基于 Rabin-Mostowski 指数的广义量词

Denis Kuperberg 法国国家科学研究中心、LIP、Plume、里昂高等师范学院，法国
Damian Niwiński 信息学研究所，华沙大学，波兰
Paweł Parys

信息学研究所，华沙大学，波兰

Michał Skrzypczak
信息学研究所，华沙大学，波兰

arXiv:2601.04739v1 [cs.LO] 8 Jan 2026

摘要

本文引入了新的广义量词，使我们能够表达自动机的 Rabin-Mostowski 指数。我们的主要结果研究了扩展了这些量词的单子二阶逻辑（MSO）的表达能力与可判定性。我们在 ω-字和无限树两种范围内研究这些问题。结果表明，这两种情形截然不同。对于 ω-字，这些新量词可以有效地用纯单子二阶逻辑表达。相比之下，在无限树的情形下，加入这些量词会导致一种不可判定的形式系统。

为了实现指标量词消去，我们考虑用博弈量词扩展 MSO。作为工具，我们为这些量词提供了一种专门的量词消去过程。此外，我们还引入了一种新颖的转换器构造，用于在带有单子参数的 ω-正则博弈中实现策略。

2012年ACM学科分类 计算理论 → 无限对象上的自动机；计算理论 → 逻辑与验证；计算理论 → 树语言

关键词和短语 单子量词、可判定性、量词消去、奇偶自动机、博弈量词、Rabin-Mostowski 指数

资助 *Denis Kuperberg*：ANR ReCiProg*Damian Niwiński*：波兰国家科学中心（资助编号：2024/55/B/ST6/00318）*Paweł Parys*：波兰国家科学中心（资助编号：2024/55/B/ST6/00318）*Michał Skrzypczak*：波兰国家科学中心（资助编号：2024/55/B/ST6/00318）

1 引言

在 ω-字或无限树上研究的单子二阶逻辑（MSO）是验证理论中的黄金标准：它是一种健壮、表达能力强且仍然可判定的形式体系。围绕这一逻辑的研究通常沿着两条路径展开。

一种研究方向聚焦于 ω-字或树的 MSO 可定义语言的性质，尤其强调可判定性问题，旨在获得有效刻画。另一条路径可能更具挑战性，即在保持可判定性的同时，尝试扩展 MSO 的表达能力。这两条路径经常相互交织，基数的研究便是一个典型例子。首先，Niwiński [34] 证明了无限树的正则语言的基数可以被有效计算。随后，Bárány、Kaiser 和 Rabinovich [1]（另见 [20]）研究了 MSO 逻辑（在二叉树上）的一个扩展，该扩展加入了基数数量词，例如$\exists^{\geq\kappa} X.\, \varphi$（$\vec{W},\ X$），表示至少有 κ 个不同的集合 X 满足 φ（$\vec{W},\ X$）。事实证明，该扩展允许对基数数量词进行消去过程：作者将带有基数数量词的 MSO 有效地转换为纯单子二阶逻辑，从而使所研究的形式体系具有可判定性 [1]。

相比之下，Bojańczyk [2],引入的无界性 量词 U$X.\, \varphi(\vec{W}, X)$ 表示公式 $\varphi(\vec{W}, X)$ 对大小无界的有限集 X 成立，它使单子二阶逻辑得到真正的扩展。经过全面研究，人们证明了，即使在 ω-字 [4]上，单子二阶逻辑+U 也是不可判定的。然而，给定正则语言的无界性很容易判定（这得益于抽引引理的应用）；一种称为对角性的相关性质，即使对于 Caucal 层次所有层上的树语言，也已证明是可判定的 [13]。

Niwiński、Parys 和 Skrzypczak [37] 的结果也属于类似情形：作者表明，单子二阶逻辑可定义的良基关系的秩满足某种二分性质，并且可以有效地加以界定，尽管秩本身无法用单子二阶逻辑直接表达。

这些情形背后有一个普遍的模式，可分为几个层次。在基础层面，我们希望判定一种 ω-字或树的语言是否满足某个特定性质，通常与某种难点有关：不可数性、无界性、序数秩 ω_1等。随后，我们会问该性质能否推广到某种量词，以及单子二阶逻辑的扩展是否是真扩展，并最终能否判定。

本文的研究由 Rabin-Mostowski 指数问题所推动，这是自动机理论中的一个重要开放问题。用奇偶自动机（见下文）来表述，该问题是要寻找一个与给定类型（确定性、非确定性或交替）的自动机等价、且优先级数量最少的自动机。出于技术上的考虑，我们还考虑最小优先级，因此将指标 定义为一对数值 (i, j)（其中 i 可假设为 0 或 1）。回顾一下，在 ω-字上，指数层次仅对确定性自动机是严格的，而对非确定性和交替自动机则塌缩为第二层。对于无限树，非确定性和交替层次都是严格的 [6, 33]；确定性层次也严格，但意义较小，因为确定性树自动机无法刻画所有正则树语言。对于 ω-字上的自动机，计算指标的问题通常是可判定的 [47], ，而对于无限树上的自动机则仍然开放。若干特殊情形已被证明可判定，尤其是当输入树自动机是确定性自动机 [35, 36]、博弈自动机 [17]或 Büchi 自动机 [14, 44]时。Colcombet 和 Löding [15] 将非确定性指标问题化约为一个关于计数器自动机渐近行为的问题；他们的论文提出了许多有趣的想法（尤其是可引导性），但原问题至今仍未解决。

在本文中，我们从“上方”着手研究指标问题，也就是说，我们引入

一类与指标性质相对应的量词。利用集合（或其元组）与其特征函数（即带标记的无限字或树）之间的对应关系，新量词的一般形式为

$$\mathtt{I}^{D}_{\mathcal{R}} X.\, \varphi(W_1, \ldots, W_k, X)$$

其中，D 表示所涉及自动机的类型（确定性或非确定性），而$\mathcal{R}$ 则决定指标。对于赋值 $\bar{w}_1, \ldots, \bar{w}_k$，若存在一个类型为 D 、指标为 $\mathcal{R}$ 的自动机 $\mathcal{A}$，使得对每个 $\bar{x}$，公式 $\varphi(\bar{w}_1, \ldots, \bar{w}_k, \bar{x})$ 成立，当且仅当$\mathcal{A}$ 接受$\langle \bar{w}_1, \ldots, \bar{w}_k, \bar{x}\rangle$时，我们就称该公式成立。请注意，在上述情形中，只有 $\bar{x}$ 会变化，而$\bar{w}_i$保持固定，充当参数。

我们的主要结果分为两部分。首先，我们证明 MSO+$\mathtt{I}$ 可以有效地归约为 ω-字上的纯单子二阶逻辑。其次，我们证明 MSO+$\mathtt{I}$ 在无限树上不可判定。据我们所知，这是关于无限树上指标相关问题的首个否定性可判定性结果。事实上，对于量词 $\mathtt{I}_{\mathrm{safety}}$，我们就已能证明其不可判定性；该量词指的是仅仅避开某些指定拒绝状态的自动机。这与判定一个正则树语言能否由安全自动机识别是直接明了的这一事实形成鲜明对比，因为这相当于检查它在无限树的标准拓扑中是否闭合（例如参见 [15, 24]）。

为了得到我们结果中的正向部分，即在ω-字上进行指标量词消去，我们依赖于针对指数层次 [25, 46]的一种 Wadge 博弈变体。这些博弈可以自然地用带有博弈量词的单子二阶逻辑来表达（例如，参见 Moschovakis 的专著 [30]）。单子二阶逻辑+ 可归约为纯单子二阶逻辑，这一点可由 Kaiser [19]的结果推出（为求完整，我们针对当前设定给出了直接证明）；然而，我们需要一个更强的性质，使我们能够构造有限记忆策略（将 Büchi-Landweber 构造 [11] 与一致化 [23, 41]联系起来）。这与 Winter 和 Zimmermann [50] 等人关于序列一致化以及由转导器实现的函数的结果思路相近。为实现目标，我们证明了一个新颖的事实，它可视为 Büchi-Landweber 构造的参数化版本（适用于变量在某种意义上彼此分离的情形）。我们认为，博弈量词的一般理论以及这一新结果，都具有独立的研究价值和应用前景。

可以探讨一下，本文中的新量词是否符合 Mostowski 提出的广义量词概念 [31] （参见 [48] 综述）。其思想是，公式 $\mathtt{Q}x.\, \varphi(\vec{w},\ x)$ 表示满足 $\varphi(\vec{w},\ x)$ 的 x（固定参数 $\vec{w}$）属于论域的某个指定子集族（例如，对 $\exists$而言，是所有非空集合；对 $\forall$而言，则是整个论域的单元素集）。更一般地，量词可以约束 k 个变量（$\mathtt{Q}x_1 \ldots x_k.\, \varphi(\vec{w},\ \vec{x})$），并对应于一个 k 元关系族。这些概念可以推广到单子二阶逻辑；此时，在量词 $\mathtt{Q}X$ （或 $\mathtt{Q}\vec{X}$）的语义中，论域被其幂集取代。前面提到的例子，即基数数量词和无界性量词，都可以很容易地用这种方式来表示。新引入的指数量词和博弈量词也同样可以表示为广义量词。感兴趣的读者可参见第4节，我们将在那里更详细地讨论这个问题。

2 预备知识

字母表 A 是一个由有限个符号组成的非空集合。通常，我们用 A^* 表示 A上的有限字集合，用 A^+ 表示 A上的非空有限字集合，并用 A^ω 表示 A上的ω-字集合，即从 $\mathbb{N} = \{0, 1, 2, \ldots\}$ 到 A 的函数。空字记作 $\varepsilon \in A^*$，两个字 u 和 v 的拼接记作 $u \cdot v$。给定 $n \in \mathbb{N}$，以及

一个至少包含 n 个符号的有限字，或一个 ω-字 $\bar{w} = w_0 w_1 w_2 \cdots$，我们用 $\bar{w}\restriction_n$ 表示有限字 $w_0 w_1 \cdots w_{n-1}$，即将 $\bar{w}$ 限制在前 n 个符号上。形如 $x \cdot y \cdot y \cdot y \cdot \ldots$ 的 ω-字，其中 $x, y \in A^+$ 为有限字，称为最终周期。字上的前缀序记作 $\preceq$；若存在 $n \in \mathbb{N}$ 使得 $\bar{w} = \bar{w}'\restriction_n$，则有 $\bar{w} \preceq \bar{w}'$。

字母表 A 上的（完全、无限、二叉）树是任意函数 $t\colon \{\mathtt{L},\mathtt{R}\}^* \to A$；这里，$\{\mathtt{L},\mathtt{R}\}^*$ 中的一个字描述了从根 ε 到节点 $x \in \{\mathtt{L},\mathtt{R}\}^*$ 的一条路径，其中 $\mathtt{L}$ 是左子节点，$\mathtt{R}$ 是右子节点。此类节点的标签为 $t(x) \in A$。所有此类树的集合记作 $\mathbf{Tr}_A$。

我们使用在树中导航的标准术语，特别是，如果 $x \succeq y$，则 x 是 y 的后代。类似地，我们使用祖先、父节点和兄弟节点这些术语。

子集 $L \subseteq A^*$、$L \subseteq A^\omega$ 或 $L \subseteq \mathbf{Tr}_A$ 称为语言。

转导器。本文使用从一个字母表到另一个字母表的（顺序、确定性、有限记忆）转导器。设 A_W、A_Y 为某些字母表。从 A_W 到 A_Y 的转导器 τ（记作 $\tau\colon A_W \twoheadrightarrow A_Y$）是一个元组 $\tau = \langle A_W, A_Y, Q_\tau, \iota_\tau, \delta_\tau \rangle$，其中：

- Q_τ 是一个有限的状态，$\iota_\tau \in Q_\tau$ 是初始状态，δ_τ：
- $Q_\tau \times A_W \to A_Y \times Q_\tau$ 是转移函数。

给定一个输入 ω-字 $\bar{w} = w_0 w_1 w_2 \cdots \in (A_W)^\omega$，我们归纳定义运行 $\bar{\rho} \overset{\text{定义}}{=} \rho_0 \rho_1 \rho_2 \cdots \in Q^\omega$ 以及输出 ω-字 $\tau(\bar{w}) \overset{\text{定义}}{=} y_0 y_1 y_2 \cdots \in (A_Y)^\omega$，令 $\rho_0 \overset{\text{定义}}{=} \iota_\tau$，并且对所有 $n \in \mathbb{N}$，有 $(y_n, \rho_{n+1}) \overset{\text{定义}}{=} \delta_\tau(\rho_n, w_n)$。

给定两个转换器 $\tau\colon A_W \twoheadrightarrow A_Y$ 和 $\tau'\colon A_Y \twoheadrightarrow A_Z$，很容易构造二者的复合，即一个转换器 $\theta\colon A_W \twoheadrightarrow A_Z$，使得对于每个 $\bar{w} \in (A_W)^\omega$，都有 $\theta(\bar{w}) = \tau'\big(\tau(\bar{w})\big)$。

奇偶索引。假设 $i, j \in \mathbb{N}$ 是满足 $i \le j$ 的自然数。（强）奇偶索引 $P_{i,j}$ 和弱奇偶指数 $W_{i,j}$ 由以下语言定义

$$P_{i,j} \overset{\text{def}}{=} \{k_0 k_1 k_2 \ldots \in \{i, i{+}1, \ldots, j\}^\omega \mid \limsup_{n\to\infty} k_n \equiv 0 \bmod 2\},$$
$$W_{i,j} \overset{\text{def}}{=} \{k_0 k_1 k_2 \ldots \in \{i, i{+}1, \ldots, j\}^\omega \mid \sup_{n\in\mathbb{N}} k_n \equiv 0 \bmod 2\}.$$

一个指标是形如 $\mathbf{R} = A_\mathcal{R}, L_\mathcal{R}$ 的二元组，它要么是 $\mathbf{P}_{i,j} = \{i, i{+}1, \ldots, j\}\, P_{i,j}$，要么是 $\mathbf{W}_{i,j} = \{i, i{+}1, \ldots, j\}\, W_{i,j}$，其中 $i, j \in \mathbb{N}$ 满足 $i \le j$。

指标的常见名称包括：*Büchi* 对应 $\mathcal{P}_{1,2}$（优先级 2 出现无限多次），余布奇 对应 $\mathcal{P}_{0,1}$（优先级 1 出现有限次），安全性 对应 $\mathcal{W}_{0,1}$（到达优先级 1 意味着我们拒绝），以及可达性 对应 $\mathcal{W}_{1,2}$（到达优先级 2 意味着我们接受）。

关于 ω-字的自动机。字母表为 A、指标为 $\mathcal{R} = \langle A_\mathcal{R}, L_\mathcal{R} \rangle$ 的非确定性奇偶 -字自动机 是一个元组 $\mathcal{D} = \langle A, \mathcal{R}\ Q_\mathcal{D}, \iota_\mathcal{D}, \delta_\mathcal{D} \rangle$，其中：

- $Q_\mathcal{D}$ 是一个有限状态集合，$\iota_\mathcal{D} \subseteq Q_\mathcal{D}$ 是初始状态集合，$\delta_\mathcal{D} \subseteq Q_\mathcal{D} \times A \times A_\mathcal{R} \times Q_\mathcal{D}$ 是转移关系。此外，该自动机是完全的[1]，也就是说，对于每个 $q \in Q_\mathcal{D}$ 和 $a \in A$，至少存在一个形如 $(q, a, k, q') \in \delta_\mathcal{D}$ 的转移。

[1] 在考察自动机的弱索引时，这一技术性假设发挥了作用。

自动机 D 在输入 ω-字 $\bar{w} = w_0 w_1 w_2 \cdots \in A^\omega$ 上运行并产生输出 ω-字 $k = k_0 k_1 k_2 \cdots \in (A_\mathcal{R})^\omega$，是指一个状态序列 $\bar{\rho} = \rho_0 \rho_1 \rho_2 \cdots \in (Q_\mathcal{D})^\omega$，满足 $\rho_0 \in \iota_\mathcal{D}$，且对每个 $n \in \mathbb{N}$，都有 $(\rho_n, w_n, k_n, \rho_{n+1}) \in \delta_\mathcal{D}$。如果存在 D 在 $\bar{w}$ 上的一次运行，产生的 ω-字 k 属于 $L_\mathcal{R}$，则 ω-字 $\bar{w}$ 被 D 接受。

此类自动机的语言记作 $\mathrm{L}(\mathcal{D}) \subseteq A^\omega$，是由 $\mathcal{D}$ 接受的 ω-字 $\bar{w} \in A^\omega$ 的集合。语言 $L \subseteq A^\omega$ 是 ω-正则的，当且仅当它是某个自动机的语言。

如果 $\iota_\mathcal{D}$ 是单元素集合，且转移关系 $\delta_\mathcal{D}$ 实际上是一个函数 $\delta_\mathcal{D} \colon Q_\mathcal{D} \times A \to A_\mathcal{R} \times Q_\mathcal{D}$，则称自动机是确定性的；在这种情况下，对于每个输入 ω-字 $\bar{w} \in A^\omega$，$\mathcal{D}$ 上都有唯一的运行。

▶ 注 2.1. 如果指标 $\mathcal{R} = \langle A_\mathcal{R}, L_\mathcal{R} \rangle$ 固定，那么在 A 上的指标为 $\mathcal{R}$ 的确定性自动机 $\mathcal{D}$ 与转导器 $\tau \colon A \twoheadrightarrow A_\mathcal{R}$ 自然一一对应，且满足 $\mathrm{L}(\mathcal{D}) = \{ \bar{w} \in A^\omega \mid \tau(\bar{w}) \in L_\mathcal{R} \}$。

拉姆齐定理。令 C 为一个有限颜色集。集合 X 的边标记是一个函数，它为每条边 $\{i, j\} \subseteq X$（其中 $i \neq j$）指定一个来自 C 的颜色。给定一个边标记，如果集合 $I \subseteq X$ 中所有边 $\{i, j\} \subseteq I$ 的颜色都相同，我们就称集合 $I \subseteq X$ 为单色的。

▶**定理 2.2** (拉姆齐)。令 C 为一个有限集合，且令 $k \in \mathbb{N}$。那么，存在一个可计算常数 $r \in \mathbb{N}$，使得对于将 $\{0, 1, \ldots, r-1\}$ 用来自 C 的颜色进行的任意边标记，都存在一个大小为 k 的单色集合 $I \subseteq \{0, 1, \ldots, r-1\}$。

此外，对于将 $\mathbb{N}$ 用来自 C 的颜色进行的任意边标记，都存在一个无限的单色集合 $I \subseteq \mathbb{N}$。

半群与幺半群。带有结合二元运算 $(\cdot)$ 的代数结构 $\langle S, (\cdot) \rangle$ 称为半群。幺半群是包含单位元 S 的半群，该单位元满足 $\varepsilon \in S$，且对每个 $s \in S$ 都有 $\varepsilon \cdot s = s \cdot \varepsilon = s$。每个半群 S 都可以通过添加一个形式中性元 ε 扩充为幺半群 $S_{+\varepsilon} = S \cup \{\varepsilon\}$，其乘积按适当方式定义。幂等元是满足 $e \cdot e = e$ 的元素 $e \in S$。

以下事实是拉姆齐定理的一个标准应用（参见定理 2.2）。

▶ **事实 2.3**。对于每个有限半群 S，都存在一个可计算常数 $r \in \mathbb{N}$，使得对于每个字 $s_0 s_1 \cdots s_{r-1} \in S^r$，都存在一对位置 $0 \leq i < j < r$，使得 $e \overset{def}{=} s_{i+1} \cdot s_{i+2} \cdot \ldots \cdot s_j$ 是一个幂等元。

特别地，令 $c \overset{定义}{=} s_0 \cdot s_1 \cdot \ldots \cdot s_j$，则有

$$c \cdot e = s_0 \cdot s_1 \cdot \ldots \cdot s_{i-1} \cdot e \cdot e = s_0 \cdot s_1 \cdot \ldots \cdot s_{i-1} \cdot e = c.$$

Wilke 代数。在本文中，我们将 Wilke 代数用作 ω-半群的表示，如 Perrin和Pin [38] 中所述。一个 *Wilke* 代数 S 由两个集合（S^{fin}，S^{inf}）和两个乘积运算组成

$$S^{\mathrm{fin}} \times S^{\mathrm{fin}} \to S^{\mathrm{fin}} \qquad \text{and} \qquad S^{\mathrm{fin}} \times S^{\mathrm{inf}} \to S^{\mathrm{inf}}$$

记作 $s \cdot s'$，用于操作数 s，s'；另一个运算 $S^{\mathrm{fin}} \to S^{\mathrm{inf}}$ 记作 s^ω，用于操作数 $s \in S^{\mathrm{fin}}$。此外，这些运算必须满足自然的结合律公理，特别是 S^{fin} 必须构成半群。每个有限 Wilke 代数 S 都唯一确定无限乘积运算 $\odot \colon (S^{\mathrm{fin}})^\omega \to S^{\mathrm{inf}}$，该运算满足结合律。特别地，$\odot(sss\cdots) = s^\omega$，且 $\odot(s_0 s_1 \cdots) = s_0 \cdot \odot(s_1 s_2 \cdots)$。

两个 Wilke 代数 S 和 T 之间的同态 α 是由两个函数 $\alpha^{\mathrm{fin}}\colon S^{\mathrm{fin}} \to T^{\mathrm{fin}}$ 和 $\alpha^{\mathrm{inf}}\colon S^{\mathrm{inf}} \to T^{\mathrm{inf}}$ 组成的一对，它们与代数的所有运算以及无限乘积 $\odot$ 相容。

识别。 Wilke 代数的一个典型例子是 $A^{\mathsf{W}} \stackrel{\mathrm{def}}{=} \langle A^+, A^\omega\rangle$，其中 A 是一个字母表。这个 Wilke 代数的运算包括拼接 $\cdot$、无限重复 $v^\omega \stackrel{\mathrm{def}}{=} v\cdot v\cdot v\cdot \ldots \in A^\omega$（对于 $v \in A^+$），以及无限乘积 $\odot(v_0v_1v_2\cdots) \stackrel{\mathrm{def}}{=} v_0\cdot v_1\cdot v_2\cdot \ldots \in A^\omega$（对于 $v_0, v_1, v_2, \ldots \in A^+$）。

结合律性质表明，若 $\alpha\colon A^{\mathsf{W}} \to S$ 是到一个有限 Wilke 代数的同态，那么对于任意有限字序列 $v_0, v_1, \ldots \in A^+$，都有

$$\alpha(v_0 \cdot v_1 \cdot v_2 \cdot \ldots) = \odot\big(\alpha(v_0)\alpha(v_1)\alpha(v_2)\cdots\big). \tag{2.1}$$

请注意，若 Wilke 代数是有限的，则可通过提供其元素列表以及所有运算的“乘法表”，将其表示为算法的输入。Wilke 代数的关键特性在于其能够识别 ω-正则语言，如下述定理所述。

▶ **定理 2.4** ([49])。给定一个由 ω-正则语言组成的元组 $(L_0, \ldots, L_{k-1})$，且对所有 $i < k$ 均有 $L_i \subseteq A^\omega$，可以有效地计算出一个有限 *Wilke* 代数 S，以及一个同态 $\alpha\colon A^{\mathsf{W}} \to S$ 和一个集合元组 $(F_0, \ldots, F_{k-1})$，其中对于每个 $i < k$，集合 $F_i \subseteq S^{\mathrm{inf}}$ 满足 $L_i = \alpha^{-1}(F_i)$。我们称 α 识别 $(L_0, \ldots, L_{k-1})$，其识别方式由 $(F_0, \ldots, F_{k-1})$ 给出。

此外，还可以要求 α 是满射，即 $\alpha(A^+) = S^{\mathrm{fin}}$ 且 $\alpha(A^\omega) = S^{\mathrm{inf}}$。

令 $\bar z = z_0z_1z_2\cdots \in (S^{\mathrm{fin}}_{+\varepsilon})^\omega$（回忆一下，$S^{\mathrm{fin}}_{+\varepsilon}$ 是通过添加一个形式中性元 ε 对 S^{fin} 进行扩充得到的）。如果 $\bar z$ 包含来自 S^{fin} 的无限多个符号，也就是说，包含不同于 ε 的符号，我们就称 $\bar z$ 是饱和的。在这种情况下，$\odot(\bar z)$ 有良好定义：我们可以从 $\bar z$ 中删去所有符号 ε，得到一个 ω-字 $\bar z' \in (S^{\mathrm{fin}})^\omega$，并令 $\odot(\bar z) \stackrel{\mathrm{def}}{=} \odot(\bar z')$。此定义同样满足公式（2.1）中的结合律性质。

前瞻与复合。 假设 $\alpha\colon A^{\mathsf{W}} \to S$ 是到有限 Wilke 代数 $S = (S^{\mathrm{fin}}, S^{\mathrm{inf}})$ 中的一个同态。对于每个 $\bar w = w_0w_1w_2\cdots \in A^\omega$，该同态定义了前瞻 $\mathsf{lk}_\alpha(\bar w) \in (S^{\mathrm{inf}})^\omega$，对每个位置 $n \in \mathbb{N}$ 定义如下：

$$\big(\mathsf{lk}_\alpha(\bar w)\big)_n \stackrel{\mathrm{def}}{=} \alpha\big(w_{n+1}w_{n+2}w_{n+3}\cdots\big) \in S^{\mathrm{inf}}.$$

请注意，在位置 $n \in \mathbb{N}$ 输出一个字母时，转导器会使用位置 0，1，…，n 上的字母。另一方面，位置 n 处的前瞻取决于位置 $n{+}1$，$n{+}2$，$n{+}3$，… 为了创建一个输出 ω-字，使其在 A_Y 中的输出字母同时取决于输入 ω-字的过去和未来，我们考虑输出字母为函数（$S^{\mathrm{inf}} \to A_Y$）的转导器，然后将这些函数应用于前瞻产生的 S^{inf} 中字母。

$^{-}$ 为简化记法，我们使用以下简写：若 $f = f_0f_1f_2\cdots \in (A_X \to A_Y)^{\omega^{-}}$ 且 $\bar x = x_0x_1x_2\cdots \in (A_X)^\omega$，则对每个位置 $n \in \mathbb{N}$，定义 $f \bullet \bar x \in (A_Y)^\omega$ 为 $(f \bullet \bar x)_n = f_n(x_n)$。

无限树上的自动机。 字母表为 A 且指标为 $\mathcal{R} = \langle A_{\mathcal{R}}, L_{\mathcal{R}}\rangle$ 的非确定性奇偶树自动机是一个元组 $\mathcal{A} = \langle A, \mathcal{R}\ Q_{\mathcal{A}}, \iota_{\mathcal{A}}, \Delta_{\mathcal{A}}\rangle$，其中 $Q_{\mathcal{A}}$ 是一个有限状态集合，$\iota_{\mathcal{A}} \subseteq Q_{\mathcal{A}}$ 是一个初始状态集合，而 $\Delta_{\mathcal{A}} \subseteq Q \times A \times A_{\mathcal{R}} \times Q \times Q$ 是一个转移关系。

同样，我们要求该自动机是完备的，也就是说，对于每个 $q \in Q_{\mathcal{A}}$ 和 $a \in A$，它都必须至少包含一个转移 $(q, a, k, q_{\mathtt{L}}, q_{\mathtt{R}}) \in \Delta_{\mathcal{A}}$。

自动机 A 在树 $\tilde{t} \in \mathrm{Tr}_A$ 上的一次运行，若产生输出树 $\tilde{\eta} \in \mathrm{Tr}^{A_{\mathcal{R}}}$，则它是一棵树 $\tilde{\rho} \in \mathrm{Tr}^{Q_{\mathcal{A}}}$，满足 $\tilde{\rho}(\varepsilon) \in \iota_{\mathcal{A}}$ 且 $(\tilde{\rho}(v)\ \tilde{t}(v)\ \tilde{\eta}(v)\ \tilde{\rho}(v\mathtt{L})\ \tilde{\rho}(v\mathtt{R})) \in \Delta_{\mathcal{A}}$ 对所有节点 $v \in \{\mathtt{L}, \mathtt{R}\}^*$ 均成立。如果存在自动机 A 在 $\tilde{t}$ 上运行并产生一棵树 $\tilde{\eta}$，使得对于每条分支 $\bar{w} \in \{\mathtt{L}, \mathtt{R}\}^\omega$，序列 $\tilde{\eta}(\bar{w}\restriction_0)\tilde{\eta}(\bar{w}\restriction_1)\tilde{\eta}(\bar{w}\restriction_2)\cdots \in (A_{\mathcal{R}})^\omega$ 属于 $L_{\mathcal{R}}$，则称树 $\tilde{t} \in \mathrm{Tr}_A$ 被 A 接受。自动机 A 的语言是它所接受的树的集合。若语言 $L \subseteq \mathrm{Tr}_A$ 是某个自动机 A 的语言，则称其为正则树语言。

如果 $\iota_{\mathcal{A}}$ 是单元素集合，且 $\Delta_{\mathcal{A}} \colon Q \times A \to A_{\mathcal{R}} \times Q \times Q$ 是一个函数，则称树自动机是确定性的（自顶向下）。

单子二阶逻辑。单子二阶逻辑的公式在适当的结构中求值；在本文的情形中，结构是 $\mathbb{N}$，配有后继关系（对于 ω-字），或者 $\{\mathtt{L}, \mathtt{R}\}^*$，配有左子关系和右子关系（对于树）。结构中的元素称为位置或节点。通常，单子二阶逻辑中的单子变量表示一个位置集合；也可以将其视为字母表为 $\{0,1\}$ 上的字或树，其中 1 表示属于该集合的位置。本文采用一种看似更一般的设定：每个单子变量 X 表示某个字母表 A_X 上的字或树，该字母表可能比 $\{0,1\}$ 更大。下文通常假设每个变量 X 都对应一个固定字母表 A_X，但有时也会在量词旁明确指定字母表（例如写作 $\exists X \in (A_X)^\omega . \varphi(X)$）。于是，对于一个字母 $x \in A_X$ 和一个一阶变量 v，我们有一个原子公式 $X(v) = x$，用于检验位置 v 上 X 的字母是否为 x。这种对单子变量的理解不会增强单子二阶逻辑的表达能力，因为取值于 A_X 的变量可以表示为 $|A_X|$ 个通常的集合变量的元组，并要求它们划分论域（甚至只需 $\lceil \log |A_X| \rceil$ 个集合变量即可）。

根据单子二阶逻辑与正则语言之间的等价性 [10, 27, 39]，我们知道，对于每个单子二阶逻辑公式 $\varphi(X_1, \ldots\ldots, X_n)$，都可以构造一个确定性奇偶 ω-字自动机（针对 ω-字的情形），或一个非确定性奇偶树自动机（针对树的情形），其定义在字母表 $A_{X_1} \times \ldots \times A_{X_n}$ 上，并且恰好接受该字母表上的那些 ω-字／树 $\bar{w}$，使得 $\varphi(\pi_1(\bar{w})\ldots\ldots, \pi_n(\bar{w}))$ 成立。其中，每个 $\pi_i(\bar{w})$ 都是由 $\bar{w}$ 将所有位置的标签投影到其第 i 个坐标后得到的。注意，所构造自动机的指标取决于公式 φ，且一般无法将其限制在某个上界之内 [5, 33, 47]。

为简化记号，我们将定义在此类乘积字母表 $A_{X_1} \times \ldots \times A_{X_n}$ 上的结构 $\bar{w}$ 与相应字母表上的结构元组视为同一对象：$\langle \pi_1(\bar{w})\ldots\ldots, \pi_n(\bar{w}) \rangle$。特别地，对于公式 $\varphi(X_1, \ldots\ldots, X_n)$，我们可以讨论该公式所定义的语言，即由满足 $\varphi(\pi_1(\bar{w})\ldots\ldots, \pi_n(\bar{w}))$ 的、定义在 $A_{X_1} \times \ldots \times A_{X_n}$ 上的结构 $\bar{w}$ 构成的集合。由于公式可以转换为自动机，这些语言总是正则的。

博弈。我们采用无限时长完美信息博弈的一般框架，由两名玩家进行（通常称为玩家 I 和玩家 II）。这样的博弈由元组 $\mathsf{G} = A, L_{\mathcal{G}}, V_{\mathcal{G}} = V_{(\mathrm{I})}\, \mathsf{G} \sqcup V_{(\mathrm{II})}\, \mathsf{G}\ \iota_{\mathcal{G}}, \delta_{\mathcal{G}}$ 给出，其中 A 是字母表，$L_{\mathcal{G}} \subseteq A^\omega$ 是胜利条件，$V_{\mathcal{G}}$ 是一个（可能无限的）位置集合，并划分为各玩家各自的位置；$\iota_{\mathcal{G}} \in V_{\mathcal{G}}$ 是初始位置，$\delta_{\mathcal{G}} \subseteq V_{\mathcal{G}} \times A \times V_{\mathcal{G}}$ 是边关系（同样满足完备性，即每个 $v \in V_{\mathcal{G}}$ 至少有一条边 $(v, a, v') \in \delta_{\mathcal{G}}$）。字母 $a \in A$ 称为边 $(v, a, v') \in \delta_{\mathcal{G}}$ 的标签。

这类博弈的对局分回合进行，初始位置为 $v_0 = \iota_{\mathcal{G}}$。在第 $n \in \mathbb{N}$ 回合，满足 P 的玩家 $v_n \in V_{(P)\,\mathcal{G}}$ 选择一条边 $(v_n, k_n, v_{n+1}) \in \delta_{\mathcal{G}}$，移动到下一个位置 v_{n+1}。在一场无限对局结束后，当且仅当 玩家 II 获胜 $\bar{k} \stackrel{\mathrm{def}}{=} k_0 k_1 k_2 \cdots$

属于 $L_{\mathcal{G}}$。经典定理 [26] 表明，如果 $L_{\mathcal{G}}$ 足够简单，那么其中一位玩家可以确保赢得这场博弈，也就是说，拥有一项获胜策略。一般而言，玩家 P 的这类策略是一个树形对象，但我们主要使用位置策略，也就是满足以下条件的函数 $\sigma^{(P)}\colon V_{(P)\mathcal{G}} \to \delta_{\mathcal{G}}$：对于每个 $v \in V_{(P)\mathcal{G}}$，都有 $\sigma^{(P)}(v) = (v, k, v')$，其中某个 $k \in A$ 和 $v' \in V_{\mathcal{G}}$ 满足条件。

指标为$R = \langle A_{\mathcal{R}}, L_{\mathcal{R}} \rangle$ 的奇偶博弈 是上述博弈$\mathcal{G}$，其中 $A = A_{\mathcal{R}}$ and $L_{\mathcal{G}} = L_{\mathcal{R}}$。

▶ **定理 2.5** ([16, 32])。 如果 G 是一个奇偶博弈，那么某位玩家 P 在 G 中具有位置获胜策略 $\sigma^{(P)}$。

3 新量词

本节介绍本文研究的两类量词。为此，我们遵循如下约定：在公式 $\mathtt{Q}X.\varphi(W_1, \dots, W_k, X)$ 中，所有参数变量 $W_1, \dots, W_k$ 都合并为乘积字母表上的一个自由变量 W，如上文所述。因此，我们关注形如 $\mathtt{Q}X.\varphi(W, X)$ 的公式，即使 W 的相应坐标来自不同的外层量词。

指数算子。 考虑一个新量词 $\mathtt{I}^{D}_{\mathcal{R}} X.\varphi(W, X)$，其中 $D \in \{\mathsf{dt}, \mathsf{nd}\}$ 决定所涉及自动机的类型，而$\mathcal{R}$ 是一个指标（可以是强奇偶指数$\mathcal{P}_{i,j}$，也可以是弱奇偶指数$\mathcal{W}_{i,j}$）。对于参数 $\bar{w} \in (A_W)^\omega$，如果存在一个指标为$\mathcal{R}$的自动机$\mathcal{A}_{\bar{w}}$，它可以是确定性的（$D = \mathsf{dt}$）或非确定性的（$D = \mathsf{nd}$），使得对于每个 $\bar{x} \in (A_X)^\omega$，公式 $\varphi(\bar{w}, \bar{x})$ 成立，当且仅当$\mathcal{A}_{\bar{w}}$接受$\langle \bar{w}, \bar{x} \rangle$。

注意，参数 $\bar{w}$ 在上述定义中有两种作用。首先，自动机 $\mathcal{A}_{\bar{w}}$ 可能取决于参数 $\bar{w}$。其次，自动机在验证给定的 $\bar{x}$ 是否使 $\varphi(\bar{w}, \bar{x})$ 为真时，不仅可以访问 $\bar{x}$，还可以访问参数 $\bar{w}$（尤其是，该自动机使用字母表 $A_W \times A_X$）。

▶ 注 3.1。我们可以思考一下，如果采用另一种指数量词语义，会有什么变化：在这种语义下，假设的自动机无法访问参数 $\bar{w}$，而只能读取被量化的 ω-字 $\bar{x}$。此时，这种形式体系立即变得不可判定。事实上，考虑最简单的公式 $\mathtt{I}^{\mathsf{dt}}_{\mathcal{W}_{0,1}} X.(X = W)$，其中包含确定性安全指数量词。那么，对于给定的 $\bar{w}$，满足 $\bar{x} = \bar{w}$ 的 ω-字集合 $\bar{x}$ 为 $\{\bar{w}\} \subseteq (A_W)^\omega$。当且仅当 $\bar{w}$ 是最终周期的，这种语言才能被确定性安全自动机识别。根据 Bojańczyk 等人 [3], 的研究，这种扩展逻辑是不可判定的。

请注意，确定性和非确定性指标量词对ω字和树都适用。令 MSO+$\mathtt{I}$ 表示通过指标量词扩展的单子二阶逻辑。

博弈量词。 作为研究指数量词的一种自然方式，我们需要在单子二阶逻辑中形式化博弈量词 的概念（见 [22, § 20.D] 和 [7, 8, 18, 21, 28, 30]）。该量词记作 $X \rightarrowtail Y$（或者，在某些论文中使用符号 $\supset$），它约束两个单子变量 X 和 Y。一个公式

$$X \rightarrowtail Y.\varphi(W, X, Y)$$

在给定参数 $\bar{w} \in (A_W)^\omega$ 时成立，当且仅当玩家 II 在博弈 G($\bar{w}, \varphi$) 中拥有获胜策略，该博弈定义如下。[2] 该博弈由无限多个回合组成。在一回合 $n \in \mathbb{N}$ 中，玩家 I 提出一个字母 $x_n \in A_X$，玩家 II 则以一个字母 $y_n \in A_Y$ 作答。最后，当且仅当 $\varphi(\bar{w}, \bar{x}, \bar{y})$ 对 $\bar{x} \text{ def} = x_0x_1x_2\cdots \in (A_X)^\omega$ 和 $\bar{y} \text{ def} = y_0y_1y_2\cdots \in (A_Y)^\omega$ 成立时，玩家 II 获胜。这个博弈很容易表示为一个形式博弈 G，其位置 $V^{(\mathrm{I})}$ G def= $\mathbb{N}$ 和 V(II) G def= $\mathbb{N} \times A_X$ 以及 $L_\mathcal{G}$ 由 φ 给出；不过，我们无需研究这个博弈的确切结构。

通常，人们在所涉及的博弈是确定的上下文中应用博弈量词，尽管即使没有这一假设，该定义仍然有意义。

令 MSO+ 表示通过博弈量词扩展的单子二阶逻辑。注意，按照其定义，博弈量词仅对 ω- 字有意义，因为无限时长博弈的时间结构的形状为 ω。

4 广义量词

在本节中，我们将本文引入的量词与广义量词这一一般概念联系起来。Mostowski [31] 提出了它们，将其作为一种抽象的逻辑构造，用于推广经典量词 $\exists$ 和 $\forall$。此后，它们成为逻辑各种应用中的重要工具（例如，参见 [48] 中的综述）。

在句法层面，量词 $\mathtt{Q}$ 通过构造 $\mathtt{Q}x.\varphi(\vec{w}, x)$扩展语言，其中 φ是任意公式。在这里，变量 x 由 $\mathtt{Q}$约束，而 $\vec{w} = (w_1, \ldots, w_k)$中的变量保持自由。在语义层面，量词与一个算子相关联；对于任意结构 M（其论域为 M），该算子定义集合族 $\mathtt{Q}^\mathcal{M} \subseteq \mathrm{P}(M)$。于是，给定赋值 $\vec{w} \mapsto \vec{a} \in M^k$，公式 $\mathtt{Q}x.\varphi(\vec{a}, x)$在 M 中成立，当且仅当集合 $\{b \in M \mid \varphi(\vec{a}, b)$ 在 M 中成立$\}$属于 $\mathtt{Q}^\mathcal{M}$。在这一设置中，$\exists^\mathcal{M}$ 是 M的所有非空子集构成的集合族，而 $\forall^\mathcal{M} = \{M\}$。作为一个不太标准的例子，可以表达这样的性质：满足 $\varphi(\vec{w}, x)$的 x的集合的基数属于某个指定的基数类（即，$\exists^\infty$ 表示该集合是无限的）；或者，满足 φ的 x与不满足它的 x具有相同的基数。通常假设集合族 $\mathtt{Q}^\mathcal{M}$在 M的置换下保持不变量，但在某些情况下，放宽这一要求也是合理的。

更一般地，可以考虑 n-元量词，其中一个量词 $\mathtt{Q}$ 同时约束 n 个变量，并且相应地，$\mathtt{Q}^\mathcal{M}$ 是定义在 M上的一族 n-元关系。例如，如果 $n = 2$ 且 $\mathtt{Q}^\mathcal{M}$ 是矩形类，即 $\mathtt{Q}^\mathcal{M} = \{X \times Y \mid X, Y \in \mathrm{P}(M)\}$，那么$\mathtt{Q}xy.\varphi(\vec{w}, x, y)$ 表示如下事实：只要 $\varphi(\vec{a}, b_1, c_1)$ 和 $\varphi(\vec{a}, b_2, c_2)$ 在 M 中成立，那么$\varphi(\vec{a}, b_1, c_2)$ 和 $\varphi(\vec{a}, b_2, c_1)$ 也成立。

上述概念可以推广到单子二阶逻辑（MSO）：在一元情形下为 $\mathtt{Q}^\mathcal{M} \subseteq \mathrm{P}(\mathrm{P}(M))$，一般情形下为 $\mathtt{Q}^\mathcal{M} \subseteq \mathrm{P}((\mathrm{P}(M))^n)$。事实上，文献中已有若干此类广义量词，其中著名的例子是弱量词，也就是将量词 $\exists$ 和 $\forall$ 限制在有限集合上的量词。引言中提到的基数数量词和无界性量词也可以在这一框架下表述。

[2] 传统上，在 Moschovakis 和 Kechris 的著作中，[22, 30] “博弈量词”要求玩家 I 赢得博弈；然而，在自动机理论语境中（例如 Church 综合问题 [11, 40]）或 Wadge 博弈 [46], 中，更常见的做法是关注玩家 II。

10 基于 Rabin-Mostowski 指数的广义量词

博弈量词。我们首先讨论如何将前文引入的博弈量词视为结构 $\mathbb{N}$ 上的广义 MSO 量词。为说明这一思路，我们先来看一阶逻辑中的一个简单例子。考虑一个公式

$$\forall x.\, \exists y.\, \forall x'.\, \exists y'.\, \varphi(\vec{w}, x, y, x', y').$$

显然，它在结构 $\mathcal{M}$ 中的含义可以看作是两个玩家（分别记为 $\exists$ 和 $\forall$）进行的一场包含 4 个回合的博弈。现在，可以用一个四元量词替换这组 4 个量词，使公式变为 $\mathtt{Q}xyx'y'.\,\varphi(\vec{w}, x, y, x', y')$。$\mathtt{Q}$ 的语义由一项性质来规定，即四元关系 r 在 $\mathtt{Q}^{\mathcal{M}}$ 中应具备的性质。从博弈的角度来看，在这场博弈中，玩家 I 和玩家 II 交替从 M中选取元素，玩家 II 应有一种策略，能够迫使选出的四元组属于 r。

现在考虑公式 $\varphi(W, X, Y)$，它在结构 $\mathbb{N}$中解释，其中 W、X、Y是集合变量（更一般地说，它们可以是一些集合变量的元组）。考虑一场无限博弈，玩家 I 和玩家 II 交替选取 $\{0, 1\}$中的比特，从而得到一个无限序列

$$x_0, y_0, x_1, y_1, x_2, y_2, \ldots, x_n, y_n, \ldots$$

序列 x_0，x_1，x_2，.....和 y_0，y_1，y_2，.....分别构成某些子集 $\bar{x}$ 和 $\bar{y}$ 的特征函数，这些子集属于 $\mathbb{N}$。现在，对于赋值 $W \mapsto \bar{w}$，由博弈量词定义的公式

$$\exists X \forall Y.\, \varphi(\bar{w}, X, Y)$$

成立，当且仅当玩家 II 有一种策略，能够确保公式 $\varphi(\bar{w}, \bar{x}, \bar{y})$ 在 $\mathbb{N}$ 中成立。博弈量词可以定义为一个二元广义单子二阶逻辑量词。其语义 $\mathbb{N}$ 由 $\mathsf{P}(\mathbb{N})$ 上的一个二元关系族定义，该关系族包含所有关系 $R \subseteq \mathsf{P}(\mathbb{N}) \times \mathsf{P}(\mathbb{N})$；对于上述博弈，玩家 II 有一种策略，能够确保最终得到的有序对 $(\bar{x}, \bar{y})$ 属于 R。因此，公式 $\exists X \forall Y.\, \varphi(\bar{w}, X, Y)$ 成立，当且仅当关系 $\{(\bar{x}, \bar{y}) \mid \varphi(\bar{w}, \bar{x}, \bar{y})$ 在 $\mathbb{N}\}$ 中属于 $\mathbb{N}$。

指数算子。为了将我们的新指数量词 $\mathtt{I}^D_{\mathcal{R}} X.\, \varphi(\vec{W}, X)$ 表述为一种广义 MSO 量词，具体起见，我们聚焦于完全二叉树的 MSO 理论，其论域为 $\{\mathtt{L}, \mathtt{R}\}^*$。由于我们假设自动机读取 W 和 X 的值，因此这一构造不属于一元情形；不过，与博弈量词类似，它可以表示为一个二元量词，或者更一般地，表示为一个 $(k+\ell)$ 元量词（如果 $\vec{W}$ 是一个 k-向量，而 X 是一个 ℓ-向量）。

为简单起见，我们考虑 $k = \ell = 1$；推广到更高的 k, ℓ 是直接的。关键在于选择 $\mathsf{P}(\{\mathtt{L}, \mathtt{R}\}^*)$ 上的一个二元关系类，作为该量词预期的语义。对于一个二元关系 $r \subseteq \mathsf{P}(\{\mathtt{L}, \mathtt{R}\}^*) \times \mathsf{P}(\{\mathtt{L}, \mathtt{R}\}^*)$，以及一个集合 $K \in \mathsf{P}(\{\mathtt{L}, \mathtt{R}\}^*)$，我们将 K 对 r 的截 定义为如下二元关系

$$r_K = r \cap \big(\{K\} \times \mathsf{P}(\{\mathtt{L}, \mathtt{R}\}^*)\big) = \big\{(K, L) \in r \mid L \in \mathsf{P}(\{\mathtt{L}, \mathtt{R}\}^*)\big\}.$$

请回想一下，在我们的量词中，我们关注的是类型为 D 且指标为 $\mathcal{R}$ 的自动机。如果一对集合 (K, L) 被一个自动机（其字母表为 $\{0,1\}^2$）接受，那么它的特征函数也被该自动机接受；而关系 $r \subseteq \mathsf{P}(\{\mathtt{L}, \mathtt{R}\}^*) \times \mathsf{P}(\{\mathtt{L}, \mathtt{R}\}^*)$ 被一个自动机识别，当且仅当它恰好由该自动机所接受的各对组成。现在考虑关系类

$$\mathcal{C}^D_{\mathcal{R}} = \big\{r_K \mid K \in \mathsf{P}(\{\mathtt{L}, \mathtt{R}\}^*) \wedge r \text{ is recognised by an automaton of type } D \text{ and index } \mathcal{R}\big\}.$$

那么不难看出，公式 $\mathtt{I}^D_{\mathcal{R}} X.\,\varphi(W,X)$ 等价于

$$\mathtt{Q}^D_{\mathcal{R}} Z X.\,\varphi(Z,X) \wedge Z = W,$$

其中，量词 $\mathtt{Q}^D_{\mathcal{R}}$ 关于树的语义由类 $\mathcal{C}^D_{\mathcal{R}}$ 给出。

显然，上述变量 Z 只起技术性作用；因此，为了使记号清晰，本文使用记号 $\mathtt{Q}^D_{\mathcal{R}} X.\,\varphi(W,X)$，而不使用 Z。

我们还要指出，我们提出的方法并非唯一可行的方案。还可以考虑一元量词 $\mathtt{Q}^D_{\mathcal{R}} X$：若公式 $\mathtt{Q}^D_{\mathcal{R}} X.\,\varphi(W,X)$ 对赋值 $W \mapsto \tilde{w}$ 成立，当且仅当所有满足 $\varphi(\tilde{w},\tilde{x})$ 的集合 $\tilde{x}$ 所构成的语言，在不读取参数 $\tilde{w}$ 的情况下可由某种（适当类型的）自动机接受，正如注 3.1 中所讨论的那样。也就是说，其语义简单地由所有可被类型为 D 且指标为 $\mathcal{R}$ 的自动机接受的语言所构成的类给出。

尽管这看起来相当自然，但我们认为，这种扩展的意义较小。它不仅会在 ω-字上引入一种不可判定的形式体系（如注 3.1 所示），还会进一步限制相关变量之间可能存在的关联。事实上，如果某个非正则的 $\tilde{w}$ 满足这样的公式，那么根据单子二阶逻辑的一般性质（即正则树定理），就存在一个正则的 $\tilde{w}'$，使得语言 $\{\tilde{x} \mid \varphi(\tilde{w},\tilde{x})\}$ 和 $\{\tilde{x} \mid \varphi(\tilde{w}',\tilde{x})\}$ 相同。因此，从某种意义上说，由公式 $\varphi(W,X)$ 定义的关系必然会弱关联其参数。这些问题仍需进一步研究。

5 ω-字上的博弈量词

我们的结果的第一部分涉及博弈量词。我们首先证明，扩展后的 MSO+ 形式体系可以约化回纯单子二阶逻辑，也就是说，可以消去博弈量词。然而，我们的目标是获得更强的性质，如定理 5.12 所述：在对公式作出适当假设的情况下，由量词描述的博弈存在可由有限记忆转换器实现的策略。

考虑一个博弈量词 $X\text{-}Y.\,\varphi(W, X, Y)$ 的实例，其中内部公式 $\varphi(W, X, Y)$ 属于单子二阶逻辑。

▶ **引理 5.1**（众所周知）。对于任意形如 $X\text{-}Y.\,\varphi(W, X, Y)$ 的公式，其中 φ 属于单子二阶逻辑，都可以有效构造一个与之等价的纯单子二阶逻辑公式。

这一构造见 Kaiser 的一篇著作 [19]。为完整起见，我们给出证明。证明中引入的概念在本文后续部分会派上用场。

证明。 设 $\mathcal{D}$ 是一个以强奇偶指数 $\mathcal{R} = \langle \{i, i{+}1, \ldots\ldots, j\}, P_{i,j} \rangle$ 为指标、字母表为 $A_W \times A_X \times A_Y$ 的确定性奇偶自动机，它等价于 φ；也就是说，当且仅当 $\varphi(\bar{w}, \bar{x}, \bar{y})$ 成立时，该自动机接受一个 ω-字 $\langle \bar{w}, \bar{x}, \bar{y} \rangle$。

给定一个 ω-字 $\bar{w} = w_0 w_1 w_2 \cdots \in (A_W)^\omega$，我们可以考虑奇偶博弈 $\mathcal{G}(\bar{w}, \mathcal{D})$，它是 $\mathcal{G}(\bar{w}, \varphi)$ 与自动机 $\mathcal{D}$ 的笛卡尔积，定义如下。

▶定义 *5.2*。令 Q 为 D 的状态集，δ 为其转移函数。于是，$G(\bar{w}, D)$ 的位置集定义为 $V(I)$ *def*$= \mathbb{N} \times Q$ 和 $V(II)$ *def*$= \mathbb{N} \times Q \times A_X$。从位置 (n, q) 出发，$\in \mathbb{N} \times Q$ 首先由玩家 I 提出 $x_n \in A_X$，博弈转移到位置 (n, q, x_n)。随后玩家 II 提出 $y_n \in A_Y$，博弈转移到位置 $(n{+}1, q')$，其中 $\delta(q, (w_n, x_n, y_n)) = (k_n, q')$。前一条边的标签等于最低优先级 i（即无关紧要），而后一条边的标签等于 k_n。

容易看出，$\mathcal{G}(\bar{w},\mathcal{D})$ 与 $\mathcal{G}(\bar{w},\varphi)$ 是等价的，含义是：某位玩家在一个博弈中获胜，当且仅当她在另一个博弈中获胜：自动机 $\mathcal{D}$ 是确定性的，因此 $\mathcal{G}(\bar{w},\varphi)$ 中的选择与 $\mathcal{G}(\bar{w},\mathcal{D})$ 中的选择之间存在一一对应关系，从而一个博弈中的策略可以直接转移到另一个博弈中。此外，由于奇偶博弈的位置确定性（见定理 2.5），玩家 II 在 $\mathcal{G}(\bar{w},\varphi)$ 中获胜，当且仅当玩家 II 在 $\mathcal{G}(\bar{w},\mathcal{D})$ 中拥有位置获胜策略。

玩家 II 在 $\mathsf{G}(\bar{w},\mathsf{D})$ 中的位置策略 $\sigma^{(\mathrm{II})}$ 可以表示为一个 ω-字 $\bar{\sigma}=\sigma_0\sigma_1\sigma_2\cdots \in \big(Q\times A_X\to A_Y\big)^{\omega}$，其中 Q 是 D 的状态集：在这个 ω-字中，字母 σ_n 满足 $\sigma_n(q,x)=y$ 其中 $\sigma^{(\mathrm{II})}(n,q,x)=\big((n,q,x)\ k,\ (n{+}1,q')\big)$ 且 $\delta_{\mathcal{D}}\big(q,\ (w_n,x,y)\big)=(k,q')$。下述断言显然成立，因为单子二阶逻辑允许我们对 $\mathsf{G}(\bar{w},\mathsf{D})$ 中的无限对局进行量化，并且能够表达奇偶条件 $\mathrm{P}_{i,j}$。

▷ 断言 5.3。存在一个单子二阶逻辑公式 $\psi^{(\mathrm{II})}(W,\Sigma)$，使得 $\psi^{(\mathrm{II})}(\bar{w},\bar{\sigma})$ 对 $\bar{w}\in(A_W)^{\omega}$ 和 $\bar{\sigma}\in\big(Q\times A_X\to A_Y\big)^{\omega}$ 成立，当且仅当 $\bar{\sigma}$ 编码了玩家 II 在 $\mathcal{G}(\bar{w},\mathcal{D})$ 中的位置获胜策略 $\sigma^{(\mathrm{II})}$。

由此可知，公式 $\exists^{\to Y}.\varphi(W,\ X,\ Y)$ 等价于

$$\exists\Sigma\in\big(Q\times A_X\to A_Y\big)^{\omega}.\,\psi^{(\mathrm{II})}(W,\Sigma),$$

其中集合 $\big(Q\times A_X\to A_Y\big)$ 是有限的，因此可以将其视为字母表。于是，该公式属于纯单子二阶逻辑。◀

利用上述引理归纳地消去最内层的博弈量词，我们立即得到以下推论。

▶**推论 5.4.** 单子二阶逻辑$+$ 的表达能力与单子二阶逻辑相同。此外，存在一个有效过程，可以消去博弈量词。

▶ 注5.5. 考虑不含参数 W 的公式，即 $\exists^{\to Y}.\varphi(X,\ Y)$，其中$\varphi$ 属于单子二阶逻辑。在这种情况下，博弈 $\mathcal{G}(\bar{w},\ \mathcal{D})$ 可以在博弈场 Q 上进行，而非在 $\mathbb{N}\times Q$ 上进行。因此，这是一个有限奇偶博弈，可以直接求解。所得策略 $\sigma^{(\mathrm{II})}$具有转导器 τ 的形式：$A_X\twoheadrightarrow A_Y$（其状态集就是 Q），使得对于每个$\bar{x}\in(A_X)^{\omega}$，都有 $\varphi\big(\bar{x},\ \tau(\bar{x})\big)$。

上述注释可视为基于奇偶博弈确定性的布希-兰德韦伯定理证明的现代版本 [11], 。这意味着，所提出的消去博弈量词的过程，可以看作 Büchi 和 Landweber 构造的一种参数化版本；在其中，我们寻找的策略可以依赖于参数$\bar{w}\in(A_W)^{\omega}$。

5.1 序贯策略

人们可能会问，在存在外部参数 W的情况下，是否可以恢复注5.5的某种版本 $\sigma^{(\mathrm{II})}$，即将策略表示为转导器。当然，确切的策略可能取决于 W的全局性质，因此不能期望存在一个能够实现该策略的单一转导器 τ：$A_W\times A_X\twoheadrightarrow A_Y$如果允许转导器依赖于给定的 ω字 $\bar{w}$，又会怎样呢?

▶ **问题5.6**。假设对于某个参数 $\bar{w}\in(A_W)^{\omega}$，公式 $\exists^{\to Y}.\varphi(\bar{w},X,Y)$成立。这是否意味着存在一个转导器 $\tau_{\bar{w}}$：$A_W\times A_X\twoheadrightarrow A_Y$，能够实现玩家 II 在 $\mathcal{G}(\bar{w},\varphi)$中的获胜策略？换句话说，我们要问的是，能否确保

$$\textit{for every } \bar{x}\in(A_X)^{\omega}\ \textit{we have}\ \varphi\big(\bar{w},\bar{x},\tau_{\bar{w}}(\bar{w},\bar{x})\big). \tag{5.1}$$

事实证明，答案是否定的——玩家 II 所使用的策略可能无法被实现为有限记忆策略，即使 $\bar{w}$ 已知。直观地说，这归结于以下事实：$\bar{w}$ 可能不是最终周期的，而 φ 可能需要在 Y 与 W 之间建立某种位置到位置对应关系。更确切地说，我们有如下事实。

▶ **事实 5.7.** 存在一个单子二阶逻辑中的公式 $\varphi(W, X, Y)$，对于某个具体的 ω字$\bar{w} \in (A_W)^\omega$，我们有 $X \to Y.\, \varphi(\bar{w}, X, Y)$，但没有转导器 $\tau\colon (A_W \times A_X) \twoheadrightarrow A_Y$满足公式 (5.1)。

证明。 令 $A_W = A_Y = \{0,1\}$，并令 $\varphi(\bar{w}, \bar{x}, \bar{y})$ 对于 $\bar{w} = w_0 w_1 w_2 \cdots$和 $\bar{y} = y_0 y_1 y_2 \cdots$表示：对每个 $n \in \mathbb{N}$，都有 $y_n = w_{n+1}$。注意，对所有 $\bar{w} \in (A_W)^\omega$，都有$X \to Y.\, \varphi(\bar{w}, X, Y)$，因为 X 在 φ 中不起作用，而且玩家 II 只需依次选择连续的值 $y_0 \stackrel{\text{def}}{=} w_1$, $y_1 \stackrel{\text{def}}{=} w_2$，依此类推。

令 $\bar{w}$ 定义为 $0^1 1 0^2 1 0^3 1 \cdots \in (A_W)^\omega$。接下来要证明，不存在转导器$\tau_{\bar{w}}\colon (A_W \times A_X) \twoheadrightarrow A_Y$满足公式（5.1）。反设 $\tau_{\bar{w}}$ 是这样一个转导器，其状态集为 Q，转移函数为 δ。固定任意一个字母 $x \in A_X$，考虑 τ 在上文定义的 ω-字 $\bar{w}$ 以及 $\bar{x} = xxx \cdots \in (A_X)^\omega$ 上的唯一运行。任取 $n \geq |Q|$，并聚焦于该运行中读取 $\bar{w}$ 的子串 $10^{n+1}1$ 的片段。转导器读取输入片段开头的 n 个零时应当输出 0（因为下一个输入字母是 0），而读取最后一个零时应当输出 1（因为下一个输入字母是 1）。设 τ 在该片段中经过的状态依次为 $\rho_0, \rho_1, \ldots, \rho_{n+1}$，其中 ρ_0 位于第一个 0 之前，ρ_{n+1} 位于最后一个 0 之后。根据鸽巢原理，对于某些 k, ℓ，其中 $0 \leq k < \ell \leq n$，有 $\rho_k = \rho_\ell$。对于 $i \in \{0, \ldots, n-1\}$，我们有 $\delta(\rho_i, (0, x)) = (0, \rho_{i+1})$；将其应用于 k 和 ℓ 之后的相邻位置，可推出 $\rho_{k'} = \rho_n$，其中 $k' = k + (n - \ell) < n$。但这样一来，$(0, \rho_{k'+1}) = \delta(\rho_{k'}, (0, x)) = \delta(\rho_n, (0, x)) = (1, \rho_{n+1})$。换言之，如果零的数量超过其状态数，转导器就无法确定应当在哪个位置输出 1。◀

这个否定答案可以从两个角度来解释。第一种观点由上述例子直接引出，着眼于前瞻的必要性——如果 $\tau_{\bar{w}}$ 能够对参数词 $\bar{w}$的未来进行某种前瞻，就能轻易实现相应的策略。引理 5.9 将这一观察形式化，其中允许进行前瞻。这种方法与 Winter 和 Zimmermann [50], 的研究思路相似，他们研究的是带前瞻的博弈。

另一种观点是，与 Büchi 和 Landweber [11]的构造（另见 [9]）不同，$\mathcal{G}(\bar{w}, \mathcal{D})$ 的博弈场是无限的。因此，变量之间可能会无限期地发生某种微妙的同步。为了避免这个问题，我们考虑一种公式的概念：该公式分别依赖于一个变量（见第 5.4 节）。事实证明，在这种情况下，博弈场的无限性不再构成问题，策略也能再次由转导器实现，正如定理 5.12 所述。

5.2 使用带前瞻的转导器进行一致化

在继续之前，我们首先需要说明，如何通过带前瞻的转导器来实现一致化的关系。我们称单子二阶逻辑公式 $\psi(W, Y)$ 是一致化的：对于每个 $\bar{w}$，至多存在一个 $\bar{y}$ 使得 $\psi(\bar{w}, \bar{y})$ 成立。下一个事实表明，由一致化的单子二阶逻辑公式所描述的部分函数，可以通过与前瞻复合的转导器来实现。这一事实相当一般，几乎已成为常识；它依赖于单子二阶逻辑的复合方法（在我们的框架中由 Wilke 代数表示）。

▶ **事实 5.8**。 假设 $\psi(W,Y)$ 是一致化的。那么，可以有效构造一个同态 $\alpha\colon (A_W)^{\mathsf{W}} \to S$，将其映射到有限 *Wilke* 代数 S，并构造一个转导器 $\tau\colon A_W \twoheadrightarrow (S^{\mathrm{inf}} \to A_Y)$，使得对于每个 $\bar{w} \in (A_W)^{\omega}$，只要 $\exists Y.\, \psi(\bar{w}, Y)$ 成立，就有

$$\psi\big(\bar{w}, \tau(\bar{w}) \bullet \mathsf{lk}_{\alpha}(\bar{w})\big).$$

换言之，对于输入 ω-字 $\bar{w} = w_0 w_1 w_2 \cdots$ 和 $f_0 f_1 f_2 \cdots \stackrel{\mathrm{def}}{=} \tau(\bar{w})$，我们考虑 $h_n \stackrel{\mathrm{def}}{=} \alpha(w_{n+1} w_{n+2} \cdots)$ 和 $y_n \stackrel{\mathrm{def}}{=} f_n(h_n)$，它们定义在 $n = 0, 1, 2, \ldots$ 上，并断言 $\psi(\bar{w}, y_0 y_1 y_2 \cdots)$ 成立。

证明。对于 $n \in \mathbb{N}$，令 $\chi_n \in \{0,1\}^{\omega}$ 表示一个 ω-字，其在位置 n 上的值为 1，其他位置的值均为 0。对于每个 $y \in A_Y$，考虑一个公式 $\psi_y(W,Z)$，使得在假设 $\exists Y.\, \psi(\bar{w}, Y)$ 的情况下，当唯一满足 $\psi(\bar{w}, \bar{y})$ 的 $\bar{y}$ 在位置 n 处的字母等于 y 时，$\psi_y(\bar{w}, \chi_n)$ 成立；这样的公式可以很容易地由 ψ 构造出来。

将定理 2.4 应用于由公式 $\big(\psi_y(W, Z)\big)_{y \in A_Y}$ 定义的语言元组，可得到一个满射到有限 Wilke 代数 S 的同态 $\beta\colon (A_W \times \{0,1\})^{\mathsf{W}} \to S$，以及一个集合元组 $(F_y)_{y \in A_Y}$，使得 $\psi_y(\bar{w}, \bar{z})$ 成立，当且仅当 $\beta\big(\langle \bar{w}, \bar{z} \rangle\big) \in F_y$。

令 $\mathsf{add}_0\colon (A_W)^{\mathsf{W}} \to (A_W \times \{0,1\})^{\mathsf{W}}$ 为一个同态，它将给定字中所有字母的第二坐标加上 0。于是，我们取 $\mathsf{add}_0 \circ \beta$ 作为 $\alpha\colon (A_W)^{\mathsf{W}} \to S$。

接下来，我们构造转导器 $\tau\colon A_W \twoheadrightarrow (S^{\mathrm{inf}} \to A_Y)$。它会记住截至目前所读前缀在 α 下的取值。为此，其状态集是从 S^{fin} 添加一个形式中性元 ε 后得到的幺半群 $S^{\mathrm{fin}}_{+\varepsilon}$。初始状态为 ε。对于 $v \in S^{\mathrm{fin}}_{+\varepsilon}$ 和 $w \in A_W$，令

$$\delta\big(v, w\big) \stackrel{\mathrm{def}}{=} \big(f, v \cdot \alpha(w)\big),$$

其中，$f\colon S^{\mathrm{inf}} \to A_Y$ 对每个 $h \in S^{\mathrm{inf}}$ 定义如下：$f(h)$ 是满足 $v \cdot \beta(w,1) \cdot h \in F_y$ 的任意固定字母 $y \in A_Y$；或者，如果对所有 $y \in A_Y$ 都有 $v \cdot \beta(w,1) \cdot h \notin F_y$，则它只是 A_Y 中的任意元素（从直观上说，应当认为这样的 y 是唯一的；然而严格来说这未必成立，这是由那些满足 $\bar{w}$ 但不满足 $\exists Y.\, \psi(\bar{w}, Y)$ 的字所导致的）。

现在固定一个满足 $\exists Y.\, \psi(\bar{w}, Y)$ 的输入 ω-字 $\bar{w} = w_0 w_1 w_2 \cdots \in (A_W)^{\omega}$。读取前缀 $w_0 w_1 \cdots w_{n-1}$ 后，τ 的状态为 $\alpha(w_0 w_1 \cdots w_{n-1})$（如果 $n = 0$，则仅为 ε）。因此，$\tau(\bar{w}) \bullet \mathsf{lk}_{\alpha}(\bar{w})$ 的第 n 个字母是满足以下条件的字母 y

$$\alpha(w_0 w_1 \cdots w_{n-1}) \cdot \beta(w_n, 1) \cdot \alpha(w_{n+1} w_{n+2} w_{n+3} \cdots) \in F_y.$$

当且仅当 $\psi_y(\bar{w}, \chi_n)$ 成立时，情况正是如此；由于 ψ 是一致化的，因此恰有一个 y 使其成立，也就是唯一满足 $\psi(\bar{w}, \bar{y})$ 的 $\bar{y}$ 在位置 n 上的字母。由此可得，$\psi\big(\bar{w}, \tau(\bar{w}) \bullet \mathsf{lk}_{\alpha}(\bar{w})\big)$ 成立，正如所需。◀

5.3 允许前瞻

利用事实 5.8，我们现在证明，玩家 II 在博弈量词对应的博弈中的获胜策略，可以由一个与前瞻组合的转导器实现。

▶ **引理 5.9**。给定一个公式 $\varphi(W, X, Y)$，可以有效构造一个同态 $\alpha\colon (A_W)^{\mathsf{W}} \to S$，其映射到一个有限 *Wilke* 代数 S，并同时构造一个转导器 $\tau\colon (A_W \times A_X) \twoheadrightarrow (S^{\mathrm{inf}} \to A_Y)$，使得对于每个满足 $X \to Y.\, \varphi(\bar{w}, X, Y)$ 的 ω-字 $\bar{w} \in (A_W)^{\omega}$，以及每个 $\bar{x} \in (A_X)^{\omega}$，我们都有

$$\varphi\big(\bar{w}, \bar{x}, \tau(\bar{w}, \bar{x}) \bullet \mathsf{lk}_{\alpha}(\bar{w})\big).$$

换言之，对于每个 $n \in \mathbb{N}$，如 $f_n \in (S^{\inf} \to A_Y)$，我们取 τ 在读取 $\bar{w}$ 的前缀 $w_0 w_1 \dots w_n$ 和 $\bar{x}$ 的前缀 $x_0 x_1 \dots x_n$ 后产生的输出字母（因此$f_0 f_1 f_2 \cdots = \tau(\bar{w}, \bar{x})$），并考虑 $h_n \stackrel{\text{def}}{=} \alpha(w_{n+1} w_{n+2} w_{n+3} \dots)$ 和 $y_n \stackrel{\text{def}}{=} f_n(h_n)$，那么$\varphi(\bar{w}, \bar{x}, y_0 y_1 y_2 \dots)$ 成立。直观地说，上述引理表明，可以通过一个转导器构造所得的 ω-字 $\bar{y}$，前提是允许对整个ω-字 $\bar{w}$ 进行前瞻（注意，对 $\bar{x}$ 不进行前瞻：不能允许玩家 II 的行动取决于玩家 I 未来的行动）。

该引理本质上是事实 5.8 与以下引理的复合。唯一的技术难点在于，前瞻是在转导器读取完整的输入 ω-字之后给出的。引理 5.9 的完整证明见附录 A。

▶ **引理 5.10** ([23, 41, 43])。对于每个单子二阶逻辑公式 $\psi(W, Y)$，可以构造一个统一化公式 $\psi_{\mathsf{u}}(W, Y)$，使得

- 对于所有 ω-字 $\bar{w}$，都有 $\big(\exists Y.\, \psi(\bar{w}, Y)\big) \Rightarrow \big(\exists Y.\, \psi_{\mathsf{u}}(\bar{w}, Y)\big)$；并
- 且对于所有 ω-字 $\bar{w}, \bar{y}$，都有 $\psi(\bar{w}, \bar{y}) \Leftarrow \psi_{\mathsf{u}}(\bar{w}, \bar{y})$。

值得一提的是，Lifsches 和 Shelah 给出的上述引理原始证明 [23, 定理 6.3] 称：“根据 [1]。”，其中“[1]”指的是 Büchi 和 Landweber 关于合成 [11] 的工作。这是不正确的，因为 Büchi 和 Landweber 展示了如何使用有限状态策略赢得博弈，而一致化一般可能取决于未来。更确切地说，即使 $X \to Y.\, \psi(X, Y)$ 不成立，也总是可以对公式 $\psi(X, Y)$ 进行一致化（设想 $\psi(X, Y)$ 表示：当且仅当 X 中有无限多个字母为 1 时，Y 的第一个字母为 1）。不过，Lifsches 和 Shelah 所犯的错误或许并非偶然：引理 5.9 表明，确实可以使用带前瞻的转导器进行一致化，而稍后证明的定理 5.12 则展示了如何在 $\psi(W, X, Y)$ 分别依赖于所涉及的变量 Y 时消除前瞻（见定义 5.11）。

5.4 分离坐标

事实 5.7 告诉我们，一般而言，当我们希望通过转导器实现获胜策略时，引理 5.9 中的前瞻 lk_α 是必要的。事实 5.7 中的例子只需要前瞻来检查参数 $\bar{w}$ 的下一个位置上的字母。还可以设想另一个例子：为了获胜，玩家 II 应输出 y_n，使其等于 $\{a, b\}$ 中的第一个字母，其中 w_{n+1}，w_{n+2}，w_{n+3}，.....（跳过它之前的所有字母 c）。在这里，由前瞻检查的未来区间需要是无界的，但仍然是有限的。另一方面，前瞻永远不需要检查 $\bar{w}$ 的“整个无限未来”，因为 $\bar{w}$ 是预先已知的。为了理解其中的直觉，假设获胜所需的 y_n 的值取决于字母 a 是否属于集合 $\{w_{n+1}, w_{n+2}, w_{n+3}, \dots\}$。在这里，获胜策略似乎取决于 $\bar{w}$ 的整个未来，但由于我们预先知道 $\bar{w}$，可以避免这一点：要么 $\bar{w}$ 包含无限多个 a（这样未来总会有某个 a），要么最后一个 a 出现在某个位置 n（这样转导器就可以数到这个固定数字 n）。这表明，应当可以创建一个不使用前瞻的转导器，但它会延迟输出 $\bar{y}$ 的字母，以便检查 $\bar{w}$ 的未来性质（形式上，这是无法实现的，因为 $\bar{y}$ 和 $\bar{x}$ 之间还存在同步问题）。

本节的主要结果表明，一旦我们不允许 φ 强制要求 $\bar{w}$ 与 $\bar{y}$之间存在任何位置到位置对应关系，就可以避免前述对前瞻（或延迟）的需求。更确切地说，我们引入以下定义。

▶**定义 5.11**。 如果公式 φ（W、 X、 Y）是公式 ψ_i（W、 X）和公式 γ_i（Y）的有限布尔组合，则称其对 Y分别依赖。

▶**定理 5.12**。 假设 $\varphi(W, X, Y)$ 分别依赖于 Y，且 $\bar{w} \in (A_W)^\omega$满足 $X \nrightarrow Y.\, \varphi(\bar{w}, X, Y)$。那么，存在一个转导器 $\tau_{\bar{w}} \colon A_W \times A_X \twoheadrightarrow A_Y$，使得公式 (5.1) 成立；也就是说，对于每个 $\bar{x} \in (A_X)^\omega$，都有 $\varphi\big(\bar{w}, \bar{x}, \tau_{\bar{w}}(\bar{w}, \bar{x})\big)$。

值得一提的是，如果 $\varphi(W, X, Y)$ 是公式 $\psi_i(W, X)$ 与公式 $\gamma_i(X, Y)$ 的布尔组合，那么仍然可以通过写成 $(\bar{x} \neq \bar{w})$ 或者对所有 $n \in \mathbb{N}$ 都有 $y_n = x_{n+1}$，来还原事实 5.7 中的例子；这种形式符合所需的结构，而且仍然没有转导器能够实现该策略。

由于每个转导器都会诱导一个策略，定理 5.12 实际上给出了一个等价 e.

▶**推论 5.13**。 取一个分别依赖于 Y 和任意 $\bar{w} \in (A_W)^\omega$ 的 $\varphi(W, X, Y)$。那么，当且仅当存在一个转导器 $\tau_{\bar{w}} \colon A_W \times A_X \twoheadrightarrow A_Y$，使得对每个 $\bar{x} \in (A_X)^\omega$，都有 $\varphi\big(\bar{w}, \bar{x}, \tau_{\bar{w}}(\bar{w}, \bar{x})\big)$，$X \nrightarrow Y.\, \varphi(\bar{w}, X, Y)$ 才成立。

定理 5.12 的完整证明见 B 节；这里我们只概述构造过程。首先应用引理 5.9，构造一个用于生成所需获胜策略的转导器。该转导器使用由同态 $\alpha \colon (A_W)^{\mathsf{W}} \to S$ 给出的前瞻，其值属于某个有限 Wilke 代数 S。我们固定参数 $\bar{w}$，并应用拉姆齐定理（参见定理 2.2），将字 $\bar{w}$ 划分为一个有限前缀，后接无限多个子字，使它们在 α 下的像都是同一个幂等元 $e \in S^{\mathrm{fin}}$。在所有划分点，转导器都能确定前瞻（即后缀在 α 下的值）为 $\alpha(e^\omega)$。但是转导器如何检测这些划分点呢？第一个点可以直接硬编码进转导器。随后，在已知某个划分点的情况下，转导器必须找到下一个点。一个大胆的猜想是，每个取值为 e 的子字都会使我们从一个划分点到达下一个划分点；但这并不成立（也许我们应该在另一个取值为 e 的子字之后进行划分）。不过，一个稍强的条件就足够了：如果转导器遇到两个连续的、取值为 e 的子字，那么它就能确定，可以选择第一个子字之后的位置（也就是说，该位置之后的后缀可以划分为无限多个取值为 e 的子字）。我们指出，Thomas 的一项工作中也出现了类似的技术技巧 [45]。

转导器检测到的这些无限多个分割点（可能存在一定延迟）是我们已知前瞻取值的位置，因此可以在这些位置生成实际输出字 ω-word $\bar{y}$ 的片段。由于这些位置以任意方式分散，我们需要能够填充这些位置之间的输出。这正是关于 Y 的依赖相互独立这一假设发挥作用之处：对于适当选取的同态$\beta \colon (A_Y)^{\mathsf{W}} \to T$到另一个有限 Wilke 代数 T，$\varphi(\bar{w}, \bar{x}, \bar{y})$ 的成立分别取决于 $\langle \bar{w}, \bar{x} \rangle$ 和 $\beta(\bar{y})$。现在，借助涉及幂等元的一些标准技巧，我们可以确保以这样的方式填充输出字 ω-word：生成过程中没有延迟（也就是说，我们确实在每一步都从 A_Y 生成一个字母），同时仍能控制 $\beta(\bar{y})$ 的最终取值，确保 $\varphi(\bar{w}, \bar{x}, \bar{y})$ 成立。

6 关于 ω-字的指数算子

现在，我们利用前面的结果来证明指数算子的量词消去过程 s.

▶**定理 6.1**。 单子二阶逻辑 $MSO{+}\mathtt{I}$ 可有效归约为 ω-字上的纯单子二阶逻辑。

首先可以观察到，非确定性指标量词 $\mathtt{I}^{\mathsf{nd}}_{\mathcal{R}}$ 要么是平凡的，要么在适当改变指标后等价于确定性指标量词 $\mathtt{I}^{\mathsf{dt}}_{\mathcal{R}'}$。更准确地说，以下等价关系成立：

- 当 R 为 $\mathsf{P}_{0,0}$ 或 $\mathsf{W}_{0,0}$，时，$\mathtt{I}^{\mathsf{nd}}_{\mathcal{R}} X.\,\varphi(W,X)$ 等价于 $\forall X.\,\varphi(W,X)$，因为这些指标的自动机接受所有 ω-字。$\mathtt{I}^{\mathsf{nd}}_{\mathcal{R}} X.\,\varphi(W,X)$ 等价于 $\forall X.\,\neg\varphi(W,X)$，当 R 为 $\mathsf{P}_{1,1}$ 或 $\mathsf{W}_{1,1}$，时，这是因为这些指标的自动机识别空语言。$\mathtt{I}^{\mathsf{nd}}_{\mathcal{P}_{1,2}} X.\,\varphi(W,X)$ 总是成立，因为非确定性 Büchi 自动机识别所有 ω-字的正则语言 [10]。对于 $\mathsf{P}_{i,j}$，只要 $i \in \{0,1\}$ 且 $j \geq 2$，同样如此。$\mathtt{I}^{\mathsf{nd}}_{\mathcal{P}_{0,1}} X.\,\varphi(W,X)$ 等价于 $\mathtt{I}^{\mathsf{dt}}_{\mathcal{P}_{0,1}} X.\,\varphi(W,X)$，因为非确定性余 Büchi 自动机与确定性余 Büchi 自动机具有相同的表达能力 [29]。当 R 为 $\mathsf{W}_{0,1}$（安全性）、$\mathsf{W}_{1,2}$（可达性）或 $\mathsf{W}_{0,2}$，时，$\mathtt{I}^{\mathsf{nd}}_{\mathcal{R}} X.\,\varphi(W,X)$ 等价于 $\mathtt{I}^{\mathsf{dt}}_{\mathcal{R}} X.\,\varphi(W,X)$；这是因为这些自动机可以在不改变指标的情况下确定化（简单的类似幂集构造便足够）。$\mathtt{I}^{\mathsf{nd}}_{\mathcal{W}_{1,3}} X.\,\varphi(W,X)$ 等价于 $\mathtt{I}^{\mathsf{dt}}_{\mathcal{W}_{0,1}} X.\,\varphi(W,X)$，因为非确定性 $\mathsf{W}_{1,3}$ 自动机与确定性余 Büchi 自动机具有相同的表达能力。对于所有 $\mathsf{W}_{i,j}$，只要 $i \in \{0,1\}$ 且 $j \geq 3$，同样如此。

这涵盖了所有情况，因为我们总可以平移指标，使其成为 $i \in \{0,1\}$。

因此，本节余下部分我们聚焦于确定性指标量词 $\mathtt{I}^{\mathsf{dt}}_{\mathcal{R}}$。与推论 5.4 类似，我们采用归纳法，也就是从内向外消去指数量词。设 $\mathcal{R} = \langle A_{\mathcal{R}}, L_{\mathcal{R}} \rangle$ 是一个指标（对于 $i \leq j$，它可以是 $\mathcal{R} = \mathcal{P}_{i,j}$，也可以是 $\mathcal{R} = \mathcal{W}_{i,j}$）。考虑一个公式 $\mathtt{I}^{\mathsf{dt}}_{\mathcal{R}} X.\,\varphi(W,X)$，其中 $\varphi(W,X)$ 属于单子二阶逻辑。我们的目标是构造一个与 $\mathtt{I}^{\mathsf{dt}}_{\mathcal{R}} X.\,\varphi(W,X)$ 等价的纯单子二阶逻辑公式。与上一节一样，这里我们只考虑单个参数 W 的情况（它可以通过乘积字母表编码多个参数）。

考虑$(A_{\mathcal{R}})^{\omega}$表示对于当且仅当$(\bar{w}, \bar{x}) \in L_{\mathcal{R}}$成立，注意$\varphi_{\mathcal{R}}(W, X, K)$分别依赖于变量 K。

▶ **引理 6.2.** 公式 $\mathtt{I}^{\mathsf{dt}}_{\mathcal{R}} X.\,\varphi(W,X)$ 和 $X \to K.\,\varphi_{\mathcal{R}}(W,X,K)$ 等价。

注意，由于后一个公式中涉及的博弈量词可根据引理 5.1 有效消去，因此这个引理完成了定理 6.1 的证明。

证明。 任取参数 $\bar{w} \in (A_W)^{\omega}$。以下条件等价：

- $\mathtt{I}^{\mathsf{dt}}_{\mathcal{R}} X.\,\varphi(\bar{w},X)$成立；存在一个指标为 R 的确定性自动机 $\mathsf{D}_{\bar{w}}$，使得对于每个 ω-字$\bar{x} \in (A_X)^{\omega}$，$\varphi(\bar{w},\bar{x})$当且仅当$(\bar{w},\bar{x}) \in \mathsf{L}(\mathsf{D}_{\bar{w}})$；存在一个转导器 $\tau_{\bar{w}} \colon (A_W \times A_X) \twoheadrightarrow A_{\mathcal{R}}$，使得对于每个 ω-字 $\bar{x} \in (A_X)^{\omega}$，$\varphi(\bar{w},\bar{x})$当且仅当 $\tau_{\bar{w}}(\bar{w},\bar{x}) \in L_{\mathcal{R}}$；$X \to K.\,\varphi_{\mathcal{R}}(\bar{w},X,K)$成立。

前两项根据指数算子的定义是等价的。第二组两项由于注 2.1 而等价。最后两项根据 $\varphi_{\mathcal{R}}$ 的选择以及推论 5.13 而等价。◀

7 树上的指数算子

本节将证明以下定理。

18 基于 Rabin-Mostowski 指标的广义量词

▶**定理 7.1**。 树上的单子二阶逻辑 $+\mathtt{I}$ 理论是不可判定的。即使我们只允许最简单的指标，即弱奇偶指数 $W_{0,1}$ （也就是安全性），并允许任意类型的确定性 $D \in \{$确定性,非确定性$\}$，这一结论仍然成立，因此我们只使用指数量词 $\mathtt{I}^{D}_{\mathcal{W}_{0,1}}$。

为简化记法，本节只考虑取值于字母表$\{0,1\}$上的变量；因此，它们的赋值可以视为集合，而不是从树节点到$\{0,1\}$的函数。因此，我们可以写作 $\tilde{x} \cup \tilde{y}$、 $\tilde{x} \subseteq \tilde{y}$ 等。

树节点集合 $I \subseteq \{\mathtt{L},\mathtt{R}\}^*$ 称为区间，如果它具有$\{u_{\mathtt{R}}{}^i \mid 0 \le i \le n\}$的形式，其中 u 是某个最顶端节点 （记作$top(I)$），$u_{\mathtt{R}}{}^n$ 是最底端节点，而 $n \in \mathbb{N}$ 是某个长度 （记作$len(I)$）。若两个区间 I_1、 I_2 的最顶端节点 u_1、 u_2 满足 $u_1 \not\preceq u_2$ 和 $u_2 \not\preceq u_1$ （即二者互不为对方的后代），则称它们独立。独立区间的并集是一个集合 $\tilde{z}$，可写作 $\bigcup_{i\in J} I_i$，其中 $(I_i)_{i\in J}$ 是两两独立的区间。注意，这种 $\tilde{z}$ 到区间的分解是唯一的。此外，还可以在单子二阶逻辑中描述这种分解（例如，我们可以用单子二阶逻辑表达“I 是 Z 中的一个区间”等）。

关键的技术贡献是下面这个引理。

▶**引理 7.2**。 在树上的单子二阶逻辑 $+\mathtt{I}^{\mathsf{nd}}_{\mathcal{W}_{0,1}}$ 和单子二阶逻辑 $+\mathtt{I}^{\mathsf{dt}}_{\mathcal{W}_{0,1}}$ 中，可以写出一个公式 $\varphi_{\mathsf{b}}(Z)$，使得对于每个独立区间的并集 $\tilde{z}$，当且仅当 $\varphi_{\mathsf{b}}(\tilde{z})$ 成立时，$\tilde{z}$ 中的区间长度有某个 $n \in \mathbb{N}$ 作为上界。

证明略述。 我们取 $\varphi_{\mathsf{b}}(Z)$

$$\forall W \subseteq Z.\ \forall Y \subseteq Z.\ \mathtt{I}^{D}_{\mathcal{W}_{0,1}} X.\ \underbrace{\big(X \subseteq W \wedge \forall x \in X.\, \exists y \in Y.\, x \preceq y\big)}_{\psi(W,Y,X)},$$

其中 D 或者是 dt，或者是 nd（即指数量词针对确定性或非确定性自动机——两者都可以）。公式 $\varphi_{\mathsf{b}}(Z)$ 给定集合 $\tilde{z}$，表达的是：对于子集 $\tilde{w}, \tilde{y} \subseteq \tilde{z}$ 的每一种选择，都存在一个安全自动机 $\mathcal{A}$，用于检查给定的 $\tilde{w}$ 的子集 $\tilde{x}$ 是否只包含拥有 $\tilde{y}$中后代的点。

注意，子公式 $\psi(W,Y,X)$ 不依赖于 $\tilde{z}$，因此假设存在的安全自动机 $\mathcal{A}$ 无法访问 $\tilde{z}$，尽管变量 Z 在子公式 $\psi(W,Y,X)$的作用域内形式上可用（可以通过在 $\mathtt{I}^{D}_{\mathcal{W}_{0,1}}$前面人为地用另一个量词 $\exists Z.$遮蔽变量 Z，确保 $\tilde{z}$ 在$\psi(W,Y,X)$内部不可见）。

首先，假设某个集合 $\tilde{z}$ 中的区间长度均由某个 $n \in \mathbb{N}$ 界定，并任取一个 $\tilde{w}$，$\tilde{y} \subseteq \tilde{z}$。$\tilde{y}$ 中能够成为 $\tilde{w}$ 中元素后代的唯一元素，是同一区间中的元素，因此它们最多位于其下方 n 层。于是，所讨论的性质可由以下确定性安全自动机识别：在 $\tilde{x}$ 的每个元素之后（检查其是否属于 $\tilde{w}$），沿着只向右的分支等待 n 层；如果未找到 $\tilde{y}$ 中的元素，则拒绝。这表明 $\varphi_{\mathsf{b}}(\tilde{z})$ 成立（无论在其定义中使用 $\mathtt{I}^{\mathsf{nd}}_{\mathcal{W}_{0,1}}$ 还是 $\mathtt{I}^{\mathsf{dt}}_{\mathcal{W}_{0,1}}$）。注意，所构造自动机的规模取决于 n。

现在假设 $\tilde{z}$ 中的区间长度无界。我们想证明 $\varphi_{\mathsf{b}}(\tilde{z})$ 不成立。我们选择 $\tilde{w}$ 为 $\tilde{z}$ 中所有区间的最顶端点。集合 $\tilde{y}$ 包含从 $\tilde{z}$ 中精心挑选的区间的最底端点。除了其他性质外，所选区间的长度还需要不断增长。首先，整个构造是以对角线方式完成的，因为假设中的安全自动机 $\mathcal{A}$ 事先未知，因此我们依次考察所有这类自动机，确保每个自动机都不能成为使 $\mathtt{I}^{D}_{\mathcal{W}_{0,1}} X.\ \psi(\tilde{w}, \tilde{y}, X)$ 成立的良好见证。

对于固定的自动机 $\mathcal{A}$，如果区间长度无界，那么某个区间的长度就会超过自动机$\mathcal{A}$的计数能力。因此，某个最底端点 y 属于 $\tilde{y}$ 的区间必须足够长，使得自动机 $\mathcal{A}$ 能够沿着该区间出现包含两个重复状态的抽取模式。这样就可以无限重复该模式（即抽取），从而有效地将节点 y 从 $\tilde{y}$ 中移除，使公式$\psi(\tilde{w},\tilde{y},\tilde{x})$ 为假，而假设中的自动机仍然接受 $\langle \tilde{w},\tilde{y},\tilde{x}\rangle$。然而，我们不能更改参数集合 $\tilde{y}$，因此实际的抽取必须在 $\tilde{y}$ 的另一部分上进行，这就需要应用类似 Ramsey 的论证，以便将抽取位置移到所考虑集合 $\tilde{y}$ 之外（这与 Carayol 和 Löding [12] 的抽取方案有些相似）。

这一蕴含关系的完整证明见附录 C.3 节。 ◀

既然我们知道，独立区间集合的有界性可以用单子二阶逻辑 MSO$^{+\mathtt{I}^{D}_{\mathcal{W}_{0,1}}}$表达，那么接下来只需调整 Bojańczyk 等人 [4]的技术构造，以便用区间集合的有界性来表达 Minsky 机的运行。这是一项相当标准的调整；更多细节请参见附录 C。

8 结论

本文的动机是引入并研究指数量词 $\mathtt{I}$，这种量词试图将指标问题纳入逻辑的语法之中。从这一角度来看，定理 7.1 所传达的一条重要信息是：逻辑 MSO+$\mathtt{I}$ 在树上是不可判定的，即使采用最简单形式的量词——确定性安全指数量词。一个正则树语言能否由指标为 $\mathcal{R}$ 的自动机识别，等价于判断公式 $\mathtt{I}^{D}_{\mathcal{R}}X.\,\varphi(X)$ 是否成立，其中指数量词只使用一次，且不允许使用参数。尽管这一片段仍可能是可判定的，但定理 7.1 表明，两者的边界已经非常接近；据我们所知，这是首个与指标问题相关的不可判定性结果。

另一方面，我们对 ω-字上的指数算子 $\mathtt{I}$ 的研究则带来了更乐观的看法。我们提出了一种有效的指标量词消去过程，从而将单子二阶逻辑+$\mathtt{I}$ 化为纯单子二阶逻辑。尽管指数算子在验证和模型检测领域是否有用尚存疑问，但我们的方法提供了一种新颖且通用的、基于博弈的证明方式，可用于证明 ω-字上的指标问题是可判定的 [47]。这与 Löding [25] 的思路相似。我们认为，这方面研究最重要的成果，是对博弈量词 进行了全面分析——我们通过先将 $\mathtt{I}$ 转换为博弈量词，再消去这些量词，来完成量词消去的证明。

博弈量词自20世纪70年代以来便已为人所知并得到研究，但其应用大多局限于描述集合论。我们以类似 Kaiser [19]，的方式，明确引入博弈量词的概念，并研究单子二阶逻辑 + 的表达能力。对于 而言，量词消去过程相当直接，类似于 Büchi 和 Landweber 的构造的参数化版本（见注5.5）。与指数算子不同，我们认为，在单子二阶逻辑中直接使用博弈量词，或许有助于表达某些与博弈相关的性质。

为了将我们的量词消去过程应用于指数算子，我们需要建立一套关于在参数化ω-正则博弈中实现获胜策略的转换器理论。为此，我们证明了一个新颖且似乎相当强的结果（见定理 5.12）：如果相关变量在某种意义上彼此分离，那么只要博弈量词成立，就可以用有限记忆转换器见证相应策略的存在。

参考文献

1 Vince Bárány、Łukasz Kaiser 和 Alexander Rabinovich。树上的单子二阶逻辑中表达基数数量词。《Fundam. Informaticae》，100(1-4):1–17，2010。 doi:10.3233/FI-2010-260。 2 Mikołaj Bojańczyk。有界量词。载于 Jerzy Marcinkowski 和 Andrzej Tarlecki（编），《计算机科学逻辑》，第18届国际研讨会 CSL 2004、EACSL第13届年会，波兰卡尔帕奇，2004年9月20日至24日，论文集，第3210卷《计算机科学讲义》，第41–55页。施普林格，2004。 doi:10.1007/978-3-540-30124-0_7。 3 Mikołaj Bojańczyk、Laure Daviaud、Bruno Guillon、Vincent Penelle 和 A. V. Sreejith。MSO+U弱版本的不可判定性。《逻辑方法与计算机科学》，16(1)，2020。doi:10.23638/LMCS-16(1:12)2020。 4 Mikołaj Bojańczyk、Paweł Parys 和 Szymon Toruńczyk。自然数集（N,<）上的 MSO+U 理论不可判定。载于 Nicolas Ollinger 和 Heribert Vollmer（编），《计算机科学理论方面》第33届研讨会，STACS 2016，2016年2月17日至20日，法国奥尔良，第47卷 LIPIcs，第21:1–21:8页。达格斯图尔城堡—莱布尼茨信息学中心，2016。 doi:10.4230/LIPICS.STACS.2016.21。 5 Julian C. Bradfield。模态 μ-演算的交替层级是严格的。《理论计算机科学》，195(2):133–153，1998。 doi:10.1016/S0304-3975(97)00217-X。 6 Julian C. Bradfield。简化模态 μ 演算的交替层级。载于 Michel Morvan 、 Christoph Meinel 和 Daniel Krob （编），STACS 98，计算机科学理论方面第 15 届年度研讨会，法国巴黎，1998 年 2 月25日至27日，论文集，第1373卷《计算机科学讲义》，第39–49页。施普林格，1998。 doi:10.1007/BFB0028547。 7 Julian C. Bradfield。不动点、博弈与差分层级。《RAIRO 理论信息学及其应用》，37(1):1–15，2003。 doi:10.1051/ITA:2003011。 8 Julian C. Bradfield、Jacques Duparc 和 Sandra Quickert。μ演算的超限扩展。载于 C.-H. Luke Ong（编），《计算机科学逻辑》，第19届国际研讨会 CSL 2005、EACSL第14届年会，英国牛津，2005年8月22日至25日，论文集，第3634卷《计算机科学讲义》，第384–396页。施普林格，2005。 doi:10.1007/11538363_27。 9 Benedikt Brütsch 和 Wolfgang Thomas。求解贝尔空间中的无限博弈。《Fundam. Informaticae》，186(1-4):63–88，2022。 doi:10.3233/FI-222119。 10 J. Richard Büchi。受限二阶算术中的一种判定方法。载于《1960年逻辑、方法论与科学哲学国际会议论文集》，第1–11页，1962。 11 J. Richard Büchi 和 Lawrence H. Landweber。用有限状态策略求解序列条件。《Trans. Amer. Math. Soc》，138:295–311，1969。doi:10.1090/S0002-9947-1969-0280205-0。 12 Arnaud Carayol 和 Christof Löding。无限二叉树上的 MSO：选择与序。载于 Jacques Duparc 和 Thomas A. Henzinger（编），《计算机科学逻辑》，第21届国际研讨会 CSL 2007、EACSL第16届年会，瑞士洛桑，2007年9月11日至15日，论文集，第4646卷《计算机科学讲义》，第161–176页。施普林格，2007。 doi:10.1007/978-3-540-74915-8_15。 13 Lorenzo Clemente、Paweł Parys、Sylvain Salvati 和 Igor Walukiewicz。高阶递归方案的对角问题是可判定的。载于 Martin Grohe、Eric Koskinen 和 Natarajan Shankar（编），第31届 ACM/IEEE 计算机科学逻辑研讨会论文集，LICS '16，美国纽约，2016年7月5日至8日，第96–105页。ACM，2016。 doi:10.1145/2933575.2934527。 14 Thomas Colcombet、Denis Kuperberg、Christof Löding 和 Michael Vanden Boom。判定 Büchi 可定义树语言的弱可定义性。载于 Simona Ronchi Della Rocca（编），《2013年计算机科学逻辑会议》（CSL 2013），CSL 2013，意大利都灵，2013年9月2日至5日，第23卷 LIPIcs，第215–230页。达格斯图尔城堡—莱布尼茨信息学中心，2013。doi:10.4230/LIPICS.CSL.2013.215。

15 Thomas Colcombet 和 Christof Löding。非确定性 Mostowski 层级与距离奇偶自动机。载于 Luca Aceto、Ivan Damgård、Leslie Ann Goldberg、Magnús M. Halldórsson、Anna Ingólfsdóttir 和 Igor Walukiewicz（编），《自动机、语言与编程》，第35届国际研讨会，ICALP 2008，冰岛雷克雅未克，2008年7月7日至11日，会议论文集，第二部分——专题B：逻辑、语义与程序设计理论及专题C：安全与密码学基础，《计算机科学讲义》第5126卷，第398–409页。施普林格，2008。 doi:10.1007/978-3-540-70583-3_33。16 E. Allen Emerson 和 Charanjit S. Jutla。树自动机、μ演算与确定性。载于第32届计算机科学基础年度研讨会，波多黎各圣胡安，1991年10月1日至4日，第368–377页。IEEE 计算机学会，1991。 doi:10.1109/SFCS.1991.185392。17 Alessandro Facchini、Filip Murlak 和 Michał Skrzypczak。Rabin-Mostowski 指数问题：迈向确定性自动机之外。载于第28届 ACM/IEEE 计算机科学逻辑研讨会，LICS 2013，美国路易斯安那州新奥尔良，2013年6月25日至28日，第499–508页。IEEE 计算机学会，2013。 doi:10.1109/LICS.2013.56。18 Olivier Finkel、Dominique Lecomte 和 Pierre Simonnet。可识别树语言复杂性的上界。《RAIRO 理论信息学及其应用》，49(2):121–137，2015。 doi:10.1051/ITA/2015002。19 Łukasz Kaiser。自动结构上的博弈量词化与分层模型检验博弈。载于 Zoltán Ésik（编），《计算机科学逻辑》，第20届国际研讨会暨第15届 EACSL 年会，CSL 2006，匈牙利塞格德，2006年9月25日至29日，会议论文集，《计算机科学讲义》第4207卷，第411–425页。施普林格，2006。 doi:10.1007/11874683_27。20 Łukasz Kaiser。《自动结构上的逻辑与博弈——使用量词与分解进行博弈》，《计算机科学讲义》第6810卷。施普林格，2011。 doi:10.1007/978-3-642-22807-0。21 Vladimir G. Kanovei。Kolmogorov 在集合运算理论中的思想。《俄罗斯数学调查》，43(6):111–155，1988。 doi:10.1070/RM1988v043n06ABEH001995。22 Alexander Kechris。《经典描述集合论》。Springer-Verlag，纽约，1995。 doi:10.1007/978-1-4612-4190-4。23 Shmuel Lifsches 和 Saharon Shelah。树类中的一致化与 Skolem 函数。《符号逻辑杂志》，63(1):103–127，1998。 doi:10.2307/2586591。24 Christof Löding。无限树上的逻辑与自动机。教授资格论文，亚琛工业大学，2009。25 Christof Löding。无限字上确定性下推自动机的判定问题。载于 Zoltán Ésik 和 Zoltán Fülöp（编），第14届自动机与形式语言国际会议论文集，AFL 2014，匈牙利塞格德，2014年5月27日至29日，第151卷 EPTCS，第55–73页，2014。 doi:10.4204/EPTCS.151.4。26 Donald A. Martin。Borel 确定性。《数学年刊》，102(2):363–371，1975。 doi:10.2307/1971035。27 Robert McNaughton。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。 doi:10.1016/S0019-9958(66)80013-X。28 Henryk Michalewski 和 Damian Niwiński。正则树语言的拓扑完备性。载于 Robert L. Constable 和 Alexandra Silva（编），《逻辑与程序语义——献给 Dexter Kozen 60岁生日的论文集》，《计算机科学讲义》第7230卷，第165–179页。施普林格，2012。 doi:10.1007/978-3-642-29485-3_11。29 Satoru Miyano 和 Takeshi Hayashi。ω词上的交替有限自动机。《理论计算机科学》，32:321–330，1984。 doi:10.1016/0304-3975(84)90049-5。30 Yiannis N. Moschovakis。《抽象结构上的初等归纳》。North-Holland 出版公司，1974。31 Andrzej Mostowski。关于量词的一种推广。《数学基础》，44(1):12–36，1957。 doi:10.4064/fm-44-1-12-36。

22 基于 Rabin-Mostowski 指数的广义量词

32 Andrzej W. Mostowski。禁位博弈。技术报告，格但斯克大学，1991。33 Damian Niwiński。关于不动点克隆。载于 Laurent Kott（编），《自动机、语言与编程》，第13届国际研讨会，ICALP86，法国雷恩，1986年7月15日至19日，会议论文集，《计算机科学讲义》第226卷，第464–473页。施普林格，1986。doi:10.1007/3-540-16761-7_96。34 Damian Niwiński。有限自动机可识别的无限树集合的基数。载于 Andrzej Tarlecki（编），《1991年计算机科学数学基础》，第16届国际研讨会，MFCS'91，波兰卡齐米日多尔尼，1991年9月9日至13日，会议论文集，《计算机科学讲义》第520卷，第367–376页。施普林格，1991。doi:10.1007/3-540-54345-7_80。35 Damian Niwiński 和 Igor Walukiewicz。确定性树语言的间隙性质。《理论计算机科学》，1(303):215–231，2003。 doi:10.1016/S0304-3975(02)00452-8。36 Damian Niwiński 和 Igor Walukiewicz。判定确定性树自动机的非确定性层级。载于 Ruy J. G. B. de Queiroz 和 Patrick Cégielski（编），第11届逻辑、语言、信息与计算研讨会论文集，WoLLIC 2004，法国枫丹白露，2004年7月19日至22日，《理论计算机科学电子札记》第123卷，第195–208页。爱思唯尔，2004。 doi:10.1016/J.ENTCS.2004.05.015。37 Damian Niwiński、Paweł Parys 和 Michał Skrzypczak。单子二阶逻辑中序数秩的二分定理。载于 Olaf Beyersdorff、Michal Pilipczuk、Elaine Pimentel 和 Kim Thang Nguyen（编），第42届计算机科学理论方面国际研讨会，STACS 2025，德国耶拿，2025年3月4日至7日，LIPIcs 第327卷，第69:1–69:18页。达格斯图尔城堡—莱布尼茨信息学中心，2025。 doi:10.4230/LIPICS.STACS.2025.69。38 Dominique Perrin 和 Jean-Éric Pin。《无限字：自动机、半群、逻辑与博弈》。纯粹与应用数学。爱思唯尔摩根考夫曼，2004。39 Michael O. Rabin。二阶理论与无限树上自动机的可判定性。《美国数学会汇刊》，141:1–35，1969。 doi:10.1090/S0002-9947-1969-0246760-1。40 Michael O. Rabin。《无限对象上的自动机与丘奇问题》。美国数学会，美国马萨诸塞州波士顿，1972。41 Alexander Rabinovich。带单子谓词扩展的自然数集上序的单子逻辑的可判定性。《信息与计算》，205(6):870–889，2007。 doi:10.1016/J.IC.2006.12.004。42 Saharon Shelah。序的单子理论。《数学年刊》，102(3):379–419，1975。doi:10.2307/1971037。43 Dirk Siefkes。算术某些单子二阶片段中的递归集。《数学与逻辑文献档案》，17(1–2):71–80，1975。 doi:10.1007/BF02280817。44 Michał Skrzypczak 和 Igor Walukiewicz。判定布奇语言的拓扑复杂性。载于 Ioannis Chatzigiannakis、Michael Mitzenmacher、Yuval Rabani 和 Davide Sangiorgi（编），第43届自动机、语言与编程国际研讨会，ICALP 2016，意大利罗马，2016年7月11日至15日，LIPIcs 第55卷，第99:1–99:13页。达格斯图尔城堡—莱布尼茨信息学中心，2016。 doi:10.4230/LIPICS.ICALP.2016.99。45 Wolfgang Thomas。ω序列的无星正则集。《信息与控制》，42(2):148–156，1979。 doi:10.1016/S0019-9958(79)90629-6。46 William Wadge。《贝尔空间中的可约性与确定性》。博士论文，加州大学伯克利分校，1983。47 Klaus W. Wagner。关于ω正则集。《信息与控制》，43(2):123–177，1979。 doi:10.1016/S0019-9958(79)90653-3。48 Dag Westerståhl。广义量词。载于 Edward N. Zalta 和 Uri Nodelman（编），《斯坦福哲学百科全书》。形而上学研究实验室，斯坦福大学，2024年秋季版，2024。URL： https://plato.stanford.edu/archives/fall2024/entries/generalized-quantifiers/。

49 Thomas Wilke. 有限字与无限字正则语言的代数理论。 代数与计算国际期刊，3(4):447–490，1993。 doi:10.1142/S0218196793000287。50 Sarah Winter 和 Martin Zimmermann。延迟博弈中的有限状态策略。 信息与计算，272:104500，2020。 doi:10.1016/J.IC.2019.104500。

引理 5.9 的证明

▶**引理 5.9**。 给定一个公式 $\varphi(W, X, Y)$，可以有效构造一个同态$\alpha\colon (A_W)^{\mathsf{W}} \to S$，映射到有限 *Wilke* 代数 S，以及一个转导器 $\tau\colon (A_W \times A_X) \twoheadrightarrow (S^{\mathrm{inf}} \to A_Y)$，使得对于每个满足 $X \dashv\!\vdash Y.\, \varphi(\bar{w}, X, Y)$ 的 ω 字 $\bar{w} \in (A_W)^\omega$ *s*，以及每个 $\bar{x} \in (A_X)^\omega$，我们有

$$\varphi\big(\bar{w}, \bar{x}, \tau(\bar{w}, \bar{x}) \bullet \mathsf{lk}_\alpha(\bar{w})\big).$$

证明。回顾定义 5.2 中的博弈 G($\bar{w}$, D)，其中 D 是一个与 φ 等价的确定性奇偶自动机。与前文一样，令 Q 为 D 的状态集。取断言 5.3 中的公式$\psi^{(\mathrm{II})}(W, \Sigma)$，该公式满足：如果 $\bar{\sigma} \in (Q \times A_X \to A_Y)^\omega$编码了玩家 II 在 G($\bar{w}$, D) 中的位置获胜策略，则 $\psi^{(\mathrm{II})}(\bar{w}, \bar{\sigma})$成立。利用引理 5.10，将 $\psi^{(\mathrm{II})}(W, \Sigma)$改写为一致化公式 $\psi_{\mathsf{u}}^{(\mathrm{II})}(W, \Sigma)$。对于每个满足 $X \dashv\!\vdash Y.\, \varphi(\bar{w}, X, Y)$的 ω-字 $\bar{w} \in (A_W)^\omega$，都存在这样的位置获胜策略，因此恰有一个 ω-字 $\bar{\sigma} \in (Q \times A_X \to A_Y)^\omega$ 使得 $\psi_{\mathsf{u}}^{(\mathrm{II})}(\bar{w}, \bar{\sigma})$成立，并且这个 ω-字编码了玩家 II 在 G($\bar{w}$, D) 中的位置获胜策略。将事实 5.8 应用于 $\psi_{\mathsf{u}}^{(\mathrm{II})}$，我们得到一个同态 $\alpha\colon (A_W)^{\mathsf{W}} \to S$ 到有限 Wilke 代数 S 的满射，以及一个转导器

$$\theta\colon A_W \twoheadrightarrow \big(S^{\mathrm{inf}} \to (Q \times A_X \to A_Y)\big)$$

使得对于每个满足 $X \dashv\!\vdash Y.\, \varphi(\bar{w},\ X,\ Y)$ 的 ω-字 $\bar{w} \in (A_W)^\omega$，结果 $\theta(\bar{w}) \bullet \mathsf{lk}_\alpha(\bar{w})$ 编码了 $\mathcal{G}(\bar{w},\ \mathcal{D})$ 中唯一的获胜策略。

同态 α 已经满足要求；它可以作为引理陈述中的同态 α。另一方面，转导器还需要改进。问题在于，θ 只读取 $\bar{w}$，并输出一个取决于 $\mathcal{D}$ 的状态和 $\bar{x}$ 当前字母的策略；然而，τ 应同时读取 $\bar{w}$ 和 $\bar{x}$，并且只生成 $\bar{y}$ 中实际对局的字母。因此，我们为 θ 增添一个部分，用来根据迄今读取的 $\bar{w}$ 和 $\bar{x}$ 的部分，以及 θ 自身生成的 $\bar{y}$ 的前面字母，追踪 $\mathcal{D}$ 的当前状态。注意，$\bar{y}$ 生成的字母，因而 $\mathcal{D}$ 的当前状态，取决于同态 α 的前瞻结果；该前瞻会检查 $\bar{w}$ 的未来部分。不过，同态在过去各处的取值可以从当前取值恢复。形式上，我们按如下方式构造新的转导器 $\tau\colon (A_W \times A_X) \twoheadrightarrow (S^{\mathrm{inf}} \to A_Y)$。

假设

$$\theta = \big\langle A_W, \big(S^{\mathrm{inf}} \to (Q \times A_X \to A_Y)\big), P, \iota_\theta, \delta_\theta \big\rangle,$$
$$\mathcal{D} = \big\langle (A_W \times A_X \times A_Y), \{i, i{+}1, \ldots, j\}, Q, \iota_{\mathcal{D}}, \delta_{\mathcal{D}} \big\rangle.$$

我们将 τ 的状态集取为笛卡尔积 $P \times (S^{\mathrm{inf}} \to Q)$。$\tau$ 的初始状态为 $(\iota_\theta, r_{\mathrm{I}})$，其中 $r_{\mathrm{I}}(h)$ 对所有 $h \in S^{\mathrm{inf}}$ 均成立。为了定义 τ 的转移函数 δ_τ，考虑一个状态 $(p, r) \in P \times (S^{\mathrm{inf}} \to Q)$ 和一个输入字母 $(w, x) \in A_W \times A_X$。首先，定义 $(\ell, p') \text{ 定义为 } = \delta_\theta(p, w)$；回想一下，$\ell\colon S^{\mathrm{inf}} \to (Q \times A_X \to A_Y)$ 是由 θ 产生的一个输出字母。接下来，对每个 $h' \in S^{\mathrm{inf}}$ 定义（如图 A.1 所示）

- $h \text{ 定义} = \alpha(w) \cdot h'$,
- $q \overset{\mathrm{def}}{=} r(h), y \text{ 定义} = \ell(h')$
- $(q, x), q' \text{ 定义} = \delta_{\mathcal{D}}\big(q, ($
- $w, x, y)\big), f(h') \text{ 定义 } = y,$
- $r'(h') \overset{\mathrm{def}}{=} q'$。
-

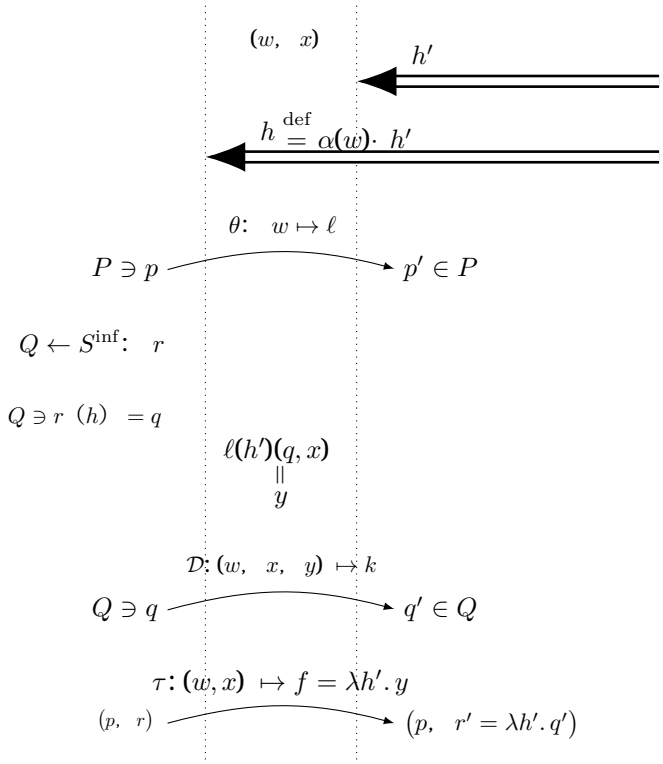

图 A.1 引理 5.9 证明中信息流的示意图。我们用记号 $q \overset{\tau\colon\ w\mapsto y}{\longrightarrow} q'$ 表示转导器 τ 有一个转移 $\delta(q,\ w) = (y,\ q')$。由 $\mathrm{lk}_\alpha(\bar{w})$ 给出的 ω-字 $\bar{w}$ 的后缀类型用双箭头表示。

有了这一切，我们取

$$\delta_\tau\big((p,r),(w,x)\big) = \big(f,(p',r')\big).$$

现在考虑输入字 $\bar{w} = w_0w_1w_2\cdots \in (A_W)^\omega$ 和 $\bar{x} = x_0x_1x_2\cdots \in (A_X)^\omega$。令 $\bar{\ell} = \ell_0\ell_1\ell_2\cdots$ def $= \theta(\bar{w})$ 为 θ 的输出，并令 $\bar{f} = f_0f_1f_2\cdots$ def $= \tau(\bar{w},\bar{x})$ 为 τ 的输出。此外，对于每个 $n \in \mathbb{N}$，令 h_n def= $\alpha(w_{n+1}w_{n+2}w_{n+3}\dots)$，令 y_n def= $f_n(h_n)$，并令 (p_n,r_n) 为 τ 读取字 $(w_0,x_0)(w_1,x_1)\cdots(w_{n-1},x_{n-1})$ 后的状态。首先，我们通过对 $n \in \mathbb{N}$ 进行归纳法观察到，每个 p_n 都是 θ 读取字 $w_0w_1\cdots w_{n-1}$ 后的状态；这显然成立，因为 p_{n+1} 被定义为 $\delta_\theta(p_n,w_n)$ 的第二坐标。

接下来，我们通过对 $n \in \mathbb{N}$ 作归纳法，观察到 $q_n = r_n(\alpha(w_n)\cdot h_n)$，并且 $y_n = \ell_n(h_n)(q_n,x_n)$。事实上，显然 $q_0 = \iota_\mathcal{D} = r_\mathrm{I}(\alpha(w_0)\cdot h_0) = r_0(\alpha(w_0)\cdot h_0)$。接着，假设对于某个 $n \in \mathbb{N}$，有 $q_n = r_n(\alpha(w_n)\cdot h_n)$。我们有

$$y_n = f_n(h_n) = \ell_n(h_n)(r_n(\alpha(w_n)\cdot h_n),x_n) = \ell_n(h_n)(q_n,x_n),$$

其中，第一个等式根据 y_n 的定义成立，第二个等式根据 δ_τ 的定义成立，因为 f_n 是从状态 (p_n,r_n) 读取 (w_n,x_n) 时由 τ 产生的输出，第三个等式则由归纳法假设成立。由此，我们得到

$$\begin{aligned} q_{n+1} &= \delta_\mathcal{D}(q_n,(w_n,x_n,y_n)) = \delta_\mathcal{D}(q_n,(w_n,x_n,\ell_n(h_n)(q_n,x_n))) \\ &= r_{n+1}(h_n) = r_{n+1}(\alpha(w_{n+1})\cdot h_{n+1}), \end{aligned}$$

其中，第一个等式根据 q_{n+1} 的定义成立，第二个等式根据此前关于 y_n 所示的等式成立，第三个等式根据 δ_τ 的定义成立，因为 r_{n+1} 是从状态 (p_n, r_n) 读取 (w_n, x_n) 后到达的 τ 的状态的第二坐标，最后一个等式根据 h_n 的定义以及 α 是同态这一事实成立。

最后，对所有 $n \in \mathbb{N}$ 都成立的等式 $y_n = \ell_n(h_n)(q_n, x_n)$ 表明：如果在 $\mathcal{G}(\bar{w}, \mathcal{D})$ 中，玩家 I 进行 ω-字对局 $\bar{x}$，那么根据由 $\bar{\sigma} = \theta(\bar{w}) \bullet \mathrm{lk}_\alpha(\bar{w})$ 编码的策略，玩家 II 会以 ω-字 $\bar{y} \overset{\text{def}}{=} y_0 y_1 y_2 \cdots = \tau(\bar{w}, \bar{x}) \bullet \mathrm{lk}_\alpha(\bar{w})$ 作出回应。上述策略是获胜策略，因此 $\varphi(\bar{w}, \bar{x}, \bar{y})$ 成立。◀

B 定理 5.12 的证明

▶ **定理 5.12**。假设 $\varphi(W, X, Y)$ 分别依赖于 Y，且 $\bar{w} \in (A_W)^\omega$ 满足 $X \twoheadrightarrow Y.\, \varphi(\bar{w}, X, Y)$ 成立。那么，存在一个转导器 $\tau_{\bar{w}}\colon A_W \times A_X \twoheadrightarrow A_Y$，使得公式 (5.1) 成立；也就是说，对每个 $\bar{x} \in (A_X)^\omega$，都有 $\varphi\big(\bar{w}, \bar{x}, \tau_{\bar{w}}(\bar{w}, \bar{x})\big)$。

在整个证明中，假设 $\bar{w} \in (A_W)^\omega$ 是固定的，并且满足 $X \twoheadrightarrow Y.\, \varphi(\bar{w}, X, Y)$。

根据定理 2.4，设 $\beta\colon (A_Y)^{\mathsf{W}} \to T$ 是一个满射到有限 Wilke 代数 T 的同态，它识别公式 $\varphi(W, X, Y)$ 中的所有语言 $\gamma_i(Y)$。于是，只要 $\beta(\bar{y}) = \beta(\bar{y}')$，就有 $\varphi(\bar{w}, \bar{x}, \bar{y}) \Leftrightarrow \varphi(\bar{w}, \bar{x}, \bar{y}')$。

回忆一下，通过向 T_{fin} 添加形式中性元 ε 得到的幺半群 $T^{\mathrm{fin}}_{+\varepsilon}$。作为构造 $\tau_{\bar{w}}$ 的第一步，我们在引理 B.1 中构造一个转导器 $\theta_{\bar{w}}$，它输出 $T^{\mathrm{fin}}_{+\varepsilon}$ 中的值（这些值旨在表示同态 β 下 $\bar{y}$ 的子串的取值），而不是具体的字母 $\bar{y}$。这一改变使得实现前述延迟更容易。随后，我们在引理 B.3 和 B.4 中说明如何将 $\theta_{\bar{w}}$ 改造为所需的转导器 $\tau_{\bar{w}}$。

对于饱和的 ω-字 $\bar{t} \in (T^{\mathrm{fin}}_{+\varepsilon})^\omega$，我们写作 $\varphi(\bar{w}, \bar{x}, \bar{t})$，表示任意满足 $\varphi(\bar{w}, \bar{x}, \bar{y})$ 的 ω-字 $\bar{y} \in (A_Y)^\omega$，其中 $\beta(\bar{y}) = \odot(\bar{t})$。此外，对于 $r \in \mathbb{N}$，我们称这样的 $\bar{t}$ 是 r-完全的，如果 $\bar{t}$ 的前 r 个字母来自 T^{fin}（即它们不是 ε）。

▶ **引理 B.1**。对于每个 $r \in \mathbb{N}$，都可以有效构造一个转导器 $\theta_{\bar{w}}\colon A_W \times A_X \twoheadrightarrow T_{\text{有限}+\varepsilon}$，使得对于每个 $\bar{x} \in (A_X)^\omega$，输出 $\theta_{\bar{w}}(\bar{w}, \bar{x})$ 是饱和的且是 r-完全的，并且 $\varphi(\bar{w}, \bar{x}, \theta_{\bar{w}}(\bar{w}, \bar{x}))$ 成立。

证明。首先，我们利用引理 5.9 得到一个同态 $\alpha\colon (A_W)^{\mathsf{W}} \to S$ 到有限 Wilke 代数 S，以及一个转导器 $\tau_Y\colon (A_W \times A_X) \twoheadrightarrow (S^{\mathrm{inf}} \to A_Y)$，使得对于每个 $\bar{x} \in (A_X)^\omega$，都有 $\varphi(\bar{w}, \bar{x}, \tau_Y(\bar{w}, \bar{x}) \bullet \mathrm{lk}_\alpha(\bar{w}))$。请注意，现在我们有两个 Wilke 代数：$S$ 对应“源”字母，T 对应“目标”字母。

我们先将 τ_Y 的输出（即 $(S^{\mathrm{inf}} \to A_Y)$ 中的字母）与 β 复合：$A_Y \to T^{\mathrm{fin}}$。得到的转导器为 $\tau_T\colon (A_W \times A_X) \twoheadrightarrow (S^{\mathrm{inf}} \to T_{\text{有限}+\varepsilon})$。记 $\bar{y} = y_0 y_1 y_2 \cdots \overset{\text{def}}{=} \tau_Y(\bar{w}, \bar{x}) \bullet \mathrm{lk}_\alpha(\bar{w})$，以及 $\bar{t} = t_0 t_1 t_2 \cdots \overset{\text{def}}{=} \tau_T(\bar{w}, \bar{x}) \bullet \mathrm{lk}_\alpha(\bar{w})$，于是对于所有 $i \in \mathbb{N}$，我们都有 $t_i = \beta(y_i)$。由于 β 是一个同态，公式（2.1）适用，因此 $\beta(\bar{y}) = \odot(\bar{t})$，这意味着 $\varphi(\bar{w}, \bar{x}, \tau_T(\bar{w}, \bar{x}) \bullet \mathrm{lk}_\alpha(\bar{w}))$ 成立。我们构造的转导器 $\theta_{\bar{w}}$ 将输出一个 ω-字 $\bar{t}'$，使得 $\odot(\bar{t}') = \odot(\bar{t})$；这也将推出 $\varphi(\bar{w}, \bar{x}, \theta_{\bar{w}}(\bar{w}, \bar{x}))$ 成立。

接下来要由 τ_T 和 α 构造 $\theta_{\bar{w}}$。回顾一下，$\bar{w} = w_0 w_1 w_2 \cdots$ 这个 ω-字是固定的。为简化记号，对于满足 $i < j$ 的 $i, j \in \mathbb{N}$，我们定义 $\alpha_{\bar{w}}(i, j) \overset{\text{定义为}}{=} \alpha(w_{i+1} w_{i+2} \cdots w_j) \in S^{\mathrm{fin}}$，并定义 $\alpha_{\bar{w}}(i, i) \overset{\text{定义为}}{=} \varepsilon \in S^{\mathrm{fin}}_{+\varepsilon}$，以及 $\alpha_{\bar{w}}(i, \infty) \overset{\text{定义为}}{=} \alpha(w_{i+1} w_{i+2} w_{i+3} \cdots) \in S^{\mathrm{inf}}$。给定 $e \in S^{\mathrm{fin}}$，如果存在位置 $i = i_0 < i_1 < i_2 < \ldots$ 使得

$\alpha_{\bar{w}}(i_k, i_\ell) = e$ 对所有满足 $k < \ell$ 的 $k, \ell \in \mathbb{N}$ 都成立（换言之，如果从位置 i 之后开始的 $\bar{w}$ 的后缀可以拆分成若干子字，使得它们各自在 α 下的像都是 e）。根据拉姆齐定理（定理 2.2；另见 [38, 定理 2.1，第 78] 页），存在一个位置 i_0 和一个值 $e \in S^{\text{fin}}$，使得 i_0 是 e-可分的，且 $i_0 \geq r$，其中 r 是引理陈述中的数值。在证明余下部分，我们固定 i_0 和 e。注意，e 是一个幂等元，也就是说，$e \cdot e = e$。

$\theta_{\bar{w}}$ 的总体思路是累积（相乘）τ_T 的输出字母，直到到达下一个可由 e 分割的位置。这是必要的，因为 τ_T 可以访问 α上的前瞻，而 $\theta_{\bar{w}}$ 无法访问；但在每个可由 e 分割的位置，转导器 $\theta_{\bar{w}}$ 可以确定 α 在该后缀上的值为 e^ω（而在以可由 e 分割的位置结尾的有限子词上，α 的值可以由 $\theta_{\bar{w}}$轻松计算）。

$\theta_{\bar{w}}$ 如何知道其当前位置是否可由 e 分割？在第一个可由 e 分割的位置，它可以记住 i_0。随后，在已知某个可由 e 分割的位置后，转导器必须找到下一个这样的位置。当然，$\theta_{\bar{w}}$ 可以维护从前一个可由 e 分割的位置开始的子词的值，并检查它是否等于 e。一个大胆的猜想是：如果 i是可由 e 分割的，并且 $\alpha_{\bar{w}}(i, j) = e$，那么 j 也同样是可由 e 分割的——但这是错误的。不过，一个略强的条件就足够了：如果转导器遇到两个连续的、值为 e的子词，那么它可以确定第一个子词之后的位置是可由 e 分割的，正如下述断言所述。

▷ 断言 B.2. 设 $i < j < k$。如果位置 i 是 e-可分的，并且 $\alpha_{\bar{w}}(i, j) = \alpha_{\bar{w}}(j, k) = e$，那么$j$ 是 e-可分的。

证明。由于 i 是 e-可分的，因此存在一个 e-可分的位置 $\ell > k$ 使得 $\alpha_{\bar{w}}(i, \ell) = e$（在将从 i 开始的后缀拆分成取值为 e的字时，ℓ 我们可以任取一个大于 k的位置）。于是有

$$\begin{aligned}\alpha_{\bar{w}}(j, \ell) &= \alpha_{\bar{w}}(j, k) \cdot \alpha_{\bar{w}}(k, \ell) = e \cdot \alpha_{\bar{w}}(k, \ell) = e \cdot e \cdot \alpha_{\bar{w}}(k, \ell) \\ &= \alpha_{\bar{w}}(i, j) \cdot \alpha_{\bar{w}}(j, k) \cdot \alpha_{\bar{w}}(k, \ell) = \alpha_{\bar{w}}(i, \ell) = e,\end{aligned}$$

这意味着 j 是 e-可分的。 ◁

这一构造大致基于 Thomas [45], 中的思路，其中介绍了一个类似的技巧。将其应用到我们的上下文中，便可用带有谓词 $\psi_s(i, j)$ 的一阶逻辑来表达值 $\alpha(\bar{w}) \in S^{\text{inf}}$，这些谓词用于描述 $\alpha_{\bar{w}}(i, j) = s$。

现在我们来详细介绍 $\theta_{\bar{w}}$ 的构造。开放扇区是一个有序对 (s, g)，其中 $s \in S^{\text{fin}}_{+\varepsilon}$和 $g\colon S^{\text{inf}} \to T$ fin$_{+\varepsilon}$。此外，给定 $\bar{x} = x_0 x_1 x_2 \cdots \in (A_X)^\omega$，并记 $f_0 f_1 f_2 \cdots$ def $= \tau_T(\bar{w}, \bar{x})$ 和 $t_0 t_1 t_2 \cdots$ def $= \tau_T(\bar{w}, \bar{x}) \bullet \text{lk}_\alpha(\bar{w})$，如果 $s = \alpha_{\bar{w}}(i, k)$ 且 $t_{i+1} \cdot t_{i+2} \cdot \ldots \cdot t_k = g(\alpha_{\bar{w}}(k, \infty))$，我们就说 (s, g) 描述位置对 (i, k)（其中 $i \leq k$）。用位置对 $(w, f) \in A_W \times (S^{\text{inf}} \to T\text{fin})$ 更新 (s, g)，所得的开放扇区为 $(s \cdot \alpha(w)\ g')$，其中对所有 $h \in S^{\text{inf}}$，$g'(h)$ def $= g(\alpha(w) \cdot h) \cdot f(h)$。我们可以看到，如果 (s, g) 描述 (i, k)，并用 (w_{k+1}, f_{k+1}) 对其进行更新，那么更新后的开放扇区就描述 $(i, k+1)$。空开放扇区是 $(\varepsilon, g_\varepsilon)$，其中对所有 $h \in S^{\text{inf}}$，$g_\varepsilon(h) = \varepsilon$。对于任意 $i \in \mathbb{N}$，它都描述位置对 (i, i)。

闭合扇区是一个三元组 (t, s, g)，其中 $t \in T^{\text{fin}}$ and (s, g) 是一个开放扇区。另给定 $\bar{x} = x_0 x_1 x_2 \cdots \in (A_X)^\omega$，并记 $f_0 f_1 f_2 \cdots$ def $= \tau_T(\bar{w}, \bar{x})$ 和 $t_0 t_1 t_2 \cdots$ def $= \tau_T(\bar{w}, \bar{x}) \bullet \text{lk}_\alpha(\bar{w})$，若 (t, s, g) 描述位置三元组 (i, j, k)（其中 $i < j \leq k$），且$\alpha_W(i, j) = e$，并且 $t_{i+1} \cdot t_{i+2} \cdot \ldots \cdot t_j = t$，且开放扇区 (s, g) 描述有序对 (j, k)，则称 (t, s, g) 描述该位置三元组。用有序对 (w, f) 更新 $(t, s, g) \in A_W \times (S^{\text{inf}} \to T$ fin$)$，得到闭合扇区 (t, s', g')，其中 (s', g') 是用 (w, f) 更新 (s, g) 的结果。同样，如果 (t, s, g) 描述 (i, j, k)，并用 (w_{k+1}, f_{k+1}) 更新它，那么更新后的闭合扇区就描述 $(i, j, k+1)$。

28 基于 Rabin-Mostowski 指数的广义量词

在读取位置 0、1、. . .、 i_0 处的字母时，新的转导器 $\theta_{\bar{w}}$ 直接模拟τ_T，并计数至 i_0；当 τ_T想要输出一个字母 f 时： $S^{\inf} \to T^{\mathrm{fin}}$ 在读取位置 $i \in \{0, 1, \ldots, i_0\}$ 处的字母时，那么 $\theta_{\bar{w}}$ 输出 $f(\alpha(w_{i+1}w_{i+2}w_{i+3}\cdots))$（其中$\alpha(w_{i+1}w_{i+2}w_{i+3}\cdots)$ 可以硬编码到转导器中，因为 $\bar{w}$ 是预先已知的）。这样，我们可以确保输出的 ω-字 $\theta_{\bar{w}}(\bar{w}, \bar{x})$ 是 r-满的（回忆一下 $i_0 \geq r$），并且对于每个$\bar{x} \in (A_X)^{\omega}$， $\theta_{\bar{w}}(\bar{w}, \bar{x})$ 中截至位置 i_0 的字母都与 $\tau_T(\bar{w}, \bar{x}) \bullet \mathrm{lk}_{\alpha}(\bar{w})$ 中的相同。

到达位置 i_0 后，转导器 $\theta_{\bar{w}}$ 会在其状态中维护以下信息：

- τ_T 的当前状态 q 、一个
- 开放扇区 o，以及闭合扇
- 区的集合 C 。

在位置 i_0， τ_T 的状态已由上一阶段确定；我们将空开放扇区作为开放扇区，并将空集作为 C 。

当收到一个新的输入字母 $(w, x) \in A_W \times A_X$ 时， $\theta_{\bar{w}}$ 的状态按如下方式更新：1. 根据其转移函数更新 τ_T 的状态；设 $f \in (S^{\inf} \to T^{\mathrm{fin}})$ 为 τ_T 产生的输出字母。2. 用 (w, f) 更新开放扇区 o 以及 C 中的所有闭合扇区。更新后，不同的闭合扇区可能变得相同；由于 C 是一个集合，我们只存储一次结果。3. 如果开放扇区 o 的形式为 (e, g)，且第一坐标为 e，则将 $(g(e^{\omega})\ \varepsilon, g_{\varepsilon})$ 加入 C，其中 $(\varepsilon, g_{\varepsilon})$ 是空开放扇区。4. 我们检查 C 中是否存在一个闭合扇区 (t, e, g)，其第二坐标为 e。如果存在，则输出字母 t，将开放扇区 o 替换为 (e, g)，并将 C 替换为 $\emptyset$。如果 C 中存在多个这样的闭合扇区，我们任意选择其中一个，并执行上述操作。如果未找到这样的闭合扇区，则输出单位元 $\varepsilon \in T^{\mathrm{fin}}_{+\varepsilon}$。

我们声称，在读完输入的 ω-字至位置 $k \geq i_0$ 后，$\mathcal{B}$ 的状态满足以下不变量：存在指标 i, j_0，满足 $i_0 \leq i \leq j_0 \leq k$，使得
(a) 位置 i 是 e-可分的，(b) 开放扇区 o 描述 (i, k)，(c) C 中的每个闭合扇区都描述某个满足 $i < j \leq k$的 j 对应的 (i, j, k)，以及 (d) 对于每个满足 $j_0 < j \leq k$ 且 $\alpha_{\bar{w}}(i, j) = e$的位置 j， C中都有一个描述 (i, j, k) 的闭合扇区。

显然，在开始时该不变量成立，此时 $k = i_0$；我们取 i_0 作为 i 和 j_0 。现在来看，在读取某个位置 $k + 1$处的字母时，该不变量如何保持。首先假设第4点中的检查结果为否。于是我们保持 i 和 j_0不变。(a)显然保持不变。(b)和(c)由于第2点中执行的更新而保持不变；对于位置 $j \leq k$，(d)也同样保持不变。如果 $\alpha_{\bar{w}}(i,\ k + 1) = e$，那么现在(d)还要求有一个描述 $(i,\ k{+}1,\ k{+}1)$ 的闭合扇区；这就是第3点中添加到 C 的扇区。

现在假设第4点的检查在 C 中遇到一个闭合扇区 $(t,\ e,\ g)$，对于某个 j 且满足 $i < j \leq k$，它描述了 $(i,\ j,\ k{+}1)$（最后一个坐标上为 $k{+}1$，因为我们已经处于第2点更新之后）。因此我们有 $\alpha_{\bar{w}}(i,\ j) = \alpha_{\bar{w}}(j,\ k{+}1) = e$，所以根据断言 B.2，我们得到 j 是 e-可分的。接着，我们取 j 作为新的 i，并取 $k{+}1$作为新的 j_0。以上证明了条目 (a)，而条目 (b)–(d) 则显然成立。

此外请注意，当 i 移至位置 j时，我们输出 t，它是 $\tau_T(\bar{w},\ \bar{x}) \bullet \mathrm{lk}_{\alpha}(\bar{w})$在位置 $i{+}1$, $i{+}2$, ……, j上的字母的笛卡尔积；而如果 i没有移动，那么

我们输出 ε。因此，如果 i 无限次移动，那么 $\theta_{\bar{w}}$ 的输出是饱和的，其无限乘积 $\odot\big(\theta_{\bar{w}}(\bar{w}, \bar{x})\big)$ 确实与无限乘积 $\odot\big(\tau_T(\bar{w}, \bar{x}) \circ \alpha(\bar{w})\big)$相同。

还需证明 i 确实会被无限次移位。假设并非如此，并考虑 i 和 j_0 的最后取值（注意，j_0 只会与 i同时变化，且最后一次变化发生的位置恰好是 j_0）。考虑将从位置 i 之后开始的后缀拆分为若干子字，使其求值为 e；根据第(a)项，这样的拆分存在。任取该拆分中大于 j_0的两个位置 j、$k{+}1$，使得 $j_0 < j < k{+}1$。由于$\alpha_{\bar{w}}(i, j) = e$，位置 k 处不变量的第(d)项表明，在 C中有一个描述 (i, j, k) 的闭合扇区。在位置 $k{+}1$ 处，它变成一个描述 $(i, j, k{+}1)$ 的闭合扇区。由于 $\alpha_{\bar{w}}(j, k{+}1) = e$，它的第二坐标于是为 e。因此，第4点会找到它，并将 i移至更高的值，这与我们的假设相矛盾。◀

为了从引理 B.1 得到定理 5.12，我们现在只需稍微调整所得的转导器，并利用 Wilke 代数的一些代数性质。接下来的两个引理将给出这些性质。

▶ **引理 B.3**。设 T 为有限 *Wilke* 代数，$T^{\mathrm{fin}}_{+\varepsilon}$ 为在 T^{fin}中添加形式中性元 ε后得到的幺半群。存在一个常数 $r \in \mathbb{N}$ 和一个转导器θ_{d}：$T_{\text{有限}+\varepsilon} \twoheadrightarrow T_{\text{有限}}$，使得对于每个饱和的 r-满 ω-字 $\bar{t} \in (T^{\mathrm{fin}}_{+\varepsilon})^{\omega}$，都有

$$\odot(\bar{t}) = \odot\big(\theta_{\mathsf{d}}(\bar{t})\big).$$

证明。令 r 为事实 2.3 中针对半群 T^{fin}给出的常数。

转导器 θ_{d} 分两个阶段运行。第一阶段的状态形如$c \in T^{\mathrm{fin}}_{+\varepsilon}$，第二阶段的状态形如 (e, c)，其中 $e \in T_{\text{有限}}$ 且$c \in T^{\mathrm{fin}}_{+\varepsilon}$。第一阶段用于输入的前缀 ω-字；在此阶段，θ_{d} 会复制输入字母，并在其状态中存储已读取部分的字的笛卡尔积 $c \in T^{\mathrm{fin}}_{+\varepsilon}$。每当存在满足条件的 $e \in T^{\mathrm{fin}}$，使 c的新值满足 $c \cdot e = c$时，它便转入第二阶段（若有多个这样的元素 e，则任意选取一个）。因此，初始状态为 ε，第一阶段中从状态 $c \in T^{\mathrm{fin}}_{+\varepsilon}$ 出发的转移函数对 $t \in T^{\mathrm{fin}}$ 定义如下：

$$\delta(c, t) \stackrel{\mathrm{def}}{=} \begin{cases} \big(t, (e, \varepsilon)\big) & \text{if } \exists e \in T^{\mathrm{fin}}.\ c \cdot t \cdot e = c \cdot t, \\ \big(t, c \cdot t\big) & \text{otherwise,} \end{cases}$$

并且 $\delta(c, \varepsilon)$ 会进入一个特殊的“错误”状态，从该状态出发，θ_{d} 持续产生某个固定的输出字母。注意，在读取了一个前缀 $t_0 t_1 \cdots t_{j-1}$ 且其中所有字母都属于 T^{fin}（即不属于 ε）之后，如果转导器仍处于第一阶段，那么其状态为 $t_0 \cdot t_1 \cdot \ldots \cdot t_{j-1}$，并且产生的输出恰好是输入的副本。

现在我们利用事实 2.3 来说明，在读取任意一个 r-满 ω-字 $\bar{t} = t_0 t_1 t_2 \cdots \in (T^{\mathrm{fin}}_{+\varepsilon})^{\omega}$ 时，转导器确实会在前 r 个位置中的某处进入第二阶段，同时避开错误状态。事实上，将事实 2.3 应用于前缀 $t_0 t_1 \cdots t_{r-1}$，可得到一个位置j和一个幂等元 e，使得（最迟）在读取 $t_0 t_1 \cdots t_j$之后，当前笛卡尔积$c \cdot t_j = t_0 \cdot t_1 \cdot \ldots \cdot t_j$满足 $c \cdot t_j \cdot e = c \cdot t_j$，从而完成第一阶段。

这一第二阶段的转移函数定义如下：

$$\delta\big((e, c), t\big) \stackrel{\mathrm{def}}{=} \begin{cases} \big(c \cdot t, (e', \varepsilon)\big) & \text{if } \exists e' \in T^{\mathrm{fin}}.\ c \cdot t \cdot e' = c \cdot t, \quad \text{(Case I)} \\ \big(e, (e, c \cdot t)\big) & \text{otherwise,} \qquad\qquad\qquad\quad \text{(Case II)} \end{cases}$$

其中，如果有多个元素 $e' \in T^{\mathrm{fin}}$，我们就任意选择其中一个。注意，对于 $e' \in T^{\mathrm{fin}}$，$\varepsilon \cdot e' \neq \varepsilon$，因此情况 I 中的输出字母 $c \cdot t$确实属于 T_{fin}。

30 基于 Rabin-Mostowski 指数的广义量词

¯现在考虑某个实际的饱和 r-满输入 ω-字 $t = t_0t_1t_2\cdots$，以及一个位置 j：在该位置之后，转导器已经进入第二阶段，处于状态 (e, c)。设 $s_0s_1\cdots s_j$ 为目前为止产生的输出字。我们保持以下不变量：存在 $j_0 \le j$ 使得

- $t_0 \cdot t_1 \cdot \ldots \cdot t_{j_0} = s_0 \cdot s_1 \cdot \ldots \cdot s_j, t_0 \cdot t_1 \cdot$
- $\ldots \cdot t_{j_0} = t_0 \cdot t_1 \cdot \ldots \cdot t_{j_0} \cdot e, t_{j_0+1} \cdot$
- $t_{j_0+2} \cdot \ldots \cdot t_j = c.$

当我们从阶段一进入阶段二时，不变量显然成立，此时有 $j_0 = j$。现在假设读入一个字母 t_{j+1}。如果转移函数采用情况 I，那么将 j_0 移至 j+1后，不变量仍成立；输出 $c \cdot t_{j+1}$之后，输出字母的乘积会乘以 $c \cdot t_{j+1} = t_{j_0+1} \cdot t_{j_0+2} \cdot \ldots \cdot t_{j+1}$，这会由 j_0的移动来抵消。如果采用情况 II，则不变量在相同的 j_0下保持不变；注意，输出 e 时，我们不会改变输出的乘积，因为不变量的第二项如此规定。

还需证明情况 I 会被无限次地使用，从而 j_0 趋于无穷。反过来，假设从某个时刻起，j_0 的值保持不变。取满足 $j_0 < i_0$和 t_{i_0}，t_{i_1}，……，$t_{i_{r-1}} \neq \varepsilon$的最前面的 r 个位置 $i_0 < i_1 < \ldots < i_{r-1}$（这样的位置有无限多个，因为 $\bar{t}$是饱和的），并考虑字 $t_{i_0}t_{i_1}\cdots t_{i_{r-1}}$。根据事实 2.3，存在一个指标 j 和一个幂等元 e'，使得对于$c = t_{i_0} \cdot t_{i_1} \cdot \ldots\ldots \cdot t_{i_{j-1}} = t_{j_0+1} \cdot t_{j_0+2} \cdot \ldots\ldots \cdot t_{i_j-1}$，有 $c \cdot t_{i_j} \cdot e' = c \cdot t_{i_j}$。这意味着读取位置 i_j时使用了情况 I，这与我们的假设相矛盾。◀

▶ **引理 B.4**。假设 $\beta\colon A^{\mathsf{W}} \to T$是到有限 *Wilke* 代数 T上的一个同态。那么存在一个转导器 $\theta_\beta\colon T^{\mathrm{fin}} \twoheadrightarrow A$，使得对于每个 ω-字 $\bar{t} \in (T^{\mathrm{fin}})^\omega$，都有

$$\odot(\bar{t}) = \beta\big(\theta_\beta(\bar{t})\big).$$

在该引理中，β 确实是满射到 T的，也就是说，T 中的所有元素都在 β的像中。

证明。首先，对于每个 $t \in T^{\mathrm{fin}}$，我们固定任意一个字 $u_t \in A^+$，使得 $\beta(u_t) = t$。然后，我们将 θ_β 的状态集取为

$$\{(c, w) \in T^{\mathrm{fin}}_{+\varepsilon} \times A^* \mid \exists t \in T^{\mathrm{fin}}, u \in A^+ . uw = u_t\},$$

其初始状态为（ε，ε）。转移函数根据以下情况定义（在这两种情况下，$a \in A$ 都是单个字母）：

$$\delta((c, aw), t) \stackrel{\mathrm{def}}{=} \big(a, (c \cdot t, w)\big), \qquad \text{and}$$
$$\delta((c, \varepsilon), t) \stackrel{\mathrm{def}}{=} \big(a, (\varepsilon, w)\big), \qquad \text{其中 } u_{c\cdot t} = aw。$$

在该构造中，c 存储尚未贡献给输出的那些输入字母的笛卡尔积；初始时 $c = \varepsilon$。读取字母 t后，转导器开始输出字 $u_{c\cdot t}$，其在 β下的像为 $c \cdot t$。该字可能由多个字母组成，因此，仍需输出的后缀存储在状态的第二坐标中，而第一坐标则与此期间读入的输入字母相乘。当整个字输出完毕后，我们读取下一个字母，并开始输出下一个字，该字对应于此期间读入的字母的笛卡尔积；如此继续。◀

定理 5.12 的证明。回想一下，我们已经固定了一个输入 ω-字 $\bar{w} \in (A_W)^\omega$，以及一个同态 $\beta\colon (A_Y)^{\mathsf{W}} \to T$到一个有限 Wilke 代数 T。取转换器$\theta_{\bar{w}}\colon A_W \times A_X \twoheadrightarrow T^{\mathrm{fin}+\varepsilon}$（来自引理 B.1）、$\theta_{\mathsf{d}}\colon T^{\mathrm{fin}}_{+\varepsilon} \twoheadrightarrow T^{\mathrm{fin}}$（来自引理 B.3），以及 $\theta_\beta\colon T^{\mathrm{fin}} \twoheadrightarrow A_Y$

（来自引理 B.4）；其中，在引理 B.1 中，参数 $r \in \mathbb{N}$ 取引理 B.3 给出的常数 r。对于 $\tau_{\bar{w}}$：$A_W \times A_X \to A_Y$，我们取由转换器 $\theta_{\bar{w}}$、 θ_{d} 和 θ_β 复合而成的转导器。现在再给定一个字 $\bar{x} \in (A_X)^\omega$，我们知道 $\varphi(\bar{w}, \bar{x}, \theta_{\bar{w}}(\bar{w}, \bar{x}))$ 成立；这些等式

$$\odot\big(\theta_{\bar{w}}(\bar{w},\bar{x})\big) = \odot\big(\theta_{\mathsf{d}}\big(\theta_{\bar{w}}(\bar{w},\bar{x})\big)\big) = \beta\big(\theta_\beta\big(\theta_{\mathsf{d}}\big(\theta_{\bar{w}}(\bar{w},\bar{x})\big)\big)\big) = \beta\big(\tau_{\bar{w}}(\bar{w},\bar{x})\big)$$

这意味着 $\varphi(\bar{w}, \bar{x}, \tau_{\bar{w}}(\bar{w}, \bar{x}))$ 也成立。 ◀

C 定理 7.1 的证明

▶**定理 7.1.** 树上的单子二阶逻辑$+\mathtt{I}$ 理论是不可判定的。即使只允许使用最简单的指标，即弱奇偶指数 $W_{0,1}$ （即安全性），以及任意类型的确定性 $D \in \{\mathsf{dt},\mathsf{nd}\}$，结论依然成立，因此我们只使用指数量词 $\mathtt{I}^D_{W_{0,1}}$。

我们首先借助 Bojańczyk 等人 [4] 的结果，论证中间形式系统单子二阶逻辑+有界是不可判定的。然后我们证明引理 7.2，说明该形式系统可以在单子二阶逻辑$+\mathtt{I}$中编码。

C.1 单子二阶逻辑+有界的不可判定性

我们依赖 Bojańczyk 等人 [4] 给出的证明：MSO+U 是不可判定的。他们论文中考察的结构是由有限深度为 4 的树构成的无限序列（以自然数集为索引）。这些树中的节点可以有任意多个子节点，且子节点之间有序；从一个节点到其后代的每条路径至多包含 4 个节点。在这种结构上定义的逻辑可以使用二元关系：“是子节点”“是下一个兄弟节点”以及“是下一棵树的根”。

现在我们给出一些定义（名称与完全二叉树中的相同，但含义不同）。深度为 4 的树构成的序列中的节点集合 I，若其所有元素都是某棵树中的连续的兄弟节点，则称为区间。若两个区间 I_1、 I_2 中，一个区间的节点都不是另一个区间中某个节点的后代、祖先或兄弟节点，则称这两个区间独立 （换言之，这些区间必须位于不同的树中，或者它们的父节点在后代序下不可比较）。最后，独立区间的并集 是指可以写成 $\bigcup_{i\in J} I_i$ 的集合 $\tilde{z}$，其中 $(I_i)_{i\in J}$ 是两两独立的区间。现在考虑一种在深度为 4 的树构成的序列上解释的单子二阶逻辑扩展，其中可以使用原子公式 bounded(Z)，表示 Z 是一个独立区间的并集，且这些区间的大小有某个 $n \in \mathbb{N}$ 上界；将此扩展记为 MSO+bounded。我们使用以下定理。

▶**定理 C.1** （[4]）。 以下问题不可判定：给定单子二阶逻辑 *MSO+bounded* 的一个句子 ψ，判断是否存在一个深度为4 的树序列，使得其中 ψ 成立。

严格来说，Bojańczyk 等人 [4] 给出的不可判定性结果针对的是 MSO+U 逻辑，但很容易得到这样的修改。具体而言，他们的不可判定性证明给出了一个归约，将一个不可判定问题（Minsky 机的语言空性）归约为 MSO+U 的可满足性。该归约中构造的 MSO+U 公式只在子公式 $\varphi_{\mathsf{b}}(Z)$ 中使用 U 量词，这些子公式说明，对于独立区间的并集 Z， Z 中的区间长度有某个上界。将每个这样的子公式替换为原子公式 bounded(Z)，即可得到一个等价的 MSO+bounded 公式。经过这一修改，我们便得到一个从不可判定问题到 MSO+bounded 可满足性的归约，从而证明后者不可判定。

C.2 从单子二阶逻辑+有界到树

现在，我们的目标是说明，定理 C.1 中的问题可以在完全二叉树上用 $\mathrm{MSO}^{+\mathtt{I}^{\mathsf{nd}}_{\mathcal{W}_{0,1}}}$ 和 $\mathrm{MSO}^{+\mathtt{I}^{\mathsf{dt}}_{\mathcal{W}_{0,1}}}$ 表达。为此，我们将深度为 4 的树序列编码为无限二叉树的节点集合 S，使得序列中各树的节点与 S中的节点一一对应。序列中第 i棵树的根编码为 $\mathtt{R}^i$。接着，使用首孩子-下一个兄弟编码来表示每棵树：若某个节点编码为 u，则它的第 i个孩子编码为 $u\mathtt{LR}^i$。于是，“是否存在一个深度为 4 的树序列”这一问题就变为“是否存在一个集合 S，它编码了一个深度为 4 的树序列”。显然，二叉树上的 MSO 可以表达 S 确实是某个深度为 4 的树序列的有效编码这一事实。此外，只要使用 MSO 构造，关于深度为 4 的树序列的句子 ψ 就可以改写为二叉树上的句子，用来描述该序列的编码 S。剩下要处理的是新的原子公式 bounded(Z)。这里很容易表达 Z 是独立区间的并集这一事实，唯一的难点在于有界性。不过我们注意到，深度为 4 的树中的兄弟节点沿着向右的分支编码；因此，深度为 4 的树中的一个区间会编码为二叉树中的一个区间，而深度为 4 的树序列中的独立区间并集则会编码为二叉树中的独立区间并集。因此，我们可以用引理 7.2 中的公式 $\varphi_{\mathsf{b}}(Z)$表达 Z 中的区间长度有界这一事实。回顾一下，公式 $\varphi_{\mathsf{b}}(Z)$既可以用 $\mathrm{MSO}^{+\mathtt{I}^{\mathsf{nd}}_{\mathcal{W}_{0,1}}}$ 书写，也可以用 $\mathrm{MSO}^{+\mathtt{I}^{\mathsf{dt}}_{\mathcal{W}_{0,1}}}$书写，因此（以引理 7.2 为前提），这就证明了这两种逻辑都不可判定。

C.3 引理 7.2 的证明

在本小节中，我们将正式证明引理 7.2 的证明中尚未完成的蕴含，即如果 $\tilde{z}$ 中的区间长度无界，那么 $\varphi_{\mathsf{b}}(\tilde{z})$ 不成立。

回顾

$$\varphi_{\mathsf{b}}(Z) \equiv \forall W \subseteq Z.\, \forall Y \subseteq Z.\, \mathtt{I}^{D}_{\mathcal{W}_{0,1}} X.\, \underbrace{\big(X \subseteq W \wedge \forall x \in X.\, \exists y \in Y.\, x \preceq y\big)}_{\psi(W,Y,X)},$$

并且，我们将 $\tilde{w}$ 取为 $\tilde{z}$ 中所有区间的所有最顶端点组成的集合。下面我们构造集合 $\tilde{y}$。

▷ 断言 C.2. 我们可以找到节点 $\varepsilon = u_0 \prec u_1 \prec u_2 \prec \ldots$ 和区间 I_0，I_1，I_2，……，它们来自 $\tilde{z}$，使得 长度(I_i) $\geq i$，并且 $u_i \preceq top(I_i)$，但对所有 $i \in \mathbb{N}$，$u_{i+1} \not\preceq top(I_i)$。

证明。假设我们已经有点 u_0、u_1、……、u_n 以及上述区间 I_0、I_1、……、I_{n-1}，其中 $\tilde{z} \cap \{u \mid u_n \preceq u\}$ 中的区间长度无界（我们通过令 $u_0 = \varepsilon$ 取 $n = 0$ 来初始化）。我们按如下方式选取下一个区间 I_n 和节点 u_{n+1}。对于 I_n，我们只需在 $\tilde{z} \cap \{u \mid u_n \preceq u\}$ 中任取一个长度至少为 n 的区间。接着，我们考察从 u_n 到 top(I_n) 的路径旁边的节点（即不在该路径上、但其父节点在该路径上的节点）。这样的节点只有有限多个，因此至少有一个这样的节点 v 应当满足：$\tilde{z} \cap \{u \mid v \preceq u\}$ 中的区间长度无界。我们任取这样的 v 作为 u_{n+1}。◁

下面我们考虑字母表为 $\{0,1\}^3$ 的非确定性安全自动机；它们读取二叉树，其中每个节点都标记为一个三元组（w, y, z） $\in \{0,1\}^3$,，其中 w 表示该节点是否属于集合 $\tilde{w}$；同样，y 表示 $\tilde{y}$，x 表示 $\tilde{x}$。如果对于每个转移（q,（$w, y,$1）0, $q_{\mathtt{L}}, q_{\mathtt{R}}$），自动机 $\mathcal{A}$ 还具有一个转移，则称该自动机是单调的

$(q, (w, y, 0)\ 0, q_{\mathrm{L}}, q_{\mathrm{R}})$。换句话说，如果 ρ 是自动机 $\mathcal{A}$ 在 $\langle \tilde{w}, \tilde{y}, \tilde{x} \rangle$ 上的一次运行，并且 $\tilde{x}' \subseteq \tilde{x}$，那么 ρ 仍然是自动机 $\mathcal{A}$ 在 $\langle \tilde{w}, \tilde{y}, \tilde{x}' \rangle$ 上的一次有效运行。

▷ 命题 C.3. 若对于某些 $\tilde{w}, \tilde{y}$，子公式 $\psi(\tilde{w}, \tilde{y}, X)$ 可由某个非确定性安全自动机 $\mathcal{A}$ 识别，那么它也可由单调非确定性安全自动机 $\mathcal{A}'$ 识别。

证明. 要构造 A′，只需向 A 中添加所有满足以下条件的转移 $(q, (w, y, 0)\ 0, q_{\mathrm{L}}, q_{\mathrm{R}})$：$(q, (w, y, 1)\ 0, q_{\mathrm{L}}, q_{\mathrm{R}})$ 是 A 的一个转移。这样，A 的每次运行都是 A′ 的一次运行，因此 A′ 接受所有满足我们性质的集合 $\tilde{x}$。反过来，假设对于某个集合 $\tilde{x}'$，我们有 A′ 的一次运行。那么，根据 A′ 的构造，存在 A 在某个集合 $\tilde{x} \supseteq \tilde{x}'$ 上的一次运行（每当 A′ 使用一个只存在于 A′ 中、读取 $(w, y, 0)$ 的转移时，我们就将该节点添加到 $\tilde{x}$ 中，并使用 A 中读取 $(w, y, 1)$ 的对应转移）。根据关于 A 的假设，集合 $\tilde{x}$ 满足我们的性质，而其子集 $\tilde{x}' \subseteq \tilde{x}$ 更是如此。◁

现在考虑字母表 $\{0, 1\}^3$ 上所有单调非确定性安全自动机 $\mathrm{A}_0, \mathrm{A}_1, \mathrm{A}2, \ldots$ 的列表。固定某个 $n \in \mathbb{N}$，并假设我们已经有一个数 m_n，也已经固定了集合 $\tilde{y} \cap \{u \mid u_{m_n} \not\preceq u\} \subseteq \tilde{z}$（即对于不是 u_{m_n} 的后代的节点，我们已经决定它们是否属于 $\tilde{y}$），且满足以下性质：无论我们如何选择 $\tilde{y} \cap \{u \mid u_{m_n} \preceq u\} \subseteq \tilde{z}$，自动机 $\mathrm{A}0, \ldots, \mathrm{A}_{n-1}$ 都无法识别子公式 $\psi(\tilde{w}, \tilde{y}, X)$（取 $m_0 = 0$ 时，上述性质对 $n = 0$ 显然成立）。现在我们以一种使上述性质对 $n+1$ 成立的方式，定义下一个数 $m_{n+1} > m_n$ 以及集合 $\tilde{y}$ 的下一个片段，即 $\tilde{y} \cap \{u \mid u_{m_n} \preceq u \wedge u_{m_{n+1}} \not\preceq u\}$。对所有 $n \in \mathbb{N}$ 完成这一构造后，我们将得到整个集合 $\tilde{y} \subseteq \tilde{z}$，使得没有单调非确定性安全自动机 A 能正确识别子公式 $\psi(\tilde{w}, \tilde{y}, X)$。因此，命题 C.3 意味着没有非确定性（因而也没有确定性）安全自动机能够识别子公式 $\psi(\tilde{w}, \tilde{y}, X)$，这意味着 $\varphi_{\mathsf{b}}(\tilde{z})$ 不成立。

对于一个状态集为 Q 的自动机 A_n，我们按如下方式进行。对于每一对 (i, j)，其中 $i < j$，我们将集合 $R_{i,j}$ 赋给满足以下条件的三元组 $(q_\top, q_\bot, q_I)$：存在 A_n 的一个接受运行（即不包含优先级为 1 的转移），它在上下文 $C_{i,j} = \{u \mid u_i \preceq u \wedge u_j \not\preceq u \wedge \mathrm{top}(I_i) \not\preceq u\}$ 上运行，其中 $\tilde{y}$ 和 $\tilde{x}$ 均为空（且 $\tilde{w}$ 包含 $\tilde{z}$ 中所有区间的最顶层节点），并且 $q_\top$ 位于根 u_i 中，$q_\bot$ 位于孔 u_j 中，而 q_I 位于孔 $\mathrm{top}(I_i)$ 中。根据拉姆齐定理（见定理 2.2），我们可以找到指标 $i_0, \ldots, i_{2|Q|}$，使得 $\max(m_n, |Q| + 1) \leq i_0 < i_1 < \ldots < i_{2|Q|}$，且所有 $R_{i_j, i_{j'}}$（其中 $0 \leq j < j' \leq 2|Q|$）都相同。我们固定 $m_{n+1} = i_{2|Q|}$，并将区间 $I_{i_0}, I_{i_2}, I_{i_4}, \ldots, I_{2|Q|}$ 的最底端点（仅偶数！）加入 $\tilde{y}$。

现在假设 $\tilde{y}$ 的其余部分（即 $u_{m_{n+1}}$ 下方的部分）也已固定。取包含区间 I_{i_0}、I_{i_2}、I_{i_4}、……、$I_{2|Q|}$ 的最顶端点的 $\tilde{x}$（同样只取偶数）。对于这样的集合 $\tilde{x}$，子公式 $\psi(\tilde{w}、\tilde{y}、\tilde{x})$ 成立。如果 $\mathcal{A}_n$ 不接受三元组 $\langle \tilde{w}、\tilde{y}、\tilde{x} \rangle$，则结论成立。假设它接受，并固定某个接受运行 ρ。根据鸽巢原理，存在指标 b、e，使得 $0 \leq b < e \leq |Q|$，且 $\rho(u_{i_{2b}}) = \rho(u_{i_{2e}})$；将此状态记为 q。再记 $q_1 = \rho(u_{i_{2b+2}})$。

▷ 断言 C.4。我们有 $\big(q、q_1、\rho(\text{顶部}(I_{i_{2b}}))\big) \in R_{i_{2b}、i_{2b+2}} = R_{i_{2b+1}、i_{2b+2}}$。

证明。这可由运行 ρ 直接证得，因为集合 $\tilde{y}$ 和 $\tilde{x}$ 在上下文 $C_{i_{2b}, i_{2b+2}}$ 中都不含任何元素。等式由指标 i_j 的定义得出。◁

▷ 断言 C.5. 对于每个 $i \geq |Q| + 1$（尤其是所有满足 $b < j < e$ 的 $i = i_{2j}$），都存在一个运行：它以 $\mathcal{A}$ 在从 $top(I_i)$ 开始的子树上运行，使用原始的 $\tilde{w}$（其中只包含 $top(I_i)$），将 $\emptyset$ 视为 $\tilde{y}$ 和 $\tilde{x}$，并且状态 $\rho(top(I_i))$ 位于 $top(I_i)$ 中。

证明。首先，根据 A_n 的单调性，我们可以将 ρ 读取 $\tilde{w}, \tilde{y}, \tilde{x}$视为读取 $\tilde{w}, \tilde{y}, \emptyset$的运行。注意，限制在我们子树上的 $\tilde{y}$ 只能包含 I_i 的最底端节点（右侧最末分支上的某个节点，与根的距离至少为 $|Q|+1$）。根据鸽巢原理，存在两个指标 r、 s，使得 $1 \leq r < s \leq |Q|+1$，且 $\rho(\mathsf{top}(I_i)_{\mathbb{R}^r}) = \rho(\mathsf{top}(I_i)_{\mathbb{R}^s})$。之后，在 $\mathsf{top}(I_i)_{\mathbb{R}^r}$ 以下，我们永远重复运行中从上下文的 $\mathsf{top}(I_i)_{\mathbb{R}^r}$到 $\mathsf{top}(I_i)_{\mathbb{R}^s}$这一部分，从而得到一个具有空 $\tilde{y}$ 和 $\tilde{x}$ 的树上的运行（并且由于 $r \geq 1$，我们不会复制包含在 $\tilde{w}$中的根节点）。◁

▷ 断言 C.6. 对于每个 $i \geq |Q|+1$（特别是对于 $i = i_{2b}$），在从 $top(I_i)$ 开始的子树上，都存在一个$\mathcal{A}_n$ 的接受运行，它具有原来的 $\tilde{w}$ 和 $\tilde{x}$，并将 $\emptyset$ 取作 $\tilde{y}$；该运行在 $top(I_i)$ 中具有状态 $\rho(top(I_i))$。

证明。 与上文相同，但我们不令 $\tilde{x}$ 为空。注意， $\tilde{x}$ 可能只包含该子树的根，因此 $\tilde{x}$ 中的元素不会被复制。◁

▷ 断言 C.7. 我们有 $\big(q,\ q,\ \rho(top(I_{i_{2b}}))\big) \in R_{i_{2b},\ i_{2e}} = R_{i_{2b},\ i_{2b+1}}$。

证明。 我们修改上下文 $C_{i_{2b},\ i_{2e}}$上的运行 ρ。原本，该上下文在从 $top(I_{i_{2j}})$ 开始的子树中包含 $\tilde{y}$ 和 $\tilde{x}$ 的一些元素（对于 $b < j < e$），但我们根据断言 C.5 将它们置空。◁

- 我们按如下方式修改原始运行 ρ。在上下文 $C_{i_{2b},i_{2b+1}}$ 中，我们将原始运行替换为任意满足以下条件的运行： q 属于 $u_{i_{2b}}$（与之前一样）， q 属于 $u_{i_{2b+1}}$，且 $\rho(\mathsf{top}(I_{i_{2b}}))$ 属于 $\mathsf{top}(I_{i_{2b}})$（与之前一样）；根据断言 C.7 以及 $R_{i_{2b},i_{2b+1}}$ 的定义，这样的运行存在。
- 在上下文 $C_{i_{2b+1},i_{2b+2}}$ 中，我们将原始运行替换为任意满足以下条件的运行： q 属于 $u_{i_{2b+1}}$（与上文定义的部分一致）， q_1 属于 $u_{i_{2b+2}}$（与之前一样），且 $\rho(\mathsf{top}(I_{i_{2b}}))$ 属于 $\mathsf{top}(I_{i_{2b+1}})$；根据断言 C.4 以及 $R_{i_{2b+1},i_{2b+2}}$ 的定义，这样的运行存在。我们将 $\mathsf{top}(I_{i_{2b+1}})$ 加入 $\tilde{x}$，从而定义 $\tilde{x}' \stackrel{\text{def}}{=} \tilde{x} \cup \{\mathsf{top}(I_{i_{2b+1}})\}$。从 $\mathsf{top}(I_{i_{2b+1}})$ 开始的子树与从 $\mathsf{top}(I_{i_{2b}})$ 开始的子树仅在一个方面有所不同：前者子树中没有 $\tilde{y}$ 的元素。因此，根据断言 C.6，我们可以用一个以 $\rho(\mathsf{top}(I_{i_{2b}}))$ 属于 $\mathsf{top}(I_{i_{2b+1}})$ 开始的运行来覆盖它。

由此，我们得到一个在 $\mathcal{A}_n$ 上运行于 $\langle \tilde{w}, \tilde{y}, \tilde{x}' \rangle$的接受运行。但是，$\tilde{x}'$的新节点下方没有任何节点属于 $\tilde{y}$，即没有节点属于 $top(I_{i_{2b+1}})$，因此子公式 $\psi(\tilde{w}, \tilde{y}, \tilde{x}')$ 不成立。这意味着 $\mathcal{A}_n$ 未能正确识别子公式 $\psi(\tilde{w}, \tilde{y}, X)$。

$\tilde{y}$ 的归纳构造至此完成，这表明当 $\tilde{z}$ 中的区间长度无界时， $\varphi_{\mathsf{b}}(\tilde{z})$不成立。因此，引理 7.2 的证明结束。