arXiv:2304.01278v1 [cs.FL] 3 Apr 2023

无限词上的跳跃自动机*

Shaull Almagor[1][0000−0001−9021−1175] 和 Omer Yizhaq[2]

[1] 以色列理工学院，以色列 shaull@technion.ac.il[2] 以色列理工学院，以色列 omeryi@campus.technion.ac.il

摘要。 跳跃自动机是有限自动机，它们以非连续的方式读取输入，不考虑词中字母的顺序。我们引入并研究无限词上的跳跃自动机。与已有大量研究的有限词情形不同，对于无限词而言，如何对词重新排序并不明确。为此，我们考虑三种语义：以某种顺序读取无限词且不遗漏任何字母的自动机；能够在给定大小 k 的窗口内重排词的自动机；以及能够在某个存在量化界限的窗口内重排词的自动机。我们研究这些模型的表达能力、闭包性质和算法性质。

关键词： 跳跃自动机 · Parikh 像 · 无限词

1 引言

传统自动机按顺序读取输入。事实上，大多数基于状态的计算模型都是如此。然而，在某些情况下，输入的顺序并不重要。例如，当输入表示可用资源，而我们只想推理其数量时。从语言理论的角度来看，这相当于考察语言的交换闭包，也就是它们的*Parikh* 像。为了在计算模型中刻画这一概念， [20] 引入了跳跃自动机。跳跃自动机可以非顺序地读取输入，在字母之间跳跃，只要每个字母都恰好读取一次即可。已有多项研究探讨了这类自动机的算法性质和表达能力 [10,11,24,9,18]。

自动机与语言理论中最令人振奋的发展之一，是将研究拓展到无限词的情形 [4,19,16], ，这为形式化方法带来了强大的工具。与有限词的情形相比，无限词的情形复杂得多，涉及多种接受条件以及错综复杂的表达能力关系。或许最值得注意的是，非确定性 Büchi 自动机无法确定化为确定性 Büchi 自动机，但可以确定化为 Rabin 自动机 [17,19,22]。

本文介绍了无限词上的跳跃自动机。首要挑战是为这一模型找到有意义的定义。事实上，这种直觉

* 本研究得到以色列科学基金会的资助（资助编号：989/22）

认为读头会在词中“跳跃”到不同位置，但这种理解并不适用于无限词，因为即使没有读取每个字母（甚至可能跳过无限多个字母），也可以构造出无限运行（见备注 2）。

为此，我们为自动机 $\mathcal{A}$ 在无限词上进行跳跃的方式提出了三种语义。

– 在跳跃语义下，若 $\mathcal{A}$ 接受 w 的某个排列，即接受一个词 w'，使得对于有限次出现的字母，w' 中每个字母的出现次数与 w 相同，并且无限次出现的字母集合也与之相同，则称词 w 被接受。– 在 k-窗口跳跃语义下，若能通过在某个固定大小的连续窗口内排列 w，使 $\mathcal{A}$ 接受 w，则称 w 被接受。– 在 $\exists$-窗口跳跃语义下，若存在某个 k，使得能通过在大小为 k 的连续窗口内排列 w，使 $\mathcal{A}$ 接受 w，则称 w 被接受。

例 *1.* 考虑一个识别语言 $\{(ab)^\omega\}$ 的 Büchi 自动机（见图3）。它的跳跃语言为 $\{w:\ w$ 中有无限多个 a 和 $b\ \}$。例如，它的 3 窗口跳跃语言包含这样的词：其中的 a 和 b 可以重新排列，从而在大小为 3 的窗口内构造出 $(ab)^\omega$，例如 $(aab\cdot bba)^\omega$。至于它的 $\exists$-窗口跳跃语言，其中包含词 $(aaaabbbb)^\omega$，该词不属于 3 窗口语言；但它不包含 $aba^2b^2a^3b^3\cdots$，而后者属于跳跃语言。

上述定义捕捉了跳跃的不同直观含义：第一个定义只考察词的 Parikh 像，对应于这样一种情形：词描述的是资源，其中一些资源是无界的。第二个（限制性更强的）定义描述了这样一种情形：词对应于需要处理的任务序列，而我们允许以有界自由度对这些任务进行重新排序；第三个定义则对应于这样一种情形：我们可以为某项任务引入任意有限延迟，但引入的总体延迟必须是有界的。

我们研究了这些语义的表达能力，以及闭包性质和判定问题。具体而言，我们证明了跳跃语义中的语言对于并集、交集和补运算都是闭合的。出人意料的是，我们还证明了在这种语义下，自动机可以进行确定化，这与标准 Büchi 自动机形成对比。我们进一步证明，这些自动机上判定问题的复杂度与其有限词跳跃对应物的复杂度相同。从技术上讲，这些结果是通过扩展半线性集以容纳 ∞，并证明跳跃语言可以借助这类集合以规范形式描述而得到的。

在 k-窗口语义下，我们证明它与 ω-正则语言之间存在对应关系，并由此推导出闭包性质以及判定问题的解。最后，我们证明 $\exists$-窗口语义的表达能力严格强于跳跃语义。

论文结构 第 2 节回顾一些基本概念，并定义我们的跳跃语义。第 3 节是本文的主体，我们在其中研究跳跃语义。在第 3.1 节中，我们介绍半线性集的一种变体，并证明它具有规范形式，且能够刻画跳跃语言。接着，在第 3.2 节和第 3.3 节中，我们研究跳跃语言的闭包性质和判定问题。在第 4 节和第 5 节中，我们分别研究 k-窗口语义和 $\exists$-窗口语义。最后，我们在第 6 节总结并展望未来研究。全文重点关注简洁的构造和可判定性，因此我们将复杂度分析放在各项结论之后的注释中。

相关工作 跳跃自动机最早在 [20] 中提出。我们指出，[20] 中包含一些错误的证明（例如关于交集和补的闭包性质，这一点也在 [11] 中有所指出）。[10,11] 中的研究确立了跳跃自动机在表达能力方面的若干结果，以及一些复杂度结果。[24] 中确立了许多其他闭包性质。[9] 研究了带有双向磁带的跳跃自动机扩展。

从技术角度来说，由于跳跃自动机对应于半线性 Parikh 像／交换闭包，[21], 关于半线性集的工具与之密切相关。[3,7,23] 中的研究给出了关于半线性集的算法结果，我们在本文中大量使用了这些结果；而在 [18] 中，半线性集的性质与自动机的状态复杂度相关。

更广泛地说，自动机与交换闭包也在 Parikh 自动机中交汇：这类自动机读取输入，并在与运行相关的某个 Parikh 像属于给定半线性集时接受。具体而言，[12] 研究了这类自动机在无限词上的一种扩展。另一种相关模型是对称变换器，即配备输出的自动机，其中输入中的排列对应于输出中的排列。[2] 以跳跃风格研究了这类模型，而 [1] 则以 k-窗口风格对其进行了研究。

2 预备知识与定义

自动机 一个非确定性自动机 是一个五元组 $\mathcal{A} = \langle \Sigma, Q, \delta, Q_0, \alpha \rangle$，其中 Σ 是有限字母表，Q 是有限状态集，$\delta\colon Q \times \Sigma \to 2^Q$ 是非确定性转移函数，$Q_0 \subseteq Q$ 是初始状态集，$\alpha \subseteq Q$是接受状态集。当对于每个 $q \in Q$ 和$\sigma \in \Sigma$，$|Q_0| = 1$ 且 $|\delta(q,\ \sigma)| = 1$ 时，我们称 $\mathcal{A}$ 是确定性的。

我们考虑在有限词和无限词上的自动机。我们用Σ^* 表示 Σ上的有限词集合，用 Σ^ω 表示无限词集合。对于一个词$w = \sigma_1,\ \sigma_2,\ \ldots$（有限或无限），令 $|w| \in \mathbb{N} \cup \{\infty\}$表示它的长度。$\mathcal{A}$在 w上的一个运行 是一个序列 $\rho = q_0,\ q_1,\ \ldots,$ 满足 $q_0 \in Q_0$，且对每个 $0 \le i < |w|$都有 $q_{i+1} \in \delta(q_i,\ w_{i+1})$（显然，若 w 有限，则运行是有限的；若w 无限，则运行是无限的）。对于有限词，运行 ρ 是接受的，当且仅当 $q_{|w|} \in \alpha$。对于无限词，我们采用 Büchi 接受条件，即运行 ρ 若无限次访问α，则称其为接受运行。形式化地，定义 $\mathrm{Inf}(\rho) = \{q \in Q \mid \forall i \in \mathbb{N}\ \exists j > i,\ \rho_j = q\}$,

那么，当 $\mathrm{Inf}(\rho) \cap \alpha \neq \emptyset$时， ρ 是接受的。如果存在 $\mathcal{A}$ 在 w上的一个接受运行，则称词 w 被 $\mathcal{A}$ 接受。

自动机 $\mathcal{A}$ 的语言 记作 $\mathfrak{L}(\mathcal{A})$，是它所接受的词集。我们强调，自动机作用于有限词时，记作 $\mathfrak{L}_{\mathrm{fin}}(\mathcal{A})$。

注 *1.* 自动机在无限词上还有其他接受条件，例如奇偶、Rabin、Streett、co-Büchi 和 Muller。正如命题 3 所示，就我们的目的而言，考虑 Büchi 条件已足够。

Parikh 像与排列 固定一个字母表 Σ。我们先为有限词定义 Parikh 像和排列。考虑一个有限词 $x \in \Sigma^*$。对于每个字母 $\sigma \in \Sigma$，我们用 $\#_\sigma[x]$ 表示 σ 在 x 中的出现次数。于是， x 的*Parikh* 像 是向量 $\Psi(x) = (\#_\sigma[x])_{\sigma\in\Sigma} \in \mathbb{N}^\Sigma$。如果 y 满足 $\Psi(y) = \Psi(x)$，我们就称词 y 是 x 的排列，此时记作 $x \sim y$（显然， $\sim$ 是一个等价关系）。我们将 Parikh 像扩展到词集，定义为 $\Psi(L) = \{\Psi(w) \mid w \in L\}$。

现在我们将定义扩展到无限词。令 $\mathbb{N} = \{0, 1, \ldots\}$ 为非负整数，并记作 $\mathbb{N}_\infty = \mathbb{N} \cup \{\infty\}$。考虑一个无限词$w \in \Sigma^\omega$。我们以自然的方式将 Parikh 像的定义扩展到无限词： $\Psi(w) = (\#_\sigma[w])_{\sigma\in\Sigma}$ 其中，若 $\sigma[w]$ 在 w 中的出现次数有限，则 $\#_\sigma[w]$ 是σ 在 w 中的出现次数；否则，$\#_\sigma[w]$ 为 ∞。因此， $\Psi(w) \in \mathbb{N}_\infty^\Sigma$。此外，由于 w是无限的， $\Psi(w)$ 至少有一个坐标为 ∞。由于我们通常只考虑无限词的 Parikh 像，因此用 $\mathbb{M}^\Sigma = \mathbb{N}_\infty^\Sigma \setminus \mathbb{N}^\Sigma$ 表示至少有一个 ∞ 坐标的向量集合。注意，当且仅当 $\boldsymbol{v}$ 是 Σ 上某个无限词的 Parikh 像时， $\boldsymbol{v} \in \mathbb{M}^\Sigma$。

对于词 $w, w' \in \Sigma^\omega$，我们稍微滥用一下记号，写作 $w \sim w'$ if $\Psi(w) = \Psi(w')$，并将 w' 称为 w的一个排列。现在，我们通过将排列限制在有限的“窗口”内来细化排列的概念：令 $k \in \mathbb{N}$，若 w 是 w'的一个 k-窗口排列，则记作 $w \sim_{k\boxplus} w'$（符号 $\boxplus$表示“窗口”），当且仅当 $w = x_1 \cdot x_2 \cdots$且 $w' = y_1 \cdot y_2 \cdots$，其中对所有 $i \geq 1$，都有 $|x_i| = |y_i| = k$ 和 $x_i \sim y_i$。也就是说， w' 可以通过对 w中的连续且互不相交的、大小为 k 的窗口进行排列，从 w 得到。注意，若 $w \sim_{k\boxplus} w'$，则特别有 $w \sim w'$，但前者的限制要严格得多。

半线性集 令 $\mathbb{N} = \{0, 1, \ldots\}$ 为自然数。对于维度$d \geq 1$，我们用粗体表示 $\mathbb{N}^d$ 中的向量（例如， $\boldsymbol{p} \in \mathbb{N}^d$）。给定向量 $\boldsymbol{b} \in \mathbb{N}^d$和 $P \subseteq \mathbb{N}^d$，由基 $\boldsymbol{b}$ 和周期 P 生成的线性集 为

$$\mathrm{Lin}(\boldsymbol{b}, P) = \{\boldsymbol{b} + \sum_{\boldsymbol{p}\in P} \lambda_p \boldsymbol{p} \mid \boldsymbol{b} \in \mathbb{N}^d \text{ and } \lambda_p \in \mathbb{N} \text{ for all } \boldsymbol{p} \in P\}$$

半线性集 即为 $\bigcup_{i\in I} \mathrm{Lin}(\boldsymbol{b_i}, P_i)$，其中 I 为有限集，且包含若干对 $(\boldsymbol{b_i}, P_i)$。

半线性集与正则语言的 Parikh 像密切相关（正则语言的 Parikh 像是半线性集） [21]。我们将在引用具体结果时，说明这些结果与这一联系之间的关系。

跳跃自动机 考虑一个自动机 $\mathcal{A} = \langle \Sigma, Q, \delta, Q_0, \alpha \rangle$。对于有限词，我们将 $\mathcal{A}$ 视为一种跳跃自动机，令其以非顺序方式读取输入，即在各字母之间“跳跃”，同时确保每个字母恰好读取一次。形式上，如果 $\mathcal{A}$ 接受 w的某种排列，那么作为跳跃自动机，它就接受词 w 。因此，我们将 $\mathcal{A}$ 的跳跃语言 定义为

$$\mathfrak{J}_{\mathrm{fin}}(\mathcal{A}) = \{w \in \Sigma^* \mid \exists w' \sim w \text{ such that } w' \in \mathfrak{L}_{\mathrm{fin}}(\mathcal{A})\}$$

现在我们来定义无限词上的跳跃自动机。如第 1 节所述，我们考虑 $\mathcal{A}$ 的三种跳跃语义，如下：

定义 1。 考虑一个自动机 $\mathcal{A} = \langle \Sigma, Q, \delta, Q_0, \alpha \rangle$。

– 其跳跃语言为：$\mathfrak{J}(\mathcal{A}) = \{w \in \Sigma^\omega \mid \exists w' \sim w$ 使得 $w' \in \mathfrak{L}(\mathcal{A})\}$。– 对于 $k \in \mathbb{N}$，其 k-窗口跳跃语言为：$\mathfrak{J}_{k\boxplus}(\mathcal{A}) = \{w \in \Sigma^\omega \mid \exists w' \sim_{k\boxplus} w$ 使得 $w' \in \mathfrak{L}(\mathcal{A})\}$。– 其 $\exists$-窗口跳跃语言为：$\mathfrak{J}_{\exists\boxplus}(\mathcal{A}) = \{w \in \Sigma^\omega \mid \exists k \in \mathbb{N} \wedge \exists w' \sim_{k\boxplus} w$ 使得 $w' \in \mathfrak{L}(\mathcal{A})\}$。

备注 *2.* 对于有限词上的跳跃自动机，有时会将模型定义为允许自动机的转移函数“跳跃”到输入中的任意字母并读取该字母 [20]。对于无限词，这一定义无法体现跳跃自动机（或许可以说是）正确的概念，因为自动机可能会跳过某些字母而永远不再读取它们，例如，对于输入 $(ab)^\omega$ ，只通过跳过所有 b来读取a^ω 。根据我们的定义，这种情况是不允许的，因为 $a^\omega \not\sim (ab)^\omega$。然而，显然仍可以通过考虑词的排列，而不是跳跃式读头，来得到我们的“跳跃”定义。

3 跳跃语言

本节研究自动机 $\mathcal{A}$ 的 $\mathfrak{J}(\mathcal{A})$ 的性质。我们首先使用半线性集的一种扩展来刻画 $\mathfrak{J}(\mathcal{A})$。

3.1 跳跃语言与掩码半线性集

回想一下， $\mathbb{M}^\Sigma = \mathbb{N}_\infty^\Sigma \setminus \mathbb{N}^\Sigma$ 并考虑一个向量 $\boldsymbol{v} \in \mathbb{M}^\Sigma$。直观地说，我们将 $\boldsymbol{v}$ 中的∞ 坐标与有限坐标分开，方法是将 $\boldsymbol{v}$ 写成一个来自 $\mathbb{N}^\Sigma$的向量与一个来自 $\{0, \infty\}^\Sigma \setminus \{\mathbf{0}\}$的向量之和。形式化地，令$= \{0, \infty\}^\Sigma \setminus \{\mathbf{0}\}$为掩码的集合，即元素为 $\mathbf{0}$和 ∞ 且不全为 0 的向量；我们将每个 $\mathfrak{m} \in$ 称为一个掩码。我们分别用 $\mathfrak{m}|_0 = \{\sigma \in \Sigma \mid \mathfrak{m}(\sigma) = 0\}$ 和 $\mathfrak{m}|_\infty = \{\sigma \in \Sigma \mid \mathfrak{m}(\sigma) = \infty\}$ 表示 $\mathfrak{m}$中值为 0 和 ∞ 的坐标。

对于 $\boldsymbol{x} \in \mathbb{N}^\Sigma$和 $\mathfrak{m} \in$，$\boldsymbol{x} \oplus \mathfrak{m} \in \mathbb{M}^\Sigma$为满足以下条件的向量：对于所有 $\sigma \in \Sigma$（$\boldsymbol{x}+\mathfrak{m}$）$_\sigma = \boldsymbol{x}_\sigma$，若 $\sigma \in \mathfrak{m}|_0$，则（$\boldsymbol{x}+\mathfrak{m}$）$_\sigma = \infty$，若 $\sigma \in \mathfrak{m}|_\infty$。注意，每个 $\boldsymbol{v} \in \mathbb{M}^\Sigma$都可以写成 $\boldsymbol{x} + \mathfrak{m}$，其中 $\boldsymbol{x}$ 由 $\boldsymbol{v}$ 替换 ∞ 为 0（事实上也可以替换为任意数值）得到，并且令 $\mathfrak{m}$ 与 $\boldsymbol{v}$的 ∞ 坐标相匹配。

现在我们为半线性集添加掩码。掩码半线性集 是形如 $S = \bigcup_{\mathfrak{m}\in} S_{\mathfrak{m}} + \mathfrak{m}$ 的并集，其中对于每个 $\mathfrak{m}$， $S_{\mathfrak{m}}$ 都是一个半线性集。我们还将 S 解释为 $\mathbb{M}^{\Sigma}$ 的一个子集，方法是将加法解释为向集合 $S_{\mathfrak{m}}$中的每个向量添加 $\mathfrak{m}$ 。注意，上述并集是不相交的，因为添加不同的掩码总会产生不同的向量。

考虑一个掩码 $\mathfrak{m} \in$ 。如果两个向量 $\boldsymbol{u}$、$\boldsymbol{v} \in \mathbb{N}^{\Sigma}$ 满足$\boldsymbol{u} + \mathfrak{m} = \boldsymbol{v} + \mathfrak{m}$（即它们在 $\mathfrak{m}|_0$上相同），则称它们 $\mathfrak{m}$-等价。我们称半线性集 $R \subseteq \mathbb{N}^{\Sigma}$是 $\mathfrak{m}$-无关的，如果对于任意 $\mathfrak{m}$-等价的向量 $\boldsymbol{u}$、$\boldsymbol{v}$，都有 $\boldsymbol{u} \in R \iff \boldsymbol{v} \in R$。直观地说，一个 $\mathfrak{m}$-无关的集合不会“查看” $\mathfrak{m}|_{\infty}$。

如果上文的掩码半线性集 S 中的每个 $S_{\mathfrak{m}}$都是 $\mathfrak{m}$-无关的，我们就称其为无关的。请注意，无关这一性质指的是 S 的一种特定表示，其中每个掩码 $\mathfrak{m}$ 对应一个半线性集 $S_{\mathfrak{m}}$ 。从某种意义上说，掩码半线性集具有无关性是很自然的，因为从语义上看，向量 $S_{\mathfrak{m}}$ 中添加$\mathfrak{m}$ 本就忽略了 $\mathfrak{m}$的 ∞ 坐标，因此 $S_{\mathfrak{m}}$ 不应“在意”它们。此外，如果集合 $S_{\mathfrak{m}}$ 是 $\mathfrak{m}$-无关的，那么直观地说，在它的每个线性分量中，我们可以将其周期向量划分为两类：一类在所有被掩码的坐标上均为 0，另一类则允许在被掩码的坐标上取任意值。我们将此称为规范的$\mathfrak{m}$-无关表示，具体如下。

定义 2。 若半线性集 $R \subseteq \mathbb{N}^{\Sigma}$对于掩码 $\mathfrak{m}$ 满足以下条件，则称其为规范的 $\mathfrak{m}$ 无关形式：对于$\mathrm{Lin}(\boldsymbol{b_i}, P_i)$，对每个 $1 \le i \le k$，都有 $P_i = Q_i \cup E_{\mathfrak{m}}$：

- $\boldsymbol{b_i}(\sigma) = 0$ 对每个 $\sigma \in \mathfrak{m}|_{\infty}$ 均成立。- $\boldsymbol{p}(\sigma) = 0$ 对每个 $p \in Q_i$ 和 $\sigma \in \mathfrak{m}|_{\infty}$ 均成立。- $E_{\mathfrak{m}} = \{\boldsymbol{e_\sigma} \mid \sigma \in \mathfrak{m}|_{\infty}\}$ 当 $\sigma = \tau$ 时为 $\boldsymbol{e_\sigma}(\tau) = 1$，否则为$0$ 。

现在我们证明，每个掩码半线性集都可以转换为与之等价的规范的无关形式（示例见图 1）。

$$\mathrm{Lin}\left(\begin{pmatrix}1\\0\\\mathbf{9}\\\mathbf{4}\end{pmatrix}, \left\{\begin{pmatrix}1\\2\\\mathbf{5}\\\mathbf{7}\end{pmatrix}, \begin{pmatrix}1\\0\\\mathbf{1}\\\mathbf{3}\end{pmatrix}\right\}\right) + \begin{pmatrix}0\\0\\\infty\\\infty\end{pmatrix} \quad \rightsquigarrow \quad \mathrm{Lin}\left(\begin{pmatrix}1\\0\\0\\0\end{pmatrix}, \left\{\begin{pmatrix}1\\2\\0\\0\end{pmatrix}, \begin{pmatrix}1\\0\\0\\0\end{pmatrix}, \begin{pmatrix}0\\0\\1\\0\end{pmatrix}, \begin{pmatrix}0\\0\\0\\1\end{pmatrix}\right\}\right) + \begin{pmatrix}0\\0\\\infty\\\infty\end{pmatrix}$$

图 1。 左侧是一个掩码半线性集（实际上是线性集），右侧是与之等价的规范的无关形式。左侧的粗体数字会被掩码去除，因此替换为 0。

引理 1. 考虑一个掩码半线性集 $S = \bigcup_{\mathfrak{m}\in} S_{\mathfrak{m}} + \mathfrak{m}$。那么存在一个无关掩码半线性集 $T = \bigcup_{\mathfrak{m}\in} T_{\mathfrak{m}} + \mathfrak{m}$，它表示 $\mathbb{M}^{\Sigma}$ 的同一子集，并且每个 $T_{\mathfrak{m}}$ 都是规范的 $\mathfrak{m}$-无关形式。

证明。 直观而言，我们通过为每个 $S_{\mathfrak{m}}$向量添加一项，得到不经意表示，其中 $\mathfrak{m}$ 的 ∞-坐标可以取任意值。

形式化地，对于每个掩码 $\mathfrak{m} \in$，定义

$$T_{\mathfrak{m}} = \{\boldsymbol{v} \in \mathbb{N}^{\Sigma} \mid \exists \boldsymbol{u} \in S_{\mathfrak{m}} \text{ such that } \boldsymbol{u} + \mathfrak{m} = \boldsymbol{v} + \mathfrak{m}\}$$

也就是说， $T_{\mathfrak{m}}$ 是通过纳入所有与 $S_{\mathfrak{m}}$中的某个向量 $\mathfrak{m}$-等价的向量而得到的。

根据构造可知， $T_{\mathfrak{m}}$ 是 $\mathfrak{m}$-无关的。此外，很容易看出 $S_{\mathfrak{m}} + \mathfrak{m} = T_{\mathfrak{m}} + \mathfrak{m}$。事实上， $S_{\mathfrak{m}} \subseteq T_{\mathfrak{m}}$，因此得到了一个包含关系；对于每个 $\boldsymbol{x} \in T_{\mathfrak{m}} + \mathfrak{m}$，将 $\boldsymbol{x} = \boldsymbol{v} + \mathfrak{m}$ 写成某个 $\boldsymbol{v} \in T_{\mathfrak{m}}$，那么就存在某个$\boldsymbol{u} \in S_{\mathfrak{m}}$ 使得 $\boldsymbol{v} + \mathfrak{m} = \boldsymbol{u} + \mathfrak{m} \in S_{\mathfrak{m}} + \mathfrak{m}$。

接下来要证明 $T_{\mathfrak{m}}$ 是半线性的，并将其表示为规范形式。为此，对每个向量 $\boldsymbol{p} \in \mathbb{N}^{\Sigma}$，定义 $\boldsymbol{p}|_{\mathfrak{m}}$ ：当 $\sigma \in \mathfrak{m}|_0$时，令 $\boldsymbol{p}|_{\mathfrak{m}}(\sigma) = \boldsymbol{p}(\sigma)$；当 $\sigma \in \mathfrak{m}|_{\infty}$时，令 $\boldsymbol{p}|_{\mathfrak{m}}(\sigma) = 0$ 。也就是说，我们将 $\boldsymbol{p}$中 $\mathfrak{m}$ 的所有 ∞ 坐标替换为 0。我们将此定义推广到向量集：对于集合 P，令 $P|_{\mathfrak{m}} = \{\boldsymbol{p}|_{\mathfrak{m}} \mid \boldsymbol{p} \in P\}$。

现在，写作 $S_{\mathfrak{m}} = \bigcup_{i \in I} \mathrm{Lin}(\boldsymbol{b_i}, P_i)$，我们断言如下：

$$T_{\mathfrak{m}} = \bigcup_{i \in I} \mathrm{Lin}(\boldsymbol{b_i}|_{\mathfrak{m}}, P_i|_{\mathfrak{m}} \cup E_{\mathfrak{m}})$$

也就是说，将 $T_{\mathfrak{m}}$ 在 $S_{\mathfrak{m}}$ 中的 ∞ 坐标置零，然后按照定义 2，任意地在这些坐标上添加数值，即可得到 $T_{\mathfrak{m}}$ ；使用 $\boldsymbol{e_\sigma} \in E_{\mathfrak{m}}$ 。

设 $\boldsymbol{v} \in T_{\mathfrak{m}}$，则存在 $\boldsymbol{u} \in S_{\mathfrak{m}}$ 使得 $\boldsymbol{v} + \mathfrak{m} = \boldsymbol{u} + \mathfrak{m}$。也就是说，对所有 $\sigma \in \mathfrak{m}|_0$，$\boldsymbol{u}(\sigma) = \boldsymbol{v}(\sigma)$。由于 $\boldsymbol{u} \in S_{\mathfrak{m}}$，我们可以将 $\boldsymbol{u} = \boldsymbol{b_i} + \lambda_1 p_1 + \ldots + \lambda_k p_k$ 写成某些 $i \in I$、 $p_1, \ldots, p_k \in P_i$ 和 $\lambda_1, \ldots, \lambda_k \in \mathbb{N}$ 的形式。于是可得

$$\boldsymbol{v} = \boldsymbol{b_i}|_{\mathfrak{m}} + \lambda_1 p_1|_{\mathfrak{m}} + \ldots + \lambda_k p_k|_{\mathfrak{m}} + \sum_{\sigma \in \mathfrak{m}|_{\infty}} \boldsymbol{v}(\sigma) \boldsymbol{e_\sigma}$$

而后一种形式属于 $\mathrm{Lin}(\boldsymbol{b_i}|_{\mathfrak{m}}, P_i|_{\mathfrak{m}} \cup E_{\mathfrak{m}})$。

反过来，设 $\boldsymbol{v} \in \mathrm{Lin}(\boldsymbol{b_i}|_{\mathfrak{m}}, P_i|_{\mathfrak{m}} \cup E_{\mathfrak{m}})$，并写作

$$\boldsymbol{v} = \boldsymbol{b_i}|_{\mathfrak{m}} + \lambda_1 p_1|_{\mathfrak{m}} + \ldots + \lambda_k p_k|_{\mathfrak{m}} + \sum_{\sigma \in \mathfrak{m}|_{\infty}} \beta_\sigma \boldsymbol{e_\sigma}$$

定义 $\boldsymbol{u} = \boldsymbol{b_i} + \lambda_1 p_1 + \ldots + \lambda_k p_k$，然后定义 $\boldsymbol{u} \in S_{\mathfrak{m}}$和 $\boldsymbol{u} + \mathfrak{m} = \boldsymbol{v} + \mathfrak{m}$（因为 $\boldsymbol{v}$ 与 $\boldsymbol{u}$ 不同的唯一坐标是 $\mathfrak{m}|_{\infty}$中的坐标），因此 $\boldsymbol{v} \in T_{\mathfrak{m}}$，证毕。

最后定义 $T = \bigcup_{\mathfrak{m} \in} T_{\mathfrak{m}} + \mathfrak{m}$。它是规范形式的无关掩码半线性集，符合要求。□

复杂度分析 *1*（引理 *1*）。T的描述需要在每个半线性集的周期中至多引入 $|\Sigma|$ 个向量，并将现有向量中的一些条目改为 0。因此， T 的描述相对于 S 的描述规模为多项式。

本节的主要结果是，跳跃语言的 Parikh 像与掩码半线性集相同。我们将在以下引理中证明这一点。

引理 2. 设 $\mathcal{A}$ 是一个自动机，则 $\Psi(\mathfrak{J}(\mathcal{A}))$ 是一个掩码半线性集。此外，我们可以根据 $\mathcal{A}$ 的表示有效地计算出它的一个表示。

证明。考虑 $\mathcal{A} = \langle \Sigma, Q, \delta, Q_0, \alpha \rangle$。根据定义 1，我们有 $\Psi(\mathfrak{J}(\mathcal{A})) = \Psi(\mathfrak{L}(\mathcal{A}))$。事实上，这两个集合都恰好包含 $\mathfrak{L}(\mathcal{A})$ 中词的 Parikh 像。回想一下（或参见附录 A.1），每个 ω-正则语言都可以（有效地）写成如下并集：$\mathfrak{L}(\mathcal{A}) = \bigcup_{i=1}^{m} S_i \cdot T_i^\omega$，其中 $S_i, T_i \subseteq \Sigma^*$ 是有限词上的正则语言。

直观而言，我们将证明 $\Psi(T_i^\omega)$ 可以分解为有限次出现的字母和无限次出现的字母，后者将导出掩码。为此，对于每个 $\emptyset \neq \Gamma \subseteq \Sigma^*$，定义 $\mathfrak{m}_\Gamma \in$，令 $\mathfrak{m}_\Gamma(\sigma) = \infty$ 当 $\sigma \in \Gamma$ 时，且 $\mathfrak{m}_\Gamma(\sigma) = 0$ 当 $\sigma \notin \Gamma$ 时。现在，对于上面并集中的每个正则语言T_i，定义

$$I(T_i) = \{\mathfrak{m}_\Gamma \mid \forall \sigma \in \Gamma \ \exists w \in T_i \cap \Gamma^* \text{ s.t. } \Psi(w)(\sigma) > 0\}$$

也就是说，$I(T_i)$ 是满足以下条件的掩码 $\mathfrak{m}_\Gamma$ 的集合：Γ 中的每个字母都出现在某个 T_i 中的词里，且该词仅包含来自 Γ 的字母。直观而言，通过无限连接来自 T_i 的词并覆盖 Γ 中的所有字母，同时不使用 Γ 以外的字母，就可以得到 $\mathfrak{m}_\Gamma \in I(T_i)$。

现在我们声称，对于每个 $1 \leq i \leq m$，都有 $\Psi(S_i \cdot T_i^\omega) = \Psi(S_i \cdot T_i^*) + I(T_i)$。为使表达简洁，本证明中省略下标 i。考虑 $\boldsymbol{v} \in \Psi(S \cdot T^\omega)$，则存在一个词 $w = x_1 x_2 \cdots$，使得 $\Psi(w) = \boldsymbol{v}$，并且 $x_1 \in S$ 和 $x_j \in T$ 对所有 $j > 1$均成立。令 Γ为在 w中无限次出现的字母集合，并令 $N_0 \geq 2$ 满足：对所有 $n \geq N_0$，都有 $x_n \in \Gamma^*$。注意到 $\mathfrak{m}_\Gamma \in I(T)$，因为对于每个 $\sigma \in \Gamma$，都存在某个词 $x_j \in T_i \cap \Gamma^*$（针对 $j \geq N_0$），使得 σ出现在 x_j中。因此，$\Psi(x_{N_0} x_{N_0+1} \cdots) = \mathfrak{m}_\Gamma$。此外，$x_1 \cdots x_{N_0-1} \in S \cdot T^*$，所以 $\boldsymbol{v} = \Psi(w) = \Psi(x_1 \cdots x_{N_0-1}) + \mathfrak{m}_\Gamma \in \Psi(S \cdot T^*) + I(T)$。

反过来，设 $\boldsymbol{v} \in \Psi(S \cdot T^*) + I(T)$，则存在 $y \in S \cdot T^*$ 和 $\mathfrak{m}_\Gamma \in I(T)$，其中某个 $\Gamma \subseteq \Sigma$满足 $\boldsymbol{v} = \Psi(y) + \mathfrak{m}_\Gamma$。由于 $\mathfrak{m}_\Gamma \in I(T)$，存在一个词集 $\{z_1, \ldots, z_r\} \subseteq T \cap \Gamma^*$，使得每个 $\sigma \in \Gamma$都出现在该集合中的某个词里。因此，我们有 $\Psi((z_1 \cdots z_r)^\omega) = \mathfrak{m}_\Gamma$和 $(z_1 \cdots z_r)^\omega \in T^\omega$，所以 $y \cdot (z_1 \cdots z_r)^\omega \in S \cdot T^\omega$，并且有 $\boldsymbol{v} = \Psi(y) + \mathfrak{m}_\Gamma = \Psi(y \cdot (z_1 \cdots z_r)^\omega) \in \Psi(S \cdot T^\omega)$，从而证明了我们的结论。

由此可得 $\Psi(\mathfrak{J}(\mathcal{A})) = \bigcup_{i=1}^{k} \Psi(S_i \cdot T_i^*) + I(T_i)$。现在我们可以按掩码而非按索引 i 重新排列求和：对于每个掩码 $\mathfrak{m} \in$，写作 $S_\mathfrak{m} = \bigcup_{i:\ \mathfrak{m} \in I(T_i)} \Psi(S_i \cdot T_i^*)$（注意，$S_\mathfrak{m}$可以为空）。由于 $S_i \cdot T_i^*$ 是正规语言，根据帕里克定理 [21]，其 Parikh 像是半线性的，因此 $S_\mathfrak{m}$ 是半线性的。因此，$\Psi(\mathfrak{J}(\mathcal{A})) = \bigcup_{\mathfrak{m} \in} S_\mathfrak{m} + \mathfrak{m}$ 是一个掩码半线性集。□

复杂度分析 *2*（引理 *2* 的复杂度分析）。将 $\mathfrak{L}(\mathcal{A})$写成并集只会导致多项式级膨胀（见附录 A.1）。接着，我们将所得表达式拆分为按 $2^{|\Sigma|}$ 掩码划分的并集。在该并集中，我们把 $S_i \cdot T_i^*$的非确定性自动机转换为半线性集。根据 [23, 定理4。1]（另见 [15]），所得半线性集的描述规模关于 $\mathcal{A}$ 的状态数n 呈多项式级，关于 $|\Sigma|$呈单指数级。此外，该转换可在 $2^{O(|\Sigma|^2 \log(n|\Sigma|))}$时间内计算。

对于反向结论，我们给出一个更强的结果，即可以构造一个确定性自动机来表示一个掩码半线性集。

引理 3。考虑一个掩码半线性集 S，则存在一个确定性自动机 $\mathcal{A}$，使得 $\Psi(\mathfrak{J}(\mathcal{A})) = S$。

证明。根据引理 1，不妨设 $S = \bigcup_{\mathfrak{m}\in} S_{\mathfrak{m}} + \mathfrak{m}$，并且每个 $S_{\mathfrak{m}}$ 都处于规范的 $\mathfrak{m}$-无关形式。令 $\mathfrak{m} \in$，并按照定义 2 将 $S_{\mathfrak{m}} = \bigcup_{i=1}^{k}$ 写作 $\mathrm{Lin}(\boldsymbol{b_i}, P_i \cup E_{\mathfrak{m}})$。考虑上述并集中的一个线性集 $L = \mathrm{Lin}(\boldsymbol{b}, P \cup E_{\mathfrak{m}})$，为简洁起见省略下标 i。我们首先构造一个确定性自动机 $\mathcal{D}$，使得 $\Psi(\mathfrak{J}_{\mathrm{fin}}(\mathcal{D})) = L$。为此，令 $w_{\boldsymbol{b}} \in \Sigma^*$ 为一个词，使得 $\Psi(w_{\boldsymbol{b}}) = \boldsymbol{b}$；类似地，对每个 $\boldsymbol{p} \in P$，令 $w_{\boldsymbol{p}} \in \Sigma^*$ 使得 $\Psi(w_{\boldsymbol{p}}) = \boldsymbol{p}$。注意，对每个 $\sigma \in \mathfrak{m}|_{\infty}$，都有 $\Psi(\sigma) = \boldsymbol{e_\sigma}$。

设 $\mathcal{D}'$ 是正则表达式 $w_{\boldsymbol{b}} \cdot (w_{\boldsymbol{p_1}} + \ldots + w_{\boldsymbol{p_n}})^*$ 的确定性自动机，其中 $P = \{\boldsymbol{p_1}, \ldots, \boldsymbol{p_n}\}$。接下来，令 $w_E = \sigma_1 \cdots \sigma_k$ 为按某种顺序连接 $\mathfrak{m}|_{\infty}$ 中所有字母得到的词。我们通过将 $\mathcal{D}'$ 的每个接受状态在读取 σ_1 后连接到一个允许读取 w_E^{ω} 的环，从而由 $\mathcal{D}'$ 构造得到 $\mathcal{D}$。$\mathcal{D}$ 的接受状态是位于 w_E 环上的那些状态。关键在于，$\mathcal{D}'$ 在读取 σ_1 后的转移仍保持确定性，因为 P 具有规范形式，因此对于每个 $\boldsymbol{p} \in P$，都有 $\boldsymbol{p}(\sigma_1) = 0$。也就是说，字母 σ_1 不会出现在 $\mathcal{D}'$ 的任何转移中，因此可以在构造 $\mathcal{D}$ 时使用它。图 2 展示了这一构造。

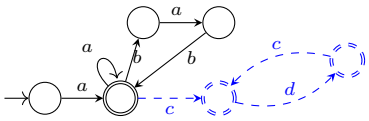


图 2. 用于表示图 1 中线性集（在 $\Sigma = \{a, b, c, d\}$ 上）的自动机 $\mathcal{D}$，其周期的代表词为 $w_{\boldsymbol{b}} = a$ 和 $\{bab, a, c, d\}$。蓝色（虚线）部分表示添加 w_E，并更改接受状态，从而由 $\mathcal{D}'$ 得到 $\mathcal{D}$。

根据构造，我们有 $\mathfrak{L}(\mathcal{D}) = w_{\boldsymbol{b}} \cdot (w_{\boldsymbol{p_1}} + \ldots + w_{\boldsymbol{p_n}})^* \cdot w_E^{\omega}$。由于$\Psi(w_E^{\omega}) = \mathfrak{m}$, 因此我们现在有

$$\Psi(\mathfrak{J}(\mathcal{D})) = \Psi(\mathfrak{L}(\mathcal{D})) = \Psi(w_{\boldsymbol{b}} \cdot (w_{\boldsymbol{p_1}} + \ldots + w_{\boldsymbol{p_n}})^* \cdot w_E^{\omega}) = \mathrm{Lin}(\boldsymbol{b}, P) + \mathfrak{m} = L + \mathfrak{m}$$

其中，最后一个等式成立是因为将 $\mathfrak{m}$ 加到向量上，会掩盖通过 ∞ 对 $E_{\mathfrak{m}}$ 中的向量进行的任何加法。

接下来，我们观察到，对于两个确定性自动机 $\mathcal{D}_1$，$\mathcal{D}_2$，我们可以构造一个确定性自动机 $\mathcal{D}_3$，使得 $\mathfrak{J}(\mathcal{D}_3) = \mathfrak{J}(\mathcal{D}_1) \cup \mathfrak{J}(\mathcal{D}_2)$。事实上，标准的乘积构造（其中接受状态是满足 $\mathcal{D}_1$ 或 $\mathcal{D}_2$ 的状态对）保持确定性。因此，我们可以先通过乘积捕获 $S_{\mathfrak{m}} = \bigcup_{i=1}^{k} \mathrm{Lin}(\boldsymbol{b_i}, P_i \cup E_{\mathfrak{m}})$，再捕获 $S = \bigcup_{\mathfrak{m}\in} S_{\mathfrak{m}} + \mathfrak{m}$，从而达到所需效果。□

复杂度分析 *3*（引理 *3*）。我们可以直接为正则表达式 $w_{\boldsymbol{b}} \cdot (w_{\boldsymbol{p_1}} + \ldots + w_{\boldsymbol{p_n}})^*$ 构造一个大小为 $\|\mathrm{Lin}(\boldsymbol{b},\ P)\| = \|\boldsymbol{b}\| + \sum_{i=1}^{n} \|\boldsymbol{p_i}\|$ 的自动机，例如，$\|\boldsymbol{b}\|$ 是 $\boldsymbol{b}$ 中各项之和（即一元表示）。随后，确定化可以得到一个大小为单指数级的确定有限自动机。接着，我们通过乘积构造取并集，副本数对应于 S 中线性集的数量，进而对应于 $2^{|\Sigma|} - 1$ 个掩码。由此得到的确定有限自动机，其大小相对于 S 为单指数级，相对于 $|\Sigma|$ 为双指数级（假设 S 采用一元表示）。最后，为掩码添加环，最多会增加 $2^{O(|\Sigma|)}$ 个状态（无论是确定性自动机还是非确定性自动机）。

结合引理2和引理3，我们得到以下结论。

定理4。 集合 S 是掩码半线性集，当且仅当存在一个自动机 $\mathcal{A}$ 使得 $S = \Psi(\mathfrak{J}(\mathcal{A}))$。

3.2 跳跃语言——闭包性质

利用第3.1节得到的刻画，我们现在可以得出跳跃语言的若干闭包性质。首先，跳跃语言显然对并集封闭：只需取各自动机的并集，并非确定性地选择从哪个自动机开始。接下来，我们将讨论交集和补运算。

注意，标准的 Büchi 自动机交集构造无法表示跳跃语言的交集，如图3所示。



图3. 自动机 $\mathcal{A}$ 和 $\mathcal{B}$ 满足 $\mathfrak{L}(\mathcal{A}) = \{(ab)^\omega\}$ 和 $\mathfrak{L}(\mathcal{B}) = \{(ba)^\omega\}$。因此，$\mathfrak{J}(\mathcal{A}) = \mathfrak{J}(\mathcal{B}) = \{w \in \Sigma^\omega \mid w$ 有无限多个 a和 $b\}$。然而，如果我们先取 $\mathcal{A}$ 和 $\mathcal{B}$的标准交集，最终得到的将是空语言。

命题 1. 设 $\mathcal{A}$ 和 $\mathcal{B}$ 为自动机，则我们可以有效地构造一个自动机 $\mathcal{C}$，使得 $\mathfrak{J}(\mathcal{C}) = \mathfrak{J}(\mathcal{A}) \cap \mathfrak{J}(\mathcal{B})$。

证明。 考虑一个词 $w \in \Sigma^\omega$，并注意到 $w \in \mathfrak{J}(\mathcal{A}) \cap \mathfrak{J}(\mathcal{B})$ 当且仅当 $\Psi(w) \in \Psi(\mathfrak{J}(\mathcal{A})) \cap \Psi(\mathfrak{J}(\mathcal{B}))$。因此，只需证明存在自动机 $\mathcal{C}$，使得 $\Psi(\mathfrak{J}(\mathcal{C})) = \Psi(\mathfrak{J}(\mathcal{A})) \cap \Psi(\mathfrak{J}(\mathcal{B}))$。由定理4可知，我们可以写出 $\Psi(\mathfrak{J}(\mathcal{A})) = \bigcup_{\mathfrak{m}\in} S_{\mathfrak{m}} + \mathfrak{m}$和 $\Psi(\mathfrak{J}(\mathcal{B})) = \bigcup_{\mathfrak{m}\in} T_{\mathfrak{m}} + \mathfrak{m}$。此外，由引理 1 可知，我们可以假设这些是无关掩码半线性集。

我们声称 $\Psi(\mathfrak{J}(\mathcal{A})) \cap \Psi(\mathfrak{J}(\mathcal{B})) = \bigcup_{\mathfrak{m}\in} S_{\mathfrak{m}} \cap T_{\mathfrak{m}} + \mathfrak{m}$。也就是说，我们可以通过对每一对半线性集取交集来求交集，同时保留掩码。直观地说，这是因为每个词都唯一确定了它可以用来加入并集的掩码，因此对并集取交集相当于（利用盲性）

对与该语言中的词的 Parikh 像的有限部分相对应的有限值向量集合取交集。

确实，令 $\boldsymbol{x} \in \bigcup_{\mathfrak{m}\in} S_{\mathfrak{m}} \cap T_{\mathfrak{m}} + \mathfrak{m}$，则存在 $\mathfrak{m} \in$ 和 $\boldsymbol{v} \in S_{\mathfrak{m}} \cap T_{\mathfrak{m}}$ 使得 $\boldsymbol{x} = \boldsymbol{v} + \mathfrak{m}$。特别地，$\boldsymbol{v} + \mathfrak{m} \in \Psi(\mathfrak{J}(\mathcal{A}))$ 和 $\boldsymbol{v} + \mathfrak{m} \in \Psi(\mathfrak{J}(\mathcal{B}))$，因此 $\boldsymbol{x} = \boldsymbol{v} + \mathfrak{m} \in \Psi(\mathfrak{J}(\mathcal{A})) \cap \Psi(\mathfrak{J}(\mathcal{B}))$。

反过来，令 $\boldsymbol{x} \in \Psi(\mathfrak{J}(\mathcal{A})) \cap \Psi(\mathfrak{J}(\mathcal{B}))$。回想一下，$\boldsymbol{x}$唯一确定了掩码 $\mathfrak{m}$，因此存在 $\boldsymbol{u} \in S_{\mathfrak{m}}$、$\boldsymbol{v} \in T_{\mathfrak{m}}$，使得 $\boldsymbol{x} = \boldsymbol{u} + \mathfrak{m} = \boldsymbol{v} + \mathfrak{m}$。由此可知，$\boldsymbol{u}$ 和 $\boldsymbol{v}$ 是 $\mathfrak{m}$-等价的。由于 $S_{\mathfrak{m}}$ 和 $T_{\mathfrak{m}}$ 是 $\mathfrak{m}$-无关的，我们还可得 $\boldsymbol{v} \in S_{\mathfrak{m}}$ 和 $\boldsymbol{u} \in T_{\mathfrak{m}}$。特别地，$\boldsymbol{v} \in S_{\mathfrak{m}} \cap T_{\mathfrak{m}}$，因此 $\boldsymbol{x} = \boldsymbol{v} + \mathfrak{m} \in \bigcup_{\mathfrak{m}\in} S_{\mathfrak{m}} \cap T_{\mathfrak{m}} + \mathfrak{m}$。

最后，半线性集在交集 [3,7]下封闭。因此，对于每个 $\mathfrak{m}$，$S_{\mathfrak{m}} \cap T_{\mathfrak{m}}$ 都是半线性的，所以 $\bigcup_{\mathfrak{m}\in} S_{\mathfrak{m}} \cap T_{\mathfrak{m}} + \mathfrak{m}$ 是一个半线性掩码集；根据引理 3，存在一个自动机 $\mathcal{C}$，使得 $\mathfrak{J}(\mathcal{C}) = \mathfrak{J}(\mathcal{A}) \cap (\mathfrak{J}(\mathcal{B})$。□

复杂度分析 *5*（命题 *1*）。半线性集交集的描述规模关于各向量的描述规模和并集的大小呈多项式级，并关于 $|\Sigma|$ [3,7] 呈单指数级。由于引理 1 和引理 2 带来的膨胀是多项式级的，因此唯一显著的膨胀来自引理 3，这是因为构造了一个大小关于$|\Sigma|$ 呈指数级的自动机。不过不难看出，$|\Sigma|$ 中的两次指数级膨胀彼此正交，可以合并为一次单指数级膨胀。

命题2。考虑一个自动机 $\mathcal{A}$。存在一个自动机 $\mathcal{B}$，使得 $\mathfrak{J}(\mathcal{B}) = \Sigma^{\omega} \setminus \mathfrak{J}(\mathcal{A})$。

证明。注意到 $w \in \Sigma^{\omega} \setminus \mathfrak{J}(\mathcal{A})$ 当且仅当 $\Psi(w) \in \mathbb{M}^{\Sigma} \setminus \Psi(\mathfrak{J}(\mathcal{A}))$。事实上，$w \in \Sigma^{\omega} \setminus \mathfrak{J}(\mathcal{A})$ 意味着 w 的每一种排列都不被 $\mathcal{A}$ 接受，也就是说，w 的 Parikh 像不是 $\mathcal{A}$ 所接受的任何词的 Parikh 像。因此，只需构造 $\mathcal{B}$，使得 $\Psi(\mathfrak{J}(\mathcal{B})) = \mathbb{M}^{\Sigma} \setminus \Psi(\mathfrak{J}(\mathcal{A}))$。

根据定理4，我们可以将 $\Psi(\mathfrak{J}(\mathcal{A})) = \bigcup_{\mathfrak{m}\in} S_{\mathfrak{m}} + \mathfrak{m}$写成这种形式。此外，根据引理 1，我们可以假设这是一个无关掩码半线性集。我们声称$\mathbb{M}^{\Sigma} \setminus \Psi(\mathfrak{J}(\mathcal{A})) = \bigcup_{\mathfrak{m}\in} (\mathbb{N}^{\Sigma} \setminus S_{\mathfrak{m}}) + \mathfrak{m}$。也就是说，我们可以通过对每个半线性集取补来对并集取补，同时保留掩码。

为证明这一点，考虑一个词 $w \in \Sigma^{\omega}$，并按照第3.1节的定义写作 $\Psi(w) = \boldsymbol{x_w} + \mathfrak{m}_w$ where$\boldsymbol{x_w} \in \mathbb{N}^{\Sigma}$。

对于第一个方向，假设 $\boldsymbol{x_w} + \mathfrak{m}_w \in \mathbb{M}^{\Sigma} \setminus \Psi(\mathfrak{J}(\mathcal{A}))$，则 $\boldsymbol{x_w} + \mathfrak{m}_w \notin \bigcup_{\mathfrak{m}\in} S_{\mathfrak{m}} + \mathfrak{m}$。特别地，$\boldsymbol{x_w} \notin S_{\mathfrak{m}_w}$（否则我们会有 $\boldsymbol{x_w} + \mathfrak{m}_w \in S_{\mathfrak{m}_w} + \mathfrak{m}_w$，而它属于并集）。由此可得 $\boldsymbol{x_w} + \mathfrak{m}_w \in (\mathbb{N}^{\Sigma} \setminus S_{\mathfrak{m}_w}) + \mathfrak{m}_w$，因此 $\Psi(w) \in \bigcup_{\mathfrak{m}\in} (\mathbb{N}^{\Sigma} \setminus S_{\mathfrak{m}}) + \mathfrak{m}$。

反过来，假设 $\Psi(w) = \boldsymbol{x_w} + \mathfrak{m}_w \in \bigcup_{\mathfrak{m}\in} (\mathbb{N}^{\Sigma} \setminus S_{\mathfrak{m}}) + \mathfrak{m}$，注意到 $\boldsymbol{x_w} + \mathfrak{m}_w \in (\mathbb{N}^{\Sigma} \setminus S_{\mathfrak{m}_w}) + \mathfrak{m}_w$。事实上，并集的其他部分都没有掩码$\mathfrak{m}_w$。反设 $\boldsymbol{x_w} + \mathfrak{m}_w \in \Psi(\mathfrak{J}(\mathcal{A})) = \bigcup_{\mathfrak{m}\in} S_{\mathfrak{m}} + \mathfrak{m}$，则根据同样的推理，$\boldsymbol{x_w} + \mathfrak{m}_w \in S_{\mathfrak{m}_w} + \mathfrak{m}_w$。因此，存在两个向量$\boldsymbol{y} \notin S_{\mathfrak{m}_w}$ 和 $\boldsymbol{z} \in S_{\mathfrak{m}_w}$，使得 $\boldsymbol{y} + \mathfrak{m} = \boldsymbol{x_w} + \mathfrak{m} = \boldsymbol{z} + \mathfrak{m}$。这与 $S_{\mathfrak{m}_w}$ 无关的假设矛盾（注意，如果 $S_{\mathfrak{m}_w}$ 不是无关的，这就不构成矛盾）。

我们得出 $\mathbb{M}^{\Sigma} \setminus \Psi(\mathfrak{J}(\mathcal{A})) = \bigcup_{\mathfrak{m}\in} (\mathbb{N}^{\Sigma} \setminus S_{\mathfrak{m}}) + \mathfrak{m}$，并且由于半线性集在补运算下封闭，[3,7], 后者也是一个掩码半线性

集。通过反向应用定理4，我们得出存在一个自动机 $\mathcal{B}$ 使得 $\Psi(\mathfrak{J}(\mathcal{B})) = \mathbb{M}^{\Sigma} \setminus \Psi(\mathfrak{J}(\mathcal{A}))$，并且如上所述，对于这样的 $\mathcal{B}$，我们有 $\mathfrak{J}(\mathcal{B}) = \Sigma^{\omega} \setminus \mathfrak{J}(\mathcal{A})$。□

复杂度分析*6*（命题*2*）。半线性集补集的描述规模在条目的次数和描述方面都是单指数的 [3,7]。我们采用与复杂度分析5类似的方法，最终得到单指数膨胀（而不仅仅是在 $|\Sigma|$ 中）。

请回忆，Büchi 自动机通常无法确定化。也就是说，存在一个自动机 $\mathcal{A}$，使得 $\mathfrak{L}(\mathcal{A})$ 无法由确定性 Büchi 自动机 [17] 识别。与此形成鲜明对比的是，引理2和3的一个直接推论是，跳跃语言确实可以确定化（另请参见复杂度分析3）。

命题3。对于每个自动机 $\mathcal{A}$，都存在一个确定性自动机 $\mathcal{D}$，使得 $\mathfrak{J}(\mathcal{A}) = \mathfrak{J}(\mathcal{D})$。

特别地，这意味着额外的接受条件（例如 Rabin、Streett、Muller 和奇偶条件）无法增强该模型的表达能力。

备注 *3.* 讨论跳跃语言的确定化容易产生误导：跳跃语言的接受定义要求存在某种排列，使自动机接受该排列。这个存在量词可以视为“语义”非确定性。因此，从某种意义上说，无论底层句法结构如何，跳跃语言本质上都是非确定性的。

最后，与有限词上的跳跃语言一样，[20,10,11]，跳跃语言与 ω-正则语言不可比较。事实上，图3中的自动机是 Büchi 自动机的例子，其语言不是排列不变的，因此不是跳跃语言；而图4则表明反过来同样不成立。

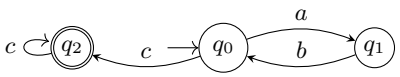


图4. 一个自动机 $\mathcal{A}$，使得 $\mathfrak{J}(\mathcal{A}) = \{u \cdot c^{\omega} \mid u \in \{a,b,c\}^{*} \wedge \#_a[u] = \#_b[u]\}$；根据简单的抽引论证，它不是 ω-正则的。

3.3 跳跃语言-算法性质

现在我们关注跳跃语言的算法性质。首先注意，对于给定的自动机 $\mathcal{A}$，当且仅当 $\mathfrak{L}(\mathcal{A}) \neq \emptyset$ 时，$\mathfrak{J}(\mathcal{A}) \neq \emptyset$。因此，$\mathfrak{J}(\mathcal{A})$ 的非空性是 NL完全问题，与 Büchi 自动机的情况相同 [8,16]。接下来，我们考虑以下标准判定问题：

– **包含关系**：给定自动机 $\mathcal{A}$，$\mathcal{B}$，$\mathfrak{J}(\mathcal{A})$ 是否 $\subseteq \mathfrak{J}(\mathcal{B})$? – **等价性**：给定自动机 $\mathcal{A}$，$\mathcal{B}$，$\mathfrak{J}(\mathcal{A})$ 是否 $= \mathfrak{J}(\mathcal{B})$? – **全域性**：给定自动机 $\mathcal{A}$，$\mathfrak{J}(\mathcal{A})$ 是否 $= \Sigma^\omega$?

我们证明，所有这些问题都可以归约为它们在有限词跳跃自动机中的对应问题。

定理 7. 跳跃语言的包含关系、等价性和全域性是 *coNP* 完全的（对于固定大小字母表）。

证明。 我们从包含关系开始。根据定理4和引理 1，对于给定的自动机$\mathcal{A}$，$\mathcal{B}$我们得到规范的无关掩码半线性表示：$\Psi(\mathfrak{J}(\mathcal{A})) = \bigcup_{\mathfrak{m}\in} S_\mathfrak{m} + \mathfrak{m}$和 $\Psi(\mathfrak{J}(\mathcal{B})) = \bigcup_{\mathfrak{m}\in} T_\mathfrak{m} + \mathfrak{m}$。于是，$\mathfrak{J}(\mathcal{A}) \subseteq \mathfrak{J}(\mathcal{B})$ 当且仅当 $\Psi(\mathfrak{J}(\mathcal{A})) \subseteq \Psi(\mathfrak{J}(\mathcal{B}))$，而这又当且仅当对于每个掩码$\mathfrak{m}$，都有 $S_\mathfrak{m} + \mathfrak{m} \subseteq T_\mathfrak{m} + \mathfrak{m}$。我们声称，后者成立当且仅当 $S_\mathfrak{m} \subseteq T_\mathfrak{m}$。

事实上，显然，如果 $S_\mathfrak{m} \subseteq T_\mathfrak{m}$ 那么 $S_\mathfrak{m} + \mathfrak{m} \subseteq T_\mathfrak{m} + \mathfrak{m}$。反过来，假设 $S_\mathfrak{m} + \mathfrak{m} \subseteq T_\mathfrak{m} + \mathfrak{m}$并令 $\boldsymbol{u} \in S_\mathfrak{m}$，因此 $\boldsymbol{u} + \mathfrak{m} \in T_\mathfrak{m} + \mathfrak{m}$。于是，存在 $\boldsymbol{v} \in T_\mathfrak{m}$使得 $\boldsymbol{u} + \mathfrak{m} = \boldsymbol{v} + \mathfrak{m}$，但由于 $T_\mathfrak{m}$是 $\mathfrak{m}$-无关的，这意味着 $\boldsymbol{u} \in T_\mathfrak{m}$。

因此，判定 $\mathfrak{J}(\mathcal{A}) \subseteq \mathfrak{J}(\mathcal{B})$ 是否成立，等价于判定对于每个 $\mathfrak{m}$，是否有 $S_\mathfrak{m} \subseteq T_\mathfrak{m}$ 。由于后者是可判定的，我们得出包含关系的可判定性。此外，我们可以在多项式时间内构造分别对应于 $S_\mathfrak{m}$ 和 $T_\mathfrak{m}$ 的非确定性自动机$\mathcal{C}_1^\mathfrak{m}$，$\mathcal{C}_2^\mathfrak{m}$（见复杂度分析 1 和 3），因此只需判定对于所有 $\mathfrak{m}$，是否有 $\mathfrak{J}_{\text{fin}}(\mathcal{C}_1^\mathfrak{m}) \subseteq \mathfrak{J}_{\text{fin}}(\mathcal{C}_2^\mathfrak{m})$。后一个问题是 coNP 完全的 [15,14]（对于固定大小的字母表），而且由于我们可以先非确定性地猜测 $\mathfrak{m}$，所以这一复杂度仍然成立。此外，coNP 困难性显然成立（例如，将给定语言 L 归约为 $L\#^\omega$，其中 # 是一个新符号）。

等价性和全域性也可通过类似论证得出，从而完成证明。

□

4 固定窗口跳跃语言

回顾定义 1：给定自动机 $\mathcal{A}$ 和 $k \in \mathbb{N}$，语言$\mathfrak{J}_{k\boxplus}(\mathcal{A})$由具有一个被 $\mathcal{A}$接受的 k-窗口排列的词组成。

由于 k 是固定的，我们可以利用有限记忆捕获 k-窗口排列。因此，正如我们接下来要证明的，$\mathfrak{J}_{k\boxplus}(\mathcal{A})$是 ω-正则的。

定理 8. 考虑一个自动机 $\mathcal{A}$，那么对于每个 $k \in \mathbb{N}$，都存在一个 *Büchi* 自动机 $\mathcal{B}_k$，使得 $\mathfrak{J}_{k\boxplus}(\mathcal{A}) = \mathfrak{L}(\mathcal{B}_k)$。

证明（概要）。直观而言，我们构造 $\mathcal{B}_k$，使其在一个状态中存储 k 个字母的 Parikh 像，然后对所有具有已存储 Parikh 像的词，非确定性地模拟 $\mathcal{A}$，同时记录是否曾经过接受状态。我们在图 5 中以 $k = 2$ 为例说明这一构造。详情见附录 A.2。□

复杂度分析 9（定理 8）。如果 $\mathcal{A}$ 有 n 个状态，那么 $\mathcal{B}$ 至多有 $k^{|\Sigma|} \times n + n$ 个状态。此外，可以有效地计算 $\mathcal{B}$：对于每个 Parikh 像，保留可达状态集合（分别为经由接受状态可达的状态集合）。

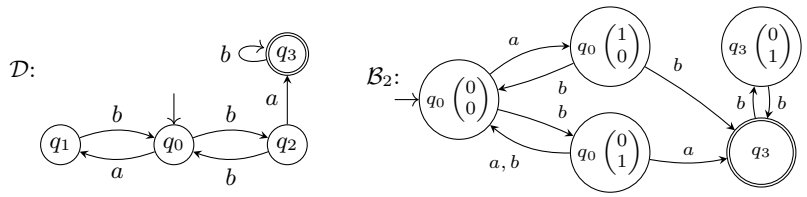


图 5. 一个自动机 $\mathcal{D}$，以及根据定理 8 为 $\mathfrak{J}_{2\boxplus}(\mathcal{D})$ 构造的 $\mathcal{B}_2$。

定理 8 的逆命题也以如下意义成立：如果对于每个 $w \sim_{k\boxplus} w'$，语言 $L \subseteq \Sigma^\omega$ 都是 $k\boxplus$ 不变的，则称其具有不变性 $w \in L \iff w' \in L$。以下结论由定义可得。

命题 4. 设 $\mathcal{A}$ 是一个自动机，且 $\mathfrak{L}(\mathcal{A})$ 具有 $k\boxplus$ 不变性，则 $\mathfrak{L}(\mathcal{A}) = \mathfrak{J}_{k\boxplus}(\mathcal{A})$。

定理 8 和命题 4 表明，k-窗口跳跃语言在通常使 ω-正则语言封闭的运算（并集、交集、补运算）下也封闭，因为这些性质保留了 $k\boxplus$ 不变性。类似地，所有算法问题都可以归约为 Büchi 自动机上的相同问题（根据复杂度分析 9，假设 $|\Sigma|$ 固定）。

注意，定理 8 中的构造得到的是一个非确定性 Büchi 自动机。鉴于备注 3，这并不令人意外。不过，这确实引出了一个问题：对于确定性自动机 $\mathcal{D}$，我们能否为 $\mathfrak{J}_{k\boxplus}(\mathcal{D})$ 找到一个确定性 Büchi 自动机？在命题 5 中，我们证明情况并非如此。也就是说，即使是确定性 k-窗口跳跃，也具有内在的非确定性。

命题 5. 不存在这样一个非确定性 *Büchi* 自动机，其语言为 $\mathfrak{J}_{k\boxplus}(\mathcal{D})$，其中 $k \geq 2$ 和 $\mathcal{D}$ 如图 *5* 所示。

证明. 考虑图 5 中的确定性自动机 $\mathcal{D}$。我们首先证明，确定性 Büchi 自动机无法识别 $\mathfrak{L}(\mathcal{B}_2) = \mathfrak{J}_{2\boxplus}(\mathcal{D})$。下文将对所有偶数 k 以相同方式证明这一点；对于所有 $k \geq 2$，证明则需要稍多一些工作。

反设存在一个确定性自动机 $\mathcal{A}$ 使得$\mathfrak{L}(\mathcal{A}) = \mathfrak{J}_{2\boxplus}(\mathcal{D})$，那么 $\mathcal{A}$ 接受 $(ba)(bb)^\omega$，因此其运行在经过某个 $(ba)(bb)^{m_1}$后经过一个接受状态。接着，$\mathcal{A}$ 也接受 $(ba)(bb)^{m_1}(ab)(bb)^\omega$，所以同样存在 m_2 使得 $\mathcal{A}$ 在经过 $(ba)(bb)^{m_1}(ab)(bb)^{m_2}$后到达一个接受状态。我们考虑词 $(ba)(bb)^{m_1}(ab)(bb)^{m_2}(ab)(bb)^\omega$，并以此方式继续构造一个在含有无限多个 (ab) 的词上的接受运行，而该词不属于 $\mathfrak{J}_{2\boxplus}(\{$v39)。

现在我们转而证明，对于任意 $k \geq 2$，结论同样成立。

反设存在 $k \in \mathbb{N}$，$k \geq 2$ 使得$\mathfrak{J}_{k\boxplus}(\mathcal{D}) = \mathfrak{L}(A)$，其中 $\mathcal{A}$ 是一个确定性自动机。k 可以是奇数，也可以是偶数。

我们首先假设 k 是偶数，并考虑词 $w = bab^{\omega}$。$w \in \mathfrak{J}_{k\boxplus}(\mathcal{D})$，因为存在一个被 $\mathcal{D}$ 接受的 $w' \sim_{k\boxplus} w$，即 $w' = w$。由于 $bab^{\omega} \in \mathfrak{L}(\mathcal{A})$，$\mathcal{A}$ 在其上的唯一运行是接受的，并令 m_1 为一个指标，使得 $|bab^{m_1}|$ 能被 k 整除，且在读取 bab^{m_1} 时至少访问过一次 α。现在，考虑词 $w = bab^{m_1}bab^{\omega}$。$w \in \mathfrak{J}_{k\boxplus}(\mathcal{D})$，因为存在一个被 $\mathcal{D}$ 接受的 $w' \sim_{k\boxplus} w$，即 $w' = abb^{m_1}bab^{\omega}$。$bab^{m_1}bab^{\omega} \in \mathfrak{L}(\mathcal{A})$，那么 $\mathcal{A}$ 在其上的唯一运行是接受的，并令 m_2 为一个指标，使得 $|bab^{m_1}bab^{m_2}|$ 能被 k 整除，且在读取 $bab^{m_1}bab^{m_2}$ 时至少访问过两次 α。考虑词 $w = bab^{m_1}bab^{m_2}bab^{\omega}$；它也属于 $\mathfrak{J}_{k\boxplus}(\mathcal{D})$，因为 $w' = abb^{m_1}abb^{m_2}bab^{\omega}$ 被 $\mathcal{D}$ 接受。照此方式，我们可以构造一个无限词 $w = bab^{m_1}bab^{m_2}bab^{m_3}\ldots$，使得 $\mathcal{A}$ 在其上的运行无限多次访问 α，尽管 $w \notin \mathfrak{J}_{k\boxplus}(\mathcal{D})$。

接下来，我们假设 $k \geq 2$ 为奇数，并考虑词 $w = ab^{\omega}$。由于存在一个被 $\mathcal{D}$接受的 $w' \sim_{k\boxplus} w$，即 $w' = bab^{\omega}$，因此有 $w \in \mathfrak{J}_{k\boxplus}(\mathcal{D})$。由于 $ab^{\omega} \in \mathfrak{L}(\mathcal{A})$，$\mathcal{A}$ 在其上的唯一运行是接受运行。令 m_1 为一个索引，使得$|ab^{m_1}|$ 能被 k 整除，并且在读取 ab^{m_1}时，α 至少被访问过一次。现在，考虑词 $w = ab^{m_1}ab^{\omega}$。同样，由于存在一个被 $\mathcal{D}$接受的 $w' \sim_{k\boxplus} w$，即 $w' = w$（其接受运行是 $\rho = q_0q_1(q_0q_2)^{k-2}q_3^{\omega}$），因此有 $w \in \mathfrak{J}_{k\boxplus}(\mathcal{D})$。由于 $ab^{m_1}ab^{\omega} \in \mathfrak{L}(\mathcal{A})$，$\mathcal{A}$ 在其上的唯一运行是接受运行。令 m_2 为一个索引，使得 $|ab^{m_1}ab^{m_2}|$ 能被 k 整除，并且在读取 $ab^{m_1}ab^{m_2}$时，α 至少被访问过两次。类似地，我们可以证明 $w = ab^{m_1}ab^{m_2}ab^{\omega} \in \mathfrak{J}_{k\boxplus}(\mathcal{D})$，并且可以继续构造一个无限词 $w = ab^{m_1}ab^{m_2}ab^{m_3}\ldots$，使得 $\mathcal{A}$在其上的运行会无限多次访问 α，尽管 $w \notin \mathfrak{J}_{k\boxplus}(\mathcal{D})$。由此可知，不存在确定性自动机 $\mathcal{A}$，使得对于任意 $k > 1$，都有 $\mathfrak{J}_{k\boxplus}(\mathcal{D}) = \mathfrak{L}(A)$。□

5 存在窗口跳跃语言

回顾定义 1，给定一个自动机 $\mathcal{A}$，语言 $\mathfrak{J}_{\exists\boxplus}(\mathcal{A})$ 包含这样的词：它们具有一个 k 窗口排列，该排列对某个 k 在 $\mathcal{A}$ 中被接受，即 $\mathfrak{J}_{\exists\boxplus}(\mathcal{A}) = \bigcup_{k\in\mathbb{N}} \mathfrak{J}_{k\boxplus}(\mathcal{A})$。我们简要说明 $\mathfrak{J}_{\exists\boxplus}(\mathcal{A})$ 的一些表达能力性质。

或许令人惊讶的是，我们证明 $\exists\boxplus$ 语言的表达能力严格强于跳跃语言。我们首先证明，每一种跳跃语言都可以定义为某个自动机的 $\exists\boxplus$ 语言。

定理10. 设 $\mathcal{A}$ 为一个自动机，则存在一个自动机 $\mathcal{B}$，使得 $\mathfrak{J}_{\exists\boxplus}(\mathcal{B}) = \mathfrak{J}(\mathcal{A})$。

证明。 直观地说，我们修改引理 3 中跳跃自动机的构造，使其生成一个自动机，其 $\exists\boxplus$ 语言与给定的半线性集一致。

考虑一个自动机 $\mathcal{A}$。根据引理 1 和引理 2，我们可以将 $\Psi(\mathfrak{J}(\mathcal{A})) = \bigcup_{\mathfrak{m}\in} S_{\mathfrak{m}} + \mathfrak{m}$ 写成规范的无关形式。与引理 3 类似，考虑上述并集中的一个特定线性集 $L = \mathrm{Lin}(\boldsymbol{b}, P \cup E_{\mathfrak{m}})$，并考虑引理 3 中针对 $\boldsymbol{p} \in P$ 的词 $w_{\boldsymbol{b}} \in \Sigma^*$ 和 $w_{\boldsymbol{p}} \in \Sigma^*$。令 $\mathcal{N}'$ 为一个非确定性自动机，使得 $\mathfrak{L}_{\mathrm{fin}}(\mathcal{N}') = w_{\boldsymbol{b}} \cdot (w_{\boldsymbol{p_1}} + \ldots + w_{\boldsymbol{p_n}})^*$。现在，通过将 $\mathcal{N}'$ 的每个接受状态连接到一个新的非确定性自动机 $\mathcal{N}_{\mathfrak{m}}$，我们得到自动机 $\mathcal{N}$。作为 Büchi 自动机，$\mathcal{N}_{\mathfrak{m}}$ 接受这样的语言：其中每个 $\sigma \in \mathfrak{m}|_\infty^\omega$ 都在 $w\}$ 中无限次出现。$\mathcal{N}$ 的接受状态就是 $\mathcal{N}_{\mathfrak{m}}$ 的接受状态。

注意，$L_{\mathfrak{m}}$ 是置换不变的，特别是 $\mathfrak{J}_{\exists\boxplus}(\mathcal{N}_{\mathfrak{m}}) = L_{\mathfrak{m}}$。事实上，显然有 $L_{\mathfrak{m}} = \mathfrak{L}(\mathcal{N}_{\mathfrak{m}}) \subseteq \mathfrak{J}_{\exists\boxplus}(\mathcal{N}_{\mathfrak{m}})$；反过来，若 $w \in \mathfrak{J}_{\exists\boxplus}(\mathcal{N}_{\mathfrak{m}})$，则对于某个 $w' \in \mathfrak{L}(\mathcal{N}_{\mathfrak{m}}) = L_{\mathfrak{m}}$ 和某个 $k \in \mathbb{N}$，有 $w \sim_{k\boxplus} w'$；但由于 $L_{\mathfrak{m}}$ 是置换不变的，且特别有 $w \sim w'$，因此 $w \in L_{\mathfrak{m}}$。

此外，$\Psi(\mathfrak{L}(\mathcal{N})) = \mathrm{Lin}(\boldsymbol{b}, P \cup E_{\mathfrak{m}}) + \mathfrak{m}$。我们通过对掩码半线性集中 $\mathfrak{m}$ 取并集来构造 $\mathcal{B}$。

我们断言 $\mathfrak{J}_{\exists\boxplus}(\mathcal{B}) = \mathfrak{J}(\mathcal{A})$。对于“简单”的方向，设 $w \in \mathfrak{J}_{\exists\boxplus}(\mathcal{B})$，则特别有 $\Psi(w) \in \Psi(\mathfrak{L}(\mathcal{N})) = \mathrm{Lin}(\boldsymbol{b}, P \cup E_{\mathfrak{m}}) \subseteq S = \Psi(\mathfrak{J}(\mathcal{A}))$（对于并集中的某个集合），因此 $w \in \mathfrak{J}(\mathcal{A})$（因为 $\mathfrak{J}(\mathcal{A})$ 是置换不变的）。

对于“困难”的方向，假设 $w \in \mathfrak{J}(\mathcal{A})$，则对并集中的某个集合，有 $\Psi(w) \in \mathrm{Lin}(\boldsymbol{b}, P \cup E_{\mathfrak{m}}) + \mathfrak{m}$。我们构造一个词 $w' \sim_{k\boxplus} w$，其对应某个 $k \in \mathbb{N}$，使得 $w' \in \mathfrak{L}(\mathcal{B})$，从而证明 $w \in \mathfrak{J}_{\exists\boxplus}(\mathcal{B})$。令 $k \in \mathbb{N}$ 足够大，使得$w = u \cdot z$ 满足 $z \in \mathfrak{m}|_\infty^\omega$ 和 $|u| \leq k$。这样的 k 确实存在，因为 w 在 $\mathfrak{m}|_\infty$之外只包含有限个字母。回顾一下，我们假设这些集合是$\mathfrak{m}$无关的，因此 $\Psi(u) \in \mathrm{Lin}(\boldsymbol{b}, P \cup E_{\mathfrak{m}})$。此外，$\Psi(z) = \mathfrak{m}$，因为$z \in L_{\mathfrak{m}}$。特别地，我们可以找到 u 的一个排列 u'，使得 $u' \in \mathfrak{L}(\mathcal{N})$。记作 $w' = u'z$，则 w' 满足上述条件，因此 $w \in \mathfrak{J}_{\exists\boxplus}(\mathcal{B})$，证毕。□

接下来，我们证明跳跃语言的表达能力严格弱于 $\exists\boxplus$语言。

例*2.* 考虑图3中的自动机 $\mathcal{A}$。我们声称，$\mathfrak{J}_{\exists\boxplus}(\mathcal{A})$ 不是任何自动机的跳跃语言。事实上，只需证明 $\mathfrak{J}_{\exists\boxplus}(\mathcal{A})$ 不是置换不变的即可。显然，$(ab)^\omega \in \mathfrak{J}_{\exists\boxplus}(\mathcal{A})$。然而，注意到 $(ab)^\omega \sim (a^n b^n)_{n=1}^\infty$（即词 $aba^2b^2a^3b^3\cdots$），因为它们的 Parikh 像均为 (∞, ∞)。但是，后者不属于 $\mathfrak{J}_{\exists\boxplus}(\mathcal{A})$，因为对 $(a^n b^n)_{n=1}^\infty$ 作任意 k-窗口排列，最终都会到达形如 a^k 的窗口（在足够长的序列 a^n 中），因此对它进行任意排列都会导致 $\mathcal{A}$ 拒绝。□

最后，我们证明 ω-正则语言与 $\exists\boxplus$lan- guages 不可比较：

例*3.* 回想图4中的自动机 $\mathcal{A}$，其中 $\mathfrak{L}(\mathcal{A}) = (ab)^* \cdot c^\omega$。注意，$\mathfrak{J}_{\exists\boxplus}(\mathcal{A}) = \mathfrak{J}(\mathcal{A}) = \{u \cdot c^\omega \mid u \in \{a,b,c\}^* \wedge \#_a[u] = \#_b[u]\}$。实际上，若$w = u \cdot c^\omega$ 如所述，那么通过重新排列数量相等的a和 b，可得 $w \in \mathfrak{J}_{|u|\boxplus}(\mathcal{A})$，即 $(ab)^* \cdot c^\omega$。反过来，若 $w \in \mathfrak{J}_{\exists\boxplus}(\mathcal{A})$，则特别有 $w \in \mathfrak{J}(\mathcal{A})$

（事实上，每个 $\exists\boxplus$-排列也是一个排列）。因此， $\mathfrak{J}_{\exists\boxplus}(\mathcal{A})$ 不是 ω-正则的。

对于逆向结论（即一个 ω-正则语言不是 $\exists\boxplus$），我们证明每个 $\mathfrak{J}_{\exists\boxplus}(\mathcal{A})$ 对任意 k 都具有 k-窗口不变性；而例如， $\{(ab)^\omega\}$ 不具有 2-窗口不变性，因此它不能被识别为一种 $\exists\boxplus$ 语言。

事实上，考虑 $w \in \mathfrak{J}_{\exists\boxplus}(\mathcal{A})$，并设 $w_1 \sim_{k\boxplus} w$ 对某个 k 成立。那么存在 $w_2 \sim_{k\boxplus} w$ ，使得 $w_2 \in \mathfrak{L}(\mathcal{A})$ 且 $k' \in \mathbb{N}$。但此时 $w_2 \sim_{k\boxplus} w$ 且 $w_1 \sim_{k\boxplus} w$，因此根据传递性， $w_1 \sim_{k\boxplus} w_2$，从而 $w_2 \in \mathfrak{J}_{\exists\boxplus}(\mathcal{A})$。因此， $\mathfrak{J}_{\exists\boxplus}(\mathcal{A})$ 具有 k-窗口不变性。□

注 *4*（替代语义）。请注意，除了 $\exists\boxplus$ 语义外，我们还可以要求存在将 w 划分为大小至多 k的窗口的划分，其中某个 $k \in \mathbb{N}$ 满足该条件。我们的所有证明在此定义下同样成立。

6 未来研究

我们为无限词上的跳跃自动机引入了三种语义。对于现有定义， $\exists\boxplus$ 语义在封闭性和判定问题方面仍有待探索。

从更广泛的角度来看，研究跳跃的定量语义将很有意思：与其粗略地区分有限次出现和无限次出现的字母，不如设想一种能够考虑字母出现频率等因素的语义。或者，也可以回到“跳跃式读头”的视角，对自动机移动读头所采用的策略施加限制（例如，仅限于由单计数器机器实现的策略）。

参考文献

1. Abu Nassar, A., Almagor, S.：逐字母转换器按轮次进行的模拟。载于：第30届 EACSL 计算机科学逻辑年度会议（2022） 2. Almagor, S.：概率转换器中的进程对称性。载于：第40届 IARCS 软件技术与理论计算机科学基础年度会议（2020） 3. Beier, S., Holzer, M., Kutrib, M.：关于半线性集上运算的描述复杂度。EPTCS 252，第41页（2017） 4. Büchi, J.R.：判定问题专题讨论会：受限二阶算术中的一种判定方法。载于：逻辑与数学基础研究，第44卷，第1–11页。爱思唯尔（1966） 5. Cadilhac, M., Finkel, A., McKenzie, P.：仿射 Parikh 自动机。RAIRO-理论信息学与应用 **46**(4), 511–545（2012） 6. Cadilhac, M., Finkel, A., McKenzie, P.：有界 Parikh 自动机。国际计算机科学基础杂志 **23**(08)，1691–1709（2012） 7. Chistikov, D., Haase, C.：驯服半线性集合。载于：第43届自动机、语言与编程国际研讨会（ICALP 2016）。达格施图尔城堡-莱布尼茨信息学中心（2016）

8. Emerson, E.A., Lei, C.L.：模型检测模态（扩展摘要）：分支时序卷土重来。载于：第12届 ACM SIGACT-SIGPLAN 编程语言原理研讨会论文集。第84–96页（1985） 9. Fazekas, S.Z., Hoshi, K., Yamamura, A.：具有跳跃模式的双向确定性自动机。理论计算机科学 **864**，92–102（2021） 10. Fernau, H., Paramasivan, M., Schmid, M.L.：跳跃有限自动机：刻画与复杂度。载于：自动机实现与应用国际会议。第89–101页。Springer（2015） 11. Fernau, H., Paramasivan, M., Schmid, M.L., Vorel, V.：关于跳跃有限自动机的刻画与复杂度结果。理论计算机科学 **679**，31–52（2017） 12. Guha, S., Jecker, I., Lehtinen, K., Zimmermann, M.：无限词上的 Parikh 自动机。载于：第42届 IARCS 软件技术与理论计算机科学基础年度会议（2022） 13. Klaedtke, F., Rueß, H.：带基数的单子二阶逻辑。载于：自动机、语言与编程：第30届国际研讨会，ICALP 2003，埃因霍温，荷兰，2003年6月30日至7月4日，会议论文集30。第681–696页。Springer（2003） 14. Kopczynski, E.：交换文法问题的复杂度。计算机科学逻辑方法 **11** （2015） 15. Kopczynski, E., To, A.W.：文法的 Parikh像：复杂度与应用。载于：2010年第25届 IEEE 计算机科学逻辑年度研讨会。第80–89页。IEEE（2010） 16. Kupferman, O.：自动机理论与模型检测。模型检测手册，第107–151页（2018） 17. Landweber, L.H.：w-自动机的判定问题。技术报告，威斯康星大学麦迪逊分校计算机科学系（1968） 18. Lavado, G.J., Pighizzini, G., Seki, S.：Parikh 等价下的操作状态复杂度。载于：形式系统描述复杂度：第16届国际研讨会，DCFS 2014，芬兰图尔库，2014年8月5日至8日，会议论文集16。第294–305页。Springer（2014） 19. McNaughton, R.：利用有限自动机测试和生成无限序列。信息与控制 **9**(5)，521–530（1966） 20. Meduna, A., Zemek, P.：跳跃有限自动机。国际计算机科学基础杂志 **23**(07)，1555–1578（2012） 21. Parikh, R.J.：关于上下文无关语言。ACM 会刊（JACM）**13**(4)，570–581（1966） 22. Safra, S.：关于 ω-自动机的复杂度。载于：第29届 IEEE 计算机科学基础研讨会论文集。第319–327页（1988） 23. To, A.W.：正则语言的 Parikh 像：复杂度与应用。arXiv 预印本 arXiv:1002.1464（2010） 24. Vorel, V.：跳跃有限自动机的基本性质。国际计算机科学基础杂志 **29**(01)，1–15（2018）

A 证明

A.1 ω-正则语言的并集

以下结论是一个广为人知的简单练习，我们将其列出以求完整。

命题。考虑一个自动机 $\mathcal{A} = \langle Q, \Sigma, \delta, q_0, \alpha \rangle$，那么我们可以写成 $\mathfrak{L}(\mathcal{A}) = \bigcup_{i=1}^{k} S_i \cdot T_i^{\omega}$，其中 S_i 和 T_i 是正则语言。

证明。对于状态 $p, q \in Q$，记 $\mathcal{A}_p^q$ 为以 p 为初始状态、以 $\{q\}$为接受状态的自动机 $\mathcal{A}$。注意，$\mathcal{A}$ 接受一个词，当且仅当 $\mathcal{A}$ 存在一条从 q_0开始、并且无限次访问某个状态 $q \in \alpha$ 的运行（实际上，由于 Q 是有限的，如果无限次访问 α，那么 α中的某个状态也会被无限次访问）。

因此，我们有 $\mathfrak{L}(\mathcal{A}) = \bigcup_{q \in \alpha} \mathfrak{L}_{\text{fin}}(\mathcal{A}_{q_0}^q) \cdot \mathfrak{L}_{\text{fin}}(\mathcal{A}_q^q)^{\omega}$。 □

A.2 定理 8 的证明

证明。直观地说，我们构造 $\mathcal{B}_k$，使其在某个状态中存储 k 个字母的 Parikh 像，然后非确定性地模拟 $\mathcal{A}$ 在所有具有所存储 Parikh 像的词上的运行，同时记录是否可能经过了某个接受状态。我们以图 5 中的 $k = 2$ 为例说明该构造。

现在我们将这一直观认识形式化。设 $\mathcal{A} = \langle Q, \Sigma, \delta, q_0, \alpha \rangle$。我们按如下方式构造 $\mathcal{B}_k = \langle Q', \Sigma, \delta', q_0', \alpha' \rangle$。– 令 $Q_\alpha = \{q_\alpha \mid q \in Q\}$ 为 Q 的一个副本，并令 $P = \{\Psi(x) | x \in \Sigma^{<k}\} \subseteq \{0, \ldots, k-1\}^{|\Sigma|}$ 表示所有长度为 $< k$ 的词的 Parikh 像集合（注意，P 是有限的），那么 $\mathcal{B}_k$ 的状态为 $Q' = (Q \times P) \cup Q_\alpha$。– 初始状态为 $q_0' = (q_0, \mathbf{0})$（注意，$\Psi(\epsilon) = \mathbf{0}$）。– $\alpha' = Q_\alpha$。– 转移函数定义如下。对于状态 $(q, \boldsymbol{v})$ 和字母 $\sigma \in \Sigma$，令 $\boldsymbol{u} = \boldsymbol{v} + \Psi(\sigma)$。如果 $\sum_{\sigma \in \Sigma} \boldsymbol{u}(\sigma) < k$，则定义 $\delta((q, \boldsymbol{v}), \sigma) = (q, \boldsymbol{u})$。如果 $\sum_{\sigma \in \Sigma} \boldsymbol{u}(\sigma) = k$，则将 $\delta^*(q, \boldsymbol{u})$ 定义为从 q 读取一个词 x 后可到达的状态集合，其中 $\Psi(x) = \boldsymbol{u}$；并将 $\delta_\alpha^*(q, \boldsymbol{u})$ 定义为从 q 读取一个词 x 后可到达的状态集合，其中 $\Psi(x) = \boldsymbol{u}$，且该运行会访问 α。于是我们有

$$\delta'((q, \boldsymbol{v}), \sigma) = \{(p, \mathbf{0}) \mid p \in \delta^*(q, \boldsymbol{v})\} \cup \{p_\alpha \mid p \in \delta_\alpha^*(q, \boldsymbol{v})\}$$

最后，对于每个 $q \in Q_\alpha$，我们定义 $\delta'(q, \sigma) = \delta'((q, \mathbf{0}), \sigma)$（也就是说，$Q_\alpha$ 副本的行为与 Q 的$\mathbf{0}$ 副本完全相同）。

现在我们断言 $\mathfrak{J}_{k\boxplus}(\mathcal{A}) = \mathfrak{L}(\mathcal{B}_k)$：令 $w \in \mathfrak{J}_{k\boxplus}(\mathcal{A})$，并记 $w = x_1 x_2 x_3 \ldots$，其中对所有 i，都有 $|x_i| = k$。那么，存在 $w' = y_1 y_2 y_3 \ldots \sim_{k\boxplus} w$（且所有 $|y_i| = k$），使得 $\mathcal{A}$在 w' 上有一个接受运行 ρ。用 q_0、q_1、.....表示对 $i \geq 1$ 而言满足 $q_i = \rho_{ik}$ 的状态。也就是说，这些状态是 ρ 在读入每个 y_i 后到达的状态。注意，对所有 i，都有 $q_{k(i+1)} \in \delta^*(q_{ik}, \psi(y_{i+1})) = \delta^*(q_{ik}, \psi(x_{i+1}))$。因此，存在 $\mathcal{B}$ 的一个运行：读入 x_{i+1} 后，若 ρ 在 y_{i+1} 上没有经过接受状态，则到达状态 $(q_{(i+1)k}, \mathbf{0})$；若经过了接受状态，则到达状态 $(q_{(i+1)k})_\alpha$。由于 ρ 会经过无限多个接受状态，$\mathcal{B}_k$ 在 w 上的运行也会如此，因此 $w \in \mathfrak{L}(\mathcal{B}_k)$。

反过来，假设 $w \in \mathfrak{L}(\mathcal{B}_k)$，并再次将 $w = x_1 x_2 x_3 \ldots$ 写作 $|x_i| = k$。于是，$\mathcal{B}_k$ 在 w上存在一条接受运行。将该运行按连续的k-窗口划分，我们可以构造词 y_1、y_2，.....，使得 $y_i \sim x_i$ 对所有 i 均成立，并且 $\mathcal{A}$ 接受 $w' = y_1 \cdot y_2 \cdots$。事实上，在每个 k-窗口中，$\mathcal{B}$ 都会模拟 $\mathcal{A}$ 在某个已读词的排列上的运行。由此可得 $w \in \mathfrak{J}_{k\boxplus}(\mathcal{A})$。□