arXiv:2602.13120v1 [cs.LO] 13 Feb 2026

# 作用域 MSO、寄存器自动机与表达式：数据字上的等价

**RadosławPiórkowski** ✉

华威大学

**摘要**

本文为无限字母表上的带猜测的非确定性寄存器自动机（NRA）所识别的语言类，建立了基于逻辑和表达式的刻画。我们引入作用域 MSO，这是一种具有新颖片段模态、并对数据比较施加句法限制的逻辑。我们证明，在能够消除“强猜测”的数据域上，该逻辑与 NRA 表达能力等价。此外，我们定义了数据正则表达式：一种由无量词区域构成、配备 $k$-收缩连接的极简正则表达式演算，并证明它在任意关系结构上与 NRA 等价。这些形式体系共同构成了寄存器自动机的稳健描述理论，弥合了自动机、逻辑与表达式之间的鸿沟。

**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 → 正则语言

**关键词和短语** 寄存器自动机、数据字、单子二阶逻辑、无限字母表、克莱尼定理、名义集合

## 1 引言

正则语言 类的一个显著特性是其出色的稳健性：同一类语言可以有多种截然不同却彼此等价的描述。在有限字母表 $\Sigma$ 上，有限自动机、正则表达式以及词结构上的单子二阶逻辑（MSO）定义的语言完全相同，恰好就是正则语言。这种三方等价（以及通过代数、时序逻辑、簇等得到的诸多细化形式）构成了经典语言理论的支柱，也解释了为什么“正则性”是一个如此稳定且可复用的概念。

从有限字母表转向无限字母表时——例如对含有进程 ID、数据库键、大型字母表（如 Unicode）或 XML 中数据值的系统进行建模——这种稳健性通常就不复存在了。此类轨迹的标准抽象是数据字，即在 $\Sigma \times \mathbb{A}$ 上的序列，其中 $\Sigma$ 是有限标签集，$\mathbb{A}$ 是称为原子 的数据域，并将其建模为具有可数无限论域的关系结构。对 $\mathbb{A}$（称为数据值）中元素的预期访问方式，仅限于测试 $\mathbb{A}$ 签名中的关系；原子的常见示例包括等式原子 $\mathbb{A}_{=} = (\mathbb{N},\ =)$ 和稠密序原子 $\mathbb{A}_{<} = (\mathbb{Q},\ <,\ =)$。针对数据字的形式体系种类繁多，呈现出支离破碎的局面：其中包含大量自动机类型和逻辑，它们的表达能力往往互不可比，其判定问题的复杂度和可判定性也各不相同 [15, 17, 1, 2, 14]。与有限字母表的情形不同，对于数据字而言，并不存在一个统一的概念来扮演“*the*”正则语言的角色。

**寄存器自动机。** 针对数据字的一种自然有限状态模型候选是寄存器自动机，由 Kaminski 和 Francez [12] 提出。这类自动机通过一组固定的寄存器扩展有限控制，能够存储数据值并比较它们是否相等。随后，该模型被扩展到任意原子 $\mathbb{A}$ [5]。本文研究带猜测的非确定寄存器自动机（$\mathrm{NRA}^{\mathrm{G}}$）这一类。“猜测”能力是寄存器自动机三种可能的非确定性变体中限制最少的一种。尽管 $\mathrm{NRA}^{\mathrm{G}}$ 是标准的基线模型，但长期以来一直缺乏 MSO 风格的逻辑刻画。

## 2 作用域 MSO、寄存器自动机与表达式：数据词上的等价性

**NRA 的 MSO 逻辑。** 要建立 $\mathrm{NRA}^{\mathrm{G}}$ 与逻辑之间的对应关系，必须审慎校准表达能力。仅仅在 MSO[<]中加入数据原子公式，就会导致不可判定性 [15, 3], ；即便是一阶（FO）片段、只含三个变量且采用最简单的原子变体 $\mathbb{A}_{=}$，也是如此：可以在数据字中编码任意维度的类网格结构，并在其上模拟图灵机的运行。反过来，若对逻辑限制过严（例如，只允许刚性守卫，或限制为二变量 FO 片段），虽然能得到可判定性，却无法刻画 $\mathrm{NRA}^{\mathrm{G}}$ [9, 4, 3]的全部表达能力。$\mathrm{NRA}^{\mathrm{G}}$ 对语言补集 [14]不封闭，这又带来了另一项挑战——任何具有相同表达能力的逻辑都必须限制否定运算。

**NRA 的表达式。** 关于表达式形式体系，Brunet 和 Silva 定义了带有显式绑定子的表达式，用于名称分配和释放 [6] ，并证明了克莱尼定理的一个类似结论：这些表达式恰好定义了 $\mathrm{NRA}^{\mathrm{G}}$ 的语言。他们的形式体系提供了一种灵活的规范语言，能够以简洁易读的表达式为代价，采用更复杂的语义定义。我们的目标与之互补：构建一种极简表达式形式体系，其语法与经典正则表达式的精神十分接近。

**贡献。** 本文针对 $\Sigma \times \mathbb{A}$上的数据词语言，提出了两种新的形式体系：作用域 MSO 逻辑（$\mathrm{MSO}_{\mathrm{S}}[<, \Sigma, \mathbb{A}]$）和数据正则表达式（$\mathrm{DRE}[\Sigma, \mathbb{A}]$），从而为 NRA 重现了经典三位一体：

$$\text{自动机}（\mathrm{NRA}^{\mathrm{G}}[\Sigma, \mathbb{A}]） \equiv \text{表达式}（\mathrm{DRE}[\Sigma, \mathbb{A}]） \equiv \text{逻辑}（\mathrm{MSO}_{\mathrm{S}}[<, \Sigma, \mathbb{A}]）。$$

**1. 数据正则表达式：** 我们在 $\Sigma \times \mathbb{A}$ 上引入一种紧凑的表达式形式。数据正则表达式由数据字的 无量词区域 构成。该演算依赖一种作用于语言的参数化运算，称为 $k$-收缩连接。参数$k$ 揭示因子之间有界的“接口”，反映了自动机的有限寄存器集。

▶ 我们证明，$\mathcal{L}$(数据正则表达式$[\Sigma, \mathbb{A}]) = \mathcal{L}(\mathrm{NRA}^{\mathrm{G}}[\Sigma, \mathbb{A}])$。

**2. 作用域 MSO：** 我们定义了一种逻辑，专为 NRA 的表达能力量身打造[G]。$\mathrm{MSO}_{\mathrm{S}}[<]$通过两项特性扩展了 MSO：

- 一种片段模态 $_X\,\varphi$，它在由一组切分位置 $X$ 定义的子区间上计算 $\varphi$ 。受限原子谓词 $R(\boldsymbol{x})$，并带有句法限制 ，以防止进行超出有限寄存器容量的任意数据比较。
-

▶ 我们证明，$\mathcal{L}(\mathrm{MSO}_{\mathrm{S}}[<, \Sigma, \mathbb{A}]) = \mathcal{L}(\mathrm{NRA}^{\mathrm{G}}[\Sigma, \mathbb{A}])$，前提是原子 $\mathbb{A}$ 允许消去强猜测——这是我们形式化的一项技术性质。标准论域满足这一条件，包括等式原子 $\mathbb{A}_{=}$ 和稠密序原子 $\mathbb{A}_{<}$；该条件还意味着 $\mathrm{MSO}_{\mathrm{S}}[<]$ 在这些原子上的可满足性是可判定的。我们讨论了针对 $\omega$-词和 $\mathrm{NRA}^{\mathrm{G}}$ 的子类对 $\mathrm{MSO}_{\mathrm{S}}[<]$ 作出的自然调整。

**意义与展望。** 这些结果为 NRA 提供了一套描述性理论，与有限字母表情形相对应。此外，这种新的刻画为解决数据语言理论中的开放问题提供了工具。例如，$\mathrm{MSO}_{\mathrm{S}}[<]$ 的逻辑形式体系或许能提供缺失的见解，从而解决 Colcombet猜想：关于能否用无歧义寄存器自动机 [8][1]分离互不相交的 $\mathrm{NRA}^{\mathrm{G}}$ 语言。此外，我们还留下一个开放问题：是否存在某种对 $\mathrm{MSO}_{\mathrm{S}}[<, \Sigma, \mathbb{A}]$ 的增强，既保留可判定的可满足性，又能超越 $\mathrm{NRA}^{\mathrm{G}}$ 语言类。

[1] 在所引用的论文中， [8] 被表述为一个定理，后来又恢复为一个猜想。

## 2 预备知识

我们使用标准记法表示自然数 $\mathbb{N}$、正自然数 $\mathbb{N}_+$ 或有理数 $\mathbb{Q}$。自然数的值域记作 $[i,j]_{\mathbb{N}} := \mathbb{N} \cap [i,j]$ 和 $[j]_{\mathbb{N}} := [1,j]_{\mathbb{N}}$。

**词。** 固定一个字母表 $\Sigma$ 和一个词 $w = a_1 a_2 \cdots a_n \in \Sigma^*$。我们记作 $w[i] := a_i$；对于一个区间 $I = [i,j]_{\mathbb{N}}$，将子串 $a_i a_{i+1} \cdots a_j$ 记作 $w[I]$。逆序词 $a_n \cdots a_1$ 记作 $w^R$。$w$ 的长度 $n$ 记作 $|w|$，$w$ 的位置集合记作 $\mathrm{pos}(w) = [n]_{\mathbb{N}}$。长度为 0 的唯一词称为空词，记作 $\varepsilon$。我们将词族集合 $(Y^*)^X$ 视为函数空间 $X \to Y^*$。相应地，我们将词族表示为映射 $\boldsymbol{u}\colon X \to Y^*$，但分量仍采用下标记法 $u_x$。我们将拼接和子串提取操作扩展到词族。

**词结构。** 给定一个 $w \in \Sigma^*$，词结构 $\mathbb{W}(w)$ 是如下定义的关系结构：$\mathbb{W}(w) = (\mathrm{pos}(w) <, (\sigma)_{\sigma\in\Sigma})$，其中“$<$”解释为 $\mathrm{pos}(w)$ 上的自然线性序；对于每个字母 $\sigma \in \Sigma$，都有一个一元关系符号 $\sigma = \{p \in \mathrm{pos}(w) \mid w[p] = \sigma\}$，其解释为 $w$ 中标记为 $\sigma$ 的位置集合。虽然我们对 $\Sigma$ 中的字母和位置谓词都使用 $\sigma$，但其含义可由上下文明确判断。

### 2.1 正则语言的形式化方法

本文以定义正则语言的以下三种形式化方法之间的等价性为出发点。固定一个有限字母表 $\Sigma$。

非确定有限自动机。非确定有限自动机（NFA）是一个元组 $\mathrm{A} = (Q, \Sigma, Q_{\mathrm{ini}}, Q_{\mathrm{fin}}, \delta)$，其中 $Q$ 是有限状态集，$Q_{\mathrm{ini}}$ 和 $Q_{\mathrm{fin}}$ 分别是初始子集和接受子集，而 $\delta \subseteq Q \times \Sigma \times Q$。我们将 $\delta$ 中的元素称为转移，并使用箭头记法 $q \xrightarrow{a} r$ 表示 $(q,a,r) \in \delta$。A 的一个长度为 $n$、对应于 $w = a_1 a_2 \cdots a_n$ 的运行，是带标签图 $(Q,\delta)$ 中的一条 $w$-标记路径 $\pi = t_1 t_2 \cdots t_n \in \delta^*$，也就是说，一个满足以下条件的序列：对每个 $i$，$t_i$ 的形式为 $(\cdot\ a_i, \cdot)$；并且对于任意两个连续转移 $p \xrightarrow{a} q$ 和 $r \xrightarrow{b} s$，都有 $q = r$。如果存在一条 A 在 $w$ 上的运行，该运行从某个初始状态 $q_{\mathrm{ini}} \in Q_{\mathrm{ini}}$ 出发并在某个接受状态 $q_{\mathrm{fin}} \in Q_{\mathrm{fin}}$ 结束，我们就称 A 接受 $w \in \Sigma^*$。A 的语言为 $\mathrm{L(A)} = \{w \in \Sigma^* \mid \text{A 接受 } w\}$。我们用 $\mathrm{NFA}[\Sigma]$ 表示所有在 $\Sigma$ 上定义的 NFA 所构成的族。

**正则表达式。** 正则表达式在 $\Sigma$ ($\mathrm{RE}[\Sigma]$) 上是由以下文法生成的项

$$E, F ::\equiv \emptyset \mid \epsilon \mid a \in \Sigma \mid E + F \mid E \cdot F \mid E^* .$$

表达式 $E$ 的语言 $\mathcal{L}(E)$ 定义如下

$$\mathcal{L}(\emptyset) = \emptyset \qquad \mathcal{L}(\epsilon) = \{\epsilon\} \qquad \mathcal{L}(a) = \{a\}$$
$$\mathcal{L}(E+F) = \mathcal{L}(E) \cup \mathcal{L}(F) \qquad \mathcal{L}(E \cdot F) = \mathcal{L}(E) \cdot \mathcal{L}(F) \qquad \mathcal{L}(E^*) = \bigcup_{n\geq 0} \mathcal{L}(E)^n$$

其中，语言连接为 $K \cdot L := \{uv \mid u \in K, v \in L\}$，对 $K, L \subseteq \Sigma^*$ 成立；$K^n$ 表示 $K$ 与自身进行 $n$ 次拼接；$K^0 := \{\varepsilon\}$。

▶ **定理 1** (Kleene [13])。$L$(非确定有限自动机$[\Sigma]$) $= L(\mathrm{RE}[\Sigma])$ 对于任意有限 $\Sigma$。

**MSO 语法。**词结构上的单子二阶逻辑(MSO$[<,\Sigma]$)包含一阶变量 $x$、$y$，……、二阶变量 $X$、$Y$，……，其文法如下：

$$\varphi, \psi ::\equiv \exists x\ \varphi \mid \exists X\ \varphi \mid \varphi \wedge \psi \mid \varphi \vee \psi \mid \neg\varphi \mid x < y \mid X(x) \mid a(x)\,,$$

其中 $a \in \Sigma$。当 $\varphi$ 由上述文法生成时，我们记作 $\varphi \in \mathrm{MSO}[<, \Sigma]$ 。若变量在 $\varphi$ 中未被任何量词约束，则称其为自由变量。我们用 $\mathsf{free}(\varphi)$（或 $\mathsf{free}_1(\varphi)$）表示 $\varphi$的自由变量集合（分别为一阶自由变量集合）。若公式没有自由变量，则称其为句子。

**MSO语义**。 $\mathrm{MSO}[<, \Sigma]$ 公式的真值取决于解释$\mathfrak{I} = (\mathbb{W}(w)\ \nu)$，该解释由词结构 $\mathbb{W} = \mathbb{W}(w)$ 和赋值 $\nu$ 组成：$\mathsf{free}(\varphi) \to \mathsf{pos}(w) \cup \mathcal{P}(\mathsf{pos}(w))$。布尔联结词和原子公式（$x < y, X(y)$ $a(x)$）以标准方式在 $\mathbb{W}$ 上进行解释。特别地， $\mathfrak{I} \models a(x)$ 当且仅当 $\nu(x) \in a^{\mathbb{W}}$。量词的值域为 $\mathsf{pos}(w)$:

- $\mathfrak{I} \models \exists x\ \varphi$ 当且仅当存在一个位置 $p \in \mathsf{pos}(w)$，使得 $\mathbb{W}(w)\ \nu[x \mapsto p] \models \varphi$;
- $\mathfrak{I} \models \exists X\ \varphi$ 当且仅当存在一个集合 $P \subseteq \mathsf{pos}(w)$，使得 $\mathbb{W}(w)\ \nu[X \mapsto P] \models \varphi$。

对于句子 $\varphi \in \mathrm{MSO}[<, \Sigma]$，当 $\mathbb{W}(w)\ \nu_0 \models \varphi$ 时，我们记作 $\mathbb{W}(w) \models \varphi$；其中 $\nu_0$ 是空赋值。 $\varphi$ 的语言定义为 $\mathcal{L}(\varphi) = \{w \in \Sigma^* \mid \mathbb{W}(w) \models \varphi\}$。

▶ **定理 2** (Büchi、Elgot、Trakhtenbrot [7, 11, 18])。 对于每个有限 $\Sigma$，可由单子二阶逻辑$[<, \Sigma]$ 定义的语言，恰好是 $\Sigma$上的正则语言。

**语法糖**。 我们用 $\forall \xi\ \varphi$ 作为 $\neg\exists \xi\ (\neg\varphi)$的简写；位置相等定义为$x = y \equiv \neg(x < y \vee y < x)$；布尔常量可定义为 $\top \equiv \exists x\ x = x \vee \neg\exists x\ x = x$和 $\bot \equiv \neg\top$。此外，我们还可以添加一些容易用单子二阶逻辑定义的关系，例如子集关系 $S' \subseteq S$。

## 2.2 无限字母表上的词

虽然词和语言的定义并不要求字母表有限，但我们在第 2.1 节回顾的所有基本语言定义形式都严格依赖这一假设。例如，对于由单字母词组成的语言 $\Sigma$ ，有限自动机必须包含针对每个 $a \in \Sigma$的转移；如果令 $\Sigma$ 为无限集，转移关系也会变成无限的。解决这一问题的一种方法是为字母表添加更多结构，并允许模型对字母进行不那么精确的查询。这便引出了原子和寄存器自动机的领域。

**原子**。 我们将任何关系结构 $\mathbb{A} = (U, R_1, \ldots, R_\ell)$ 称为原子，其论域 $U$ 是可数无限集，签名中包含有限个关系符号。我们假设其中一个 $R_i$ 是论域元素的相等关系。我们有意将 $\mathbb{A}$ 与其论域 $U$混为一谈，并记作 $\alpha \in \mathbb{A}$。

**数据字**。 数据字是定义在双轨字母表 $\Sigma \times \mathbb{A}$上的词，由来自 $\Sigma$ 的有限标签和来自 $\mathbb{A}$的数据值组成。通常可以使用卷积将多轨词分解为独立的轨迹。对于字母表 $A_1, \ldots, A_k$，等长词$w_1 \in A_1^*, \ldots, w_k \in A_k^*$的卷积是词 $\otimes(w_1, \ldots, w_k) \in (A_1 \times \cdots \times A_k)^*$，定义为 $\otimes(w_1, \ldots, w_k)[i] := (w_1[i], \ldots, w_k[i])$。因此，数据字可以看作是来自 $\Sigma$的标签词与来自 $\mathbb{A}$的数据序列的卷积。

▶ **例3.** 设 $\mathbb{A} = (\mathbb{N}, =)$ 和 $\Sigma = \{a, b\}$。在 $\Sigma \times \mathbb{A}$ 上的双轨词 $\otimes(w, \varpi)$ 是两条轨迹 $w \in \Sigma^*$ 和 $\varpi \in \mathbb{A}^*$ 的卷积:

$$w = \mathrm{abaab}$$
$$\varpi = 10, 42, 5, 97, 42$$
$$w = \begin{bmatrix} \mathrm{a} \\ 10 \end{bmatrix} \begin{bmatrix} \mathrm{b} \\ 42 \end{bmatrix} \begin{bmatrix} \mathrm{a} \\ 5 \end{bmatrix} \begin{bmatrix} \mathrm{a} \\ 97 \end{bmatrix} \begin{bmatrix} \mathrm{b} \\ 42 \end{bmatrix}$$

上文中， $\varpi[2] = \varpi[5] = 42$ 和 $w[3]$ 是 $a$。

**关于 $\mathbb{A}$ 的无量词公式和区域**。 设 $\mathcal{X}$ 为变量的有限集合。使用变量$\mathcal{X}$的、关于 $\mathbb{A}$ 的原子公式 集合由所有表达式 $R(x_1,\dots,x_r)$ 组成，其中 $R$是 $\mathbb{A}$ 的签名中的关系符号，且 $x_i \in \mathcal{X}$。使用变量$\mathcal{X}$ 的、关于 $\mathbb{A}$ 的无量词公式 是原子公式的布尔组合：

$$\Phi, \Psi ::\equiv \top \mid \neg\Phi \mid \Phi \wedge \Psi \mid R(x_1,\dots,x_r)\,.$$

这些公式按标准方式在一对 $(\mathbb{A}, \nu)$ 上解释，其中 $\nu\colon \mathcal{X} \to \mathbb{A}$。

对于一个词 $w = a_1 \cdots a_k \in \mathbb{A}^k$， $w$ 的无量词类型 记作 type($w$)，它是所有使用变量 $\mathcal{X}_k := \{x_1,\dots,x_k\}$的文字（原子公式或其否定）的集合；当变量被解释为 $w$ 中的数据值时，这些文字均成立。形式上，type($w$) :=$\{\phi \in \mathrm{Lit}(\mathcal{X}_k) \mid \mathbb{A}, \nu \models \phi\}$ 其中 $\nu(x_i) = a_i$。词 $w \in \mathbb{A}^*$ 的无量词区域 记作 region$_{\mathbb{A}}$($w$)，它是所有与 $w$ 具有相同无量词类型的词的集合。直观地说，region$_{\mathbb{A}}$($w$) 中的词无法通过无量词公式与 $w$ 区分。我们以自然的方式将 region$_{\mathbb{A}}$ 推广到数据字。容易看出，对于任意 $k$， $\mathbb{A}^k$ 都是有限个无量词区域的并。

**数据词结构**。 固定原子 $\mathbb{A} = (U,\ R_1^{\mathbb{A}},\ \dots\dots,\ R_\ell^{\mathbb{A}})$，并令 $\otimes(w,\ \varpi) \in (\Sigma \times U)^*$ 满足 $|w| = |\varpi|$。数据词结构是 $\mathbb{W}(w) = (\mathrm{pos}(w) <, (P_a)_{a\in\Sigma})$ 的扩张

$$\mathbb{W}(w,\varpi) := (\mathrm{pos}(w), <, (\sigma)_{\sigma\in\Sigma}, (R_i)_{i\in[\ell]_{\mathbb{N}}})$$

其中，若 $R_i$ 的元数为 $m_i$，则对于所有 $p_1,\ \dots\dots,\ p_{m_i} \in$，pos($w$)，

$$R_i(p_1,\dots,p_{m_i}) \quad \text{iff} \quad R_i^{\mathbb{A}}(\varpi[p_1],\dots,\varpi[p_{m_i}]).$$

## 2.3 寄存器自动机——一种数据词语言的形式体系

定义数据字语言的一种常见形式体系是非确定性寄存器自动机（NRA$^{\mathrm{G}}$）。固定原子 $\mathbb{A} = (U,\ R_1,\ \dots\dots,\ R_\ell)$，以及一个有限的$\mathcal{R}$，其元素称为寄存器。令 $k := |\mathcal{R}|$，并令 $\mathbb{A}_{\mathrm{nil}} := \mathbb{A} \cup \cdot\ \{$空值$\}$，其中 nil 是一个新元素，用来表示空寄存器；并将签名 $\mathbb{A}_{\mathrm{nil}}$ 扩展为包含一个一元符号 nil，其解释为{空值}。寄存器赋值是一个函数 $\mu\colon \mathcal{R} \to \mathbb{A}_{\mathrm{nil}}$。

寄存器约束。设 $\mathrm{V}_{\mathcal{R}} := \bigcup_{r\in\mathrm{R}} \{\overleftarrow{r}, \overrightarrow{r}\}$ 为表示转移前后寄存器值的变量，并设 $\overrightarrow{y}$ 表示当前输入符号。寄存器约束 $\Phi$ 是一个关于 $\mathbb{A}_{\mathrm{nil}}$ 的无量词公式，其变量集为 $\{\overrightarrow{y}\} \cup \mathrm{V}_{\mathcal{R}}$。我们将 R 的所有寄存器约束构成的集合记为 constr$_{\mathbb{A}}$(R)。为对约束求值，考虑两个赋值 $\mu, \mu'\colon \mathrm{R} \to \mathbb{A} \cup \{\mathrm{nil}\}$ 以及一个输入符号 $\alpha \in \mathbb{A}$。它们在变量 $\{\overrightarrow{y}\} \cup \mathrm{V}_{\mathcal{R}}$ 上诱导出一个组合赋值 $\nu$，定义为 $\nu(\overleftarrow{r}) = \mu(r)$， $\nu(\overrightarrow{r}) = \mu'(r)$，以及 $\nu(\overrightarrow{y}) = \alpha$。

寄存器自动机。带猜测的非确定性寄存器自动机（NRA$^{\mathrm{G}}$）是一个元组 $\mathrm{A} = (Q, \Sigma, \mathbb{A}, \mathrm{R}\ Q_{\mathrm{ini}}, Q_{\mathrm{fin}}, \delta)$，包含有限状态集 $Q$、有限字母表 $\Sigma$、初始状态 $Q_{\mathrm{ini}} \subset Q$、接受状态 $Q_{\mathrm{fin}} \subseteq Q$ 以及一组转移规则 $\delta$： $Q \times \Sigma \times \mathrm{constr}_{\mathbb{A}}(\mathrm{R}) \times Q$。转移规则 $(p, \sigma, \Phi, q) \in \delta$ 采用双箭头记法 $p \overset{\sigma,\Phi}{\Longrightarrow} q$ 书写。语义由配置$C_{\mathcal{A}} := Q \times (\mathrm{R} \to \mathbb{A}_{\mathrm{nil}})$ 上的无限转移系统 $[\![\mathrm{A}]\!] := (C_{\mathcal{A}}, \Delta_{\mathcal{A}})$ 定义，其中 $\Delta \subseteq C_{\mathcal{A}} \times \Sigma \times \mathbb{A} \times C_{\mathcal{A}}$。转移 $(p,\mu) \xrightarrow{\sigma\ \alpha} (q,\mu') \in \Delta_{\mathcal{A}}$存在，当且仅当 $\delta$ 中存在规则 $p \overset{\sigma,\varphi}{\Longrightarrow} q$，使得 $\mathbb{A}, \nu(\mu,\alpha,\mu') \models \Phi$。若 $q \in Q_{\mathrm{ini}}$，且对每个 $r \in \mathrm{R}$ 均有 $\mu(r) =$ 空值，则配置 $(q,\mu)$ 是初始配置。当 $q \in Q_{\mathrm{fin}}$ 时，配置 $(q,\mu)$ 是接受配置。运行的概念与非确定有限自动机中类似；若某次运行从某个初始配置开始并在某个接受配置结束，则该运行是接受的。语言 L(A) 由标注 $[\![\mathrm{A}]\!]$ 之运行的数据字构成。 $[\![\mathrm{A}]\!]$ 的每个运行 $\pi$ 都关联着一个转移规则序列 $t_1 \cdots t_n \in \delta^*$。我们将 $\Sigma \times \mathbb{A}$ 上的 $k$-寄存器 NRA 族记作 NRA$^{\mathrm{G}}_k[\Sigma, \mathbb{A}]$。

寄存器自动机中非确定性的变体。固定 A $\in$ $\mathrm{NRA}^{\mathrm{G}}_k$ 及其在 $w \in (A \times \Sigma)^n$ 上的运行 $\pi = (q_0, \mu_0) \xrightarrow{\sigma_1\ \alpha_1} (q_1, \mu_1) \to \cdots \to (q_{n-1}, \mu_{n-1}) \xrightarrow{\sigma_n\ \alpha_n} (q_n, \mu_n)$。若 $\xi = \mu_a(r_j)$ 不同于所有先前的寄存器值 $\mu_{a-1}$(R) 以及数据值 $\alpha_a$，则称 A 正在猜测赋值 $\mu_a$ 中第 $j$ 个寄存器的值。该猜测可以是弱的或强的，取决于从配置 $(q_a, \mu_a)$ 起的后续运行。每当对所有 $m \in [a, b]_{\mathbb{N}}$ 有 $\xi \in \mu_m$(R) 且 $\xi \notin \mu_{b+1}$(R) 时，我们称 $\xi$ 的存活区间为 $[a, b]_{\mathbb{N}}$。每当对存活区间 $[i, j]_{\mathbb{N}}$ 中的每个 $m$ 都有 $\alpha_m \neq \xi$ 时，赋值 $\mu_a$ 中第 $j$ 个寄存器之值的猜测是强的。

我们区分 $\mathrm{NRA}^{\mathrm{G}}$ 的两个受限子类：弱猜测非确定性寄存器自动机($\mathrm{NRA}^{\mathrm{W}}$) 和无猜测确定性寄存器自动机(NRA)。它们分别由转移约束 $\Phi$ 和 $\delta$ 的确定性方面的语法限制所定义。我们称$\mathcal{A} \in \mathrm{NRA}^{\mathrm{G}}[\Sigma, \mathbb{A}]$ 为弱猜测型，如果$\mathcal{A}$ 在其接受运行中不会进行强猜测。如果它在任何运行中都不会猜测某个值，则称其为无猜测型。

▶**定义 4** (强猜测消去。)。我们称原子 $\mathbb{A}$ 具有强猜测消去性质，当且仅当对于每个 $A \in \mathrm{NRA}^{\mathrm{G}}[\Sigma, \mathbb{A}]$，都存在一个弱猜测型 $B \in \mathrm{NRA}^{\mathrm{G}}[\Sigma, \mathbb{A}]$，使其识别相同的语言。

## 3 与寄存器自动机等价的新形式系统

### 3.1 数据正则表达式

固定原子 $\mathbb{A}$ 和一个有限 $\Sigma$。我们引入一种关于 $\Sigma \times \mathbb{A}$ 的数据正则表达式，它们由以下文法生成

$$E, F ::\equiv \emptyset \mid [w] \mid E + F \mid E \cdot_k F \mid E^{k,*},$$

其中 $k \in \mathbb{N}$ 和 $w \in (\Sigma \times \mathbb{A})^*$。令 $\mathrm{DRE}[\Sigma, \mathbb{A}]$ 为所有此类数据表达式的集合。$E \in \mathrm{DRE}[\Sigma, \mathbb{A}]$ 的语言 $\mathcal{L}(E)$ 定义如下：

$$\mathcal{L}(\emptyset) = \emptyset \qquad \mathcal{L}([w]) = \mathrm{region}_{\mathbb{A}}(w)$$

$$\mathcal{L}(E + F) = \mathcal{L}(E) \cup \mathcal{L}(F) \qquad \mathcal{L}(E \cdot_k F) = \mathcal{L}(E) \cdot_k \mathcal{L}(F) \qquad \mathcal{L}\left(E^{k,*}\right) = \bigcup_{n \geq 0} \mathcal{L}(E)^{k,n}$$

其中 $\cdot_k$ 是 $k$-收缩连接，定义为 $K \cdot_k L \coloneqq \{uw \mid uv \in K, v^R w \in L, |v| = k\}$对于数据词语言 $K, L$ 和 $K^{n,k}$，对于 $n > 0$ 表示 $n$ 重 $k$ 收缩连接 $K$ 与自身，而对于 $n = 0$，我们定义 $K^{0,k} \coloneqq \{vv^R \mid |v| = k\}$，即长度为 $2k$ 的回文集合。

▶**定理 5.** $L\left(\mathrm{NRA}^{\mathrm{G}}[\Sigma, \mathbb{A}]\right) = L(\mathrm{DRE}[\Sigma, \mathbb{A}])$ 对于任意有限 $\Sigma$ 和原子 $\mathbb{A}$。

⋆

### 3.2 作用域 MSO

**作用域MSO语法**。固定有限的 $\Sigma$ 和原子 $\mathbb{A} = (U, R_1, \ldots\ldots, R_\ell)$。作用域 MSO 的语法使用一阶变量 $x, y, \ldots\ldots$、二阶变量 $X, Y, \ldots\ldots$，并由以下规则生成

$$\varphi, \psi ::\equiv \exists x\, \varphi \mid \exists X\, \varphi \mid \mathfrak{S}_X\, \varphi \mid \varphi \wedge \psi \mid \varphi \vee \psi \mid \neg\varphi \mid x < y \mid X(x) \mid a(x) \mid R(\boldsymbol{x}),$$

对于所有字母 $a \in \Sigma$、关系 $R \in \{R_1, \ldots\ldots, R_\ell\}$，且其中 $\boldsymbol{x} = (x_1, \ldots\ldots, x_r)$ 与 $R$ 的元数相匹配。与 MSO 的文法（见第 2.1 节）相比，新增了两项：$\mathfrak{S}_X$ 是作用域模态，而 $R(\boldsymbol{x})$ 是数据原子公式。

$\mathfrak{S}$

$\varphi$中的一阶变量若其绑定存在量词不出现在否定的作用域内，则为顶层 变量；否则为嵌套变量。（注意，$\forall x\, \varphi \equiv \neg\exists x\, \neg\varphi$中的变量本质上都是嵌套变量。）

▶**条件 6** (良构性). 若公式 $\varphi$ 中出现的每个数据原子公式$R(\boldsymbol{x})$ 至多包含一个嵌套变量，则称其为良构的 。

▶ **例 7**。 固定 $\mathbb{A} = (\mathbb{N}, \sim)$，其中 $\sim$ 是 $\mathbb{N}$上的相等关系。公式 $\exists x\, \neg[\exists y.\ x \sim y \wedge x \neq y]$ 是良构的，因为 $x$ 处于顶层。然而， $\neg[\exists x\, \exists y.\ x \sim y \wedge x \neq y]$ 并非如此，因为外部否定使 $x$ 和 $y$ 都处于嵌套层级。模态 $\wr_X$保留变量的顶层状态；因此， $\exists X\, \wr_X\, \exists x\, \neg[\exists y.\ x \sim y \wedge x \neq y]$ 仍然是良构的。

我们将所有良构公式的集合记为 $\mathrm{MSO}_{\mathrm{S}}[<, \Sigma, \mathbb{A}]$。

**子作用域**。 设 $I = [a, b]_{\mathbb{N}}$ 且设 $X \subseteq \mathbb{N}$。我们将 $X$ 视为一组分割点，并定义通过在 $X$中的每个位置切割 $I$ 所得到的连续子区间集合 subs($I$， $X$)。形式化地，令 subs($I$， $X$) 表示所有 $[a', b']_{\mathbb{N}}$的集合，其中 $a'$， $b'+1$是排序集合 $(X \cap I) \cup \{a,\ b+1\}$中的两个相邻元素。

▶ **例 8.** 令 $I = [a, b]_{\mathbb{N}} = [10, 20]_{\mathbb{N}}$和 $X = \{5、7、11、13、17,23\}$。分割点集合 $(X \cap I) \cup \{a、\ b+1\}$ 为 $\{10、\ 11、\ 13、\ 17、\ 21\}$，因此

$$\mathrm{subs}(I, X) = \{\underset{\substack{\| \\ a}}{10}\}, \{\underset{\substack{\Cap \\ X}}{11}, 12\}, \{\underset{\substack{\Cap \\ X}}{13}, 14, 15, 16\}, \{\underset{\substack{\Cap \\ X}}{17}, 18, 19, \underset{\substack{\| \\ b+1}}{20}\}.$$

带作用域 MSO 的语义。我们在元组 $\mathfrak{I} = (\mathbb{W}(w, \varpi)\ \nu, I)$ 上解释 $\mathrm{MSO}_{\mathrm{S}}[<, \Sigma, \mathbb{A}]$ 公式，该元组由数据词结构 $\mathbb{W} = \mathbb{W}(w, \varpi)$、赋值 $\nu$: free($\varphi$) $\to$pos($w$) $\cup\cdot$ P(pos($w$)) 以及一个称为作用域的离散区间 $I = [i, j]_{\mathbb{N}} \subseteq$ pos($w$) 组成。布尔联结词和原子公式 $(x < y, X(y)\ a(x)\ R(\boldsymbol{x}))$ 按照标准方式在 $(\mathbb{W}, \nu)$ 上解释，并忽略 $I$。特别地， $\mathfrak{I} \models R(\boldsymbol{x})$ 当且仅当 $\nu(\boldsymbol{x}) \in R^{\mathbb{W}}$。量词的值域严格限制在当前作用域 $I$ 内:

- $\mathfrak{I} \models \exists x\, \varphi$ 当且仅当存在一个位置 $p \in I$使得 $\mathbb{W}(w,\ \varpi)\ \nu[x \leftarrow p]$， $I \models \varphi$；当且仅
- 当存在一个位置集合 $P \subseteq I$使得 $\mathbb{W}(w,\ \varpi)\ \nu[X \leftarrow P]$， $I \models \varphi$。

最后， $\wr_X$ 要求该公式在所有诱导子作用域上都成立:

- $\mathfrak{I} \models \wr_X\, \varphi$ iff $\mathbb{W}(w, \varpi), \nu, I' \models \varphi$ for every $I' \in \mathrm{subs}(I, \nu(X))$.

当作用域区间是整个 pos($w$) 时，我们会在解释元组中省略它，并写作$\mathbb{W}(w,\ \varpi)\ \nu \models \varphi$。当 $\varphi$是一个句子时，我们会省略赋值，并写作 $\mathbb{W}(w,\ \varpi) \models \varphi$，并将 $\varphi$的语言 定义为 $\mathcal{L}(\varphi) = \{\otimes(w,\ \varpi) \mid \mathbb{W}(w,\ \varpi) \models \varphi\}$。

▶ **定理 9.** $L\big(\mathrm{NRA}^{\mathrm{G}}[\Sigma, \mathbb{A}]\big) = L$(单子二阶逻辑$_S[<, \Sigma, \mathbb{A}]$) 适用于任意有限的 $\Sigma$和具有强猜测消去性质（依据定义 *4*）的原子 $\mathbb{A}$。

▶**推论 10**。 如果原子 $\mathbb{A}$ 具有强猜测消去性质，那么当且仅当 $\mathrm{NRA}^{\mathrm{G}}[\Sigma, \mathbb{A}]$ 的空性可判定时，单子二阶逻辑$_S[<, \Sigma, \mathbb{A}]$ 的可满足性才可判定。

▶ **引理 11**。 等式原子 $\mathbb{A}_= = (\mathbb{N}, =)$ 和稠密序原子 $\mathbb{A}_< = (\mathbb{Q}, <)$ 都具有强猜测消去性质。

第 5 节讨论如何以使逻辑更加复杂为代价，推广关于强猜测消去的假设。

## 4 主要定理的证明

我们简要概述定理 5 和 9 背后的核心思想，而详细证明则放在附录（附录 A 和 B）中。

## 4.1 表达式与自动机

一个 $k$-寄存器自动机只能将 $k$ 数据值传输过输入中的一个切分点。DRE 的算子 $\cdot_k$ 正是为准确体现这一点而设计的：一个词属于 $K \cdot_k L$，当且仅当它可以分割为 $uw$，使得存在一个重叠 $v \in (\Sigma \times \mathbb{A})^k$，满足 $uv \in K$ 和 $v^R w \in L$。直观而言，$v$ 是一个长度为 $k$ 的接口，用于传递在右侧继续进行识别所需的有限个数据值。我们给出了 DRE 与 $\mathrm{NRA}^{\mathrm{G}}$ 之间的归约，因为带绑定子的表达式 [6] 这种形式在概念上难以用 $k$-收缩连接来模拟。

**从 数据正则表达式 到 $\mathbf{NRA}^{\mathbf{g}}$。** 编译过程是对表达式进行结构归纳。唯一非经典的步骤是 $\cdot_k$（以及相应的（$k$, $*$）迭代）：为实现存在量化的重叠，自动机会猜测 接口词 $v$，将其数据值非确定性地存入寄存器，然后在启动右侧分量时反向重放 该重叠。

**从 $\mathbf{NRA}^{\mathbf{g}}$ 到 数据正则表达式。** 这里的主要思路是将有限控制 与数据一致性分离。我们首先忽略数据，将一次运行视为由转移规则构成的有限字母表上的一个词；控制一致的规则序列集合是正则的，因此可以用普通正则表达式描述（借助 Kleene）。然后，我们通过将每条转移规则替换为一个局部块语言 来重新引入数据；该语言的长度为 $2k{+}1$，其数据轨道编码了 $k$寄存器的前赋值、当前数据值和后赋值，并将转移约束作为关于该元组的无量词条件进行检查。

## 4.2 逻辑与自动机

等价关系 $\mathcal{L}$(单子二阶逻辑$_{\mathrm{S}}[<]) = \mathcal{L}\big(\mathrm{NRA}^{\mathrm{G}}\big)$ 面临一个核心障碍：如果逻辑可以自由比较无关位置上的数据值，可满足性很快就会变得不可判定，而且无论如何，其表达能力都超过了有限个寄存器。我们的两项关键设计正是为了解决这一问题。首先，良构性限制每个数据原子至多包含一个嵌套（一阶）变量，从而避免否定之下出现任意的位置间依赖关系。其次，模态 $\textcircled{2}_X$ 提供了对子区间进行受控递归的机制，使我们能够在作用域中施加性质，而这些作用域大致对应于寄存器的存活区间。

**从 单子二阶逻辑$_{\mathbf{S}}[<, \Sigma, \mathbb{A}]$ 到 $\mathbf{NRA}^{\mathbf{g}}$。** 我们的思路是消除作用域特征，并提取位置之间的数据值依赖关系，直到只剩下局部数据测试。通过引入辅助轨道来处理顶层变量：在每个作用域上，这些轨道携带一个选定的见证值，并使其在整个作用域内传播。这样，非局部引用（“使用当前作用域内为 $x$ 选定的值”）就转化为对局部轨道的访问。

接下来，$\textcircled{2}_X \varphi$ 被展开为对 MSO 可定义子作用域的量化，以及这些子作用域上若干个 $\varphi$ 副本的合取；经过这一步，翻译后的公式中的数据谓词变为局部测试，也就是说，它们只检查（扩展后）词中的当前位置。局部测试公式可以由 0 寄存器非确定性寄存器自动机识别：每个位置在数据签名上只有有限多个无量词类型，因此我们可以将问题化简为一个有限字母表 MSO 公式，再化简为有限自动机，最后将这些类型重新解释为关于当前数据值的无量词约束。辅助数据值轨道通过非确定性投影消除：NRA 在线地在寄存器中猜测这些轨道并检查局部约束，同时确保“见证轨道”在预期的片段内保持恒定。由此得到一个关于原字母表 $\Sigma \times \mathbb{A}$ 的 $\mathrm{NRA}^{\mathrm{G}}$。

**从弱猜测 $\mathbf{NRA}^{\mathbf{g}}$ 到 单子二阶逻辑$_{\mathbf{S}}[<]$。** 反向地，我们假设采用弱猜测（在强猜测消去后不失一般性），并首先将其规范化为一种简单形式

，在这种形式下，寄存器更新是显式的，且值不会在寄存器之间调换。单子二阶逻辑部分猜测一个运行骨架：二阶变量为每个位置标记所选的转移规则，并确保控制一致性。剩下的挑战是表达所有寄存器约束均成立，同时不违反良构性。对于每个寄存器值变量（以及当前数据值），运行骨架都会将词划分为若干存活区间，在每个区间内，该值保持不变。弱猜测保证了可见性：每个携带某个值的存活区间内，都有一个见证位置，该位置上的输入数据轨道包含此值。随后，我们利用 $X$ 作为逐区间选择见证位置的机制，而不是将其视为全称量化位置上的函数。粗略地说，我们将作用域划分为一族互不相交的“操作区间”，并在每个这样的区间内，针对当前尚未获得见证的值，存在地选取见证位置，然后递归处理。正是这些操作区间简单的不相交／交替性质使这种分段方式可行：它让我们能够在许多区间内独立选择见证，而不会相互干扰。一旦为所有相关值确定了见证，最后的检查就很简单：我们对当前位置作全称量化，并通过引用固定的见证位置来评估每个转移守卫，因此每个数据原子只包含一个嵌套变量。由此得到一个与该自动机等价的单子二阶逻辑 $_{\mathrm{S}}[<]$ 句子。

# 5 扩展和特殊情形

## 5.1 无限词

固定一个有限字母表 $\Sigma$ 和原子 $\mathbb{A}$。关于 $\Sigma \times \mathbb{A}$ 的无限 词是一个序列 $(a_i)_{i\in\mathbb{N}_+}$，其中 $a_i \in \Sigma \times \mathbb{A}$。我们用 $(\Sigma \times \mathbb{A})^\omega$ 表示无限数据词的集合。将位置集合取为 $\mathbb{N}_+$，数据词结构的概念即可原样扩展到这种情形。

寄存器自动机在处理 $\omega$-词时无需进行语法上的修改；只需将接受条件替换为标准的 *Büchi* 条件。对于一个 $\in \mathrm{NRA}^{\mathrm{G}}[\Sigma, \mathbb{A}]$，我们将$\mathcal{L}^\omega(()\mathcal{A})$定义为这样的 $\omega$-词的集合：它们存在一个从初始配置开始、并无限多次访问 $Q_{\mathrm{fin}}$ 中的接受状态的无限运行。

**作用域 MSO（适用于 $\omega$-词）**。在无限数据词结构上，$\mathrm{MSO}_{\mathrm{S}}[<, \Sigma, \mathbb{A}]$ 的语义保持不变：公式在位置 $\mathbb{N}_+$上进行解释，其中作用域仍与之前一样是 $\mathbb{N}_+$ 的区间。运行与 MSO 可定义的运行骨架之间的对应关系同样适用：任何一个 $\in \mathrm{NRA}^{\mathrm{G}}[\Sigma, \mathbb{A}]$ 的无限运行，都可以通过 $\mathrm{MSO}_{\mathrm{S}}[<, \Sigma, \mathbb{A}]$ 加以描述，方法是在每个位置约束转移标记，并添加常见的 Büchi 重现条件。

对于反向方向，我们重用引理 19 中的翻译：对于每个句子$\varphi_0 \in$ 单子二阶逻辑 $_{\mathrm{S}}[<, \Sigma, \mathbb{A}]$，我们构造一个公式 $F(\varphi_0)$，使得 $\mathcal{L}^\omega(()\varphi_0)$ 是 $\mathcal{L}^\omega(()F(\varphi_0))\cap\mathcal{L}^\omega()\big(\bigwedge_{x\in\mathcal{U}_\uparrow}\eta_x\big)$，的存在量化轨道投影；此外，$F(\varphi_0)$ 属于有限字母表 MSO 片段（基于扩展字母表 $\Sigma''$）。因此，根据 Büchi–Elgot–Trakhtenbrot 定理的 $\omega$-正则类比版本 [16]，$F(\varphi_0)$ 可由在$\Sigma''$上的 Büchi 自动机识别。按照 B.1 节中的相同乘积投影构造，我们得到一个带有 Büchi 接受条件的 $\mathrm{NRA}^{\mathrm{G}}[\Sigma, \mathbb{A}]$，它识别 $\mathcal{L}^\omega(()\varphi_0)$。特别地：

▷断言 12. 对于每个有限 $\Sigma$ 和原子 $\mathbb{A}$，

$$\mathcal{L}^\omega\big(\mathrm{MSO}_{\mathrm{S}}[<, \Sigma, \mathbb{A}]\big) \;=\; \mathcal{L}^\omega\big(\mathrm{NRA}^{\mathrm{G}}[\Sigma, \mathbb{A}]\big).$$

## 5.2 需要强猜测的原子

弱猜测对于我们的主要逻辑刻画至关重要：$\mathrm{MSO}_{\mathrm{S}}[<, \Sigma, \mathbb{A}]$ 只能引用输入词中出现的数据值，而具有强猜测能力的 NRA 可能

引入输入轨道上从未出现过的新数据值。为了容纳这种行为，可以将（引理 16 中使用的）$\mathrm{MSO}_{\mathrm{S}}[<,\Sigma,\mathbb{A}]$ 的多轨变体扩展为允许对新的数据轨道进行顶层存在量化。形式上，将语法扩展为形如 $\exists\ t[\cdot]\,\varphi$ 的量词，其中 $t[\cdot]$ 是一个新的轨道访问器项，其取值范围为 $\mathbb{A}$。

语义。设 $\boldsymbol{u}\colon A\to\Sigma^{L}$ 和 $\boldsymbol{v}\colon B\to\mathbb{A}^{L}$ 是长度相等 $L$ 的轨迹族。那么

$$\mathbb{W}(\boldsymbol{u},\boldsymbol{v}),\nu,I\models\exists\ t[\cdot]\ \varphi\quad\text{iff}\quad\exists\ v'\in\mathbb{A}^{L}\ \text{such that}\ \mathbb{W}(\boldsymbol{u},\otimes(\boldsymbol{v},v')),\nu,I\models\varphi,$$

其中在 $\varphi$ 中，术语 $t[x]$ 对每个一阶变量 $x$ 都解释为 $v'[\nu(x)]$。直观而言，$\exists\ t[\cdot]$ 为逻辑提供了一组新鲜数据值，对应于自动机执行的强猜测。

## 5.3 无猜测的 NRA

只需对逻辑做简单改进，即可刻画 NRA 形式体系。我们将 $\mathrm{MSO}'_{\mathrm{S}}[<,\Sigma,\mathbb{A}]$ 记作一种变体：它以作用域量词 $SX,y$ 取代作用域模态，并相应调整顶层变量的概念：

- $\mathbb{W},\nu,I\models SX,y\,\varphi$ 当且仅当对于每个子作用域 $I'=[a,b]_{\mathbb{N}}\in\mathsf{subs}(I,\nu(X))$，都有 $\mathbb{W},\nu[y\leftarrow a],I'\models\varphi$；若一阶变量由某个 $,y$ 绑定，且该绑定在语法树中不处于否定之下，则
- 该变量称为顶层变量；否则称为嵌套变量。数据原子的良构性条件保持不变：$R(\boldsymbol{x})$ 的每次出现至多包含一个嵌套变量。
-

与前文相同，作用域旨在对应运行中的存活区间。由 $x,y$ 绑定的变量 $y$将每个寄存器值固定在存活区间的左端点。直观来说，这符合“不可猜测”的规则：值可以从较早的位置向前复制（或取自 $\overset{\triangledown}{y}$），但不能凭空猜出。

### 参考文献

1 Henrik Björklund 和 Thomas Schwentick。《关于数据语言正则性概念》。理论计算机科学，411(4):702–715，2010。《计算理论基础》。网址：https://www.sciencedirect.com/science/article/pii/S0304397509007518, doi:10.1016/j.tcs.2009.10.009。2 Mikołaj Bojańczyk。《略微无限的集合》。书稿，2019年9月11日版本，2019年。网址：https://www.mimuw.edu.pl/~bojan/paper/atom-book。3 Mikołaj Bojańczyk、Claire David、Anca Muscholl、Thomas Schwentick 和 Luc Segoufin。《数据词上的二变量逻辑》。*ACM*计算逻辑汇刊，12(4)，2011年7月。doi:10.1145/1970398.1970403。4 Mikołaj Bojańczyk 和 Rafał Stefański。《无限字母表上的单次使用自动机与转换器》。载于 Artur Czumaj、Anuj Dawar 和 Emanuela Merelli（编），第*47*届自动机、语言与程序设计国际研讨会（*ICALP 2020*），第168卷，莱布尼茨国际信息学会议录（*LIPIcs*），第113:1–113:14页，达格施图尔，德国，2020年。Schloss Dagstuhl – Leibniz-Zentrum für Informatik。doi:10.4230/LIPIcs.ICALP.2020.113。5 Mikołaj Bojańczyk、Bartek Klin 和 Sławomir Lasota。《名义集合中的自动机理论》。《计算机科学中的逻辑方法》，第10卷，第3期，2014年8月。网址：https://lmcs.episciences.org/1157, doi:10.2168/LMCS-10(3:4)2014。

6 Paul Brunet 和 Alexandra Silva。《名义自动机的克莱尼定理》。载于 Christel Baier、Ioannis Chatzigiannakis、Paola Flocchini 和 Stefano Leonardi（编），第46届自动机、语言与程序设计国际研讨会（ICALP 2019），第132卷，《莱布尼茨国际信息学会议录》（LIPIcs），第107:1–107:13页，达格施图尔，德国，2019年。Schloss Dagstuhl – Leibniz-Zentrum für Informatik。 doi:10.4230/LIPIcs.ICALP.2019.107。7 J. Richard Büchi。《弱二阶算术与有限自动机》。《数学逻辑季刊》，6(1-6):66–92，1960年。网址：https://onlinelibrary.wiley.com/doi/abs/10.1002/malq.19600060105，doi:10.1002/malq.19600060105。8 Thomas Colcombet。《自动机理论中的无歧义性》。载于 Jeffrey Shallit 和 Alexander Okhotin（编），《形式系统描述复杂性》，第3–18页，查姆，2015年。施普林格国际出版社。9 Thomas Colcombet、Clemens Ley 和 Gabriele Puppis。《具有刚性守卫数据测试的逻辑》。《计算机科学中的逻辑方法》，11(3)，2015年。doi:10.2168/LMCS-11(3:10)2015。10 Loris D'Antoni、Tiago Ferreira、Matteo Sammartino 和 Alexandra Silva。《符号寄存器自动机》。载于 Isil Dillig 和 Serdar Tasiran（编），《计算机辅助验证》，第3–21页，查姆，2019年。施普林格国际出版社。11 Calvin C Elgot。《有限自动机设计的判定问题及相关算术》。《美国数学学会汇刊》，98(1):21–51，1961年。12 Michael Kaminski 和 Nissim Francez。《有限记忆自动机》。《 *[1990] 31*届计算机科学基础年会论文集》，第683–688页，第2卷，1990年。网址：https://api.semanticscholar.org/CorpusID:205095872。13 Stephen Cole Kleene。《神经网络与有限自动机中的事件表示》，第34卷。普林斯顿大学出版社，普林斯顿，1956年。14 Bartek Klin、Sławomir Lasota 和 Szymon Toruńczyk。《对数据语言而言，非确定性与余非确定性蕴含确定性》。载于 Stefan Kiefer 和 Christine Tasson（编），《软件科学与计算结构基础》，第365–384页，查姆，2021年。施普林格国际出版社。15 Frank Neven、Thomas Schwentick 和 Victor Vianu。《无限字母表上字符串的有限状态机》。《ACM计算逻辑汇刊》，5(3):403–435，2004年7月。 doi:10.1145/1013560.1013562。16 J. Richard Büchi。《判定问题研讨会：受限二阶算术中的一种判定方法》。载于 Ernest Nagel、Patrick Suppes 和 Alfred Tarski（编），《逻辑、方法论与科学哲学》，第44卷，《逻辑与数学基础研究》，第1–11页。Elsevier，1966年。网址： https://www.sciencedirect.com/science/article/pii/S0049237X09705646， doi:10.1016/S0049-237X(09)70564-6。17 Luc Segoufin。《无限字母表上的字与树的自动机和逻辑》。载于 Zoltán Ésik（编），《计算机科学逻辑》，第41–57页，柏林、海德堡，2006年。Springer Berlin Heidelberg。18 B. A. Trakhtenbrot。《有限自动机与一元谓词逻辑》。《西伯利亚数学杂志》，3:103–131，1962年。

## A 表达式与自动机的等价性

在本节余下部分，固定一个有限字母表 $\Sigma$ 和原子 $\mathbb{A} = (U, R_1, \ldots, R_\ell)$。

### A.1 从表达式到自动机

本节中，我们证明定理 5 的一个方向。

▶ **引理 13**。对于每个数据正则表达式 $E \in$ 数据正则表达式$[\Sigma, \mathbb{A}]$，都存在一个 $A \in \mathrm{NRA}^{\mathrm{G}}[\Sigma, \mathbb{A}]$，使得 $L(E) = L(A)$。

带有 $\varepsilon$-转移的 NRA。$\mathrm{NRA}^{\mathrm{G}}$ 的基础模型（如第 2.3 节所述）每步恰好读取一个输入符号。对于下面的构造，允许不消耗输入的 $\epsilon$-转移会很方便。带有 $\varepsilon$-转移的非确定性寄存器自动机（$\mathrm{NRA}^{\mathrm{G}}_{\varepsilon}[\Sigma, \mathbb{A}]$）是一个元组 $\mathrm{A} = (Q, \Sigma, \mathbb{A}, \mathrm{R}\ Q_{\mathrm{ini}}, Q_{\mathrm{fin}}, \delta, \delta_\varepsilon)$，其中 $(Q, \Sigma, \mathbb{A}, \mathrm{R}\ Q_{\mathrm{ini}}, Q_{\mathrm{fin}}, \delta)$ 是前文所述的 $\mathrm{NRA}^{\mathrm{G}}$，而 $\delta_\varepsilon \subseteq Q \times \mathrm{constr}_{\mathbb{A}}(\mathrm{R}) \times Q$ 是一组有限的 $\varepsilon$- 转移规则，其约束不使用变量 $\overset{\triangledown}{y}$。我们将 $(p, \sigma, \Phi, q) \in \delta$ 记作 $p \overset{\sigma,\Phi}{\Longrightarrow} q$，并将 $(p, \Phi, q) \in \delta_\varepsilon$ 记作 $p \overset{\Phi}{\Longrightarrow} q$。由此导出的无限转移系统使用与第 2.3 节相同的配置 $C_{\mathcal{A}} = Q \times (\mathrm{R} \to \mathbb{A}_{\mathrm{nil}})$。对于数据字 $u = (\sigma_1, \alpha_1) \cdots (\sigma_n, \alpha_n) \in (\Sigma \times \mathbb{A})^*$，一次运行是一个由消耗输入的转移和 $\varepsilon$-转移交替组成的序列：

- 消耗输入的转移：$(p, \mu) \xrightarrow{\sigma\ \alpha} $，当且仅当存在 $p \overset{\sigma,\Phi}{\Longrightarrow} q \in \delta$ 且 $\nu(\mu, \alpha, \mu') \models \Phi$，$\varepsilon$-转移：$(p, \mu) \xrightarrow{\varepsilon} (q, \mu')$ 当且仅当存在 $p \overset{\Phi}{\Longrightarrow} q \in \delta_\varepsilon$ 且 $\nu(\mu, \mu') \models \Phi$，其中 $\nu(\mu, \mu')$ 的定义方式与 $\nu(\mu, \cdot\ \mu')$ 类似，但不指定 $\overset{\triangledown}{y}$ 的值。

接受的定义与通常相同：如果存在一条由 $u$ 标记的运行，从某个 $(q_{\mathrm{ini}}, \mu_{\mathrm{ini}})$ 出发，并以某个 $(q_{\mathrm{fin}}, \mu)$ 结束，其中 $q_{\mathrm{fin}} \in Q_{\mathrm{fin}}$，则接受 $u$。

▶ **引理 14** ($\varepsilon$-消除)。固定 $N \in \mathbb{N}$。对于每个 $A \in \mathrm{NRA}^{\mathrm{G}}_{\varepsilon}[\Sigma, \mathbb{A}]$，若其连续执行的 $N\varepsilon$-转移至多为 $N$ 次，则存在 $B \in \mathrm{NRA}^{\mathrm{G}}[\Sigma, \mathbb{A}]$，使得 $L(A) = L(B)$。

**证明思路**。这是标准的 $\varepsilon$-消除构造，在转移规则层面应用。需要对连续执行的 $N \in \mathbb{N}$ 数量设定 $\varepsilon$-步的上界，因为 $\varepsilon$-转移之间经过的寄存器赋值需要存储在额外寄存器中。

直观来说，将任意“$N$ $\varepsilon$-步，接着一个输入步，再接着 $N$ $\varepsilon$-步”的模式替换为单个输入步，其约束是相关约束的（有限）合取，并为中间赋值使用新鲜的寄存器副本；自动机通过存在量化猜测中间寄存器值。◀

因此，只需构造一个带有 $\varepsilon$-转移的 $\mathrm{NRA}^{\mathrm{G}}$，并确保这些转移在消耗输入的步之间不会连续使用超过有界次数；随后由引理 14 得到一个 $\mathrm{NRA}^{\mathrm{G}}$。

**引理 13 的证明**。我们定义一个映射 $A$，通过结构归纳法将数据正则表达式 $E$ 编译为$A(E) \in \mathrm{NRA}^{\mathrm{G}}_{\varepsilon}[\Sigma, \mathbb{A}]$，并确保，从而有 $\mathcal{L}(\mathcal{A}_E) = \mathcal{L}(E)$。应用引理 14 可得到所需的 $\mathrm{NRA}^{\mathrm{G}}$。

全文中，我们约定每个转移约束 $\Phi \in \mathrm{constr}_{\mathbb{A}}(\mathcal{R})$ 都隐含地保证：未在 $\Phi$ 中提及的寄存器 $r \in \mathcal{R}$ 的值保持不变，即 $\overset{\triangleright}{r} = \overset{\triangleleft}{r}$。

**情形1** ($E = \emptyset$)。设 $A(E)$ 没有接受状态。那么 $\mathcal{L}(A(E)) = \emptyset = \mathcal{L}(E)$。

情形2（$E = [\otimes(w, \varpi)]$）。设 $\mathrm{R} = \{r_1, \ldots, r_n\}$。定义 $A(E) = (Q, \Sigma, \mathbb{A}, \mathrm{R}, \{0\}\ Q_{\mathrm{fin}}, \delta, \emptyset)$，其中 $Q = [0, n]_{\mathbb{N}}$ 且 $Q_{\mathrm{fin}} = \{n\}$。对于每个 $i \in [n]_{\mathbb{N}}$，写下规则 $i - 1 \xrightarrow{\sigma_i, \Phi_i} i$，其中

$$\Phi_i := \left(\overset{\triangleright}{r}_i = \overset{\triangledown}{y}\right) \wedge \begin{cases} \top & \text{if } i < n, \\ \bigwedge_{\psi \in \mathrm{type}_{\mathbb{A}}(\varpi)} \psi(\overset{\triangleright}{r}_1, \ldots, \overset{\triangleright}{r}_n) & \text{if } i = n. \end{cases}$$

显然，$A(E)$ 恰好读取 $n$ 个输入符号，其数据轨道为 $\Sigma$ 轨 $\sigma_1 \cdots \sigma_n$，将数据轨道存入 $(r_1, \ldots, r_n)$，最后检查所存元组是否属于区域 $\mathbb{A}(\varpi)$。因此 $\mathcal{L}(A(E)) = $ 区域 $\mathbb{A}(\otimes(w, \varpi)) = \mathcal{L}(E)$。

**情形 3** $(E = E' + E'')$。 利用标准乘积构造即可直接处理此情形 .

情形 4 $(E = E' \cdot_k E'')$。根据引理 14，我们假设子表达式对应的自动机不使用 $\varepsilon$-转移。令 $\mathrm{A}' = A(E') = (Q', \Sigma, \mathbb{A}, \mathrm{R}', Q'_{\mathrm{ini}}, Q'_{\mathrm{fin}}, \delta')$，$\mathrm{A}'' = A(E'') = (Q'', \Sigma, \mathbb{A}, \mathrm{R}'', Q''_{\mathrm{ini}}, Q''_{\mathrm{fin}}, \delta'')$，且 $\mathrm{R}'$、$\mathrm{R}''$ 不相交。令 $\mathrm{R}''' = \{s_1, \ldots, s_k\}$ 为新寄存器，用于存储猜测的重叠词的数据部分。我们构造 $A(E) = (Q, \Sigma, \mathbb{A}, \mathrm{R}\ Q_{\mathrm{ini}}, Q_{\mathrm{fin}}, \delta, \delta_\varepsilon)$，使其分四个阶段运行：

(P1) 在真实输入前缀 $u$上模拟$\mathcal{A}'$；(P2) 使用 $\varepsilon$ 步猜测一个重叠词 $v \in (\Sigma \times \mathbb{A})^k$，并继续在 $v$ 上模拟$\mathcal{A}'$，直到其接受；然后，(P3) 通过使用 $\varepsilon$ 步向$\mathcal{A}''$ 输入 $v^R$ 来开始模拟；最后，(P4) 在剩余的真实输入后缀 $w$上运行$\mathcal{A}''$。

令 $\Sigma^{\leq k} = \bigcup_{j=0}^{k} \Sigma^j$。定义状态空间

$$\begin{aligned} Q := \ & (Q' \times \{\mathtt{P1}\}) \cup (Q' \times [0, k]_{\mathbb{N}} \times \Sigma^{\leq k} \times \{\mathtt{P2}\}) \cup \\ & (Q'' \times \{\mathtt{P4}\}) \cup (Q'' \times [0, k]_{\mathbb{N}} \times \Sigma^{k} \times \{\mathtt{P3}\}). \end{aligned}$$

直观地说：

- $(q, \mathtt{P1})$ 和 $(q, \mathtt{P4})$ 分别在真实输入上模拟$\mathcal{A}'$ 和$\mathcal{A}''$；$(q, j, v, \mathtt{P2})$ 猜测了 $j$ 个重叠字母，记
- 住 $v \in \Sigma^j$，并在猜测的重叠部分上模拟$\mathcal{A}'$；而 $(q, j, v, \mathtt{P3})$ 反向重放记住的 $v \in \Sigma^k$，以模拟$\mathcal{A}''$。
-

令 $Q_{\mathrm{ini}} = \{(q'_{\mathrm{ini}}, \mathtt{P1}) \mid q_{\mathrm{ini}} \in Q'_{\mathrm{ini}}\}$ 和 $Q_{\mathrm{fin}} = \{(q_{\mathrm{fin}}, \mathtt{P4}) \mid q_{\mathrm{fin}} \in Q''_{\mathrm{fin}}\}$。寄存器集为$\mathcal{R} = \mathcal{R}' \cup \cdot\ \mathcal{R}'' \cup \cdot\ \mathcal{R}'''$。我们按如下方式构造 $\delta$ 和 $\delta'$。

- 阶段 P1 和 P4 很容易实现：对于规则 $p \xrightarrow{\sigma, \Phi} q \in \delta'$，添加输入规则 $(p, \mathtt{P1}) \xrightarrow{\sigma\ \Phi} (q, \mathtt{P1})$，其中 $\Phi$ 可视为关于 $\mathcal{R}$ 的约束（它只涉及$\mathcal{R}'$中的寄存器）。我们为$\mathcal{A}''$ 和 P4添加类似的转移规则。
- 从 P1 切换到 P2。对于每个 $q \in Q'$，添加 $\varepsilon$-规则 $(q, \mathtt{P1}) \xrightarrow{\top} (q, 0, \varepsilon, \mathtt{P2})$。
- 阶段 P2。对于每个 $j \in [0, k-1]_{\mathbb{N}}$、每个 $v \in \Sigma^j$、每个 $\sigma \in \Sigma$以及每条规则 $p \xrightarrow{\sigma, \Phi} q \in \delta'$，令 $\Phi \equiv \Phi[\overset{\triangledown}{y} \leftarrow \overset{\triangleright}{s}_{j+1}] \wedge (\overset{\triangleright}{s}_{j+1} = \overset{\triangleleft}{s}_{j+1} \vee \overset{\triangleright}{s}_{j+1} \neq \overset{\triangleleft}{s}_{j+1})$，并添加 $\varepsilon$-规则

$$(p, j, v, \mathtt{P2}) \xrightarrow{\Phi'} (q, j+1, v\sigma, \mathtt{P2}).$$

具体而言，这使自动机能够猜测一个新的 $s_{j+1}$ 值，并在状态中记住猜测的 $\Sigma$字母。

- 从 P2 切换到 P3：经过 $k$ 次猜测后，要求接受$\mathcal{A}'$ 并跳转到重放阶段：对于每个 $v \in \Sigma^k$、每个 $q'_{\mathrm{fin}} \in Q'_{\mathrm{fin}}$ 和 $q''_{\mathrm{ini}} \in Q''_{\mathrm{ini}}$，以及每个 $\sigma \in \Sigma$，添加 $(q'_{\mathrm{fin}}, k, v, \mathsf{P2}) \xRightarrow{\top} (q''_{\mathrm{ini}}, 0, v, \mathsf{P3})$。
- 阶段 P3：重放 $v^R$ 以启动 $\mathcal{A}''$。固定 $v = \sigma_1 \cdots \sigma_k \in \Sigma^k$。对于每个 $j \in [0, k-1]_{\mathbb{N}}$ 以及每条规则 $p \xrightarrow{\sigma_{k-j},\Phi} q \in \delta''$，令 $\Phi' \equiv \Phi[\overset{\triangledown}{y} \leftarrow \overset{\triangleleft}{s}_{k-j}]$ 添加该 $\varepsilon$-规则

$$(p, j, v, \mathsf{P3}) \xRightarrow{\Phi} (q, j+1, v, \mathsf{P3}).$$

（因此，第 $j$ 次重放步会强制猜测数据 $\overset{\triangledown}{y}$ 等于存储的重叠数据。）从 P3 切换到 P4：
- 对于每个 $q \in Q''$ 和每个 $\sigma \in \Sigma$，添加 $(q, k, v, \mathsf{P3}) \xRightarrow{\top} (q, \mathsf{P4})$。

**正确性**。我们证明 $\mathcal{L}(A(E)) = \mathcal{L}(E' \cdot_k E'')$。
方向“$\subseteq$”。设 $uw \in \mathrm{L}(A(E))$，并固定一条接受运行，使得 $u$ 是运行停留在 P1-分量时所消耗的 $uw$ 的极大前缀。随后，运行执行切换，并在 P2中恰好进行 $k$ $\varepsilon$ 次转移，从而确定某个 $\sigma_1 \cdots \sigma_k \in \Sigma^k$，并将某些数据值 $\alpha_1, \ldots, \alpha_k \in \mathbb{A}$ 存储在 $s_1, \ldots, s_k$ 中。设 $v = (\sigma_1, \alpha_1) \cdots (\sigma_k, \alpha_k) \in (\Sigma \times \mathbb{A})^k$。根据构造，将该运行投影到 R′-寄存器上，可见 A′ 在 $uv$上有一条接受运行，因此根据归纳假设，$uv \in \mathrm{L}(A') = \mathrm{L}(E')$。接下来，在 P3 期间，自动机将 $v^R$（通过 $\varepsilon$-转移）输入 A″，然后在真实输入上模拟 A″并消耗 $w$，最终到达一个终态。因此 $v^R w \in \mathrm{L}(A'') = \mathrm{L}(E'')$。由此，根据 $k$-收缩连接的定义，$x = uw \in \mathrm{L}(E') \cdot_k \mathrm{L}(E'') = \mathrm{L}(E' \cdot_k E'')$。

方向“$\supseteq$”。反过来，设 $uw \in \mathrm{L}(E' \cdot_k E'')$，且对于某个 $v \in (\Sigma \times \mathbb{A})^k$，有 $uv \in \mathrm{L}(E')$ 和 $v^R w \in \mathrm{L}(E'')$。根据归纳假设，A′ 接受 $uv$，A″ 接受 $v^R w$。我们以直接的方式构造 $A(E)$ 在 $uw$ 上的一次运行。

**情形 5** $(E = F^{k,*})$。设 $K = \mathcal{L}(F)$。回忆一下，$\mathcal{L}\left(F^{k,*}\right) = \bigcup_{n \geq 0} K^{k,n}$ 其中 $K^{k,0}$ 是长度为 $2k$ 的回文串集合。
针对 $K^{k,0}$ 的自动机。注意，长度为 $2k$ 的回文串语言可以由表达式 $[w_1] + [w_2] + \cdots + [w_m]$ 定义：枚举所有长度为 $2k$ 的词的轨道的代表元 $w_1, \ldots, w_m$ 即可。因此，根据归纳假设，它有一个对应的自动机 $\mathrm{A}_0$。针对 $\bigcup_{n \geq 1} K^{k,n}$ 的自动机。这是针对 $E \cdot_k F$ 的构造的一种迭代变体。我们构造一个 $\mathrm{A}_{\geq 1} \in \mathrm{NRA}^{\mathrm{G}}_{\varepsilon}[\Sigma, \mathbb{A}]$，使其开始时在真实输入上模拟 $A(F)$，并且可以非确定性地决定执行与情形 $E \cdot_k F$ 中相同的“猜测+重放”组件；不过现在，它从 $A(E)$ 返回到 $A(E)$ 中的模拟。具体而言，取上述针对 $AF \cdot_k F$ 的构造，并合并状态 $(q, \mathsf{P1})$ 和 $(q, \mathsf{P4})$。由此得到的自动机识别 $\bigcup_{n \geq 1} K^{k,n}$。正确性论证与上文相同，通过计算运行中已完成的组件的数量即可：每个已完成的“猜测+重放”组件对应于 $\cdot_k$ 的一次应用，而最终接受片段对应于最后一个因子。

最后，定义 $A(E)$ 为 $\mathcal{A}_0$ 和 $\mathcal{A}^{\geq 1}$ 的并（情形 $+$）。于是 $\mathcal{L}(\mathcal{A}_E) = \mathcal{L}\left(F^{k,*}\right)$。
至此，归纳构造完成，因此对于每个 $E$，$\mathcal{L}(A(E)) = \mathcal{L}(E)$。应用引理 14 可得一个不含 $\varepsilon$-规则、识别相同语言的 $\mathrm{NRA}^{\mathrm{G}}$。◀

## A.2 从自动机到表达式

本节证明定理 5 的逆向结论。

▶ **引理 15**。设 $A \in \mathrm{NRA}^{\mathrm{G}}[\Sigma, \mathbb{A}]$。存在一个数据正则表达式 $E(A) \in$数据正则表达式 $[\Sigma, \mathbb{A}]$，使得 $L(A) = L(E(A))$。

**证明**。 固定一个 $\mathcal{A} = (Q,\ \Sigma,\ \mathbb{A},\ \mathcal{R}\ Q_{\mathrm{ini}},\ Q_{\mathrm{fin}},\ \delta) \in \mathrm{NRA}_k^{\mathrm{G}}[\Sigma, \mathbb{A}]$。固定一个枚举 $\mathcal{R} = \{r_1,\ \ldots,\ r_k\}$。另外，固定一个特殊的 **填充字母** $\# \in \Sigma$ (假设 $\Sigma \neq \emptyset$)。

**转移标签非确定有限自动机**。令 $\Gamma := \delta$ 为 $\mathcal{A}$ 的有限转移规则集。定义 $\mathcal{B} = (Q,\ \Gamma,\ Q_{\mathrm{ini}},\ Q_{\mathrm{fin}},\ \delta') \in$ 非确定有限自动机$[\Gamma]$，其中 $\delta' \subseteq Q \times \Gamma \times Q$ 包含转移 $p \xrightarrow{t} q$ 当且仅当 $t \in \delta$ 是规则 $p \overset{\sigma,\ \Phi}{\Longrightarrow} q$ 对于某个 $\sigma \in \Sigma$ 且 $\Phi \in$ 满足构造条件$_{\mathbb{A}}(\mathcal{R})$。因此，$\mathcal{B}$ 读取字母表 $\Gamma$ 上的词，并且恰好接受控制一致的规则序列：

$$\mathcal{L}(\mathcal{B}) = \{\pi \in \delta^* \mid \pi \text{ is a path from some } q_{\mathrm{ini}} \in Q_{\mathrm{ini}} \text{ to some } q_{\mathrm{fin}} \in Q_{\mathrm{fin}}\}.$$

根据克莱尼定理（定理 1），存在一个（经典）正则表达式 $E_{\mathcal{B}} \in \mathrm{RE}[\Gamma]$，使得 $\mathcal{L}(E_{\mathcal{B}}) = \mathcal{L}(\mathcal{B})$。

**区域的并集**。注意，$\mathbb{A}^m$ 中词的无量词区域构成有限集。令 $\mathrm{repr}_m$ 为这些区域的代表元集合。设 $\psi(x_1,\ \ldots,\ x_m)$ 是关于 $\mathbb{A}$ 的无量词一阶公式，其自由变量属于 $x_1,\ \ldots,\ x_m$。对于 $w \in \Sigma^m$，定义 $\mathrm{any}(m,\ \psi) \in$ 数据正则表达式$[\Sigma, \mathbb{A}]$ 为满足 $\psi$ 的无量词区域之并

$$\mathrm{any}(w, \psi) := \sum_{\substack{\varpi \in \mathrm{repr}_m \\ \mathbb{A} \models \psi(\varpi)}} \Big[\otimes(w, \varpi)\Big].$$

最后，定义守卫 $G := \text{任意}(\#^k,\ \top)$。

**编码转移规则**。设 $t \in \delta$ 为规则 $p \overset{\sigma,\ \Phi}{\Longrightarrow} q$。我们通过代入以下内容，将 $\Phi$ 转换为公式 $\hat{}\Phi(x_0,\ \ldots\ldots,\ x_{2k})$：$\overset{\triangleleft}{r}_i \mapsto x_{k-i}$，$\overset{\triangledown}{y} \mapsto x_k$，$\overset{\triangleright}{r}_i \mapsto x_{k+i}$，对所有 $i \in [k]_{\mathbb{N}}$均如此，并将每个原子公式 $\mathrm{nil}(r)$替换为 $\top$。直观地说，前 $k$ 个变量按逆序编码执行 $t$之前的寄存器赋值，最后 $k$ 个变量则编码执行 $t$之后的赋值。初始活动区间向前延伸至编码的最开头，因此无需使用 nil。定义与 $t$ 关联的数据表达式如下

$$\mathrm{Block}(t) := \mathrm{any}\left(\#^k \sigma \#^k, \hat{\Phi}\right) \in \mathrm{DRE}[\Sigma, \mathbb{A}].$$

直观地说，$\mathrm{Block}(t)$由所有 $(2k+1)$ 字母块组成，这些块的中间符号具有有限字母 $\sigma$，填充字母均为 #，且其数据值在被解释为某个赋值 $\mu$、当前数据值以及某个 $\mu'$的编码时，满足约束 $\Phi$。

**将 $\varepsilon$ 解释为 $(\cdot_k)$的幺元**。我们将把 $E_{\mathcal{B}}$ 转换为一个 $k$-收缩数据表达式；为此，我们需要一个充当 $\varepsilon$ 角色的表达式。定义公式

$$\mathrm{pal}_k(x_1, \ldots, x_{2k}) := \bigwedge_{i \in [k]_{\mathbb{N}}} x_i = x_{2k-i+1}$$

并令 $\mathrm{Pal}(k) := \mathrm{any}(\#^{2k}, \mathrm{pal}_k)$。容易看出，$\mathcal{L}(\mathrm{Pal}(k))$ 是以下集合

$$\left\{ \otimes(\#^k, \varpi) \otimes (\#^k, \varpi)^R \;\middle|\; \varpi \in \mathbb{A}^k \right\}.$$

**将 $E_{\mathcal{B}}$ 转换为数据正则表达式**。定义一个转换 $\hat{}\mathrm{E}(\cdot)$，将 $\Gamma$ 上的经典正则表达式映射为 $\mathrm{DRE}[\Sigma, \mathbb{A}]$ 中的数据表达式，定义如下：

$$\hat{\mathrm{E}}(\emptyset) := \emptyset \qquad \hat{\mathrm{E}}(E + F) := \hat{\mathrm{E}}(E) + \hat{\mathrm{E}}(F)$$

$$\hat{\mathrm{E}}(\epsilon) := \mathrm{Pal}(k) \qquad \hat{\mathrm{E}}(E \cdot F) := \hat{\mathrm{E}}(E) \cdot_k \hat{\mathrm{E}}(F)$$

$$\hat{\mathrm{E}}(t) := \mathrm{Block}(t) \quad (t \in \Gamma) \qquad \hat{\mathrm{E}}(E^*) := \hat{\mathrm{E}}(E)^{k,*}.$$

最后，令 $E(\mathcal{A}) := G \cdot_k \hat{}\mathrm{E}(E_{\mathcal{B}}) \cdot_k G$。

**正确性**。 我们证明 $\mathcal{L}(E(\mathcal{A})) = \mathcal{L}(\mathcal{A})$ 两者互相包含。
**方向**$\mathcal{L}(\mathcal{A}) \subseteq \mathcal{L}(E(\mathcal{A}))$。令 $w = (\sigma_1, \alpha_1)\cdots(\sigma_n, \alpha_n) \in (\Sigma \times \mathbb{A})^*$ 由$\mathcal{A}$接受。固定一条接受运行

$$(Q_{\mathrm{ini}}, \mu_0) \xrightarrow{\sigma_1,\alpha_1} (q_1, \mu_1) \xrightarrow{\sigma_2,\alpha_2} \cdots \xrightarrow{\sigma_n,\alpha_n} (q_n, \mu_n),$$

其中 $\mu_0(r) =$ 对所有 $r \in \mathcal{R}$ 和 $q_n \in Q_{\mathrm{fin}}$均为 nil。固定相应的转移规则序列 $t_1 t_2 \cdots t_n$。根据构造，它属于 $\mathcal{L}(\mathcal{B}) = \mathcal{L}(E_{\mathcal{B}})$，并且对每个 $i \in [n]_{\mathbb{N}}$，我们有

$$t_i = (q_{i-1} \xrightarrow{\sigma_i, \Phi_i} q_i) \quad \text{with} \quad \nu(\mu_{i-1}, \alpha_i, \mu_i) \models \Phi_i.$$

对每个 $i$，考虑 $(2k+1)$-块

$$B_i \coloneqq \otimes\big(\#^k\, \sigma_i\, \#^k,\ \ \mu_{i-1}(r_k)\cdots\mu_{i-1}(r_1)\ \alpha_i\ \mu_i(r_1)\cdots\mu_i(r_k)\big)\,.$$

根据构造及 $\hat{}\ \Phi_i$ 的定义，我们有 $B_i \in \mathcal{L}(\mathrm{Block}(t_i))$。此外，$k$-收缩连接恰好对应于后赋值和前赋值：$B_i$ 的最后 $k$ 个符号是 $\otimes\big((\#^k, \mu_i(r_1)\cdots\mu_i(r_k))\big)$，反转后即等于 $B_{i+1}$的前 $k$ 个符号。因此，词 $W \coloneqq B_1 \cdot_k B_2 \cdot_k \ldots \cdot_k B_n$ 定义良好，并满足 $W \in \mathcal{L}(\mathrm{Block}(t_1) \cdot_k \cdots {}_k\,\mathrm{Block}(t_n)) \subseteq \hat{}\mathrm{E}(E_{\mathcal{B}})$。直接计算 $k$-收缩可得

$$W = \otimes\big(\#^k, \varpi'\big) \cdot w \cdot \otimes\big(\#^k, \varpi''\big)$$

对于某些 $\varpi'$、$\varpi'' \in \mathbb{A}^k$。根据守卫 $G$ 的定义，$w \in \mathrm{L}\big(G \cdot_k \hat{}\mathrm{E}(E_{\mathcal{B}}) \cdot_k G\big) = \mathrm{L}(E(\mathrm{A}))$。方向为 $\mathrm{L}(E(\mathrm{A})) \subseteq \mathrm{L}(\mathrm{A})$。令 $w \in \mathrm{L}(E(\mathrm{A}))$。根据 $\cdot_k$ 的语义，存在词 $u$、$v \in \mathrm{L}(G)$ 和 $W \in \hat{}\mathrm{E}(E_{\mathcal{B}})$，使得 $w = u \cdot_k W \cdot_k v$。由于 $\mathrm{L}(G) = (\{\#\} \times \mathbb{A})^k$，两个外部 $k$-收缩的作用是移除 $W$ 的长度为 $k$ 的前缀和长度为 $k$ 的后缀。特别地，$W = u^R w v^R$。

现在，由于 $W \in \hat{}\mathrm{E}(E_{\mathcal{B}})$，展开 $\hat{}\mathrm{E}(\cdot)$ 的定义可得

$$W = u_0^R a_1 u_1 \ \cdot_k\ u_1^R a_2 u_2 \ \cdot_k\ u_2^R \cdots u_{n-1} \ \cdot_k\ u_{n-1}^R a_n u_n$$

对于某些 $u_i \in$ (<style id="2">{#<style id="4">} $\times\, \mathbb{A})^k$，使得 $B_i \coloneqq u_{i-1}^R a_i u_i \in$ <style id="10">L(Block($t_i$)) 对于 $i \in [n]_{\mathbb{N}}$ 成立；以及某个序列 $t_1$ <style id="15">$\cdots\ t_n \in$ <style id="18">L(<style id="20">B)，其中包含从某个初始 $q_{\mathrm{ini}} \in Q_{\mathrm{ini}}$ 开始并以某个接受 $q_{\mathrm{fin}} \in Q_{\mathrm{fin}}$ 结束的转移规则。每个块 $B_i$ 通过其数据轨道确定一个寄存器赋值三元组和一个当前数据值（$\mu_{i-1}$<style id="26">, $\alpha_i$<style id="29">, $\mu_i$），使得 $\nu(\mu_{i-1}$<style id="34">, $\alpha_i$ <style id="37">, $\mu_i) \models \Phi_i$ 满足 $t_i$ 的约束（可能需要先将某些数据值替换为空值）。

因此，我们得到了自动机 $\mathcal{A}$ 在词 $w = a_1 a_2 \cdots a_n$ 上一个定义良好的接受运行：

$$(q_0, \mu_0) \xrightarrow{\sigma_1,\alpha_1} (q_1, \mu_1) \cdots \xrightarrow{\sigma_n,\alpha_n} (q_n, \mu_n)\,,$$

其中 $\mu_0$ 是初始赋值，$q_0 = q_{\mathrm{ini}}$ 和 $q_n = q_{\mathrm{fin}}$。因此 $w \in \mathcal{L}(\mathcal{A})$。 ◀

## B 逻辑-自动机等价

在本节余下部分，固定一个有限字母表 $\Sigma$ 和原子 $\mathbb{A} = (U, R_1, \ldots, R_\ell)$。

### B.1 从公式到自动机

本节证明定理 9 的一个方向，具体由下面的引理给出。

▶ **引理 16**。 对于每个单子二阶逻辑句子 $\varphi_0 \in MSO_S[<, \Sigma, \mathbb{A}]$，都存在一个 $A \in \mathrm{NRA}^{\mathrm{G}}[\Sigma, \mathbb{A}]$，使得 $L(\varphi_0) = L(A)$。

在本节余下部分中，固定一个单子二阶逻辑句子 $\varphi_0 \in \mathrm{MSO}_{\mathbb{S}}[<, \Sigma, \mathbb{A}]$。不失一般性， $\varphi_0$ 中由量词绑定的所有变量两两不同。令 $\mathcal{U}$ 为这些变量构成的集合，并记作 $\mathcal{U} = \mathcal{U}_{\uparrow} \cup \mathcal{U}_{\downarrow}$，其中 $\mathcal{U}_{\uparrow}$ 由顶层变量组成。设 $N := |\mathcal{U}_{\uparrow}|$。

▶ 注释 17. 根据条件 6， $\varphi_0$ 中的每个形如 $R(x_1, \ldots, x_r)$ 的原子公式至多包含一个来自 $\mathcal{U}_{\downarrow}$ 的变量。

**多轨字母表**。在本节中，我们使用单子二阶逻辑 $\mathrm{MSO}_{\mathbb{S}}[<, \Sigma', \mathbb{A}']$ 的公式，其中 re

$$\Sigma' := \Sigma \times \{\bullet, \circ\}^N, \qquad \mathbb{A}' := \mathbb{A}^{(N+1)}.$$

这里， $\mathbb{A}^{(N+1)}$ 表示一个论域为 $U^{N+1}$ 且具有扩展签名的结构：对于 $\mathbb{A}$ 的每个元数为 $r$ 的关系符号 $R$，以及每个元组 $(i_1, \ldots, i_r) \in [N+1]^r_{\mathbb{N}}$， $\mathbb{A}'$ 的签名中都包含一个关系符号 $R^{(i_1,\ldots,i_r)}$，用于测试相应坐标上的 $R$。

给定一个词 $\otimes(w, \varpi) \in (\Sigma \times \mathbb{A})^*$ 以及长度相同的词族 $\boldsymbol{u} : \mathcal{U}_{\uparrow} \to \{\bullet, \circ\}^*$ 和 $\boldsymbol{v} : \mathcal{U}_{\uparrow} \to \mathbb{A}^*$，我们用 $\mathbb{W}(w, \boldsymbol{u}, \varpi, \boldsymbol{v})$ 表示数据词结构 $\mathbb{W}(\otimes(w, \boldsymbol{u}) \otimes(\varpi, \boldsymbol{v}))$。将这类结构视为多轨词会很方便。

我们将 $\Sigma'$ 和 $\mathbb{A}'$ 的分量称作：

- 有限轨道： $w$ 和 $u_x$ 对应 $x \in \mathcal{U}_{\uparrow}$；
- 无限轨道： $\varpi$ 和 $v_x$ 对应 $x \in \mathcal{U}_{\uparrow}$。

假设轨道名称与元组索引之间存在固定的双射，我们使用以下语法糖：

$$R(t_1[x_1], \ldots, t_r[x_r]) \equiv R^{(i_1,\ldots,i_r)}(x_1, \ldots, x_r), \qquad t[x] = a \equiv a_1(x) \vee \cdots \vee a_n(x),$$

其中，每个 $t_j$ 都是与索引 $i_j$ 对应的轨道， $t$ 是有限轨道， $a \in \Sigma \cup \{\bullet, \circ\}$，并且 $\{a_1, \ldots, a_n\}$ 是 $\Sigma'$ 中坐标 $t$ 等于 $a$ 的字母集合。

这种记法既可以看作单子二阶逻辑 $\mathrm{MSO}_{\mathbb{S}}[<, \Sigma', \mathbb{A}']$ 的语法糖，也可以等价地看作一种带有轨道访问器 项 $t[x]$ 的温和多排序扩展。我们可以在这两种观点之间自由切换。

**预备定义**。对于每个 $x \in \mathcal{U}_{\uparrow}$，定义公式

$$\eta_x := \underbrace{\forall s, t\ (\mathrm{Succ}(s,t) \wedge u_x[t] = \circ) \to v_x[s] = v_x[t]}_{\text{For all } s, s+1, \text{ if } u_x[s+1] = \circ \text{ then } v_x \text{ is constant.}} \wedge \underbrace{\forall s\ \mathrm{First}(s) \to u_x[s] = \bullet}_{\text{The track } u_x \text{ starts with } \bullet.},$$

$$\theta_{x,S} := \underbrace{v_x[x] = \varpi[x]}_{\substack{\text{The value } \varpi[x] \text{ is also} \\ \text{present on the } v_x \text{ track.}}} \wedge \underbrace{\forall s\ S(s) \to (u_x[s] = \bullet \leftrightarrow \mathrm{FirstIn}(s, S))}_{\substack{\text{On the scope } S, \text{ the track } u_x \text{ forms} \\ \text{a word in } \bullet\circ^*.}}.$$

其中 $\mathrm{Succ}(s,t) \equiv \neg\exists p.\ s < p \wedge p < t$， $\mathrm{First}(s) \equiv \neg\exists p.\ p < s$，以及 $\mathrm{FirstIn}(s, S) \equiv \neg\exists p.\ S(p) \wedge p < s$。如果公式 $\varphi$ 是局部测试的，则称其为局部测试公式；若其中出现的每个数据原子公式都具有形式 $R(t_1[x], \ldots, t_r[x])$，其中某个变量为 $x$，则该公式 $\varphi$ 永远不会比较词中不同位置的数据值。

▶ **引理 18** （子作用域可由单子二阶逻辑定义）。存在一个单子二阶逻辑 $[<, \Sigma]$ 公式 $\mathrm{Subs}(S', X, S)$，使得对于每个数据词结构 $fS$ 和赋值 $\nu$，满足关系 $fS, \nu \models \mathrm{Subs}(S', X, S)$ 成立，当且仅当 $\nu(S)$ 是非空值域，并且 $\nu(S') \in \mathrm{subs}(\nu(S)\ \nu(X))$。

**公式翻译**。为证明引理 19，我们定义一个翻译 $F(\varphi)$，它将 $\varphi_0$ 的每个子公式 $\varphi$ 映射到单子二阶逻辑 $\mathrm{MSO}_{\mathbb{S}}[<, \Sigma', \mathbb{A}']$ 的一个公式，该公式

- 是局部测试，并且

不包含作用域模态。 ∎

我们通过对 $\varphi$ 进行结构递归来定义 $F$，并使用一个辅助函数 $G(\varphi, S)$，该函数以一个额外的作用域变量作为参数。考虑一个数据原子公式 $\varphi \equiv R(x_1, \dots, x_n)$。根据注记 17，$|\{x_1, \dots, x_n\} \cap \mathcal{U}_\downarrow| \le 1$。我们令

$$G(\varphi, S) \coloneqq \begin{cases} R(t_{1,x}, \dots, t_{n,x}) & \text{if } \{x_1, \dots, x_n\} \cap \mathcal{U}_\downarrow = \{x\}, \\ \exists z.\ z \in S \wedge R(t_{1,z}, \dots, t_{n,z}) & \text{if } \{x_1, \dots, x_n\} \cap \mathcal{U}_\downarrow = \emptyset, \end{cases} \tag{1}$$

其中 $z$ 是新鲜变量，$t_{i,\,y}$ 是轨道访问器项；当变量符号 $x_i$ 和 $y$ 相同时为 $\varpi[y]$，否则为 $v_{x_i}[y]$。对于其余构造，定义：

$$G(a(x), S) \coloneqq w[x] = a, \tag{2}$$
$$G(X(y), S) \coloneqq X(y), \tag{3}$$
$$G(x < y, S) \coloneqq x < y, \tag{4}$$
$$G(\gamma \wedge \gamma', S) \coloneqq G(\gamma, S) \wedge G(\gamma', S), \tag{5}$$
$$G(\gamma \vee \gamma', S) \coloneqq G(\gamma, S) \vee G(\gamma', S), \tag{6}$$
$$G(\neg\gamma, S) \coloneqq \neg G(\gamma, S), \tag{7}$$
$$G(\exists x\ \gamma, S) \coloneqq \begin{cases} \exists x.\ x \in S \wedge G(\gamma, S) \wedge \theta_{x,S} & \text{if } x \in \mathcal{U}_\uparrow \\ \exists x.\ x \in S \wedge G(\gamma, S) & \text{otherwise,} \end{cases} \tag{8}$$
$$G(\exists X\ \gamma, S) \coloneqq \exists X.\ X \subseteq S \wedge G(\gamma, S), \tag{9}$$
$$G(\mathcal{2}_X\ \gamma, S) \coloneqq \forall S'.\ \mathrm{Subs}(S', X, S) \to G(\gamma, S'), \quad \text{where } S' \text{ is a fresh SO variable.} \tag{10}$$

最后，定义

$$F(\varphi_0) \coloneqq \begin{cases} \neg\,\mathrm{Empty} \wedge (\exists S.\ G(\varphi_0, S) \wedge \forall y\ S(y)) \vee \mathrm{Empty} & \text{if } \varepsilon \in \mathcal{L}(\varphi_0) \\ \neg\,\mathrm{Empty} \wedge (\exists S.\ G(\varphi_0, S) \wedge \forall y\ S(y)) & \text{otherwise,} \end{cases}$$

其中 $S$ 和 $y$ 是新鲜变量符号，且 $\mathsf{Empty} \equiv \neg\exists x.\ x = x$。

引理 16 的证明可分解为以下引理。

▶ 引理 19. 词 $\otimes(w, \varpi) \in (\Sigma \times \mathbb{A})^*$ 属于 $L(\varphi_0)$，当且仅当存在族 $\boldsymbol{u}: U_\uparrow \to \{\bullet, \circ\}^*$ 和 $\boldsymbol{v}: U_\uparrow \to \mathbb{A}^*$，使得 $\otimes(w, \boldsymbol{u}, \varpi, \boldsymbol{v}) \in L(F(\varphi_0)) \cap L\big(\bigwedge_{x \in U_\uparrow} \eta_x\big)$。

▶ **引理 20.** 对于每个不使用作用域模态的局部测试 $\varphi \in$ 单子二阶逻辑$_S[<, \Sigma', \mathbb{A}']$，都存在一个 $A \in \mathrm{NRA}_0^{\mathrm{G}}[\Sigma', \mathbb{A}']$，使得 $L(\varphi) = L(A)$。

▶ **引理 21.** 对于每个 $A \in \mathrm{NRA}_0^{\mathrm{G}}[\Sigma', \mathbb{A}']$，都存在一个$\mathrm{NRA}^{\mathrm{G}}$，它识别语言

$$\left\{ \otimes(w, \varpi) \;\middle|\; \exists \boldsymbol{u}, \boldsymbol{v}.\ \otimes(w, \boldsymbol{u}, \varpi, \boldsymbol{v}) \in \mathcal{L}(\mathcal{A}) \cap \mathcal{L}\big(\textstyle\bigwedge_{x \in \mathcal{U}_\uparrow} \eta_x\big) \right\}.$$

引理 21 并不令人意外，因为 $\eta_x$ 可以通过一种直接的方式由 NRA 识别，而该构造需要实现交集（乘积构造）和存在量化轨道投影。为避免处理 $\mathbb{A}$ 和 $\mathbb{A}'$所带来的技术困难，我们决定给出一个整体构造：在 $\mathbb{A}'$ 上的寄存器自动机在形式上需要在寄存器中保存原子值的元组，这会很繁琐。假设引理 19–21 成立，通过一个简短的组合论证即可推出引理 16。

引理 16 的证明（假设引理 19–21 成立）。固定任意一个词 $\otimes(w, \varpi) \in (\Sigma \times \mathbb{A})^*$。根据引理 19，$\otimes(w, \varpi) \in \mathrm{L}(\varphi_0)$ 当且仅当存在族 $\boldsymbol{u} \in \mathrm{U}_\uparrow \to \{\bullet, \circ\}^*$ 和 $\boldsymbol{v} \in \mathrm{U}_\uparrow \to \mathbb{A}^*$，使得 $\otimes(w, \boldsymbol{u}, \varpi, \boldsymbol{v}) \in \mathrm{L}(F(\varphi_0)) \cap \mathrm{L}\big(\bigwedge_{x \in \mathrm{U}_\uparrow} \eta_x\big)$。

根据引理 20，我们得到 $\mathcal{A} \in \mathrm{NRA}_0^{\mathrm{G}}[\Sigma', \mathbb{A}']$，它识别 $\mathcal{L}(F(\varphi_0))$。最后，引理 21 得出 $\mathcal{B} \in \mathrm{NRA}^{\mathrm{G}}[\Sigma, \mathbb{A}]$，使得 $\otimes(w, \varpi) \in \mathcal{L}(\mathcal{B})$ 当且仅当 $\otimes(w, \varpi) \in \mathcal{L}(\varphi_0)$。因此$\mathcal{L}(\mathcal{D}) = \mathcal{L}(\varphi_0)$，证毕。◀

**证明路线图。** 我们先证明较为直接的引理 20–21，然后转向主要技术步骤——引理 19。

引理 20 的证明。固定一个不使用模态 的局部测试单子二阶逻辑句子 $\varphi \in \mathrm{MSO}_{\mathbb{S}}[<, \Sigma', \mathbb{A}']$。我们的目标是构造一个$\mathfrak{A} \in \mathrm{NRA}_0^{\mathrm{G}}[\Sigma', \mathbb{A}']$，使得 $\mathrm{L}(\varphi) = \mathrm{L}(\mathrm{A})$。等价地，对于每个词 $\otimes(w, \varpi)$，我们希望 $\otimes(w, \varpi) \in \mathrm{L}(\mathrm{A})$ 当且仅当 $\mathbb{W}(w, \varpi) \models \varphi$。注意，$\varphi$ 的所有子公式都在初始作用域 $\mathrm{pos}(w)$ 上解释。

**翻译 $H$。** 令 $\mathcal{R}$ 为 $\mathbb{A}'$ 的签名中所有关系名称的集合。我们用数据值的无量词类型扩展 $\Sigma'$：$\Sigma'' = \Sigma' \times \mathcal{P}(\mathcal{R})$。定义一个翻译 $H(\psi)$，将 $\varphi$ 的子公式映射为单子二阶逻辑公式 $[<, \Sigma'']$。请注意，由于局部测试这一假设，所有数据原子公式都测试单个位置变量 $x$。我们定义

$$H(R^{(i_1,\dots,i_r)}(x,\dots,x)) \equiv \bigvee_{\substack{\sigma=(a,\mathcal{R}')\in\Sigma'' \\ R^{(i_1,\dots,i_r)}\in\mathcal{R}'}} \sigma(x) \qquad H(a(x)) \equiv \bigvee_{\sigma=(a,\mathcal{R}')\in\Sigma''} \sigma(x)\,.$$

并对其余所有类型的公式平凡地定义 $H$： $H(\psi \wedge \gamma) \equiv H(\psi) \wedge H(\gamma)$,$H(\neg\psi) \equiv \neg H(\psi)$, $H(\exists x\ \psi) = \exists x\ H(\psi), \dots$.

给定一个长度为 $L$ 的数据词结构 $\mathbb{W}(w, \boldsymbol{u}, \varpi, \boldsymbol{v})$，我们定义一个词 $\rho(\varpi, \boldsymbol{v}) \in \mathcal{P}(\mathcal{R})^L$，使得对于每个位置 $i \in [L]_{\mathbb{N}}$，字母 $\rho[i]$ 都是 $\{R \in \mathcal{R} \mid \alpha_i^{\mathrm{ar}(R)} \in R\}$，其中 $\alpha_i$ 是元组 $\alpha_i = (\varpi[i], \boldsymbol{v}[i]) \in \mathbb{A}'$。

▷ 断言 22。固定 $\varphi$ 的一个子公式 $\psi$，以及一个解释 $(\mathbb{W}(w, \boldsymbol{u}, \varpi, \boldsymbol{v})\ \nu)$。那么$\mathbb{W}(w, \boldsymbol{u}, \varpi, \boldsymbol{v})\ \nu \models \varphi$ 当且仅当 $\mathbb{W}(w, \boldsymbol{u}, \rho(\varpi, \boldsymbol{v}))) \models G(\varphi)$。

我们通过结构归纳来证明该断言。固定一个子公式 $\psi$ of $\varphi$，以及一个解释 $(\mathbb{W}(w, \boldsymbol{u}, \varpi, \boldsymbol{v})\ \nu)$。
情形1 $(\psi \equiv R^{(i_1,i_2,\dots,i_r)}(x, x, \dots, x))$。令 $i = \nu(x)$。以下陈述等价：$\mathbb{W}(w, \boldsymbol{u}, \varpi, \boldsymbol{v})\ \nu \models \varphi$ 当且仅当 $(\varpi[i], \boldsymbol{v}[i])^{\mathrm{ar}(R)} \in R$ 当且仅当 $R \in \rho(\varpi, \boldsymbol{v})[i]$ 当且仅当 $\mathbb{W}(w, \boldsymbol{u}, \rho(\varpi, \boldsymbol{v}))\ \nu \models G(\psi)$。情形2 $(\psi \equiv a(x))$。令 $i = \nu(x)$。以下陈述等价： $\mathbb{W}(w, \boldsymbol{u}, \varpi, \boldsymbol{v})\ \nu \models \varphi$ 当且仅当 $w[i] = a$ 当且仅当 $\mathbb{W}(w, \boldsymbol{u}, \rho)\ \nu \models G(\psi)$。
**情形 3-10** (其他情况)。 其余情形均为平凡情形，故略去。
令 $\mathcal{A} = (Q, \Sigma'', Q_{\mathrm{ini}}, Q_{\mathrm{fin}}, \delta) \in \mathrm{DFA}[\Sigma'']$ 为通过定理 2 得到的、关于 $G(\varphi)$ 的自动机。

▷ 断言 23. 令 $\otimes(w, \boldsymbol{u}, \varpi, \boldsymbol{v})$ 为 $(\Sigma' \times \mathbb{A}')^*$ 上的任意词。那么， $\otimes(w, \boldsymbol{u}, \varpi, \boldsymbol{v}) \in \mathcal{L}(\varphi)$ 当且仅当 $\otimes(w, \boldsymbol{u}, \rho(\varpi, \boldsymbol{v})) \in \mathcal{L}(\mathcal{A})$。

定义 $\mathcal{B} = (Q, \Sigma', \mathbb{A}', Q_{\mathrm{ini}}, F, \delta') \in \mathrm{NRA}_0^{\mathrm{G}}[\Sigma', \mathbb{A}']$，使得对每个 $\mathcal{R}' \subseteq \mathcal{R}$， $\sigma \in \Sigma'$ 并且$p \in Q$ 关系 $\delta'$ 包含一条规则 $p \xrightarrow{\sigma,\ \Phi(\mathcal{R}')} q$，其中 $\delta(p, (\sigma, \mathcal{R}'))$ 且

$$\Phi(\mathcal{R}') = \textstyle\bigwedge_{R\in\mathcal{R}'} R(\overset{\triangledown}{y},\dots,\overset{\triangledown}{y}) \wedge \bigwedge_{R\in\mathcal{R}\setminus\mathcal{R}'} \neg R(\overset{\triangledown}{y},\dots,\overset{\triangledown}{y})$$

且 $\delta'$ 没有其他规则。不难看出，对于任意 $(\sigma, \alpha) \in \Sigma' \times \mathbb{A}'$ 和 $p, q \in Q$，自动机 A 存在转移 $p \xrightarrow{\sigma,\rho(\alpha))} q$，当且仅当 B 存在转移 $p \xrightarrow{\sigma,\ \alpha} q$。因此，通过直接归纳法可得，$\otimes(w, \boldsymbol{u}, \rho(\varpi, \boldsymbol{v})) \in \mathrm{L}(\mathrm{A})$ 当且仅当 $\otimes(w, \boldsymbol{u}, \varpi, \boldsymbol{v}) \in \mathrm{L}(\mathrm{B})$。于是，B 就是所述的、识别 $\mathrm{L}(\varphi)$ 的 $\mathrm{NRA}_0^{\mathrm{G}}[\Sigma', \mathbb{A}']$。◀

**引理 21 的证明**。令 $\mathcal{A} = (Q, \Sigma', \mathbb{A}', \emptyset, Q_{\mathrm{ini}}, Q_{\mathrm{fin}}, \delta)$。由于 $\mathcal{A}$ 有 0 个寄存器，每条规则 $p \xrightarrow{\sigma', \Phi} q$ 只检查当前的 $\mathbb{A}'$-字母。我们构造一个运行在 $\Sigma \times \mathbb{A}$ 上的 $\mathcal{B}$，使其能够：(i) 即时猜测被投影掉的轨道 $\boldsymbol{u}, \boldsymbol{v}$；(ii) 根据猜测的字母检查 $\mathcal{A}$；(iii) 通过在寄存器中记住 $v_x[i-1]$ 来强制执行 $\bigwedge_x \eta_x$。

**寄存器及其 $\eta$-守卫。在读取位置**$i$ 之后，我们有 $\mu_i(r_x) = v_x[i]$。对于标记元组 $\gamma = (\gamma_x)_{x \in \mathcal{U}_\uparrow}$，其中 $\gamma_i \in \{\bullet, \circ\}$，定义

$$\mathrm{Eta}(\gamma) \coloneqq \bigwedge_{\substack{x \in \mathcal{U}_\uparrow \\ \gamma_x = \circ}} \left( \overset{\triangleleft}{r}_x = \overset{\triangleright}{r}_x \right).$$

这正是 $\eta_x$ 的单步形式：如果 $u_x[i] = \circ$，那么 $v_x[i] = v_x[i-1]$。

**编译 $\mathbb{A}'$-测试。固定** $(a, \gamma, \Phi, q)$ 规则 $p$ of $\mathcal{A}$。由于 $\mathcal{A}$ 是 0-寄存器的，$\Phi$ 等价于关于 $\mathbb{A}$ 的某个约束，其自由变量为 $\bar{y}$（输入数据值）以及来自 $\mathbb{A}'$ 的 $(N+1)$-元组的分量 $v_x(\overset{\triangledown}{y})$。令 $\tau(\Phi) \in \mathsf{constr}_{\mathbb{A}}(\mathcal{R})$ 表示相应的寄存器约束，其中将 $v_x(\overset{\triangledown}{y})$ 读取为 $\overset{\triangleright}{r}_x$。因此，$\tau(\Phi)$ 检查 $\mathcal{A}$ 的 $\mathbb{A}'$-条件是否满足猜测到后置寄存器值中的元组。

B 的构造。我们为状态扩展一个一位的“首位”标记，以避免在 $i = 1$ 处强制要求 $\eta_x$（因为 $\eta_x$ 是对 $i > 1$ 进行量化的）。令 $Q' \coloneqq Q \times \{0, 1\}$，初始状态为 $Q'_{\mathrm{ini}} \coloneqq Q_{\mathrm{ini}} \times \{0\}$，接受状态为 $Q'_{\mathrm{fin}} \coloneqq Q_{\mathrm{fin}} \times \{1\}$。定义 $\mathrm{B} = (Q', \Sigma, \mathbb{A}, \mathrm{R}, Q'_{\mathrm{ini}}, Q'_{\mathrm{fin}}, \delta_B)$，其中对于 A 的每条规则 $p \xrightarrow{(a, \gamma, \Phi)} q$，我们都加入：

$$(p, 0) \xrightarrow{a,\ \tau(\Phi)} (q, 1) \qquad \text{and} \qquad (p, 1) \xrightarrow{a,\ \tau(\Phi) \wedge \mathrm{Eta}(\gamma)} (q, 1).$$

仅读取 $a \in \Sigma$ 就能实现对 $\boldsymbol{u}$ 的投影：自动机 $\mathcal{B}$ 通过选择要采用的规则，非确定性地选择标记元组 $\gamma$。通过在每一步猜测寄存器值 $\overset{\triangleright}{r}_x$，即可完成对数据值轨道 $v_x$ 的投影。

**正确性**。我们需要证明，$\otimes(w, \varpi) \in \mathcal{L}(\mathcal{B})$ 当且仅当存在 $\boldsymbol{u}, \boldsymbol{v}$，使得 $\otimes(w, \boldsymbol{u}, \varpi, \boldsymbol{v}) \in \mathcal{L}(\mathcal{A}) \cap \mathcal{L}(\bigwedge_x \eta_x)$。

**方向“⇒”**。固定 $\mathcal{B}$ 在 $\otimes(w, \varpi)$ 上的一个接受运行：

$$((q_0, 0), \mu_0) \xrightarrow{w[1], \varpi[1]} ((q_1, 1), \mu_1) \to \cdots \xrightarrow{w[n], \varpi[n]} ((q_n, 1), \mu_n),$$

其中，对于所有 $r \in \mathrm{R}$，$\mu_0(r) = \mathrm{nil}$。设 $\boldsymbol{u}[i] = \gamma_i$ 为步骤 $i$ 的转移所选的标记元组，并固定满足集合 $v_x[i] \coloneqq \mu_i(r_x)$ 的 $\boldsymbol{v}$。在每一步 $i$，B 中所选的规则都来自 A 的某条规则 $q_{i-1} \xrightarrow{w[i]\ \boldsymbol{u}[i]\ \Phi_i} q_i$，且 $\tau(\Phi_i)$ 在 $\overset{\triangleright}{r}_x = v_x[i]$ 下成立，因此 A 沿着相同的状态序列接受 $\otimes(w, \boldsymbol{u}, \varpi, \boldsymbol{v})$。此外，对于 $i > 1$，$\mathrm{Eta}(\boldsymbol{u}[i])$ 会在 $u_x[i] = \circ$ 时强制满足 $v_x[i] = v_x[i-1]$，因此 $\otimes(w, \boldsymbol{u}, \varpi, \boldsymbol{v}) \models \bigwedge_x \eta_x$。

**方向“⇐”**。反过来，假设 $\otimes(w, \boldsymbol{u}, \varpi, \boldsymbol{v}) \in \mathcal{L}(\mathcal{A}) \cap \mathcal{L}(\bigwedge_x \eta_x)$，并固定 A 的一个接受运行

$$q_0 \xrightarrow{(w[1], \boldsymbol{u}[1]), \Phi_1} q_1 \Rightarrow \cdots \xrightarrow{(w[n], \boldsymbol{u}[n]), \Phi_n} q_n.$$

定义寄存器赋值如下：$\mu_0(r) =$空值，以及 $\mu_i(r_x) = v_x[i]$。由于 $\mathcal{A}$ 在步骤 $i$ 的规则在当前 $\mathbb{A}'$ 字母上已启用，因此 $\tau(\Phi_i)$ 在 $\mathcal{B}$ 的步骤 $i$ 成立。并且对于 $i > 1$，无论何时 $u_x[i] = \circ$，$\bigwedge_x \eta_x$ 都会推出 $v_x[i] = v_x[i-1]$，也就是说，$\mathrm{Eta}(\boldsymbol{u}[i])$ 成立。因此，$\mathcal{B}$ 可以沿着相同的状态序列运行，并接受 $\otimes(w, \varpi)$。这就证明了所需的语言相等关系。◀

为了证明关于句子 $F(\varphi_0)$ 的引理 19，我们先证明下面这个技术性更强的引理 25。该引理还刻画了对于 $\varphi_0$ 的子公式 $\varphi$， $G(\varphi, S)$ 的行为，从而能够对 $\varphi_0$ 的结构进行归纳证明。对于 $\varphi_0$ 的子公式 $\varphi$，我们区分 $\mathcal{U}_\uparrow$ 的以下两个子集：

$$\mathcal{X}(\varphi) \coloneqq \mathcal{U}_\uparrow \cap \mathrm{free}_1(\varphi)\,, \qquad \mathcal{Y}(\varphi) \coloneqq \mathcal{U}_\uparrow \cap \mathrm{var}_1(\varphi) \setminus \mathrm{free}_1(\varphi)\,.$$

固定 $L \in \mathbb{N}$ 和一个作用域 $I \subseteq [L]_{\mathbb{N}\circ}$。令 $\boldsymbol{u}\colon A \to B^L$ 和 $\check{\boldsymbol{u}}\colon A' \to B^{|I|}$ 为词族，其中 $A' \subseteq A$。若对于每个 $x \in A'$， $u_x[I] = \check{u}_x$，则称 $\boldsymbol{u}$ 与 $\check{\boldsymbol{u}}$ 在 $I$ 上一致。

▶ **例24**。令 $X = \{\mathtt{x}, \mathtt{y}, \mathtt{z}\}$、$X' = \{\mathtt{y}, \mathtt{z}\}$和 $I = [3,5]_{\mathbb{N}\circ}$。令 $\boldsymbol{u}$ 和 $\check{\boldsymbol{u}}$ 为以下词族：

$$\otimes(\check{\boldsymbol{u}}) = \begin{matrix}\check{u}_\mathtt{y}\colon\\ \check{u}_\mathtt{z}\colon\end{matrix}\begin{bmatrix}\mathtt{d}\\ \mathtt{g}\end{bmatrix}\begin{bmatrix}\mathtt{e}\\ \mathtt{h}\end{bmatrix}\begin{bmatrix}\mathtt{f}\\ \mathtt{i}\end{bmatrix}, \qquad \otimes(\boldsymbol{u}) = \begin{matrix}u'_\mathtt{x}\\ :\,u'_\mathtt{y}\colon\\ u'_\mathtt{z}\colon\end{matrix}\begin{bmatrix}\mathtt{a}\\ \mathtt{b}\\ \mathtt{c}\end{bmatrix}\begin{bmatrix}\mathtt{a}\\ \mathtt{b}\\ \mathtt{c}\end{bmatrix}\begin{bmatrix}\mathtt{a}\\ \mathtt{d}\\ \mathtt{g}\end{bmatrix}\begin{bmatrix}\mathtt{a}\\ \mathtt{e}\\ \mathtt{h}\end{bmatrix}\begin{bmatrix}\mathtt{a}\\ \mathtt{f}\\ \mathtt{i}\end{bmatrix}\begin{bmatrix}\mathtt{a}\\ \mathtt{b}\\ \mathtt{c}\end{bmatrix}\circ$$

1 2 3 4 5 6 ($[3,5]_{\mathbb{N}}$)

那么， $\boldsymbol{u}$ 与 $\check{\boldsymbol{u}}$ 在 $I$ 上一致。

▶ 引理 25。固定 $\varphi_0$ 的一个子公式 $\varphi$，以及一个解释 $\mathfrak{I} = (\mathbb{W}(w, \varpi)\ \nu, I)$，其中 $I$ 非空。令 $L = |w|$。固定一个超作用域族 $(I_x)_{x \in \mathcal{X}(\varphi)}$，使得 $I \subseteq I_x$，且对于每个 $x \in$，都有 $\nu(y) \in I_x$ $X(\varphi)$ 和 $U_\downarrow \cap \mathit{free}_1(\varphi)$。令 $S$ 为一个新的二阶变量，并令 $\mathfrak{I}_{\boldsymbol{u},\boldsymbol{v}}$ 为 MSO $_S[<, \Sigma', \mathbb{A}']$ 的解释 $(\mathbb{W}(w, \boldsymbol{u}, \varpi, \boldsymbol{v})\ \nu[S \leftarrow I])$。定义两个性质 $P_\mathcal{X}$ 和 $P_\mathcal{Y}$：

- $P_\mathcal{X}(\varphi, \boldsymbol{u}, \boldsymbol{v})$ *holds iff* $u_x[I_x] \in \bullet\circ^*$ *and* $v_x[I_x] = \varpi[\nu(x)]^{|I_x|}$ *for every* $x \in \mathcal{X}(\varphi)$,
- $P_\mathcal{Y}(\varphi, \check{\boldsymbol{u}}, \check{\boldsymbol{v}})$ *holds iff* $\mathbb{W}(w[I], \check{\boldsymbol{u}}, \varpi[I], \check{\boldsymbol{v}}) \models \eta_y$ *for every* $y \in \mathcal{Y}(\varphi)$.

那么以下两个断言成立：
*(C1)* 若 $\mathfrak{I} \models \varphi$，则存在中缀族 $\check{\boldsymbol{u}}\colon Y(\varphi) \to \{\bullet, \circ\}^{|I|}$ 和 $\check{\boldsymbol{v}}\colon Y(\varphi) \to \mathbb{A}^{|I|}$，满足 $P_\mathcal{Y}(\varphi, \check{\boldsymbol{u}}, \check{\boldsymbol{v}})$；并且对于所有与 $\check{\boldsymbol{u}}$ 和 $\check{\boldsymbol{v}}$ 在 $I$ 上一致的 $\boldsymbol{u}\colon U_\uparrow \to \{\bullet, \circ\}^L$ 和 $\boldsymbol{v}\colon U_\uparrow \to \mathbb{A}^L$，若 $P_\mathcal{X}(\varphi, \boldsymbol{u}, \boldsymbol{v})$ 成立，则 $\mathfrak{I}_{\boldsymbol{u},\boldsymbol{v}} \models G(\varphi, S)$。*(C2)* 若 $\mathfrak{I}_{\boldsymbol{u},\boldsymbol{v}} \models G(\varphi, S)$，且对于某个 $\boldsymbol{u}\colon U_\uparrow \to \{\bullet, \circ\}^L$ 和 $\boldsymbol{v}\colon U_\uparrow \to \mathbb{A}^L$,，性质 $P_\mathcal{X}(\varphi, \boldsymbol{u}, \boldsymbol{v})$ 和 $P_\mathcal{Y}(\varphi, \boldsymbol{u}[I], \boldsymbol{v}[I])$ 均成立，则 $\mathfrak{I} \models \varphi$。

请注意，在引理 25 中，我们总是在环境作用域 $\mathrm{pos}(w)$ 上求值翻译后的公式 $G(\varphi, S)$，因此在相应的解释中省略区间分量。我们首先说明引理 25 如何推出引理 19。

**假设引理 25，证明引理 19**。固定任意一个词 $\otimes(w, \varpi) \in (\Sigma \times \mathbb{A})^*$。当 $w$ 为空时，断言显然成立：事实上，空当且仅当$\mathrm{pos}(w) = \emptyset$。以下假设 $w$ 非空。我们需要证明

$$\otimes(w, \varpi) \in \mathcal{L}(\varphi_0) \iff \exists \check{\boldsymbol{u}}, \check{\boldsymbol{v}}.\ \otimes(w, \check{\boldsymbol{u}}, \varpi, \check{\boldsymbol{v}}) \in \mathcal{L}\big(\exists S.\ G(\varphi_0, S) \wedge \forall y\ S(y) \wedge \textstyle\bigwedge_{x \in \mathcal{U}_\uparrow} \eta_x\big)\,.$$

根据左侧语言成员关系的定义，并且由于 $I_0 \coloneqq \mathrm{pos}(w)$ 是满足 $\forall y\ S(y)$ 的唯一位置集合，只需证明

$$\mathbb{W}(w, \varpi), \nu_0, I_0 \models \varphi_0 \iff \exists \check{\boldsymbol{u}}, \check{\boldsymbol{v}}.\ \mathbb{W}(w, \check{\boldsymbol{u}}, \varpi, \check{\boldsymbol{v}}), \nu_0[S \leftarrow I_0] \models G(\varphi_0, S) \wedge \textstyle\bigwedge_{x \in \mathcal{U}_\uparrow} \eta_x\,,$$

其中 $\nu_0$ 是空赋值。注意，$\mathcal{X}(\varphi_0) = \emptyset$ 以及 $\mathcal{Y}(\varphi_0) = \mathcal{U}_\uparrow$。所需的等价性由引理 25 结合 $I = I_0$直接得到。

方向“⇒”。根据断言（C1），我们得到满足 $P_{\mathcal{Y}}(\varphi_0, \check{\boldsymbol{u}}, \check{\boldsymbol{v}})$ 的 $\check{\boldsymbol{u}}$、$\check{\boldsymbol{v}}$。任何与 $I_0$ 上的 $\check{\boldsymbol{u}}$、$\check{\boldsymbol{v}}$ 一致的族 $\boldsymbol{u}$、$\boldsymbol{v}$ 都必须分别与 $\check{\boldsymbol{u}}$、$\check{\boldsymbol{v}}$ 一致。此外，由于 $\mathrm{X}(\varphi_0) = \emptyset$，$P_{\mathcal{X}}(\varphi_0, \check{\boldsymbol{u}}, \check{\boldsymbol{v}})$ 为空真。因此，$\mathbb{W}(w, \check{\boldsymbol{u}}, \varpi, \check{\boldsymbol{v}})\ \nu_0[S \leftarrow I_0] \models G(\varphi_0, S)$，正如所需。方向“⇐”。固定 $\check{\boldsymbol{u}}$、$\check{\boldsymbol{v}}$，使得 $\mathbb{W}(w, \check{\boldsymbol{u}}, \varpi, \check{\boldsymbol{v}})\ \nu_0[S \leftarrow I_0] \models G(\varphi_0, S) \wedge \bigwedge_{x \in \mathrm{U}_\uparrow} \eta_x$。根据定义，$P_{\mathcal{Y}}(\varphi_0, \check{\boldsymbol{u}}, \check{\boldsymbol{v}})$ 成立；而由于 $\mathrm{X}(\varphi_0) = \emptyset$，$P_{\mathcal{X}}(\varphi_0, \check{\boldsymbol{u}}, \check{\boldsymbol{v}})$ 为空真。根据断言（C2），我们得出 $\mathbb{W}(w, \varpi)\ \nu_0, I_0 \models \varphi_0$。◀

引理 25 的证明。固定 $\varphi_0$ 的一个子公式 $\varphi$ 以及一个解释 $\mathfrak{I} = (\mathbb{W}(w, \varpi)\ \nu, I)$。固定一族超作用域 $(I_x)^{x \in \mathcal{X}(\varphi)}$，使得 $I \subseteq I_x$，并且对于每个 $x \in \mathrm{X}(\varphi)$ 和 $y \in \mathrm{U}_\downarrow \cap \mathrm{free}_1(\varphi)$，都有 $\nu(y) \in I_x$。我们对 $\varphi$ 的结构进行归纳。共有十种主要情形，分别对应于 $\mathrm{MSO}_{\mathrm{S}}[<, \Sigma, \mathbb{A}]$ 的语法规则。在每种情形中，我们都证明断言 (C1) 和 (C2)。

**情形1** $(\varphi \equiv R(x_1, x_2, \ldots, x_r))$。令 $\mathcal{V} := \{x_1, x_2, \ldots, x_r\}$。回忆一下，$|\mathcal{V} \cap \mathcal{U}_\downarrow| \leq 1$。

**情形1A** $(\mathcal{V} \cap \mathcal{U}_\downarrow = \{z\}$ 对某个 $z)$。令 $p = \nu(z)$。

（C1）假设 $\mathfrak{I} \models \varphi$。根据定义，$(\varpi[\nu(x_1)], \ldots, \varpi[\nu(x_r)]) \in R$。注意，$\mathrm{Y}(\varphi) = \emptyset$。令 $\check{\boldsymbol{u}}$、$\check{\boldsymbol{v}}$ 为唯一的空族。于是 $P_{\mathcal{Y}}(\varphi, \check{\boldsymbol{u}}, \check{\boldsymbol{v}})$ 显然成立。任取满足 $P_{\mathcal{X}}(\varphi, \boldsymbol{u}, \boldsymbol{v})$ 的 $\boldsymbol{u}, \boldsymbol{v}$。这些族显然与 $\check{\boldsymbol{u}}$ 和 $\check{\boldsymbol{v}}$ 在 $I$ 上一致。剩下要证明的是，对每个 $i$，$G(\varphi, S)$ 中的项 $t_{i,z}$ 的求值结果为 $\varpi[\nu(x_i)]$。当 $x_i \equiv z$ 时，这一点显而易见。否则，$t_{i,z}$ 的求值结果为 $v_{x_i}[p]$。由于 $p \in I_{x_i}$，性质 $P_{\mathcal{X}}$ 得出 $v_{x_i}[p] = \varpi[\nu(x_i)]$，，正如所需。

（C2）固定 $\boldsymbol{u}, \boldsymbol{v}$，使得 $P_{\mathcal{X}}(\varphi, \boldsymbol{u}, \boldsymbol{v})$、$P_{\mathcal{Y}}(\varphi, \boldsymbol{u}[I], \boldsymbol{v}[I])$，并且 $\mathfrak{I}_{\boldsymbol{u}, \boldsymbol{v}} \models G(\varphi, S)$。如断言（C2）中所述，我们对每个 $i$ 都得到 $\varpi[\nu(x_i)] = v_{x_i}[p]$，因此 $\mathfrak{I} \models \varphi$。

**情形1B** $(\mathcal{V} \cap \mathcal{U}_\downarrow = \emptyset)$。

(C1) 假设 $\mathfrak{I} \models \varphi$。令 $\check{\boldsymbol{u}}$、$\check{\boldsymbol{v}}$ 为由 $\mathrm{Y}(\varphi)$ 索引的唯一空中缀族，并令 $\boldsymbol{u}$、$\boldsymbol{v}$ 为任意词族，使其在 $I$ 上与它们一致，并满足 $P_{\mathcal{X}}(\varphi, \boldsymbol{u}, \boldsymbol{v})$。于是，$P_{\mathcal{Y}}(\varphi, \check{\boldsymbol{u}}, \check{\boldsymbol{v}})$ 真空成立。我们必须证明 $\mathfrak{I}_{\boldsymbol{u}, \boldsymbol{v}} \models \exists z (z \in S \wedge R(t_1, z, \ldots\ldots, t_r, z))$。任取 $p \in I$（回忆 $I \neq \emptyset$）。由与情形1A 中相同的论证，我们有 $\mathbb{W}(w, \boldsymbol{u}, \varpi, \boldsymbol{v})\ \nu[z \leftarrow p, S \leftarrow I] \models z \in S \wedge R(t_1, z, \ldots\ldots, t_r, z)$。因此，$\mathfrak{I}_{\boldsymbol{u}, \boldsymbol{v}} \models G(\varphi, S)$。

(C2) 固定 $\boldsymbol{u}, \boldsymbol{v}$，使得 $\mathfrak{I}_{\boldsymbol{u}, \boldsymbol{v}} \models G(\varphi, S)$，并且性质 $P_{\mathcal{X}}(\varphi, \boldsymbol{u}, \boldsymbol{v})$ 和 $P_{\mathcal{Y}}(\varphi, \boldsymbol{u}[I], \boldsymbol{v}[I])$ 成立。根据定义，存在 $p \in I$，使得 $\mathbb{W}(w, \boldsymbol{u}, \varpi, \boldsymbol{v})\ \nu[z \leftarrow p, S \leftarrow I] \models R(t_1, z, \ldots\ldots, t_r, z)$。由与情形1A 中相同的论证，我们得出 $\mathfrak{I} \models \varphi$。

**情形2** $(\varphi \equiv x < y)$。显然。

(C1) 若 $\mathfrak{I} \models x < y$，则 $\nu(x) < \nu(y)$，因此对于任意 $\boldsymbol{u}$，$\mathfrak{I}_{\boldsymbol{u}, \boldsymbol{v}} \models x < y$ $\boldsymbol{v}$ 与由 $\mathcal{Y}(\varphi) = \emptyset$ 索引的唯一平凡中缀族 $\check{\boldsymbol{u}}$，$\check{\boldsymbol{v}}$ 一致。(C2) 若对于某个 $\boldsymbol{u}$，$\mathfrak{I}_{\boldsymbol{u}, \boldsymbol{v}} \models x < y$ $\boldsymbol{v}$，则 $\mathfrak{I} \models x < y$。

**情形 3** $(\varphi \equiv a(x))$。显然。

(C1) 若 $\mathfrak{I} \models a(x)$，则 $w[\nu(x)] = a$，因此对于任意 $\boldsymbol{u}$，$\mathfrak{I}_{\boldsymbol{u}, \boldsymbol{v}} \models w[x] = a$ $\boldsymbol{v}$ 与由 $\mathcal{Y}(\varphi) = \emptyset$ 索引的唯一平凡中缀族 $\check{\boldsymbol{u}}$，$\check{\boldsymbol{v}}$ 一致。
(C2) 若对于某个 $\boldsymbol{u}$，$\mathfrak{I}_{\boldsymbol{u}, \boldsymbol{v}} \models w[x] = a$ $\boldsymbol{v}$，则 $\mathfrak{I} \models a(x)$。

**情形 4** $(\varphi \equiv X(y))$。显然。

（C1）如果 $\mathfrak{I} \models X(y)$，那么 $\nu(y) \in \nu(X)$，因此对于任意与由 $\check{\boldsymbol{u}}, \check{\boldsymbol{v}}$（由 $\mathcal{Y}(\varphi) = \emptyset$ 索引）构成的唯一平凡中缀族相一致的 $\boldsymbol{u}, \boldsymbol{v}$，都有 $\mathfrak{I}_{\boldsymbol{u}, \boldsymbol{v}} \models X(y)$。（C2）如果对于某个 $\boldsymbol{u}, \boldsymbol{v}$，$\mathfrak{I}_{\boldsymbol{u}, \boldsymbol{v}} \models X(y)$，那么 $\mathfrak{I} \models X(y)$。

**情形 5** $(\varphi \equiv \psi \wedge \gamma)$。注意 $\mathcal{Y}(\varphi) = \mathcal{Y}(\psi) \cup \cdot\, \mathcal{Y}(\gamma)$。

（C1）假设 $\mathfrak{I} \models \psi \wedge \gamma$。那么 $\mathfrak{I} \models \psi$ 和 $\mathfrak{I} \models \gamma$。根据归纳假设中的断言（C1），存在中缀族 $\check{}\boldsymbol{u}'$、$\check{}\boldsymbol{v}'$，满足 $P_{\mathcal{Y}}(\psi, \check{}\boldsymbol{u}', \check{}\boldsymbol{v}')$，使得对于每个与它们在 $I$ 上一致并满足 $P_{\mathcal{X}}(\psi, \boldsymbol{u}', \boldsymbol{v}')$ 的 $\boldsymbol{u}'$、$\boldsymbol{v}'$，都有 $\mathfrak{I}_{\boldsymbol{u}',\boldsymbol{v}'} \models G(\psi, S)$。同理，存在 $\check{}\boldsymbol{u}''$、$\check{}\boldsymbol{v}''$，可作为 $G(\gamma, S)$ 对应命题的见证。令 $\check{}\boldsymbol{u} = \check{}\boldsymbol{u}' \cup \check{}\boldsymbol{u}''$，且 $\check{}\boldsymbol{v} = \check{}\boldsymbol{v}' \cup \check{}\boldsymbol{v}''$。则 $P_{\mathcal{Y}}(\varphi, \check{}\boldsymbol{u}, \check{}\boldsymbol{v})$ 成立。任取满足 $P_{\mathcal{X}}(\varphi, \boldsymbol{u}, \boldsymbol{v})$，且与 $\check{}\boldsymbol{u}$、$\check{}\boldsymbol{v}$ 在 $I$ 上一致的 $\boldsymbol{u}$、$\boldsymbol{v}$。由于 $\boldsymbol{u}$、$\boldsymbol{v}$ 也与 $\check{}\boldsymbol{u}'$、$\check{}\boldsymbol{v}'$ 在 $I$ 上一致，并满足 $P_{\mathcal{X}}(\psi, \boldsymbol{u}, \boldsymbol{v})$ 和 $P_{\mathcal{Y}}(\psi, \boldsymbol{u}[I], \boldsymbol{v}[I])$，因此可得 $\mathfrak{I}_{\boldsymbol{u},\boldsymbol{v}} \models G(\psi, S)$。同样，$\boldsymbol{u}$、$\boldsymbol{v}$ 与 $\check{}\boldsymbol{u}''$、$\check{}\boldsymbol{v}''$ 在 $I$ 上一致，并满足 $P_{\mathcal{X}}(\gamma, \boldsymbol{u}, \boldsymbol{v})$ 和 $P_{\mathcal{Y}}(\gamma, \boldsymbol{u}[I], \boldsymbol{v}[I])$，故 $\mathfrak{I}_{\boldsymbol{u},\boldsymbol{v}} \models G(\gamma, S)$。因此 $\mathfrak{I}_{\boldsymbol{u},\boldsymbol{v}} \models G(\varphi, S)$。

（C2）设 $\boldsymbol{u}$、$\boldsymbol{v}$ 满足 $P_{\mathcal{X}}(\varphi, \boldsymbol{u}, \boldsymbol{v})$ 和 $P_{\mathcal{Y}}(\varphi, \boldsymbol{u}[I], \boldsymbol{v}[I])$，且 $\mathfrak{I}_{\boldsymbol{u},\boldsymbol{v}} \models G(\varphi, S)$。则 $\mathfrak{I}_{\boldsymbol{u},\boldsymbol{v}} \models G(\psi, S)$ 且 $\mathfrak{I}_{\boldsymbol{u},\boldsymbol{v}} \models G(\gamma, S)$。此外，$P_{\mathcal{X}}(\psi, \boldsymbol{u}, \boldsymbol{v})$ 和 $P_{\mathcal{Y}}(\psi, \boldsymbol{u}[I], \boldsymbol{v}[I])$ 成立，$\gamma$ 的情形也类似。根据归纳假设中的断言（C2），有 $\mathfrak{I} \models \psi$ 和 $\mathfrak{I} \models \gamma$，因此 $\mathfrak{I} \models \psi \wedge \gamma$。

**情形 6** ($\varphi \equiv \psi \vee \gamma$)。证明与情形 5 类似，只需将 $\wedge$ 替换为 $\vee$。

**情形 7** ($\varphi \equiv \neg\psi$)。注意，$\mathcal{Y}(\psi) = \mathcal{Y}(\varphi) = \emptyset$，因为否定之下不能量化顶层变量。

（C1）假设 $\mathfrak{I} \models \neg\psi$。令 $\check{}\boldsymbol{u}$、$\check{}\boldsymbol{v}$ 为唯一的空中缀族；则 $P_{\mathcal{Y}}(\varphi, \check{}\boldsymbol{u}, \check{}\boldsymbol{v})$ 显然成立。反设，对于某个满足 $P_{\mathcal{X}}(\varphi, \boldsymbol{u}, \boldsymbol{v})$ 且与 $\check{}\boldsymbol{u}$、$\check{}\boldsymbol{v}$ 平凡相容的 $\boldsymbol{u}$，$\boldsymbol{v}$，有 $\mathfrak{I}_{\boldsymbol{u},\boldsymbol{v}} \models G(\psi, S)$。于是断言（C2）推出 $\mathfrak{I} \models \psi$，这与假设矛盾。

（C2）假设对于某些 $\boldsymbol{u}$、$\boldsymbol{v}$，$\mathfrak{I}_{\boldsymbol{u},\boldsymbol{v}} \models \neg G(\psi, S)$ 成立，且 $P_{\mathcal{X}}(\varphi, \boldsymbol{u}, \boldsymbol{v})$ 和 $P_{\mathcal{Y}}(\varphi, \boldsymbol{u}[I], \boldsymbol{v}[I])$ 成立。为导出矛盾，假设 $\mathfrak{I} \models \psi$。根据归纳假设的断言（C1），对于唯一的空族 $\check{}\boldsymbol{u}$、$\check{}\boldsymbol{v}$（它们与 $\boldsymbol{u}$、$\boldsymbol{v}$ 相容），我们有 $\mathfrak{I}_{\boldsymbol{u},\boldsymbol{v}} \models G(\psi, S)$，这与假设矛盾。

**情况 8** （$\varphi \equiv \exists X\ \psi$）。显然。

（C1）假设 $\mathfrak{I} \models \exists X\ \psi$。那么存在 $P \subseteq I$，使得 $\mathbb{W}(w, \varpi)\ \nu[X \leftarrow P], I \models \psi$。根据归纳假设的断言（C1），存在一些中缀族 $\check{}\boldsymbol{u}$、$\check{}\boldsymbol{v}$，满足$P_{\mathcal{Y}}(\psi, \check{}\boldsymbol{u}, \check{}\boldsymbol{v})$，因此对于每个与它们在 $I$ 上相容且满足 $P_{\mathcal{X}}(\psi, \boldsymbol{u}, \boldsymbol{v})$ 的 $\boldsymbol{u}, \boldsymbol{v}$，满足关系 $\mathbb{W}(w, \boldsymbol{u}, \varpi, \boldsymbol{v})\ \nu[X \leftarrow P, S \leftarrow I] \models G(\psi, S)$ 成立。我们沿用 $\check{}\boldsymbol{u}$、$\check{}\boldsymbol{v}$；任取与它们在 $I$ 上相容且满足 $P_{\mathcal{X}}(\varphi, \boldsymbol{u}, \boldsymbol{v})$ 的 $\boldsymbol{u}, \boldsymbol{v}$。注意到$P_{\mathcal{Y}}(\varphi, \check{}\boldsymbol{u}, \check{}\boldsymbol{v})$ 和 $P_{\mathcal{X}}(\psi, \boldsymbol{u}, \boldsymbol{v})$ 成立，因此 $\mathbb{W}(w, \boldsymbol{u}, \varpi, \boldsymbol{v})\ \nu[X \leftarrow P, S \leftarrow I] \models X \subseteq S$。根据定义，$\mathfrak{I}_{\boldsymbol{u},\boldsymbol{v}} \models G(\varphi, S)$ 成立。

（C2）假设对于某些满足$P_{\mathcal{X}}(\varphi, \boldsymbol{u}, \boldsymbol{v})$和 $P_{\mathcal{Y}}(\varphi, \boldsymbol{u}[I], \boldsymbol{v}[I])$的 $\boldsymbol{u}, \boldsymbol{v}$，有$\mathfrak{I}_{\boldsymbol{u},\boldsymbol{v}} \models G(\varphi, S) \equiv \exists X.\ X \subseteq S \wedge G(\psi, S)$。根据定义，存在某个 $P \subseteq \mathrm{pos}(w)$，使得 $\mathbb{W}(w, \boldsymbol{u}, \varpi, \boldsymbol{v})\ \nu[X \leftarrow P, S \leftarrow I] \models X \subseteq S \wedge G(\psi, S)$。容易看出，$P_{\mathcal{X}}(\psi, \boldsymbol{u}, \boldsymbol{v})$且 $P_{\mathcal{Y}}(\psi, \boldsymbol{u}[I], \boldsymbol{v}[I])$。注意 $P \subseteq I$。根据归纳假设的断言（C2），我们得到 $\mathbb{W}(w, \varpi)\ \nu[X \leftarrow P], I \models \psi$，因此 $\mathfrak{I} \models \varphi$。

**情况 9** ($\varphi \equiv \exists x\ \psi$)。

**情况 9A** ($x \notin \mathcal{U}\!\uparrow$)。显然。注意 $\mathcal{Y}(\varphi) = \emptyset$，否则 $x$ 将处于顶层。

（C1）假设 $\mathfrak{I} \models \varphi$；根据定义，存在 $p \in I$ 使得 $\mathbb{W}(w, \varpi)\ \nu[x \leftarrow p], I \models \psi$。根据归纳假设中的断言（C1），存在 $\check{}\boldsymbol{u}$、$\check{}\boldsymbol{v}$ 满足 $P_{\mathcal{Y}}(\psi, \check{}\boldsymbol{u}, \check{}\boldsymbol{v})$，使得对于每个与它们在 $I$ 上一致且满足 $P_{\mathcal{X}}(\psi, \boldsymbol{u}, \boldsymbol{v})$ 的 $\boldsymbol{u}, \boldsymbol{v}$，满足关系 $\mathbb{W}(w, \boldsymbol{u}, \varpi, \boldsymbol{v})\ \nu[x \leftarrow p, S \leftarrow I] \models G(\psi, S)$ 都成立，因此根据定义，$\mathfrak{I}_{\boldsymbol{u},\boldsymbol{v}} \models G(\varphi, S)$；这就是所需的断言。

（C2）固定 $\boldsymbol{u}$，$\boldsymbol{v}$使得 $P_{\mathcal{X}}(\varphi, \boldsymbol{u}, \boldsymbol{v})$ 且 $P_{\mathcal{Y}}(\varphi, \boldsymbol{u}[I], \boldsymbol{v}[I])$，并且 $\mathfrak{I}_{\boldsymbol{u},\boldsymbol{v}} \models G(\varphi, S)$。根据定义，存在某个 $p \in \mathrm{pos}(w)$，使得 $\mathbb{W}(w, \boldsymbol{u}, \varpi, \boldsymbol{v})\ \nu[x \leftarrow p, S \leftarrow I] \models x \in S \wedge G(\psi, S)$。

注意，这意味着 $p \in I$。根据归纳假设的断言（C2），我们得到 $\mathbb{W}(w,\ \varpi)\ \nu[x \leftarrow p],\ I \models \psi$；根据定义，$\mathfrak{I} \models \varphi$，如所需。

**情况 9B** ($x \in \mathcal{U}$)。注意 $\mathcal{Y}(\varphi) = \mathcal{Y}(\psi) \cup \{x\}$。此处，$G(\varphi, S) \equiv \exists x.\ x \in S \wedge G(\psi, S) \wedge \theta_{x,S}$ .

（C1）假设 $\mathfrak{I} \models \varphi$。那么存在 $p \in I$，使得 $\mathbb{W}(w, \varpi)\ \nu[x \leftarrow p], I \models \psi$。根据归纳假设的断言（C1），取满足 $P_{\mathcal{Y}}(\psi, \check{\boldsymbol{u}}', \check{\boldsymbol{v}}')$ 的 $\check{\boldsymbol{u}}'$、$\check{\boldsymbol{v}}'$，其中 $I_x = I$；对于所有在 $I$ 上与它们一致且满足 $P_{\mathcal{X}}(\psi, \boldsymbol{u}, \boldsymbol{v})$ 的 $\boldsymbol{u}, \boldsymbol{v}$，都有 $\mathbb{W}(w, \boldsymbol{u}, \varpi, \boldsymbol{v})\ \nu[x \leftarrow p, S \leftarrow I] \models G(\psi, S)$。令 $\check{\boldsymbol{u}} = \check{\boldsymbol{u}}' \cup \{u_x \mapsto \bullet \circ {}^{|I|-1}\}$，且令 $\check{\boldsymbol{v}} = \check{\boldsymbol{v}}' \cup \{v_x \mapsto \varpi[p]^{|I|}\}$。那么 $P_{\mathcal{Y}}(\varphi, \check{\boldsymbol{u}}, \check{\boldsymbol{v}})$ 成立。任取与 $\check{\boldsymbol{u}}, \check{\boldsymbol{v}}$ 一致且满足 $P_{\mathcal{X}}(\varphi, \boldsymbol{u}, \boldsymbol{v})$ 的 $\boldsymbol{u}, \boldsymbol{v}$。令 $\mathfrak{I}'_{\boldsymbol{u},\boldsymbol{v}} = (\mathbb{W}(w, \boldsymbol{u}, \varpi, \boldsymbol{v})\ \nu[x \leftarrow p, S \leftarrow I])$。根据中缀轨道 $\check{u}_x, \check{v}_x$ 的构造，性质 $P_{\mathcal{X}}(\psi, \boldsymbol{u}, \boldsymbol{v})$ 和 $P_{\mathcal{Y}}(\psi, \check{\boldsymbol{u}}, \check{\boldsymbol{v}})$ 也成立。因此，$\mathfrak{I}'_{\boldsymbol{u},\boldsymbol{v}} \models G(\psi, S)$。此外，根据 $\check{u}_x$ 和 $\check{v}_x$ 的选取，我们还有 $\mathfrak{I}'_{\boldsymbol{u},\boldsymbol{v}} \models \theta_{x,S}$。因此 $\mathfrak{I}_{\boldsymbol{u},\boldsymbol{v}} \models G(\varphi, S)$。

（C2）固定 $\boldsymbol{u}, \boldsymbol{v}$，使得 $P_{\mathcal{X}}(\varphi, \boldsymbol{u}, \boldsymbol{v})$、$P_{\mathcal{Y}}(\varphi, \boldsymbol{u}[I], \boldsymbol{v}[I])$ 以及 $\mathfrak{I}_{\boldsymbol{u},\boldsymbol{v}} \models G(\varphi, S)$ 均成立。令 $\mathfrak{I}'_{\boldsymbol{u},\boldsymbol{v},p} = (\mathbb{W}(w, \boldsymbol{u}, \varpi, \boldsymbol{v})\ \nu[x \leftarrow p, S \leftarrow I])$。根据定义，存在 $p \in \mathrm{pos}(w)$，使得 $\mathfrak{I}'_{\boldsymbol{u},\boldsymbol{v},p} \models x \in S \wedge G(\psi, S) \wedge \theta_{x,S}$。特别地，$p \in S$。由于 $\mathfrak{I}'_{\boldsymbol{u},\boldsymbol{v},p} \models \theta_{x,S}$，我们有 $v_x[p] = \varpi[p]$ 和 $u_x[I] \in \bullet \circ {}^*$。利用假设 $\mathbb{W}(\boldsymbol{u}[I], \boldsymbol{v}[I]) \models \eta_x$，通过简单的归纳法可证明，对每个 $q \in I$，$v_x[q] = \varpi[p]$。因此，$P_{\mathcal{X}}(\psi, \boldsymbol{u}, \boldsymbol{v})$ 和 $P_{\mathcal{Y}}(\psi, \boldsymbol{u}[I], \boldsymbol{v}[I])$ 均成立。根据归纳假设中的断言（C2），我们得到 $\mathbb{W}(w, \varpi)\ \nu[x \leftarrow p], I \models \psi$。由于 $p \in I$，我们得出结论：$\mathfrak{I} \models \varphi$。

**情形 10** ($\varphi \equiv \mathcal{E}_X\ \psi$)。令 $S'$ 为一个新的二阶变量。由于 $I \neq \emptyset$，我们有 $\mathrm{subs}(I, \nu(X)) \neq \emptyset$。令 $\mathrm{subs}(I, \nu(X)) = \{J_1, \ldots, J_k\}$，按递增顺序枚举。

（C1）假设 $\mathfrak{I} \models \mathcal{E}_X\ \psi$。那么，对于每个 $i$，$\mathbb{W}(w, \varpi)\ \nu, J_i \models \psi$。根据归纳假设中的断言（C1），对于每个 $i \in [k]_{\mathbb{N}}$，都存在中缀族 $\check{\boldsymbol{u}}^{(i)}$、$\check{\boldsymbol{v}}^{(i)}$，满足 $P_{\mathcal{Y}}\big(\psi, \check{\boldsymbol{u}}^{(i)}, \check{\boldsymbol{v}}^{(i)}\big)$，使得对于每个与它们在 $J_i$ 上一致且满足 $P_{\mathcal{X}}\big(\psi, \boldsymbol{u}^{(i)}, \boldsymbol{v}^{(i)}\big)$ 的 $\boldsymbol{u}^{(i)}, \boldsymbol{v}^{(i)}$，都有 $\mathbb{W}\big(w, \boldsymbol{u}^{(i)}, \varpi, \boldsymbol{v}^{(i)}\big),\ \nu[S' \leftarrow J_i] \models G(\psi, S')$。令 $\check{\boldsymbol{u}} = \check{\boldsymbol{u}}^{(1)}\check{\boldsymbol{u}}^{(2)} \cdots \check{\boldsymbol{u}}^{(k)}$ 和 $\check{\boldsymbol{v}} = \check{\boldsymbol{v}}^{(1)}\check{\boldsymbol{v}}^{(2)} \cdots \check{\boldsymbol{v}}^{(k)}$，其中拼接是逐点应用于相应的轨迹族。我们断言 $P_{\mathcal{Y}}(\varphi, \check{\boldsymbol{u}}, \check{\boldsymbol{v}})$ 成立。事实上，$\check{\boldsymbol{u}}$ 中的每条轨迹都以 $\bullet$ 开始，因为根据 $P_{\mathcal{Y}}(\psi, \check{\boldsymbol{u}}^{(1)}, \check{\boldsymbol{v}}^{(1)})$，$\check{\boldsymbol{u}}^{(1)}$ 中的每条轨迹都是如此。此外，只要 $s, s+1$ 分别位于某个 $i$ 对应的单个块 $\check{\boldsymbol{u}}^{(i)}$、$\check{\boldsymbol{v}}^{(i)}$ 内，蕴含式 $\check{u}_x[s+1] = \circ \rightarrow \check{v}_x[s] = \check{v}_x[s+1]$ 就成立。但由于 $\check{u}_x^{(i)}$ 的首位字母全都是 $\bullet$，因此该式在块边界处也显然成立。于是，$P_{\mathcal{Y}}(\varphi, \check{\boldsymbol{u}}, \check{\boldsymbol{v}})$ 成立。现在任取与 $\check{\boldsymbol{u}}$、$\check{\boldsymbol{v}}$ 在 $I$ 上一致且满足 $P_{\mathcal{X}}(\varphi, \boldsymbol{u}, \boldsymbol{v})$ 的 $\boldsymbol{u}, \boldsymbol{v}$，并固定任意一个$i \in [k]_{\mathbb{N}}$。由于 $\boldsymbol{u}, \boldsymbol{v}$ 与 $\check{\boldsymbol{u}}^{(i)}$、$\check{\boldsymbol{v}}^{(i)}$ 在 $J_i$ 上一致，且满足 $P_{\mathcal{X}}(\psi, \boldsymbol{u}, \boldsymbol{v})$，我们有$\mathbb{W}(w, \boldsymbol{u}, \varpi, \boldsymbol{v})\ \nu[S \leftarrow I, S' \leftarrow J_i] \models G(\psi, S')$。因此，$\mathfrak{I}_{\boldsymbol{u},\boldsymbol{v}} \models G(\varphi, S)$。

（C2）固定 $\boldsymbol{u}, \boldsymbol{v}$，使得 $\mathfrak{I}_{\boldsymbol{u},\boldsymbol{v}} \models G(\varphi, S)$，且性质 $P_{\mathcal{X}}(\varphi, \boldsymbol{u}, \boldsymbol{v})$ 和 $P_{\mathcal{Y}}(\varphi, \boldsymbol{u}[I], \boldsymbol{v}[I])$ 成立。那么，对于每个 $i \in [k]_{\mathbb{N}}$，$\mathbb{W}(w, \boldsymbol{u}, \varpi, \boldsymbol{v})\ \nu[S \leftarrow I, S' \leftarrow J_i] \models G(\psi, S')$。显然，$P_{\mathcal{X}}(\psi, \boldsymbol{u}, \boldsymbol{v})$ 和 $P_{\mathcal{Y}}(\psi, \boldsymbol{u}[J_i], \boldsymbol{v}[J_i])$ 成立。根据归纳假设中的断言（C2），我们得到 $\mathbb{W}(w, \varpi)\ \nu, J_i \models \psi$ 对每个 $i$ 都成立。因此，$\mathfrak{I} \models \mathcal{E}_X\ \psi$，证毕。◀

## B.2 从自动机到公式

本节证明定理 9 的另一个方向，具体表述如下：

▶ **引理 26.** 固定有限的 $\Sigma$ 和原子 $\mathbb{A}$。对于每个弱猜测 $A \in \mathrm{NRA}^{\mathrm{G}}[\Sigma, \mathbb{A}]$，都存在一个公式 $\varphi \in$ 单子二阶逻辑$_S[<, \Sigma, \mathbb{A}]$，使得 $L(A) = L(\varphi)$。

**证明路线图**。我们将论证分为以下几部分：

- 介绍简单NRA$^{G}$，这是一个表达能力与完整NRA$^{G}$相同的子类；用标准的单子二阶逻辑编
- 码运行骨架；以及一种由寄存器的存活区间引导的几何递归，它借助模态，使我们能够
- 在单子二阶逻辑中检查所有寄存器约束$_{S}[<, \Sigma, \mathbb{A}]$。

ƨ

**简单NRA**。 我们首先定义一类简单*NRA*，它要求寄存器赋值为单射，且不进行寄存器重排。这是文献中关于其他类型寄存器机的一个著名构造 [10,定理 1], ，并且可以直接推广到 NRA$^{G}$ 。因此，我们陈述该引理而不提供证明。

设$\mathcal{A} = (Q, \Sigma, \mathbb{A}, R, Q_{\text{ini}}, Q_{\text{fin}}, \delta) \in \text{NRA}^{G}_{k}[\Sigma, \mathbb{A}]$。若$\mathcal{A}$的每条转移规则($p\ \sigma\ \Phi\ q \in \delta$)$\Phi \equiv \Phi' \wedge \bigwedge_{i \neq j} \overset{\triangleleft}{r}_j \neq \overset{\triangleleft}{r}_i \neq ,,,$)都具有如下形式的寄存器约束，则$\overset{\triangleright}{r}_j \neq \overset{\triangleright}{r}_i \wedge \bigwedge_i \overset{\triangleleft}{r}_i \sim_i \overset{\triangleright}{r}_i \qquad \sim_i \in \ = \neq$称$\mathcal{A}$是简单的：其中{,}。

直观地说，简单NRA不会同时将同一个数据值存储在两个不同的寄存器中，不会在寄存器之间移动数据值，并且会明确说明对于每个 $i$，是$\overset{\triangleleft}{r}_i = \overset{\triangleright}{r}_i$ 还是 $\overset{\triangleleft}{r}_i \neq \overset{\triangleright}{r}_i$ 。

▶ **事实27**。 对于每个 $A \in \text{NRA}^{G}[\Sigma, \mathbb{A}]$，都存在一个简单的 $A' \in \text{NRA}^{G}[\Sigma, \mathbb{A}]$，识别相同的语言；若 $A$ 是弱猜测型的，则 $A'$ 也是如此。

在本节余下部分，固定一个简单且弱猜测型的$\mathcal{A} = (Q, \Sigma, \mathbb{A}, \mathcal{R}\ Q_{\text{ini}}, Q_{\text{fin}}, \delta) \in \text{NRA}^{G}[\Sigma, \mathbb{A}]$。固定 $\mathcal{R} = \{r_1, \ldots, r_k\}$ 的一个枚举。

**运行骨架**。 令 $m = |\delta|$，并固定转移 $\kappa : \delta \to [m]_{\mathbb{N}}$ 的一个枚举。我们引入二阶变量 $\Delta_1, \ldots, \Delta_m$，并令 $\text{Run}_{\mathcal{A}}(\Delta_1, \ldots, \Delta_m)$ 表示：

- 每个位置恰好携带一个转移规则标签，所有转移规则都
- 与字母 $\Sigma$ 匹配，相邻的转移规则标签所对应的状态相匹
- 配，且首条规则从 $Q_{\text{ini}}$ 开始，末条规则以 $Q_{\text{fin}}$ 结束。
-

我们按标准方式描述这些性质：

$$\text{Run}_{\mathcal{A}}(\Delta_1, \Delta_2, \ldots, \Delta_m) := \forall i, j.$$

$$\underbrace{\Big[\bigvee_{t=(\cdot,\sigma,\cdot,\cdot)\in\delta} \Delta_{\kappa(t)}(i) \wedge \sigma(i) \wedge \bigwedge_{\substack{t'\in\delta \\ t'\neq t}} \neg\Delta_{\kappa(t')}(i)\Big]}_{\substack{\text{Every position has exactly one associated transition} \\ \text{rule and it matches the letter } \sigma \in \Sigma.}} \wedge$$

$$\underbrace{\Big[\ \text{succ}(i,j) \to \bigvee_{\substack{t=(q,\cdot,\cdot,r)\in\delta \\ t'=(r,\cdot,\cdot,s)\in\delta}} \Delta_{\kappa(t)}(i) \wedge \Delta_{\kappa(t')}(j)\Big]}_{\text{Consecutive transition rules have matching states.}} \wedge$$

$$\underbrace{\Big[\ \text{first}(j) \to \bigvee_{\substack{t=(q,\cdot,\cdot,\cdot)\in\delta \\ q\in Q_{\text{ini}}}} \Delta_{\kappa(t)}(j)\Big]}_{\text{First transition rule begins in } q \in Q_{\text{ini}}.} \wedge \underbrace{\Big[\ \text{last}(i) \to \bigvee_{\substack{t=(\cdot,\cdot,\cdot,q)\in\delta \\ q\in Q_{\text{fin}}}} \Delta_{\kappa(t)}(i)\Big]}_{\text{Last transition rule ends with } q \in Q_{\text{fin}}.} .$$

接下来需要定义 RegOK($\Delta_1$，……， $\Delta_m$) $\in$ 单子二阶逻辑$_{S}[<, \Sigma, \mathbb{A}]$，使其满足：

$$\otimes(w, \varpi) \models \exists \Delta_1 \ \cdots \exists \Delta_m \ \Big(\text{Run}_{\mathcal{A}}(\Delta_1, \ldots, \Delta_m) \wedge \text{RegOK}(\Delta_1, \ldots, \Delta_m)\Big) \text{ iff } \otimes(w, \varpi) \in \mathcal{L}(\mathcal{A})$$

**剩余目标**。 直观地说，公式 RegOK 需要通过检查词 $\varpi$ 中相关寄存器值出现的位置，来验证所有转移约束是否成立。如果我们可以不受限制地使用全称量词，这件事就很简单。然而，这会导致我们在数据原子公式中使用多个嵌套变量

。因此，我们需要采用一种复杂得多的构造，并利用模态。

ƨ

**由以下内容诱导的活动区间** $\Delta_1, \ldots, \Delta_m$。令 $\mathcal{V}' \coloneqq \{\overset{\triangledown}{y}\} \cup \{\overset{\triangleleft}{r_i}, \overset{\triangleright}{r_i} \mid i \in [k]_{\mathbb{N}}\}$。固定 $\mathcal{V}'$ 上的平局打破顺序 $\prec$。根据 $\Delta$-标签，对于每个 $x \in \mathcal{V}'$，我们用 MSO 定义集合 $\mathcal{I}_x$，其中包含 $x$ 的活动区间：在由 $\Delta$ 描述的（唯一）运行中，$x$ 的值保持不变的极大值域 $I = [a, b]_{\mathbb{N}}$。将 MSO 谓词 $\mathrm{Live}_x(a, b)$ 记作“$[a, b]_{\mathbb{N}} \in \mathcal{I}_x$”。

▷ 断言28（活动区间划分 pos($w$)）。每个 $\mathcal{I}_x$ 都是 pos($w$) 的不相交划分

▷ 断言29（弱猜测见证）。如果 $I \in \mathcal{I}_{r_i^{\triangleleft}}$ 或 $\mathcal{I}_{r_i^{\triangleright}}$ 携带数据值 $\alpha \in \mathbb{A}$，那么存在一个位置 $p \in I$，使得 $\varpi[p] = \alpha$。

根据 $\Delta_i$ 标签定义 $\mathrm{Live}_x(a,\ b)$ 虽然很常规，但记法繁琐，因此省略了具体公式。该公式使用由所选规则是否强制要求 $r_i^{\triangleleft} \neq r_i^{\triangleright}$ 而诱导的、可用单子二阶逻辑定义的“变化点”。

**区间配对**。我们定义类型为 $(x,\ y)$ 的操作区间 这一概念。固定 $x,\ y \in \mathcal{V}'$。我们将 $\mathrm{Pairs}(x,\ y) \subseteq \mathcal{I}_x \times \mathcal{I}_y$ 定义为满足以下条件的配对 $(I_x,\ I_y)$ 的集合：

$$I_x \cap I_y \neq \emptyset, \qquad \min I_x \leq \min I_y, \qquad \max I_x \leq \max I_y.$$

对于 $(I_x, I_y) \in \mathrm{Pairs}(x, y)$ 定义操作区间

$$J(I_x, I_y) \ \coloneqq\ I_x \cup I_y \ =\ [\min I_x, \max I_y]_{\mathbb{N}}.$$

若 $x = y$，则 $\mathrm{Pairs}(x,\ x)$ 简化为 $\{(I_x,\ I_x) \mid I_x \in \mathcal{I}_x\}$，进而 $J(I_x,\ I_x) = I_x$。

▶ **引理30**（由类型 $(x, y)$ 诱导的双射）。假设 $x \neq y$。若 $(I_x, I_y) \in \mathrm{Pairs}(x, y)$，则 $\max I_x \in I_y$ 且 $\min I_y \in I_x$。因此：

*1.* 对于出现在某个配对中的每个 $I_x$，至多存在一个 $I_y$，使得 $(I_x, I_y) \in \mathit{Pairs}(x, y)$；*2.* 对于出现在某个配对中的每个 $I_y$，至多存在一个 $I_x$，使得 $(I_x, I_y) \in \mathit{Pairs}(x, y)$；*3.* 若按 $\mathit{min}\ I_x^i$递增的顺序枚举配对 $(I_x^1, I_y^1), (I_x^2, I_y^2) \ldots$，则 $\mathit{min}\ I_y^1 < \mathit{min}\ I_y^2 < \cdots$，且这些 $I_y^i$ 两两不相交。

证明。设 $(I_x, I_y) \in \mathrm{Pairs}(x, y)$。由于 $I_x \cap I_y \neq \emptyset$，我们有 $\min I_y \leq \max I_x$。结合 $\min I_x \leq \min I_y$，可得 $\min I_y \in I_x$。类似地，$\max I_x \leq \max I_y$ 且 $\min I_y \leq \max I_x$可推出 $\max I_x \in I_y$。

对于 (1)，若 $(I_x, I_y)$,$(I_x, I'_y) \in \mathrm{Pairs}(x, y)$，则 $I_y$ 和 $I'_y$ 都包含 $\max I_x$。由于 $\mathcal{I}_y$ 是一个划分，因此 $I_y = I'_y$。(2) 可通过 $\min I_y \in I_x$对称地证明。对于 (3)，按 $\min I_x^i$递增的顺序枚举。于是 $I_x^i$ 是来自一个划分的互不相交区间，因此 $\max I_x^i < \min I_x^{i+1}$。由于 $\max I_x^i \in I_y^i$，包含 $\max I_x^i$ 的 $y$-区间必须在包含 $\min I_x^{i+1}$的 $y$-区间开始之前结束（同样依据 $\mathcal{I}_y$的划分/不相交性），因此 $I_y^i$ 严格位于 $I_y^{i+1}$之前。◀

▶ **引理31** (交替不相交性)。假设 $x \neq y$。将 配对$(x, y)$ 按$(I_x^1, I_y^1)$、$(I_x^2, I_y^2)$ .....的顺序，按 $\min I_x^i$ 递增排列。令 $J_i \coloneqq J(I_x^i, I_y^i) = I_x^i \cup I_y^i$。于是，对于所有 $i$，都有 $J_i \cap J_{i+2} = \emptyset$。因此，操作区间分成两个由两两不相交区间组成的族：$\{J_1, J_3, \ldots\}$ 和 $\{J_2, J_4, \ldots\}$。

关于操作区间交替不相交性的概念示意图，请参见图 1。

**证明。** 根据引理30，$(I_x^i)_i$ 和 $(I_y^i)_i$ 都是两两不相交的区间序列，并按 $\min I_x^i$ 和 $\min\ I_y^i$递增排列，且 $I_x^i \cap I_y^i \neq \emptyset$。反设 $J_i \cap J_{i+2} \neq \emptyset$。那么 $\min I_x^{i+2} = \min J_{i+2} \leq \max\ J_i = \max\ I_y^i$。由于 $x$-区间按序排列且两两不相交，$\max\ I_x^{i+1} < \min I_x^{i+2}$，因此

$$\max I_x^{i+1} < \min I_x^{i+2} \leq \max I_y^i.$$

由于 $y$区间是有序且不相交的，$\max\ I_y^i < \min I_y^{i+1}$，因此

$$\max I_x^{i+1} < \max I_y^i < \min I_y^{i+1}.$$

因此 $I_x^{i+1}$严格位于 $I_y^{i+1}$的左侧，所以 $I_x^{i+1} \cap I_y^{i+1} = \emptyset$，这与 $(I_x^{i+1},\ I_y^{i+1}) \in$配对$(x,\ y)$ 矛盾。◀

**用于逐次消除两个见证变量。**$\mathrm{MSO}_{\mathbb{S}}[<, \Sigma, \mathbb{A}]$ 条件禁止将见证位置作为全称量化位置 $p$的函数来选择。相反，我们将用互不相交的操作区间覆盖该词，在每个操作区间内选择一次见证，然后进行递归。固定一个递归阶段，并将其划分为$\mathcal{P} \cup \cdot\ \mathcal{Q}\ = \mathcal{V}'$。直观地说，$\mathcal{P}$ 中的变量在当前作用域中已经具有固定的见证位置 $(p_x)_{x\in\mathcal{P}}$，而$\mathcal{Q}$中的变量仍有待见证。

**情形1** （基本情形）。如果 $\mathcal{Q} = \emptyset$，我们可以使用单个嵌套变量在所有位置检查守卫 $p$:

$$\Psi_{\mathcal{V}',\emptyset}(\bar{p})\ :=\ \forall p\ \bigwedge_{t\in\delta}\Big(\Delta_{\kappa(t)}(p) \to \widehat{\Phi}_t(p,\bar{p})\Big),$$

其中，$\Phi_t$ 由寄存器约束 $\Phi_t$ 替换得到：将每个寄存器值变量$x \in \mathcal{V}_{\mathcal{R}}$ 替换为其见证位置 $p_x$，并将 $\overset{\triangledown}{y}$ 替换为 $p$。这样，每个数据原子至多包含一个嵌套变量（即 $p$），因此这是良构的。

**情形2**(归纳步骤)。 假设 $\mathcal{Q}\ \neq \emptyset$。对于每个有序对 $(x,\ y)\ \in \mathcal{Q}\ \times\ \mathcal{Q}$ (包括$x = y$)，我们将:

**1.** 定义与当前阶段相关的操作区间集合 $J(I_x,\ I_y)$；**2.** 使用引理31（或对 $x = y$取平凡情形）将它们分为两个不相交的族（奇数/偶数）；**3.** 量化分割点集$X^0_{x,\ y}$，$X^1_{x,\ y}$，将当前作用域切分为若干子作用域，使每个子作用域恰好包含这些操作区间；**4.** 使用$X^b_{x,\ y}$ 对每个操作区间运行一次子公式，在其中我们为 $x$ 和 $y$选择见证位置，从而在 $x \neq y$时消去两个变量，否则只消去一个变量。

对于 $x, y \in \mathcal{Q}$，我们用 $\mathrm{Op}_{\mathcal{Q}}(x, y; a_x, b_x, a_y, b_y)$ 表示定义集合 $\mathrm{Pairs}(x, y)$ 的单子二阶逻辑可定义谓词:

- $\mathrm{Live}_x(a_x, b_x) \wedge \mathrm{Live}_y(a_y, b_y) \wedge [a_x, b_x]_{\mathbb{N}} \cap [a_y, b_y]_{\mathbb{N}} \neq \emptyset$,$a_x \leq a_y$ 和 $b_x \leq b_y$，并且$[a_x, b_x]_{\mathbb{N}}$
- 是左极端的：对于任何其他 $a_z, b_z$，只要满足 $\mathrm{Live}_z(a_z, b_z)$ 和 $a_y \in [a_z, b_z]_{\mathbb{N}}$，其左端点
- $a_x \leq a_z$ （若有并列，则由 $\prec$打破并列）$[a_y, b_y]_{\mathbb{N}}$是右极端的：对于任何其他 $a_z, b_z$，只要满足 $\mathrm{Live}_z(a_z, b_z)$ 和 $b_x \in [a_z, b_z]_{\mathbb{N}}$，其右端点 $a_z \leq a_y$ （若有并列，则由 $\prec$打破
- 并列）。

同样，这个条件可以用单子二阶逻辑定义，但记号较为繁复，因此我们省略确切的公式。现在我们可以定义$\Psi_{\mathcal{P},\mathcal{Q}}(\bar{p})$，其中 $\bar{p}$ 是来自 $\mathcal{P}$ 的变量元组，如下：

$$\Psi_{\mathcal{P},\mathcal{Q}}(\bar{p}) \coloneqq \bigwedge_{(x,y)\in\mathcal{Q}\times\mathcal{Q}} \bigwedge_{b\in\{0,1\}} \exists X^b_{x,y} \left(\mathrm{Split}^b_{\mathcal{Q}}(x,y,X^b_{x,y}) \wedge \mathcal{C}_{X^b_{x,y}} \Theta_{\mathcal{P},\mathcal{Q}}(x,y)\right),$$

其中：

- $\mathrm{Split}^b_{\mathcal{Q}}(x,\ y,\ X)$ 表示切点 $X$ 满足：所诱导的子作用域包含（可能是严格子集）所有源自 $(x,\ y)$ 的极值区间对、按顺序列出的 $b$-奇偶操作区间 $J(I^i_x,\ I^i_y)$。这由引理31证明（并且由 $\mathcal{I}_x$ 对于 $x=y$ 的不相交性证明）。
- $\Theta_{\mathcal{P},\ \mathcal{Q}}(x,\ y)$ 是子作用域动作：

  $$\Theta_{\mathcal{P},\mathcal{Q}}(x,y;\bar{p}) \coloneqq \left[\exists a_x,b_x,a_y,b_y \ \ \mathrm{Op}_{\mathcal{Q}}(x,y;a_x,b_x,a_y,b_y)\right] \to \Theta'_{\mathcal{P},\mathcal{Q}}(x,y;\bar{p})$$

  其中，对于 $x\neq y$ 公式 $\Theta'_{\mathcal{P},\ \mathcal{Q}}(x,\ y;\ \bar{p})$ 定义为

  $$\Theta'_{\mathcal{P},\mathcal{Q}}(x,y;\bar{p}) \coloneqq \exists p_x\ \exists p_y \Bigg( \underbrace{p_x\in[a_x,b_x]_{\mathbb{N}}}_{\substack{\text{Witness for } x \text{ in its}\\ \text{live range } [a_x,b_x]_{\mathbb{N}}}} \wedge \underbrace{p_y\in[a_y,b_y]_{\mathbb{N}}}_{\substack{\text{Witness for } y \text{ in its}\\ \text{live range } [a_y,b_y]_{\mathbb{N}}}} \wedge \Psi_{\mathcal{P}\cup\{x,y\},\ \mathcal{Q}\setminus\{x,y\}}(\bar{p},p_x,p_y)\Bigg),$$

  并且当 $x=y$ 时，我们定义

  $$\Theta'_{\mathcal{P},\mathcal{Q}}(x,y;\bar{p}) \coloneqq \exists p_x \left(p_x\in[a_x,b_x]_{\mathbb{N}} \wedge \Psi_{\mathcal{P}\cup\{x\},\ \mathcal{Q}\setminus\{x\}}(\bar{p},p_x)\right).$$

最后，我们定义 $\mathrm{RegOK}(\Delta_1,\ \ldots,\ \Delta_m) \coloneqq \Psi_{\emptyset,\ \mathcal{V}'}$。

**正确性（概要）**。

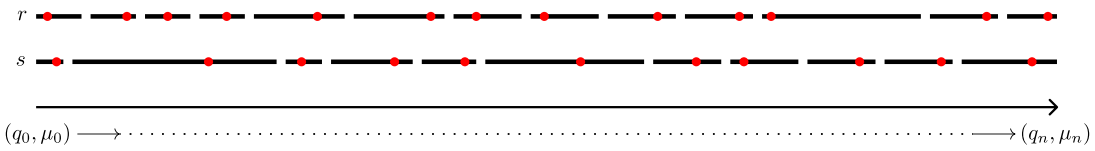
**可靠性**。$(\mathcal{L}(\mathcal{A})\ \subseteq\mathcal{L}(\varphi))$：给定 $\mathcal{A}$ 的一个接受运行，相应地解释 $\Delta_i$，从而使$\mathrm{Run}_{\mathcal{A}}$ 成立。然后，在 产生的每个操作区间中，选取见证位置 $p_x,\ p_y$，这些位置位于相应的活动区间内，并在数据轨道上携带正确的寄存器值（其存在性由弱猜测保证）。基本情形检查每个位置上的守卫，因此公式成立。

**完备性**。$(\mathcal{L}(\varphi)\ \subseteq\mathcal{L}(\mathcal{A}))$：从 $\mathrm{Run}_{\mathcal{A}}$ 在每个位置获得一个规则标记（运行骨架）。递归为 $\mathcal{V}'$ 中的所有变量在每个操作区间内提供见证位置，并且基本情形在各处强制使用翻译后的守卫。利用简单性（不存在混洗 $+$ 显式 $\overset{\triangleleft}{r_i}\sim_i\overset{\triangleright}{r_i}$ $+$ 单射性），可以沿着骨架重构出一致的寄存器赋值，从而得到 $\mathcal{A}$ 的一个接受运行。这证明 $\mathcal{L}(\mathcal{A})$ $=\mathcal{L}(\varphi)$。

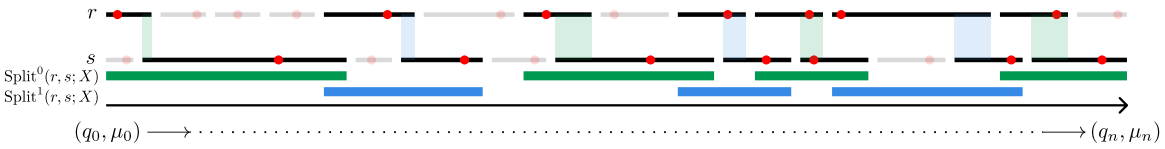
## C 其他证明

**引理 11 的证明概要**。 等式原子的强猜测消去性质$\mathbb{A}_{=}$ 已作为练习 7 的解答予以证明。 [2, 第 10, 207]。我们概述稠密序原子的证明 $\mathbb{A}_{<}=(\mathbb{Q},\ <,\ =)$。

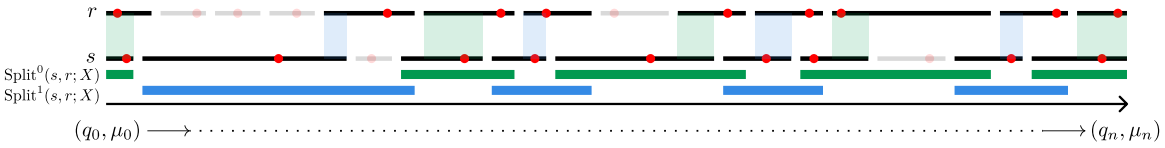
固定一个自动机 $\mathcal{A}\in\mathrm{NRA}^{\mathrm{G}}[\Sigma,\mathbb{A}_{<}]$，其中某个有限 $\Sigma$。考虑一段运行 $\pi=(q_0,\ \mu_0)\xrightarrow{\sigma_1\ \alpha_1},(q_1,\ \mu_1)\xrightarrow{\sigma_2\ \alpha_2},\cdots\xrightarrow{\sigma_n\ \alpha_n},$，它属于 $\mathcal{A}$，且 $\mathcal{A}$ 在 $\pi$ 上进行强猜测。考虑其所有寄存器的所有存活区间组成的集合 $\mathcal{I}$；我们在图 2 中将它们表示为水平线段。区间的 $y$ 坐标对应于寄存器中存储的值。其中一些区间包含一个位置 $i$，使得寄存器值等于数据值 $\alpha_i$。我们称这类区间为弱猜测区间；其余区间为强猜测区间



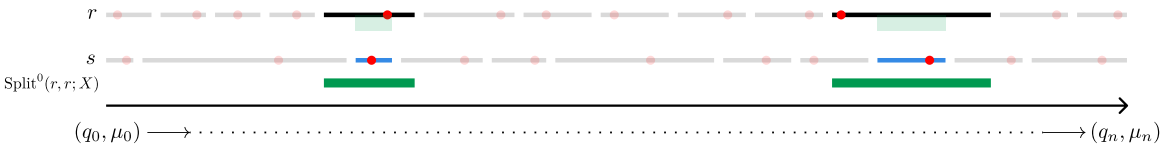

**(a)** A 的一段运行 $\pi$，以及将寄存器的存活区间划分为两类的两种方式 $r$ 和 $s$。



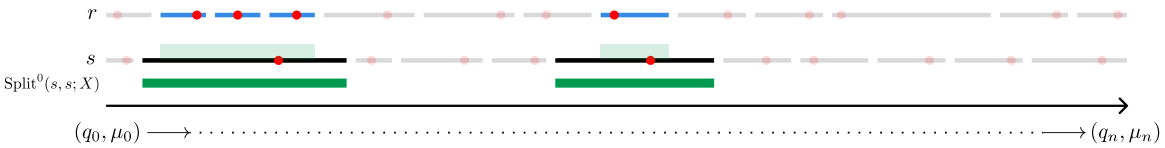

**(b)** 类型（$r$，$s$）的操作区间。直观而言，$r$-区间向左延伸得更远，而 $s$-区间向右延伸得更远。并行使用两个作用域模态，我们可以捕获该类型每个操作区间的见证点（红点）。




**(c)** 类型（$s$，$r$）的操作区间。直观而言，$s$-区间向左延伸得更远，而 $r$-区间向右延伸得更远。并行使用两个作用域模态，我们可以捕获该类型每个操作区间的见证点（红点）。



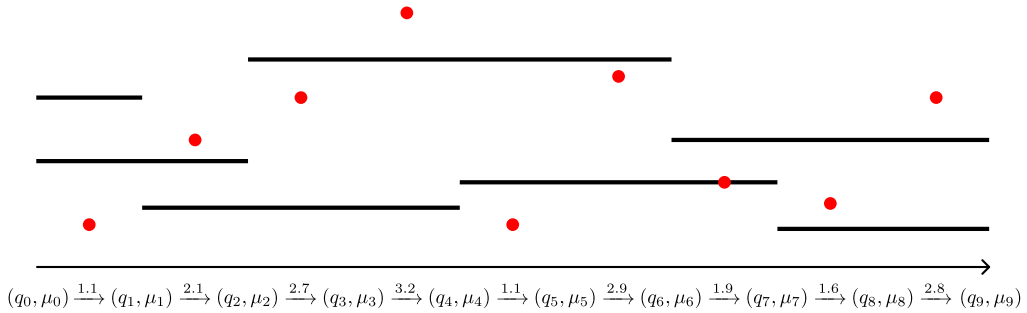

**(d)** 类型（$r$，$r$）的操作区间。这种退化情形只需消除一个寄存器 $r$。



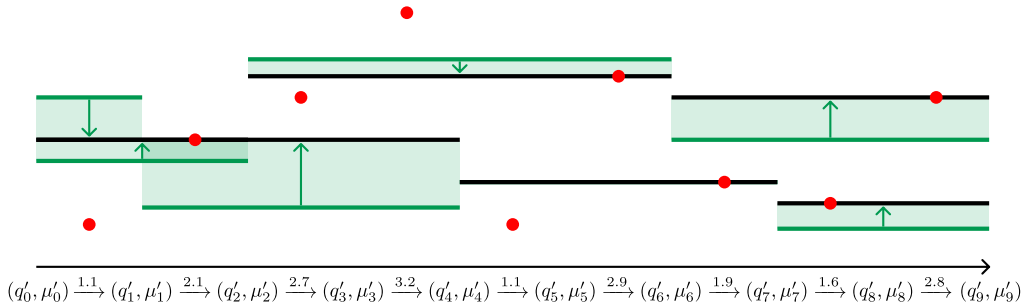

**(e)** 类型（$s$，$s$）的操作区间。这种退化情形只需消除一个寄存器 $s$。

**图 1** 两个寄存器 $\{r, s\}$的四种类型各自的操作区间构造。

猜测。其思路是构造一个自动机 $\mathcal{B}$，放宽与强猜测区间对应的部分转移守卫。具体来说，转移约束 $r_1 < r_2$ 中的某些严格不等式需要放宽为 $r_1 \leq r_2$，使所有强猜测区间都能垂直移动到词中最近的数据值处，从而使猜测变为弱猜测（参见图 2b）。显然，如果$\mathcal{A}$ 有一个接受运行，那么$\mathcal{B}$ 也会有一个。对于$\mathcal{L}(\mathcal{B}) \subseteq \mathcal{L}(\mathcal{A})$，我们需要确保可以进行反向操作，即加强约束。我们通过跟踪存活区间在反方向上可移动距离的上界来实现这一点。这可以通过为原自动机的每个寄存器 $r$ 添加一个新的 $r'$ 来实现。◀




**（a）运行** $\pi$ 中的活动区间，属于$\mathcal{A}$。




**（b）运行** $\pi'$ 中的活动区间，属于$\mathcal{B}$。

**图 2** 执行强猜测的$\mathcal{A}$ 的运行 $\pi$ 与仅进行弱猜测的$\mathcal{B}$ 的运行 $\pi'$ 之间的对应关系。