分层自动机：无穷词上自动机的典范模型

Antonio Casares 泽斯劳滕-
兰道大学，德国**Christof Löding**亚琛
工业大学，德国**Igor**
Walukiewicz法国国家科学研究中心、
波尔多大学，法国

arXiv:2601.15940v1 [cs.FL] 22 Jan 2026

摘要

我们引入分层自动机这一交替奇偶自动机的子类，它是确定性自动机的推广。假设满足一致性性质，这些自动机便具有历史确定性和 0-1 概率性。我们证明，每个 ω-正则语言都由唯一的最小一致分层自动机识别，并且可以在多项式时间内从任意分层自动机或确定性自动机计算出这种典范形式。我们还证明，对于分层自动机，一致性检查和包含性测试都可以在多项式时间内完成。与确定性有限自动机十分相似，最小一致分层自动机可以通过同余关系来刻画。

2012 ACM 学科分类 **分类** 计算理论 → 逻辑与 验证

关键词和短语 ω 自动机、历史确定性、规范自动机

资助 *Antonio Casares*：部分研究得到德国科学基金会（资助编号 522843867）和欧洲研究委员会（资助编号 101089343）的支持。本研究的部分工作是在 Casares 任职于波兰华沙大学期间完成的，并获得波兰国家科学中心（NCN）“多项式有限状态计算”项目资助（2022/46/A/ST6/00072）。

致谢 感谢 Thomas Colcombet、Aditya Prakash 和 Pierre Ohlmann 对这一主题进行了有益的讨论。

本文档包含超链接。每次出现概念时，都会链接到其定义。在电子设备上，读者可以点击单词或符号（某些 PDF 阅读器中只需将鼠标悬停在其上），以查看其定义。

2 分层自动机：无穷词上自动机的一个典范模型

目录

1 引言 3 2 预备知识 6

3 分层自动机 8

3.1 分层自动机的句法定义.8 3.2 通过交替奇偶自动机定义语义.9 3.2.1 定义.10 3.2.2 示例.11 3.2.3 从交替奇偶自动机到分层自动机.12 3.3 通过安全语言定义语义.13 3.4 分层自动机具有历史确定性和 0-1 概率性 15 3.5 多项式时间内的若干判定过程.18 3.6 与其他模型的联系和比较.19

4 规范的最小分层自动机 23

4.1 正规性、中心性与安全最小性.23 4.2 中心序列.24 4.3 极小分层自动机的唯一性.25

5 多项式时间内的最小化 26

5.1 分层自动机的态射.27 5.2 规范化.28 5.2.1 分离强连通分量.28 5.2.2 降低.29 5.3 安全最小化.31 5.4 中心化.33 5.5 子自动机与满射态射.34

6 基于同余的刻画 3 **5**

6.1 同余关系的定义和基本性质.35 6.2 自动机 $\mathcal{A}_{/\equiv_L}$ .41 6.3 $\mathcal{A}_{/\equiv_L}$ 的正确性.42 6.4 $\mathcal{A}_{/\equiv_L}$ 的最小性.45

7 结论 **48**

1 引言

无穷词上的自动机（下文称为 ω-自动机）最早由 Büchi 引入，用以证明关于 $(\mathbb{N}, <)$ 的单子二阶逻辑的可判定性 [Büc62]。此后，它们在反应式系统的验证与综合中得到了广泛应用 [CES86, PR89, Kup18]。

当前 ω-自动机理论的一个关键局限，是缺少最小的典范模型。最小化对于实际应用至关重要，特别是在通过分层转换从规约（例如线性时序逻辑公式）导出自动机时。在这一过程中，规约中的每个构造子都对应自动机上的一种操作，其中一些操作需要进行幂集构造。有限词和树的情形表明，每次操作后进行最小化是实现高效转换的关键 [KMS02]。最小化对于理论研究也很重要。一个直接的例子是针对 ω-正则语言的学习算法；更广泛地说，它表明我们已经对接受器的结构性质有了深入认识。这些认识最终或许能让我们重新审视一些长期悬而未决的问题，例如树自动机的指标问题 [CL08, IL25]。

与有限词上的自动机不同，确定性 ω-自动机没有最小的典范模型，因为一种语言可能存在多个非同构的确定性自动机，且它们的状态数都达到最小。此外，对于表达能力足以识别所有 ω- ω正则语言 [Sch10, Cas22, AE25]的各种确定性 ω-正则语言的典范表示方法有一些，例如 ω-半群或同余关系，但这些表示可能比确定性奇偶自动机大得多，甚至可能大出指数级；此外，它们无法直接用于验证和综合领域的应用。

历史确定性 (HD) 自动机是非确定性自动机的一个子类，其非确定性可以即时消解。它们最初以 游戏适用 [HP06][1], 之名引入，因为这类自动机恰好能够与任意博弈组合，同时保持胜者不变。这一性质使它们能够在验证和综合应用中取代确定性自动机。每个 确定性自动机都是历史确定性的，但一个历史确定性自动机可能比最小的确定性自动机小一个指数级 [KS15]。 历史确定性已推广至交替 ω自动机 [Col13, BL19]。历史确定性交替自动机具有同样良好的组合性质，因此适用于验证和综合应用，而且规模甚至小于非确定性 HD 自动机 [BKLS20，引理 6]。

近期受到广泛研究的其他类型自动机包括：其非确定性可以通过随机方式消解的自动机 [AK15, HPT25, PPS+25]，以及能够与马尔可夫决策过程忠实组合的自动机 [HPS+20]。特别是，*0-1 概率性自动机*是指在这类自动机中，无限词上的随机游走以概率 1 接受，当且仅当该词属于自动机的语义；否则，它以概率1拒绝。这类自动机尤其适合用于研究马尔可夫决策过程 [HPT25]。

在这项工作中，我们提出了一种新的 ω-正则语言接受器模型，并将其称为分层自动机。这一模型的出发点是 Abu Radi 和 Kupferman 的结果 [AK22] ，该结果指出，（基于转移的）历史确定性 coBüchi 自动机可以在多项式时间内最小化，并且具有规范最小表示。在[LW25] 中，研究进一步表明，这个规范自动机可以用一种

[1] 树良好自动机 这一相关概念此前已在 [KSV96]中提出。

自然的方式来描述，即使用有限词对上的同余。然而，历史确定性 coBüchi 自动机只能识别 ω-正则语言中的一个片段：即属于 Σ^0_2 博雷尔层级中的该层级，或者等价地说，属于 Manna-Pnueli 层级中的持久性质 [MP90]。

我们提出的分层自动机模型可视为 Abu Radi 和 Kupferman 的最小化结果在所有 ω-正则语言上的扩展。这与一些近期研究一脉相承，例如*coBüchi* 自动机链 [ES22]和重轨自动机 [Ehl25]，我们将在第 3.6 节中与之比较。另一方面，分层自动机也可视为 Muller 语言的 Zielonka 树 [Zie98] 以及位置语言的签名自动机 [CO24]的一种推广。我们提供了充分证据，表明分层自动机具有良好的结构和算法性质。

贡献

我们引入了分层自动机，这是一种表示所有 ω-正则语言的形式体系。可以将分层自动机视为一种具有良好结构性质的交替奇偶自动机表示。如果分层自动机满足一项额外的一致性性质，那么与之关联的交替自动机就是历史确定的。

一致分层自动机结合了两种结构：最小的历史确定性 coBüchi 自动机（命题 3.9）和 Zielonka 树。此外，它们还涵盖了确定性奇偶自动机（命题 3.8）。它们具有非常优良的性质。最重要的是，它们存在一种典范的最小形式，且可在多项式时间内计算得出。

这一典范性结果可表述如下：

▶**定理 1.1**。 每个 ω正则语言 L 都可以由一个一致分层自 动机 $\mathcal{A}_L$ 识别，使得每个与之等价的一致分层自动机都包含一个子自动机，该子自动机存在一个满射态射映射$\mathcal{A}_L$。此外，$\mathcal{A}_L$ 可从任意给定的一致分层自动机在多项式时间内计算得出，该自动机识别 L。

我们将进一步详细介绍分层自动机的性质。

适用性。一致分层自动机是交替奇偶自动机的表示形式，并且保证同时具有历史确定性和 0-1 概率性（命题 3.19 和 3.20）。因此，一致分层自动机具备关键性质，适用于验证和综合应用。此外，它们还提供了一种易于检验的充分条件，以确保其具有历史确定性且为 0-1 概率性，因为分层自动机以语法方式定义，而检验其一致性可在多项式时间内完成（命题 3.26）。

由于分层自动机是最小 coBüchi 自动机的推广，其规模可能比最小的确定性奇偶自动机 [KS15]。

我们还证明，一致分层自动机的空性与包含问题一致分层自动机可以在多项式时间内解决（命题 3.27 和 3.28）。

典范性、态射与同余关系。我们证明，每个 ω-正则语言都可以由唯一的最小一致分层自动机识别。更确切地说，我们证明，对于每个 ω-正则语言 L，都存在一个一致分层自动机 A_L，它是正规、中心化且安全最小的；任何具有这三种性质的分层自动机都与 A_L 同构（定理 4.1）。此外，每个识别L 的一致分层自动机都包含一个子自动机，该子自动机存在一个到 A_L 的满射态射（推论 5.2）。此外，我们还给出了基于同余关系的最小分层自动机 A_L 的刻画（定理 6.14）。这一刻画完全基于语言 L，并且

不依赖于给定的自动机表示形式。据我们所知，这是此类自动机模型的首批成果；该模型能够识别所有ω-正则语言，且其最小形式的规模从不超过确定性奇偶自动机。

多项式时间内的最小化。我们证明，给定一个一致分层自动机$\mathcal{A}$，我们可以在多项式时间内构造出其最小分层自动机，即 $L(\mathcal{A})$ （定理 5.1）。由于每个确定性奇偶自动机显然都是一个一致分层自动机，因此这提供了一种高效的最小化过程，可将确定性奇偶自动机转换为分层自动机。这并不与前文提及的关于最小化的下界矛盾，因为该过程的结果是一个交替自动机。

与其他模型的关系。我们通过示例和结果，比较并阐释分层自动机与其他模型（例如 coBüchi 自动机链和 Zielonka 树）之间的关系。我们还讨论了如何利用分层自动机刻画 ω-正则语言的一些重要性质，例如位置性；这方面可参见 [CO24]。虽然我们将对这些关系的更详细研究留待未来开展，但我们的讨论表明，分层自动机与 ω-正则语言理论中各种成熟且较新的形式体系有许多有趣的联系。

相关工作

我们工作的出发点是 Abu Radi 和 Kupferman提出的 [AK22] 的 历史确定性 coBüchi 自动机最小化过程。通过对偶性，这也导出了全称历史确定性 Büchi自动机的最小化过程，而分层自动机 这一概念可以看作对任意奇偶指标的一种自然扩展，它混合了非确定性分支和全称分支。 分层自动机 还受到了签名自动机以及 ε-完备自动机的启发，这些自动机见于 [CO24，第 3 节。3]，以及针对它们的最小化过程。关于所有分层自动机 都是历史确定性的 证明，推广了 [KS15，引理 3]中的思想。基于同余的刻画扩展了 coBüchi 自动机的同余关系，该结果见于 [LW25，第 3], 节；其所使用的条件与 [BL24，定义 3.1、3.5]中针对有限字定义的优先级映射密切相关。

文献中存在 ω-正则语言的各种表示，它们在某种意义上具有典范模型。这些表示包括一些基于同余关系的代数方法，例如 Arnold 同余 [Arn85], Wilke 代数和 ω-半群 [Wil91，PP95], 中的右同余族 [MS97，定义 5] 以及确定性有限自动机族 [Kla94，AF16]。这些对象的缺点是，它们的规模可能比确定性奇偶自动机大一个指数级差距（众所周知，即使对于有限字上的正则语言，句法幺半群的大小也可以是 n^n ，而最小确定性有限自动机的大小为 n，参见 [HK02，定理 3] 中的具体例子；这一指数差距也会传递到上述所有形式体系中）。虽然它们的大小与某个语言的最小 DPA大小之间不存在可比性，但它们不能直接用于综合等应用。

另一种表示形式以特定的确定性奇偶自动机（DPA）的形式给出，见 [BL24，第 3]节。文中将语言的“精确 DPA”定义为由确定性米利机的乘积构成，这些米利机根据着色函数直接从语言中提取。然而，精确 DPA 的规模可能比最小的DPA大一个指数级。

近来，一些基于历史确定性 自动机的规范表示已被提出。2022 年，Ehlers 和 Schewe 提议将一个 ω-正则语言表示为

余 Büchi 可识别语言的递减链 [ES22]。其中每种语言都可以由 Abu Radi 和 Kupferman 提出的最小历史确定性 co-Büchi 自动机表示，从而得到一种规范表示。此后，这种形式体系以$COCOA$ [EK24a, EK24b]之名得到了研究。最小分层自动机显然与 COCOA 有关，但二者存在一些重要区别。一个关键区别是，分层自动机与交替历史确定性自动机直接相关——这种自动机被广泛认可为适用于验证和综合的 ω-自动机。值得注意的是，我们猜想规范的分层自动机在所有历史确定性交替奇偶自动机中都是最小的（猜想 7.1）。两种模型的大小比较略显微妙。尽管 COCOA 可能比分层自动机更小，但若要从 COCOA 得到一个历史确定性 自动机，就需要对其所有分量取乘积 [EK24a]。我们在第 3.6 节中对 COCOA 作了更详细的比较。

2025 年，Ehlers 引入了 重轨自动机 [Ehl25，定义 1]，并证明可以高效地对其进行最小化。这类自动机为每个运行关联一个优先级（即无限次出现的最小优先级）；当某个运行的最大优先级为偶数时，词即被接受。此外，它们必须满足一种语义上的“再轨道性质”，其灵感来自最小 历史确定性 coBüchi 自动机的性质。在 [Ehl25，第 VI 节], 中，研究表明重轨自动机可用于反应式综合，但其与 历史确定性 自动机之间的确切关系尚未确立。

2 预备知识

令 $[d] = \{1, \ldots, d\}$，并称其中的元素为 $[d]$ 优先级。我们用变量 x, y表示优先级。该奇偶条件 是定义在 $[d]^\omega$ 上的语言，定义如下：

奇偶 $= \{x_1 x_2 \cdots \in [d]^\omega \mid \liminf x_i$ 为偶数$\}$。（注意，本文使用最小奇偶条件。）

迁移系统。设 Σ 为一个集合。定义在其上的迁移系统是一个边标记有向图 $\mathrm{T} = (Q, \Delta)$，其中 $\Delta \subseteq Q \times \Sigma \times Q$。如果 $(q, a, q') \in \Delta$，我们也写作 $q \xrightarrow{a} q'$。如果对所有 $q \in Q$ 和 $a \in \Sigma$，都存在某个 $(q, a, q') \in \Delta$，则称其为完全的。如果对每个 $q \in Q$ 和 $a \in \Sigma$，至多存在一个 q' 使得 $(q, a, q') \in \Delta$，则称其为确定性的。我们更倾向于用函数 $\delta\colon Q \times \Sigma \to Q \cup \{\bot\}$ 表示确定性迁移系统的转移，其中若 $(q, a, q') \in \Delta$，则 $\delta(q, a) = q'$；若不存在这样的转移，则 $\delta(q, a) = \bot$。

对于一个词 $w \in \Sigma^*$，我们写作 $q \xrightarrow{w} q'$，如果存在一条从 q 到 q' 的路径，其标签为 w，我们称之为一次运行。有时我们省略 q'，只写作 $q \xrightarrow{w}$，如果 q' 无关紧要。若不存在从 $q \xrightarrow{w} \bot$ 出发且标签为 w 的路径，我们写作 q。

若 T 是定义在乘积字母表 $\Sigma \times [d]$ 上的迁移系统，略去记号上的区别，我们称 $w \in \Sigma^\omega$ 上的任意路径为运行，只要它在 Σ 分量中的标签为 w。在这种情况下，如果运行的 $[d]$ 标签序列满足奇偶条件，我们也称该运行为接受。我们记 $q \xrightarrow{a} q'$，如果 $(q, (a, x)\ q') \in \mathrm{T}$；对于某个 $y \geq x$，若 $(q, (a, y)\ q') \in \mathrm{T}$，则类似地记作 $q \xrightarrow{a} q'$；对于 $> x$ 等也类似。若路径上的最小优先级为 $\geq x$，我们记 $q \xrightarrow{w} q'$；对于$> x$ 等也类似。

两个定义在相同集合 Σ 上的迁移系统 $\mathrm{T} = (Q, \Delta)$、$\mathrm{T}' = (Q', \Delta')$ 之间的态射，是一个函数 $\mu\colon Q \to Q'$，满足：若 $(p, a, q) \in \Delta$，则 $(\mu(p)\ a, \mu(q)) \in \Delta'$。注意，在 T 中从 q 出发、定义在 w 上的任意运行，都会映射为 T′ 中从 $\mu(q)$ 出发、定义在 w 上的运行。

确定性奇偶自动机。 一个 确定性奇偶自动机是一个完备的确定性迁移系统 $\mathcal{A} = (Q, \Delta)$ 定义在集合 $\Sigma \times [d]$ 上，并带有指定的初始状态

$q_{\mathsf{init}} \in Q$。

如果一个词 $w \in \Sigma^\omega$ 在 w 上的唯一运行满足 奇偶条件，则称该词被自动机接受。自动机的语言是其所接受的词的集合。如果一个语言可以由 ω-正则 确定性奇偶自动机识别，则称其为 。 ω-正则语言有许多等价的定义方式，例如通过在 MSO逻辑中的可定义性，或通过非确定性Büchi自动机进行识别；例如参见 [GTW02]。

交替奇偶自动机。 一般而言， 交替奇偶自动机 由一组状态以及一个形式如下的转移函数给出： δ: $Q \times \Sigma \to \mathsf{Bool}^+(Q \times [d])$。在本文中，所有交替自动机都采用一种更简单的形式，我们称之为按优先级简化的交替自动机。这些自动机的所有转移都属于以下两种形式之一：

$$\delta(q,a) = \bigwedge_{p \in Q'} (p,x) \text{ for some even } x, \quad \text{or} \quad \delta(q,a) = \bigvee_{p \in Q'} (p,x) \text{ for some odd } x.$$

一般情形的定义可参见 [BL19] 。下面我们给出 按优先级简化的交替自动机这一子类的一种形式化的替代性表述。

简单优先级自动机是定义在集合$\Sigma \times [d]$ 上的完全转移系统 $\mathsf{A} = (Q, \Delta)$，具有指定的初始状态 $q_{\mathsf{init}} \in Q$，并满足：对于每个 $q \in Q$ 和 $a \in \Sigma$，从 q 出发的所有a-转移具有相同的优先级。因此， q中存在一个动作 a 的优先级，这意味着对于所有 $(q,(a,y)\ q') \in \Delta$，都有 $y = x$。

简单优先级自动机的语义通过博弈来定义，其中动作的优先级决定由谁行动。给定一个输入词 $w = a_1 a_2 \cdots \in \Sigma^\omega$，我们定义博弈$\mathcal{G}(\mathcal{A}\ w)$，其中两位参与者 Eve 和 Adam 构造一个运行，沿着词 w 在$\mathcal{A}$中进行，具体如下：

- 初始位置是状态 $q_1 = q_{\mathsf{init}}$。假设在一局对局中构造的当前运行以状态 q_i结束，且 x_i 是
- a_i 在 q_i中的优先级。如果 x_i 是奇数，则由 Eve 选择一条转移 $q_i \xrightarrow{a_i:\ x_i} q_{i+1}$；否则由 Adam 作出选择。如果构造出的路径是接受的，则 Eve 获胜。

我们称 w 被 A 接受，当且仅当 Eve 在博弈 G(A w) 中拥有获胜策略。注意，根据奇偶博弈的确定性， [相反，Adam 拥有获胜策略。A 所识别的语言，记作 L(A)，是它所接受的词的集合。状态 $q \in Q$ 的语言，是将 q 固定为初始状态后得到的自动机所识别的语言。我们将其记作 L(A q)；如果上下文明确，也简写为 $L(q)$。

语义确定性。 自动机 $\mathcal{A}$ 是 语义确定的，如果只要有 $p \xrightarrow{a} q_1$ 和 $p \xrightarrow{a} q_2$ ，那么 L（$\mathcal{A}\ q_1$） $= L$（$\mathcal{A}\ q_2$）。等价地，如果 $p \xrightarrow{a} q$ 那么L（$\mathcal{A}\ q$） $= a^{-1}L$（$\mathcal{A}\ p$）。

▶ 注 2.1。 设 $\mathcal{A}$ 是一个 语义确定的 自动机 ， 它识别 语言 L。设 u 是一个从初始状态出发、为路径标记的词，且该路径通向 q。那么， 语言 q 为：

$$u^{-1}L = \{w \in \Sigma^\omega \mid uw \in L\}.$$

解析器与历史确定性。考虑一个简单优先级自动机 A。A 中 Eve 的一个解析器是函数 σ: $(\{q_{\mathsf{init}}\} \cup \Delta^+) \times \Sigma \to Q$，满足：每当一条运行$\rho \in (\{q_{\mathsf{init}}\} \cup \Delta^+)$ 以状态 q 结束，且 a 在 q 中的优先级为奇数时，那么 $q' = \sigma(\rho, a)$ 是 q 的一个a-后继状态，即 $q \xrightarrow{a} q'$。如果一条运行的每个前缀

$\rho \cdot (q, \ (a, \ x) \ q')$ 与 x 的优先级为奇数，则有 $\sigma(\rho, \ a) = q'$。由 Eve 解析器接受的语言 σ 是

$$L(\mathcal{A}, \sigma) = \{w \in \Sigma^\omega \mid \text{every } \sigma\text{-run over } w \text{ is accepting}\}.$$

注意，我们总有 $L(\mathrm{A}\ \sigma) \subseteq L(\mathrm{A})$。如果 Eve解析器 σ 满足 $L(\mathrm{A}\ \sigma) = L(\mathrm{A})$，我们就称其为有效解析器。对 Adam 及其语言的解析器，我们也作对称定义（注意，若 τ 是 Adam解析器，则 $L(\mathrm{A}\ \tau) \supseteq L(\mathrm{A})$）。对于分别为 Eve 和 Adam 固定的两个解析器 σ 和 τ，每个词都有唯一一个同时与 σ 和 τ 一致的运行。我们用 $L(\mathrm{A}\ \sigma, \tau)$ 表示这些运行为接受运行的词的集合。

一个交替自动机是历史确定的(HD)，如果 Eve 和 Adam 都有有效解析器。也就是说，存在解析器 σ 和 τ 分别属于 Eve 和 Adam，使得

$$L(\mathcal{A}) = L(\mathcal{A}, \sigma) = L(\mathcal{A}, \tau) = L(\mathcal{A}, \sigma, \tau).$$

3 分层自动机

本节将介绍分层自动机，这是本文的中心概念。我们分两个阶段给出其定义：分层自动机的句法结构（第 3.1 节），以及它们通过关联的分层自动机（第 3.2 节）所识别的语言。得益于交替奇偶自动机的结构，关联的自动机保证分层自动机具有简单的优先级结构。

接下来，我们引入一致语义确定性，这是一个略强于语义确定性的概念。语义确定性是自动机成为历史确定性自动机的必要条件。对于分层自动机而言，一致语义确定性带来两个重要结论：相应的交替自动机同时具有历史确定性和0-1 概率性（命题 3.19 和 3.20）。

我们在第 3.3 节介绍的主要技术工具，是对分层自动机的接受进行刻画，且不依赖交替。为此，我们引入一致性这一概念，用于刻画一致语义确定性。对于分层自动机，我们可以在多项式时间内检验一致性（命题 3.26）。此外，对于一致分层自动机，空性和语言包含可以在多项式时间内判定（命题 3.27 和 3.28）。

3.1 分层自动机的句法定义

一个分层自动机 $\mathcal{A}$ 是由以下内容组成的元组：若干个确定性的迁移系统 $\mathcal{T}_1$，……，$\mathcal{T}_d$，以及态射 $\mu_x : \mathcal{T}_{x+1} \to \mathcal{T}_x$，其中 $1 \leq x < d$。此外，我们要求 $\mathcal{T}_1$ 是完备的，具有一个特殊的初始状态 q_{init}，位于 $\mathcal{T}_1$，中，并且 $\mathcal{T}_1$ 中的所有状态都可从 q_{init} 到达。[2]

我们称 T_x 为自动机的第 x 层。我们使用 Q_x 表示 T_x 的状态集合，并将其称为 x-状态。我们假设 Q_1、……、Q_d 两两不相交。我们令 $\delta_x : Q_x \times \Sigma \to Q_x \cup \{\bot\}$ 为 T_x 的（部分）转移函数（对于 T_1，该函数必须是全函数）。我们写作 $p \xrightarrow{a}_x q$，以强调转移发生在自动机的第 x 层。有时我们用 Q_{x+1} 表示 x 某一层；当 $x = d$ 时，我们假设该集合为空集。

我们将 $Q_{\geq x}$ 记作 Q_x、……、Q_d 的并集，因此 $Q_{\geq 1}$ 是所有状态集合的并集。

[2] 从索引 1 开始（而不是例如从 0开始）是任意的。之所以这样选择，是为了使具有两层的分层自动机对应于广为人知的 HD coBüchi 自动机。

对于每个 x，我们定义 $\widehat{\mu}_x$： $Q_{\geq x} \to Q_x$ 如下

$$\widehat{\mu}_x(q) = \mu_x(\mu_{x+1}(\dots\mu_{y-1}(q)\dots)), \text{ for } q \in Q_y,\ y > x \ ; \quad \text{and} \quad \widehat{\mu}_x(q) = q, \text{ for } q \in Q_x.$$

对于状态 q，令 layer(q) 为满足 $q \in Q_x$ 的层级 x。对于状态与字母构成的对 (q, a)，令 layer(q, a) 为满足 $\delta_x(\widehat{\mu}_x(q)\ a)$ 有定义的最大层级 x。注意，由于 T_1 是完备的，layer(q, a) 总是有定义的。

若 $\widehat{\mu}_x(q) = p$，对于 $x = \text{layer}(p)$，则称 p 是 q 的祖先。给定 $q, p \in Q_{\geq x}$，若 $\widehat{\mu}_x(q) = \widehat{\mu}_x(p)$，则记作 $q \sim_x p$，并令 $[q]_x = \widehat{\mu}_x^{-1}(\widehat{\mu}_x(q))$。若状态 $q \in Q_{\geq 1}$ 不是任何 μ_x的像，则称其为叶状态。

分层自动机的森林。我们可以将一个分层自动机看作一个森林，其节点为 $Q_{\geq 1}$。T$\mathcal{T}_1$ 中的每个状态都是一棵树的根节点，而节点 $q \in Q_{x+1}$ 以 μ_x（q） $\in Q_x$ 为父节点。该森林的层级从1 （最顶层）到 d标记。

这样，层(q)就是该森林中 q 的层级。函数 $\widehat{\mu}_x(q)$ 指定了节点 在层级 x 上的祖先 $q \in Q_{\geq x}$。我们有：如果 p 和 q 在层级 x上具有相同的祖先，那么 $q \sim_x p$。等价地，$[q]_x$ 是以 祖先为根的子树中节点的集合，其中 q 位于层级 x。正如预期，叶状态就是在该森林中作为叶节点的状态。请注意，多个层级可能都具有叶状态。

▶ **例 3.1**。 在图 1 中，我们展示了两个分层自动机，它们表示为树。每个迁移系统构成树的一个层级。对应于父关系的态射 μ_x 用虚线箭头表示。

在这两个例子中，所有叶状态都位于同一层。这并非必须如此。图 8（第 6 节，例 6.1）展示了一个分层自动机，其叶状态位于不同的层。

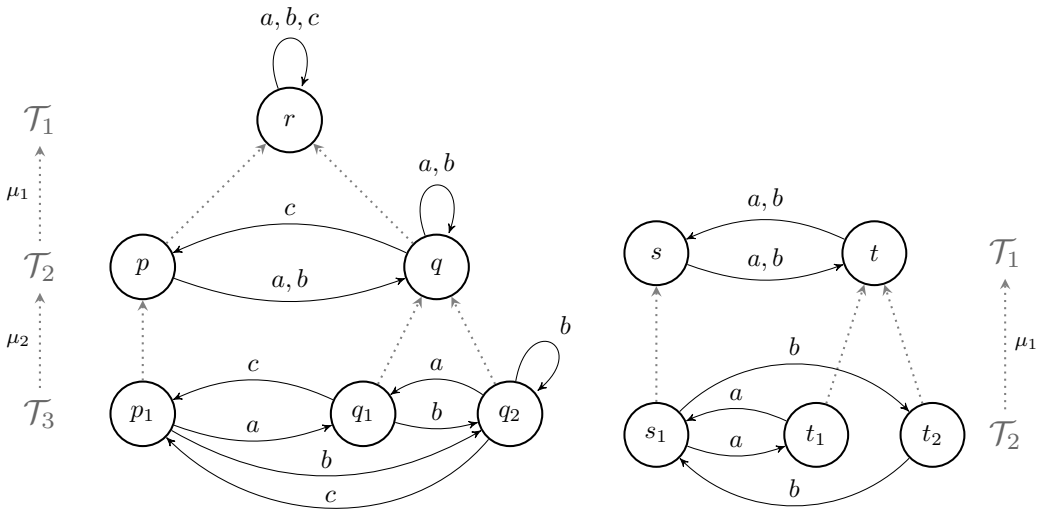


图 1 两个 分层自动机，以树的形式表示。

3.2 通过交替奇偶自动机定义的语义

现在，我们定义分层自动机的接受方式：为每个分层自动机$\mathcal{A}$关联一个交替自动机 $[\![\mathcal{A}]\!]$, ，称为A 的语义$\mathcal{A}$；$\mathcal{A}$ 的语言定义为

语言 $[\![\mathcal{A}]\!]$。 $\mathcal{A}$ 的树结构约束 $[\![\mathcal{A}]\!]$ 的转移按优先级保持简单。基于这一语义，我们引入一致语义确定性，这是对语义确定性的略微加强。该性质确保，所有叶状态在第1层中具有相同祖先的叶状态都接受相同的语言——我们稍后将通过一致性（第 3.3 节）来刻画这一约束。

3.2.1 定义

我们为一个分层自动机$\mathcal{A}$ 关联其语义自动机 $[\![\mathcal{A}]\!]$, ，它是一个由优先级定义的简单交替奇偶自动机，定义如下。状态 Q 是叶状态，属于$\mathcal{A}$。其初始状态是任意一个叶状态 q ，满足 $\widehat{\mu}_1(q) = q_{\text{init}}$ 是初始状态，属于$\mathcal{T}_1$。[3]

转移为

$$\Delta = \left\{ (q, a, x, q') \in Q \times \Sigma \times [d] \times Q \;\middle|\; \begin{array}{l} x = \mathit{layer}(q, a), \text{ and} \\ \widehat{\mu}_x(q) \xrightarrow{a} \widehat{\mu}_x(q') \text{ in } \mathcal{T}_x \end{array} \right\}.$$

直观地说，当处于叶节点 q 并给定一个字母 a 时，我们考虑层$(q,\ a)\ = x$，即满足下列条件的最大 x ： 转移 $\widehat{\mu}_x(q) \xrightarrow{a}_x p$ 有定义。从 q 到每个叶节点 $p' \in \widehat{\mu}_x^{-1}(p)$都有一条转移。所有这些转移的优先级都是 x。如果 x 是奇数，则由 Eve 选择其下的一个叶节点 p；否则由 Adam 选择。

▶ 注 3.2。每当存在一条转移 $q \xrightarrow{a:x} q'$ 在 $[\![\mathcal{A}]\!]$, 时，也存在一条从 q 出发、指向每个叶状态 $q'' \in [q']_x$的转移，且不存在指向其他状态的转移。此性质说明了记号 $q \xrightarrow{a:x} [q']_x$的合理性。

在下文中，我们几乎专门关注分层自动机，使得 $[\![\mathcal{A}]\!]$是语义确定的。实际上，我们研究的是一个稍强的性质。

▶ **定义 3.3**。 分层自动机$\mathcal{A}$ 是一致语义确定的，如果对于每一对叶状态 $p \sim_1 q$ ，都有 L（$[\![\mathcal{A}]\!]$，p）$= L$（$[\![\mathcal{A}]\!]$，q）。

顾名思义，我们有如下结论。

▶**引理 3.4**。 如果一个分层自动机$\mathcal{A}$ 是一致语义确定的，那么它是语义确定的。

证明。 如果 $q \xrightarrow{a} p_1$ 且 $q \xrightarrow{a} p_2$ 在 $[\![\mathcal{A}]\!]$, 中，那么 $\widehat{\mu}_1$（p_1）$= \widehat{\mu}_1$（p_2），因此结论成立。 ◀

在命题 3.26 中，我们证明了可以在多项式时间内检查给定的分层自动机是否一致语义确定。

▶ 注 3.5。 如果 $\mathcal{A}$ 是 一致语义确定的 ，那么 $\mathcal{T}_1$ 存在一个态射 映射到 剩余自动机 $L = L$ $([\![\mathcal{A}]\!])$：该 迁移系统 的状态为集合 $u^{-1}L$，转移为 $u^{-1}L \xrightarrow{a} (ua)^{-1}L$。这是因为我们假设 $\mathcal{T}_1$中的所有状态都可以从 q_{init}到达。

[3] 形式上，这定义了一个自动机族。一般而言，这些自动机不一定等价。然而，对于本文所研究的自动机（参见下文关于一致性均匀 SD一致性的概念），初始状态的选择无关紧要。

3.2.2 示例

▶ **例 3.6**。 图 2 展示了与图 1 左侧的 分层自动机 相对应的交替奇偶自动机。它识别的语言为

$L =$含有限多个 cc 和无限多个 aa.的词

状态是树的叶子。每当从状态 c 读到字母 p_1时，就产生优先级1，Eve 可以选择转移到任意状态。每当从状态 a 读到字母q_1时，就产生优先级2，Adam 可以选择转移到 q_1 或 q_2。其余转移产生优先级3，对应于3层上的分层自动机中的转移。

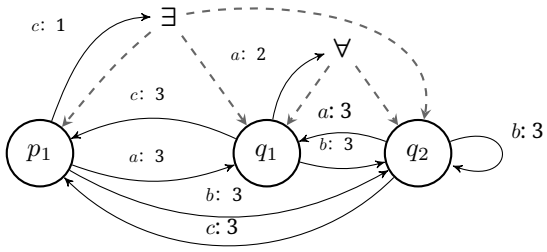


图 2 图 1 左侧分层自动机对应的交替奇偶自动机。

▶ **例 3.7**。 图 3 展示了分层自动机对应的交替奇偶自动机，该自动机对应于图 1 右侧的分层自动机。它识别语言

$$L = (\Sigma^2)^*(aa + bb)^\omega.$$

在此情况下，该自动机最终是确定性的。迁移系统$\mathcal{T}_1$ 是该语言的剩余自动机，用于计算词中位置的奇偶性（偶数或奇数）。

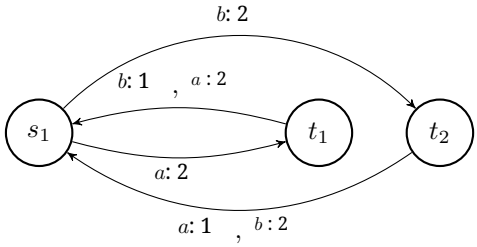


图 3 图 1 右侧分层自动机对应的交替奇偶自动机。

确定性奇偶自动机作为分层自动机。 确定性自动机可以看作分层自动机的一种特殊情况，下一条命题将对此进行说明。

▶**命题 3.8**。 设 $\mathcal{B}$ 是一个 确定性奇偶自动机。存在一个 分层自动机 $\mathcal{A}$ 使得 $[\![\mathcal{A}]\!] = \mathcal{B}$。

证明。令 $\mathrm{B}_{\geq x}$ 为将 B 限制在优先级为$\geq x$的转移上所得到的迁移系统。令 A 为分层自动机，其迁移系统为 $\mathrm{T}_x = \{x\} \times \mathrm{B}_{\geq x}$，态射为$\mu_x((x+1,q)) = (x,q)$，其中涵盖所有优先级 x 和状态 q。（注意，x-分量仅为使集合 Q_x互不相交而添加。）不难验证 $[\![\mathrm{A}]\!] = \mathrm{B}$。◀

最小 HD coBüchi 自动机。 一个（非确定性的）*coBüchi* 自动机是一个（$\Sigma \times [1,2]$）-迁移系统。Kuperberg 和 Skrzypczak 表明，历史确定性 coBüchi 自动机不仅可以剪枝为语义确定的，还可以同时剪枝为安全的确定性自动机。后一性质意味着，非确定性选择仅出现在携带优先级 1 [KS15]的转移中。此类自动机是基于优先级的简单自动机，其优先级 $[1,2]$ 满足：对于每个 $q \in Q$ 和 $a \in \Sigma$，至多有一个转移 $q \xrightarrow{a,2} q'$。

在此工作的基础上，Abu Radi 和 Kupferman 表明，HD coBüchi 自动机可以在多项式时间内最小化；并且在假设其1-饱和 [AK22]时，最小自动机是唯一的。自动机1-饱和，当且仅当 $q \xrightarrow{a,1} p$时，$q \xrightarrow{a,1} p'$ 对于所有p' 满足 $L(p') = L(p)$。这种最小化是此处所介绍的分层自动机最小化的一个特例。

▶**命题 3.9。** 设 $\mathcal{B}$ 是一个 语义确定的、安全确定的、1-饱和余 *Büchi* 自动机。存在一个 分层自动机 $\mathcal{A}$ 使得 $[\![\mathcal{A}]\!] = \mathcal{B}$。

证明。 设 $\mathcal{T}_1$ 是 $\mathcal{B}$ 的剩余自动机，且 $\mathcal{T}_2 = \mathcal{B}_{\geq 2}$，其中 $\mu_1(q) = L(q)$。显然可以验证 $[\![\mathcal{A}]\!] = \mathcal{B}$。◀

3.2.3 从交替奇偶自动机到分层自动机

我们已将分层自动机引入为一种树形结构，并将其解释为交替奇偶自动机。这里我们从另一个角度探讨这一等价类。我们通过状态上的、满足良好性质的等价关系，直接刻画由分层自动机 产生的交替奇偶自动机。

考虑一个 按优先级简化的 交替奇偶自动机，并配以一族 d定义在其状态上的嵌套等价关系，即：

$$Q \times Q \supseteq \sim_1 \supseteq \sim_2 \supseteq \cdots \supseteq \sim_d .$$

我们考虑以下性质。对于所有的 $a \in \Sigma$、$x \in [d]$ 和 p，$q \in Q$:

- (x-*determinism*) If $p \xrightarrow{a:x} q$ and $p \xrightarrow{a:x} q'$, then $q' \sim_x q$.
- (x-一致性) 若 $p \sim_x p'$ 且 $p \xrightarrow{a:\geq x} q$，则 $p' \xrightarrow{a,\geq x} q'$ 对某个 $q' \sim_x q$。
- (x-饱和) 若 $p \xrightarrow{a,x} q$，则对所有 $q' \sim_x q$，都存在一个转移 $p \xrightarrow{a,x} q'$。

▶**命题 3.10。** 设 $\mathcal{B}$ 是一个 优先级简单 自动机。那么，存在一个 分层自动机 $\mathcal{A}$ 使得 $\mathcal{B} = [\![\mathcal{A}]\!]$ 当且仅当 $\mathcal{B}$ 可以配备 d 嵌套等价关系 $\sim_1$、……、$\sim_d$ 满足 x-确定性、x-一致性 和 x-饱和。

证明。 假设 $\mathcal{B} = [\![\mathcal{A}]\!]$ 对于一个 分层自动机 $\mathcal{A}$。那么，我们可以定义 $q \sim_x p$ 若$\widehat{\mu}_x(q) = \widehat{\mu}_x(p)$ 以及 q，$p \in Q_{\geq x}$。根据 $[\![\mathcal{A}]\!]$, 的定义，显然这三个性质均成立。

假设 $\mathcal{B}$可以赋予 d嵌套等价关系 $\sim_1, \ldots, \sim_d$，且满足上述三个性质。令 $\mathcal{T}_x$ 为具有 迁移系统，其状态为 $\sim_x$-类的

B，且 $[p]_x \xrightarrow{a}_x [q]_x$ 若 $p \overset{a:\geq x}{\Longrightarrow} q'$ 属于 B 中某个 $q' \in [q]_x$（根据 x-饱和性，这意味着$p \overset{a:\geq x}{\Longrightarrow} q'$ 对所有 $q' \in [q]_x$都成立）。这些转移由 x-相干性良好定义。迁移系统 T_x 由 x-确定性可知是确定性的。我们定义 $\mu_{x+1}([q]_{x+1}) = [q]_x$，根据 T_x的转移定义，它是一个态射。$\mathrm{B} = [\![\mathrm{A}]\!]$ 这一事实很容易验证。◀

不过，我们注意到，也可以为$\mathcal{B}$ 赋予满足上述三个性质的不同等价关系。在这种情况下，上述证明会给出两个不同的分层自动机 $\mathcal{A}$，$\mathcal{A}'$ 使得 $[\![\mathcal{A}]\!] = [\![\mathcal{A}']\!] = \mathcal{B}$。

3.3 通过安全语言刻画语义

本节提出一种对分层自动机语言的替代刻画方式，避免明确提及交替。此前的定义采用了一致语义确定性这一交替自动机的经典概念，并借此引入相关概念；现在，我们提出一种更直接的方法。我们直接在分层自动机上施加一项称为一致性的限制，并证明它等价于一致语义确定性。对于一致的自动机，我们可以通过一种不依赖交替的 $[\![\mathcal{A}]\!]$接受条件来刻画其语言。这种接受概念与词的自然颜色 [ES22] 以及重轨自动机的接受机制 [Ehl25]，十分相似，更多讨论见第 3.6 节。

安全语言。 状态 x-安全语言 定义为状态 $q \in Q_{\geq x}$ 的

$$L_x(q) = \{w \in \Sigma^* \cup \Sigma^\omega \mid \text{在 } w \text{ 上从 } \widehat{\mu}_x(q) \text{ 开始的运行定义于 } \mathcal{T}_x\}。$$

也就是说，$L_x(q)$ 是从 祖先（即 q 在层级 x）出发存在路径的词。注意，根据 完备性，$\mathcal{T}_1$ 我们有 $L_1(q) = \Sigma^* \cup \Sigma^\omega$ 对每个状态 q均成立。当我们需要指明自动机 $\mathcal{A}$时，记作 $L_x^{\mathcal{A}}(q)$。

▶ **例 3.11。** 对于 分层自动机 （图 1 左侧），我们有

$$L_2(q) = \text{不含因子 } cc, \quad L_3(q_2) = \text{不含因子}$$
的词 cc 和 aa.

我们有以下等式，描述标记着路径的词（其中路径上没有优先级 $< x$）与 $[\![\mathcal{A}]\!]$ 的 x-安全语言之间的关系。回想一下，若路径上的最小优先级为 $\geq x$，我们记作 $\xrightarrow{w\,\geq x}$ 。

$$\begin{aligned} L_x(q) &= \{w \in \Sigma^* \cup \Sigma^\omega \mid \text{there is a path } \widehat{\mu}_x(q) \xrightarrow{w}_x \text{ in } \mathcal{T}_x\} \\ &= \{w \in \Sigma^* \cup \Sigma^\omega \mid \text{there is a path } [q]_x \xrightarrow{w:\geq x} \text{ in } [\![\mathcal{A}]\!]\} \\ &= \{w \in \Sigma^* \cup \Sigma^\omega \mid \text{no path } [q]_x \xrightarrow{w} \text{ in } [\![\mathcal{A}]\!] \text{ sees a priority } < x\}. \end{aligned}$$

▶ 定义 *3.12*。考虑一个分层自动机 A。若存在一个分解 $uw' = w$ 以及层 x 上的一个状态 p，使得 $q_{\text{init}} \xrightarrow{u}_1 \widehat{\mu}_1(p)$ 且 $w' \in L_x(p)$，则称词 $w \in \Sigma^\omega$ 在层 x 中是最终安全的。若词 w 最终安全的最大层为偶数，则称自动机 A 接受词 w 。若该最大层为奇数，则称自动机 A 拒绝词 w 。

我们还引入了一个相关概念：强接受，借此可以定义一致性，并最终证明由 $\mathcal{A}$ 接受与 $[\![\mathcal{A}]\!]$ 的语言之间的等价关系。

强接受。我们称 $w \in \Sigma^\omega$ 被强接受 p，如果 $w \in L_x(p)$，p 位于某个偶层 x，并且对于每个分解 $w = uw'$，都不存在 $p' \in Q_{x+1}$ 满足 $\widehat{\mu}_x(p) \xrightarrow{u}_x \widehat{\mu}_x(p')$ 且 $w' \in L_{x+1}(p')$ 的 w。对于强拒绝的定义类似，只需将 x 奇数。

直观地说，强接受意味着，从 $[p]_x$ 经由 w 在 $[\![\mathcal{A}]\!]$ 出发的每一次运行中，只有优先级 $\geq x$ 会出现，且 x 会无限次出现，而不论参与者如何选择。由于 x 是偶数，所有这些运行都被接受。

原则上，一个词可能既被某个状态强接受，又被另一个状态强拒绝。我们通过施加一个称为一致性的额外条件来避免这种情况，具体定义如下。我们将证明，在一致性假设下，我们引入的所有接受机制都等价：由交替自动机 $[\![\mathcal{A}]\!]$，接受于 $\mathcal{A}$，或者存在（分别为不存在）一个后缀，它在某个适当的剩余中被某个状态强接受（分别被强拒绝）（见推论 3.18）。

▶ **例 3.13**。在左侧的分层自动机中，任何包含无限多个因子 cc 的词都会被状态强拒绝。$r \in Q_1$ 状态 $q \in Q_2$ 强接受不包含因子 cc 且包含无限多个因子 aa 的词。

接下来的两个引理将强接受在 $\mathcal{A}$ 与由 $[\![\mathcal{A}]\!]$ 的接受联系起来。

▶引理 *3.14*。设 *A* 是一个分层自动机，p 是一个叶状态。若词 w 被 $\widehat{\mu}_x(p)$ 强接受，其中某个 x 为偶数，则 $w \in L([\![A]\!], p)$。类似地，若 w 被 $\widehat{\mu}_x(p)$ 强拒绝，其中某个 x 为奇数，则 $w \notin L([\![A]\!], p)$

证明。我们证明，任意运行从 p 出发，在 w 上运行时，都会无限多次产生优先级 x（无论 Eve 和 Adam 采用何种策略）。由于 $w \in L_x(p)$，从 p 出发，在 w 上运行的任何运行只会产生优先级 $\geq x$。如果这样的运行没有无限多次产生优先级 x，那么它的形式为 $p \xrightarrow{u} q \xrightarrow{w'} \xrightarrow{>x}$，这就导致矛盾。◀

▶ **引理 3.15**。若 $\mathcal{A}$ 接受 w，则存在一个分解 $uw' = w$ 以及一个状态 p 位于某个偶层 x，使得 $q_{\text{init}} \xrightarrow{u}_1 \widehat{\mu}_1(p)$，且 w' 从被强接受于 p。

证明。考虑最大层 x，其中 w 是最终安全的；由于 w 是接受的。根据定义，我们有一个分解 $uw' = w$，其中包含一个状态 p、一条路径 $q_{\text{init}} \xrightarrow{u}_1 \widehat{\mu}_1(p)$，以及 $w' \in L_x(p)$。不难验证，w' 从强接受 p，否则 x 就不会是 w 最终安全的最大层。◀

为了得到所需的刻画，我们还需要一个对应于一致语义确定性的条件。

一致性 如果不存在一对状态 分层自动机 $\mathcal{A}$ 是 一致的：状态 p、$p' \in Q_{\geq 1}$ 满足 $\widehat{\mu}_1(p) = \widehat{\mu}_1(p')$，且存在一个词 $w \in \Sigma^\omega$ 使得 w 被强接受，由 p 而 w 被强拒绝，由 p'。

如下文所述，这一概念等价于 一致语义确定性，针对 $[\![\mathcal{A}]\!]$。我们将在命题 3.26 中证明，可以在多项式时间内检验这一性质。

我们陈述这一关键结果：对于一致分层自动机，通过强接受获得的语义与通过交替奇偶自动机获得的语义相同。证明需要引入最长后缀解析器，其中 $[\![\mathcal{A}]\!]$，相关内容留待下文第 3.4 节给出。

▶ **定理 3.16**。设 $\mathcal{A}$ 是一个一致分层自动机。词 $w \in \Sigma^\omega$ 被 接受 当且仅当它被 $\mathcal{A}$ 接受，当且仅当 $w \in L([\![\mathcal{A}]\!])$。此外，$[\![\mathcal{A}]\!]$ 是 一致语义确定的。

▶**推论 3.17**。 分层自动机 是 一致 当且仅当它是 一致语义确定的。

证明。 从左到右的蕴含由上述定理给出。反方向则可直接由定义和引理 3.14 得出。◀

▶ **推论 3.18**。 设 $\mathcal{A}$ 是一个一致分层自动机，并设 $w \in \Sigma^\omega$。以下命题等价：

- $w \in L([\![\mathcal{A}]\!])$，$w$ 被 A 接受；存在一个分解 $uw' = w$ 和一个状态 p，使得 $q_{\text{init}} \xrightarrow{u}_1 \widehat{\mu}_1($
- $p)$ 且 p 强接受 w'；不存在一个分解 $uw' = w$ 和一个状态 p，使得 $q_{\text{init}} \xrightarrow{u}_1 \widehat{\mu}_1(p)$ 且
- p 强拒绝 w'。
-

3.4 分层自动机是历史确定的且具有 0-1 概率性

本节将证明定理 3.16 以及以下命题。

▶**命题 3.19**。 对于每个 一致分层自动机 $\mathcal{A}$，交替奇偶自动机 $[\![\mathcal{A}]\!]$ 是 历史确定性的。

我们称一个 按优先级简单的奇偶自动机$\mathcal{A}$ 是 *0-1*概率自动机，如果对于每个从 ρ 开始、在词 $w \in \Sigma^\omega$ 上进行的 q_{init}随机游走，都有：

- $\Pr(\rho \text{ 被接受 }) = 1 \iff w \in L(\mathcal{A})$，并且
- $\Pr(\rho \text{ 被接受 }) = 0 \iff w \notin L(\mathcal{A})$。

我们注意到，0-1概率自动机 对于 马尔可夫决策过程 是有用的，其意义在于 [HPS+20]参见 [BGB12, HPT25] 了解相关概念。

▶**命题 3.20**。 对于每个 一致分层自动机 $\mathcal{A}$，交替奇偶自动机 $[\![\mathcal{A}]\!]$ 是 *0-1*概率自动机。

我们首先介绍最长后缀解析器。这是一类解析器，它们见证了历史确定性 的分层自动机，并具有一些有用的性质。

最长后缀解析器。 对于 优先级 x、有限词 u和一个 x状态 q，我们将最长 x后缀 u 到 q 的最长后缀定义为 v 中 u 满足以下条件的后缀：对于某个 $p \xrightarrow{v}_x q$，在 $\mathcal{T}_x$中，存在某个 p。注意，该后缀可能为空。

我们记 Q和 Δ 分别为 $[\![\mathcal{A}]\!]$, 的状态集合和转移集合。

Eve 的最长后缀解析器[4] 是一个解析器 $\sigma : (\{q_{\text{init}}\} \cup \Delta^+) \times \Sigma \to Q$，满足以下条件：每当在 [A] 中有一局 $p_0 \xrightarrow{v} p \xrightarrow{a,x} [q]_x$，且 x 为奇数时，Eve 就选择任意一个状态 $q' \in [q]_x$，使其能够接受 va 到 $\widehat{\mu}_{x+1}(q')$ 的最长（$x+1$）后缀。Adam 的最长后缀解析器定义对称地定义。

下一个引理陈述了最长后缀解析器的主要性质。它保证，只要存在满足此性质的运行，对局就会离开奇数层。该陈述使用了更一般的概念：将最长后缀解析器初始化为 $u \in \Sigma^*$。它是一个函数

[4] 形式化地说，我们定义一个解析器族，因为定义中存在一些非确定性选择。稍微滥用一下记号，我们将其中任意一个称为 最长后缀解析器。

$\sigma_u : (\{q_{\mathsf{init}}\} \cup \Delta^+) \times \Sigma \to Q$ 满足以下条件：每当有一个对局 $p_0 \xrightarrow{v} p \xrightarrow{a:x} [q]_x$ 在 $[\![\mathcal{A}]\!]$ 中，且x 为奇数时，Eve 都会选择任意一个 $q' \in [q]_x$ 具有最长$(x+1)$后缀的 uva 以 $\widehat{\mu}_{x+1}(q')$结尾的状态，而非选择 va 的最长后缀，如同未初始化的解析器所做的那样。因此，先前的概念相当于用空字进行初始化。

▶引理 *3.21*。设 x 是一个奇数优先级，p 是分层自动机 A 中的一个叶状态。假设 $w = u\tilde{w} \in \Sigma^\omega$ 存在一条运行 $p \xrightarrow{u:\geq x} q \xrightarrow{\tilde{w}:\geq x+1}$ 在 $[\![A]\!]$ 中（等价地，$w \in L_x(p)$，但未被 $\widehat{\mu}_x(p)$ 强拒绝）。那么，对于每个 u_0，任何初始化为 u_0 、为 *Eve* 服务的最长后缀解析器 σ 都能确保：在从 p出发、以词 w 为输入的任何 σ-运行中，最终所有优先级都是 $\geq x+1$。

证明。考虑一场从 p 开始的对局，它遵循一个 为 Eve 服务的最长后缀解析器，其初始化为某个u_0，并在某个前缀 uw' 之后看到了 x 优先级。该前缀是 $u\tilde{w}$：

$$p \xrightarrow{u:\geq x} q \xrightarrow{w':\geq x} q' \xrightarrow{a:x} q'' \ .$$

在这种情况下，初始化为 u_0 的最长后缀解析器会找到 u_0uw' 的某个后缀 v，使得其中带有$p_{va} \xrightarrow{va}_{x+1} \widehat{\mu}_{x+1}(q'')$，且位于 T_{x+1} 中（该自动机中存在 p_{va} 某个状态）。注意，根据引理的假设，w' 是 v 的一个后缀，因为 $\widehat{\mu}_{x+1}(q) \xrightarrow{w'a}_{x+1} \widehat{\mu}_{x+1}(q'')$。因此，这条在 v 上的路径可以表示为 $p_{va} \xrightarrow{v'}_{x+1} q_{w'a} \xrightarrow{w'a}_{x+1} \widehat{\mu}_{x+1}(q'')$，其中 v'，且 $q_{w'a}$ ∈属于 T_{x+1}。我们称 $q_{w'a}$ 为 $w'a$的支撑。现在考虑最长后缀运行下一次看到优先级 x的时刻，即

$$p \xrightarrow{u:\geq x} q \xrightarrow{w':\geq x} q' \xrightarrow{a:x} q'' \xrightarrow{w'':>x} q^{(3)} \xrightarrow{b:x} \ .$$

这意味着我们有 $q_{w'a} \xrightarrow{w'a}_{x+1} \widehat{\mu}_{x+1}(q'') \xrightarrow{w''}_{x+1} \widehat{\mu}_{x+1}(q^{(3)})$，且在 T_{x+1} 中不存在从 $\widehat{\mu}_{x+1}(q^{(3)})$ 出发的 b- 转移。因此，$q_{w'a}$ 不能作为 $w'aw''b$ 的支撑，并且可以通过上述推理找到一个新的支撑。这一论证表明，每当一场遵循初始化为 u_0 的最长后缀解析器的对局遇到优先级 x 时，就会有一个 $(x+1)$-状态被排除为潜在支撑。因此，这样的对局能够看到优先级 x 的次数受 $(x+1)$-状态数量的限制。◀

陈述下一个引理时，请回顾 $(\mathcal{A}\ q)$ 表示 $\mathcal{A}$ 其中 q 被视为初始状态。对于分层自动机，初始状态应来自第 1 层，而 $[\![\mathcal{A}]\!]$ 的初始状态是 $\mathcal{A}$ 的叶节点。

▶引理 *3.22*。设 *A* 是一个一致分层自动机，q 是 $[\![A]\!]$, 的一个状态，σ 和 τ分别是 $[\![A]\!]$, 中 *Eve* 和 *Adam* 的最长后缀解析器。如果词 w 被 $(A\ \widehat{\mu}_1(q))$ 接受，那么 $([\![A]\!], q)$ 在 w 上的每个 σ-运行都是接受运行。对称地，如果词 w 被 $(A\ \widehat{\mu}_1(q))$ 拒绝，那么 $([\![A]\!], q)$ 在 w 上的每个 τ-运行都是拒绝运行。

证明。我们先回顾引理 3.15：如果 w 被 接受 于 $(\mathcal{A}\ q)$ 那么存在一个分解 $u'w' = w$ 以及一个状态 p' 位于某个偶层 y，使得 $q \xrightarrow{u'}_1 \widehat{\mu}_1(p')$，且$w'$ 从被强接受，其起点为 p'。

假设 Eve 的最长后缀解析器 在 Eve 于 w 进行对局时拒绝 $([\![\mathcal{A}]\!], q)$。取一个与该解析器一致的非接受运行，

$$q \xrightarrow{u''} p'' \xrightarrow{w'':\geq x} \quad \text{in } [\![\mathcal{A}]\!],$$

其中，对于某个奇数 x，最终只出现优先级 $\geq x$，且 x 无限次出现。我们可以假设 u' 是 u'' 的前缀。

首先，我们证明 w'' 在 A 中从 $\widehat{\mu}_x(p')$ 被强拒绝。由于 $[\![\mathcal{A}]\!]$ 中的运行避开了优先级 $< x$，因此我们有 $w'' \in L_x(p')$。如果 w'' 未被强拒绝，那么根据引理 3.21，初始化为 u'' 的 Eve 最长后缀解析器最终只会看到来自 p'' 的优先级 $\geq x+1$；这与 $[\![\mathcal{A}]\!]$ 中的运行里 x 无限次出现这一事实矛盾。

现在设 $u'' = u'v$ (u' 是 u'') 的前缀。我们有一条路径 $p' \xrightarrow{v}_y p_y$ 位于偶层 $\mathcal{T}_y$ 中，且 w'' 从强接受 p_y。此外，$\widehat{\mu}_1(p'') = \widehat{\mu}_1(p_y)$。这与一致性 $\mathcal{A}$ 相矛盾。◀

上述引理给出了定理 3.16 的第一部分，即对于任意一致分层自动机 $\mathcal{A}$，由 $\mathcal{A}$ 接受 $[\![\mathcal{A}]\!]$ 的结果与由其接受的结果相同。它还证明了 $[\![\mathcal{A}]\!]$的历史确定性（命题 3.19）。

▶**推论 3.23**。 设 $\mathcal{A}$ 是一个一致分层自动机。那么，$[\![\mathcal{A}]\!]$ 是历史确定性的，且最长后缀解析器是有效解析器。

接下来需要证明，一致的 $[\![\mathcal{A}]\!]$ 是一致语义确定的。

▶**引理 3.24**。 若 $\mathcal{A}$ 是一个一致的自动机，则它是一致语义确定。

证明。 注意，接受（通过安全语言）$(\mathcal{A}\ q)$ 仅取决于 $\widehat{\mu}_1(q)$。由于引理 3.22 对 $\mathcal{A}$ 中任意初始状态的选择均成立，因此对任意$p \sim_1 q$

$$L([\![\mathcal{A}]\!], p) = L(\mathcal{A}, p) = L(\mathcal{A}, q) = L([\![\mathcal{A}]\!], q).$$ ◀

随机策略。 命题 3.20 的证明几乎完全相同。

▶ **引理 3.25**。 设 p 是一个分层自动机 A中的。设 $uw \in \Sigma^\omega$ 接受一个运行 $p \overset{u:\geq x}{\longrightarrow} q \overset{w:\geq x+1}{\longrightarrow}$ 于 $[\![\mathcal{A}]\!]$中。那么，从 uw 出发，在词 p 上的任意随机游走，最终只会产生优先级 $\geq x+1$。

证明。 设 n 为叶状态的数量。设 $w = uw'a\tilde{w}$ 并设

$$p \xrightarrow{u:\geq x} q' \xrightarrow{w':\geq x} q'' \xrightarrow{a:x} [\tilde{q}]_x$$

是在 uw的某个前缀上、从 p 出发并产生优先级 x的随机游走。根据假设，在 $[\tilde{q}]_x$ 中至少存在一个状态 $\tilde{q}'$，使得 $\tilde{w} \in L_{x+1}(\tilde{q}')$。因此，随机游走到达该状态的概率至少为 $1/n$。所以，在前缀 u 之后看到具有优先级 x的 k 次转移的概率小于 $(1-1/n)^k \xrightarrow[k\to\infty]{} 0$。◀

命题 3.20 的证明。 设 $w \notin L(\mathcal{A})$（情形 $w \in L(\mathcal{A})$ 是对称的），并设 x 是一个偶数 优先级。我们将证明，随机游走在 w 上产生 x 作为最小优先级无穷多次的概率为 0。该事件发生的概率，就是获得如下形式的路径的概率：

$$q_{\text{init}} \xrightarrow{u} p \xrightarrow{w':\geq x} \quad \text{producing } x \text{ infinitely often, with } w = uw'.$$

对于每条这样的路径 $q_{\text{init}} \xrightarrow{u} p$，由于 $w' \in L_x(p)$ 且未被 $\widehat{\mu}_x(p)$强拒绝，存在一次运行 $p \overset{u':\geq x}{\longrightarrow} q \overset{w'':\geq x+1}{\longrightarrow}$，满足 $w' = u'w''$。因此，根据引理 3.25，获得一条路径 $p \overset{w':\geq x}{\longrightarrow}$ 并使 x 无限多次出现的概率为 0。◀

3.5 多项式时间内的一些判定过程

我们证明，可以在多项式时间内检查给定的分层自动机 A 是否一致。我们还可以在多项式时间内检查一致分层自动机的空性和语言包含。

▶**命题 3.26**。 判断给定的分层自动机是否一致，

证明。 我们检查是否存在一对 p，$p' \in Q_{\geq 1}$ 与一致性的定义冲突，即 $\widehat{\mu}_1(p) = \widehat{\mu}_1(p')$ 且存在一个词 w 被强接受，由 p 并且 被强拒绝，由p'。候选项数量是多项式级的，因此我们可以逐一检查。给定$p \in Q_x$ 和 $p' \in Q_{x'}$，其中 x 为偶数且 x' 为奇数，我们可以通过博弈检查是否存在一个冲突词 $w \in \Sigma^\omega$ 。在这场博弈中，一名玩家 证明者 试图找出一个词作为不一致的见证，而 反驳者 则试图证明构造出的词不是见证。

博弈从 (p, p')开始。每一轮中，证明者选择下一个字母，反驳者则根据所选字母和转移函数选择下一对状态。如果从 p 出发的运行中仅出现优先级 $\geq x$，且 x 无限次出现；并且从 p' 出发的每条运行在 w上都仅出现优先级 $\geq x'$，且 x' 无限次出现，那么证明者获胜。这是一个广义 Büchi 博弈，可以在多项式时间内求解。我们声称，证明者有获胜策略，当且仅当 w 被 强接受，由 p 且 被 强拒绝，由 p'。显然，如果满足此条件，那么证明者可以选择词 $w \in \Sigma^\omega$，而反驳者所能选择的所有运行都使证明者赢得这场博弈。

对于另一个方向，假设证明者在该博弈中拥有一个获胜策略，并让她按照该策略进行对局。由该策略产生的任何词都具有以下性质：从 p 访问的所有优先级都是 $\geq x$，而从 p' 访问的所有优先级都是 $\geq x'$。让反驳者在两个分量中都使用一个最长后缀解析器来选择运行，并令 $w \in \Sigma^\omega$ 为证明者的策略对抗反驳者这一策略时产生的词。如果在 w 上存在一个从 p 开始、且不无限次地产生 x 的运行，那么从某个位置开始，该运行只产生优先级 $\geq x+1$。反驳者用于选择运行的最长后缀解析器随后会导致一个从某个位置开始只看到优先级 $\geq x+1$ 的运行（见引理 3.21）。对于 p'也是如此。◀

▶**命题 3.27**。 给定一个一致分层自动机$\mathcal{A}$，我们可以在多项式时间内检查 $L(\mathcal{A}) = \emptyset$。

证明。 由于我们假设 $\mathcal{T}_1$ 中的所有状态均可达，$L(\mathcal{A}) \neq \emptyset$ 当且仅当存在某个状态 $p_0 \in Q_{\geq 1}$ 位于偶层 x 且 强接受 某个词。

我们可以通过一个与上文所讨论的博奇博弈类似的博弈来检验情况是否如此。两位参与者在 $(p, q) \in Q_x \times Q_{x+1}$ 上进行对局。起始位置为 (p_0, q_0)，其中 q_0 是 $\mu_x^{-1}(p_0)$ 中的任意状态。每一步，证明者都会给出下一个字母，使得得到的词对于 p_0 是安全的。下一个状态 $p \xrightarrow{a}_x p'$ 由确定性的方式给出。如果转移 $q \xrightarrow{a}_{x+1} q'$ 存在，就执行该转移；否则，反驳者可以选择重置到 $\mu_x^{-1}(p')$ 中的某个状态。若在层级 $x+1$ 发生无限多次重置，则证明者获胜。显然，证明者在且仅在 p_0 强接受某个词时，才能赢得这场博弈。实际上，只要生成的词未被强接受，反驳者就可以使用最长后缀解析器获胜。◀

▶**命题 3.28**。给定两个一致分层自动机$\mathcal{A}$ 和$\mathcal{A}'$，我们可以在多项式时间内检查 L（$\mathcal{A}$）$\subseteq L$（$\mathcal{A}'$）。

证明。 我们证明，可以将该问题归约为求解一个具有 3 个多项式规模 Rabin 对的拉宾博弈。由此可以得出结论，因为求解具有固定数量 Rabin 对的拉宾博弈可以在多项式时间内完成（例如参见 FAA，定理 3.）。

获胜条件实际上略简单一些：如果博弈一直停留在第 1 阶段，证明者将输掉；否则，在第 2 阶段，如果满足两个 Büchi 条件，证明者将获胜。

在这个博弈中，证明者试图通过给出 $L(\mathcal{A}) \setminus L(\mathcal{A}')$ 中的一个词，来证明 $L(\mathcal{A}) \not\subseteq L(\mathcal{A}')$。证明者逐个给出字母。在第一个阶段，这些字母确定两条运行：

$$q_{\mathsf{init}} \xrightarrow{a_1}_1 q_2 \xrightarrow{a_2}_1 q_3 \xrightarrow{a_3}_1 \dots \text{ in } \mathcal{T}_1 \quad \text{and} \quad q'_{\mathsf{init}} \xrightarrow{a_1}_1 q'_2 \xrightarrow{a_2}_1 q'_3 \xrightarrow{a_3}_1 \dots \text{ in } \mathcal{T}'_1.$$

在任意时刻，证明者都可以决定在 $\mathcal{A}$ 中进入第 2 阶段：他转到某个对 (p_i, s_i)，其中 $p_i \in \widehat{\mu}_1^{-1}(q_i)$ 位于某个偶层 $\mathcal{T}_x$，且 $s_i \in \mu_x^{-1}(p_i)$。类似地，在任意时刻，证明者都可以在 $\mathcal{A}'$ 中进入第 2 阶段，转到 (p'_i, s'_i)，其中 $p'_i \in \widehat{\mu}_1^{-1}(q'_i)$ 位于某个奇数层 $\mathcal{T}_{x'}$，且 $s'_i \in \mu_{x'}^{-1}(p'_i)$。

如果对于 $\mathcal{A}$ 或 $\mathcal{A}'$，证明者始终没有进入第 2 阶段，那么他就会输。

当自动机 $\mathcal{A}$ 处于第 2 阶段且位于一对 (p_i, q_i) 中时，博弈与前一个命题中的相同。转移 $p_i \xrightarrow{a_i}_x p_{i+1}$ 必须存在（否则，证明者自动输掉博弈）。如果转移 $q_i \xrightarrow{a_i}_{x+1} q_{i+1}$ 在 $\mathcal{T}_{x+1}$ 中存在，那么我们就转到第二分量中的 q_{i+1}。如果该转移不存在，那么这就是第一条 Büchi 条件的一条接受边，反驳者会将第二分量重置为任意一个 $q_{i+1} \in \mu_x^{-1}(p_{i+1})$。

$\mathcal{A}'$ 中的第 2 阶段是相同的。我们在第一分量中沿着 $\mathcal{T}'_{x'}$ 中的运行进行，而该运行必须存在（否则，证明者输掉博弈）。必要时，反驳者可以选择如何在 $\mathcal{T}'_{x'+1}$中重置第二分量。每次重置发生时，我们都会看到第二条 Büchi 条件的一条接受边。

显然，当且仅当证明者能够无限多次看到两条 Büchi 条件中的边时，$L(\mathcal{A}) \setminus L(\mathcal{A}')$中才存在一个词。◀

3.6 与其他模型的联系和比较

本节将讨论分层自动机与文献中的其他模型之间的关系。有时我们会提到第 4 节中针对某个语言所引入的最小分层自动机。关于分层自动机与其他表示形式相比，在规模方面的 ω-正则语言的详尽分析将留待未来工作；这里只对这一问题作一些说明。

确定性自动机和交替自动机。 首先，我们注意到，一致分层自动机可能比确定性的自动机小一个指数级数量级，这一点对于历史确定性 coBüchi 自动机 [KS15，定理 1]中已经如此。另一方面，它们可能比交替奇偶自动机大两个指数级数量级。对于有限字上的自动机，这种双指数差距已经出现 [CKS81，定理 5。3]。这一差距同样适用于一致分层自动机，因为在第一层上，它们至少需要拥有与该语言中的剩余语言数量一样多的状态。

最小HD coBüchi 自动机。如命题 3.9 所述，最小HD coBüchi自动机可视为分层自动机的一个特例。我们在第 4 节中介绍的、刻画最小分层自动机的性质，是对 Abu Radi 和 Kupferman 提出的安全最小性与安全中心性概念的推广 [AK22]。

Muller 语言。 熟悉 Zielonka 树的读者 [Zie98] 及其奇偶自动机 [DJW97, CCFL24] 可能已经注意到，分层自动机 与这些概念十分相似。事实上，不难看出，Muller 语言 L 的 Zielonka 树

可以作为 树，即 识别 L 的分层自动机的树。在这种情况下，相关自动机 $[\![\mathcal{A}]\!]$通过剪枝即可成为确定性的。任何这样的剪枝都会得到语言 L 的Zielonka 树对应的确定性奇偶自动机。关于 [CCFL24，第 4]节中给出了 Zielonka 树的奇偶自动机的定义。

位置语言。 2024 年，Casares 和 Ohlmann 刻画了哪些 ω-正则语言 具有 位置性，也就是说，哪些语言 L 使得在以 L [CO24] 为获胜条件的博弈中，存在玩家总能使用位置策略进行最优对局。在这一刻画中，他们使用了所谓的 签名自动机。事实证明，签名自动机其实只是分层自动机，外加一些确保其具有位置性的性质。

更确切地说，我们可以将他们的刻画（ [CO24，定理 3.1] 中的第 (2) 点）重述如下：一个 ω-正则语言 L 具有位置性，当且仅当其最小分层自动机$\mathcal{A}_L$ 满足以下条件：

1. 偶层上的节点至多有一个子节点，**2.** 对于每个 x，$\mathcal{T}_x$ 的每个强连通分量中的状态都按 x-安全语言的包含关系全序排列，**3.** 剩余语言按包含关系全序排列（这可以视为前一条件的一个特例），并且**4.** 每个偶层 x 都进展一致。也就是说，如果 $p \xrightarrow{u}_x q$ 且 $L_x(p) \subsetneq L_x(q)$，那么 u^ω就被 强接受于 p。

coBüchi 自动机链（COCOA）。在 [ES22], Ehlers 和 Schewe 提议用一个递减的序列来表示 ω-正则语言，该序列由coBüchi 语言 $L_1 \supseteq L_2 \supseteq \cdots \supseteq L_d$，[5]其中 L_x 是那些在层级不适合 $x-1$ 的单词（定义请参见 [ES22，定义 1、定理 7][6]）。若单词 w 属于由该链表示的语言 L，则最大的索引 x 使得 $w \in L_x$ 为偶数。

从一个一致的分层自动机得到 COCOA 表示很容易。对于每一层 x，定义一个历史确定性 coBüchi 自动机，它基于迁移系统$\mathcal{T}_x$，将现有转移解释为该自动机的2-转移，并为所有缺失的转移，在相应剩余类的所有状态上添加1-转移。因此，语言的 COCOA 表示的大小绝不会超过该语言的最小分层自动机的大小。但如下面的例 3.29 所示，语言的最小分层自动机可能比最小 COCOA 大得多，甚至呈指数级增长。

由于 COCOA 将语言表示为 coBüchi 自动机的布尔组合，因此它们比用单个自动机表示更小也就不足为奇了。为了在综合等应用中使用 COCOA，需要从中得到一个历史确定的自动机。这需要将构成 COCOA 的所有 coBüchi 自动机做乘积 [EK24a]。因此，来自 COCOA 的最终交替奇偶自动机 的状态数，可能远多于由交替奇偶自动机 所表示的分层自动机。例 3.30 中给出了这样一个例子。[7]

[5] [ES22] 中的定义以0作为最小颜色，而我们以1作为最小层。如[ES22], 所述，这些定义可以很容易地调整为以1作为最小颜色的设定。[6] 我们认为，在 [ES22的定理 7] 中，应写作“即 w 在颜色 $i-1$中不处于本地”。[7] 我们在此考虑 [EK24a]的朴素乘积。他们还定义了一种约简乘积和一种忽略某些乘积状态的优化乘积。对这些乘积与分层自动机进行更细粒度的比较留待未来工作。

此外，这个例子的语言是前缀无关的（它只有一个剩余语言），因此，这种膨胀并非仅仅是对该语言的所有剩余语言取乘积所致。

ES 中的一个核心定义是 [ES22] 的自然颜色，即一个词 w 相对于语言 L的自然颜色。我们认为，这种自然颜色体现在最小分层自动机中，针对 L而言，它恰好与最大层 x 相同，在该层上 w 是最终安全的。

▶ **例 3.29**。本例中的语言是对 [ABF18] 中所用语言的轻微修改，该语言用于展示确定性有限自动机与确定性优先自动机的饱和族之间存在指数差距。令 $\Sigma = \{1, \ldots, k\}$，并考虑语言 L，它由所有满足以下两个条件的 $w \in \Sigma^\omega$ 组成：

- w 从某个位置开始，之后无限次出现的不同字母数为偶数；对于每个双字母子串
- ij，都有 $j \in \{1, \ldots, i+1\}$。

第二个条件确保，对于 L 中的所有词，无限次出现的符号集合构成 $\{1, \ldots, k\}$ 中的一个区间。对于 $k = 4$，该语言的最小分层自动机如图 4 所示。这个自动机有 $k+1$ 层。由于该语言前缀无关，第一层只有一个状态，该状态在所有字母上循环。对于 $x \geq 1$，第 x 层上的强连通分量对应于大小为 $k - x + 2$ 的区间。不过，一个区间可能会出现多次。例如，第 4 层上区间 $[2, 3]$ 出现了两次，因为它分别包含于 $[1, 3]$ 和 $[2, 4]$ 中。第 5 层上，单元素区间 $[3, 3]$ 出现了三次，因为它包含于 $[3, 4]$ 和 $[2, 3]$ 中，而后者在前一层上已经出现过两次。由于从第 2 层开始呈二叉树结构，第$k+1$层上有 2^{k-1} 个状态。这些状态由相应的交替历史确定性自动机使用。如果为该语言构造 COCOA，那么每个区间在 COCOA 的每个层级上只出现一次，因此 COCOA 中每个余 Büchi 自动机的状态数至多关于 k 呈二次增长。

进一步说明一下：对于熟悉 ω-半群的读者，这个语言的句法ω-半群的大小为 k^4，因为半群中的一个元素只需存储一个词中出现的首个、最后一个、最小和最大数字。词中的首个和最后一个数字是连接两个词时追踪第二个条件所必需的。并且，假设第二个条件已满足，那么无限多次出现的不同字母数可以由无限多次出现的最小和最大数字推导出来，因为每一步中字母最多只能增加一。

▶ **例 3.30.** 令 $\Sigma = \{a_1, \ldots, a_k, b_1, \ldots, b_k, c\}$，并考虑由满足以下两个条件之一的所有单词组成的语言：

- $\{b_1, \ldots, b_k\}$ 有限次出现，并且 $\min(\{k+1\} \cup \{x \mid a_x \text{无限多次出现}\})$ 为偶数；或者$\{a_1, \ldots, a_k\}$ 有限次出现，并且 $\min(\{k+1\} \cup \{x \mid b_x \text{无限多次出现}\})$ 为偶数。
-

也就是说，这就是这样一类词的语言：一种编号符号只出现有限次，而另一种符号满足奇偶条件（最小值中的 $k+1$ 用于处理只有 c 无限次出现的情况）。最小的分层自动机需要两条分支来检查这两个奇偶条件，如图 5 中的 $k = 4$所示。最小 COCOA 的构造方法是，取每一层 x，将状态上的循环解释为2转移，并为该层缺失的字母补上1转移。对这些 coBüchi 自动机取（朴素）乘积，会产生指数级数量的状态，其中大多数状态只有一个 c 循环。

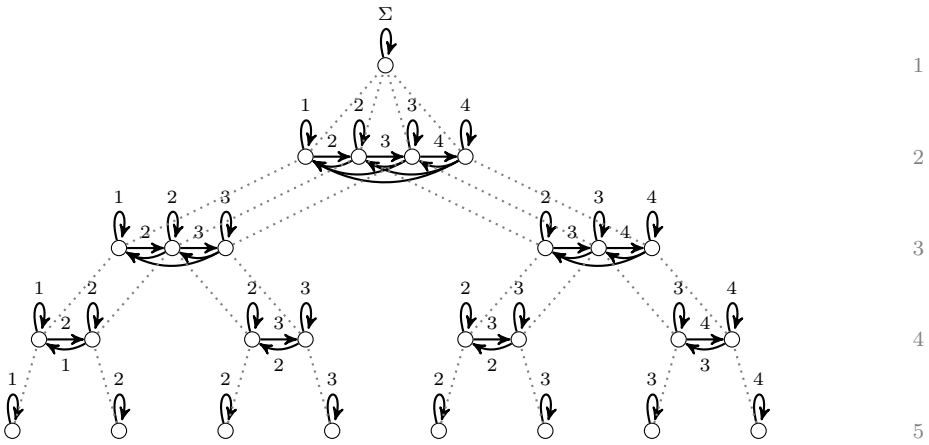


图 4 例 3.29 中 $k = 4$ 的分层自动机。在每一层的每个强连通分量中，所有指向同一状态的边都具有相同的标签。为便于阅读，我们省略了第 2 层和第 3 层中从右向左的边的标签。

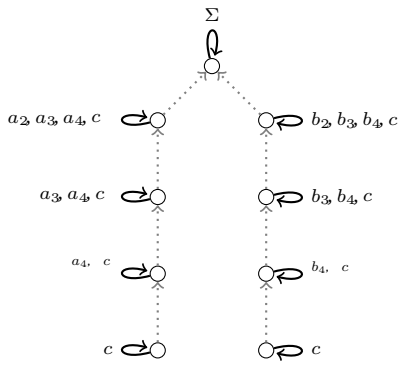


图 5 例 3.30 中 $k = 4$ 的分层自动机。

重轨自动机。2025 年，Ehlers 引入了重轨自动机 [Ehl25]，，这是一类具有特定语义的自动机：当某个词的运行中的最大优先级为偶数时，该词就被接受。

我们认为，一致分层自动机$\mathcal{A}$ 可以通过进一步饱和 $[\![\mathcal{A}]\!]$，转化为重轨自动机。更准确地说，每当存在转移 $p \xrightarrow{a:x} [q]_x$，我们就可以添加转移 $p \xrightarrow{a:y} [q]_y$，对所有 $y < x$ 都如此，且不会改变该自动机所识别的语言。由此得到的自动机具有再轨道性质。此外，当且仅当一个词在饱和后采用重轨自动机的语义时会被接受，它才会被接受。$\mathcal{A}$ 接受。

对于另一个方向，我们不清楚重轨自动机是否是分层的，甚至是否仅仅是HD 交替的。不过，一个语言的最小再轨道自动机与其最小分层自动机或许本质上是同一个对象。

4 一个典范的最小分层自动机

在本节中，我们证明每个 ω-正则语言都存在唯一的最小一致分层自动机。该自动机由三个性质刻画：正规性、中心性和安全最小性（扩展了最小HD coBüchi 自动机 [AK22]的性质）。

▶**定理 4.1**。设$\mathcal{A}$ 和$\mathcal{B}$ 是两个等价的一致分层自动机，且它们正规、中心且安全最小。那么，$\mathcal{A}$ 和$\mathcal{B}$ 是同构的（作为分层自动机）。

两个分层自动机$\mathcal{A}$、$\mathcal{B}$ 同构，当且仅当它们的状态之间存在一个双射，该双射同时保持迁移系统的结构和树结构。也就是说，存在函数 $\varphi\colon \mathcal{A} \to \mathcal{B}$，其在每一层$\mathcal{T}_x$ 上是迁移系统的同构，并且保持由态射给出的子关系 μ_x。（关于分层自动机的态射的进一步定义，参见第 5.1 节。）

4.1 正规性、中心性与安全最小性

我们定义用于刻画最小一致分层自动机的三个性质。

▶ **定义 4.2**。如果一个分层自动机满足以下条件，则称其处于标准形式：对于所有 $x \geq 2$ 和 $q \in Q_x$，以下两个性质均成立。

N1 若 $q \xrightarrow{u}_x q'$ 对于某个 $u \in \Sigma^*$ 成立，则 $q' \xrightarrow{v}_x q$ 对于某个 $v \in \Sigma^+$成立。**N2** 对于所有 $p \in \mu_x^{-1}(q)$，都存在某个 u，使得 $q \xrightarrow{u}_x$ 在$\mathcal{T}_x$ 中有定义，但 $p \xrightarrow{u}_{x+1} \perp$ 在$\mathcal{T}_{x+1}$中无定义。

处于标准形式可确保不存在明显无用的转移；此外，不改变所识别的语言，就无法降低优先级为 $[\![\mathcal{A}]\!]$中的转移优先级。第一个性质意味着，每个$\mathcal{T}_x$ 都是非平凡强连通分量的并。从直观上说，只需删除改变强连通分量的转移即可实现这一点。删除此类转移会使其优先级在 $x-1$ 中降为 $[\![\mathcal{A}]\!]$中的。这不会改变所接受的语言，因为在一次运行中，如果没有另一个优先级 $\leq x-1$ 的转移也无限次出现，此类转移就不可能无限次出现。

第二个性质表明，μ_x 未完全覆盖$\mathcal{T}_x$中的某个强连通分量的所有边，换句话说，子节点的安全语言始终严格包含于父节点的安全语言。否则，p 不会强烈接受/拒绝任何内容。在语义自动机中，该性质意味着如果 $p \xrightarrow{a:\geq x}$，那么存在 u 以及一次运行

$p \overset{u}{\longrightarrow}_{:x}$ from p以 x 作为最小优先级。

一个 x-强连通分量 是 分层自动机 T_x的一个强连通分量。下一个性质要求，任何 x-强连通分量 都不能由另一个强连通分量模拟。下面给出形式化定义。

▶ **定义 4.3**。 对于 $x \geq 2$，以及 $p, q \in Q_x$ 我们定义 $p \preccurlyeq_x q$ 当且仅当 $p \sim_{x-1} q$（即，$\mu_{x-1}(q) = \mu_{x-1}(p)$）且 $L_x(p) \subseteq L_x(q)$。

▶ **定义 4.4**。 若对于每个 分层自动机 都是 中心化的，则称其为 $x \geq 2$，并且对于所有状态p，$q \in Q_x$，若 $p \preccurlyeq_x q$则 p，q属于同一个 x-强连通分量。

直观地说，如果 $p \preccurlyeq_x q$ 属于不同的 x-强连通分量，那么 x-强连通分量 p 及其所有后代都可以被移除。由于 $p \preccurlyeq_x q$，这种移除不会改变其父强连通分量中词被强接受/拒绝的情况，其中父强连通分量是 p 和 q的父强连通分量。因此，可以移除这些状态，而不会改变该自动机的语言。

最后，在每一层上，我们都可以像对有限自动机那样进行类似的最小化构造，将该层上 x 在所有层上语言等价的状态 $y \leq x$合并。我们通过归纳定义 x 关系 $p \approx_x q$，对于 p，$q \in Q_{\geq x}$:

$p \approx_1 q$ 当且仅当 $L(p) = L(q)$，并且

$p \approx_x q$ 当且仅当 $p \approx_{x-1} q$ 并且 $L_x(p) = L_x(q)$（对于 $x > 1$）。

▶ **定义 4.5**。 一个 分层自动机 是 安全最小的，当对于所有 $x \geq 1$ 和状态p，$q \in Q_x$: 如果 $p \approx_x q$ 那么 $p = q$,

▶ 注 4.6。在一致分层自动机中，$\sim_x \subseteq \approx_x$ 对于所有 x均成立。这是因为 $p \sim_x q$意味着 $\widehat{\mu}_x(p) = \widehat{\mu}_x(q)$。根据一致性假设，$L(p) = L(q)$，并且对于 $1 < y \leq x$，等式 $L_y(p) = L_y(q)$ 显然成立。安全最小性表明，安全语言等价蕴含 $\sim_x$。也就是说，如果 p、$q \in Q_x$ 满足: 对于所有$y \leq x$，$L(p) = L(q)$ 且 $L_y(p) = L_y(q)$，那么 $p = q$。

4.2 中心序列

作为研究最小分层自动机的一项重要技术工具，我们引入状态 p的中心序列概念。它是一个有限词，能够唯一确定状态 p（将同步性质扩展自 [CO24，第 4.2 节。2] 以及完整版本 [LW25]中针对 coBüchi 自动机的中心序列概念）。正规、安全最小和中心化这三种性质刻画了最小分层自动机（如定理 4.1 所述），并且还保证每个状态都有一个中心序列（引理 4.7）。这是证明定理 4.1 的重要一步。

设$\mathcal{A}$ 是一个分层自动机，且 $p \in Q_x$ 是一个 x-状态。我们称 $z_p \in \Sigma^+$ 是中心序列，对应于 p，如果满足:

- $p \xrightarrow{z_p}_x p$ 在$\mathcal{T}_x$中；对于所有 $q \in Q_x$ 满足 $q \sim_{x-1} p$的情况，
- $q \xrightarrow{z_p}_x p$ 或 $q \xrightarrow{z_p}_x \bot$；对于所有 $q' \in Q_{x+1}$ 满足 $q' \sim_{x-1} p$
- 的情况，我们有 $q' \xrightarrow{z_p}_{x+1} \bot$。

在 $[\![\mathcal{A}]\!]$，中，这意味着 $p \overset{z_p}{\longrightarrow}_{:x} [p]_x$，并且从 $[p]_{x-1}$ 中的任何其他状态出发，要么$q \overset{z_p}{\longrightarrow}_{:x} [p]_x$，要么 $q \overset{z_p}{\longrightarrow}_{:<x}$，这意味着 z_p 会产生优先级 $< x$。注意，第三个条件是必要的，以确保相关运行的优先级不为 $> x$。

▶**引理 4.7**。 设 $\mathcal{A}$ 是一个 正规、 中心化 且 安全最小分层自动机。那么，对于每个 $x \geq 2$，每个状态 $p \in Q_x$ 都存在一个 中心序列。

证明。 设 $P_x = \{p_1, \ldots, p_k\}$ 是 x-状态 属于 $[p]_{x-1}$。

首先假设 p 在 P_x中是 $\preccurlyeq_x$-极大的。由 A 的安全最小性可知，不存在满足 $L_x(p') = L_x(p)$ 的 $p' \in P_x \setminus \{p\}$。由于 p 是 $\preccurlyeq_x$- 极大的，对于每个 $p' \in P_x \setminus \{p\}$，我们都可以找到 $v_{p'} \in L_x(p) \setminus L_x(p')$。由正规性可知，我们可以假设 $p \xrightarrow{v_{p'}}_x p$。

设 $u_0 \in \Sigma^+$ 使得 $p \xrightarrow{u_0}_x p$ 且 $\mu_x^{-1}(p) \xrightarrow{u_0}_{x+1} \perp$ (这由 正规性保证存在)。定义u_i 如下归纳定义：

- 如果 $p_i \xrightarrow{u_{i-1}}_x \perp$ 或 $p_i \xrightarrow{u_{i-1}}_x p$ ，则 $u_i = u_{i-1}$。如果 $p_i \xrightarrow{u_{i-1}}_x p' \neq p$，那
- 么我们取上面选定的 $v_{p'}$ 并定义 $u_i = u_{i-1}v_{p'}$。

不难验证 u_k 是一个中心序列对于 p。

现在假设 p 不是 $\preccurlyeq_x$-极大的。令 p' 为 $\preccurlyeq_x$-极大状态，在 P 中，且 $p \preccurlyeq_x p'$。由于该自动机是中心化的， p 和 p' 属于$\mathcal{T}_x$的同一个强连通分量。令 u， v 为满足 $p \xrightarrow{u}_x p' \xrightarrow{v}_x p$的词。由前一种情况可知， p' 存在一个中心序列 $z_{p'}$。不难验证$uz_{p'}v$ 是中心序列对于 p。◀

4.3 最小分层自动机的唯一性

现在我们可以证明定理 4.1。

▶**定理 4.1**。设$\mathcal{A}$ 和$\mathcal{B}$ 是两个等价的一致分层自动机，且它们是正规、中心且安全最小。那么，$\mathcal{A}$ 和$\mathcal{B}$ 是同构的（作为分层自动机）。

设$\mathcal{A}$ 和$\mathcal{B}$ 是两个一致分层自动机，对应于 L ，且它们是正规、中心且安全最小。我们要构造一个同构 φ: $\mathcal{A} \to \mathcal{B}$。这样的同构也可以等价地写成一列迁移系统的同构 φ_x : $\mathcal{T}_x^{\mathcal{A}} \to \mathcal{T}_{\mathcal{B}x}$，并且与 μ_x-函数相容。我们按照 x归纳地构造这一序列。

第一层 T A1 和 $\mathrm{T}_1^{\mathcal{B}}$ 都由 L 的剩余类组成，因此存在一个同构 $\varphi_1 : \mathrm{T\,A}_1 \to \mathrm{T}_1^{\mathcal{B}}$。现在假设我们已经构造了同构$\varphi_1, \ldots, \varphi_{x-1}$， 直至第 $x-1$ 层，并希望将其扩展至第 x 层。首先，我们证明，对于 A 中每个以 $\mu_{x-1}^{\mathcal{A}}(p) = r$ 为父节点的 x-状态 p ，B 中都存在一个以 $\varphi_{x-1}(r)$ 为父节点的 x-状态 q ，使得 q 和 p 具有相同的安全语言（命题 4.8）。根据安全最小性，这样的状态 q 是唯一的。然后我们定义 $\varphi_x(p) = q$。证明中两个自动机的作用是对称的，因此可知两个自动机在第 x 层都有对应于相同安全语言的状态。这意味着 φ_x 是 $\mathrm{T}_x^{\mathcal{A}}$ 与 T Bx 之间的一个同构（命题 4.9）。此外，根据定义，φ_x 与 μ_x 函数相容： $\varphi_{x-1}(\mu_{x-1}^{\mathcal{A}}(p)) = \mu_{x-1}^{\mathcal{B}}(\varphi_x(p))$。定理 4.1 得证。

▷ 命题 4.8。 对于每个 x状态 p ，属于 $\mathcal{A}$ 且父节点为 $\mu_{x-1}^{\mathcal{A}}(p) = r$的，都存在唯一一个 x 状态 q，属于 $\mathcal{B}$ 且父节点为 $\varphi_{x-1}(r)$，使得 $L_x{}^{\mathcal{A}}(p) = L_x{}^{\mathcal{B}}(q)$。

证明。根据安全最小性， q 的唯一性可直接得到。我们通过以下构造证明其存在性，该构造见图 6。设 p 是 A 中父节点为 r的一个 x-状态，并设 $\varphi_{x-1}(r) = s$ 是 B 的第 $x-1$ 层上与之对应的状态。设 z_p 是 p的一个中心序列。那么，从任意状态 $p' \sim_x p$开始在 A 中读取 z_p^ω 时， z_p^ω 会生成最小优先级 x 。此外，由于 z_p 在 p上循环，因此 z_p 在 r 上也循环；又由于 A 和 B 的 $x-1$ 层同构，所以 z_p 在 B 中的 s 上也循环。因此，从 B 中的 s 读取 z_p^ω时，它生成的最小优先级至少为 $x-1$。由于 r 和 s 语言等价，

必须存在某个子节点 q' 属于 s 使得 z_p^ω 是 x-安全的，从 q'开始。令 q 满足$q' \xrightarrow{z_p}_x q$。注意， q 的父节点也是 s，因为 z_p 在 s上循环。

令 z_q 为 q的中心序列，并令 w 满足 $q \xrightarrow{w}_x q'$（这是存在的，因为 B 是正规的）。注意， z_q 和 w 在 s 上循环，因此也在 r 上循环。 ω-词（$z_q w z_p$）$^\omega$ 从 B 中的 q 开始读取时，会产生优先级 x。由于 p 和 q 语言等价（因为它们的父节点语言等价），因此在 A 中必须能够从 r 的一个子节点，在（$z_q w z_p$）$^\omega$上产生优先级 $\geq x$。由于 z_p 是 p 的中心序列，这样的运行在第一个 $z_q w z_p$ 之后属于 p，因此（$z_q w z_p$）$^\omega$ 必须对 p 安全。令 p_1、 p_2 满足 $p \xrightarrow{z_q} p_1 \xrightarrow{w} p_2$。由于 z_p 是 p 的中心序列，我们得到 $p_2 \xrightarrow{z_p} p$。

我们断言 $L_x^{\mathcal{A}}(p) = L_x^{\mathcal{B}}(q)$。反设不然。第一种情况是存在$u \in L_x(p) \setminus L_x(q)$。由于 A 是正规的，我们可以选择 u，使其在 p上循环。因此，它也在 r 和 s上循环。于是， ω-词 ($z_q w z_p u)^\omega$ 从 p 产生优先级 x，并从 q产生优先级 $x-1$。后者成立，是因为优先级 $x-1$ 要么在 z_q上产生；若没有，则 z_q 之后的状态为 q，因此 $z_q w z_p$之后的状态也如此，从而优先级 $x-1$ 在 u上产生。由于 p和 q 语言等价，这就构成了矛盾。

第二种情况是存在 $v \in L_x(q) \setminus L_x(p)$。于是我们得到 $(z_q w z_p v)^\omega$ 产生优先级 x，来自 q，以及来自 优先级 $x-1$，来自 p，论证方式与第一种情况相同。◁

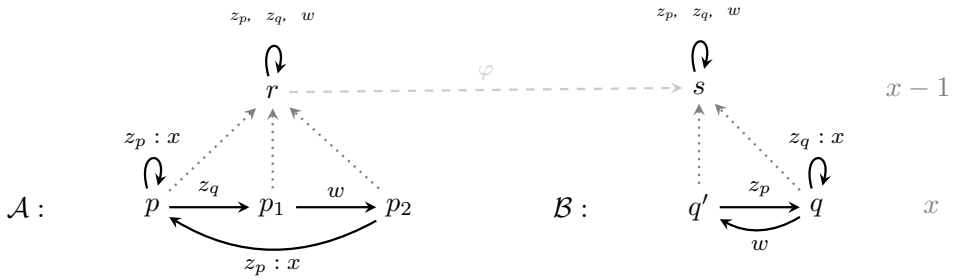


图 6 命题 4.8 的证明中同构构造的示意图。

令 $\varphi_x(p) = q$，对于 q 唯一的 x-状态，在 $\mathcal{B}$ 中满足 $\varphi_{x-1}(\mu_{x-1}^{\mathcal{A}}(p)) = \mu_{x-1}^{\mathcal{B}}(q)$ 以及 $L_x{}^{\mathcal{A}}(p) = L_x{}^{\mathcal{B}}(q)$。

▷ 命题 4.9。 函数 φ_x 是一个 迁移系统的同构。

证明。我们首先证明 φ_x 是一个态射。设 $p \xrightarrow{a}_x p'$ 属于 A 且 $q = \varphi_x(p)$。由于$L_x{}^{\mathcal{A}}(p) = L_x{}^{\mathcal{B}}(q)$，存在一条迁移 $q \xrightarrow{a}_x q'$ 属于 B，并且 $L_x{}^{\mathcal{A}}(p') = L_x{}^{\mathcal{B}}(q')$。若 r 是 p 的父节点，则对于某个 r' 属于 A，有 $r \xrightarrow{a} r'$。根据 $\varphi_x(p)$的定义， $q = \varphi_x(p)$的父节点是 $\varphi_{x-1}(r)$。根据同构， $\varphi_{x-1}(r) \xrightarrow{a} \varphi_{x-1}(r')$，因此 $\varphi_{x-1}(r')$是 q'的父节点。所以根据定义，我们有 $\varphi_x(p') = q'$。因此， φ_x 是一个态射，证毕。

由命题 4.8， φ_x 是一个双射，且函数 φ_x^{-1}也由对称性可知是态射。

◁

5 多项式时间内的最小化

本节介绍一种用于一致分层自动机的最小化过程。更准确地说，该过程构造出定理 4.1 中出现的唯一最小分层自动机。该过程证明，每个一致分层自动机都包含一个

子自动机，且存在一个到该最小自动机的满射态射，从而证明其典范性。我们通过分层自动机的态射形式化这些概念，并在第 5.1 节中对其进行介绍。

▶**定理 5.1**。 设 $\mathcal{A}$ 是一个 一致分层自动机。我们可以在多项式时间内构造一个等价的一致分层自动机$\mathcal{A}_L$，该自动机是 正规、中心 且 安全最小。
此外，$\mathcal{A}$ 包含一个 子自动机[8]，该子自动机允许存在一个满射 强态射 到$\mathcal{A}_L$。特别地，$\mathcal{A}_L$ 的状态数和 叶状态 数都不多于 $\mathcal{A}$。

结合定理 4.1，这为 正规、中心且安全最小一致分层自动机的最小性和典范性提供了结论。

▶**推论 5.2**。 设 $\mathcal{A}_L$ 是一个 一致分层自动机，且该自动机是 正规、中心 且 安全最小。那么，任意等价的 一致分层自动机 $\mathcal{A}$ 都包含一个子自动机，该子自动机允许存在一个满射 强态射 到$\mathcal{A}_L$。因此，$\mathcal{A}$ 的状态数至少与 $\mathcal{A}_L$一样多，并且 $[\![\mathcal{A}]\!]$ 的状态数至少与 $[\![\mathcal{A}_L]\!]$一样多。

本节余下部分将致力于证明定理 5.1。我们首先介绍态射，并证明一些用于说明最小化过程正确性的技术性引理。

5.1 分层自动机的态射

两个 分层自动机 A 和 A′ 之间的态射是一个函数 $\varphi\colon\ Q_{\geq 1} \to Q'_{\geq 1}$ 它保持 迁移系统 以及树表示中的 祖先 关系。即：

- $\varphi(q_{\mathsf{init}}) = q'_{\mathsf{init}}$，如果 $q \xrightarrow{a}_x q'$ 在 T_x 中，则对于 $y = 层(\varphi(q))$，$\varphi(q) \xrightarrow{a}_y \varphi(q')$ 在 T'_y 中；并且如果 p 是 A 中 q 的祖先，则 $\varphi(p)$ 是 A′ 中 $\varphi(q)$ 的祖先。

如果此外它还保持非转移，我们称其为 强态射，即：

- if $q \xrightarrow{a}_x \bot$, then $\varphi(q) \xrightarrow{a}_y \bot$.

请注意，尽管状态可以通过 φ 改变层，但这些条件施加了一定的刚性。特别是，如果 $q \in Q_x$ 被映射到 $\mathcal{T}_y$，那么从 $\mathcal{T}_x$ 中的 q 可达的所有状态也都会被映射到层 y。特别地，根据我们的可达性假设，$\mathcal{T}_1$ 中的所有状态都会被映射到 $\mathcal{T}_1$。
一个分层自动机的态射 φ 是同构，当且仅当它是双射且 φ^{-1} 也是一个态射。注意，在这种情况下，它保持层不变，也就是说，$\varphi|_{Q_x}$ 是一个从迁移系统$\mathcal{T}_x$ 到$\mathcal{T}'_x$的同构。

子自动机。一个分层自动机$\mathcal{A}'$ 是子自动机$\mathcal{A}$，如果存在一个单射态射 $\varphi\colon\ \mathcal{A}' \to \mathcal{A}$。

注意，$\mathcal{A}'$ 不仅可以通过从$\mathcal{A}$的某些层中移除状态来得到，也可以通过将某些强连通分量从一层移到另一层来得到。

[8] 我们注意到，这个子自动机可能不一致。

保持强接受的态射。在接下来的章节中，我们从一个一致分层自动机$\mathcal{A}$ 开始，并对其进行调整，以得到修改后的自动机 $\mathcal{A}'$。除一种修改外，我们都通过$\mathcal{A}'$，即移除 $\mathcal{A}$ 的状态或转移，使得存在一个单射$\mathcal{A}' \hookrightarrow \mathcal{A}$。在三种情形中的两种情形下，引理 5.3 将确保此操作的正确性。在安全最小化步骤（第 5.3 节）中，$\mathcal{A}'$ 通过合并 $\mathcal{A}$的状态得到，从而产生一个满射 $\mathcal{A} \twoheadrightarrow \mathcal{A}'$。

我们称一个 分层自动机的态射 $\varphi : \mathcal{A}' \to \mathcal{A}$ 保持强接受 当且仅当对于每个 $p' \in Q'_{\geq 1}$ 以及每个 $w \in \Sigma^\omega$ 都有：

- 若 w 被 强接受 于 p'，则 w 被 强接受于 $\varphi(p')$；若 w 被 强拒绝
- 于 p'，则 w 被 强拒绝于 $\varphi(p')$。

▶ **引理 5.3.** 设 $\varphi : \mathcal{A}' \to \mathcal{A}$ 是一个 保持强接受的态射。若 $\mathcal{A}$ 是一致的，则 $\mathcal{A}'$ 是 一致的，且 $L(\mathcal{A}') = L(\mathcal{A})$。

证明。取 $p'_1 \sim_1 p'_2$ 属于 A′。为导出矛盾，假设 w 被p'_1 强接受且被 p'_2强拒绝。根据态射性质，有 $\varphi(p_1) \sim_1 \varphi(p_2)$。于是，由于 φ 保持强接受，w 被 $\varphi(p'_1)$强接受，且被 $\varphi(p'_2)$强拒绝。这与 A 的一致性矛盾。

如果 w 被 A′ 接受，那么根据引理 3.15，存在一个状态 p 以及一个分解$uw' = w$，使得 $q'_{\text{init}} \xrightarrow{u} \widehat{\mu}_1(p)$，且 w' 从 p 强接受。于是，在 A 中有 $q_{\text{init}} \xrightarrow{u} \widehat{\mu}_1(\varphi(p))$，并且 w' 从 $\varphi(p)$ 强接受。根据 A′ 的一致性和引理 3.15，w 被 A′ 接受。当 w 被 A′ 拒绝时，论证类似。◀

5.2 规范化

令 $\mathcal{A}$ 为一个 一致分层自动机。我们分两步构造一个与之等价、同时满足 N1 和 N2的自动机。首先，通过分离彼此之间存在转移的强连通分量来确保满足 N1 。然后，在不影响N2 的情况下，建立N1。

5.2.1 分离强连通分量

为了确保性质 N1，我们将每个 $\mathcal{T}_x$ 分离为互不连通的非平凡强连通分量。固定 $x > 1$，并假设在 $\mathcal{T}_x$ 中存在一个转移 $p \xrightarrow{a}_x q$，它属于 $\mathcal{T}_x$，且其两个状态不属于同一个强连通分量。（这种转移可通过执行强连通分量分解，在 [Tar72]中以线性时间检测出来。）此操作只需从 $p \xrightarrow{a}_x q$中移除该转移，并从 $\mathcal{T}_x$ 中移除经由 $\widehat{\mu}_x$ 映射到该转移的所有转移，针对 $\mathcal{T}_y$ 中的 $y > x$即可。我们还要移除可能因此产生的平凡强连通分量（即 $\mathcal{T}_y$ 中没有出边的状态），直到结果中不再存在此类状态。将得到的自动机记为 $\mathcal{A}^{sep}$。

▶**引理 5.4**。设 $\mathcal{A}$ 是一个一致分层自动机。那么，$\mathcal{A}^{sep}$ 是一致的。此外，$L(\mathcal{A}^{sep}) = L(\mathcal{A})$，并且 $\mathcal{A}^{sep}$ 是 $\mathcal{A}$ 的子自动机。

证明。我们证明映射 φ：$\mathcal{A}^{sep} \to \mathcal{A}$ 由 $\varphi(p) = p$ 是一个保持强接受的单射态射。由引理 5.3 即得结论。显然，φ 是一个态射，且是单射。

设 w 被 p 在 A^{sep} 中强接受（强拒绝的情形是对称的）。特别地，$w \in L_x^{\mathcal{A}^{sep}}(p)$，因此根据态射性质，$w \in L_x^{\mathcal{A}}(p)$。反设 w 未被 p 在 A 中强接受，也就是说，存在一个分解 $w = uw'$ 和状态 $q \in Q_{x+1}$，使得在 A 中有 $p \xrightarrow{u}_x \widehat{\mu}_x(q)$，且 $w' \in L_{x+1}^{\mathcal{A}}(q)$。令 $w' = u'w''$ 满足：运行 $q \xrightarrow{u'}_{x+1} q' \xrightarrow{w''}_{x+1}$ 在后缀 w''期间不改变其强连通分量。于是，$w'' \in L_{x+1}^{\mathcal{A}^{sep}}(q')$。

因此，分解 $w = uu' \cdot w''$ 见证了 w 未被 强接受 p 在$\mathcal{A}^{sep}$，这就产生了矛盾。◀

应用此操作后，我们至少会移除一条转移。因此，重复此操作至多 $|Q_{\geq 1}|^2$ 次，我们便可得到一个等价的一致子自动机，满足N1。

5.2.2 降低

我们假设 A 满足性质 N1，并在不影响 N1 的情况下使其满足 N2。假设 N2 对层级 x 上的状态 q 不成立。这意味着 q 有一个子节点 p，使得对于每个满足 $q \xrightarrow{u}_x$ 有定义的 $u \in \Sigma^*$， $p \xrightarrow{u}_{x+1}$ 也有定义。（我们可以在对数空间内检测是否存在这样一对父节点和子节点，因为这只需判断确定性安全自动机是否等价。）取 S_{x+1}，即 T_{x+1} 中 p 的强连通分量，以及 S_x，即 T_x 中 $q = \mu_x(p)$ 的强连通分量。我们有 S_x 被 S_{x+1} 覆盖，也就是说，对于每个$p' \in S_{x+1}$，只要存在转移 $\mu_x(p') \xrightarrow{a}_x$，就也有 $p' \xrightarrow{a}_{x+1}$。在 $[\![\mathrm{A}]\!]$ 中，这意味着映射到 S_{x+1} 的状态没有优先级为 x 的出边；若$\widehat{\mu}_{x+1}(p') \in S_{x+1}$，则不存在任何 a 使得 $[\![\mathrm{A}]\!]$ 中有转移 $p' \overset{a}{:\!\!\longrightarrow}_x$。我们将说明如何消除层级 x 上这样一个被覆盖的强连通分量 S_x。

这里有一个简单的引理，说明较高层级的强连通分量包含在较低层级的强连通分量中。

▶**引理 5.5**。 设 $S \subseteq Q_x$ 是 T_x中的一个强连通分量，且 $S' \subseteq Q_{x+1}$ 是 T_{x+1}中的一个强连通分量。如果$\mu_x(S') \cap S \neq \emptyset$，那么 $\mu_x(S') \subseteq S$。

证明。 这是因为 μ_x 保持路径不变。 ◀

假设 A 满足 N1，并固定 S_x, S_{x+1}，使得 S_x 如上所述被 S_{x+1} 覆盖。对于每个 $y \geq x+2$，令 S_y 为 T_y 中由映射到S_{x+1}的 T_y 状态所诱导的部分，即 $\{p \in \mathrm{T}_y : \widehat{\mu}_{x+1}(p) \in S_{x+1}\}$。类似地，对于 $y \geq x+1$，令 Z_y 为 T_y 中映射到 S_x的部分，也就是由状态 $\{p \in Q_{\geq x} : \widehat{\mu}_x(p) \in S_x\}$诱导的部分。我们需要区分映射到 S_x 和映射到 S_{x+1}的状态，因为即使 S_x 被 S_{x+1}覆盖，也可能存在映射到 S_x、但位于 S_{x+1} 之外的状态。注意，根据引理 5.5，每个 S_y 和Z_y 都是 T_y中的一组强连通分量。

▶ 例 5.6。在给出降低操作的一般定义之前，先考虑图 7 左侧所示的自动机。在其中有几个候选强连通分量可以应用降低操作。对于本例，我们选择 $S_x = \{p_2\}$ 和 $S_{x+1} = \{p_3\}$，因此 $x = 2$ 在本例中成立。注意，这两个状态都在 a 上循环，b，因此 S_x 被 S_{x+1}覆盖。映射到 S_{x+1} 的强连通分量集合（通过 $\widehat{\mu}_3$）为 $S_4 = \{p_4, q_4\}$ 和$S_5 = \{q_5\}$；映射到 S_x 的强连通分量集合（通过 $\widehat{\mu}_2$）为 $Z_3 = \{p_3, r_3\}$、$Z_4 = \{p_4, q_4, r_4\}$ 和 $Z_5 = \{q_5, r_5\}$。自动机 A^{low} 的构造方式是：从层 y中移除 Z_y，并将 S_{y+2} 添加到层 y，对每个 $y \geq x$均如此。结果如图 7 右侧所示。除 S_{x+2}中的状态外，所有状态的父关系都得以保留；在本例中，S_4。 p_4 和 q_5 的新父节点现在分别是 $\widehat{\mu}_{x-1}(p_4)$ 和 $\widehat{\mu}_{x-1}(q_5)$，二者均为 p_1。这对应于下面 μ^{low}_{y-1} 定义中的第三种情况。 q_5 的父节点仍然是 q_4，但由于这些状态改变了层次，与 μ相比，μ^{low}_{y-1} 的定义需要调整索引。因此我们得到 $\mu^{low}_2(q_5) = \mu_4(q_5) = q_4$。对于 $y = 3$，这是下面 μ^{low}_{y-1} 定义中的第二种情况。未改变层次且未被删除的状态（即位于 Q^{low}_y 和 Q_y中的状态）的父节点直接复制，例如本例中的 q_2：$\mu^{low}_1(q_2) = \mu_1(q_2) = p_1$。这是下面 μ^{low}_{y-1} 定义中的第一种情况。

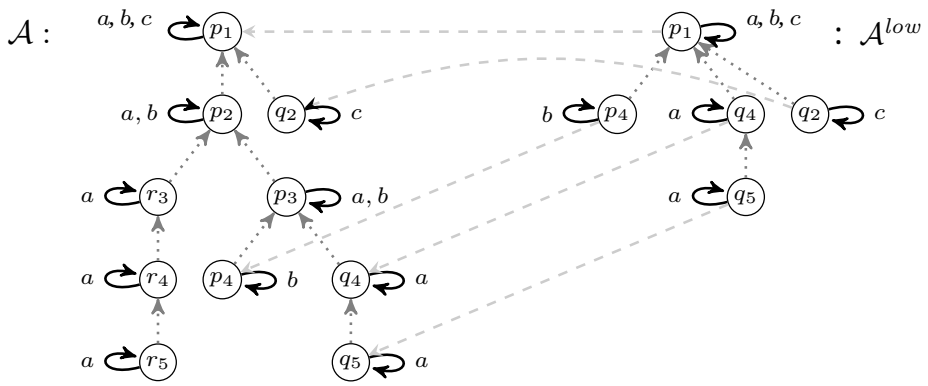


图 7 用于建立 N2的降低操作示意图，取自例 5.6。

通过降低某些分量得到一个新的分层自动机 $\mathcal{A}^{low}$ 。

- $\mathcal{T}_y^{low} = \begin{cases} (\mathcal{T}_y - Z_y) \cup S_{y+2} & y \geq x, \\ \mathcal{T}_y & \text{otherwise.} \end{cases}$
- For $p \in \mathcal{T}_y^{low}$, $y \geq 2$ we set $\mu_{y-1}^{low}(p) = \begin{cases} \mu_{y-1}(p) & \text{if } p \in Q_y, \\ \mu_{y+1}(p) & \text{if } p \in S_{y+2},\ y > x, \\ \widehat{\mu}_{x-1}(p) & \text{if } p \in S_{x+2}. \end{cases}$

也就是说，$\mathcal{T}_y$ 中以 S_x 为祖先的强连通分量（映射到 S_x 的那些强连通分量 $\widehat{\mu}_x$）会被 $\mathcal{T}_{y+2}$中的强连通分量覆盖。因此， S_x 会被替换为 S_{x+2}；而被替换的不仅有 S_{x+1} ，还有层 $x+1$ 上所有映射到 S_x 的强连通分量，它们会被替换为 S_{x+3}。就 $[\![\mathcal{A}^{low}]\!]$ 而言，这意味着，从映射到 优先级 的状态出发的转移，其优先级会降低 S_x 个 2，并且会移除一些状态。

▶**引理5.7.** 设 $\mathcal{A}$ 是一个满足 *N1* 的一致分层自动机。那么，$\mathcal{A}^{low}$ 是一致的，并且满足 *N1*。此外， $L(\mathcal{A}^{low}) = L(\mathcal{A})$，并且 $\mathcal{A}^{low}$ 是 的子自动机 $\mathcal{A}$。

证明。 自动机 $\mathcal{A}^{low}$ 满足 N1，因为没有添加强连通分量之间的转移。

要验证 A^{low} 是 A 的子自动机，我们需要证明恒等映射 φ: A^{low} →A 是一个单射态射。显然，若 $p \xrightarrow{a}_y q$ 属于 A^{low} ，那么 $\varphi(p) \xrightarrow{a}_z \varphi(q)$ 属于 A，其中 $z = y+2$ 或 $z = y$ ，具体取决于 $p \in S_{y+2}$ 是否成立。保持祖先关系这一性质由 μ_{y-1}^{low}的定义可知。这是因为 A 满足 N1，且 A^{low} 是通过移除整个强连通分量得到的。注意，若 $p \in S_{x+2}$ ，则 p 位于 A^{low} 的第 x 层，位于 A 的第 $x+2$ 层，而 $\varphi(\mu_{x-1}(p))$ 位于 A 的第 $x-1$ 层。因此， φ 保持祖先关系，但不保持父关系。

我们命题如下：如果 w 在强接受自 p 的第 y 层上的$\mathcal{A}^{low}$ ，那么存在一个分解 $uw' = w$ 和一个状态 q ，使得 $p \xrightarrow{u}_z \widehat{\mu}_z(q)$ ，并且 w' 从 q 在$\mathcal{A}$上被强接受；此处 $z = y+2$ 或 $z = y$ 取决于 $p \in S_y$ 是否如此。将接受/拒绝的角色互换后，我们也有类似陈述。

为证明该命题，取 w ，即强接受或强拒绝，从 p 位于层 y的$\mathcal{A}^{low}$。如果 $p \in S_{y+2}$ ，那么在 w 上的整个运行，从 p 都属于 S_{y+2} ，因为 S_{y+2} 根据引理 5.5 是强连通分量的并集。但这样一来，层 $y+1$ 的$\mathcal{A}^{low}$ 映射到 S_{y+2} 的所有状态都属于 S_{y+3}。因此， w 从 p 的强接受/拒绝在$\mathcal{A}$中也成立。因此，该命题对

平凡分解 $u = \varepsilon$、$w' = w$ 和 $q = p$成立。如果 $p \in \mathcal{T}_y$ 对于 $y \neq x-1$，则论证相同，因为在 w 上的运行始终停留在$\mathcal{T}_y$中。

完成命题证明所需处理的剩余情形是 $p \in \mathrm{T}^{low}_{x-1}$。为便于书写，假设 $x-1$ 为偶数，因此 w 从 p 在 A^{low} 中强接受。另一种情形相同。假设 w 没有从 p 在 A 中强接受。这意味着存在一个分解 $uw' = w$，以及层 x 上的一个状态 q，使得在 A 中有 $p \xrightarrow{u}_{x-1} \mu_{x-1}(q)$ 和 $w' \in L_x(q)$。显然，$q \in S_x$；否则，w 就不会从 p 在 A^{low} 中强接受。由于 S_x 被 S_{x+1} 覆盖，因此存在 S_{x+1} 中的 q'，使得 $\mu_x(q') = q$、$w' \in L_{x+1}(q')$。如果 w' 从 q' 强接受，那么我们就证明了命题，因为 $p \xrightarrow{u}_{x-1} \widehat{\mu}_{x-1}(q')$。否则，存在一个分解 $u'w'' = w'$，以及层 $x+2$ 上的一个状态 q''，使得 $q' \xrightarrow{u'}_{x+1} \mu_{x+1}(q'')$ 且 $w'' \in L_{x+2}(q')$。但此时在 A^{low} 中有 $p \xrightarrow{u}_{x-1} \mu_{x-1}(q) \xrightarrow{u'} \widehat{\mu}_{x-1}(q'')$，因为 A 和 A^{low} 的 $x-1$ 层相同；并且有 $w'' \in L_x(\mu^{low}_x(q''))$，因为 $q'' \in S_{x+2}$。此外，根据定义，$\widehat{\mu}_{x-1}(q'') = \mu^{low}_{x-1}(q'')$。这与 w 从 p 在 A^{low} 中强接受相矛盾。

借助我们的命题，可以证明 A^{low} 是一致的。为此，反设存在 $p_1 \sim p_2$ 和 w，使得 w 从 p_1 出发在 A^{low}中被强接受，而从 p_2 出发被强拒绝。根据我们的命题，存在状态 q_1、q_2 以及分解$u_1w_1 = u_2w_2 = w$，使得 $p_1 \xrightarrow{u_1}_1 \widehat{\mu}_{y_2}(q_1)$、$p_2 \xrightarrow{u_2}_1 \widehat{\mu}_{y_2}(q_2)$，并且 w_1 从 q_1 出发在 A 中被强接受，w_2 从 q_2 出发在 A 中被强拒绝。由于 A 是一致的，根据推论 3.18，这意味着 $u_1w_1 = w$ 被 A 接受，而 $u_2w_2 = w$ 被 A 拒绝。矛盾。

最后，我们证明 $L(\mathrm{A}) = L(\mathrm{A}^{low})$。如果 $w \in L(\mathrm{A}^{low})$，那么根据推论 3.18，存在一个分解 $uw' = w$ 和一个状态 p，使得 $q_{\mathsf{init}} \xrightarrow{u}_1 \widehat{\mu}_1(p)$，并且 w' 从 p 出发在 A^{low}中被强接受。由我们的命题可知，存在一个分解 $u'w'' = w'$ 和 A 中的一个状态 q，使得 $q_{\mathsf{init}} \xrightarrow{uu'}_1 \widehat{\mu}_1(q)$，并且 w'' 从 q 出发在 A 中被强接受。因此，根据推论 3.18，$uu'w'' = w$ 被 A 接受。$w \notin L(\mathrm{A}^{low})$ 的情形同理。◀

每次执行降低操作后，我们都会移除整个强连通分量，这是由引理 5.5 得出的，因此 N1性质仍然成立。当不存在 覆盖 强连通分量时，我们得到一个等价的 子自动机，它处于标准形式，即同时满足 N1 和 N2。显然，可以在多项式时间内找到并移除一个被覆盖的强连通分量。

5.3 安全最小化

我们将展示如何在保持规范化的同时，对一个分层自动机进行安全最小化。回顾一下，$p \approx_x q$ 若 p 和 q 语言等价，且 $L_y(p) = L_y(q)$ 对所有 $y \leq x$均成立，并且当安全最小当且仅当 $\sim_x$ 与 $\approx_x$ 一致。首先，根据定义可直接得到，$\approx_x$ 关系是 T_x中的一个双模拟。

▶ **引理 5.8**。设 p，$p' \in Q_x$ 使得 $p \approx_x p'$。若 $p \xrightarrow{a}_x q$ 则 $p' \xrightarrow{a}_x q'$ 且 $q' \approx_x q$。

所需的安全最小自动机$\mathcal{A}^{\mathrm{smin}}$ 是通过使用 $\approx_x$等价类作为 x-状态得到的自动机。下文使用记号 $[q]_{\approx_x}$时，我们假设 $q \in Q_x$。形式上，$\mathcal{A}^{\mathrm{smin}}$ 定义如下：

- $\mathcal{T}^{\mathrm{smin}}_x = (Q^{\mathrm{smin}}_x, \delta^{\mathrm{smin}}_x)$ with
  - $Q^{\mathrm{smin}}_x = \{[q]_{\approx_x} \mid q \in Q_x\}$,
  - $\delta^{\mathrm{smin}}_x([q]_{\approx_x}, a) = \begin{cases} [\delta_x(q,a)]_{\approx_x} & \text{if } \delta_x(q,a) \text{ defined,} \\ \bot & \text{else} \end{cases}$

- 初始状态为 $[q_0]_{\approx_1}$，其中 q_0 是$\mathcal{T}_1$中的初始状态。
- $\mu_x^{\mathrm{smin}}([q]_{\approx_{x+1}}) = [\mu_x(q)]_{\approx_x}$.

转移函数 δ^{smin} 由引理 5.8 良好定义，且 μ_x^{smin}定义良好，因为 $q \approx_{x+1} q'$ 蕴含 $q \approx_x q'$ 根据定义成立，因此 $\mu_x(q) \approx_x \mu_x(q')$。直接应用定义，并利用 μ_x 是一个态射这一事实，可以很容易地证明 μ_x^{smin}是一个态射。引理 5.8 蕴含 $L_x{}^{\mathcal{A}}(p) = L_x{}^{\mathcal{A}^{\mathrm{smin}}}([p]_{\approx_x})$。因此，$\mathcal{A}^{\mathrm{smin}}$ 是安全最小的。此外，我们已经确立的其他性质不会因这一构造而失效，下一条引理将对此加以说明。

▶**引理 5.9**。 若 $\mathcal{A}$ 是 正规的，则 $\mathcal{A}^{\mathrm{smin}}$也如此。

证明。 我们检查 N1和 N2这两个性质对 $x \geq 2$是否成立。

N1。 设 $[q]_{\approx_x} \xrightarrow{u}_x [q']_{\approx_x}$ 属于 $\mathcal{T}_{\mathrm{smin}x}$。那么，可以假设代表元 q, q' 的选取满足 $q \xrightarrow{u}_x q'$ 属于 $\mathcal{T}_x$。因此 $q' \xrightarrow{v}_x q$ 对于某个 $v \in \Sigma^+$成立，因为 $\mathcal{A}$ 满足 N1，故 $[q']_{\approx_x} \xrightarrow{v}_x [q]_{\approx_x}$。

N2. 设 $[q]_{\approx_x}$ 是 $\mathrm{A}^{\mathrm{smin}}$中的一个状态，$[p]_{\approx_{x+1}}$ 是它的一个子节点。那么，我们可以假设代表元的选取满足 $\mu_x(p) = q$ 属于 A。由于 A 满足 N2，存在 u，使得 $q \xrightarrow{u}_x$ 在 T_x 中有定义，但在 T_{x+1}中 $p \xrightarrow{u}_{x+1} \perp$。于是，$[q]_{\approx_x} \xrightarrow{u}_x$ 在 $\mathrm{T}_x^{\mathrm{smin}}$中有定义，但在 $\mathrm{T}_{x+1}^{\mathrm{smin}}$中 $[p]_{\approx_x} \xrightarrow{u}_{x+1} \perp$。◀

下面的引理给出了确保$\mathcal{A}^{\mathrm{smin}}$正确性的关键性质。

▶ **引理 5.10**。设$\mathcal{A}$ 是一个一致分层自动机，且 $p \in Q_x$ 是一个 x-状态。若一个词$w \in \Sigma^\omega$ 被强接受（分别被拒绝） $[p]_{\approx_x}$ 在$\mathcal{A}^{\mathrm{smin}}$中，则它也被强接受（分别被拒绝） p在$\mathcal{A}$中。

证明。假设 w 被 $[p]_{\approx_x}$ 在 $\mathrm{A}^{\mathrm{smin}}$ 中强接受（因此 x 为偶数）。特别地，$w \in L_x^{\mathcal{A}^{\mathrm{smin}}}([p]_{\approx_x})$，从而 $w \in L_x^{\mathcal{A}}(p)$。反设 w 未被 p 强接受。那么，存在一个分解 $w = uw'$ 和一个状态 $q' \in Q_{x+1}$，使得 $p \xrightarrow{u}_x q$，其中 $q = \widehat{\mu}_x(q')$，且 $w' \in L_{x+1}^{\mathcal{A}}(q')$。我们断言，分解 $w = uw'$ 见证了 w 未被 $[p]_{\approx_x}$ 在 $\mathrm{A}^{\mathrm{smin}}$ 中强接受，这就产生了矛盾。事实上，根据 δ_x^{smin} 的定义，我们有 $[p]_{\approx_x} \xrightarrow{u}_x [q]_{\approx_x}$，并且 $w' \in L_{x+1}^{\mathcal{A}^{\mathrm{smin}}}([q']_{\approx_{x+1}})$。由于 $\mu_x^{\mathrm{smin}}([q']_{\approx_{x+1}}) = [q]_{\approx_x}$，证明得出。◀

我们现在可以证明该构造的正确性：$\mathcal{A}^{\mathrm{smin}}$。

▶**引理 5.11**。 设 $\mathcal{A}$ 是一个正规且一致分层自动机。那么，$\mathcal{A}^{\mathrm{smin}}$ 是安全最小的、正规的且一致分层自动机。此外，$L(\mathcal{A}) = L(\mathcal{A}^{\mathrm{smin}})$，并且$\mathcal{A}$存在一个满射强态射指向$\mathcal{A}^{\mathrm{smin}}$。

证明。 由引理 5.8，映射 φ：$\mathcal{A} \to \mathcal{A}^{\mathrm{smin}}$，其中 $\varphi(p) = [p]_{\approx_x}$ 对于 $q \in Q_x$ 是一个强态射，显然它是满射。自动机$\mathcal{A}^{\mathrm{smin}}$ 根据引理 5.9是正规的。

我们证明 $\mathrm{A}^{\mathrm{smin}}$的一致性。设 $[p]_{\approx_x}$ 和 $[p']_{\approx_{x'}}$ 在 $\mathrm{A}^{\mathrm{smin}}$中 $\sim_1$-等价，也就是说，$L(\mathrm{A}\ p) = L(\mathrm{A}\ p')$。反设有一个词 w 在 $\mathrm{A}^{\mathrm{smin}}$中被 $[p]_{\approx_x}$强接受且被 $[p']_{\approx_{x'}}$ 强拒绝。由引理 5.10，w 在 A 中被 p 强接受且被 p' 强拒绝。因此，根据引理 3.14，$w \in L(\mathrm{A}\ p)$ 和 $w \notin L(\mathrm{A}\ p')$，这与这些状态的语言等价相矛盾。

我们证明 $L(\mathrm{A}^{\mathrm{smin}}) = L(\mathrm{A})$。设 w 是 $\mathrm{A}^{\mathrm{smin}}$ 接受的一个词（对于 w拒绝的情况，证明类似），也就是说，存在一个 x 偶数以及一个分解 $w = uw'$，使得 $[q_{\mathrm{init}}]_{\approx_1} \xrightarrow{u}_1 [q]_{\approx_1}$，并且 w' 被 $[q']_{\approx_x}$强接受，其中 $\widehat{\mu}_1(q') \approx_1 q$。根据引理 5.10，$w'$ 是强

接受的，由 q' 中的$\mathcal{A}$接受，因此根据引理 3.14，$w' \in L(\mathcal{A}\ q') = u^{-1}L(\mathcal{A})$，其中最后一个等式来自于A 的一致语义确定性$\mathcal{A}$。

最后，安全最小性的$\mathcal{A}^{\text{smin}}$ 由语言等价以及以下事实可知：$L_x(p) = L_x^{\mathcal{A}^{\text{smin}}}$ $([p]_{\approx_x})$。

◀

▶**引理 5.12**。 给定一个一致分层自动机$\mathcal{A}$，我们可以在$\mathcal{A}^{\text{smin}}$ 多项式时间内计算。

证明。 为构造$\mathcal{A}^{\text{smin}}$，我们只需计算 $\approx_x$ 等价类$\mathcal{A}$，因为转移和态射函数直接定义在这些等价类上。要计算 $\approx_1$的等价类，我们需要检查状态的语言等价性，这可根据命题 3.28 在多项式时间内完成。对于 $\approx_x$，$x > 1$ 的计算包括检查确定性安全自动机的等价关系，这可在对数空间内完成。◀

5.4 中心化

现在我们展示如何将自动机中心化，同时保持规范化和安全最小性。以下引理表明，我们可以扩展 $\preccurlyeq_x$ 关系以比较x-强连通分量。

▶ **引理 5.13.** 设 $\mathcal{A}$ 是一个一致分层自动机 满足 $N1$。设 S_1，S_2 是x中的两个$\mathcal{T}_x$-强连通分量 $p_1 \in S_1$，且存在 $p_2 \in S_2$ 使得 $p_1 \preccurlyeq_x p_2$，那么对于每个 $q_1 \in S_1$，都存在 $q_2 \in S_2$ 满足 $q_1 \preccurlyeq_x q_2$。

证明。取一条路径 $p_1 \xrightarrow{u}_x q_1$ in T_x。由于 $p_1 \preccurlyeq_x p_2$, $u \in L_x(p_2)$。设 q_2 是沿路径 $p_2 \xrightarrow{u}_x q_2$在 T_x 中到达的状态。由于 A 满足 N1，$q_2 \in S_2$。显然，$q_1 \sim_{x-1} q_2$，因为 $p_1 \sim_{x-1} p_2$。最后，$L_x(q_1) = u^{-1}L_x(p_1) \subseteq u^{-1}L_x(p_2) = L_x(q_2)$，因此 $q_1 \preccurlyeq_x q_2$。◀

我们记作 $S \preccurlyeq_x S'$，如果存在 $p \in S$ 和 $p' \in S'$，使得 $p \preccurlyeq_x p'$。

我们定义一种中心化操作，用于移除 x-强连通分量中不具备 $\preccurlyeq_x$-极大性的分量。固定$x \geq 2$ 奇数和两个分量 $S \preccurlyeq_x S'$。令$\mathcal{A}^{ctl}$ 为移除了 S 的分层自动机。这意味着对于每个 $y \geq x$，我们移除所有 $p \in Q_y$ 满足 $\widehat{\mu}_x(p) \in S$的对象。

▶**引理 5.14**。 设$\mathcal{A}$ 是一个安全最小、正规且一致分层自动机。那么，$\mathcal{A}^{ctl}$ 也具有安全最小、正规且一致的性质。此外，$L(\mathcal{A}^{ctl}) = L(\mathcal{A})$，并且$\mathcal{A}^{ctl}$ 是$\mathcal{A}$的子自动机。

证明。 由于$\mathcal{A}^{ctl}$ 是通过移除整个 x-强连通分量得到的，因此现存强连通分量之间不会增加转移，也不会有任何 x-强连通分量被覆盖于$\mathcal{A}^{ctl}$中。因此，$\mathcal{A}^{ctl}$ 也具有标准形式。

对于所有满足 $\approx_x$ 的关系 $x \geq 2$，关系 $\approx_x$ 在 $\mathcal{A}^{ctl}$ 是 $\approx_x$ 在新自动机状态集合上的限制，因为对于每个 p 属于 $\mathcal{A}^{ctl}$ 的状态，$L_x^{\mathcal{A}^{ctl}}(p) = L_x^{\mathcal{A}}(p)$。因此，该构造无法生成两个 $\approx_x$ 等价状态。

对于引理的第二部分，我们证明映射 $\varphi\colon \mathcal{A}^{ctl} \to \mathcal{A}$ 由下式给出：$\varphi(p) = p$ 是一个单射态射，并且保持强接受。由引理 5.3 即可得出结论。显然，φ 是一个态射且为单射。

为了证明 φ 保留强接受，假设 w 从 某个状态强接受 (或被拒绝)，且该状态位于 $\mathcal{A}^{ctl}$中。这里唯一有趣的情况是该状态是 S的父节点，因为对于所有其他状态，它们的强连通分量及其子节点的强连通分量都没有改变。由引理 5.5 可知，S 的所有父节点都处于同一个强连通分量中，因此令 $R \subseteq Q_{x-1}$ 为 S的父强连通分量。假设 $x-1$ 是偶数，另一种情况可对称地证明。令 w从 某个状态强接受，该状态属于 $r \in R$ 中的 。假设 w 未从 某个状态强接受，该状态属于 r

在 A 中。于是存在一个分解 $w = uw'$、一次运行 $r \xrightarrow{u}_{x-1} q$，以及某个$p \in Q_x$ 对应的 $w' \in L_x(p)$，满足 $\mu_{x-1}(p) = q$。由于 w 从 r 在 A^{ctl}中强接受，因此我们得到 p 的强连通分量已在 A^{ctl}中被移除，也就是说，$p \in S$。由于 $S \preccurlyeq_x S'$，存在一个状态 $p' \in S'$，满足$L_x(p) \subseteq L_x(p')$，且 $\mu_{x-1}(p) = \mu_{x-1}(p')$。然而，$w' \in L_x(p')$ 且 $\mu_{x-1}(p') = q$，这意味着 w 无法从 r 在 A^{ctl} 中强接受，因为 S' 在 A^{ctl}中仍是R 的子强连通分量。◀

▶**引理 5.15**。给定一个一致的规范化分层自动机$\mathcal{A}$，我们可以在$\mathcal{A}^{ctl}$多项式时间内计算。

证明。要得到$\mathcal{A}^{ctl}$ 只需确定 $x \geq 2$ 以及两个状态 p，$p' \in Q_x$ 它们位于$\mathcal{T}_x$ 的不同强连通分量中，并且满足 $p \preccurlyeq_x p'$。找到这样的状态后，只需从$\mathcal{A}$ 中移除 x-强连通分量中 p的状态，以及通过 $\widehat{\mu}_x$映射到该分量的状态。

为此，我们计算关系 $\preccurlyeq_x$。对于每个 $x \geq 2$，我们检查每一对 $p \sim_{x-1} p'$是否 $L_x(p) \subseteq L_x(p')$，这实际上就是确定性安全自动机的语言包含。◀

每次应用中心化操作都会移除一些状态。当该操作无法再应用时，经过至多 $|Q_{\geq 1}|$ 步，我们得到一个等价的自动机，它是正规的、安全最小的且中心化的。

5.5 子自动机与满射态射

应用上一小节中介绍的多项式时间变换，我们得到一个自动机 $\mathcal{A}_L$，它是正规、中心且安全最小，这证明了定理 5.1 的第一部分。下面证明该定理的第二部分，即初始自动机$\mathcal{A}$ 包含一个 子自动机，它允许存在一个满射强态射映射到$\mathcal{A}_L$。

这一构造分为 3 个步骤，产生了以下自动机序列：

$$\mathcal{A} \rightsquigarrow \mathcal{A}_{\mathsf{normal}} \rightsquigarrow \mathcal{A}_{\mathsf{safeMin}} \rightsquigarrow \mathcal{A}_{\mathsf{central}} = \mathcal{A}_L.$$

此外，我们已经证明这些自动机满足以下关系：

- $\mathcal{A}_{\mathsf{normal}}$ 是子自动机，属于$\mathcal{A}$，$\mathcal{A}_{\mathsf{normal}}$ 允许存在一个满射强态射映射到$\mathcal{A}_{\mathsf{safeMin}}$。$\mathcal{A}_{\mathsf{central}}$
- 是子自动机，属于$\mathcal{A}_{\mathsf{safeMin}}$ 通过移除后一个自动机中层级为 ≥ 2 的子树得到。
-

我们首先指出以下事实。

▶ **引理 5.16.** 设 $\varphi\colon \mathcal{A} \to \mathcal{B}$ 是一个态射，其定义在两个一致分层自动机之间，并设$\mathcal{B}'$ 是子自动机，它由$\mathcal{B}$ 在若干层 ≥ 2移除一些子树得到。那么，$\varphi^{-1}(\mathcal{B}')$是一个分层自动机，也是子自动机，属于$\mathcal{A}$。

证明。显然，$\mathrm{A}' = \varphi^{-1}(\mathrm{B}')$ 是 A 的子自动机，因为它是通过移除一些节点得到的。要验证它是分层自动机，首先注意到 B' 的第 1 层与 B 的第 1 层相同，因此 A' 的第 1 层与 A 的第 1 层相同。接下来只需验证，将这些 μ_x 函数限制在 A'上时，它们仍然是态射。取某个 $x \geq 2$的 $p \in \mathrm{T}_x^{\mathcal{A}'}$。我们有 $\mu_{x-1}(p) \in \mathrm{T}^{\mathcal{A}'}$，因为 B' 是通过从 B 中移除子树得到的。显然，转移在 A' 中仍由 μ_x 保持，因为它们在 A 中得到保持。◀

令 $\mathcal{A}'$ 为由前一个引理应用于态射 $_{\mathsf{normal}}$ 所得的子自动机 $\varphi\colon \mathcal{A}_{\mathsf{normal}} \to \mathcal{A}_{\mathsf{safeMin}}$ 以及 子自动机 $\mathcal{A}_{\mathsf{central}}$ 的 $\mathcal{A}_L$。于是，$\mathcal{A}' = \varphi^{-1}(\mathcal{A}_{\mathsf{central}})$ 是 子自动机，且为 $\mathcal{A}_{\mathsf{normal}}$的子自动机，因此也是子自动机，属于 $\mathcal{A}$。由于 φ 是一个 强态射

其在 $\mathcal{A}'$ 上的限制也是强的。此外，φ 是满射，因为$\mathcal{A}'$ 是 φ的原像。由此完成定理 5.1 的证明，因为$\mathcal{A}_{\mathsf{central}} = \mathcal{A}_L$。

▶ 备注 5.17（操作的顺序）。我们安排各项操作的顺序，以确保每一步都保留前面步骤所获得的性质。规范化首先进行，因为降低操作可能会破坏中心性和安全最小性。有趣的是，安全最小化并不总能保留中心化——这与 co-Büchi 情形不同（见 [AK22]），在该情形下，中心化可以先于安全最小化进行。相比之下，引理 5.14 证明了中心化既保留规范化，也保留安全最小性，因此自然适合作为最后一步。

6 基于同余的刻画

本节将基于有限词元组上的同余关系，给出最小分层自动机的另一种刻画。这种方法为任意分层自动机构造相应的典范最小 ω-正则语言。基于同余的视角补充了第 5 节介绍的句法最小化过程，并从有限词序列上的等价关系这一角度，进一步揭示分层自动机的结构。例如，这类序列的最大长度反映了该语言的奇偶优先级指标。

6.1 同余的定义和基本性质

设 $L \subseteq \Sigma^\omega$（这些定义可应用于所有无限词语言，但关于分层自动机 的结果仅适用于正则 ω-语言）。在我们基于同余对最小分层自动机 的描述中，针对 L，第 x 层上的状态是关于 $\equiv_L$ 上的 x-元组 $u = (u_1, \dots, u_x) \in (\Sigma^*)^x$ 的有限词等价类。关系$\equiv_L$ 可视为定义在所有有限词的有限元组（长度至少为一）集合上的一个等价关系，其中等价元组的长度必须相同。为了定义各层上的转移关系以及层间的态射，我们考虑这类有限元组上的两种运算：

连接： 我们用 $u \cdot u_{x+1}$ 表示与词 $u_{x+1} \in \Sigma^*$ 在最后一个分量上的连接，即 $u \cdot u_{x+1} = (u_1, \dots, u_{x-1}, u_x u_{x+1})$ 是一个元组，其前 $x-1$ 个分量与 u 相同，最后一个分量为 $u_x u_{x+1}$（若 $x = 1$，则结果元组就是$(u_1 u_2)$）。我们用这一运算来定义各层上的转移关系。

合并： 我们将 (u, u_{x+1}) 记作通过添加 u_{x+1} 作为额外分量得到的元组 $(u_1, \dots, u_x, u_{x+1})$。合并元组 (u, u_{x+1}) 中最后两个分量（该元组长度至少为 2）的操作，是移除最后一个分量，并将其连接到倒数第二个分量上，从而得到元组 $u_{_} u_{x+1}$。我们用此操作表示从第 $x+1$ 层到第 x层的态射。

在介绍正式定义之前，我们先来看一个例子，以便直观理解接下来要定义的概念。

▶ **例 6.1**。考虑语言 L：它由所有满足以下条件的、在字母表 $\Sigma = \{a, b\}$ 上的词组成：aba 只出现有限次，而 a 和 b 都出现无限次。L 的同余自动机如图 8 所示。该语言是前缀无关的，因此第 1 层上只有一个等价类，而其他层中所有等价类的代表元都可以将 ε 作为第一分量。第 2 层确保所有不包含 aba 的词都从某个状态开始是安全的。为了

描述这一点，我们引入元组为（等价于） $\perp$的概念。在该例中，$(u_1, u_2) \equiv_L \perp$ 当且仅当 u_2 包含以 aba 为中缀的内容。这可以通过考察将 u_2 扩展为词 $u_1(u_2w)^\omega \in L$的可能扩展u_2w 来刻画。当且仅当 u_2 不包含 aba 时，这样的扩展才存在。而不存在此类扩展的元组被声明为 $\perp$（在下文进一步讨论该例的第 3 层时，我们会对 $\perp$的定义作更多说明）。元组为 $\perp$ 的这一概念，为每个元组定义了一种安全语言：可以附加到最后一个分量之后、而不会使该元组变为 $\perp$的有限词集合。在第 x 层，我们的等价关系依据安全语言合并 x-元组中彼此等价的元组。这使得该例的第 2 层上形成了三个等价类，它们跟踪 aba的前缀，分别为 $[(\varepsilon, \varepsilon)]_{\equiv_L}$、$[(\varepsilon, a)]_{\equiv_L}$、$[(\varepsilon, ab)]_{\equiv_L}$。在图中，代表元 (ε, bb) 被用于等价类 $[(\varepsilon, \varepsilon)]_{\equiv_L}$。直观地说，这是因为 bb 是一个能够确保以下事实的词：将其连接到任意一个（第 2 层上的）元组后，所得元组要么是 $\perp$，要么等价于 (ε, bb)。因此，从这个意义上说，第二分量中的 bb 指向一个唯一的等价类。下文对带标记元组的定义刻画了这一点。我们利用这一概念选出用于构造自动机的等价类：只有包含带标记元组的等价类才会被使用。有些等价类可能只包含带标记元组，或者同时包含带标记元组和非带标记元组（如该例中的 $(\varepsilon, \varepsilon)$ 和 (ε, bb)），但也可能存在不包含任何带标记元组的等价类，而这些等价类不会被用作状态。后一种情况不会在本例中发生；关于这一点，请参见例6.6。第 3 层负责检查 a 和 b 是否出现无限次，因此只包含这两个字母中的一个的词必须在第 3 层上是安全的（因而会被拒绝，因为 3 是奇数）。注意，第 3 层上没有状态映射到第 2 层的 $[(\varepsilon, ab)]_{\equiv_L}$。原因是，如果 $[(\varepsilon, ab)]_{\equiv_L}$ 在第 2 层的一次安全运行中出现无限次，那么该词必定包含无限多个 a 和 b，因为在第 2 层该等价类上循环的所有词都包含 a 和 b。因此，在这种情况下，第 3 层无需检查任何内容。我们的 $\perp$ 定义如下地刻画了这一点。我们称第 3 层上的元组 (u_1, u_2, u_3) 是 $\perp$，当且仅当以下二者之一成立：(u_1, u_2u_3) 在第 2 层上是 $\perp$；或者，能够在 (u_1, u_2) 的等价类上闭合一个循环的每个扩展 $w \in \Sigma^+$（记作 $(u_1, u_2u_3w) \equiv_L (u_1, u_2)$）都满足：所得词 $u_1u_2(u_3w)^\omega$ 属于L。直观地说，这意味着我们无需在第 3 层上跟踪任何内容，因为如果第 2 层上的等价类出现无限次，就没有任何需要拒绝的情况。考虑该例中的元组 $(\varepsilon, ab, \varepsilon)$。合并最后两个分量得到 (ε, ab)。我们的定义确保 $(\varepsilon, ab, \varepsilon) \equiv_L \perp$，因为任何非空的 w，只要满足 $(\varepsilon, abw) \equiv_L (\varepsilon, ab)$，就必须使$abw$ 不包含 aba（否则 (ε, abw) 就会是 $\perp$，从而不会等价于 (ε, ab)），并且 w 以 ab结尾。于是显然 $ab(w)^\omega \in L$，因为 w 同时包含 a 和 b；而 $ab(w)^\omega$不包含 aba，这是由以下事实推出的： abw 不包含 aba，且 w以 ab结尾。

现在我们给出形式化定义，将 [LW25].[9] 中针对 co-Büchi 情形、定义在词对上的定义加以推广。

对于元组 $\bar{u} = (u_1, \ldots, u_x)$、$\bar{v} = (v_1, \ldots, v_x)$（其中各项均为有限词），我们按照元组的长度归纳定义以下概念：$\bar{u} \equiv_L \perp$、$\bar{u} \equiv_L \bar{v}$以及 $\bar{u}$ 是 带指针的。

对于归纳基 $x = 1$，定义

- $(u_1) \not\equiv_L \perp$ 对于所有 u_1，$(u_1) \equiv_L (v_1)$ 当且仅当 $u_1 \sim_L v_1$，
- (u_1)带指针的对于所有 u_1。

[9] 这里有一个小小的区别。在 [LW25] 的 co-Büchi 情形中，同余自动机的所有状态都是长度为 2 的元组的等价类。将这里的定义应用于 co-Büchi 情形时，在某些情况下也可能出现由长度为 1 的元组构成的等价类作为状态。元组长度上的这种灵活性实际上使定义更简单，因为它避免了像 [LW25] 中那样对 ε 作特殊处理。

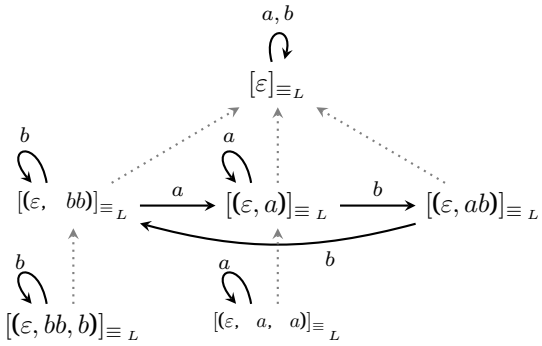


图 8 针对语言的同余自动机：该语言由含有限个 aba 、且含无限多个 a 和 b 的单词构成，见例 6.1。

底元素和安全语言。对于归纳步骤，令$(u, u_{x+1}) \equiv_L \bot$ 如果

$\overline{u} \cdot u_{x+1} \equiv_L \bot$，或者

$$\forall w \in \Sigma^+ : \overline{u} \cdot u_{x+1} w \equiv_L \overline{u} \Rightarrow [u_1 \cdots u_x (u_{x+1} w)^\omega \in L \Leftrightarrow x \text{ even}].$$

请注意，我们要求 w 非空，以确保循环部分 $u_{x+1}w$ 非空。

▶ 注 6.2。1. 设 (u, u_{x+1}) 满足不存在 $w \in \Sigma^+$ 使得 $u \cdot u_{x+1} w \equiv_L u$。那么，$(u, u_{x+1}) \equiv_L \bot$，因为量词“$\forall w$ [...]”是空泛的。2. 在证明中，我们常通过找出一个满足 $u \cdot u_{x+1} w \equiv_L u$ 且 $[u_1 \cdots u_x (u_{x+1} w)^\omega \in L \Leftrightarrow x + 1 \text{ even}]$ 的 $w \in \Sigma^+$，来证明一个元组是 $\not\equiv_L \bot$（注意这里改为“$x + 1$ even”）。有时我们把这样的 w 称为该元组不是 $\bot$ 的见证。

下一个引理指出，$\bot$ 在最后一个分量中对右侧连接保持不变，因此我们可以用它来定义一个安全语言。

▶ **引理 6.3**。设 $u \in (\Sigma^*)^x$ 且 $a \in \Sigma$。那么 $u \equiv_L \bot \Rightarrow u \cdot a \equiv_L \bot$。

证明。对 x 进行归纳。对于 $x = 1$，命题成立，因为长度为 1 的元组都不是 $\bot$。对于 $x+1$，假设 $(u, u_{x+1}a) \not\equiv_L \bot$。那么 $u \cdot u_{x+1}a \not\equiv_L \bot$，因此根据归纳假设，$u \cdot u_{x+1} \not\equiv_L \bot$。此外，存在 $w \in \Sigma^+$，使得 $u \cdot u_{x+1}aw \equiv_L u$ 且 $u_1 \cdots u_x (u_{x+1}aw)^\omega \in L$ 当且仅当 $x + 1$ 为偶数。于是，aw 是 $(u, u_{x+1}) \not\equiv_L \bot$ 的见证。◀

对于无限词 $w \in \Sigma^\omega$，我们定义 $(u, w) \not\equiv_L \bot$，当且仅当 $(u, v) \not\equiv_L \bot$ 对所有前缀 v 都成立，其中 w。元组的安全语言 是所有能够与该元组连接且不会使该元组变为 $\bot$ 的有限词和无限词的集合：

$$L^s((\overline{u}, u_{x+1})) := \{w \in \Sigma^* \cup \Sigma^\omega \mid (\overline{u}, u_{x+1} w) \not\equiv_L \bot\}.$$

请注意，安全语言完全由其中包含的有限词决定（安全语言中的无限词是指其所有有限前缀都属于安全语言的那些词）。

等价关系。基于此，令

$$(\overline{u}, u_{x+1}) \equiv_L (\overline{v}, v_{x+1}) \text{ if } \quad \overline{u} \cdot u_{x+1} \equiv_L \overline{v} \cdot v_{x+1}, \text{ and } L^s((\overline{u}, u_{x+1})) = L^s((\overline{v}, v_{x+1}))$$

这是相对于安全语言的剩余等价，并附加条件：这些元组在前一层映射到同一个等价类。元组的等价类记作 $[u]_{\equiv_L}$。

▶ 例 6.4。设 $L =$ 奇偶性$(1, 2, \ldots, d)$。那么，$\equiv_L$ 的元组是指长度为 $> d$ 或在某个索引为 $> x$的分量中包含 x 的元组。因此，对于每个长度 $x \in \{1, \ldots, d\}$，恰好有两个等价类：$[(\varepsilon, 1, 2, \ldots, x-1)]_{\equiv_L} \equiv_L \bot$，以及 $[(\varepsilon, \ldots, \varepsilon)]_{\equiv_L} \not\equiv_L \bot$。

下面的引理表明，$\equiv_L$ 是关于我们在词元组上考虑的操作的同余，即在最后一个分量中进行右连接、合并最后两个分量，以及扩展出一个新的最后分量。注意，这意味着$\equiv_L$ 在任意此类操作序列下保持不变，例如先为元组扩展出一个新的最后分量，然后通过反复右连接将一个词填入这个新分量中。

▶ **引理 6.5**。令 $u, v \in (\Sigma^*)^x$，$a \in \Sigma$，$u_{x+1}, v_{x+1} \in \Sigma^*$。以下性质成立。

1. $\overline{u} \equiv_L \overline{v} \Rightarrow \overline{u} \cdot a \equiv_L \overline{v} \cdot a$。**2.** $(u, u_{x+1}) \equiv_L (v, v_{x+1}) \Rightarrow u \cdot u_{x+1} \equiv_L v \cdot v_{x+1}$。**3.** $u \equiv_L v \Rightarrow (u, \varepsilon) \equiv_L (v, \varepsilon)$。

证明。 证明只是对定义的简单应用。

1. 显然，若 $L^s(u) = L^s(v)$，则也有 $L^s(u \cdot a) = L^s(v \cdot a)$。基于这一观察，对 x进行归纳即可直接得出证明。2. 这一性质属于 $\equiv_L$ 的归纳定义。3. 假设 $u \equiv_L v$。那么显然 $u \cdot \varepsilon \equiv_L v \cdot \varepsilon$。为了证明 $L^s(u, \varepsilon) = L^s(v, \varepsilon)$，令$u_{x+1} \in L^s(u, \varepsilon)$。那么存在 w，使得 $u \cdot u_{x+1} w \equiv_L u$，且 $u_1 \cdots u_x (u_{x+1} w)^\omega \in L$ 当且仅当$x+1$ 为偶数。根据假设，我们有 $u \equiv_L v$，由 1 可推出 $u \cdot u_{x+1} w \equiv_L v \cdot u_{x+1} w$，因此 $v \cdot u_{x+1} w \equiv_L v$。此外，$v_1 \cdots v_x (u_{x+1} w)^\omega \in L$ 当且仅当 $u_1 \cdots u_x (u_{x+1} w)^\omega \in L$，因为 $v_1 \cdots v_x \sim_L u_1 \cdots u_x$。因此，$u_{x+1} \in L^s(v, v_{x+1})$。由对称性可得 $L^s(u, \varepsilon) = L^s(v, \varepsilon)$。这表明 $(u, \varepsilon) \equiv_L (v, \varepsilon)$。

◀

带指针的。构造接受 L的分层自动机时，可以使用 $\equiv_L$ 的所有等价类（$\bot$除外）。不过，这个自动机不一定是最小自动机（见下文例6.6）。现在我们给出带标记元组的定义，它们用于选出 $\equiv_L$ 中实际用于最小自动机 A$_{\equiv_L}$定义的等价类。带指针的定义旨在使我们能够为包含带标记元组的等价类构造中心序列。考虑一个元组 $(u, u_{x+1}) \not\equiv_L \bot$。它可以看作是从 (u, ε) 读取 u_{x+1} 后到达的元组。(u, ε) 在层级 x 上的父节点是 u。现在，如果从任何一个以 u 为第 x层父节点的 $x+1$元组 (v, v_{x+1}) 出发，读取 u_{x+1} 后得到的元组 $(v, v_{x+1} u_{x+1})$ 要么是 $\bot$，要么等价于 (u, u_{x+1})，那么我们就称 (u, u_{x+1}) 是带指针的。换句话说，从 u的某个子节点出发，u_{x+1} 唯一地指向 (u, u_{x+1}) 的等价类。

形式上，$(\overline{u}, u_{x+1}) \not\equiv_L \bot$ 是带指针的，如果 u 是带指针的，并且对于所有带指针的 v 以及所有 v_{x+1} 满足$v \cdot v_{x+1} \equiv_L u$：

$$(\overline{v}, v_{x+1} u_{x+1}) \not\equiv_L \bot \Rightarrow (\overline{v}, v_{x+1} u_{x+1}) \equiv_L (\overline{u}, u_{x+1}).$$

如图 9 所示，灰色三角形表示定义中用到的所有子节点 $(v, v_{x+1})[u]_{\equiv_L}$。

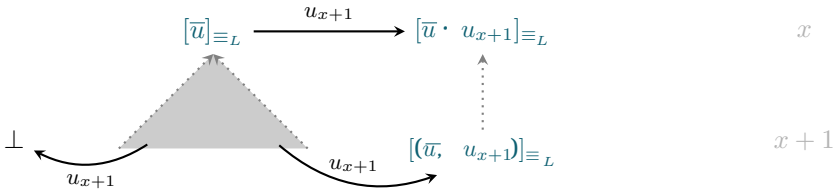


图 9 对定义的说明：(u, u_{x+1}) 是带指针的。基本上，从 $[u]_{\equiv_L}$ 的每个子节点出发，词 u_{x+1} 会到达 $\bot$ 或到达 (u, u_{x+1})。

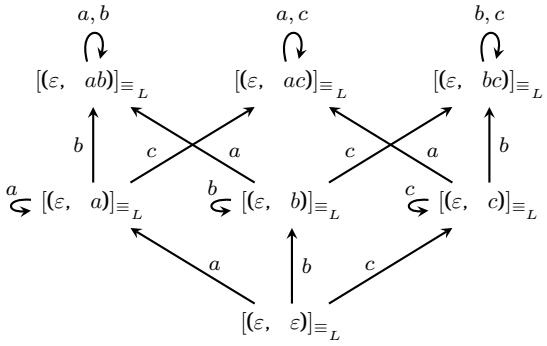


图10 语言 $\equiv_L$ 在第2 层上的所有 L 等价类示意图；该语言出自例6.6（至少一个字母出现有限次）。

▶ 例6.6。考虑字母表 $\Sigma := \{a, b, c\}$，以及由所有至少有一个字母只出现有限次的 ω-词组成的语言。这个语言是前缀无关的，因此第一层只有一个等价类，而且我们只需考虑第一分量中的 ε。此外，$(\varepsilon, u_2) \equiv_L \bot$ 当且仅当 u_2 包含所有字母；如果相同的字母出现在 u_2 和 v_2中，则 $(\varepsilon, u_2) \equiv_L (\varepsilon, v_2)$。长度大于 2 的所有元组都是 $\bot$。否则，若 $(\varepsilon, u_2, u_3) \not\equiv_L \bot$，则 $(\varepsilon, u_2u_3) \not\equiv_L \bot$，且存在 $w \in \Sigma^+$，使得 $(\varepsilon, u_2u_3w) \equiv_L (\varepsilon, u_2)$ 并且 $u_2(u_3w)^\omega \notin L$。由 $(\varepsilon, u_2u_3w) \equiv_L (\varepsilon, u_2)$ 可知，u_2和 u_2u_3w 包含相同的字母。由 $(\varepsilon, u_2u_3) \not\equiv_L \bot$ 可得，u_2u_3 不包含所有字母。因此 u_3w 不包含所有字母，所以 $u_2(u_3w)^\omega \in L$，这就产生了矛盾。第 2 层上 $\equiv_L$ 的完整图如图10所示（除了等价类 $\bot$，缺失的转移会指向该处）。所有这些等价类的父节点都是第1层唯一的等价类。对于这个自动机，我们只需要等价类 $[(\varepsilon, ab)]_{\equiv_L}$、$[(\varepsilon, ac)]_{\equiv_L}$和 $[(\varepsilon, bc)]_{\equiv_L}$，它们分别用于检查 c、b或 a 是否只出现有限次。而且，恰好只有这些等价类包含带标记元组。例如，考虑 (ε, ab)。从图中的任意等价类出发读取 ab，要么会到达 $\bot$，要么会停在 $[(\varepsilon, ab)]_{\equiv_L}$，因此 (ε, ab) 是带指针的。但例如，由元组 (ε, a^i) 组成的 (ε, a) 的等价类 $i \geq 1$不包含任何带标记元组。若 (ε, a^i) 是带指针的，那么根据指向的定义，若有 $(v, v_{x+1}) = (\varepsilon, ab)$，我们就会得到 $(\varepsilon, aba^i) \equiv_L \{\bot, (\varepsilon, a)\}$，但实际上 $(\varepsilon, \overline{ab}a^i) \equiv_L (\varepsilon, ab)$。换句话说，从 (ε, ab) 出发读取 a^i 会回到 (ε, ab)，而不是到达 (ε, a) 或 $\bot$。

下面的引理断言，带指针的性质在最后一个分量中进行右连接以及合并最后两个分量时都能保持不变。

▶ **引理 6.7**。设 $u \in \overline{(\Sigma^*)^x}$, $a \in \Sigma$, $u_{x+1} \in \Sigma^*$。以下性质成立。

1. 若 $\overline{u}$ 是 带指针的 且 $\overline{u}\cdot a \not\equiv_L \bot$，则 $u\overline{\cdot} a$ 是 带指针的。
2. 若 $(\overline{u}, u_{x+1})$ 是 带指针的，则 $\overline{u}\cdot u_{x+1}$ 是 带指针的。

证明。 注意，性质 2 可由性质 1 推出：若 $(u,\ u_{x+1})$ 是带指针的，则根据定义，u 是带指针的。反复应用性质 1，可得 $u\cdot u_{x+1}$ 是带指针的。

性质 1 显然对 $x=1$ 成立，因为所有长度为 1 的元组都是带指针的。现在我们证明长度至少为 2 的元组满足该命题，因此假设 (u,u_{x+1}) 是带指针的。为证明 $(u,u_{x+1}a)$ 是带指针的，取 $v\in(\Sigma^*)^x$，$v_{x+1}\in\Sigma^*$，使得 $v\cdot v_{x+1}\equiv_L u$。那么 $(v,\overline{v}_{x+1}u_{x+1}) \equiv_L \{\bot,(u,u_{x+1})\}$，因为 $(\overline{u},u_{x+1})$ 是带指针的。

若 $(v,\ v_{x+1}u_{x+1}) \equiv_L \bot$，则根据引理 6.3，$(v,\ v_{x+1}u_{x+1}a) \equiv_L \bot$。若 $(v,\ v_{x+1}u_{x+1}) \equiv_L (u,\ u_{x+1})$，则根据引理 6.5(1)，$(v,\ v_{x+1}u_{x+1}a) \equiv_L (u,\ u_{x+1}a)$。◀

正则性与有限性。

我们证明，对于 ω-正则语言，$\equiv_L$-等价类的数量是有限的。证明分为两部分：第一部分（引理 6.8）证明，在每一层上（对于元组的每个固定长度），$\equiv_L$-等价类的数量是有限的。第二部分（引理 6.9）证明，只有有限多个层包含不同于 $\bot$的等价类。

▶ **引理 6.8。** 如果 L 是 ω-正则的，那么对于所有 $\equiv_L$，$\equiv_L$ 在 $(\Sigma^*)^x$ 上的索引是有限的。$x\geq 1$

证明。设 $L=L(\mathrm{A})$ 为一个非确定性 Büchi 自动机（NBA）A。对于有限词$u\in\Sigma^*$，令 u 的转移概况包含 A 的每一对 p、q 状态的信息，即是否存在一条在 u 上从 p 到 q 的运行，以及是否存在一条经过接受状态的此类运行。通过对 x 进行归纳，不难证明：如果 u、$v\in(\Sigma^*)^x$ 满足对于所有 i，u_i、v_i 具有相同的转移概况，那么 $u\equiv_L v$。由于可能的转移概况只有有限多个，命题得证。因此，设 u、$v\in(\Sigma^*)^x$ 满足u_i、v_i 对于每个 i都具有相同的转移概况。如果 u_1、v_1 具有相同的转移概况，那么它们也属于同一个剩余类，因为从 A 的初始状态经由 u_i 和经由 v_i可达的状态相同。因此，对于 $x=1$，命题得证。对于归纳步骤，考虑具有相同转移概况的u_{x+1}、$v_{x+1}\in\Sigma^*$，并证明 $(u,\ u_{x+1}) \equiv_L (v,\ v_{x+1})$。由于$u_x$、$v_x$ 以及 u_{x+1}、v_{x+1} 分别具有相同的转移概况，所以 $\overline{u}_x u_{x+1}$、$v_x\overline{v}_{x+1}$也具有相同的转移概况，因此根据归纳法，$u\cdot u_{x+1}\equiv_L v\cdot v_{x+1}$。由引理 6.5(1)可知，这意味着对于所有 $w'\in\Sigma^*$，$u\cdot u_{x+1}w'\equiv_L v\cdot v_{x+1}w'$。于是，根据归纳法，$(u,\ u_{x+1}w') \equiv_L u$ 当且仅当 $(v,\ v_{x+1}w') \equiv_L v$，因为 $u\equiv_L v$ 也成立。此外，由于所有对 u_i、v_i 都具有相同的转移概况，我们得到 $u_1\cdots u_x(u_{x+1}w')^\omega\in L$ 当且仅当 $v_1\cdots v_x(v_{x+1}w')^\omega\in L$。这意味着，对于每个 $w\in L^s(u,\ u_{x+1})$，见证 $(u,\ u_{x+1}w) \not\equiv_L \bot$ 的相同单词也见证 $(v,\ v_{x+1}w) \not\equiv_L \bot$，反之亦然。因此 $L^s(u,\ u_{x+1}) = L^s(v,\ v_{x+1})$。◀

下面的引理表明，$\equiv_L$ 具有“有限深度”，如果 L 是 ω-正则的，那么未 $\bot$ 元组的长度有上界，该上界由语言的奇偶优先级指标给出。

▶ **引理 6.9。** 如果 L 是 ω-正则的，且由一个 *DPA* 接受，该 *DPA* 的优先级属于 $[1,d]$，那么 $u\equiv_L \bot$对所有 $u\in(\Sigma^*)^{d+1}$（因此，对所有更长的元组也成立）。

证明。设 A 是一个用于 L 的 DPA，并令 n 大于 A 的状态数。反设存在 $u=(u_1,\dots,u_{d+1})\in(\Sigma^*)^{d+1}$ 满足 $u\not\equiv_L \bot$。我们归纳地定义词 $v_x\in\Sigma^*$，其中 $x\in\{2,\dots,d+2\}$，并证明 A 在$u_1v_2^n$ 上的运行所进入的循环必定访问 d 个不同的优先级，且其中最小的优先级为偶数。如果 A 仅使用优先级 $1,\dots,d$，这是不可能的。

我们先给出 $v_{d+2}:=\varepsilon$ 的定义，这仅用于一个简单的归纳基。

假设 v_{x+1} 已经对 $x \in \{2, \ldots, d+1\}$ 定义，并且 $(u_1, \ldots, u_x) \cdot v_{x+1} \equiv_L (u_1, \ldots, u_x)$。由于 $v_{d+2} := \varepsilon$，显然当序列以 $x = d+1$ 开始时，此条件成立。由 $(u_1, \ldots, u_x) \cdot v_{x+1} \equiv_L (u_1, \ldots, u_x)$，根据引理 6.5(1)，我们还可得，对于所有 i，$(u_1, \ldots, u_x) \cdot v_{x+1}^i \equiv_L (u_1, \ldots, u_x)$。特别地，$(u_1, \ldots, u_x) \cdot v_{x+1}^n \not\equiv_L \bot$。因此，存在一个词 $w_x \in \Sigma^+$，使得 $(u_1, \ldots, u_{x-1}) \cdot u_x v_{x+1}^n w_x \equiv_L (u_1, \ldots, u_{x-1})$，且 $[u_1 \cdots u_{x-1}(u_x v_{x+1}^n w_x)^\omega \in L \Leftrightarrow x \text{ 是偶数}]$。令 $v_x := u_x v_{x+1}^n w_x$。注意

$$(u_1, \ldots, u_{x-1}) \cdot v_x = (u_1, \ldots, u_{x-1}) \cdot u_x v_{x+1}^n w_x \equiv_L (u_1, \ldots, u_{x-1}),$$

因此，v_x 满足归纳定义所假设的性质。

我们的目标是证明，v_x 在 $\mathcal{A}$ 中诱导出 $(d+1)$ 个具有不同奇偶性的嵌套循环，其接受状态交替变化。为此，我们需要从正确类型的状态开始读取 v_x。设 $\mathcal{A}$ 的初始状态为 q_0，并设在 $\mathcal{A}$ 中读取 $u_1 \cdots u_i$ 后到达的状态为 q_i：

$$q_0 \xrightarrow{u_1} q_1 \xrightarrow{u_2} q_2 \xrightarrow{u_3} \cdots \xrightarrow{u_{d+1}} q_{d+1}$$

现在我们证明，对于每个 $q \sim q_{x-1}$，只要 v_x 的某个非空幂在 q 上形成循环，该循环就包含 $d+2-x$ 个不同的优先级，其中最小优先级的奇偶性与 x 相同。设 $q \xrightarrow{v_x^+} q$，其中加号表示某个非空幂。根据 v_x 的选择，我们有 $u_1 \cdots u_{x-1} v_x^\omega \in L$ 当且仅当 x 为偶数。因此，v_x^ω 被接受当且仅当 q 为偶数，这意味着循环上的最小优先级与 x 具有相同的奇偶性。

我们通过从 $x = d+1$ 开始向下归纳，证明循环中至少包含 $d+2-x$ 个优先级。归纳基显然成立，因为循环中至少必须包含 1 个优先级。

若 $x < d+1$，则考虑经过前 $v_x = u_x v_{x+1}^n w_x$ 个因子后到达的状态：

$$q \xrightarrow{u_x} p_0 \xrightarrow{v_{x+1}} p_1 \xrightarrow{v_{x+1}} p_2 \cdots \xrightarrow{v_{x+1}} p_n \xrightarrow{w_x} q'$$

由于 $q \sim q_{x-1}$，且 p_0 可经由 u_x 从 q 到达，我们得到 $p_0 \sim q_x$。此外，如上所述，$(u_1, \cdots u_x) \cdot v_{x+1}^i \equiv_L (u_1, \cdots u_x)$。因此，对于所有 $i \in \{0, \ldots\ldots, n\}$，我们都有 $p_i \sim q_x$。根据 n 的选取，必存在 $i \neq j$，使得 $p_i = p_j =: p$，因此我们找到了一个状态 $p \sim q_x$，使得 v_{x+1} 的某个非零次幂在 p 上循环。根据归纳，我们知道在这个循环上至少访问 $d+2-(x+1)$ 个优先级，其中最小优先级与 $x+1$ 具有相同的奇偶性。因此，在循环 $q \xrightarrow{v_x^+} q$ 上访问到的最小优先级必定严格小于循环 $p \xrightarrow{v_{x+1}^+} q$ 上的最小优先级，从而得到所需的命题。

由此我们立即得到，在 $u_1 v_2^n$ 上进入的循环至少包含 $d = d+2-2$ 个优先级，且其中最小的优先级为偶数。◀

6.2 自动机 $\mathcal{A}_{\equiv_L}$

对于 $\mathcal{A}_{\equiv_L}$ 的定义，我们假设 L 是 ω-正则的。因此，在所有涉及 $\mathcal{A}_{\equiv_L}$ 的陈述中，我们都假设 L 是一个 ω-正则语言（在主要结果的定理表述中，我们会再次明确说明这一点，但并非在所有辅助引理中都如此）。令 d 为最大的值，使得存在 $u \in (\Sigma^*)^d$ 满足 $u \not\equiv_L \bot$。根据引理 6.9，这是使得 d 最小的值，且 L 可由一个 $[1, d]$-DPA 接受。

第 x 层由以下内容组成：$\mathcal{A}_{\equiv_L}$ 中维数为带指针的元组的等价类 x，其转移对应于最后一个分量上的连接。从第 $x+1$ 层到第 x 层的态射通过合并最后两个分量得到。初始状态则照例为空字的等价类。形式化地，我们定义分层自动机 $\mathcal{A}_{\equiv_L} = (q_0^{\equiv_L}, \mathcal{T}_1^{\equiv_L}, \mu_1^{\equiv_L}, \ldots\ldots, \mathcal{T}_{d-1}^{\equiv_L}, \mu_{d-1}^{\equiv_L}, \mathcal{T}_d^{\equiv_L})$ 如下：

- $\mathcal{T}_x^{\equiv_L} = (Q_x^{\equiv_L}, \delta_x^{\equiv_L})$ with
  - $Q_x^{\equiv_L} = \{[\overline{u}]_{\equiv_L} \mid \overline{u} \in (\Sigma^*)^x \text{ pointed}\}$
  - $\delta_x^{\equiv_L}([\overline{u}]_{\equiv_L}, a) = \begin{cases} [\overline{u}\cdot a]_{\equiv_L} & \text{if } \overline{u}\cdot a \not\equiv_L \bot \\ \text{undefined} & \text{else.} \end{cases}$
- $q_0^{\equiv_L} = [\varepsilon]_{\equiv_L}$
- $\mu_x^{\equiv_L} : Q_{x+1}^{\equiv_L} \to Q_x^{\equiv_L}$ with $\mu_x^{\equiv_L}([(\overline{u}, u_{x+1})]_{\equiv_L}) = [\overline{u}\cdot u_{x+1}]_{\equiv_L}$.

注意，在 $\delta_x^{\equiv_L}$和 $\mu_x^{\equiv_L}$的定义中，根据引理 6.5(1) 和 (2)，结果与所选代表元无关。此外，$\mu_x^{\equiv_L}$是一个态射，因为先连接再合并，与先合并再连接并无区别：

$$\begin{aligned} \mu_x^{\equiv_L}(\delta_{x+1,\equiv_L}([(\overline{u}, u_{x+1}]_{\equiv_L}), a)) &= \mu_x^{\equiv_L}([(\overline{u}, u_{x+1}a)]_{\equiv_L}) \\ &= [\overline{u}\cdot u_{x+1}a]_{\equiv_L} \\ &= \delta_x^{\equiv_L}([\overline{u}\cdot u_{x+1}]_{\equiv_L}, a) \\ &= \delta_x^{\equiv_L}(\mu_x^{\equiv_L}([(\overline{u}, u_{x+1})]_{\equiv_L}, a)) \end{aligned}$$

因此 $\mathcal{A}_{\equiv_L}$ 是一个分层自动机。

6.3 $\mathcal{A}_{\equiv_L}$的正确性

我们证明$\mathcal{A}_{\equiv_L}$ 是一个一致分层自动机，它接受 L （定理 6.14）。首先需要一些辅助定义和引理。

我们的第一个目标是证明，元组的安全语言被带指针的元组所覆盖。对于有限词和无限词，这分别由引理 6.11 和 6.12 表述。为了构造相应的元组，我们从辅助引理 6.10 开始；该引理将在引理 6.11 的证明中应用（因此读者现在可以跳过引理 6.10，阅读引理 6.11 的证明时再回来查看）。

▶ **引理 6.10**。设 $u, v \in (\Sigma^*)^x$ 以及 $u_{x+1}, v_{x+1} \in \Sigma^*$。若 $\overline{u} \equiv_L v\cdot v_{x+1}$，则 $L^s(u, u_{x+1}) \supseteq L^s(v, v_{x+1}u_{x+1})$。

证明。设 $w \in L^s(v, v_{x+1}u_{x+1})$，也就是说，$(v, v_{x+1}u_{x+1}w) \not\equiv_L \bot$。我们证明 $w \in L^s(u, u_{x+1})$成立。只需对有限词 w 证明这一点，因为安全语言中的无限词由有限词确定。

若 $(v, v_{x+1}u_{x+1}w) \not\equiv_L \bot$，则存在 $w' \in \Sigma^+$，使得 $v\cdot v_{x+1}u_{x+1}ww' \equiv_L v$ 且$v_1\cdots v_x(v_{x+1}u_{x+1}ww')^\omega \in L$ 当且仅当 $x+1$ 是偶数。注意，$u \equiv_L v\cdot v_{x+1}$ 尤其意味着$u_1\cdots u_x \equiv_L v_1\cdots v_xv_{x+1}$。由于 $v_1\cdots v_x(v_{x+1}u_{x+1}ww')^\omega = v_1\cdots v_xv_{x+1}(u_{x+1}ww'v_{x+1})^\omega$，我们得到 $u_1\cdots u_x(u_{x+1}ww'v_{x+1})^\omega \in L$ 当且仅当 $x+1$ 是偶数。进一步

$$\overline{u}\cdot u_{x+1}ww'v_{x+1} \equiv_L \underbrace{\overline{v}\cdot v_{x+1}u_{x+1}ww'}_{\equiv_L \overline{v}} v_{x+1} \equiv_L \overline{v}\cdot v_{x+1} \equiv_L \overline{u}$$

因此，后缀 $w'v_{x+1}$ 见证了 $(u, u_{x+1}w) \not\equiv_L \bot$，进而也见证了 $w \in L^s(u, u_{x+1})$。 ◀

下面的引理指出，如果我们从一个长度为 x 的带标记元组 u 出发，那么空的第 $(x+1)$ 个分量的长度为 $(x+1)$ 的元组 (u, ε) 的安全语言，被 $[u]_{\equiv_L}$ 的带标记子节点 (v, v_{x+1}) 覆盖，也就是说，那些长度为 $x+1$ 且满足 $\mu_x^{\equiv_L}([(v, v_{x+1})]_{\equiv_L}) = [u]_{\equiv_L}$ 的带标记 (v, v_{x+1})。回想一下，$\mu_x^{\equiv_L}([(v, v_{x+1})]_{\equiv_L}) = [u]_{\equiv_L}$ 等价于 $v\cdot v_{x+1} \equiv_L u$。

▶ **引理 6.11.** 考虑 (u, u_{x+1}) $\not\equiv_L \bot$ 使得 u 是带指针的。那么存在带指针的 (v, v_{x+1}) 使得 $v \cdot v_{x+1} = u$ 且 $(v, v_{x+1}u_{x+1}) \not\equiv_L \bot$。

证明。假设我们可以证明存在 $v \in (\Sigma^*)^x$, $v_{x+1} \in \Sigma^*$，其中 $v \cdot v_{x+1} \equiv_L u$，且 $(v, v_{x+1}u_{x+1})$ 是带指针的（注意，该引理声称，末尾没有 u_{x+1} 的元组必须是带指针的）。由此，我们可以按如下方式构造所需的元组。由于 $(v, v_{x+1}u_{x+1}) \not\equiv_L \bot$（因为它是带指针的），因此存在 $w \in \Sigma^+$，使得 $v \cdot v_{x+1}u_{x+1}w \equiv_L v$，且 $v_1 \cdots v_x(v_{x+1}u_{x+1}w)^\omega \in L$ 当且仅当 $x+1$ 是偶数。于是 $(v, v_{x+1}u_{x+1}wv_{x+1}u_{x+1}) \not\equiv_L \bot$。此外，$v \cdot \underbrace{v_{x+1}u_{x+1}w}_{\equiv_L} v_{x+1} \equiv_L v \cdot v_{x+1} \equiv_L u$ 根据引理 6.5，且 $(v, v_{x+1}u_{x+1}wv_{x+1})$ 是带指针的 作为一个 带指针的 元素的扩展（根据引理 6.7）。因此，有了 $v'_{x+1} = v_{x+1}u_{x+1}wv_{x+1}$ 我们便得到一个元组 (v, v'_{x+1}) 如引理所述。

剩下要证明的是，我们可以找到 $v \in (\Sigma^*)^x$, $v_{x+1} \in \Sigma^*$，使得 $v \cdot v_{x+1} \equiv_L u$，且 $(v, v_{x+1}u_{x+1})$ 带指针。思路相当简单。我们从 (u, u_{x+1}) 开始，将其作为当前元组。只要当前元组不带指针，就存在一个见证：一个新的元组，其最后一个分量以 u_{x+1} 结尾，且其安全语言不同于当前元组的安全语言。根据引理 6.10，新元组的安全语言必定严格包含于前一个元组的安全语言中。这种情况只会发生有限次，因为元组的安全语言仅取决于其 $\equiv_L$ 等价类，而 $\equiv_L$ 等价类只有有限多个。我们将在下文形式化这一思路。

从 $i = 0$ 开始，我们定义一个由元组组成的序列 $(v^{(i)}, v^{(i)}_{x+1})$，其中 $v^{(i)} \in (\Sigma^*)^x$ 和 $v^{(i)}_{x+1} \in \Sigma^*$，使得

(1) $v^{(i)}$ 是带指针的，(2) $v^{(i)} \cdot v^{(i)}_{x+1} \equiv_L u$，(3) $(v^{(i)}, v^{(i)}_{x+1}u_{x+1}) \not\equiv_L \bot$，并且 (4) 如果 $i > 0$，那么 $L^s(v^{(i)}, v^{(i)}_{x+1}u_{x+1}) \subsetneq L^s(v^{(i-1)}, v^{(i-1)}_{x+1}u_{x+1})$。

我们证明，如果 $(v^{(i)}, v^{(i)}_{x+1}u_{x+1})$ 不是带指针的，我们就可以将该序列再延长一个元组。由于最后一个性质，每个这样的序列都必须是有限的，因为一个元组的安全语言只取决于它的 $\equiv_L$ 等价类，而根据引理 6.8，对于每种元组长度，$\equiv_L$ 等价类的数量都有限（我们假设 L 是正则的）。

我们从 $v^{(0)} = u$ 和 $v^{(0)}_{x+1} = \varepsilon$ 开始，很容易看出它具备所有这些性质。如果 $(v^{(i)}, v^{(i)}_{x+1}u_{x+1})$ 是带指针的，那么我们就找到了所需的元组，其中 $v := v^{(i)}$ 和 $v_{x+1} := v^{(i)}_{x+1}$。否则，我们将说明如何获得序列中的下一个元组。

因此，假设 $(v^{(i)}, v^{(i)}_{x+1}u_{x+1})$ 不是带指针的。由于 $v^{(i)}$ 是带指针的，必定存在 $v' \in (\Sigma^*)^x$ 和 $v'_{x+1} \in \Sigma^*$ 使得

(i) v' 是带指针的，(ii) $v' \cdot v'_{x+1} \equiv_L v^{(i)}$，(iii) $(v', v'_{x+1}v^{(i)}_{x+1}u_{x+1}) \not\equiv_L \bot$，以及 (iv) $(v', v'_{x+1}v^{(i)}_{x+1}u_{x+1}) \not\equiv_L (v^{(i)}, v^{(i)}_{x+1}u_{x+1})$。

若定义 $v^{(i+1)} := v'$ 和 $v^{(i+1)}_{x+1} := v'_{x+1}v^{(i)}_{x+1}$，则新元组的性质 (1) 和 (3) 可由 (i) 和 (iii) 立即得出。性质 (2) 根据 (ii) 可由 $v^{(i+1)} \cdot v^{(i+1)}_{x+1} = v' \cdot v'_{x+1}v^{(i)}_{x+1} \equiv_L v^{(i)}v^{(i)}_{x+1} \equiv_L u$ 得出，其中倒数第二个等价关系来自 (ii)，最后一个等价关系则来自 i 的 (2)。性质 (4) 由 (iv) 和引理 6.10 得出，因为 $(v', v'_{x+1}v^{(i)}_{x+1}u_{x+1}) \not\equiv_L (v^{(i)}, v^{(i)}_{x+1}u_{x+1})$ 与 $L^s(v', v'_{x+1}v^{(i)}_{x+1}u_{x+1}) \neq L^s(v^{(i)}, v^{(i)}_{x+1}u_{x+1})$ 相同，因此由引理 6.10 $L^s(v', v'_{x+1}v^{(i)}_{x+1}u_{x+1}) \subsetneq L^s(v^{(i)}, v^{(i)}_{x+1}u_{x+1})$ 得出。根据 (ii)，我们可以应用引理 6.10。◀

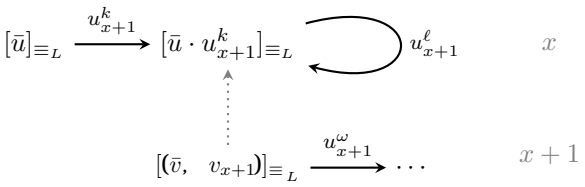


图 11 在以下假设下，引理 6.13 证明中的矛盾示意图：$(u\cdot v^k,\ v^\omega)\not\equiv_L \bot$。

以下是引理 6.11 关于无限词的一个类似结论。

▶ 引理 *6.12*。设 $u\in(\Sigma^*)^x$ 是带标记的，且 $w\in\Sigma^\omega$，满足 $(u,w)\not\equiv_L\bot$。则存在 $v\in(\Sigma^*)^x$，$\bar{v}_{x+1}\in\Sigma^*$，使得 $v\cdot v_{x+1}\equiv_L u$，$(v,\bar{v}_{x+1})$ 是带标记的，且 $(v,v_{x+1}\bar{w})\not\equiv_L\bot$。

证明。我们可以对 w 的每个有限前缀 $u^{(i)}=u_{x+1}$ 应用引理 6.11，从而得到 $v^{(i)}\in(\Sigma^*)^x$，$v_{x+1}^{(i)}\in\Sigma^*$ $\bar{v}^{(i)}$ $v_{x+1}^{(i)}\equiv_L\bar{u}$ $\bar{v}^{(i)}$ $v_{x+1}^{(i)}$ $\bar{v}^{(i)}$ $v_{x+1}^{(i)}u^{(i)}$ $\not\equiv_L\bot$，满足 $\cdot$，$(\ ,\ )$ 是带标记的，且 $(\ ,\ )$。我们可以假设每个 $(v^{(i)},v_{x+1}^{(i)})$ 都是其等价类 $\equiv_L$ 中最短的带标记元组，因为所需性质对于某个等价类 $\equiv_L$ 的所有带标记元组要么都成立，要么都不成立。因此，$(v^{(i)},v_{x+1}^{(i)})$ 的候选项只有有限个，故必定存在一个元组 (v,v_{x+1})，对无限多个前缀 $u^{(i)}$ 都等于 $(v^{(i)},v_{x+1}^{(i)})$。于是，这个元组满足所有所需性质。◀

下一个引理用于定理 6.14 的证明中，以说明 $\mathcal{A}_{\equiv_L}$ 是一致且接受语言 L。

▶ **引理 6.13**。设 x 为偶数，且 $u\in(\Sigma^*)^x$ 为带指针的，使得 u_{x+1}^ω 在从起 $[u]_{\equiv_L}$ 于 $\mathcal{A}_{\equiv_L}$ 中被强接受。那么 $u_1\cdots u_xu_{x+1}^\omega\in L$。对于 x 奇数，u_{x+1}^ω 强拒绝以及 $u_1\cdots u_xu_{x+1}^\omega\notin L$，相应的结论也成立。

证明。我们证明 x 为偶数的情形。由于我们从不涉及层 $x-1$，x 为奇数时的证明完全对称。

由于 u_{x+1}^ω 从 $[u]_{\equiv_L}$ 起被强接受，因此有 $u\cdot u_{x+1}^\omega\not\equiv_L\bot$。设 $k\geq 0$、$\ell\geq 1$ 满足 $u\cdot u_{x+1}^k\equiv_L u\cdot u_{x+1}^{k+\ell}$。假设 $(u\cdot u_{x+1}^k,u_{x+1}^\omega)\not\equiv_L\bot$。那么，根据引理 6.12，存在 $v\in(\Sigma^*)^x$、$\bar{v}_{x+1}\in\Sigma^*$，使得 $v\cdot v_{x+1}\equiv_L u\cdot u_{x+1}^k$，$(v,\bar{v}_{x+1})$ 带指针且 $(v,v_{x+1}\bar{u}_{x+1}^\omega)\not\equiv_L\bot$。这将构成一个见证，表明 u_{x+1}^ω 并非从 $[u]_{\equiv_L}$ 起被强接受，如图 11 所示。因此，必有 $(u\cdot u_{x+1}^k,u_{x+1}^\omega)\equiv_L\bot$，这意味着对于某个 m，有 $(u\cdot u_{x+1}^k,u_{x+1}^m)\equiv_L\bot$。反复应用 $u\cdot u_{x+1}^k\equiv_L u\cdot u_{x+1}^{k+\ell}$，可得 $u\cdot u_{x+1}^{k+\ell\cdot m}\equiv_L u\cdot u_{x+1}^k$。根据 $(u\cdot u_{x+1}^k,u_{x+1}^m)\equiv_L\bot$ 的定义，这意味着 $u_1\cdots u_xu_{x+1}^\omega=u_1\cdots u_xu_{x+1}^k(u_{x+1}^{\ell\cdot m})^\omega\in L$，因为 x 为偶数。◀

▶**定理 6.14.** 设 L 是一个 ω-正则语言。那么 $\mathcal{A}_{\equiv_L}$ 是一个满足 $L(\mathcal{A}_{\equiv_L})=L$。

证明。假设 $\mathrm{A}_{\equiv_L}$ 不是一致语义确定的，因此根据推论 3.17，它不一致。于是存在一个偶数 x、一个奇数 y 以及带标记元组 $u\in(\Sigma^*)^x$、$v\in(\Sigma^*)^y$，满足 $u_1\cdots u_x\sim_L v_1\cdots v_y$，并且某个 $w\in\Sigma^\omega$ 从 $[u]_{\equiv_L}$ 出发被强接受，而从 $[v]_{\equiv_L}$ 出发被强拒绝。显然，所有这些 w 的集合是 ω-正则的，因此也存在一个具有此性质的最终周期 $w=uv^\omega$。于是，v^ω 从 $[u\cdot u]_{\equiv_L}$ 出发被强接受，而从 $[v\cdot u]_{\equiv_L}$ 出发被强拒绝。根据引理 6.13，我们得到 $u_1\cdots u_xuv^\omega\in L$

以及 $v_1 \cdots v_y uv^\omega \notin L$，这与 $u_1 \cdots u_x \sim_L v_1 \cdots v_y$ 矛盾。我们得出 $\mathcal{A}_{\equiv_L}$ 是一致的。

为了证明 $L(\mathrm{A}_{\equiv_L}) = L$，设 uv^ω 为一个最终周期词。根据推论 3.18，uv^ω 包含一个后缀，该后缀被 $\mathrm{A}_{\equiv_L}$ 中的某个状态强接受或强拒绝，但不会同时被两者接受和拒绝，因为我们已经证明 $\mathrm{A}_{\equiv_L}$ 是一致的。于是，v^ω 从某个 $[u]_{\equiv_L}$ 处被强接受或强拒绝，对于一个带指针的元组 $u = (u_1, \ldots, u_x) \in \Sigma^x$，使得 $(\overline{u_1 \cdots u_x})$ 可通过 uv^k 中的前缀从 (ε) 在第 1 层上到达。由于 $\mathrm{A}_{\equiv_L}$ 中的所有转移都尊重 $\sim_L$ 等价类，我们得到 $u_1 \cdots u_x \sim_L uv^k$。根据引理 6.13，我们得到$uv^k v^\omega \in L$ 当且仅当 x 为偶数。因此，$L(\mathrm{A}_{\equiv_L})$ 和 L 包含相同的最终周期词，因而相等（这由 ω-正则语言的闭包性质得出，参见例如 [CNP93，事实 1]）。◀

6.4 $\mathcal{A}_{\equiv_L}$的最小性

我们证明，$\mathcal{A}_{\equiv_L}$ 是安全最小、正规且中心化的（引理 6.18），然后使用定理 4.1 得出结论：$\mathcal{A}_{\equiv_L}$ 是层次自动机的最小 L。

我们首先证明，$\mathcal{A}_{\equiv_L}$ 具有中心序列。证明分为三个引理：引理 6.15 说明，如何为每个具有一个子节点且该子节点是中心序列的带指针的等价类 $\perp$构造中心序列。引理 6.16 说明，$\mathcal{A}_{\equiv_L}$ 的叶节点的所有子节点都是 $\perp$；在引理 6.17 的证明中，我们证明引理 6.15 也适用于$\mathcal{A}_{\equiv_L}$的内部状态。

▶ 引理 *6.15*。设 $x \geq 2$，$\overline{u} \in (\Sigma^*)^{x-1}$ 和 $u_x, u_{x+1} \in \Sigma^*$，其中 (u, u_x) 是带指针的，$(u, \overline{u_x}, u_{x+1}) \equiv_L \perp$ 且 $(u, u_x\overline{u_{x+1}}) \not\equiv_L \perp$。那么 $[(u, u_x\overline{u_{x+1}})]_{\equiv_L}$ 有一个中心序列。

证明。由于 $(\overline{u}, u_x u_{x+1}) \not\equiv_L \perp$，存在 w 满足 $\overline{u \cdot u_x u_{x+1} w} \equiv_L \overline{u}$ 且 $u_1 \cdots u_{x-1}(u_x u_{x+1} w)^\omega \in L$ 当且仅当 x 为偶数。我们证明 $z := wu_x u_{x+1}$ 是 中心序列，对于 $[(u, u_x u_{x+1})]_{\equiv_L}$。

首先，$[u \cdot u_x u_{x+1}]_{\equiv_L}$的每个子节点都被映射到 $[(\overline{u}, u_x u_{x+1})]_{\equiv_L}$ 或 $\perp$，这是因为$(u, u_x u_{x+1})$ 是带指针的：令(v, v_x)为带指针的，使得 $\overline{v \cdot v_x} \equiv_L u \cdot u_x u_{x+1}$。于是根据引理 6.5，$v \cdot v_x w \equiv_L u \cdot u_x u_{x+1} w \equiv_L u$，从而

$$(\overline{v}, v_x w u_x u_{x+1}) \equiv_L \{(\overline{u}, u_x u_{x+1}), \perp\}.$$

其次，根据 w，$(u, u_x \overline{u_{x+1} w u_x u_{x+1}}) \not\equiv_L \perp$，因此 z 在 $[(u, u_x u_{x+1})]_{\equiv_L}$上循环。

第三，$[u \cdot u_x u_{x+1}]_{\equiv_L}$ 的每个孙节点都被 z映射到 $\perp$：设 (v, v_x, v_{x+1}) 是带指针的，且 $v \cdot v_x v_{x+1} \equiv_L u \cdot u_x u_{x+1}$。那么再次由引理 6.5，$v \cdot v_x v_{x+1} w \equiv_L u \cdot u_x u_{x+1} w \equiv_L u$。由于 (u, u_x) 是带指针的，我们得到 $(\overline{v}, v_x v_{x+1} w u_x) \equiv_L \{(u, u_x), \perp\}$。如果 $(v, v_x \overline{v_{x+1} w u_x}) \equiv_L \perp$，那么显然，由引理 6.3，$(v, v_x v_{x+1} z) = (v, v_x v_{x+1} w u_x u_{x+1}) \equiv_L \perp$。如果 $(v, \overline{v_x} v_{x+1} w u_x) \equiv_L (u, u_x)$，那么由引理 6.10

$$L^s((\overline{v}, v_x, v_{x+1} z)) = L^s((\overline{v}, v_x, v_{x+1} w u_x u_{x+1})) \subseteq L^s((\overline{u}, u_x, u_{x+1})) = \emptyset.$$

这意味着$(v, v_x, \overline{v_{x+1} z}) \equiv_L \perp$，因此 z 满足中心序列关于 $[(u, u_x u_{x+1})]_{\equiv_L}$的所有性质。◀

现在我们通过证明：在叶节点等价类中，任意带指针的元组再扩展一个分量所得的元组都是 $\perp$，来给出叶状态的刻画。证明只需简单应用引理 6.11。

▶**引理 6.16.** 设 $x \geq 1$ 和 $u \in (\Sigma^*)^x$ 是 带指针的。那么 $[u]_{\equiv_L}$ 是 $\mathcal{A}_{\equiv_L}$ 的叶节点，当且仅当$(u, \varepsilon) \equiv_L \perp$。

证明。设 $u \in (\Sigma^*)^x$ 是带指针的。若 $(u, \varepsilon) \not\equiv_L \bot$，则应用引理 6.11，其中 $u_{x+1} = \varepsilon$。我们得到 $v \in (\Sigma^*)^x$, $v_{x+1} \in \Sigma^*$，使得 (v, v_{x+1}) 是带指针的，且 $v \cdot v_{x+1} \equiv_L u$。因此，$[(v, v_{x+1})]_{\equiv_L}$ 是 $\mathsf{T}^{\equiv_L}_{x+1}$ 中的一个状态，满足 $\mu_x^{\equiv_L}([(v, v_{x+1})]_{\equiv_L}) = [v \cdot v_{x+1}]_{\equiv_L} = [u]_{\equiv_L}$。因此，$[u]_{\equiv_L}$ 不是叶节点。

对于另一个方向，假设 $[u]_{\equiv_L}$ 不是叶节点，并令 $v \in (\Sigma^*)^x$, $v_{x+1} \in \Sigma^*$，使得 (v, v_{x+1}) 是带指针的，且 $v \cdot v_{x+1} \equiv_L u$，也就是说，$\mu_x^{\equiv_L}([(v, v_{x+1})]_{\equiv_L}) = [v \cdot v_{x+1}]_{\equiv_L} = [u]_{\equiv_L}$。由引理 6.10，结合 $u_{x+1} = \varepsilon$，我们得到 $L^s(v, v_{x+1}) \subseteq L^s(u, \varepsilon)$。由于 $L^s(v, v_{x+1}) \neq \emptyset$，我们也有 $L^s(u, \varepsilon) \neq \emptyset$，这意味着 $(u, \varepsilon) \not\equiv_L \bot$。◀

▶ **引理 6.17**。在 $\mathcal{A}_{\equiv_L}$ 层上的所有状态 $x \geq 2$ 都有一个中心序列。

证明。若 $[(u, u_x)]_{\equiv_L}$ 是 $\mathcal{A}_{\equiv_L}$ 的叶节点，那么根据引理 6.16，$(u, u_x, \varepsilon) \equiv_L \bot$，因此 $[(u, u_x)]_{\equiv_L}$ 根据引理 6.15 具有一个中心序列。

为了证明内部状态具有中心序列，设 (v, v_x, v_{x+1}) 是带指针的，并证明其父节点 $[(v, v_x v_{x+1})]_{\equiv_L}$ 具有中心序列，假设 $[(v, v_x, v_{x+1})]_{\equiv_L}$ 具有中心序列 z。我们想要证明可以应用引理 6.15，因此构造一个词 z'，它在 $[(v, v_x v_{x+1})]_{\equiv_L}$ 上循环，并将 (v, v_x, v_{x+1}) 映射到 $\bot$，所以它具有以下性质：

(i) $(\overline{v}, v_x v_{x+1} z') \equiv_L (\overline{v}, v_x v_{x+1})$ and
(ii) $(\overline{v}, v_x, v_{x+1} z') \equiv_L \bot$.

我们可以将引理 6.15 应用于 $(v, v_x, v_{x+1} z')$，从而得到一个中心序列，用于 $[(v, v_x v_{x+1} z')]_{\equiv_L} = [(v, v_x v_{x+1})]_{\equiv_L}$，如所需。

下面用于证明 (i) 和 (ii) 的论证并不困难，但由于涉及多个元组，可能会令人困惑。图 12 概述了下文构造的情形。左侧层 x 上的元组正是我们要为其构造词 z' 的元组。

我们先解释图 12 中的 z-边。由于 z 是 $[(v, v_x, v_{x+1})]_{\equiv_L}$ 的中心，$[(v, v_x, v_{x+1})]_{\equiv_L}$ 的每个子节点都通过 z 被取到 $\bot$（这由图 12 中层 $x+2$ 上的 $[(\cdots)]_{\equiv_L}$ 表示）。此外，我们有 $(v, v_x, v_{x+1} z) \equiv_L (v, v_x, v_{x+1})$，因此根据引理 6.5，也有 $(v, v_x v_{x+1} z) \equiv_L (v, v_x v_{x+1})$ 以及 $v \cdot v_x \overline{v}_{x+1} z \equiv_L v \cdot \overline{v}_x v_{x+1}$（这些就是图 12 中的 z-循环）。于是，$(v, v_x v_{x+1} z) \not\equiv_L \bot$，并且存在 w，使得 $(v, v_x v_{x+1} z w) \equiv_L v$ 且 $\overline{v}_1 \cdots \overline{v}_{x-1} (v_x v_{x+1} z w)^\omega \in L$ 当且仅当 x 为偶数。这就是图 12 中层 $x-1$ 上的 w-边。此外，对 w 的选择保证了 $(v, v_x v_{x+1} (z w v_x v_{x+1})^\omega) \not\equiv_L \bot$。

我们令 $z' := z w v_x v_{x+1}$。z' 在层 $x-1$、x 和 $x+1$ 上诱导的路径如图 12 所示。层上显示的大多数边，只需将相应的词附加到相应元组的最后一个分量即可得到。父边由态射性质得到。层上有两条粗边，其目标等价类由带指针的定义确定。首先，考虑层 x 上的粗 v_x-边。由于 (v, v_x) 是带指针的，且 $v \cdot v_x v_{x+1} z w \equiv_L v$（如图 12 所示），我们得到 $(v, v_x v_{x+1} z w v_x) \equiv_L \{(v, v_x)\} \not\equiv_L \bot$。并且，正如前面所述，$w$ 的选择确保了 $(v, v_x v_{x+1} (z w v_x v_{x+1})^\omega) \not\equiv_L \bot$，因此特别有 $(v, v_x v_{x+1} z w v_x) \not\equiv_L \bot$。于是，我们得到 $(v, v_x v_{x+1} z w v_x) \equiv_L (v, v_x)$，因此也有

$$(\overline{v}, v_x v_{x+1} z') = (\overline{v}, v_x v_{x+1} z w v_x v_{x+1}) \equiv_L (\overline{v}, v_x v_{x+1}).$$

这解释了图 12 中第 x 层的边，并展示了 (i)。

现在我们证明 (ii)。实际上，粗 v_{x+1} 边在层 $x+1$ 上是基于 (ii) 不成立这一假设绘制的，随后用于导出矛盾。由于 (v, v_x, v_{x+1}) 是带指针的，并且 $(v, v_x v_{x+1} z w v_x) \equiv_L (v, v_x)$（如前所述），我们可得

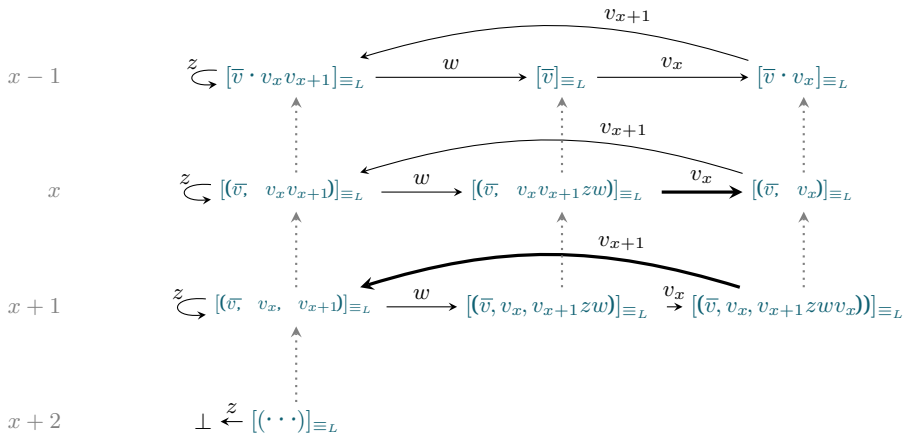


图 12 用于支持引理 6.17 证明中推理的示意图。虚线边表示到父层的态射。在一个会导致矛盾的假设下绘制层 $x+1$ 上的 v_{x+1}-边，因此层 $x+1$ 上的 wv_xv_{x+1}-循环不存在。

$(\overline{v}, v_x, v_{x+1} z w v_x v_{x+1}) \equiv_L \{(v, v_x, \overline{v_{x+1}}) \bot\}$。若 $(v, v_x, \overline{v_{x+1}} z w v_x v_{x+1}) \equiv_L \bot$，则 (ii) 成立。否则，我们得到 $(v, v_x, v_{x+1} z w v_x \overline{v_{x+1}}) \equiv_L (v, v_x, v_{x+1})$，这由层 $\overline{x}+1$ 上的粗v_{x+1}-边表示。现在从图 12 可以看出，在这种情况下，$(z w v_x v_{x+1})^\omega$ 从 $[(v, v_x, v_{x+1})]_{\equiv_L}$ 强接受/拒绝，当且仅当 $x+1$ 为偶数/奇数。根据引理 6.13，这意味着 $v_1 \cdots v_{x+1}(z w v_x \overline{v_{x+1}})^\omega \in L$ 当且仅当 $x+1$ 为偶数。但我们选择 w 时使得 $v_1 \cdots v_{x-1}(v_x v_{x+1} z w)^\omega = v_1 \cdots v_{x+1}(z w v_x v_{x+1})^\omega \in L$ 当且仅当 x 为偶数，这就产生了矛盾。◀

▶**引理 6.18**。 设 L 是一个 ω-正则语言。则 $\mathcal{A}_{\equiv_L}$ 是 安全最小、 正规且中心化。

证明。若分层自动机的第一层对应于 L 的剩余等价，并且对于所有 $x \geq 2$，对于第一层 x 上任意两个具有相同父节点的状态 p、q，其安全语言 p 和 q 均不相同，则该分层自动机是安全最小的。因此，设 (u, u_x) 和 $(v, \overline{v_x})$ 是两个满足 $u \cdot u_x \equiv_L v \cdot \overline{v_x}$ 的带标记元组。由 $\equiv_L$ 的定义可得，$(u, u_x) \equiv_L (v, v_x)$ 当且仅当 $L^s(u, u_x) = L^s(v, v_x)$。因此，层 x 上具有相同父类的两个带标记元组所对应的状态不同，当且仅当它们具有不同的安全语言。

为了证明 $\mathsf{A}_{\equiv_L}$ 是正规的，我们首先需要证明定义 4.2 中的性质 N1，即每一层 $x \geq 2$ 都是非平凡强连通分量的并集。设 (u, u_x) 是层级 x 上的某个带标记元组，并设 $v \in \Sigma^*$ 满足 $(u, u_x v) \not\equiv_L \bot$，也就是说，在 $\mathsf{A}_{\equiv_L}$ 中存在路径 $[(u, \overline{u_x})]_{\equiv_L} \xrightarrow{v} [(u, u_x v)]_{\equiv_L}$。由于 $(u, u_x v) \not\equiv_L \bot$，存在 $w \in \Sigma^+$ 使得 $u \cdot u_x v w \equiv_L u$ 且 $u_1 \cdots u_{x-1}(u_x v w)^\omega \in L$ 当且仅当 x 为偶数。于是 $(u, u_x v w u_x) \not\equiv_L \bot$，并且由于 (u, u_x) 是带指针的，我们得到 $(u, u_x v w u_x) \equiv_L (u, u_x)$，因此在 $\mathsf{A}_{\equiv_L}$ 中得到以下路径 $[(u, u_x v)]_{\equiv_L} \xrightarrow{w u_x} [(u, u_x)]_{\equiv_L}$。

性质 N2 所要求的正规性可由中心序列的存在推出（状态的中心序列具有 N2所要求的性质）。这也意味着 $\mathcal{A}_{\equiv_L}$ 是中心化，这同样可由中心序列的存在推出：如果 $[u]_{\equiv_L} \preccurlyeq_x [v]_{\equiv_L}$ 对于两个长度为$x \geq 2$的元组，那么对于 z，中心序列 $[u]_{\equiv_L}$ 无法将 $[v]_{\equiv_L}$ 映射到 $\bot$，因此 $[v \cdot z]_{\equiv_L} = [u]_{\equiv_L}$，这意味着 $[v]_{\equiv_L} \xrightarrow{z}_x [u]_{\equiv_L}$。根据正规的性质 N1，$[u]_{\equiv_L}$ 和 $[v]_{\equiv_L}$ 属于同一个x-强连通分量。◀

结合引理 6.18、定理 6.14 和定理 4.1，可得本节的主要结果。

▶**定理 6.19**。 设 L 是一个 ω-正则语言。则 $\mathcal{A}_{\equiv_L}$ 是 分层自动机 L的最小者。

7 结论

我们引入了分层自动机，这是一种合并并推广最小 HDcoBüchi 自动机和 Zielonka 树的模型。该模型为ω-正则语言提供了典范表示。我们通过定义在有限词元组上的、与机器无关的同余关系，刻画了这些表示。除了这一基础性刻画之外，一致分层自动机还展现出强大的算法性质。对于一致分层自动机，空性和包含性测试均可在多项式时间内完成。此外，一致分层自动机可以在多项式时间内最小化，并且所得结果从不大于最小的等价确定性自动机。这些性质使一致分层自动机 成为 ω-正则语言的接受器的0-1 概率性模型，因为它们具有良好的算法性质，同时还兼具历史确定性和0-1 概率性。

虽然我们证明了规范的 分层自动机 在一致分层自动机 等价类中是最小的（推论 5.2），但我们猜想，它们在所有交替历史确定性 自动机中也是最小的。

▶**猜想 7.1**。 设 $\mathcal{B}$ 是一个 历史确定性交替奇偶自动机。那么，存在一个与之等价的 分层自动机 $\mathcal{A}$，使得 $[\![\mathcal{A}]\!]$ 的状态数不多于 $\mathcal{B}$。

特别地，一个语言的最小 分层自动机 也是所有历史确定性交替奇偶自动机中最小的。[10]

如果该猜想成立，就可以证明 分层自动机 为所有 d
历史确定性 自动机 提供了一种算法高效的表示方式。

其他未来方向。 显然，分层自动机理论本身还有许多方面有待发展。我们相信，分层自动机为更广泛的实际应用和理论应用提供了一个前景广阔的框架。一个自然的首要方向是，为语言的布尔运算提供高效构造，并将线性时序逻辑等逻辑形式体系直接转换为分层自动机。第二个方向是利用EK24a],中的技术，直接从 COCOA 表示构造最小分层自动机，从而避开第 3.6 节讨论的完整乘积构造。最后，我们设想将分层自动机用于 ω-正则语言的被动学习和主动学习，并拓展针对 coBüchi 自动机的被动学习结果 [LW25]。

[10]我们已有一份猜想 7.1 在优先级为 2（Büchi 和 coBüchi 自动机）时的证明草稿。也就是说，Abu Radi 和 Kupferman 提出的最小 历史确定性 co-Büchi 自动机 [AK22] 在所有HD 交替 coBüchi 自动机中都是最小的。

参考文献

ABF18 Dana Angluin、Udi Boker 和 Dana Fisman。作为 ω-正则语言接受器的确定性有限自动机族。《计算机科学中的逻辑方法》，第 14 卷（第 1 期），2018 年。doi: http://doi.org/10.23638/LMCS-14(1:15)2018。AE25 Bader Abu Radi 和 Rüdiger Ehlers。可在多项式时间内最小化的ω-自动机的刻画。CoRR，第 abs/2504.20553 卷，2025 年。doi: http://doi.org/10.48550/ARXIV.2504.20553。AF16 Dana Angluin 和 Dana Fisman。正则 ω 语言的学习。《理论计算机科学》，第 650 卷：57–72，2016 年。doi: http://doi.org/10.1016/J.TCS.2016.07.031。AK15 Guy Avni 和 Orna Kupferman。加权自动机的随机化。在《2015 年计算机科学数学基础——第 40 届国际研讨会（MFCS 2015）论文集》上，意大利米兰，2015 年 8 月 24–28 日，第一部分，第 9234 卷《计算机科学讲义》，第 89–102 页。施普林格，2015 年。doi: http://doi.org/10.1007/978-3-662-48057-1_7。AK22 Bader Abu Radi 和 Orna Kupferman。游戏适用的基于转移的自动机的最小化与规范化。《计算机科学中的逻辑方法》，第 18 卷，第 3 期：7587，2022 年。doi: http://doi.org/10.46298/lmcs-18(3:16)2022。Arn85 André Arnold。有理ω语言的句法同余。《理论计算机科学》，第 39 卷：333–335，1985 年。doi: http://doi.org/10.1016/0304-3975(85)90148-3。BGB12 Christel Baier、Marcus Grösser 和 Nathalie Bertrand。概率 ω–自动机。《ACM 杂志》，第 59 卷（第 1 期），2012 年。doi: http://doi.org/10.1145/2108242.2108243。BKLS20 Udi Boker、Denis Kuperberg、Karoliina Lehtinen 和 Michal Skrzypczak。交替奇偶游戏适用自动机的简洁性。在 Nitin Saxena 和 Sunil Simon 编，《FSTTCS》，LIPIcs 第 182 卷，第 41:1–41:13 页。达格施图尔城堡——莱布尼茨计算机科学中心，2020 年。doi: http://doi.org/10.4230/LIPICS.FSTTCS.2020.41。BL19 Udi Boker 和 Karoliina Lehtinen。游戏适用自动机：从非确定性到交替。在 Wan J. Fokkink 和 Rob van Glabbeek 编，《CONCUR》，LIPIcs 第 140 卷，第 19:1–19:16 页。2019 年。doi: http://doi.org/10.4230/LIPICS.CONCUR.2019.19。BL24 León Bohn 和 Christof Löding。根据正例和反例构造确定性奇偶自动机。《TheoretiCS》，第 3 卷，2024 年。doi: http://doi.org/10.46298/THEORETICS.24.17。Büc62 J. Richard Büchi。受限二阶算术中的一种判定方法。《国际逻辑、方法论与科学哲学大会论文集》，第 1–11 页，1962 年。Cas22 Antonio Casares。基于转移的 Rabin 自动机的最小化，以及 Muller 条件的色彩记忆需求。在《CSL》，第 216 卷，第 12:1–12:17 页。2022 年。doi: http://doi.org/10.4230/LIPIcs.CSL.2022.12。CCFL24 Antonio Casares、Thomas Colcombet、Nathanaël Fijalkow 和 Karoliina Lehtinen。从 Muller 自动机到奇偶自动机和 Rabin 自动机：保持（历史）确定性的最优变换。《TheoretiCS》，第 3 卷，2024 年。doi: http://doi.org/10.46298/THEORETICS.24.12。CES86 E. M. Clarke、E. A. Emerson 和 A. P. Sistla。利用时序逻辑规范对有限状态并发系统进行自动验证。《ACM 编程语言与系统汇刊》，第 8 卷（第 2 期）：244–263，1986 年。doi: http://doi.org/10.1145/5397.5399。CKS81 Ashok K. Chandra、Dexter C. Kozen 和 Larry J. Stockmeyer。交替。《ACM 杂志》，第 28 卷（第 1 期）：114–133，1981 年。doi: http://doi.org/10.1145/322234.322243。CL08 Thomas Colcombet 和 Christof Löding。非确定性 Mostowski 层级与距离奇偶自动机。在《自动机、语言与编程》第 35 届国际研讨会（ICALP 2008）论文集上，《计算机科学讲义》第 5126 卷，第 398–409 页。施普林格，2008 年。doi: http://doi.org/10.1007/978-3-540-70583-3_33。CNP93 Hugues Calbrix、Maurice Nivat 和 Andreas Podelski。有理 w-语言的最终周期词。在 Stephen D. Brookes、Michael G. Main、Austin Melton、Michael W.

50 分层自动机：无穷词上的自动机的典范模型

Mislove 和 David A. Schmidt（编），《程序语义的数学基础》，第九届国际会议，美国路易斯安那州新奥尔良，1993年4月7日至10日，论文集，计算机科学讲义第802卷，第554–566页。施普林格，1993。doi: http://doi.org/10.1007/3-540-58027-1_27。CO24 Antonio Casares 和 Pierre Ohlmann。位置式 ω-正则语言。载于 LICS，第21:1–21:14页。ACM，2024。doi: http://doi.org/10.1145/3661814.3662087。Col13 Thomas Colcombet。代价正则函数。授权论文（HDR），巴黎第七大学，2013。DJW97 Stefan Dziembowski、Marcin Jurdziński 和 Igor Walukiewicz。赢得无限博弈需要多少记忆？载于 LICS，第99–110页。1997。doi: http://doi.org/10.1109/LICS.1997.614939。Ehl25 Rüdiger Ehlers。重轨自动机。CoRR，第abs/2503.08438卷，2025。doi: http://doi.org/10.48550/ARXIV.2503.08438。EK24a Rüdiger Ehlers 和 Ayrat Khalimov。完全广义反应性（1）综合。载于 Bernd Finkbeiner 和 Laura Kovács（编），《系统构造与分析的工具和算法》，第14570卷，第83–102页。2024。doi: http://doi.org/10.1007/978-3-031-57246-3_6。EK24b Rüdiger Ehlers 和 Ayrat Khalimov。从 LTL 到奇偶性和 COCOA 的自然着色转换。（arXiv:2410.01021），2024。doi: http://doi.org/10.48550/ARXIV.2410.01021。ES22 Rüdiger Ehlers 和 Sven Schewe。《无限词的自然颜色》。LIPIcs，第250卷，FSTTCS 2022，第250卷：36:1–36:17，2022。doi: http://doi.org/10.4230/LIPICS.FSTTCS.2022.36。FAA+25 Nathanaël Fijalkow、C. Aiswarya、Guy Avni、Nathalie Bertrand、Patricia Bouyer、Romain Brenguier、Arnaud Carayol、Antonio Casares、John Fearnley、Paul Gastin、Hugo Gimbert、Thomas A. Henzinger、Florian Horn、Rasmus Ibsen-Jensen、Nicolas Markey、Benjamin Monmege、Petr Novotný、Pierre Ohlmann、Mickael Randour、Ocan Sankur、Sylvain Schmitz、Olivier Serre、Mateusz Skomra、Nathalie Sznajder 和 Pierre Vandenhove。《图上的博弈：从逻辑和自动机到算法》，2025。doi: https://arxiv.org/abs/2305.10546。即将由剑桥大学出版社出版。GTW02 Erich Grädel、Wolfgang Thomas 和 Thomas Wilke（编）。《自动机、逻辑与无限博弈》。施普林格，柏林、海德堡，2002。doi: http://doi.org/10.1007/3-540-36387-4。HK02 Markus Holzer 和 Barbara König。关于确定性有限自动机与句法幺半群的大小。载于《语言理论进展》，DLT 2002，计算机科学讲义第2450卷，第258–269页。施普林格，2002。doi: http://doi.org/10.1007/3-540-45005-X_22。HP06 Thomas A. Henzinger 和 Nir Piterman。无需确定化求解博弈。载于《计算机科学逻辑》，第395–410页。2006。doi: http://doi.org/10.1007/11874683_26。HPS+20 Ernst Moritz Hahn、Mateo Perez、Sven Schewe、Fabio Somenzi、Ashutosh Trivedi 和 Dominik Wojtczak。用于概率分析和强化学习的对MDP有利的自动机。载于 Armin Biere 和 David Parker（编），TACAS，计算机科学讲义第12078卷，第306–323页。2020。doi: http://doi.org/10.1007/978-3-030-45190-5_17。HPT25 Thomas A. Henzinger、Aditya Prakash 和 K. S. Thejaswini。用随机性消解非确定性。载于 Pawel Gawrychowski、Filip Mazowiecki 和 Michal Skrzypczak（编），MFCS，LIPIcs第345卷，第57:1–57:18页。2025。doi: http://doi.org/10.4230/LIPICS.MFCS.2025.57。IL25 Olivier Idir 和 Karoliina Lehtinen。利用博弈和通用树刻画树语言的非确定性指标。载于第52届自动机、语言与程序设计国际研讨会，ICALP 2025，LIPIcs第334卷，第160:1–160:19页。

达格施图尔城堡 - 莱布尼茨信息学中心，2025。doi: http://doi.org/10.4230/LIPICS.ICALP.2025.160。Kla94 Nils Klarlund。关于 ω-正则性的同态概念。载于《计算机科学逻辑》，第八届国际研讨会，CSL '94，卡齐米日，波兰，1994年9月25日至30日，精选论文，计算机科学讲义第933卷，第471–485页。施普林格，1994。doi: http://doi.org/10.1007/BFb0022276。KMS02 Nils Klarlund、Anders Møller 和 Michael I. Schwartzbach。MONA 实现的奥秘。《国际计算机科学基础杂志》，第13(4)卷：571–586，2002。doi: http://doi.org/10.1142/S012905410200128X。KS15 Denis Kuperberg 和 Michał Skrzypczak。关于游戏适用自动机的确定化。载于 ICALP，第299–310页。2015。doi: http://doi.org/10.1007/978-3-662-47666-6_24。KSV96 Orna Kupferman、Shmuel Safra 和 Moshe Y. Vardi。词自动机与树自动机的关联。载于 LICS，第 322–332 页。1996。doi: http://doi.org/10.1109/LICS.1996.561360。Kup18 Orna Kupferman。自动机理论与模型检验。载于 Edmund M. Clarke、Thomas A. Henzinger、Helmut Veith 和 Roderick Bloem（编），《模型检验手册》，第107–151页。施普林格国际出版，2018。doi: http://doi.org/10.1007/978-3-319-10575-8_4。LW25 Christof Löding 和 Igor Walukiewicz。最小历史确定性 co-Büchi 自动机：同余关系与被动学习。载于 LICS，第431–443页。IEEE，2025。doi: http://doi.org/10.1109/LICS65433.2025.00039。Mar75 Donald A. Martin。博雷尔确定性。《数学年刊》，第102(2)卷：363–371，1975。MP90 Zohar Manna 和 Amir Pnueli。时序性质层级。载于 Cynthia Dwork（编），PODC，第377–410页。ACM，1990。doi: http://doi.org/10.1145/93385.93442。MS97 Oded Maler 和 Ludwig Staiger。关于 ω-语言的句法同余。《理论计算机科学》，第183(1)卷：93–112，1997。doi: http://doi.org/10.1016/S0304-3975(96)00312-X。形式语言理论。PP95 Dominique Perrin 和 Jean-Éric Pin。半群与无限词上的自动机。载于《半群、形式语言与群》（约克，1993），第933卷，第49–72页。Kluwer 学术出版社，1995。PPS+25 Soumyajit Paul、David Purser、Sven Schewe、Qiyi Tang、Patrick Totzke 和 Di-De Yen。借助机遇消解非确定性。载于 Patricia Bouyer 和 Jaco van de Pol（编），CONCUR，LIPIcs第348卷，第32:1–32:22页。Schloss Dagstuhl - Leibniz-Zentrum für Informatik，2025。doi: http://doi.org/10.4230/LIPICS.CONCUR.2025.32。PR89 Amir Pnueli 和 Roni Rosner。关于反应式模块的综合。载于 POPL，第179–190页。1989。doi: http://doi.org/10.1145/75277.75293。Sch10 Sven Schewe。超最小化之外——确定性 Büchi 自动机和确定性优先自动机的最小化是 NP 完全。载于 FSTTCS，第8卷，第400–411页。2010。doi: http://doi.org/10.4230/LIPIcs.FSTTCS.2010.400。Tar72 Robert Tarjan。深度优先搜索与线性图算法。《SIAM 计算杂志》，第1(2)卷，1972。doi: http://doi.org/10.1137/0201010。Wil91 Thomas Wilke。关于无限语言的艾伦伯格定理。载于 ICALP，第510卷，第588–599页。1991。doi: http://doi.org/10.1007/3-540-54233-7_166。Zie98 Wiesław Zielonka。有限着色图上的无限博弈及其在无限树上的自动机中的应用。《理论计算机科学》，第200(1-2)卷：135–183，1998。doi:http://doi.org/10.1016/S0304-3975(98)00009-7。