

定理 4.4。 设 L 是一个 ω-语言。则以下陈述等价。

1. L 可由 SPBA 识别。
2. L 可由 MSPBA 识别。
3. L可由 ε-SPBA 识别。
4. L可由 ε-MSPBA 识别。
5. L 可由 WPBA 识别。
6. L 可由 MWPBA 识别。
7. L可由 ε-WPBA 识别。
8. L可由 ε-MWPBA 识别。

回顾一下，$\mathcal{L}_{\mathsf{Reset}}$ 表示所有可由 SPBA 识别的 ω-语言的类；根据上述定理，该类也可以等价地使用任意重置模型来定义。

5. PPBA 与盲计数器机的等价性及其闭包性质 s

本节证明，PPBA 描述的 ω-语言类与 Fernau 和 Stiebe [5] 引入的（同步）盲计数器机相同。一个盲 k-计数器机 (计数器机)是一个五元组 $\mathcal{M} = (Q,\ \Sigma,\ q_0,\ \Delta,\ F)$，其中 Q、Σ、q_0 和 F 的定义与非确定有限自动机中的相同，而 $\Delta \subseteq Q \times (\Sigma \cup \{\varepsilon\}) \times \mathbb{Z}^d \times Q$ 是整数标记转移的集合。特别地，Δ 的转移标记为可能含负数的整数向量。此外，允许使用ε-转移。

无限词 $\alpha = \alpha_1\alpha_2\alpha_3\ldots$ 的一个配置是形如 $c = (p, \alpha_1\ldots\alpha_i, \alpha_{i+1}\alpha_{i+2}\ldots, \mathbf{v})$ $\in Q \times \Sigma^* \times \Sigma^\omega \times \mathbb{Z}^k$ 的元组，其中 $i \geq 0$。如果 $c' = (q, \alpha_1\ldots\alpha_{i+1}, \alpha_{i+2}\ldots, \mathbf{v} + \mathbf{u})$ 且 $(p, \alpha_{i+1}, \mathbf{u}\ q) \in \Delta$，或者 $c' = (q, \alpha_1\ldots\alpha_i, \alpha_{i+1}\alpha_{i+2}\ldots, \mathbf{v} + \mathbf{u})$ 且 $(p, \varepsilon, \mathbf{u}\ q) \in \Delta$，那么配置 c 可以推导出配置 c'，记作 $c \vdash c'$。$\mathcal{M}$ 接受无限词 α，当且仅当存在一个无限的配置推导序列 $c_1 \vdash c_2 \vdash c_3 \vdash \ldots$，其起始配置为 $c_1 = (q_0, \varepsilon, \alpha, \mathbf{0})$，并且对于无限多个 i，都有 $c_i = (p_i, \alpha_1\ldots\alpha_j, \alpha_{j+1}\alpha_{j+2}\ldots, \mathbf{0})$，其中 $p_i \in F$；此外，对于所有 $j \geq 1$，该序列中都存在形如 $(p, \alpha_1\ldots\alpha_j, \alpha_{j+1}\alpha_{j+1}\ldots, q)$ 的配置，其中 $p, q \in Q$。也就是说，如果计数器为 0 时无限次访问接受状态，并且在某个时刻读取了 α 的每个符号，那么该字就被接受。我们将由 $\mathcal{M}$ 识别的 ω-语言定义为 $L_\omega(\mathcal{M}) = \{\alpha \in \Sigma^\omega \mid \mathcal{M}$ 接受 $\alpha\}$。

我们证明，可以将每个计数器机有效地转换为等价的 ε-PPBA；根据定理 4.2，它与一个 PPBA 等价。对于反方向，我们将使用ε-转移，也就是说，我们的结果并未给出适用于计数器机的 ε-消除方案。据我们所知，尚不清楚不含 ε-转移的计数器机是否与含 ε-转移的计数器机具有同等的能力。

> 引理 5.1。对于每个计数器机 $\mathcal{M}$，都存在一个等价的 ε-PPBA $\mathcal{A}$。

证明。令 $\mathcal{M} = (Q, \Sigma, q_0, \Delta, F)$ 为一个 k-计数器机。对于向量 $(x_1, \ldots, x_k) \in \mathbb{Z}^k$，我们定义向量 $\mathbf{v}^{\pm} = (x_1^+, \ldots x_k^+, x_1^-, \ldots x_k^-) \in \mathbb{N}^{2k}$ 如下：如果 x_i 为正，则 $x_i^+ = x_i$ 且 $x_i^- = 0$。否则，$x_i^+ = 0$且 $x_i^- = |x_i|$。我们构造一个维数为 $2k$ 的等价 ε-PPBA $\mathcal{A} = (Q, \Sigma, q_0, \Delta', \mathcal{E}', F, C)$，其中 $\Delta' = \{(p, a, \mathbf{v}^{\pm}, q) \mid (p, a, \mathbf{v}\ q) \in \Delta\}$，且 $\mathcal{E}' = \{(p, \varepsilon, \mathbf{v}^{\pm}, q) \mid (p, \varepsilon, \mathbf{v}\ q) \in \Delta\}$。最后，我们取 $C = \{(x_1, \ldots, x_k, x_1, \ldots, x_k) \mid x_i \in \mathbb{N}\}$。现在很容易验证，$L_\omega(\mathcal{M}) = P_\omega(\mathcal{A})$。□

接下来，我们证明可以通过引入一个新状态，将每个带有 d 个计数器的 PPBA 转换为等价的 d-计数器机。

> 引理 5.2. 对于每个 PPBA $\mathcal{A}$，都存在一个等价的计数器机 CM $\mathcal{M}$。

证明。令 $\mathcal{A} = (Q, \Sigma, q_0, \Delta, F, C)$ 为一个维数为 d 的 PPBA。根据引理 3.1，且由于计数器机对并集封闭，我们可以假设 C 是线性的。我们构造一个 d-计数器机 $\mathcal{M}$，按如下方式模拟 $\mathcal{A}$：$\mathcal{M}$ 由 $\mathcal{A}$ 的一个副本组成，其中接受状态带有额外的 ε-转移，这些转移以 C 的周期向量的相反数为标签。我们只需考虑一次 C 的基向量，因此引入一个新的初始状态 q_0'，并添加一条从 q_0' 到 q_0 、以 C的基向量的相反数为标签的 ε-转移。注意，向量 $\mathbf{v}$ 属于 $C = \{b_0 + b_1 z_1 + \cdots + b_\ell z_\ell \mid z_1, \ldots, z_\ell\}$ 当且仅当$\mathbf{v} - b_1 z_1 - \cdots - b_\ell z_\ell - b_0 = \mathbf{0}$ 对某个 z_i 成立。直观来说，$\mathcal{M}$ 在这些副本中计算向量 $\mathbf{v}$

中的 Q，并猜测接受状态中的 z_i。我们构造 $\mathcal{M} = (Q \cup \{q_0'\}, \Sigma, q_0', \Delta', F)$，其中

$$\Delta' = \Delta \cup \{(q_0', \varepsilon, -b_0, q_0\} \cup \{(q_f, \varepsilon, -b_i, q_f) \mid q_f \in F, i \leq \ell\}.$$

现在很容易验证，$P_\omega(\mathcal{A}) = L_\omega(\mathcal{M})$。 □

由前两个引理可得，PPBA 与计数器机等价。

推论 5.1。可由 PPBA 识别的 ω-语言类与可由计数器机识别的ω-语言类相同。

正如 Fernau 和 Stiebe 所示，可由计数器机识别的 ω-语言类对并集封闭，并且对与 ω-正则语言的交集封闭，但对交集和补集不封闭。因此，可由 PPBA 识别的 ω-语言类具有相同的封闭性质。此处我们简要指出，可由 SPBA 识别的 ω-语言类也对并集以及与 ω-正则语言的交集封闭，但对交集和补集不封闭（这可以通过非常相似的论证证明）。

观察 5.1。可由 PPBA 识别的 ω-语言类与可由 SPBA 识别的ω-语言类对并集封闭，并且对与 ω-正则语言的交集封闭，但对交集和补集不封闭。

我们通过证明可由 PPBA 识别的和可由 SPBA 识别的 ω-语言类也在与 Parikh 可识别语言进行左连接下封闭，来结束本节。我们证明以下辅助引理，它简化了这一结论的证明，也简化了后续各节中的一些证明。

引理 5.3. 设 $\mathcal{A} = (Q, \Sigma, q_0, \Delta, F, C)$ 是一个（针对有限词的）概率自动机，且 $L(\mathcal{A}) = L$。那么，我们可以在多项式时间内构造一个概率自动机 $\mathcal{A}_N$，使其具有以下性质。

- $L(\mathcal{A}_N) = L \setminus \{\varepsilon\}$。
- $\mathcal{A}_N$ 只有一个接受状态 f，并且 f 没有出边。

我们称 $\mathcal{A}_N$ 为规范化的。

证明。设 $\mathcal{A}_N = (Q \cup \{f\}, \Sigma, q_0, \Delta_N, \{f\}, C)$，其中 $\Delta_N = \Delta \cup \{(p, a, \mathbf{v}\ f) \mid (p, a, \mathbf{v}\ q) \in \Delta, q \in F\}$，也就是说，$\mathcal{A}_N$ 会猜测输入字中最后一个符号的位置，并在读到最后一个符号时转移到新的接受状态（该状态没有出边）。由于至少需要读取一个符号才能到达 f，因此我们永远不会接受空词。显然，按上述方式构造的 $\mathcal{A}_N$ 满足规范化的条件，且可在多项式时间内计算得出。 □

引理 5.4. 设 $L_1 \subseteq \Sigma^*$ 是 Parikh 可识别的，且 $L_2 \subseteq \Sigma^\omega$ 是 SPBA 可识别的（可由 PPBA 识别的）。那么 $L_1 L_2$ 是 SPBA 可识别的（可由 PPBA 识别的）。

证明。我们先给出关于 SPBA 的论证。设 $\mathcal{A}_1 = (Q_1, \Sigma, q_1, \Delta_1, F_1, C_1)$ 是一个维数为 d_1、满足 $L(\mathcal{A}_1) = L_1$ 的概率自动机，且 $\mathcal{A}_2 = (Q_2, \Sigma, q_2, \Delta_2, F_2, C_2)$ 是一个维数为 d_2、满足 $S_\omega(\mathcal{A}_2) = L_2$ 的 SPBA。不失一般性，我们假设 $\varepsilon \notin L$，因此假设 $\mathcal{A}_1$ 是规范化的，特别是 $F_1 = \{f\}$（若 $\varepsilon \in L$，我们使用 $L_1 L_2 = (L_1 \setminus \{\varepsilon\}) L_2 \cup L_2$ 以及对并运算封闭性）。

我们选取 SPBA $\mathcal{A} = (Q_1 \cup Q_2, \Sigma, q_1, \Delta, F_1 \cup F_2, C)$，其中

$$\begin{aligned}\Delta = &\{(p, a, \mathbf{v} \cdot \mathbf{0}^{d_2}, q) \mid (p, a, \mathbf{v}, q) \in \Delta_1\} \\ &\cup \{(p, a, \mathbf{0}^{d_1} \cdot \mathbf{v}, q) \mid (p, a, \mathbf{v}, q) \in \Delta_2\} \\ &\cup \{(f, a, \mathbf{0}^{d_1} \cdot \mathbf{v}, q) \mid (q_2, a, \mathbf{v}, q) \in \Delta_2\}\end{aligned}$$

以及 $C = C_1 \cdot \{\mathbf{0}^{d_2}\} \cup \{\mathbf{0}^{d_1}\} \cdot C_2$。不难证明，$S_\omega(\mathcal{A}) = L(\mathcal{A}_1) \cdot S_\omega(\mathcal{A}_2)$，这里我们仅给出证明概要。SPBA $\mathcal{A}$的构造方式是：它从 $\mathcal{A}_1$ 的一个副本开始，并且可以从 f 转移到 $\mathcal{A}_2$副本的初始状态。由于 $\mathcal{A}_1$ 是规范化的，它只有一个接受状态 f，且不存在指向 $\mathcal{A}_1$ 的出边。特别是，要接受一个无限词，自动机 $\mathcal{A}$ 必须从 $\mathcal{A}_1$ 的副本转移到 $\mathcal{A}_2$的副本。由于 f 是 $\mathcal{A}_1$ 唯一的接受状态，也是唯一能够转移到 $\mathcal{A}_2$副本中各状态的状态，因此根据 C的选择，到达 f 时，我们必定已经读入了一个来自 $L(\mathcal{A}_1)$ 的字。在 f中重置之后，只有当无限词剩余部分属于 $S_\omega(\mathcal{A}_2)$ 时，它才会接受（注意，前 d_1 个计数器在 f中被重置，而重置之后只使用最后的 d_2 个计数器）。

PPBA 的证明类似。我们从一个 PPBA $\mathcal{A}_2$ for L_2出发，而不是从 SPBA 出发，并构造一个 PPBA $\mathcal{A}$ for $L_1 L_2$。构造中唯一的区别是设置$C = C_1 \cdot C_2$，因为自动机访问 f时，计数器不会重置。□

6. PPBA 和 SPBA 的表达能力

本节将证明，可由 PPBA 识别的 ω-语言类严格包含于可由 SPBA 识别的 ω-语言类。考虑受 B¨uchi 定理启发的类 $\mathcal{L}^\omega_{\mathsf{PA}} = \{U_1 V_1^\omega \cup \cdots \cup U_n V_n^\omega \mid n \geq 1, U_i, V_i \in \mathcal{L}_{\mathsf{PA}}\}$ 将很有帮助。注意，这个类等价于 Fernau 和 Stiebe 提到的类 $\mathcal{K}_*$；他们证明了，可由 CM 识别的 ω-语言类严格包含于 $\mathcal{K}_*$ [5, 引理 3.2 和 3.3]。由推论 5.1 可得，可由 PPBA 识别的 ω-语言类严格包含于 $\mathcal{L}^\omega_{\mathsf{PA}}$。

我们证明，$\mathcal{L}^\omega_{\mathsf{PA}}$ 是 SPBA 可识别语言类的真子集。首先证明，Parikh 可识别语言对 ω 闭包后是 SPBA 可识别的。

> 引理 6.1。设 $L \subseteq \Sigma^*$ 是 Parikh 可识别的。那么 L^ω 是 SPBA 可识别的。

证明。设 $\mathcal{A} = (Q, \Sigma, q_0, \Delta, F, C)$ 是一个 PA，且 $L(\mathcal{A}) = L$。我们证明可以构造一个 SPBA $\mathcal{A}'$，使其无限次模拟 $\mathcal{A}$ 的接受运行，从而推出 $L_\omega(\mathcal{A}') = L^\omega$。由于根据定义，对所有语言 L，都有 $L^\omega = (L \setminus \{\varepsilon\})^\omega$，我们假设根据引理 5.3，$\mathcal{A}$ 已规范化。

我们选择 $\mathcal{A}' = (Q, \Sigma, q_0, \Delta', F, C)$，其中 $\Delta' = \Delta \cup \{(f, a, \mathbf{v}\ q) \mid (q_0, a, \mathbf{v}\ q) \in \Delta\}$。我们证明 $S_\omega(\mathcal{A}') = L(\mathcal{A})^\omega$。

$\Rightarrow$ 我们首先证明 $S_\omega(\mathcal{A}') \subseteq L(\mathcal{A})^\omega$。设 $\alpha \in S_\omega(\mathcal{A}')$ 的接受运行是$r = r_1 r_2 r_3 \ldots$，其中 $r_i = (p_{i-1}, \alpha_i, \mathbf{v}_i, p_i)$。令 $0 = k_0 < k_1 < k_2 < \ldots$ 表示 r中的重置位置。我们证明，对于所有 $j \geq 1$，都可以修改 $w_j = \alpha[k_{j-1} + 1, k_j]$ 上的部分运行 $r_{j-1,\ j} = r_{k_{j-1}+1} r_{k_{j-1}+2} \ldots r_{k_j}$，使其成为 w_j上的 $\mathcal{A}$的接受运行。注意，$r_{0,\ 1}$ 已经是 $\mathcal{A}$ 在 w_1上的接受运行。观察到

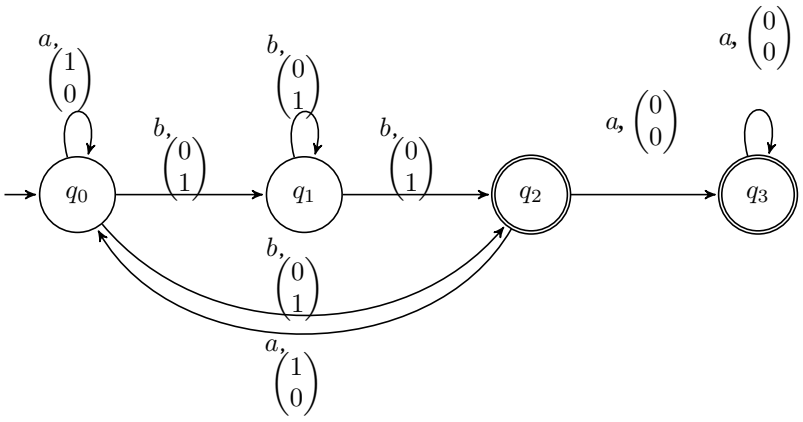

图 4：用于 $L = \{a^n b^n \mid n \geq 1\}^* \cdot \{a\}^\omega \cup \{a^n b^n \mid n \geq 1\}^\omega$ 且带有 $C = \{(z,\ z)$ 的 SPBA $\mid z \in \mathbb{N}\}$。

$j > 1$ 转移 $r_{k_{j-1}+1}$ 是一个形如 $(f,\ \alpha_{k_{j-1}+1},\ \mathbf{v}^{k_{j-1}+1},\ p_{k_{j-1}+1})$ 的新转移，因为 f 在 $\mathcal{A}$中没有出边。特别地，存在一个形如$\delta = (q_0,\ \alpha_{k_{j-1}+1},\ \mathbf{v}^{k_{j-1}+1},\ p_{k_{j-1}+1}) \in \Delta$的转移。因此，对于所有 $j > 1$，$\delta r_{k_{j-1}+2} \dots r_{k_j}$都是 w_j 上的 $\mathcal{A}$的接受运行。因此我们有 $\alpha \in L(\mathcal{A})^\omega$。」

$\Leftarrow$为了证明 $L(\mathcal{A})^\omega \subseteq S_\omega(\mathcal{A}')$，令 $w_1 w_2 w_3 \dots \in L(\mathcal{A})^\omega$，其中对所有 $j \geq 1$，$w_j \in L(\mathcal{A})$。令 $n_j = |w_1| + \dots + |w_j|$表示 $w_1 \dots w_j$的长度，并令$r_{j-1,j} = r_{n_{j-1}+1} r_{n_{j-1}+2} \dots r_{n_j}$，其中 $r_i = (p_{i-1}, \alpha_i, \mathbf{v}_i, p_i)$ 是 $\mathcal{A}$ 在 w_j上的一个接受运行。注意，p_{n_j} 是 $r_{j-1,j}$中唯一的接受状态。此外，存在一个转移$\delta_j = (p_{n_j}, \alpha_{n_j+1}, \mathbf{v}^{n_j+1}, p_{n_j+1}) \in \Delta'$。令对所有 $j \geq 2$，$r'_{j-1,j} = \delta_{j-1} r_{n_{j-1}+2} \dots r_{n_j}$。那么$r = r_{0,1} r'_{1,2} r'_{2,3} \dots$是 $\mathcal{A}'$ 在 $w_1 w_2 w_3 \dots$上的一个运行，并且我们在每个 r_{n_j}上都进行一次重置。由于对所有 $j \geq 1$，$\rho(w_j) \in C$，运行 r是接受运行，因此 $w_1 w_2 w_3 \dots \in S_\omega(\mathcal{A}')$。 □

备注 6.1。 由于 ω-语言 $\{a^n b^n \mid n \geq 1\}^\omega$ 不是可由 PPBA 识别的（这是推论 5.1 和 [5, 引理 3.3]的推论），因此该引理不适用于 PPBA 可识别的语言。

至此，我们已具备所有必要条件，可以证明类 $\mathcal{L}^\omega_{\mathsf{PA}}$ 是 SPBA 可识别的 ω-语言类的子集。我们将证明这种包含关系是严格的。

定理 6.1。 $\mathcal{L}^\omega_{\mathsf{PA}} \subsetneq \mathcal{L}_{\mathsf{Reset}}$。

证明。 该包含关系可直接由观察 5.1、引理 5.4 和引理 6.1 得出。

我们将证明这种包含关系是严格的。设 $L = \{a^n b^n \mid n \geq 1\}^* \cdot \{a\}^\omega \cup \{a^n b^n \mid n \geq 1\}^\omega$。我们证明 $L \in \mathcal{L}_{\mathsf{Reset}}$，但 $L \notin \mathcal{L}^\omega_{\mathsf{PA}}$。$L \in \mathcal{L}_{\mathsf{Reset}}$ 可由图 4 中满足 $C = \{(z,\ z) \mid z \in \mathbb{N}\}$ 的 SPBA 见证。

我们聚焦于 $L \notin \mathcal{L}^\omega_{\mathsf{PA}}$ 并用反证法论证。假设 $L \in \mathcal{L}^\omega_{\mathsf{PA}}$，即存在 Parikh 可识别语言 U_1, V_1, ..., U_n, V_n，使得 $L = U_1 V_1^\omega \cup \dots \cup U_n V_n^\omega$。那么存在某个 $i \leq n$，使得对于无穷多个 $j \geq 1$，无限词 $\alpha_j = aba^2b^2 \dots a^j b^j \cdot a^\omega \in U_i V_i^\omega$。那么 V_i必定包含一个形如 $v = a^k$、$k > 0$ 的字。此外，V_i 中不可能有带有中间子串 b 的字。为此，假设

反设存在一个字 $w \in V_i$，满足 $\ell = |w|_b > 0$。令 $\beta = (v^{\ell+1}w)^\omega$。注意到 β 有一个中间子串，其中包含至少 $\ell+1$ 个 a，后接至多 ℓ 个、但至少一个 b，因此，L中不存在任何形如 $u\beta$ 且满足 $u \in U_i$ 的字。这就产生了矛盾，因此 $V_i \subseteq \{a\}^+$。

由于 $U_i \in \mathcal{L}_{\mathsf{PA}}$，存在一个具有 $L(\mathcal{A}_i) = U_i$ 的概率自动机 $\mathcal{A}_i$。令 m 为 $\mathcal{A}_i$ 和 $w' = aba^2b^2 \dots a^{m^4+1}b^{m^4+1}$ 中的状态数。于是，w' 是被 $\mathcal{A}_i$ 接受的某个字的前缀。现在考虑因子 $a^\ell b^\ell$ 以及状态对 q_1，q_2，其中我们从 a^ℓ 开始读取，并在读取 a^ℓ 时结束；以及状态对 q_3，q_4，其中我们分别从读取 b^ℓ 开始，并在读取 b^ℓ 时结束。第一对有 m^2 种选择，第二对有 m^2 种选择，因此总共有 m^4 种可能。由于这样的因子多于 m^4 个，必定存在两个因子具有相同的关联状态 q_1，q_2，q_3，q_4。然后我们可以交换这两个因子，得到一个形如 $ab \dots a^r b^s \dots a^s b^r \dots a^{m^4+1}b^{m^4+1}$ 的字，它是 $L(\mathcal{A}_i) = U_i$ 中某个字的前缀。但 L 中没有任何字具有这样的前缀，这就产生了矛盾。因此，$U_1V_1^\omega \cup \dots \cup U_nV_n^\omega \neq L$。 □

7. 判定问题

在本节中，我们将说明，[9, 6] 中关于有限词上的 PA 的常见判定问题的（不可）判定性结果可以转移到 PBA。这些问题包括：

1. 空性。 $\mathcal{A}$ 是否识别空语言?
2. 普遍性。 $\mathcal{A}$ 是否接受每个字?
3. 等价性。 $\mathcal{A}_1$ 和 $\mathcal{A}_2$ 是否识别相同的语言?
4. 包含关系。 $\mathcal{A}_2$ 是否识别 $\mathcal{A}_1$ 识别的每个字?

下文假设，所有半线性集均以基向量 b_0 和周期向量 b_1、……、b_k 的列表集合形式给出（有限并中的每个线性集对应一个列表）。所有数字均以二进制编码。

> 定理 7.1。SPBA、WPBA 和 PPBA（以及与它们等价的模型）的空性问题是 coNP 完全的。SPBA、WPBA 和 PPBA（以及与它们等价的模型）的普遍性、包含关系和等价性问题均不可判定。

根据引理 3.4 和引理 3.5，每个 SPBA 都能高效地转换为与之等价的 WPBA，反之亦然，因此 SPBA 的可判定性结果同样适用于 WPBA。此外，我们将展示如何在多项式时间内把任意 PPBA 转换为与之等价的 WPBA（进而转换为 SPBA）（请注意，这并不能直接从我们的结果推出，因为将 PPBA 转换为与之等价的计数器机这一中间步骤会引入 ε-转移，而消除这些转移需要超多项式时间）。

因此，要证明空性的 coNP-完全性，只需证明 ε-MSPBA 的空性属于 coNP（从而证明我们所有模型的 coNP-归属），并证明 PPBA（甚至是一维 PPBA）的 coNP-困难性（从而证明所有模型都具有困难性）。同理，只需证明 PPBA 的不可判定性结果。

> 引理 7.1。设 $\mathcal{A} = (Q, \Sigma, q_0, \Delta, F, C)$ 是维数为 d 的 PPBA。那么存在一个等价的 WPBA，可在多项式时间内构造。

证明。思路如下。我们有 $\mathcal{A}$ 的一个副本，此外，对于半线性集 C 中的每个线性集 C_i，我们引入 $\mathcal{A}$ 的一个副本 $\mathcal{A}_i$。$\mathcal{A}$ 的副本中的所有状态都是非重置的，而 F 中的所有状态在 $\mathcal{A}_i$ 中都是重置的。运行仍处于 $\mathcal{A}$ 且将要进入状态 $f \in F$ 时，我们允许非确定性地转移到某个 $\mathcal{A}_i$ 并重置计数器。为此，我们先转移到 f 的一个特殊重置副本，然后再转移到 $\mathcal{A}_i$。我们使用附加计数器来标记当前所处的半线性集 $\mathcal{A}_i$，并修改该半线性集，以便在运行继续进入 $\mathcal{A}_i$ 时不再重复计算基向量。

形式化地，假设 $C = \bigcup_{i \le k} C_i$。令 $\mathcal{A}' = (Q', \Sigma, (q_0, \mathbf{0})\ \Delta', F', C')$ 为维数为 $d + k + 1$ 的，其中

$$
\begin{aligned}
Q' &= Q \times \{0, 1, \ldots, k\} \cup F \times \{1, \ldots, k\} \times \{R\},\\
F' &= F \times \{1, \ldots, k\} \times \{R\} \cup F \times \{1, \ldots, k\},\\
\Delta' &= \{((p,0), a, \mathbf{v} \cdot \mathbf{0}^{k+1}, (q,0)), ((p,i), a, \mathbf{v} \cdot \mathbf{e}_i^{k+1}, (q,i)) \mid (p,a,\mathbf{v},q) \in \Delta, 1 \le i \le k\}\\
&\quad \cup \{((q,0), a, \mathbf{v} \cdot (\mathbf{e}_i^{k+1} + \mathbf{e}_{k+1}^{k+1}), (f,i,R)\} \mid (q,a,\mathbf{v},f) \in \Delta, f \in F\}\\
&\quad \cup \{((f,i,R), a, \mathbf{v} \cdot \mathbf{e}_i^{k+1}, (q,i)\} \mid (f,a,\mathbf{v},q) \in \Delta, f \in F\}, \text{ and}\\
C' &= \bigcup_{i \le k} C_i \cdot \{(\mathbf{e}_i^{k+1} + \mathbf{e}_{k+1}^{k+1})\} \cup \bigcup_{i \le k} \vec{C}_i \cdot \{z \cdot \mathbf{e}_i^{k+1} \mid z \in \mathbb{N}\},
\end{aligned}
$$

其中，$\vec{C}_i$ 定义为 C_i，但不含任何基向量。我们证明 $P_\omega(\mathcal{A}) = W_\omega(\mathcal{A}')$.

$\Rightarrow$ 我们首先证明 $P_\omega(\mathcal{A}) \subseteq W_\omega(\mathcal{A}')$。令 $\alpha \in P_\omega(\mathcal{A})$ 具有接受运行 $r = r_1 r_2 r_3 \ldots$，其中 $r_i = (p_{i-1},\ \alpha_i,\ \mathbf{v}_i,\ p_i)$，并用 $f_1 < f_2 < \ldots$ 表示 r 的接受命中位置。根据无限鸽巢原理，存在某个 $\ell \le k$，使得这些接受命中中有无穷多个属于线性集 C_ℓ。用 $k_1 < k_2 < \ldots$ 表示 f_1，f_2，..... 中所有属于 C_ℓ 的接受命中所构成的子序列。

这意味着 $p_{k_j} \in F$ 和 $\rho(r_1 \cdots r_{k_j}) \in C_\ell$ 对所有 $j \ge 1$ 都成立。对于 $i < j$，令 $r_{i,j} = r_{k_i+1} \ldots r_{k_j}$。现在注意到，对所有 $\ell \ge 1$，$\rho(r_1 \ldots r_{k_\ell}) = \rho(r_1 \ldots r_{k_1}) + \rho(r_{1,2}) + \rho(r_{2,3}) + \cdots + \rho(r_{\ell-1,\ell})$。特别地，对所有 $i < \ell$，我们有 $\rho(r_1 \ldots r_{k_1}) \in C_\ell$ 和 $r_{i,i+1} \in \vec{C}_\ell$。

令 $\hat{}r = \hat{}r_1 \ldots \hat{}r_{k_1-1}((p_{k_1-1}, \mathbf{0})\ \alpha_{k_1}, \mathbf{v}^{k_1} \cdot (\mathbf{e}_\ell^{k+1} + \mathbf{e}_{k+1}^{k+1}), (f, \ell, R))$，其中对所有 $i \le k_1 - 1$，$\hat{}r_i = ((p_{i-1}, \mathbf{0}), \alpha_i, \mathbf{v}_i \cdot \mathbf{e}_0^{k+1}, (p_i, \mathbf{0}))$；并令 $r' = ((f, \ell, R)\ \alpha_{k_1+1}, \mathbf{v}^{k_1+1} \cdot \mathbf{e}_\ell^{k+1}, (p'_{k_1+1}, \ell))$ $r'_{k_i+2} r'_{k_i+3} \ldots$，其中对所有 $i \ge k_1 + 1$，$r'_i = ((p'_{i-1}, \ell)\ \alpha_i, \mathbf{v} \cdot \mathbf{e}_\ell^{k+1}, (p'_i, \ell))$。那么，$\hat{}r r'$ 是 $\mathcal{A}'$ 在 α 上的一次运行。此外，这次运行是接受的，因为我们可以将 r 的接受命中位置与 $\hat{}r r'$ 的重置位置一一对应；这由 Δ'、C' 的选取以及上述观察可知。因此 $\alpha \in W_\omega(\mathcal{A}')$。

$\Leftarrow$ 为了证明 $W_\omega(\mathcal{A}') \subseteq P_\omega(\mathcal{A})$，设 $\alpha \in W_\omega(\mathcal{A}')$ 有接受运行 $r' = r'_1 r'_2 r'_3 \ldots$，其中 $r'_i = (p'_{i-1}, \alpha_i, \mathbf{v}_i \cdot u, p'_i)$，且 $p'_i \in Q'$ 和 $u \in \{\mathbf{0}^{k+1}, \mathbf{e}_\ell^{k+1},\ (\mathbf{e}_\ell^{k+1} + \mathbf{e}_{k+1}^{k+1}) \mid 1 \le \ell \le k\}$，并用 $0 = k_0 < k_1 < k_2 \ldots$ 表示 r' 的重置位置。由于 $\mathcal{A}$ 的第一个副本中没有状态会重置，且根据 $\mathcal{A}'$ 的进一步构造，对于所有 $i < k_1$，$p'_{k_1} = (p_{k_1}, \ell, R)$，都有 $p'_i = (p_i, \mathbf{0})$；并且对于某个 $\ell \le k$，对所有 $i > k_1$，都有 $p'_i = (p_i, \ell)$。由于从 p'_{k_1-1} 到 p'_{k_1} 的转移是唯一一个将新计数器中的第一个计数器设为 1 的转移，且该转移在 r' 中只出现一次（无法从副本 $\mathcal{A}_\ell$ 返回），因此有 $\rho(r'_1 \ldots r'_{k_1}) \in C_\ell \cdot \{(\mathbf{e}_\ell^{k+1} + \mathbf{e}_{k+1}^{k+1})\}$，并且对所有 $i > 1$，都有 $\rho(r'_{k_{i-1}+1} \ldots r'_{k_i}) \in \vec{C}_\ell \cdot \{z\mathbf{e}_\ell^{k+1} \mid z \in \mathbb{N}\}$。对于每个 $i \ge 1$，令 $r_i = (p_{i-1}, \alpha_i, \mathbf{v}_i, p_i)$。那么 $r = r_1 r_2 r_3 \ldots$ 在 α 上是 $\mathcal{A}$ 的。此外，r 是接受的，因为 r' 的重置位置

在 r 中一一对应地映射到接受命中。为此，注意对于所有向量 $\mathbf{u} \in C_\ell$ 和 $\mathbf{v} \in \vec{C}_\ell$，都有 $\mathbf{u} + \mathbf{v} \in C_\ell$。由于 $\rho(r_1 \dots r_{k_1}) = \mathbf{v}_1 + \dots + \mathbf{v}_{k_1} \in C$ 且 $\rho(r_{k_{i-1}+1} \dots r_{k_i}) = \mathbf{v}_{k_{i-1}+1} + \dots + \mathbf{v}_{k_i} \in \vec{C}$ 对所有 $i > 1$ 都成立，因此运行 r 确实是接受运行，从而 $\alpha \in P_\omega(\mathcal{A})$。

自动机 $\mathcal{A}'$ 至多有 $(k+2)|Q|$ 个状态，并且显然可以在多项式时间内构造。

□

现在我们证明，PPBA 的非空性是 NP-hard 的。类似的证明曾在 [6, 命题 III.2]中略作说明。

引理 7.2。 PPBA 的非空性是 NP-hard 的。

证明。 我们将从子集和问题进行（多项式时间多对一）归约，该问题已知是 NP完全的 [8]。该问题的一个实例由一个有限集 $M \subseteq \mathbb{N}$ 和一个阈值 ℓ构成，问题是判断是否存在一个子集 $S \subseteq M$，使得 $\sum_{s \in S} s = \ell$。设 $\mathcal{I}$ 是子集和问题的一个实例，其中 $M = \{m_1, \dots, m_k\}$，阈值为 ℓ。直观地说，我们将 $\mathcal{I}$的一个解编码为一个由 $\Sigma = \{x_1, \dots, x_k\}$上的有限词构成的字，其长度为 k，并使用由虚符号 x_D组成的无限序列对其进行填充。之后，我们构造一个维数为 1 的 PPBA $\mathcal{A}_\mathcal{I}$，它仅接受 $\mathcal{I}$ 的有效解。

我们选择 $\mathcal{A}_\mathcal{I} = (\{q_i \mid 1 \le i \le m\} \cup \{q_0, q_f\},\ \Sigma \cup \{x_D\},\ q_0,\ \Delta_\mathcal{I},\ \{q_f\},\ \{\ell\})$，其中

$$\begin{aligned}\Delta_\mathcal{I} = &\{(q_{i-1}, x_i, m_i, q_i), (q_{i-1}, x_i, 0, q_i) \mid 1 \le i \le m\} \\ &\cup \{(q_m, x_D, 0, q_f), (q_f, x_D, 0, q_f)\}.\end{aligned}$$

我们声称，$\mathcal{I}$ 有解，当且仅当 $P_\omega(\mathcal{A}_\mathcal{I}) \neq \varnothing$。

$\Rightarrow$ 设 $S \subseteq M$ 是 $\mathcal{I}$的一个解。容易看出，$\mathcal{A}_\mathcal{I}$ 接受无限词 $x_1 \dots x_m x_D^\omega$，因为运行 $r = r_1 \dots r_m (q_m, x_D, 0, q_f)(q_f, x_D, 0, q_f)^\omega$是接受运行，其中当 $m_i \in S$ 时为$r_i = (q_{i-1}, x_i, m_i, q_i)$，否则为 $r_i = (q_{i-1}, x_i, 0, q_i)$。因此，$P_\omega(\mathcal{A}_\mathcal{I}) \neq \varnothing$。

$\Leftarrow$ 设 α 是由 $\mathcal{A}_\mathcal{I}$接受的一个无限词。根据 $\mathcal{A}_\mathcal{I}$的选择，我们有 $\alpha = x_1 \dots x_m x_D^\omega$，其接受运行为 $r = r_1 \dots r_m (q_m, x_D, 0, q_f)(q_f, x_D, 0, q_f)^\omega$。我们根据 r构造一个解 S，如下所示。对于每个 $1 \le i \le m$，当且仅当 $r_i = (q_{i-1}, x_i, m_i, q_i)$时，我们将 m_i 添加到 S 中。与上文类似，集合 S 是 $\mathcal{I}$的一个解。

$\mathcal{A}_\mathcal{I}$ 的大小与 $\mathcal{I}$的大小呈线性关系，因此，这一变换是一个多项式时间归约。

□

接下来，我们证明 ε-MSPBA 的非空性属于 NP。正如 Fernau 和 Stiebe 所证明的，所有至少接受一个无限词的计数器机，也至少接受一个最终周期的无限词，也就是说，接受一个形如 uv^ω 的字，其中 $u \in \Sigma^*$ 且 $v \in \Sigma^+$。我们证明这一结论对 ε-MSPBA 也成立（因此对我们的所有模型都成立）。

引理 7.3. 设 $\mathcal{A}$ 是一个字母表为 Σ 的 ε-MSPBA。若 $S_\omega(\mathcal{A}) \neq \varnothing$，则存在一个形如 $uv^\omega \in S_\omega(\mathcal{A})$ 的字，其中 $u \in \Sigma^*$ 且 $v \in \Sigma^+$。

证明。 假设 $S_\omega(\mathcal{A}) \neq \varnothing$。那么存在一个无限词 $\alpha \in S_\omega(\mathcal{A})$，其接受运行为 $r = r_1 r_2 r_3 \ldots$，其中 $r_i = (p_{i-1}, \gamma_i, \mathbf{v}_i, p_i)$。设 $k_1 < k_2 < \ldots$ 为 r 中所有重置状态所在的位置。设 $k_i < k_j$ 是其中两个位置，满足 $p_{k_i} = p_{k_j}$，且 $\mathcal{A}$ 在离开 p_{k_i} 并进入 p_{k_j} 后，至少读入了一个来自 α 的符号。设 $u = \gamma_1 \ldots \gamma_{k_i}$ 是访问 p_{k_i} 时读入的 α 的前缀，$v = \gamma_{k_i+1} \cdots \gamma_{k_j}$ 是在 p_{k_i} 和 p_{k_j} 之间读入的中间子串。注意，根据 k_j 的选择，$v \neq \varepsilon$。于是 $\mathcal{A}$ 也接受 uv^ω，因为根据定义，$r_1 \ldots r_{k_i} (r_{k_i+1} \cdots r_{k_j})^\omega$ 是 $\mathcal{A}$ 在 uv^ω 上的接受运行。 □

这有助于给出一个 NP 算法，用于解决 ε-MSPBA 的非空性问题

引理 7.4. ε-MSPBA 的非空性问题属于 NP。

证明。根据引理 3.3，我们可以将任意 ε-MSPBA 转换为等价的 ε-SPBA（容易看出，该引理中的构造可在多项式时间内完成）。因此，我们说明如何解决 ε-SBPA 的非空性问题。设 $\mathcal{A} = (Q, \Sigma, q_0, \Delta, \mathcal{E}, F, C)$是一个 ε-SPBA。根据引理 7.3，只需检查 $\mathcal{A}$ 是否接受一个无限词uv^ω，使得 $u \in \Sigma^*$ 且 $v \in \Sigma^+$。若存在这样的字，我们可以假设存在 $\mathcal{A}$ 在 uv 上的一个接受运行 $r_u r_v^\omega$，其中 r_u 和 r_v 都不会两次访问同一个接受状态。对于任意 p、$q \in Q$，我们定义 $\mathcal{A}_{p \Rightarrow q} = (Q \cup \{q_0'\}, \Sigma, q_0', \Delta', \mathcal{E}', \{q\}, C)$，其中$\Delta' = \{(q_1, a, \mathbf{v}\ q_2) \mid (q_1, a, \mathbf{v}\ q_2) \in \Delta, q_1 \notin F\} \cup \{(q_0', a, \mathbf{v}\ q_2) \mid (p, a, \mathbf{v}\ q_2) \in \Delta\}$，类似地，$\mathcal{E}' = \{(q_1, \varepsilon, \mathbf{v}\ q_2) \mid (q_1, \varepsilon, \mathbf{v}\ q_2) \in \mathcal{E}, q_1 \notin F\} \cup \{(q_0', \varepsilon, \mathbf{v}\ q_2) \mid (p, \varepsilon, \mathbf{v}\ q_2) \in \Delta\}$。

以下 NP 算法可解决非空性问题:

1. 猜测一个接受状态序列 f_1、……、f_k，满足 $k \leq 2|F|$，其中 $f_i = f_k$ 对某个 $i \leq k$ 成立。 2. 验证对于所有 $1 \leq j < k$（将其视为有限词上的概率自动机），$L(\mathcal{A}_{q_0 \Rightarrow f_1}) \neq \varnothing$ 且 $L(\mathcal{A}_{f_j \Rightarrow f_{j+1}}) \neq \varnothing$。 3. 验证 $L(\mathcal{A}_{f_i \Rightarrow f_{i+1}}) \cdot \ldots \cdot L(\mathcal{A}_{f_{k-1} \Rightarrow f_k}) \not\subseteq \{\varepsilon\}$。

第二步可以通过向自动机中添加一个全新的符号（比如 e），并将每个 ε-转移替换为一个 e-转移来完成（注意，这种构造不会改变空性行为；与引理 4.4 的构造不同，它可以在多项式时间内计算）。之后，我们使用用于判定概率自动机 [6] 非空性的 NP 算法。

第三步的核心是指出，并非所有 $L(\mathcal{A}_{f_j \Rightarrow f_{j+1}})$ 在 $j \geq i$ 下都只能接受空词，因为我们要求 $v \neq \varepsilon$。为检验这一性质，我们可以构造一个 PA[1]，识别 $L(\mathcal{A}_{f_i \Rightarrow f_{i+1}}) \cdot \ldots \cdot L(\mathcal{A}_{f_{k-1} \Rightarrow f_k})$，并再次将每个 ε-转移替换为一个 e-转移。最后，我们构造该 PA 与识别语言 $\{w \in (\Sigma \cup \{e\})^* \mid w$ 中至少包含 1 个来自 $\Sigma\}$ 的符号的 PA（非确定有限自动机）的积自动机，并检验所得 PA 的非空性。 □

现在我们证明，PPBA 的普遍性（因此也包括我们所有模型的普遍性）是不可判定的。

引理 7.5。 PPBA 的普遍性是不可判定的。

[1]通过一种与引理 5.4 中的方法非常相似的构造，可以在多项式时间内完成。不过，据我们所知，文献中尚无针对有限词上的 PA 的拼接的显式构造。

证明。 我们从概率自动机的普遍性问题归约而来，该问题已知是不可判定的 [9]。设 $\mathcal{A} = (Q, \Sigma, q_0, \Delta, F, C)$ 是任意一个概率自动机。我们构造一个 PPBA $\mathcal{A}'$，使其是全称的，当且仅当 $\mathcal{A}$ 是全称的。思路如下。PPBA $\mathcal{A}'$ 接受所有由 $\mathcal{A}$ 接受、后接一个由虚符号组成的无限序列的字 x_D。此外， $\mathcal{A}'$ 接受所有不属于 $\Sigma^* \cdot \{x_D\}^\omega$ 的字，也就是说，所有“格式错误”的字。第一步，我们构造一个 PPBA $\mathcal{A}_1$，其中

$$\mathcal{A}_1 = (Q \cup \{q_f\}, \Sigma \cup \{x_D\}, q_0, \Delta_1, \{q_f\}, C)$$

with

$$\Delta_1 = \Delta \cup \{(q, x_D, \mathbf{0}, q_f) \mid q \in F\} \cup \{(q_f, x_D, \mathbf{0}, q_f)\}.$$

注意， $P_\omega(\mathcal{A}_1) = L(\mathcal{A}) \cdot \{x_D\}^\omega$。令 $\mathcal{A}_2$ with $P_\omega(\mathcal{A}_2) = \Sigma^\omega$，并令 $\mathcal{A}_3$ 为一个 PPBA，它接受所有包含一个虚设符号后接一个非虚设符号的无限词，即 $P_\omega(\mathcal{A}_3) = (\Sigma \cup \{x_D\})^* \cdot \{x_D\} \cdot \Sigma \cdot \{\Sigma \cup \{x_D\}\}^\omega$ （注意，这些语言甚至是 ω-正则的）。令 $L = L_\omega(\mathcal{A}_1) \cup L_\omega(\mathcal{A}_2) \cup L_\omega(\mathcal{A}_3)$。根据观察 5.1，我们可以构造一个识别 L 的 PPBA。令 $\mathcal{A}'$ 为这个 PPBA。那么， $L(\mathcal{A}) = \Sigma^*$ 当且仅当 $P_\omega(\mathcal{A}') = (\Sigma \cup \{x_D\})^\omega$。

$\Rightarrow$ 令 $\mathcal{A}$ 为全称的，即 $\mathcal{A}$ 接受所有字 $w \in \Sigma^*$；也就是说，根据 $\mathcal{A}_1$ 的构造，所有形如 $w \cdot x_D^\omega$ 的无限词都将被 $\mathcal{A}'$ 接受。由于 $\mathcal{A}_2$和 $\mathcal{A}_3$ 接受 $(\Sigma \cup \{x_D\})^\omega \setminus (\Sigma^* \cdot \{x_D\}^\omega)$ 中的所有无限词，因此 PPBA $\mathcal{A}'$ 接受 $\Sigma \cup \{x_D\}$ 上的所有无限词，即 $\mathcal{A}'$ 是全称的。

$\Leftarrow$ 设 $\mathcal{A}'$ 是全称的。我们用反证法进行论证。假设 $\mathcal{A}$ 不是全称的，也就是说，存在一个字 $w \in \Sigma^*$ 被 $\mathcal{A}$拒绝。那么 $\alpha = w \cdot x_D^\omega$ 也会被 $\mathcal{A}_1$拒绝。此外， $\mathcal{A}_2$ 和 $\mathcal{A}_3$ 也不接受 α。因此， $\alpha \notin P_\omega(\mathcal{A}')$，矛盾。 □

注7.1。如果将 $\mathcal{A}'$ 解释为 SPBA 或 WPBA，这个证明也同样成立。唯一的区别在于，我们考虑 $\mathcal{A}$ 的规范化自动机 $\mathcal{A}_N$，并单独检查 ε 的成员资格；由于 PA 的成员资格问题是可判定的（属于 NP） [9, 6]，这一检查仍然可以计算。

既然 PPBA 的普遍性是不可判定的，我们可以轻易地如下推出等价性和包含关系的不可判定性：设 Σ 是任意字母表， $\mathcal{A}_\Sigma$ 是一个满足 $P_\omega(\mathcal{A}_\Sigma) = \Sigma^\omega$ 的 PPBA。设 $\mathcal{A}$ 是任意 PPBA。我们注意到， $\mathcal{A}$ 是全称的，当且仅当 $P_\omega(\mathcal{A}) = P_\omega(\mathcal{A}_\Sigma)$；当且仅当 $P_\omega(\mathcal{A}_\Sigma) \subseteq P_\omega(\mathcal{A}, C)$。同样，这些结论也适用于 SPBA、WPBA 以及所有等价模型。

8. 结论与展望

我们引入了具有不同接受条件的 Parikh-B¨uchi 自动机。我们证明，具有强重置接受条件和弱重置接受条件的 PBA 描述了相同的一类 ω-语言，而这一类严格包含可由 PPBA 识别的ω-语言。后一类等价于 Fernau 和 Stiebe 引入的盲 k-计数器机器类。作为一个副产品，我们证明了所有模型都允许ε-消除，并且等价于其多计数器对应形式。最后，我们研究了若干常见的判定问题。

一个有趣的开放问题是，寻找一种能够恰好刻画 $\mathcal{L}_{\mathsf{PA}}^{\omega}$ 的（直观）模型。反之，关于 $\mathcal{L}_{\mathsf{Prefix}}$ 和 $\mathcal{L}_{\mathsf{Reset}}$ 、遵循 B¨uchi 定理思路的刻画仍有待发现。此外，研究确定性 PBA 也很有意思，因为确定性 B¨uchi 自动机已经弱于（非确定性）B¨uchi 自动机。最后，Klaedtke 和 Ruess 提议研究一种 PBA 变体：在半线性集中加入一个表示无穷大的符号 ∞ ，并通过超限归纳相应地计算无限词的扩展 Parikh 像。

参考文献

[1] J. Richard B¨uchi. 弱二阶算术与有限自动机。 *Mathematical Logic Quarterly*, 6 1 6 66(-): –92, 1960. [2] Micha ¨ *RAIRO Theor.* el Cadilhac、Alain *Informatics Appl.* 46 4 511 545, 2012 Finkel 和 Pierre McKenzie。仿射 Parikh 自动机。, ( ): –。[3] Micha ¨ *Int. J. Found.* el Cadilhac、Alain *Comput. Sci.* 23 8 1691 1710, 2012 Finkel 和 Pierre McKenzie。有界 Parikh 自动机。, ( ): –。[4] Luc Dartois、Emmanuel Filiot 和 Jean-Marc Talbot。带有可见下推栈的双向 Parikh 自动机。载于 Mikolaj Bojanczyk 和 Alex Simpson 编， *Foundations of Software Science and Computation Structures - 22nd International Conference, FOSSACS 2019, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2019, Prague, CzechRepublic, April 6-11,* 2019, *Proceedings*, 第11425卷，收录于 *Lecture Notes in Computer Science*，第189–206页。施普林格，2019。[5] Henning Fernau 和 Ralf Stiebe。ω-字上的盲计数器自动机。 *Fundam. Inform.*, 83 51: –64, 2008。[6] Diego Figueira 和 Leonid Libkin。用于查询图的路径逻辑：兼顾表达能力与效率。载于 *Proceedings of the 2015 30th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)* 15, 329 340, 2015 , LICS ' pages – 京都, 日本，。IEEE。[7] Emmanuel Filiot、Shibashis Guha 和 Nicolas Mazzocchi。双向 Parikh 自动机。载于 Arkadev Chattopadhyay 和 Paul Gastin 编， *39th IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS 2019, December 11-13,* 2019, *Bombay India* 150 *LIPIcs* 40 1 40 14 ¨ , , 第 卷，收录于 ，第: – :。Schloss Dagstuhl - Leibniz-Zentrum fur Informatik，2019。[8] Michael R. Garey 和 David S. Johnsons。 *Computers and Intractability: A Guide to the Theory of NP-Completeness* 1 1979 。W. H. Freeman，美国纽约，第 st 版，。[9] Felix Klaedtke 和 Harald Rueß。带基数的单子二阶逻辑。载于 Jos C. M. Baeten、Jan Karel Lenstra、Joachim Parrow 和 Gerhard J. Woeginger 编， *Automata Languages and* ，第 – 页，柏林，海德堡，。施普林格。[10] Rohit J Parikh。论上下文无关语言。 *Journal of the ACM (JACM)*, 13(4):570–581, 1966。[11] Wolfgang Thomas 等。 *Automata, logics, and infinite games: a guide to current research*，第2500卷。施普林格科学与商业媒体，2002。