arXiv:2602.17309v1 [cs.FL] 19 Feb 2026

关于边缘码语言的一些评述

Stavros Konstantinidis[0000−0002−6628−067X]

数学与计算机科学系，圣玛丽大学，哈利法克斯，新斯科舍省，加拿大，
s.konstantinidis@smu.cahttps://cs.smu.ca/~stavros/

摘要。 前缀码 L 满足这样一个条件：L 中没有任何一个字是 L 中另一个字的真前缀。最近，Ko、Han 和 Salomaa 放宽了这一条件，允许 L 中的一个字作为 L 中至多 k 个字的真前缀，其中 k 是某个“边缘”值，由此引入了 k-前缀无关语言类，以及类似的 k-后缀无交和 k-因子无交语言类。本文通过两种方式将这三个语言类的定义统一为一个统一的定义：一种基于偏序方法，另一种基于转导器方法。因此，对于任何已知的、可通过转导器方法定义的码相关语言类，都可以得到该类的边缘版本。在 Ko、Han 和 Salomaa 所用技术的基础上，我们讨论边缘语言类的统一满足性问题和极大性问题。

关键词： 独立语言 · 码 · 极大性 · 可判定性 · 自动机 · 形式语言

1 引言

中定义的前缀无关和有限前缀无关语言的概念 [8]自然地拓展了前缀码的概念；类似地，也可以定义后缀码和内因子码的相应概念， [8]。这三个概念可以统一为一个统一的概念，即 k-α 以及有限α 语言，它自然地拓展了任何现有的独立 α 语言——无论是通过偏序方法 [17], 还是通过转导器方法 [2,11]定义的。前缀码和后缀码这类码的统称是独立语言。沿用 [8] 中对上述参数所用的“边缘”一词 k，本文采用统称边缘独立语言来指 k-α 以及有限α 形式的现有独立语言。基于 [8], 中的技术，我们讨论边缘语言类的一致满足问题和极大性问题。

组织结构与主要结果。 本文结构如下。下一节介绍自动机、形式语言和码的一些基本概念。在第3节中，我们回顾定义独立语言类的两种方法——这些类称为独立性质:一偏序方法，以及范围更广的 Jürgensen 独立性方法。我们将讨论如何把偏序方法扩展为一种用于“边缘”偏序的方法，

其思路见 [8]。在定理 1 中，我们给出如下在数学上颇具意义的结果：任何满足某种可通过右无限偏序定义的有限α 独立性质的语言，都无法嵌入某个极大语言，因此这些有限α 独立性质不属于 Jürgensen 独立性方法的范畴。在第4节中，我们回顾满足性问题，即判定给定的正则语言是否满足 [8] 中三种边缘性质之一的问题。在第5节中，我们回顾用转换器定义独立性质的“形式化”方法，其中该性质可通过表示转换器的有限表达式来描述。这引出了满足性问题和极大性判定问题的统一版本，其中独立性质是输入的一部分。在定理 4 和 5 中，我们展示如何判定边缘性质的统一满足性问题。在第 6节中，我们以若干总结性意见和未来研究方向作结。

2 基本概念与符号

我们假设读者熟悉形式语言的基础知识，参见例如 [4,13,14,16]。一些记号： λ 表示空字； Σ、Γ 表示任意字母表； L 表示语言 L 的补集。设 L 是一种语言， u、v、w、x 是任意字。若 $w \in L$，则称 w 为一个 L-字。我们使用字和语言的标准运算与记号；特别地， $|w|$表示 w 的长度。若 w 的形式为 uv，则称 u 为 w 的前缀，称 v 为 w 的后缀。若 w 的形式为 uxv，则称 x 为 w 的内因子，称 uv 为 w 的外因子。若前缀 u 满足 $u \neq w$，则称其为 w 的真前缀。真后缀、真内因子和真外因子的概念与此类似。

我们将标准意义上的非确定有限自动机（NFA）简称为（有限）自动机，并用 DFA 指确定有限自动机。我们假设读者熟悉转导器的基础知识，例如参见 [1,15,19]。一个（有限）转导器是一个六元组 $\boldsymbol{t} = (Q, \Sigma, \Gamma, E, I, F)$，其中 Q 是状态集， $I \subseteq Q$ 是非空的起始（初始）状态集， $F \subseteq Q$是终态集， Σ, Γ 分别是输入字母表和输出字母表，而 E 是有限的转移（边）集。本文中，所有转导器的输入字母表和输出字母表都相同： $\Sigma = \Gamma$。我们假设所有转导器都采用标准形式：对于每个转移 $(p, x/y, q) \in E$，其输入标签 x 要么是空字 λ，要么是 Σ中的一个符号；输出标签 y也同样如此。我们将转导器 $\boldsymbol{t}$视为一个标记有向图，因此可以讨论其中的路径。路径的标签是字对 (u, v)，其中 u （分别是 v）为该路径上各转移的输入（分别为输出）标签的连接。若一条路径的第一个状态是初始状态，且最后一个状态是终态，则该路径为接受路径。 $\boldsymbol{t}$ 所实现的关系 $R(\boldsymbol{t})$ 是 $\boldsymbol{t}$的接受路径上标签 (u, v) 所构成的集合。 $\boldsymbol{t}$ 在输入 u 上的输出集合记为 $\boldsymbol{t}(u)$，也就是说， $v \in \boldsymbol{t}(u)$ 当且仅当 $(u, v) \in R(\boldsymbol{t})$。 $\boldsymbol{t}$ 的逆 $\boldsymbol{t}^{-1}$ 实现关系 $R(\boldsymbol{t})$ 的逆，即 $R(\boldsymbol{t}^{-1}) = \{(v, u) \mid (u, v) \in R(\boldsymbol{t})\}$。如果 $\boldsymbol{s}$ 是自动机或转导器，那么$|\boldsymbol{s}|$ 表示 $\boldsymbol{s} =$ 的大小，即 $\boldsymbol{s}$的状态数与转移数之和。

我们用 $\mathbb{N}$、$\mathbb{N}_0$ 分别表示正整数集和非负整数集。若 S 是一个集合，则 $|S|$ 表示 S的基数。对于集合 S，记号 $[S]^n$ 用于表示 S的 n重笛卡尔积 $S \times \cdots \times S$。使用这种记号是为了避免与语言 L的 n次幂 L^n 混淆。

语言性质就是语言的一个类（集合）。独立性质是指任何满足该性质的语言 $\mathbb{P}$ 都包含在一个关于 $\mathbb{P}$ 极大的语言 $M \in \mathbb{P}$ 中。所谓关于 $\mathbb{P}$ 的极大语言（或相对于 $\mathbb{P}$ 的极大语言），是指满足以下条件的语言 M：对所有字 $w \notin M$，$M \cup \{w\} \notin \mathbb{P}$。文献中关于码的各种概念，除极少数例外，都可以用独立性的概念来描述。当 $L \in \mathbb{P}$ 时，我们说 L 满足性质 $\mathbb{P}$，或者说 L 是相对于 $\mathbb{P}$ 的独立语言。通常，独立性质有一个名称 φ，并记作 $\mathbb{P}_\varphi$；在这种情况下，满足 $\mathbb{P}_\varphi$ 的语言 L 也称为 φ-独立语言。

如前所述 [6], 独立语言的典型例子是码理论所研究的那些语言。每一类码都由码（语言）中字所满足的某个条件来定义。

例 *1.* 我们回顾几类众所周知的码。
– 如果没有任何一个 L字是另一个 L字的前缀，那么语言 L 就是前缀码。– 如果没有任何一个 L字是另一个 L字的后缀，那么语言 L 就是后缀码。– 如果没有任何一个 L字是另一个 L字的内因子，那么语言 L 就是内因子码。– 如果没有任何一个 L字是另一个 L字的外因子，那么语言 L 就是外因子码。– 如果不存在这样的 L 字：它是从另一个 L 字中删除若干符号后得到的，那么语言 L 就是超码。– 如果每个字都至多能以一种方式写成 L字的串接，那么语言 L 就是唯一可译／可解码，简称唯一可译码（UD-code）。

缩写： iff =“当且仅当”，w.r.t. = “关于”

3 Jürgensen 独立性与边缘码语言

独立语言研究中的一个重要方向，是探索系统的方法，以便定义和研究这些语言的不同码类。接下来我们将回顾其中的一些方法。

通过偏序定义独立语言。据我们所知，定义码性质的第一种方法是通过字上的某些二元关系来定义的方法。二元关系 ω 是字对的集合，即 $\omega \subseteq \Sigma^* \times \Sigma^*$。在 [17], 中，作者介绍了通过作为偏序（按标准定义）的二元关系来定义的方法。设 $\leq$ 是 Σ^* 上的一个偏序。如果一个语言 L 满足以下条件，则称其为 $\leq$-独立语言：

$$u, v \in L \text{ and } u \leq v \text{ imply } u = v. \tag{1}$$

同一类语言也可通过‘$\leq$’的严格版本‘$<$’来定义：

$$< \cap [L]^2 = \emptyset. \tag{2}$$

上文中， $<_= \leq -\{(u,u) \mid u \in \Sigma^*\}$，这是‘$\leq$’的非自反且非对称版本。前缀码和后缀码的经典示例分别通过偏序 $\leq_{\text{px}}, \leq_{\text{sx}} \subseteq [\Sigma^*]^2$ 定义：

$$\leq_{\text{px}} = \{(w,z) \mid z \text{ is a prefix of } w\} \quad \text{and} \quad \leq_{\text{sx}} = \{(w,z) \mid z \text{ is a suffix of } w\};$$

或通过这些序关系的严格形式

$<_{\text{px}} = \{(w,\ z) \mid z$ 是 $w\}$ 的真前缀，且
$<_{\text{sx}} = \{(w,\ z) \mid z$ 是 $w\}$.的真后缀

因此， L 是前缀码，当且仅当没有两个不同的字 $u,\ v \in L$ 通过 $\leq_{\text{px}}$相关；等价地，当且仅当没有两个字 $u,\ v \in L$ 通过 $<_{\text{px}}$相关。

Jürgensen 独立性。设 n是正整数或 $\aleph_0$。一个 n-独立性质是一个语言类 $\mathbb{P}$，满足

$$L \in \mathbb{P} \quad \text{iff} \quad \text{for all } L' \subseteq L \text{ with } |L'| < n\text{: } L' \in \mathbb{P}. \tag{3}$$

Jürgensen 性质就是某个 $n \in \mathbb{N} \cup \{\aleph_0\}$的 n-独立性质。因此，要判断语言 L 是否关于某个 $\mathbb{P}$独立，只需（也必须）确保 L 的每个含有 $< n$ 个元素的子集都关于 $\mathbb{P}$独立。例如，前缀码类是一种 3-独立性质，因为语言 L 是前缀码，当且仅当 L 的每个至多含有 2 个字的子集都是前缀码。为什么 Jürgensen 定义值得关注？原因之一是，它可以用来刻画以往所有方法定义的独立性质。另一个原因是，Jürgensen 方法保证每个独立语言都包含在某个极大语言 [6]中。第三个原因是，唯一可译码类无法用以往的方法刻画，因为它是一种 $\aleph_0$-独立性（语言 L 是唯一可译码，当且仅当 L 的每个有限子集都是唯一可译码），但唯一可译码类并非任何$n < \aleph_0$、 [6]下的 n -独立性质。

3.1 边缘性质。

由 $\leq$-独立性质（2）要求任意一对 L-字之间不存在通过序关系‘$<$’关联的情况。例如，当 L 是前缀码时，没有任何 L-字是某个 L-字的真前缀。在 [8], 中，Ko、Han 和 Salomaa 放宽了这一条件，允许一个 L-字成为至多 k 个字的真前缀，其中‘误差’余量为 $k \in \mathbb{N}_0$。他们定义了 k-前缀无歧义语言这一类，在本文中记作 $\mathbb{P}_{k\text{-px}}$，定义如下

$$L \in \mathbb{P}_{k\text{-px}}, \text{ 当且仅当 } L \text{ 中的每个字都是 } L\text{中至多 } k \text{ 个字的真前缀。} \tag{4}$$

由此可知， $\mathbb{P}_{0\text{-px}}$ 恰好是前缀码这一类。有限前缀无关语言的类 $\mathbb{P}_{\text{fin-px}}$ 定义如下

$$L \in \mathbb{P}_{\text{fin-px}}, \text{ 如果 } \forall w \in L, w \text{ 是至多有限个 } L\text{-字的真前缀。} \tag{5}$$

与作为唯一可译码的前缀码（0-前缀语言）的情况不同，值得注意的是，对于 $k > 0$，这些语言不一定是唯一可译码。例如，$\{0, 00\}$ 是一个 1-前缀无歧义语言，但它不是唯一可译码。

类 $\mathbb{P}_{k\text{-sx}}$ 和 $\mathbb{P}_{k\text{-ix}}$ 的 k-后缀无歧义语言 和 k-无子串语言的定义类似，此外还有类 $\mathbb{P}_{\text{fin-sx}}$ 和 $\mathbb{P}_{\text{fin-ix}}$，[8]。

我们通过偏序机制扩展上述概念。

定义1。设 $\leq_\alpha$ 是 Σ^* 上的一个偏序，其中 α 用作该偏序的名称。类 $\mathbb{P}_{k\text{-}\alpha}$ 的 k-α 语言定义如下：

$L \in \mathbb{P}_{k\text{-}\alpha}$，如果 L 中的每个字相对于“$<_\alpha$”至多比 L 中的 k 个字小。

有限-α 语言的类 $\mathbb{P}_{\text{fin-}\alpha}$ 定义同上，只需将条件“ L 中的每个字相对于‘$<_\alpha$’至多比 L 中有限多个字小”作为定义条件。

因此，我们可以基于适当的偏序来讨论 k-超码语言类和 k-外因子无关语言类。注意，对于每个偏序 $\leq_\alpha$ 和每个 $k \in \mathbb{N}_0$，每个单元素语言$\{w\}$ 都是一个 k-α 语言。尽管对于所有 $k \in \mathbb{N}_0$，类 $\mathbb{P}_{k\text{-}\alpha}$ 是独立性质，但类 $\mathbb{P}_{\text{fin-}\alpha}$ 则并非如此。下一个定理将说明这一点，其中术语“右无限序” $\leq_\alpha$ 表示，对于每个字 u，都存在一个字 v，使得 $u <_\alpha v$。

定理 1. 设 $\leq_\alpha$ 是字上的任意偏序，且设 $k \in \mathbb{N}_0$。以下陈述成立。

1. 类 $\mathbb{P}_{k\text{-}\alpha}$ 是一个 *(k+3)*-独立性质；因此，每个 k-α 语言都包含在某个极大语言中。*2.* 如果 $\leq_\alpha$ 是右无限序，那么对于每个有限-α 语言 L，都存在一个字 $z \notin L$，使得 $L \cup \{z\}$ 是有限-α 语言。因此，类$\mathbb{P}_{\text{fin-}\alpha}$ 不是 *Jürgensen* 性质。

证明。第一个陈述可根据 Jürgensen 性质和 k-α 语言的定义，运用标准逻辑推理得出。对于第二个陈述，考虑任意一个有限-α 语言 L。我们证明存在一个字 $z \notin L$，使得 $L \cup \{z\}$ 是一个有限-α 语言。如果 L 为空，那么 z 可以是任意一个字：$L \cup \{z\} = \{z\}$，它是有限-α的。如果 L 非空，则考虑任意一个字 $u \in L$。对于任意字 x，我们用记号 $V(x) = \{v \mid x <_\alpha v\}$。由于 L 是有限-α的，集合 $L \cap V(u)$ 是有限的；因此它具有如下形式

$$L \cap V(u) = \{v_1, \ldots, v_n\}.$$

考虑任意一个在 $\{v_1, \ldots, v_n\}$ 中极大的元素 v_r；也就是说，对于所有 $j = 1, \ldots, n$，都有 $v_r \not<_\alpha v_j$。由于 $\leq_\alpha$ 是右无穷的，存在一个字 z，使得 $v_r <_\alpha z$。此外，我们有 $u <_\alpha v_r <_\alpha z$和 $z \notin \{v_1, \ldots, v_n\}$。接下来我们证明，$z \notin L$ 和 $L \cup \{z\}$ 是一个有限-α 语言。

作为 $z \in V(u)$ 和 $z \notin L \cap V(u)$，我们有 $z \notin L$。

设 w 是 $L \cup \{z\}$ 中的任意一个字。我们需要证明，就“<”而言， w 至多比 $L \cup \{z\}$ 中有限多个字小；也就是说，集合 $\big(L \cup \{z\}\big) \cap V(w)$ 是有限的。首先注意到

$$\big(L \cup \{z\}\big) \cap V(w) \;=\; \big(L \cap V(w)\big) \cup \big(\{z\} \cap V(w)\big).$$

如果 $w \neq z$ 那么 $w \in L$ 和 $L \cap V(w)$ 是有限的，因此 $\big(L \cup \{z\}\big) \cap V(w)$ 也是有限的。如果 $w = z$ 那么只需证明 $L \cap V(z)$ 为空。如果存在 $z_1 \in L \cap V(z)$，那么 $u <_\alpha z <_\alpha z_1$，这意味着 $z_1 \in V(u)$，并且 $z_1 \in \{v_1, \ldots, v_n\}$，这是不可能的。

例 1 中与各码类对应的所有偏序（唯一可译码类除外）都是右无限序。

值得注意的是，如果一种语言 L 是正则的且有限前缀无关，那么 L 对于某个 k 是无前缀的 $k \geq 0$；但一般而言，如果 L 是上下文无关的 [8]，则情况并非如此。

4 满足性的可判定性

对于一个固定的独立性质 $\mathbb{P}$，满足性问题是判定给定语言 L是否满足 $\mathbb{P}$；也就是说，是否 $L \in \mathbb{P}$。该语言通过一种定义良好的描述方法作为输入给出。通常，输入是描述正则语言的有限自动机。这在编码理论中是一个合理的假设，因为所关注的语言通常是有限的或正则的。当然，完全可以假设描述该语言的输入是上下文无关文法或下推自动机；但在这种情况下，满足性问题通常是不可判定的 [6]。

在第5节中，我们讨论统一版本的上述判定问题，其中除了语言 L的描述之外，所需算法还将独立性质 $\mathbb{P}$的描述作为输入。

边缘性质满足性的可判定性。这些性质来自 [8]，并在陈述（4）和（5）中定义，见第3.1节。如 [8], 所示， k-边缘性质难以判定——有限边缘性质见下文：

– 判定对于给定的自动机 $\boldsymbol{a}$ 和整数 $k \geq 0$，语言 $L(\boldsymbol{a})$ 是否为无前缀的，是 PSPACE完全的。– 判定对于给定的自动机 $\boldsymbol{a}$ 和整数 $k \geq 0$，语言 $L(\boldsymbol{a})$ 是否为无后缀的，是 PSPACE完全的。– 判定对于给定的自动机 $\boldsymbol{a}$ 和整数 $k \geq 0$，语言 $L(\boldsymbol{a})$ 是否为无子串的，是 PSPACE困难的。

在 [8] 中，这仍是一个悬而未决的问题：满足 k-无子串性质是否属于 PSPACE 类。随后， [5] 对此问题给出了肯定的回答——事实上，即使对于更大的机器类也成立，其中 $\boldsymbol{a}$ 是一台有限交叉双向机，并带有若干个反转有界计数器。

对于有限边际性质的情形——见陈述（5）中的定义——满足性问题可以高效判定：

– 假设字母表固定（但任意），则存在一个 $O(|\boldsymbol{a}|^2)$ 算法，用以判定给定自动机 $\boldsymbol{a}$ 是否接受有限前缀无关语言（同样适用于有限后缀无关、有限因子无关）。

我们将在第5.1节重新讨论边缘性质的满足性问题。

5 形式独立性质与边缘版本

这里我们考虑这样一种设想：将一个独立性质作为输入提供给某个算法，该算法会针对该性质提出的指定问题返回答案。为此，必须有一种句法方法，通过某种表达式 φ 来描述独立性质。我们将用 $\mathbb{P}_\varphi$ 表示由 φ 描述的独立性质（关于用于描述独立性质的句法方法选择，见下文进一步讨论）。我们将使用术语 φ-独立语言来指称属于 $\mathbb{P}_\varphi$ 的语言（满足 $\mathbb{P}_\varphi$ 的语言）。

我们关注一致满足问题：

输入：独立性质的描述 φ，以及语言 L 的描述

问题： L 是否满足 $\mathbb{P}_\varphi$。

我们还关注一致极大性问题：

输入：独立性质 φ 的描述，以及语言 L的描述。问题： L是否是满足 $\mathbb{P}_\varphi$的极大元素。

转导器的形式化方法。 方法 [2,11,12] 使用转导器描述有理二元关系（包括若干偏序），从而定义若干3-独立性质，如（1）中所示。令 $\boldsymbol{t}$ 为一个转导器。由 $\boldsymbol{t}$ 描述的性质 $\mathbb{P}_{\boldsymbol{t}}$ 定义如下

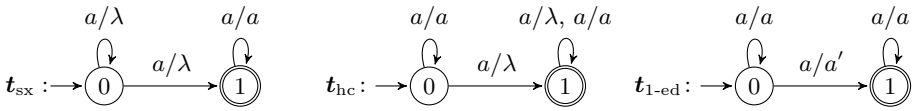
$$L \in \mathbb{P}_{\boldsymbol{t}}, \quad \text{if } v \in \boldsymbol{t}(u) \text{ implies } u = v, \text{ for all } u, v \in L. \tag{6}$$

如果转导器 $\boldsymbol{t}$ 是输入改变型的，即对所有字 w， $w \notin \boldsymbol{t}(w)$，那么我们可以对 $\mathbb{P}_{\boldsymbol{t}}$给出一个更简单的刻画：

$$L \in \mathbb{P}_{\boldsymbol{t}} \quad \text{iff} \quad \boldsymbol{t}(L) \cap L = \emptyset. \tag{7}$$

下图中的转导器 $\boldsymbol{t}_{\text{sx}}$ 是输入改变型的，它描述了后缀码这一类；也就是说， L 是后缀码，当且仅当 $\boldsymbol{t}_{\text{sx}}(L) \cap L = \emptyset$。转移标签 a/λ实际上表示多个转移，每个 $a \in \Sigma$对应一个转移，其他标签也类似。标签 a/a' 也表示多个转移，每个满足 $a \neq a'$的对a， $a' \in \Sigma$ 对应一个转移。下方的转导器 $\boldsymbol{t}_{\text{hc}}$ 是输入改变型的，它描述了超码这一类。转导器 $\boldsymbol{t}_{1\text{-ed}}$ 是输入改变型的，它描述了1-错误检测语言[1]

[1] 一种语言 L 具有 k错误检测能力，当且仅当任意两个长度相同且不同的 L-字之间的汉明距离为 $> k$。对于许多错误组合，包括 [9] 中的错误组合，都可以充分运用转导器方法来描述错误检测语言。



我们有 $\boldsymbol{t}_{\mathrm{sx}}(v) = $ 是 v 的真后缀集合。请注意，转导器方法实现了第 3 节中有理偏序的逆。例如，我们有 $\left(<_{\mathrm{sx}}\right)^{-1} = R(\boldsymbol{t}_{\mathrm{sx}})$。

在 [2], 文中证明，对于每个正则轨迹性质 $\mathbb{P}_\alpha$，都存在一个输入改变型转导器 $\boldsymbol{t}$，它描述相同的独立性质，即 $\mathbb{P}_\alpha = \mathbb{P}_{\boldsymbol{t}}$。此外，有些自然的错误检测性质无法用轨迹来描述。上述转导器方法只能描述 3-独立性质。当所涉及的转导器为输入改变型时，该方法将通过 n元关系实现独立性的方法形式化，其中这些关系对于 $n = 2$是有理的——参见公式（??）中的定义，并注意公式（7）等价于 $R(\boldsymbol{t}) \cap [L]^2 = \emptyset$。

使用转导器可以判定正则语言的一致满足问题和一致极大性问题：

定理 2（[2,7]）。假设字母表固定（但任意），则存在[2] 一个$O(|\boldsymbol{t}||\boldsymbol{a}|^2)$时间算法，用于判定给定的转导器 $\boldsymbol{t}$ 和自动机 $\boldsymbol{a}$后，$L(\boldsymbol{a}) \in \mathbb{P}_{\boldsymbol{t}}$是否成立。

定理3（[2]）。给定转导器[3] $\boldsymbol{t}$ 和自动机 $\boldsymbol{a}$，判定 L（$\boldsymbol{a}$）是否在 $\mathbb{P}_{\boldsymbol{t}}$中为极大是 *PSPACE* 完全的。

5.1 边缘独立性质的一种形式化方法。

定义1 以自然的方式扩展了 [8] 的边缘独立性质。然而，当我们要处理统一的满足性问题的可判定性时，需要允许将偏序的描述作为输入提供给所需的算法。我们使用转导器来描述许多偏序（事实上，任何有理二元关系）。现在，我们将（6）中的 $\mathbb{P}_{\boldsymbol{t}}$ 定义扩展为边缘性质的定义。注意，（6）中的条件等价于“$\boldsymbol{t}$（u）$\cap$（$L - \{u\}$）$= \emptyset$，对所有 $u \in L$均成立”。扩展后的定义如下：

$$L \in \mathbb{P}_{\boldsymbol{t},k}, \quad \text{if } |\boldsymbol{t}(u) \cap (L - \{u\})| \leq k, \text{ for all } u \in L. \tag{8}$$

.....的边缘性质 [8] 以及更多性质，都可以像上面的（8）一样，通过使用适当的转换器来定义 $\boldsymbol{t}$。引入边缘

[2] 算法的多项式时间复杂度记作 [2], ，而 O（$|\boldsymbol{t}||\boldsymbol{a}|^2$）时间复杂度由 [7, 定理 21]确立。

[3] 的定理9 [2] 指出该问题是 PSPACE 困难的，但很容易看出它实际上是 PSPACE 完全的。此外，[2] 的定理9假设给定的转导器 $\boldsymbol{t}$ 是输入保持型，但这一限制并非必要——当且仅当对所有字$\boldsymbol{t}$，若 $\boldsymbol{t}$（w）$\neq \emptyset$ 则 $w \in \boldsymbol{t}$（w）时，转导器$\boldsymbol{t}$ 是输入保持型。 w事实上，即使 $\boldsymbol{t}$ 是输入改变型，该问题仍然是 PSPACE 困难的。

性质的动机是放宽严格的空集条件 $\boldsymbol{t}(u) \cap (L - \{u\}) = \emptyset$，改为条件 $|\boldsymbol{t}(u) \cap (L - \{u\})| \leq k$，因此假设 k 很小，甚至是固定的，是有意义的。对于固定的 k，正则语言的转导器边缘性质的一致满足问题可在多项式时间内判定：

定理 4。设 k 为一个固定的非负整数。存在一个多项式时间算法，用于判定给定转导器 $\boldsymbol{t}$ 和自动机 $\boldsymbol{a}$时，$L(\boldsymbol{a}) \in \mathbb{P}_{\boldsymbol{t},\, k}$是否成立。

证明。如果 $k = 0$，那么 $\mathbb{P}_{\boldsymbol{t},\, k} = \mathbb{P}_{\boldsymbol{t}}$，且该陈述立即由定理 2得出。现在令 $k > 0$。我们使用转导器 $\uparrow$ 与 $\downarrow$ 自动机 $\boldsymbol{s}$ 之间的以下运算 $\boldsymbol{a}$，[10,11]:

– $\boldsymbol{s} \downarrow \boldsymbol{a}$ 是一个定义域为 $L(\boldsymbol{a})$ 的转导器，使得当且仅当 $v \in \boldsymbol{s}(u)$ 且 $u \in L(\boldsymbol{a})$ 时，$v \in (\boldsymbol{s} \downarrow \boldsymbol{a})(u)$ 成立。转导器 $\boldsymbol{s} \downarrow \boldsymbol{a}$ 可以在时间 $O(|\boldsymbol{t}||\boldsymbol{a}|)$ 内计算得到。– $\boldsymbol{s} \uparrow \boldsymbol{a}$ 是满足以下条件的转导器：$v \in (\boldsymbol{s} \uparrow \boldsymbol{a})(u)$ 当且仅当 $v \in \boldsymbol{s}(u)$ 且 $v \in L(\boldsymbol{a})$。转导器 $\boldsymbol{s} \uparrow \boldsymbol{a}$ 可以在时间 $O(|\boldsymbol{s}||\boldsymbol{a}|)$ 内计算得到。

注意，$v \in (\boldsymbol{s} \downarrow \boldsymbol{a} \uparrow \boldsymbol{a})(u)$当且仅当 $u, v \in L(\boldsymbol{a})$且 $v \in \boldsymbol{s}(u)$。我们还使用 [3]的如下结果：对于固定整数 $k > 0$，存在一个多项式时间算法，用以判定任意转导器是否为 k-值的；其中，若对所有字 w均有 $|\boldsymbol{s}(w)| \leq k$，则转导器 $\boldsymbol{s}$是 k 值的。我们希望 $L(\boldsymbol{a})$属于$\mathbb{P}_{\boldsymbol{t},k}$ 当且仅当转导器$(\boldsymbol{t} \downarrow \boldsymbol{a} \uparrow \boldsymbol{a})$是$(k+1)$-值的。然而，这并不正确。我们按如下方式加以修正：设 $\boldsymbol{t}^{=}$ 是由 $\boldsymbol{t}$ 得到的转导器，即添加一个既为初始又为终态的新状态 q，并且对所有字母表符号 $\sigma \in \Sigma$具有环路转移$(q, \sigma/\sigma, q)$。则对所有字 w，有 $\boldsymbol{t}^{=}(w) = \{w\} \cup \boldsymbol{t}(w)$。设$\boldsymbol{s} = (\boldsymbol{t}^{=} \downarrow \boldsymbol{a} \uparrow \boldsymbol{a})$。可以验证

– $\mathbb{P}_{\boldsymbol{t},\, k} = \mathbb{P}_{\boldsymbol{t}^{=},\, k}$。– $\boldsymbol{s}(u) = \boldsymbol{t}^{=}(u) \cap L(\boldsymbol{a})$，对所有 $u \in L(\boldsymbol{a})$ 均成立。– $L(\boldsymbol{a}) \in \mathbb{P}_{\boldsymbol{t},\, k}$ 当且仅当转导器 $\boldsymbol{s}$ 是 $(k+1)$-值的。

该陈述可由 [3]上述结果推出。

我们注意到，在 [5, 命题 35], 中，作者证明：对于固定的 p、r、c、k，存在一个多项式时间算法，可以判定给定的“在输入带上至多有 p个计数器、至多进行 r 次反转且至多发生 c 次穿越的双向机”是否接受一种 k-无子串语言（同样也适用于 k-无前缀和 k-无后缀）。这意味着，对于固定的 k，当给定机器为标准自动机时，该问题可在多项式时间内判定。在上述定理中，我们证明了，对于 $\mathbb{P}_{\boldsymbol{t},\, k}$中的任意边缘性质，该问题仍可在多项式时间内判定。

在上述定理中，如果 k 不固定，则该问题是可判定的，但不太可能能在多项式时间内判定——正如第 4节所述，对于固定性质 $\mathbb{P}_{\boldsymbol{t}_{\mathrm{sx}},\, k}$，该问题已经是 PSPACE完全的。

现在，我们按照 [8], 中对8的做法，扩展 k-边缘性质的有限边缘性质定义：

$$L \in \mathbb{P}_{\boldsymbol{t},\mathrm{fin}}, \quad \text{if } \boldsymbol{t}(u) \cap L \text{ is finite, for all } u \in L. \tag{9}$$

同样， $\boldsymbol{t}$ 可以是任意转换器（例如，若 $\boldsymbol{t} = \boldsymbol{t}_{\mathrm{sx}}$ 则$\mathbb{P}_{\boldsymbol{t}_{\mathrm{sx}},\ \mathrm{fin}}$ 就是有限后缀无歧义语言类）。如第4节所述， [8] 的作者证明，对于正则语言，三种边缘性质——有限前缀无关、有限后缀无歧义和有限无子串——的满足性问题可以在二次时间内判定。下文将证明，一致的满足性问题也可以在二次时间内判定，其中时间复杂度相对于给定自动机的大小而言。该证明以适用于所有转换器 [8] 的方式统一了 $\boldsymbol{t}$中各证明的思路。对于转换器$\boldsymbol{s}$，证明会用到以下术语：

– λ-输入路径： $\boldsymbol{s}$ 中的任意路径，其转移标签均具有 λ/b for$b \in \Sigma \cup \{\lambda\}$的形式。– 适当的 λ-输入环： $\boldsymbol{s}$ 中的任意环，其转移标签均具有λ/b的形式（对于 $b \in \Sigma \cup \{\lambda\}$），且至少有一个标签具有 $b \neq \lambda$。– $\boldsymbol{s}$的修剪部分：如果消除 $\boldsymbol{s}$中无法从初始状态到达且无法到达终态的状态，所得到的转导器。

定理5。 假设字母表固定但任意，则存在一个 $O(|\boldsymbol{t}||\boldsymbol{a}|^2)$时间算法，用于判定对于给定的转导器 $\boldsymbol{t}$ 和自动机 $\boldsymbol{a}$， $L(\boldsymbol{a}) \in \mathbb{P}_{\boldsymbol{t},\ \mathrm{fin}}$。

证明。 令 Σ 为字母表，并令 $\Sigma_\lambda = \Sigma \cup \{\lambda\}$。我们声称，陈述（9）中需要检验的条件等价于

$$\boldsymbol{s} \text{ has no proper } \lambda\text{-input cycle, where } \boldsymbol{s} = \text{ the trim part of } \boldsymbol{t} \downarrow \boldsymbol{a} \uparrow \boldsymbol{a}.$$

关于操作“↓”和“↑”，请参见定理4 的证明。关于上述断言的正确性，详见下文。注意到以下几点即可得到时间复杂度：(i) $\boldsymbol{s}$ 的规模为 $O(|\boldsymbol{t}||\boldsymbol{a}|^2)$，且可在 $O(|\boldsymbol{t}||\boldsymbol{a}|^2)$时间内构造；(ii) 可以使用广度优先搜索在线性时间内检验一个图是否包含环；(iii) 用于检验环的图，是将 $\boldsymbol{s}$ 视为自动机（将转移标签看作原子符号）后得到的转导器，并将其与任何一个接受语言 $(\{\lambda\} \times \Sigma_\lambda)^*(\{\lambda\} \times \Sigma)(\{\lambda\} \times \Sigma_\lambda)^*$的固定自动机求交所得。

现在，我们证明关于转导器 $\boldsymbol{s}$ 的断言是正确的。首先，假设 $\boldsymbol{s}$ 有一个适当的 λ-输入环，其标签为 $(\lambda,\ z) \neq (\lambda,\ \lambda)$，从某个状态 q 出发并终止于 q。由于 $\boldsymbol{s}$ 是修剪的，从某个初始状态到 q 存在一条标签为 $(x,\ y)$ 的路径，且从 q 到某个终态存在一条标签为 $(x',\ y')$ 的路径。于是， $\boldsymbol{s}(xx')$ 是无限的，因为它包含 yz^*y'。因此，对于 $u = xx'$，我们有 $u \in L(\boldsymbol{a})$ 和 $\boldsymbol{t}(u) \cap L(\boldsymbol{a}) = \boldsymbol{s}(u)$ 是无限的，从而 $L(\boldsymbol{a}) \notin \mathbb{P}_{\boldsymbol{t},\ \mathrm{fin}}$。

现在假设 $L(\boldsymbol{a}) \notin \mathbb{P}_{\boldsymbol{t},\ \mathrm{fin}}$；那么存在 $u \in L(\boldsymbol{a})$，使得 $\boldsymbol{s}(u)$是无限的。考虑集合 π_u ，其中包含 $\boldsymbol{s}$ 中所有满足以下条件的接受路径：不含任何 $(\lambda,\ \lambda)$- 环，且其标签具有 $(u,\ v)$ 的形式。由于 $\boldsymbol{s}(u)$ 是无限的，集合 π_u 也是无限的。由于所有 π_u 路径的输入部分 u 都相同，因此路径 π_u中包含任意长的 λ-输入子路径。所以，这些 λ-输入子路径中至少有一条含有重复状态，因此该子路径含有一个环，且此环不是 $(\lambda,\ \lambda)$-环；因此它是一个真正的 λ-输入环。

6 结论性备注

.....的概念 k-前缀无歧义、 k-后缀无歧义以及 k-中缀无歧义语言，以及 [8] 中的相关结果，提供了重要见解，由此可以建立一种形式化方法来表达边缘性质 $\mathbb{P}_{\boldsymbol{t},\ k}$，并研究相关的一致满足问题和极大性问题。具体而言，我们有以下研究方向。

– 用于判定语言 $L \in \mathbb{P}_{\boldsymbol{t}}$ 是否为极大的测试条件，其中 $\boldsymbol{t}$ 是转导器， [2], 是

$$L \cup \boldsymbol{t}(L) \cup \boldsymbol{t}^{-1}(L) = \Sigma^*.$$

是否存在类似的条件，用于测试语言 $\mathbb{P}_{\boldsymbol{t},\ k}$-极大性 L? – 第4 节中关于 k-无前缀性质满足性的 *PSPACE*完全性结果表明，对于给定的$\mathbb{P}_{\boldsymbol{t},\ k}$、 $\boldsymbol{t}$ 和 k，正则语言是否属于$\boldsymbol{t}$ 的统一满足性问题是 PSPACE困难的。统一满足性问题是否属于 PSPACE? 如果不是，对于哪些类型的转换器$\boldsymbol{t}$ 满足性问题属于 PSPACE——请再次参见4定理之后的讨论?

致谢。 本研究由加拿大自然科学与工程研究委员会（资助编号 RGPIN-2020-05996）的 Discovery Grant 资助。

利益披露。 作者声明不存在与本文内容相关的任何利益冲突。

参考文献

1. Berstel, J.: 《转换与上下文无关语言》。B.G. Teubner，斯图加特（1979） 2. Dudzinski, K.，Konstantinidis, S.: 代码性质的形式描述：可判定性、复杂性与实现。《国际计算机科学基础杂志》**23:1**，67–85（2012），这是在加拿大萨斯卡通举行的 DCFS 2010 会议上受邀报告的期刊版本。编辑：Ian McQuillan 和 Giovanni Pighizzini。3. Gurari, E.M.，Ibarra, O.H.: 关于有限值变换器和有限歧义变换器的注记。《数学系统理论》**16**（1），61–66（1983）。 https://doi.org/10.1007/BF01744569，01744569 4101007https://doi.org/./BF。 Hopcroft, J.E.，Ullman, J.D.: 《自动机理论、语言与计算导论》。Addison-Wesley（1979） 5. Ibarra, O.H.，McQuillan, I.: 关于某些机器类的判定问题的复杂性及其应用。《信息与计算》**294**，105080（2023）。 https://doi.org/https://doi.org/10.1016/j.ic.2023.105080， https://www.sciencedirect.com/science/article/pii/S0890540123000834 6. Jürgensen, H.，Konstantinidis, S.: 码。载于：Rozenberg 和 Salomaa [14], 第 511–607 页 7. Kari, L.，Konstantinidis, S.，Kopecki, S.: DNA 码性质的转导器描述与反形态问题的不可判定性。《信息与计算》**259**（2），237–258（2018）。 https://doi.org/10.1016/J.IC.2017.09.004， https://doi.org/10.1016/j.ic.2017.09.004

8. Ko, S., Han, Y., Salomaa, K.: 带有边缘错误的代码语言推广。《计算机科学基础国际期刊》**32**（5），509–529（2021）。https://doi.org/10.1142012905412141005790129054121410057，101142/Shttps://doi.org/./S。Konstantinidis, S.: 涉及三种基本错误类型组合的离散信道代数。《信息与计算》**167**（2），120–131（2001）。https://doi.org/103035 10100620013035，1010062001./INCO..https://doi.org/./inco..。Konstantinidis, S.: 转换器与错误检测、错误纠正和有限延迟可解码性的性质。《通用计算机科学期刊》**8**，278–291（2002） 11. Konstantinidis, S.: 转换器在独立语言、词距离和码中的应用。载于：Pighizzini, G., Câmpeanu, C.（编）：《形式系统描述复杂性——第 19 届 IFIP WG 1.02 国际会议，DCFS 2017，意大利米兰，2017 年 7 月 3–5 日，会议录》。《计算机科学讲义》，第 10316 卷，第 45–62 页。Springer（2017）。https://doi.org/10.1007/978 3 319 60252 3_4 10 1007 978 3 319 60252 3_4 ----，12 https://doi.org/./----。Konstantinidis, S., Meijer, C., Moreira, N., Reis, R.: 通过转换器实现码性质。载于：Han, Y.S., Salomaa, K.（编）：CIAA 2016 会议录。第 1–13 页。《计算机科学讲义》第 9705 卷（2016），arXiv 版本：《码性质的符号操作》。arXiv:1504 1, 2015 13 04715。v 。Mateescu, A., Salomaa, A.: 形式语言：导论与概要。载于：Rozenberg 和 Salomaa [14], 1 39 第 – 页。14 Rozenberg, G., Salomaa, A.（编）：《形式语言手册》，第 I 卷。Springer-Verlag，柏林（1997） 15. Sakarovitch, J.: 《自动机理论要素》。剑桥大学出版社，柏林（2009） 16. Salomaa, A.: 形式语言。学术出版社，纽约（1973） 17. Shyr, H.J.: 《自由幺半群与语言》。Hon Min 图书公司，台中，第二版（1991） 18. Shyr, H.J., Thierrin, G.: 码与二元关系。载于：Malliavin, M.P.（编）：《Paul Dubreil 代数研讨会》，巴黎 1975–1976（第 29 届）。《数学讲义》，第 586 卷，第 180–188 页（1977） 19. Yu, S.: 正则语言。载于：Rozenberg 和 Salomaa [14], 第 41–110 页