arXiv:2601.04902v1 [cs.FL] 8 Jan 2026

单时钟综合问题

Sławomir Lasota华沙大学
Mathieu Lehaut华沙大学
Julie Parreaux雷恩大学、
Inria、法国国家科学研究中心、
IRISA**Radosław**
Piórkowski牛津大学

摘要

我们研究了 Büchi-Landweber 博弈在定时场景中的一种推广。获胜条件由非确定性定时自动机指定，且其中一名玩家可以让时间流逝。我们系统地研究了定时博弈各种变体中的综合问题，具体取决于获胜条件由哪名玩家指定，以及要寻找哪名玩家的策略（或控制器，即有限记忆策略）。

作为主要结果，我们证明，在所有变体中，无论是策略综合还是控制器综合，都普遍存在不可判定性；即使获胜条件由单时钟自动机指定，结论仍然成立。这强化并推广了此前已知的不可判定性结果。我们还完整刻画了有限记忆足以获胜的情形，即存在策略就意味着存在控制器。

我们的所有结果均在定时 设置下表述，而在数据 设置下也有类似结果，此时单时钟自动机被单寄存器自动机所取代。

2012年ACM学科分类 计算理论 → 定时与混合模型；计算理论 → 无限对象上的自动机；计算理论 → 定量自动机；计算理论 → 逻辑与验证

关键词和短语 定时自动机、寄存器自动机、Büchi-Landweber 博弈、Church 综合问题、反应式综合问题

数字对象标识符 10.4230/LIPIcs.CVIT.2016.23

资助 *Sławomir Lasota*：部分由欧洲研究理事会 INFSYS 项目资助（协议编号 950398），并由波兰国家科学中心项目 2024/55/B/ST6/01674 资助。*Mathieu Lehaut*：部分由波兰国家科学中心项目 2021/41/B/ST6/00535 资助。*Julie Parreaux*：部分由法国国家研究署 BisoUS 项目（ANR-22-CE48-0012）资助。

© Sławomir Lasota、Mathieu Lehaut、Julie Parreaux、Radosław Piórkowski；根据知识共享署名4.0国际许可协议授权
第42届重要主题会议（CVIT 2016）。编辑：John Q. Open 和 Joan R. Access；文章编号：23；页码：23:1–23:30 莱布尼茨国际信息学论文集 Schloss Dagstuhl – Leibniz-Zentrum für Informatik，达格施图尔出版社，德国

1 引言

非确定性定时自动机（NTA）是应用最广泛的实时反应式系统模型之一，相关文献极为丰富。它由扩展了实值时钟的有限自动机构成，这些时钟可以重置，并可通过不等式约束进行比较。自开创性论文证明可达性问题具有 PSpace 完全性以来 [2], ，NTA 已应用于定时系统的自动验证，并最终催生了 UPPAAL [5], 、UPPAAL Tiga（定时博弈） [11], 和 PRISM（概率定时自动机） [24]等成熟工具。该模型的一项限制是，NTA 语言无法确定化，也不具备对补运算的闭包性；此外，普遍性／包含问题也不可判定 [2]。对于只有一个时钟的 NTA 这一受限子类（NTA_1） [30], ，可恢复可判定性；对于只有一个时钟的交替定时自动机也是如此 [25]。

确定性定时自动机 (DTA) 构成了非确定性定时自动机的一个真子类，其中下一个配置由当前配置和定时输入符号唯一确定。此类自动机具有更强的性质，例如包含问题可判定且可取补集 [2],，并应用于若干领域，如测试生成 [29], 、故障诊断 [8],、学习 [36, 35]、定时博弈 [3, 22, 9], 以及定时语言的可识别性 [26]。

定时博弈与综合问题。 文献中有许多不同形式的定时博弈，具体取决于玩家是否必须确保对局非 Zeno、由谁控制时间流逝、是否存在并发动作、获胜条件如何设定等 [3, 22, 9, 37, 27, 4, 16, 14, 32, 21]。遵循先前的工作 [13, 31], ，我们考虑非对称（只有一名玩家可以让时间流逝）、无限时长、轮流进行的博弈，可将其视为 Büchi-Landweber 博弈的定时推广 [10]。有两名玩家，分别称为计时器和监视器，他们严格交替轮流行动。在第 i轮中，计时器从有限字母表中选择一个字母 a_i ，并选择一个非负有理数时间延迟 τ_i；监视器则从有限字母表中回应一个字母 b_i 。两名玩家共同生成一场无限对局$\pi = (a_1,\ b_1,\ \tau_1)(a_2,\ b_2,\ \tau_2)\cdots$。计时器或监视器的获胜集合由一个非确定性 定时自动机指定。作为比较，先前已有研究探讨较简单的特殊情形：获胜集合由确定性 或历史确定性 定时自动机给出（例如， [16, 7]）。鉴于我们的不可判定性结果，我们不考虑更一般的对称变体，即双方都可以让时间流逝。

我们研究定时反应式综合问题：该问题要求判断给定玩家是否拥有一种策略，使得每一场遵循该策略的对局对该玩家而言都是获胜的。根据指定的是哪位玩家的获胜集合（注意非确定性定时自动机语言不具备补集闭包性质），以及要为哪位玩家寻找获胜策略，我们区分了该问题的四种变体。我们还研究定时 Church 综合问题：该问题要求判断给定玩家是否拥有一种获胜的有限记忆策略（控制器）。对于监视器而言，控制器是一个确定性定时自动机，其迁移输出来自监视器的字母表；对于计时器而言，控制器是一个确定性有限自动机，其迁移输出来自计时器的字母表以及时间延迟。

对布奇–兰德韦伯博弈的定时推广的研究始于 [13] ，当时考虑的是这样一种情形：计时器的获胜集合由非确定性定时自动机指定，目标是为监视器综合一个控制器。该研究的主要结果是，资源受限 的定时 Church 综合问题是可判定的；该问题旨在为固定的 k，综合一个至多使用 k 个时钟的控制器。随后， [31] 证明了不受限制（资源不受限）版本不可判定，即使计时器的获胜集合由双时钟非确定性定时自动机给出，且目标是为监视器综合控制器也是如此。对于非确定性定时自动机$_1$ （单时钟非确定性定时自动机）获胜集合，这一问题的可判定性仍悬而未决，其他变体以及定时反应式综合的状况也同样如此。本文以否定结果解决了所有这些悬而未决的问题。

贡献。 面对多达八种不同的判定问题，人们可能会认为其可判定性与复杂性图景会十分复杂。作为主要技术贡献，我们证明事实并非如此：即使在最简单的情形下，也就是获胜集合由非确定性定时自动机$_1$指定时，这八种判定问题全都不可判定。这些不可判定性结果显著加强并推广了先前已知的下界。它们还表明，将范围限制为非确定性定时自动机$_1$ 并不能使综合（博弈求解）问题恢复可判定性；同样，将范围限制为非确定性定时自动机$_1$ ，也不能使无限字上的语言普遍性及相关问题变得可判定（与非确定性定时自动机$_1$ 所接受的有限定时词语言上的普遍性及相关问题不同） [30]。

证明八个问题不可判定需要四种归约。对于监视器的获胜集合由非确定性定时自动机$_1$指定的情况，通过从非确定性定时自动机$_1$ 普遍性问题和采样普遍性问题 [1, 25]进行归约，可以轻松证明其不可判定。另一种情况，即计时器的获胜集合由非确定性定时自动机指定时，其不可判定性证明构成了本文的技术核心；证明通过从有损计数器机的两个不可判定问题——有界性和重复可达性 [28, 33]——进行归约来实现。

最后，我们还探讨有限记忆何时足以获胜，也就是说，策略的存在何时意味着控制器的存在。一方面，我们证明获胜集合由该玩家指定的玩家具有有限记忆性质：当该玩家是监视器时，所有非确定性定时自动机$_1$ 规格都具有此性质；当该玩家是计时器时，则只有在受限的可达性 非确定性定时自动机$_1$ 规格下才具有此性质。另一方面，我们证明在其他所有情况下，有限记忆性质均不成立：当计时器指定自己的获胜集合时，通常不满足该性质；当对手指定获胜集合时，无论是计时器还是监视器都不满足该性质，即使在可达性非确定性定时自动机$_1$ 规格下也是如此。

众所周知，NTA 与非确定性寄存器自动机（NRA）非常相似，后者也称为有限记忆自动机 [23] （例如，参见 [20] ，其中正式讨论了这两种模型之间的联系）。寄存器自动机以数据值（而非时间戳）作为输入，并使用寄存器（而非时钟）存储数据值（[34] 是关于数据场景下自动机模型的一篇优秀综述）。对于许多与语言相关的问题，例如空性（可达性）、普遍性或包含，寄存器自动机与定时自动机已知的（不可）判定性结果完全相同 [15] [30, 25]。两种场景下的语言闭包性质也类似。进一步印证这些深层相似之处的是， [31] 证明了：对于由具有两个时钟的 NTA 指定获胜集合的定时 *Church* 综合 问题，以及对于由具有两个寄存器的 NRA 指定获胜集合的数据 *Church* 综合问题，二者均不可判定。 [17] 也独立开展了关于 Büchi-Landweber 博弈的数据推广的类似研究，相关文献见其中引用的作品 [19, 18]。我们的所有不可判定性结果都可从定时场景推广到数据场景：对于数据响应式/ *Church* 综合 问题的八种变体，不可判定性均成立；即使获胜集合由 NRA 指定也已如此$_1$（即仅使用一个寄存器的 NRA）。（这些进一步的结果超出了本文的讨论范围，计划收入本文即将发表的完整版中。）

概述。 我们首先定义定时博弈的设定和综合问题（第 2 节），并在该节讨论有限记忆性质。第 3 节是预备部分，我们在其中处理综合问题的简单情形（给定监视器的获胜集合）。随后，在第 4 节中，我们定义有损计数器机的不可判定问题，并将在论文的主要技术部分，即第 5 节和第 6 节中使用这些问题。在后两节中，我们针对较难的情形给出不可判定性证明，即给定计时器的获胜集合。最后一节包含总结性评论。

2 定时综合问题

设 $\mathbb{Q}_{\geq 0}$ 和 $\mathbb{N}_{>0}$ 分别为非负有理数和正整数。令 X 为一个有限变量集，称其元素为时钟。赋值是一个映射 ν: X $\to \mathbb{Q}_{\geq 0}$（我们更倾向于使用有理数值而非实数值）。对于赋值 ν、延迟 $\tau \in \mathbb{Q}_{\geq 0}$ 和时钟的子集 $Y \subseteq$X，我们定义赋值 $\nu + \tau$ 为 $(\nu + \tau)(x) = \nu(x) + \tau$，对所有 $x \in$ X 均成立；并定义赋值 $\nu[Y := 0]$ 为 $(\nu[Y := 0])(x) = 0$（若 $x \in Y$），否则为 $(\nu[Y := 0])(x) = \nu(x)$。时钟 X 上的守卫是由形如 $x \bowtie c$ 的原子约束合取而成，其中 $x \in$X、$\bowtie \in \{\leq, <, =, >, \geq\}$ 且 $c \in \mathbb{N}$。空守卫记作 $\top$。若 $\nu(x) \bowtie c$，则赋值 ν 满足原子约束 $x \bowtie c$。满足关系自然地扩展到所有守卫 g，并记作 $\nu \models g$。令 Guard(X) 表示 X 上守卫的集合。

设 Σ 为有限字母表。我们将 Σ上的一个定时词 定义为有限或无限序列 $w = (a_1, t_1)(a_2, t_2) \cdots \in (\Sigma \times \mathbb{Q}_{\geq 0})^\omega$，其中时间戳 t_i 是单调的：$t_1 \leq t_2 \leq t_3 \leq \cdots$。有时将二元组 $(a, t) \in \Sigma \times \mathbb{Q}_{\geq 0}$ 称为定时字母。除非另有说明，我们处理无限定时词。时间戳序列由延迟 $\tau_1, \tau_2, \ldots \in \mathbb{Q}_{\geq 0}$的序列唯一确定，其中 $\tau_i = t_i - t_{i-1}$（假设 $t_0 = 0$）。对 w进行去定时化 所得的结果记为 untime(w)，即删除时间戳后得到的词 $a_1 a_2 \ldots \in \Sigma^\omega$。定时语言是在固定字母表 Σ上任意定时词的集合。

定时自动机。非确定性定时自动机（NTA）A $= L,$ X Σ, L_i, L_f, Δ由 L组成：一个有限的位置集，其中 $L_i, L_f \subseteq L$ 分别表示初始位置集和接受位置集；X，即有限时钟集； Σ，即有限字母表；以及 $\Delta \subseteq L \times \Sigma \times$ Guard(X) $\times 2^{\mathcal{X}} \times L$，即有限转移集。转移 $(\ell, a, g, Y, \ell') \in \Delta$记作 $\ell \xrightarrow{a, g, Y} \ell'$，当 g 和 Y 分别为 $\top$ 或 ∅时，可省略它们。用 a 代替 A 表示一组转移。A $\in$ NTA 的语义是一个定时迁移系统 $[\![$A$]\!] = (Q, Q_I, \to)$，其定义如下： $Q = L \times (\mathbb{Q}_{\geq 0})^{\mathcal{X}}$是配置集（即位置—赋值对）； $Q_I = L_i \times \{0^{\mathcal{X}}\}$是初始配置集； $\to$是边集。若且唯若，存在 A 的一个转移 $\delta = (\ell, a, g, Y, \ell') \in \Delta$ 满足 $\nu + \tau \models g$，且 $\nu' = (\nu + \tau)[Y := 0]$，则有 $(\ell, \nu) \xrightarrow{\delta, \tau} (\ell', \nu')$。A$]\!]$ 中的运行 ρ 是边序列 $\rho = (\ell_1, \nu_1) \xrightarrow{\delta_1, \tau_1} (\ell_2, \nu_2) \xrightarrow{\delta_2, \tau_2} \ldots$，并满足 $\ell_1 \in L_i$。定时词$w = (a_1, t_1)\ (a_1, t_2)$ .....在 Σ 上标记 A 的运行 ρ，当且仅当对所有 $i \in \mathbb{N}_{>0}$，$\delta_i = (\ell_i, a_i, g, Y, \ell_{i+1})$ 且 $t_i = \sum_{j=1}^{i} \tau_j$。我们采用 Büchi 接受条件：若一次运行无限多次访问接受位置，则称该运行为接受运行；此时，标记该运行的词称为被 A 接受。A $\in$ NTA 的语言记作 L(A)，是 A 所接受的定时词集合。

我们还采用<style id='1'>可达性接受条件：一旦运行至少访问过一次接受位置（类似于有限词的情形），该运行就被视为接受。在这种接受条件下，NTA 所接受的<style id='4'>无限定时词的语言L 由那些存在以接受位置结束的运行的 L' 有限定时词的语言唯一确定。我们记作 L =Reach(L')。如此定义的<style id='11'>可达性<style id='13'>NTA可以看作 NTA 的一个（严格）子类，记作<style id='15'>reach-<style id='17'>NTA。

除了 reach-NTA 之外，我们还考虑 NTA 的其他子类。例如，NTA$_k$由具有固定数量 $k = |\mathcal{X}|$ 个时钟的非确定性定时自动机构成。DTA 是<style id='6'>确定性的定时自动机类，其中，对于所有位置 $\ell \in L$ 和字母 $a \in \Sigma$，出现在形如 $(\ell, a, g, _, _)$ 的迁移中的守卫 g 构成 $\mathbb{Q}_{\geq 0}^{|\mathcal{X}|}$ 的一个划分。我们还考虑 NTA$^{\text{res}}$ 的一个超类，即<style id='27'>1-重置定时自动机（定义见第 6 节）；它通过一种有限形式的 ε-迁移扩展而来，这种迁移在时钟每次等于 1 时将其重置。带有 ε-迁移的 NTA 类的表达能力严格强于 NTA [6]；我们将在本节末尾以及第 6 节和第 7 节讨论这一选择。

我们结合子类和超类的记号，并将 reach-NTA$_1$, reach-NTA$_1^{\text{res}}$ 写作

表示 reach-NTA 中的单时钟自动机，以及其 1-重置扩展。

给定 L 一个定时语言，我们用 L 表示它的补集。普遍性问题询问：给定$\mathcal{A}$，是否有$\mathcal{L}(\mathcal{A}) = \emptyset$。已知对于非确定性定时自动机，该问题是不可判定的$_1$ [25]。

定时博弈。本文研究两名玩家之间的轮流行动博弈，分别称为计时器和监视器，其中获胜条件由非确定性定时自动机给出$_1$ （分别为非确定性定时自动机$_1^{\text{res}}$）。

▶ **定义 1**。一个定时博弈G =⟨T、M、A、所有者、代理⟩由有限字母表T和M组成，分别属于计时器和监视器；A ∈ 非确定性定时自动机$_1$ （分别为A ∈ 非确定性定时自动机$_1^{res}$）是定义在乘积字母表 T ×M 上的定时自动机，其语言 $L(A)$ 定义了获胜条件；并且所有者、代理 ∈{计时器、监视器}。

直观地说，如果一场对局属于$\mathcal{L}(\mathcal{A})$，那么 Owner 就是该对局的获胜者。由于非确定性定时自动机对补集不封闭， [2], 获胜条件的所有者选择并非无关紧要，而且对手的获胜条件可能不再是非确定性定时自动机语言。我们研究关于代理是否存在获胜策略的判定问题。同样，代理的选择也并非无关紧要，因为我们不知道所研究的博弈是否具有确定性。鉴于我们的不可判定性结果，我们不考虑定时博弈的一种对称变体，即双方都提交时间延迟；因为这种变体会推广上述博弈。

定时博弈按回合进行。每个第 i 回合（$i \in \mathbb{N}_{>0}$）都由计时器选择 $a_i \in$ T 和一个延迟 $\tau_i \in \mathbb{Q}_{\geq 0}$ 开始，随后监视器作出响应 $b_i \in$ M。由此产生一场对局，它是字母表 T × M 上的一个定时词（a_1, b_1, t_1） （a_2, b_2, t_2） ⋯，其中$t_i = \sum_{j=1}^{i} \tau_j$。如果对局属于$\mathcal{L}(\mathcal{A})$，则 Owner 获胜；否则由其对手获胜。

玩家的策略是一个函数，它根据对手迄今为止所走的步，给出该玩家的下一步。形式上，计时器的策略是一个函数 σ_{T}: M* → T × $\mathbb{Q}_{\geq 0}$而监视器的策略是一个函数 σ_{M}: (T × $\mathbb{Q}_{\geq 0}$)$^+$ → M。我们称一局对局$w = (a_1, b_1, t_1)\ (a_2, b_2, t_2) \ldots$遵循计时器的策略 σ_{T} （分别地，监视器的策略 σ_{M}），如果其中所有（带时间的）字母都遵循该策略的输出：$(a_i, \tau_i) = \sigma_{\text{T}}(b_1 \cdots b_{i-1})$ 对所有 $i \in \mathbb{N}_{>0}$都成立（分别地， $b_i = \sigma_{\text{M}}((a_1, t_1) \cdots (a_i, t_i))$ 对所有 $i \in \mathbb{N}_{>0}$都成立）。当且仅当每一局 w 遵循该策略的对局都对该玩家而言是胜局时，该策略才是该玩家的制胜策略（即， $w \in$L(A) 当且仅当该玩家是所有者）。

控制器是一种有限记忆策略，由带输出的定时自动机表示。具体而言，监视器的控制器是一个来自 M 的带输出定时自动机，其迁移形式为$\delta = (\ell, a, g, Y, \ell', b)$，其中 $b \in$ M 是输出。该控制器导出策略 σ_{M}，将 $w = (a_1, t_1) \cdots (a_i, t_i)$ 映射为在 w 上运行的最后一个输出。对于计时器，控制器的概念更为复杂，因为它需要生成时间戳。我们令计时器的控制器输出对 $(a, \tau) \in$ T × $\mathbb{Q}_{\geq 0}$，其中 τ被解释为下一次延迟。形式上，它定义为一个来自 T × $\mathbb{Q}_{\geq 0}$ 的带输出确定性有限自动机（不带时钟的定时自动机），其迁移形式为 $\delta = (\ell, b, \ell', a, \tau)$，其中 $(a, \tau) \in$T × $\mathbb{Q}_{\geq 0}$是输出，此外还配备一个初始动作 $(a_1, \tau_1) \in$ T × $\mathbb{Q}_{\geq 0}$。请注意，与一般策略相比，计时器控制器存在一个显而易见的限制，即控制器使用的延迟集合是有限的。控制器导出策略 σ_{T}，将空词映射为 (a_1, τ_1)，并将非空词 $b_1 \cdots b_i$ 映射为在 w 上运行的最后一个输出。

定时综合问题。 我们研究判定问题，以确定代理是否获胜，也就是说，代理是否拥有获胜策略。根据所有者 ∈{计时器，监视器}和代理 ∈{计时器，监视器}的选择，我们区分四种情形。此外，在这四种情形中，我们分别考虑两个不同的判定问题：给定一个定时博弈，

- 定时反应式综合定时反应式综合询问代理是否拥有获胜策略；

	代理 =计时器	代理 =监视器
所有者 = 计时器	对于 $\mathcal{A}$ ∈reach-NTA 一致，但并非对于所有 $\mathcal{A}$ ∈非确定性定时自动机$_1$	不重合，即使对于 $\mathcal{A}$ ∈ reach-NTA$_1$
所有者 = 监视器	不重合，即使对于 $\mathcal{A}$ ∈ reach-NTA$_1$	对于所有 $\mathcal{A}$ ∈ 非确定性定时自动机均重合

表1 反应式综合与丘奇综合。

- 定时 Church 综合 询问代理是否拥有获胜控制器。

在某些情况下，这些问题是等价的（总结见表1）。最重要的是，如果监视器既是获胜条件的所有者，也是综合问题中的代理，那么只要它有获胜策略，就也有控制器。计时器的情况也类似，但仅限于受限的 reach-NTA$_1$ 获胜条件：

▶**定理 2.** 在以下情况下，这两个综合问题是等价的，也就是说，如果代理有获胜策略，那么它就有获胜控制器：
1. 所有者 = 代理 = 计时器和$\mathcal{A} \in reach\text{-}NTA_1$。
2. 所有者 = 代理 = 监视器和$\mathcal{A} \in NTA_1$。

证明。 我们给出该定理所述两种情况的证明。在这两种情况下，有获胜控制器都意味着有获胜策略，因此只需证明反方向即可。

第一种情况的证明：所有者 = 代理 = 计时器

▶ **引理3。** 设 $G = \langle$T、M、A、计时器、计时器$\rangle$ 是一个博弈，其中 $A \in$ 是 $reach\text{-}NTA_1$。如果计时器具有获胜策略 σ_{T}，那么它具有获胜控制器。

证明。 我们考虑$\mathcal{G}$ 中所有符合 σ_{T}的对局所构成的树。存在 计时器 节点只有一个子节点（即在当前对局中由 $\mathrm{T} \times \mathbb{Q}_{\geq 0}$ 输出的唯一序对），而监视器 节点则对于M中的每个字母都有一个子节点。由于我们考虑的是无限对局，因此这棵树也是无限的。我们将证明，只需考虑这棵树的有限部分，就能构造出一个获胜控制器。

根据树上的柯尼希引理，这棵树中存在一条无限分支。特别地，存在一个符合 σ_{T} 的无限词 w，并且由于 σ_{T} 是获胜的，该词属于$\mathcal{L}(\mathcal{A})$。由于$\mathcal{A}$ ∈是 reach-NTA，因此存在 w 的一个有限前缀 w'，使得 w' 的任意延续也属于$\mathcal{L}(\mathcal{A})$。此外，w' 符合 σ_{T}，因为它是 w 的前缀。考虑一棵新树，在 w'之后截断该分支。我们对所有无限分支重复这一过程，直到只剩下一棵有限树为止。这棵树中的所有分支都是符合 σ_{T} 的对局，并且任何延续都会被$\mathcal{A}$接受。我们构造一个控制器$\mathcal{A}$，只要对局仍在这棵有限树中，就让它模仿 σ_{T}；离开这棵树后则任意行动。因此，我们得到一个有限的获胜控制器。◀

第二种情况的证明：所有者 = 代理 = 监视器

▶ **引理 4。** 设 $G = \langle$T, M, A,监视器, 监视器$\rangle$ 是一个博弈，且 $A \in$ 为非确定性定时自动机$_1$。如果监视器有获胜策略 σ_{M}，那么它就有一个获胜控制器。

本节剩余部分将致力于证明。直观地说，我们考虑与 σ_{M} 一致的 $\mathcal{G}$ 的对局树。与前一种情况不同，这棵树具有无界

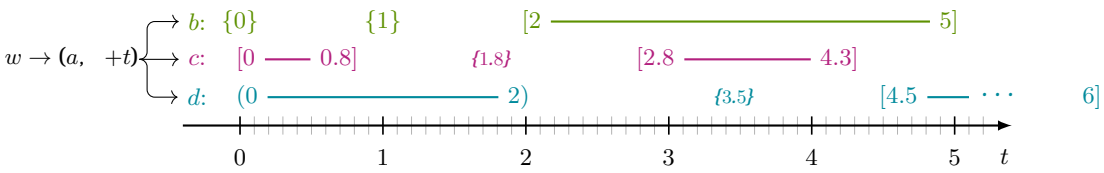


图1 一个示例，展示了当计时器以 $t \in \mathbb{Q}_{\geq 0}$的延迟进行 a时，来自 $\mathrm{M} = \{b, c, d\}$ 的一个字母作为获胜响应的区间结构。请注意，由于监视器拥有获胜策略，如果它遵循该策略，那么在任何时刻，对于所有延迟 $t \in \mathbb{Q}_{\geq 0}$，它都将拥有一个获胜响应。

分支：每个计时器节点针对每一对 $(a, t) \in \mathrm{T} \times \mathbb{Q}_{\geq 0}$ 都有一个子节点。此外，σ_{M} 对不同的 (a, t) 对作出的响应可能表现得高度不一致。

直观地说，我们利用获胜条件$\mathcal{A}$ 只能区分有限多个点这一事实。这表明，应该存在一种更加统一的获胜策略：对于较大的可能时间戳区间，采取相同的步，仅在少数几个特别区分出来的单点集上例外。我们将说明如何利用$\mathcal{A}$的区域构造这种策略，以及如何界定实现它所需的时钟数和记忆量，从而为监视器构造一个控制器。

步骤 1：将时钟数缩减为有限个。 策略先验上拥有无限多个时钟：底层树的每一层对应一个时钟。要限制此类策略中的时钟数，就要让监视器在某个时间戳区间上的响应更加统一：为此，我们采用区域机制。更具体地说，带有 k个时钟的策略只使用k 个时钟来监测时间并据此作出决策。形式上，带有 k个时钟的策略 是确定性定时自动机$_k$ 的一种扩展，但其位置数（因此转移数）可能是无限的。

为了限制时钟的数量，我们根据计时器在固定双方各自所出两个字母（a, b）$\in \mathrm{T} \times \mathrm{M}$ 时所选择的时间戳，为获胜配置定义一个等价关系。给定一个定时词 w、两个字母（a, b）和一个时间戳 $t \in \mathbb{Q}_{\geq 0}$，如果存在一种策略，能够从 $w \cdot (a, b, t)$ 获胜，我们就称 (a, b, t) 是 w 的获胜延续。于是，对固定的 (a, b)，我们将获胜时间戳集合 t 划分为有限个长度大于 1 的区间的并、有限个单点集的并，以及至多一个从 w 的最后一个时间戳开始且长度小于 1 的区间（如图1所示）。特别地，我们得到以下引理。

▶**引理 5.** 如果 σ_{M} 对 监视器 在 G 中是获胜的，那么存在一种获胜策略，使 监视器 至多使用 k 个时钟，其中 $k = 2|\mathrm{T}| \cdot K$，且 K 是 A 的守卫中的最大常数。

为了证明前一个引理，我们首先根据获胜时间戳区间的大小，引入两个字母的紧和大延续这一概念。

▶ 定义6。设 $w \in (T \times M \times \mathbb{Q}_{\geq 0})^*$，且 $(a, b, t) \in T \times M \times \mathbb{Q}_{\geq 0}$ 是一个定时字母。*1.* w 对监视器而言是获胜的，当且仅当存在一种获胜策略，使监视器能够完成 w。*2.* 当 $w \cdot (a, b, t)$ 对监视器而言是获胜的时，(a, b, t) 就是 w 对监视器而言的获胜延续。

令 $\mathrm{winCont}^{\mathrm{M}}_{\mathcal{A}}(w)$ 表示 w 对于监视器的获胜延续集合，w 为一个

监视器的获胜词，且 $(a,\ b,\ t)$ 为 w 对于监视器的一个获胜延续，我们固定

$$t_{\inf}(w,a,b,t) = \inf\{t' \mid \forall t'' \in [t',t], (a,b,t'') \in \mathrm{winCont}^{\mathrm{M}}_{\mathcal{A}}(w)\}$$
$$t_{\sup}(w,a,b,t) = \sup\{t' \mid \forall t'' \in [t,t'], (a,b,t'') \in \mathrm{winCont}^{\mathrm{M}}_{\mathcal{A}}(w)\}$$
$$\mathrm{size}(w,a,b,t) = t_{\sup}(w,a,b,t) - t_{\inf}(w,a,b,t)$$

现在，我们可以定义定时词的紧延续和大延续。

▶ **定义 7.** 设 w 是关于$(\mathrm{T} \times \mathrm{M})$ 的一个获胜定时词，且 $(a,b,t) \in \mathrm{T} \times \mathrm{M} \times \mathbb{Q}_{\geq 0}$ 是关于 w 的一个获胜延续。若延续(a,b,t) 满足$\mathrm{size}(w,a,b,t) = 0$，则称其为紧的；若满足$\mathrm{size}(w,a,b,t) \geq 1$，则称其为大的 。

那么，给定一个定时词 w，我们注意到，关于 w 的监视器获胜延续总是紧的、大的，或者与 w的最后一个时间戳过于接近。

▶ **引理 8.** 设 w 是关于$(\mathrm{T} \times \mathrm{M})$ 的一个获胜定时词，且$(a,b,t) \in \mathrm{winCont}^{\mathrm{M}}_{\mathcal{A}}(w)$。那么，$t_{\sup}(w,a,b,t) - t_n \leq 1$和 $t_{\inf}(w,a,b,t) = t_n$，或者(a,b,t)，是 w的一个紧的或大的获胜延续。

证明。令 t 为一个时间戳，使得 $(a,\ b,\ t)$ 是一个既不紧也不大的获胜延续，即满足 $0 < \mathrm{size}(w,\ a,\ b,\ t) < 1$。我们区分两种情况。

首先，假设该延续接近 t_n，但并非从 t_n 开始，即 $t_{\sup}(w,a,b,t) - t_n \leq 1$ 且 $t_{\inf}(w,a,b,t) > t_n$。由于 (a,b,t_n) 不属于 t 的获胜延续，因此存在某个 $t_n \geq t' < t$，使得 (a,b,t') 不是 w 的获胜延续。因此，当进行对局 (a,b,t) 时，A 中出现的守卫不同于进行对局 (a,b,t_n) 时所使用的守卫。这个新守卫有两种情况：

- 该守卫是一个等式约束，因此 $\mathrm{size}(w,\ a,\ b,\ t) = 0$，由此得到矛盾。否则，该守卫
- 在一段赋值的开区间内成立。因此，要么 $\mathrm{size}(w,\ a,\ b,\ t) \geq 1$，要么 $t_{\inf}(w,\ a,\ b,\ t) = t_n$，两者都会导致矛盾。

其次，假设延续远离 t_n，即 $t_{\sup}(w,\ a,\ b,\ t) - t_n > 1$。由于$t_{\sup}(w,\ a,\ b,\ t) - t_n > 1$，获胜条件 $\mathcal{A}$中的时钟 x 的区域沿着新定时字母的读数发生了变化，我们用 r 表示这个新区域[1]。

- 如果 r是一个开区域（即，对于某个 $n < K$，它是区间 $(n,\ n+1)$，或者是区间 $(K,\ +\infty)$，其中 K是 $\mathcal{A}$ 中最大的常数），并且 t'是一个时间戳，使得 $(a,\ b,\ t')$ 的读数到达 r，那么根据由区域给出的等价关系，我们有 $t_{\inf}(w,\ a,\ b,\ t) \leq t' \leq t_{\sup}(w,\ a,\ b,\ t)$，换言之，$t'$是 t的获胜延续。此外，由于r 的大小至少为一，因此能够到达 r的 t' 集合的大小也至少为一。由此，我们推得 t的获胜延续很大。
- 否则，r 是一个单点集。在这种情况下，要么与 r 相邻的某个开区域包含一个针对 (a,b) 的获胜时间戳，此时我们应用上一条；要么这些区域都不包含获胜时间戳，且获胜延续是紧的。

在这两种情况下，我们都得出矛盾。 ◀

[1] 我们采用 [2], 引入的经典概念：对于非确定性定时自动机，区域 是 $\mathbb{R}^{\mathcal{X}}_{\geq 0}$ 的等价类，它们在可达关系下保持不变，因此在语言接受下也保持不变。注意，在非确定性定时自动机中，$_1$,区域是单位区间 $(n, n+1)$，其中 $n < K$；是满足 $n < K$的单点集 $\{n\}$；或是无限区间 $(K, +\infty)$，其中 K是非确定性定时自动机$_1$中出现的最大常数。

为了完成引理 5 的证明，我们希望说明，只需跟踪有限数量的紧延续，所有其他延续都可以由一个使用 M 中不同字母的较大延续替代。形式上，设 (a, b, t) 是 w 的紧获胜延续；如果不存在 $b' \in \mathrm{M}$，使得 (a, b', t) 是一个较大的获胜延续，则称其为不可替代的 。

▶ **引理 9**。 设 w 是一个在 $\mathrm{T} \times \mathrm{M}$ 上针对 监视器 的获胜定时词，其最后一个时间戳为t_n。那么，至多存在 $|\mathrm{T}| \cdot K$ 个不可替代的紧获胜延续，其中 K 是 A 的守卫中的最大常数。

证明。 直观地说，当所应用迁移上存在等式守卫，或当前赋值位于所应用迁移的守卫边界上时，就会出现紧获胜延续。

现在注意到，在 $t_n + K$ 之后，对于某个字母 $b' \in \mathrm{M}$，整个区间 $(t_n + K, +\infty)$ 上存在一个较大的获胜延续。因此，所有不可替代的紧延续都位于 t_n 和 $t_n + K$ 之间。此外，监视器在此博弈中获胜；因此，对于所有 $(a, t) \in \mathrm{T} \times \mathbb{Q}_{\geq 0}$，都存在某个 $b \in \mathrm{M}$，使得 (a, b, t) 是一个获胜延续。最后，我们证明：如果 (a, b, t) 和 (a, b', t')（相同的 $a \in \mathrm{T}$，不同的 $b, b' \in \mathrm{M}$ 和 $t, t' \in \mathbb{Q}_{\geq 0}$）是两个紧延续，那么 t 和 t' 之间的距离至少为一个时间单位。由此可知，每个字母 $a \in \mathrm{T}$ 至多有 K 个紧延续。

反证，假设存在两个不同的时间戳 t 和 t'，它们之间的距离小于 1，并且 (a, b, t)、(a, b', t') 是不可替代的紧延续。但是，如前所述，对于 t 和 t'之间的每个时间戳 t''，都存在某个 $b'' \in \mathrm{M}$，使得 (a, b'', t'') 是获胜延续。如果这些延续中有一个是大的，那么它必定包含 t 和 t' 中的至少一个。不妨设包含 t，那么要么 $b'' \neq b$，此时 (a, b, t) 不是不可替代的；要么 $b'' = b$，此时 (a, b, t) 不是紧延续，由此得到矛盾。因此，假设对于所有 $t'' \in [t, t']$，形如 (a, b'', t'') 的获胜延续都是紧延续。由于 M 是有限的，存在某个$b \in \mathrm{M}$，使得有无穷多个时间戳 $t_b \in [t, t']$ 满足 (a, b, t_b) 是紧的获胜延续。此外，我们注意到，要将一个时间单位内的两个时间戳分隔到两个不同的区域，某个守卫必须在分别应用每个时间戳后，将 x的赋值分隔开来。因此，由于迁移的数量有限，我们可以推知存在 $t_1 < t_2 \in [t, t']$，使得 (a, b, t_1) 和 (a, b, t_2) 是两个紧的获胜延续，并且由 $w \cdot (a, b, t_1)$ 和$w \cdot (a, b, t_2)$ 到达的区域集合包含相同的开区域。因此，由于所有到达该开区域的时间戳都会使 (a, b) 构成获胜延续，我们可以得出结论：(a, b, t_1) 是一个大的延续。于是，按照前面相同的论证，我们得到矛盾。◀

现在我们可以完成引理 5 的证明。

引理 5 的证明。 根据引理 9，我们知道，在博弈的每一步中，最多存在 $k = 2|\mathrm{T}| \cdot K$ 个重要时间戳，即不可替代的紧延续（其中 监视器 的决策可能针对单个点发生变化），以及大型获胜延续的上界（t_{sup}）和下界（t_{inf}）。特别是，在 计时器的每次对局之后，监视器最多只需保存 k 个新配置即可作出选择，而这至多可以使用 k 个时钟完成。◀

第 2 步：在区域上进行对局。现在起，假设监视器拥有一个至多使用 k 个时钟的获胜策略。根据该策略，我们通过首先限制每个位置的分支度（借助区域等价），然后限制每个分支的大小（借助良拟序），来定义一个获胜控制器。

23:10 单时钟合成问题

首先，我们定义基于区域的策略这一概念。为此，我们根据 s$\mathcal{A}$引入两个定时词之间的一个（等价）关系。我们用$\mathcal{A}(w)$表示$\mathcal{A}$ 在读入 w 时可到达的区域配置（位置与区域组成的对）；由于$\mathcal{A}$ 不是确定性的，该集合不一定是单元素集合。设 w 和 w' 是两个定时词，我们称 w 和 w' 对$\mathcal{A}$等价，记作 $w \sim_{\mathcal{A}} w'$，当$\mathcal{A}(w) = \mathcal{A}(w')$时。

▶**定义 10.** 如果对所有满足 $w \sim_{\mathcal{A}} w'$ 的 w 和 w' 定时词，以及所有 $(a,t) \in \mathrm{T} \times \mathbb{Q}_{\geq 0}$ 个计时器选择，都有 $\sigma(w,(a,t)) = \sigma(w',(a,t))$，则称 k-时钟策略 σ 为基于区域的。

现在，我们证明监视器可以使用这样的策略获胜。

▶**引理 11.** 如果监视器在 G 中拥有一个获胜的 k-时钟策略，那么它就拥有一个获胜的、基于区域的 k-时钟策略。

证明. 令 σ 为一个获胜的 k-时钟策略，且它不是基于区域的。对于每个定时词w，令 $[w]$ 为根据 $\sim_{\mathcal{A}}$定义的 w 等价类的代表元。我们定义σ'使得 $\sigma'(w,(a,t)) = \sigma([w],(a,t))$。根据定义， σ' 是一个基于区域的 k-时钟策略。

最后，我们证明 σ' 是获胜的。反设存在一个获胜定时词 w 以及一个计时器的选择（a, t） $\in \mathrm{T} \times \mathbb{Q}_{\geq 0}$，使得（$a, \sigma'(w,(a,t))$ t）不是获胜延续。然而，由于 σ 是获胜的，我们有（$a, \sigma(w,(a,t))$ t）是一个获胜延续。特别地， $\sigma(w,(a,t))$ 不一定与 $\sigma'(w,(a,t))$ 相同，但$\sigma'($ $w,(a,t))$ $= \sigma([w],(a,t))$。此外， $w \sim_{\mathcal{A}} [w]$ 和区域保留获胜延续。因此（$a, \sigma([w],(a,t))$ t）也必定是获胜的，这就产生了矛盾。◀

现在，我们可以结束引理 4 的证明。

引理 4 的证明。 假设 监视器 在 $\mathcal{G}$ 中有一个获胜策略，并令 K 为 $\mathcal{A}$ 的守卫中的最大常数。特别地，监视器 有一个获胜的 k-时钟、基于区域的获胜策略 σ_{M}，其具有 $k = 2|\mathrm{T}| \cdot K$，由引理 5 和 11 得出。

为完成证明，唯一剩下的步骤是在保持获胜性质的同时剪除 σ_{M} 的无限分支，从而得到一个获胜的 k时钟控制器。为此，我们注意到，一个 k时钟的基于区域的策略在其配置上定义了一个良拟序（我们可以将其限制为 k）。因此，对于每个分支，每当给出一个决策时，我们都会根据该序到达一个更小的配置集合。由于该序不存在无限反链，所有分支最终都会结束，或到达树的另一部分。因此，所得到的策略是一棵有限树，它定义了一个带输出的定时自动机。

最后，我们证明该控制器是获胜的。反设存在一个针对计时器的策略，使得生成的定时词不属于$\mathcal{L}(\mathcal{A})$。特别地，面对该策略，控制器进行的对局不再具有获胜延续。当控制器到达 σ_{M}的树中的一个新部分时，就会发生这一事实。否则，我们将与 σ_{M} 是获胜的这一事实矛盾。但是，这一操作由良拟序给出，其含义是，控制器所到达的配置集合保持了接受条件。因此，我们得出与 σ_{M} 是获胜的这一事实相矛盾。◀

至此，定理 2 的证明结束。 ◀

在所有其余情况下，这两个综合问题并不等价，也就是说，存在获胜策略并不意味着存在控制器：第一，当计时器既是获胜条件的所有者，也是综合问题中的代理，但获胜条件是任意的非确定性定时自动机$_1$ 语言时；第二，当所有者与代理不同时，即使获胜条件是可达性非确定性定时自动机$_1$ 获胜条件：

▶**定理 12**。 以下情况下，这两个综合问题可能并不等价：**1.** 所有者 $\neq$代理 且 $\mathcal{A} \in$ 可达性-非确定性定时自动机$_1$。**2.** 所有者 $=$ 代理 $=$ 计时器 且 $\mathcal{A} \in$ 非确定性定时自动机$_1$。

证明。**1.** 我们先考虑所有者 $=$ 计时器、代理 $=$ 监视器的情形。设计时器的字母表是平凡的（单元素集合），因此下文省略；所以计时器实际上只选择时间戳。设监视器的字母表 $\mathrm{M} = \{a, b\}$，并设计时器的获胜条件要求定时词中包含某个字母（a 或 b），且两者间隔为 1；也就是说，对于某个 $t \in \mathbb{Q}_{\geq 0}$，定时词要么同时包含 (a, t) 和 $(a, t+1)$，要么同时包含 (b, t) 和 $(b, t+1)$。不难看出，该条件可由可达性非确定性定时自动机$_1$识别。监视器在此博弈中有获胜策略：除非时间戳 $t-1$ 先前已在对局中出现，否则它总是选择同一个字母（例如 a）；若该时间戳已经出现，则选择当时所用字母的相反字母。另一方面，监视器没有获胜控制器，因为获胜需要存储无界多个不同的时间戳（以及与之对应使用的字母）。

对于所有者 $=$ 监视器且代理 $=$计时器的情况，我们选择两个字母表都为平凡字母表（因此下文省略）。因此，对局仅为由计时器选择的时间戳单调序列 $t_1 \leq t_2 \leq \cdots$。令监视器的获胜条件由满足以下任一条件的序列组成：某一时刻超过 1（$t_i > 1$ 对于某个 $i \in \mathbb{N}_{>0}$），或重复某个时间戳（$t_i = t_{i+1}$ 对于某个 $i \in \mathbb{N}_{>0}$）。这两个条件及其并集都可由可达性非确定性定时自动机$_1$识别。此外，计时器可以通过生成一个有上界 1 的芝诺词获得获胜策略，但计时器没有获胜控制器。事实上，若要获胜，计时器只能使用严格正延迟；因此，由于每个控制器只使用有限种不同的延迟，最终必然会超过上界 1。

2. 考虑所有者 $=$ 代理 $=$计时器的情况。我们采用与前一种情况相同的思路：计时器只有通过生成芝诺词才能获胜。令计时器的字母表为平凡字母表（因此下文省略）；于是计时器实际上只选择时间戳。令监视器的字母表为 $\mathrm{M} = \{\ ,\ \}$，并令计时器的获胜条件为以下两个集合的并集：时间戳绝不超过 1、且仅标记为 , 的定时词$w = (b_1, t_1)\,(b_2, t_2) \cdots \in (\mathrm{M} \times \mathbb{Q}_{\geq 0})^\omega$，即 $t_i < 1$且对所有 $i \in \mathbb{N}_{>0}$均有 $b_i =$；以及满足以下条件的定时词：对于某个 $i \in \mathbb{N}_{>0}$，有 $t_i \lessdot t_{i+1}$,$b_1 = \cdots = b_i =$且 $b_{i+1} =$。计时器可以通过生成一个有上界 1 的严格单调 芝诺词获得获胜策略。另一方面，违反严格单调性（即出现相等情形 $t_i = t_{i+1}$）会受到监视器出招 $b_{i+1} =$ 的惩罚。因此，和前一种情况一样，计时器没有获胜控制器。◀

结果总结。 作为我们的主要贡献，我们证明了反应式综合问题和 Church 综合问题在所有情况下都是不可判定的，即使在非常受限的情形下也是如此：定时胜利条件由 NTA$_1$ 指定（在其中一个变体中则由 reach-NTA$_1^{\text{res}}$ 指定；总结请见表 2）。无论哪一方是代理，我们针对反应式综合问题的归约都以统一方式进行。因此，表 2 中有六个条目，而不是八个。这六种情况需要四种不同的归约，因此表中的每个条目都标明了我们所归约的问题。当计时器是获胜条件的所有者时，有损计数器机（LCM）中的一些不可判定问题适合用于归约，即重复可达性和有界性 [33]（这些结果构成了技术核心）；而当监视器是所有者时，我们使用 NTA$_1$语言普遍性的两种变体（这些归约相对简单）。

该表规定了我们证明不可判定性的获胜条件类别$\mathcal{A}$，适用于$\mathcal{A} \in$NTA$_1$,，但有一个例外：我们需要*1-重置* 可达性非确定性定时自动机$_1$,，即对可达性非确定性定时自动机$_1$ 进行轻微扩展，使其使用一种受限形式的 ε-迁移，并在时钟达到值 1 时将其重置。我们认为，采用 NTA$_1^{\text{res}}$并不会削弱

	反应式综合	丘奇综合	
	任意代理	代理 =计时器	代理 =监视器
所有者 || 计时器	$\mathcal{A} \in$ 非确定性定时自动机$_1$ 最小公倍数重复可达性。 (定理 16，第 5 节)	$\mathcal{A} \in$ 非确定性定时自动机$_1$ 最小公倍数重复可达性。 （定理 16，第 5 节）	$\mathcal{A} \in$reach-NTA$_1^{\text{res}}$ LCM 有界性 （定理 23，第 6 节）
所有者 || 监视器	$\mathcal{A} \in$ 非确定性定时自动机$_1$ 非确定性定时自动机$_1$ 普遍性 (定理 13，第 3 节)	$\mathcal{A} \in$ 非确定性定时自动机$_1$ 非确定性定时自动机$_1$ 采样普遍性 （定理 15，第 3 节）	$\mathcal{A} \in$ 非确定性定时自动机$_1$ 非确定性定时自动机$_1$ 普遍性 （定理 13，第 3 节）

表 2 我们的不可判定性结果总结。

我们的结果，理由有二：首先，尽管带有 NTA$_1$ ε-迁移的自动机对应于带猜测的 非确定性单寄存器自动机（NRA$_1$）（见 [20]），但 NTA$_1^{\text{res}}$仍对应于 NRA$_1$ 不带猜测；其次，我们所有的不可判定性结果都可以从定时场景转换到数据场景，在数据场景中，获胜集合由不带猜测的 NRA$_1$ 指定。

3 监视器是所有者

作为预备，我们通过从非确定性定时自动机普遍性问题的两个变体归约（在 [25]和 [1] 中证明为不可判定），证明在监视器拥有获胜条件时，这两个综合问题都是不可判定的。

▶ **定理 13.** 当 所有者 = 监视器时，以下问题不可判定：1. 定时反应式综合问题，不论 代理是谁；2. 定时 *Church* 综合问题，当 代理 =监视器时。

证明。我们从非确定性定时自动机普遍性问题归约而来（给定的 A $\in$非确定性定时自动机$_1$ 的语言是否包含所有无限定时词？） [25]。给定一个字母表为 Σ 的非确定性定时自动机$_1$ A，我们构造一个定时博弈，其中监视器的字母表是单元素集合 {□}，计时器的字母表是 Σ，而监视器的获胜条件是 L(A)（忽略字母“□”）。因此，如果计时器生成的定时词不属于 L(A)，计时器就获胜。由此，若 L(A) 不包含所有定时词，计时器就有获胜策略；否则，监视器有获胜控制器（即始终发出“□”的平凡无记忆控制器）。◀

▶ 备注 14。该归约并不保证计时器存在控制器，因为计时器的策略可能需要生成一个无限定时词。然而，在 reach-NTA$_1$ 的情况下，该策略只需生成一个有限前缀，因此无论所有者和代理为何，我们都能得到两个综合问题的 HyperAckermann 困难性。

为了证明代理为 = 计时器时定时 Church 综合问题的不可判定性，我们转向采样 普遍性问题；这是一种变体，采用定时自动机的采样 语义$\mathcal{L}_\delta(_)$。给定 $\delta > 0$，如果一个定时词的所有时间戳都是 δ 的倍数，我们称其具有粒度 δ，并将$\mathcal{L}_\delta(\mathcal{A}) \subseteq \mathcal{L}(\mathcal{A})$定义为$\mathcal{L}(\mathcal{A})$中所有具有粒度 δ 的定时词所构成的语言。给定一个非确定性定时自动机 $\mathcal{A}$，采样 普遍性问题询问：对于所有 $\delta > 0$，语言$\mathcal{L}_\delta(\mathcal{A})$是否包含所有具有粒度 δ 的定时词。该问题（在 [1] 中称为普遍采样普遍性 ）已知对于非确定性定时自动机$_1$ [1,是不可判定的（定理 3]）。

▶ **定理 15。** 当所有者 =监视器与代理 =计时器时，定时 *Church* 综合问题是不可判定的。

证明。 我们采用与定理 13 类似的方法，但将非确定性定时自动机$_1$ 采样普适性问题作为归约源。给定一个非确定性定时自动机$_1$ $\mathcal{A}$ 其字母表为 Σ，我们构造与定理 13 中相同的定时博弈：M $=\{\square\}$，T $=\Sigma$，且监视器的获胜条件为$\mathcal{L}(\mathcal{A})$（忽略“□”）。

如果 Timer 在此博弈中有一个获胜控制器 $\mathcal{C}$，我们就取任意一个有理数 $\delta > 0$，使得$\mathcal{C}$ 所使用的所有正延迟都是 δ 的整数倍，并由此推得$\mathcal{C}$ 输出的任意无限定时词都具有粒度 δ，且不属于$\mathcal{L}_\delta(\mathcal{A})$。

反过来，假设定时词 w 的粒度为 δ，且不被 A 接受。不失一般性地，假设 A 有一条以 w 为标签的运行。我们构造一个词 w'，使其运行只访问有限个不同的配置。令 M 为 A 的守卫中出现的最大常数。如果以 w 为标签的运行中的赋值从未超过 M，也就是说，如果 A 总是在时钟超过 M 时将其重置，那么我们取 $w' = w$ 即可，因为每个赋值始终都是 δ 的整数倍，且在 0 与 M 之间这样的数只有有限个。否则，令 $t_M > M$ 为运行中第一个超过 M 的赋值。我们通过修改 w 得到 w'：每当运行中的赋值大于 M 时，就将其设为恰好 t_M（必要时使用重复的时间戳），而当赋值不超过 M 时，运行保持不变。由于 A 无法区分大于 M 的赋值，w' 不被 A 接受。因此，从某个时刻起，运行中不会再出现接受位置，所以该运行必然形成一个不含接受位置的循环。我们用定时词 w'' 替换 w'，该定时词是通过让 A 无限重复这个循环得到的。和 w' 一样，词 w'' 的粒度为 δ，且不属于 L$_\delta$(A)；此外，当将 w'' 看作字母-延迟对的序列 $w'' = (a_1, \tau_1)\,(a_2, \tau_2)\cdots$ 时，它还是最终周期的。因此，Timer 有一个获胜控制器，可以生成 w''。◀

4 有损计数器机

现在我们介绍后续归约中将使用的模型。有损计数器机（LCM）是一个元组 M $= \mathrm{C}\ S, s_0,$ I，其中 C 是有限个计数器组成的集合，S 是有限个控制位置组成的集合，$s_0 \in S$ 是初始控制位置，I 是有限个指令组成的集合 $I = (s, \mathrm{op}\ s')$，其中 $s, s' \in S$ 且 op $\in \{c, c, c{=}0? \mid c \in \mathrm{C}\}$ 表示对某个计数器 $c \in$ C 执行递增、递减或零测试操作。我们分别用 $\mathrm{I}^c_{\mathrm{inc}}$、$\mathrm{I}^c_{\mathrm{dec}}$、$\mathrm{I}^c_{\mathrm{zt}}$ 表示递增、递减以及对计数器 c进行零测试的指令集合。我们还用 $\mathrm{I}^c = \mathrm{I}^c_{\mathrm{inc}} \cup \mathrm{I}^c_{\mathrm{dec}} \cup \mathrm{I}^c_{\mathrm{zt}}$ 表示所有影响 c的指令集合。

有损计数器机 M 的一个配置是一个对 $(s, \nu) \in S \times \mathbb{N}^{\mathrm{C}}$，其中 s 是一个控制位置，ν 是一个计数器赋值。给定计数器赋值 $\mu, \nu \in \mathbb{N}^{\mathrm{C}}$，当且仅当$\mu$对每个 $c \in$C 都有 $\mu \le \nu\,(c) \le \nu(c)$ 时，我们记作 $\mu \le \nu$。我们定义两种等价的语义：有损语义和自由测试语义（直观来说，后者对应于将损失推迟到零测试时再发生）。在任一种语义下，有损计数器机的一次运行都是从初始配置 (s_0, ν_0) 开始的极大（有限或无限）配置序列，其中 $\nu_0 = 0^{\mathrm{C}}$，且对于任意两个相邻配置 (s, ν) 和 (s', ν')，都存在一条指令 $I = (s, \mathrm{op}\ s') \in$I，使得 $(s, \nu) \xrightarrow{I} (s', \nu')$。在有损语义下，$(s, \nu) \xrightarrow{I} (s', \nu')$ 成立，当且仅当

1. $I \in \mathcal{I}^c_{\mathrm{inc}}$ 和 $\nu' \le \nu[c{+}{+}]$，其中 $\nu[c{+}{+}]$ 与 ν 相同，只是 c 递增 1；或2. $I \in \mathcal{I}^c_{\mathrm{dec}}$ 和 $\nu' \le \nu[c{-}{-}]$，其中 $\nu[c{-}{-}]$ 与 ν 相同，只是 c 递减 1；或3. $I \in \mathcal{I}^c_{\mathrm{zt}}$，$\nu(c) = 0$，且 $\nu' \le \nu$。

在自由测试语义下，$(s, \nu) \xrightarrow{I} (s', \nu')$ 成立，当且仅当

1. $I \in \mathcal{I}^c_{\mathrm{inc}}$ 和$\nu' = \nu[c{+}{+}]$；或2. $I \in \mathcal{I}^c_{\mathrm{dec}}$ 和$\nu' = \nu[c{-}{-}]$；或3. $I \in \mathcal{I}^c_{\mathrm{zt}}$ 和 $\nu' = \nu[c := 0]$ 与.....相同，ν 除了 c 被设为 0。

可以将自由测试语义理解为：损失会延迟到下一次零测试时才发生；届时，计数器会完全失去其值，然后继续进入下一个配置。

不失一般性地，我们假设 $\mathcal{M}$ 是确定性的，即对于任意配置 (s, ν)，至多存在一条指令 I 可将其导向另一个配置。在自由测试语义下，这意味着给定配置至多只有一个后继配置。在有损语义下，可能存在多个后继配置，它们仅在计数器赋值上有所不同（即指令执行后每个计数器损失了多少）。令 $\mathrm{Runs}_{\mathcal{M}}$表示 $\mathcal{M}$的运行。不失一般性地，我们假设这些运行全都是无限的。

有界性问题询问：给定的有损计数器机 $\mathcal{M}$ 的所有运行是否只访问有限多个配置，或者等价地，$\mathcal{M}$ 的所有运行在其赋值上是否有界。重复可达性问题询问：$\mathcal{M}$ 是否存在一条无限运行，无限多次访问给定的控制位置 $s \in S$。已知这两个问题对于 4 个计数器 [28,时就已经是不可判定的 10,12]（优秀的综述见 [33]），并且所选择的语义无关紧要。

5 计时器既是所有者也是代理

本节中，我们证明当计时器拥有获胜条件时，两个综合问题均不可判定；但 Agent = Monitor 的情况除外（将在下一节讨论）。

▶ **定理 16**。 当 所有者 =计时器时，以下问题不可判定：定时反应式综合，无论 代理为何；定时 *Church* 综合，当 代理 = 计时器时。

（注意定理 13 和定理 16 在交换计时器与监视器的角色后具有对称性。）

本节余下部分将通过从有损计数器机的重复可达性问题进行归约来证明定理 16。为此，我们固定一台有损计数器机$\mathcal{M} = \langle \mathcal{C}\ S, s_0, \mathcal{I} \rangle$，它有四个计数器$\mathcal{C} = \{c_1, c_2, c_3, c_4\}$ 以及一个位置 $s \in S$，并构造一个定时博弈，使得当$\mathcal{M}$ 重复到达 s 时，计时器拥有获胜控制器；否则，监视器拥有获胜策略（见引理 22）。证明中，我们假设$\mathcal{M}$采用有损语义。我们的方法受到 [31, 定理 8.4] 证明的启发（但需注意，两者存在显著差异：该文讨论的是定时 Church 综合，其获胜条件由非确定性定时自动机指定，且计时器拥有该获胜条件$_2$ ；而我们只考虑非确定性定时自动机$_1$，。此外，该文寻求的是监视器的控制器，而我们寻求的是计时器的控制器）。

归约的思想。 在博弈过程中，计时器负责生成越来越长的定时词，该定时词应当是$\mathcal{M}$的一次运行的编码。然而，计时器也可能作弊 并生成一个包含错误 的定时词，因此它不是正确的运行编码。我们区分四种错误。为了防止计时器作弊，监视器会验证计时器提出的编码是否正确；如果不正确，监视器必须正确且立即声明所检测到的错误类型，也就是说，必须在错误发生的同一轮中声明。因此，计时器的一种获胜方式是无限多次看到 s，同时没有看到监视器声明任何错误（无论是否作弊）；或者通过作弊误导监视器，使其无法立即声明错误，或声明错误的类型不正确。另一方面，当监视器能够立即声明正确的错误类型时，无论后续延续如何，对局都由监视器获胜。

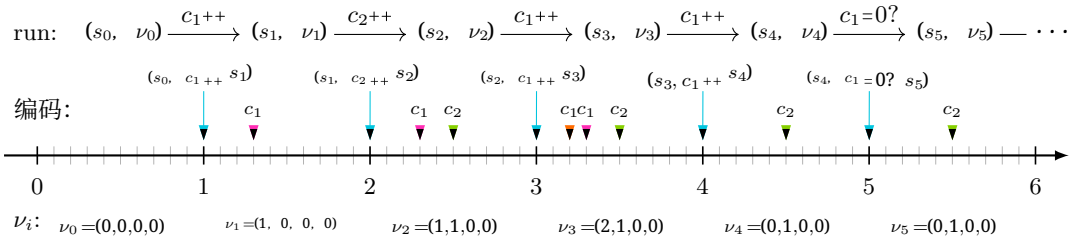


图 2 本节所用运行编码的示意图。

运行编码 $\mathcal{M}$。 我们归约的一个核心要素，是将$\mathcal{M}$的运行编码为定时词。令 $\Sigma_{\mathcal{M}} = \mathcal{C} \cup \mathcal{I}$。我们首先介绍赋值编码。对于一个赋值$\nu = (v_1,\ v_2,\ v_3,\ v_4) \in \mathbb{N}^{\mathcal{C}}$，定义 $\mathrm{enc}(\nu) = c_1^{v_1} c_2^{v_2} c_3^{v_3} c_4^{v_4}$ 为一个由四个片段组成的有限非定时词。所有此类编码的集合为 $\mathrm{ValEnc}_{\mathcal{M}} = c_1^* c_2^* c_3^* c_4^*$。

接下来，我们定义运行编码。函数 $\mathrm{enc}\colon \mathrm{Runs}_{\mathcal{M}} \to \Sigma_{\mathcal{M}}^{\omega}$ 将一个运行 $\rho = (s_0,\ \nu_0) \xrightarrow{I_1} (s_1,\ \nu_1) \xrightarrow{I_2} \cdots$ 映射为无限非定时词 $\mathrm{enc}(\rho) = \mathrm{enc}(\nu_0)\ I_1\ \mathrm{enc}(\nu_1)\ I_2 \cdots$。令 $\mathrm{RunEnc}_{\mathcal{M}} = \{\mathrm{enc}(\rho) \mid \rho \in \mathrm{Runs}_{\mathcal{M}}\}$ 为有效非定时编码的集合。

直观而言，我们利用词的定时结构来保证运行编码的正确性（见引理 17）。为此，我们定义定时语言 $\mathrm{RunEnc}^{\mathbb{T}}_{\mathcal{M}}$。我们使用一个无定时正则语言 $\mathrm{Reg}_{\mathcal{M}}$ 来描述它，该语言规定了$\mathcal{M}$的运行的结构形式；此外，还使用定时语言 $A^{\mathbb{T}}_{\mathcal{M}}$、$B^{\mathbb{T}}_{\mathcal{M}}$ 和 $C^{\mathbb{T}}_{\mathcal{M}}$，它们施加额外的定时性质。令 $\mathrm{Reg}_{\mathcal{M}}$ 由所有满足以下正则条件的（无定时）词 $w \in \Sigma_{\mathcal{M}}^{\omega}$ 组成：

块结构：w 是赋值编码和指令的无限交错，即，w具有以下形式：$w = \mathrm{enc}(\nu_0)\ I_1\ \mathrm{enc}(\nu_1)\ I_2 \cdots \in (\mathrm{ValEnc}_{\mathcal{M}}\, \mathrm{I})^{\omega}$，其中 I_1 从 s_0开始。指令兼容性：每条指令的目标状态都是下一条指令的源状态，即，对于每个子串 $I_i\, \mathrm{C}^*\, I_{i+1}$，存在某个 s，使得 $I_i = (_, _, s)$且 $I_{i+1} = (s, _, _)$。零测试一致性：w 中每个形如 $I_i\ \mathrm{enc}(\nu_i)\ I_{i+1}$ 的子串，其中包含 $I_{i+1} \in \mathrm{I}^c_{\mathrm{zt}}$，都验证 $\mathrm{enc}(\nu_i) \in (\mathrm{C} \setminus \{c\})^*$，即符号 c 不会出现在 $\mathrm{enc}(\nu_i)$ 中。

显然，$\mathrm{Reg}_{\mathcal{M}} \supseteq \mathrm{RunEnc}_{\mathcal{M}}$。令 $\mathrm{Reg}^{\mathbb{T}}_{\mathcal{M}} = \mathrm{untime}^{-1}(\mathrm{Reg}_{\mathcal{M}})$ 包含所有去定时化属于 $\mathrm{Reg}_{\mathcal{M}}$ 的定时词。我们将 $\mathrm{RunEnc}^{\mathbb{T}}_{\mathcal{M}}$ 定义为四种定时语言的交集：

$$\mathrm{RunEnc}^{\mathbb{T}}_{\mathcal{M}} = \mathrm{Reg}^{\mathbb{T}}_{\mathcal{M}} \cap A^{\mathbb{T}}_{\mathcal{M}} \cap B^{\mathbb{T}}_{\mathcal{M}} \cap C^{\mathbb{T}}_{\mathcal{M}} \subseteq (\Sigma_{\mathcal{M}} \times \mathbb{Q}_{\geq 0})^{\omega}$$

其中 $A^{\mathbb{T}}_{\mathcal{M}}$、$B^{\mathbb{T}}_{\mathcal{M}}$ 和 $C^{\mathbb{T}}_{\mathcal{M}}$ 是满足下述条件的无限定时词 w 的语言。为简单起见，我们假设 $w \in L = \mathrm{Reg}^{\mathbb{T}}_{\mathcal{M}} \cap A^{\mathbb{T}}_{\mathcal{M}} \cap B^{\mathbb{T}}_{\mathcal{M}}$，并定义 $C^{\mathbb{T}}_{\mathcal{M}}$（最复杂的条件）的条件，因为无关紧要的是它如何处理 L 之外的词。

Stri时钟单调 icity ($A^{\mathbb{T}}_{\mathcal{M}}$): w i串严格ictl单调 ic (no ti时间戳重复).
B时钟与单位区间对齐（$B^{\mathbb{T}}_{\mathcal{M}}$）： 符号来自 $\mathcal{I}$ 出现在 w 中，时间戳为从 1 开始的连续整数。

赋值编码的良好对齐（$C^{\mathbb{T}}_{\mathcal{M}}$）： 对于 w 中形如 $\mathrm{enc}(\nu_i) I_i \mathrm{enc}(\nu_{i+1})$ 的每个极大连续子串，对于所有计数器 $c \in \mathcal{C}$，均满足以下条件：

1. 如果 $I_i \notin \mathrm{I}^c_{\mathrm{dec}} \cup \mathrm{I}^c_{\mathrm{inc}}$，那么 $\nu_{i+1}(c) \leq \nu_i(c)$，并且只有当 $(c, t-1)$ 出现在 $\mathrm{enc}(\nu_i)$ 中时，定时字母 (c, t) 才会出现在 $\mathrm{enc}(\nu_{i+1})$ 中；2. 如果 $I_i \in \mathrm{I}^c_{\mathrm{inc}}$，那么 $\nu_{i+1}(c) \leq \nu_i(c) + 1$，并且该编码验证情形 1，但允许一个（可能的）额外字母 c 出现在 $\mathrm{enc}(\nu_{i+1})$ 中 c-段的开头，且距离 $\mathrm{enc}(\nu_i)$ 的第一个符号 c 不到一个时间单位。

3. 如果 $I_i \in \mathcal{I}^c_{\mathrm{dec}}$，那么 $\nu_{i+1}(c) \le \nu_i(c) - 1$，并且该编码验证情形 **1**，但第一个字母 c 出现在 编码(ν_i) 它不能出现在 编码(ν_{i+1})。

根据 $C^{\mathbb{T}}_{\mathcal{M}}$ 的定义，计数器 c 的每次递增（分别地，递减）都会在 c-段的开头 添加（分别地，移除）一个字母 c。图 2 给出了一个示例。

► **引理 17。** $\mathsf{RunEnc}_{\mathcal{M}} = \{\mathsf{untime}(w) \mid w \in \mathsf{RunEnc}^{\mathbb{T}}_{\mathcal{M}}\}$。

证明。 我们通过双向包含进行论证，首先证明 $\mathrm{RunEnc}_{\mathcal{M}} \supseteq \{\mathrm{untime}(w) \mid w \in \mathrm{RunEnc}^{\mathbb{T}}_{\mathcal{M}}\}$。设 $w \in \mathrm{RunEnc}^{\mathbb{T}}_{\mathcal{M}} \subseteq \mathrm{Reg}^{\mathbb{T}}_{\mathcal{M}}$（根据 $\mathrm{RunEnc}^{\mathbb{T}}_{\mathcal{M}}$ 的定义）。特别地，我们推得 $\mathrm{untime}(w) \in \mathrm{Reg}_{\mathcal{M}} \subseteq \mathrm{RunEnc}_{\mathcal{M}}$，因此，$\mathrm{Reg}_{\mathcal{M}} = \mathrm{RunEnc}_{\mathcal{M}}$。

反过来，我们证明 $\mathsf{RunEnc}_{\mathcal{M}} \subseteq \{\mathsf{untime}(w) \mid w \in \mathsf{RunEnc}^{\mathbb{T}}_{\mathcal{M}}\}$。设 $w \in \mathsf{RunEnc}_{\mathcal{M}}$，我们希望定义一个定时词 $w^{\mathbb{T}} \in \mathsf{RunEnc}^{\mathbb{T}}_{\mathcal{M}}$，使得 $\mathsf{untime}(w^{\mathbb{T}}) = w$。由于 $w \in \mathsf{RunEnc}_{\mathcal{M}} \subseteq \mathsf{Reg}_{\mathcal{M}}$，根据定义，这样的 $w^{\mathbb{T}}$ 应属于 $\mathsf{Reg}^{\mathbb{T}}_{\mathcal{M}}$。因此，只需验证 $\mathsf{untime}(w^{\mathbb{T}}) = w$ 且 $w^{\mathbb{T}} \in A^{\mathbb{T}}_{\mathcal{M}} \cap B^{\mathbb{T}}_{\mathcal{M}} \cap C^{\mathbb{T}}_{\mathcal{M}}$。构造 $w^{\mathbb{T}}$ 需要定义 w 中各字母的时间戳，更具体地说，需要定义来自$\mathcal{C}$的字母的时间戳（$\mathcal{I}$ 的时间戳由 $B^{\mathbb{T}}_{\mathcal{M}}$ 决定）。

设 $\rho = (s_0, \nu_0) \xrightarrow{I_1} (s_1, \nu_1) \xrightarrow{I_2} \cdots$ 是 $\mathcal{M}$ 的一次运行，且 $w = \mathsf{enc}(\nu_0) I_1 \mathsf{enc}(\nu_1) I_2 \cdots$。我们递归定义一个定时词序列 $\mathsf{enc}^{\mathbb{T}}(\nu_0), \mathsf{enc}^{\mathbb{T}}(\nu_1) \ldots$，使得对所有 $n \ge 0$，均有 $\mathsf{untime}(\mathsf{enc}^{\mathbb{T}}(\nu_n)) = \mathsf{enc}(\nu_n)$。基础情况下，$\mathsf{enc}^{\mathbb{T}}(\nu_0) = \varepsilon$。假设 $\mathsf{enc}^{\mathbb{T}}(\nu_n)$ 已定义，那么 $\mathsf{enc}^{\mathbb{T}}(\nu_{n+1})$ 是满足以下条件的最短词：对所有 $c \in \mathcal{C}$：

- 若 $I_n \notin \mathsf{I}^c_{\mathrm{inc}} \cup \mathsf{I}^c_{\mathrm{dec}}$，则对于 $\mathsf{enc}^{\mathbb{T}}(\nu_n)$ 中最先出现的 $\nu_{n+1}(c)$ 个定时字母 (c, t)，有 $(c, t+1)$ 属于 $\mathsf{enc}^{\mathbb{T}}(\nu_{n+1})$；若 $I_n \in \mathsf{I}^c_{\mathrm{dec}}$，则对于 $\mathsf{enc}^{\mathbb{T}}(\nu_n)$ 中从第二次出现到第 $(\nu_{n+1}(c) + 1)$ 次出现为止的每个定时字母 (c, t)，有 $(c, t+1)$ 属于 $\mathsf{enc}^{\mathbb{T}}(\nu_{n+1})$；若 $I_n \in \mathsf{I}^c_{\mathrm{inc}}$，且 $\nu_{n+1}(c) \le \nu_n(c)$，则应用第一种情况。否则，$\nu_{n+1}(c) = \nu_n(c) + 1$。接着，首先，对于 $\mathsf{enc}^{\mathbb{T}}(\nu_n)$ 中的每个定时字母 (c, t)，有 $(c, t+1)$ 属于 $\mathsf{enc}^{\mathbb{T}}(\nu_{n+1})$。此外，假设 $c = c_i$ 对于 $i \in \{1, 2, 3, 4\}$ 成立。令 $t_{\min} = n + (i-1)/4$，且 t_1 为 $\mathsf{enc}^{\mathbb{T}}(\nu_n)$ 中 c 首次出现的时间戳；若不存在这样的出现，则令其为 $n + i/4$。最后，我们还有 $(c, (t_{\min} + t_1)/2)$ 属于 $\mathsf{enc}^{\mathbb{T}}(\nu_{n+1})$。

容易验证，这些定时词确实满足 $\mathsf{untime}(\mathsf{enc}^{\mathbb{T}}(\nu_n)) = \mathsf{enc}(\nu_n)$，并且 $\mathsf{enc}^{\mathbb{T}}(\nu_n)$ 中的所有时间戳都在区间 $(n, n+1)$ 内，对所有 $n \ge 0$ 均成立。最后，我们令$w^{\mathbb{T}} = \mathsf{enc}^{\mathbb{T}}(\nu_0) \cdot (I_1, 1) \cdot \mathsf{enc}^{\mathbb{T}}(\nu_1) \cdot (I_2, 2) \cdot \ldots$。证明 $w^{\mathbb{T}}$ 属于 $A^{\mathbb{T}}_{\mathcal{M}}$ 且 $B^{\mathbb{T}}_{\mathcal{M}}$ 属于 $w^{\mathbb{T}}$ 显而易见；对指令进行分类讨论可知 $w^{\mathbb{T}}$ 也属于 $C^{\mathbb{T}}_{\mathcal{M}}$，从而完成证明。◄

定时博弈。我们定义定时综合博弈，其中计时器的动作是 $\mathsf{T} = \Sigma_{\mathcal{M}}$，监视器的动作是 $\mathsf{M} = \{\ , \mathrm{R}, \mathrm{A}, \mathrm{B}, \mathrm{C}\}$。定义获胜条件时，关键在于所有语言 $\mathsf{Reg}^{\mathbb{T}}_{\mathcal{M}}, A^{\mathbb{T}}_{\mathcal{M}}, B^{\mathbb{T}}_{\mathcal{M}}$ 和 $C^{\mathbb{T}}_{\mathcal{M}}$ 都可以局部定义：当且仅当无限词 w 的所有（有限）前缀都属于某个定时词局部语言 $\mathsf{Reg}^{\mathbb{T},\ell}_{\mathcal{M}} \subseteq (\Sigma_{\mathcal{M}} \times \mathbb{Q}_{\ge 0})^*$ 时，无限词 w 才属于 $\mathsf{Reg}^{\mathbb{T}}_{\mathcal{M}}$；此外，该语言由一个 $\mathsf{reach\text{-}NTA}_1$ 识别；对于 $A^{\mathbb{T}}_{\mathcal{M}}$、$B^{\mathbb{T}}_{\mathcal{M}}$ 和 $C^{\mathbb{T}}_{\mathcal{M}}$，情况也类似。具体而言：

- $\mathsf{Reg}^{\mathbb{T},\ell}_{\mathcal{M}}$：有限前缀满足 $\mathsf{Reg}_{\mathcal{M}}$ 的定义条件。$A^{\mathbb{T},\ell}_{\mathcal{M}}$：最后两个时间戳不相等（或者词的长度为 1，即边界情况）。$B^{\mathbb{T},\ell}_{\mathcal{M}}$：如果最后一个字母来自 $\mathcal{I}$，那么最后一个开区间内不会出现 $\mathcal{I}$，并且在最后一个时间戳之前恰好一个时间单位处出现 $\mathcal{I}$；如果最后一个字母不来自 $\mathcal{I}$，那么在最后一个时间戳之前不到一个时间单位处会出现某个 $\mathcal{I}$。（我们省略边界情况。）

- $C^{\mathbb{T},\ell}_{\mathcal{M}}$：如果词的最后一个定时字母是 $(c,\ t) \in \mathcal{C} \times \mathbb{Q}_{\geq 0}$，那么它也会在之前一个时间单位处以 $(c,\ t-1)$ 的形式出现；但若它是 c 段中的第一个字母，且最后一条指令是对 c 的递增，则不在此列。此外，当 $(c,\ t)$ 是 c 段中的第一个字母，且最后一条指令是对 c 的递减时，语言 $C^{\mathbb{T},\ell}_{\mathcal{M}}$ 要求前一个 c 段中的第一个字母已被移除，即其时间戳严格小于 $t-1$。和之前一样，我们方便地假设词属于 $L = \mathrm{Reg}^{\mathbb{T},\ell}_{\mathcal{M}} \cap A^{\mathbb{T},\ell}_{\mathcal{M}} \cap B^{\mathbb{T},\ell}_{\mathcal{M}}$，因为 $\widehat{L} = \mathrm{Reg}^{\mathbb{T},\ell}_{\mathcal{M}} \cup \widehat{A^{\mathbb{T},\ell}_{\mathcal{M}}} \cup \widehat{B^{\mathbb{T},\ell}_{\mathcal{M}}}$ 来自属于 $C^{\mathbb{T},\ell}_{\mathcal{M}}$。请注意，$C^{\mathbb{T},\ell}_{\mathcal{M}}$ 是最难的一种，也是唯一一种需要用到非确定性的情况。

获胜条件也主要通过局部方式定义，即结合对计时器或监视器的步施加的限制。为此，我们使用有限词的投影 $\mathrm{proj}_{\mathrm{M}} : (\mathrm{T} \times \mathrm{M} \times \mathbb{Q}_{\geq 0})^* \to \mathrm{M}^*$ 和 $\mathrm{proj}_{\mathrm{T},\mathbb{T}} : (\mathrm{T} \times \mathrm{M} \times \mathbb{Q}_{\geq 0})^* \to (\mathrm{T} \times \mathbb{Q}_{\geq 0})^*$，以及它们的逆投影 $\mathrm{proj}^{-1}_{\mathrm{M}}$ 和 $\mathrm{proj}^{-1}_{\mathrm{T},\mathbb{T}}$。我们将计时器的获胜集合定义为

$$W_{\mathcal{M}} = \mathrm{Reach}(V^{\mathbb{T}}_{\mathcal{M}}) \cup \left(\mathrm{proj}^{-1}_{\mathrm{T},\mathbb{T}}(\mathrm{Inf}(s)) \cap \mathrm{proj}^{-1}_{\mathrm{M}}(\{\nu\}^{\omega})\right)$$

其中，$\mathrm{Inf}(s)$ 是（非定时）正则语言“位置 s 无限次出现”，$\mathrm{Reach}(V^{\mathbb{T}}_{\mathcal{M}})$ 表示那些具有某个前缀属于 $V^{\mathbb{T}}_{\mathcal{M}}$ 的无限定时词所构成的语言，而 $V^{\mathbb{T}}_{\mathcal{M}}$ 本身是以下有限定时词语言的并集：

$$\begin{aligned}
V^{\mathbb{T}}_{\mathcal{M}} = {} & \left(\mathrm{proj}^{-1}_{\mathrm{M}}(\nu^* \chi_{\mathrm{R}}) \cap \mathrm{proj}^{-1}_{\mathrm{T},\mathbb{T}}(\mathrm{Reg}^{\mathbb{T},\ell}_{\mathcal{M}})\right) \cup && \text{(Monitor wrongly claims an Reg error)} \\
& \left(\mathrm{proj}^{-1}_{\mathrm{M}}(\nu^* \chi_{\mathrm{A}}) \cap \mathrm{proj}^{-1}_{\mathrm{T},\mathbb{T}}(A^{\mathbb{T},\ell}_{\mathcal{M}})\right) \cup && \text{(Monitor wrongly claims an } A \text{ error)} \\
& \left(\mathrm{proj}^{-1}_{\mathrm{M}}(\nu^* \chi_{\mathrm{B}}) \cap \mathrm{proj}^{-1}_{\mathrm{T},\mathbb{T}}(B^{\mathbb{T},\ell}_{\mathcal{M}})\right) \cup && \text{(Monitor wrongly claims a } B \text{ error)} \\
& \left(\mathrm{proj}^{-1}_{\mathrm{M}}(\nu^* \chi_{\mathrm{C}}) \cap \mathrm{proj}^{-1}_{\mathrm{T},\mathbb{T}}(C^{\mathbb{T},\ell}_{\mathcal{M}} \cup \widehat{\mathrm{Reg}^{\mathbb{T},\ell}_{\mathcal{M}}} \cup \widehat{A^{\mathbb{T},\ell}_{\mathcal{M}}} \cup \widehat{B^{\mathbb{T},\ell}_{\mathcal{M}}})\right) && \\
& && \text{(监视器错误地声称发生了 } C \text{ 错误)}
\end{aligned}$$

因此，如果 s 无限次出现且监视器没有声明错误（即只进行规定的步），或者对局的某个有限前缀属于 $V^{\mathbb{T}}_{\mathcal{M}}$，则计时器获胜；也就是说，监视器声明了错误类型的错误。具体而言，在相应局部条件成立的回合中，监视器声明 χ_{R}、χ_{A} 或 χ_{B}；而在局部条件 $C^{\mathbb{T},\ell}_{\mathcal{M}}$ 成立，或其他三个局部条件之一不成立的回合中，监视器声明 χ_{C}。直观来说，χ_{R}、χ_{A} 或 χ_{B} 优先于 χ_{C}：若要通过声明错误获胜，监视器必须在相应局部条件不成立时声明 χ_{R}、χ_{A} 或 χ_{B}，而只有在其他情况下才可以声明 χ_{C}。

▶**引理18**。 $W_{\mathcal{M}}$ 可由非确定性定时自动机$_1$识别。

证明。 如前所述，我们需要证明，定义 $V^{\mathbb{T}}_{\mathcal{M}}$ 的并集中的 $\mathrm{proj}^{-1}_{\mathrm{T},\mathbb{T}}$ 内的每个定时语言都可以由非确定性定时自动机$_1$表示。

首先，我们定义局部语言的概念。给定一个无限字定时语言 L，其局部版本记为 L^{ℓ}，是满足以下条件的有限词定时语言：$\widehat{\mathrm{L}} = \widehat{\mathrm{L}^{\ell} \cdot \mathbb{T}(\Sigma^{\omega}_{\mathcal{M}})}; w \in \mathrm{L}$ 当且仅当 w 的所有前缀都属于 L^{ℓ}。直观而言，$\mathrm{Reg}^{\mathbb{T}}_{\mathcal{M}}$、$A^{\mathbb{T}}_{\mathcal{M}}$、$B^{\mathbb{T}}_{\mathcal{M}}$ 和 $C^{\mathbb{T}}_{\mathcal{M}}$ 的局部版本可识别 Reg、A、B 和 C 类型错误的缺失。

现在，我们证明该获胜条件可以用非确定性定时自动机$_1$表示。根据 $\mathrm{Reg}^{\mathbb{T},\ell}_{\mathcal{M}}$ 的定义，该定时语言是在其无定时字上定义的正则语言。因此，它可以用有限自动机表示。现在，我们证明 $A^{\mathbb{T},\ell}_{\mathcal{M}}$ 和 $B^{\mathbb{T},\ell}_{\mathcal{M}}$ 可以通过取补集用确定性定时自动机$_1$ 表示。

23:18 单时钟综合问题

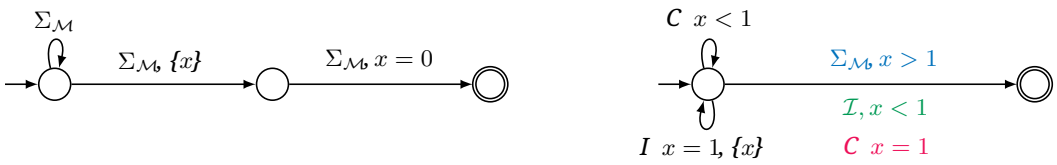


图3 两个识别 $A^{\mathbb{T},\ell}\,\mathcal{M}$ 的确定性定时自动机$_1$（左侧），以及 $B^{\mathbb{T},\ell}\,\mathcal{M}$（右侧）。

▶ **引理19**。 $A^{\mathbb{T},\ell}\,\mathcal{M}$ ∈确定性定时自动机$_1$ 和 $B^{\mathbb{T},\ell}\,\mathcal{M}$ ∈ 确定性定时自动机$_1$。

证明。 我们从 $A^{\mathbb{T},\ell}\,\mathcal{M}$ 开始。根据定义，当 w 中的最后两个连续符号具有相同的时间戳时，$w \in A^{\mathbb{T},\ell}\,\mathcal{M}$。因此，识别 $A^{\mathbb{T},\ell}\,\mathcal{M}$ 的确定性定时自动机$_1$ 如图3左侧所示（标记为 $\Sigma_{\mathcal{M}}$ 的迁移表示对$\Sigma_{\mathcal{M}}$中的每个字母各有一个迁移，且守卫和时钟重置集合相同）。

最后，我们证明 $B^{\mathbb{T},\ell}\,\mathcal{M}$ ∈ 是确定性定时自动机$_1$。根据定义，如果 $w \in B^{\mathbb{T},\ell}\,\mathcal{M}$，则根据 w 的最后一个字母（记作 (a,t)）可能出现几种情况：

- a 是第一条指令，且 $t \neq 1$：第一条指令不是从 1 开始；$a \in \mathcal{I}$，且 $t \notin \mathbb{N}$：某条指令出现在非整数时间戳处（图3中以绿色表示）；$a \in \mathcal{C}$ 且 $n \in \mathbb{N}$：另一个符号出现在整数时间戳处（图3中的橙色边表示）；令 $w = w' \cdot (a',t') \cdot (a,t)$，其中 $t' < n < t$ 对某个 $n \in \mathbb{N}$ 成立：没有符号出现在整数时间戳处（图3中以蓝色边表示）。

因此，识别该语言的确定性定时自动机$_1$ $B^{\mathbb{T},\ell}\,\mathcal{M}$ 如图3右侧所示。 ◀

▶ **推论20**。 $A^{\mathbb{T},\ell}\,\mathcal{M}$ ∈非确定性定时自动机$_1$ 和 $B^{\mathbb{T},\ell}\,\mathcal{M}$ ∈ 非确定性定时自动机$_1$。

为完成证明，我们必须证明：在假设词属于 $\mathsf{Reg}^{\mathbb{T},\ell}\,\mathcal{M} \cap A^{\mathbb{T},\ell}\,\mathcal{M} \cap B^{\mathbb{T},\ell}\,\mathcal{M}$ 的情况下，可以用非确定性定时自动机$_1$ 判断一个（有限[2]）定时词是否满足$C^{\mathbb{T},\ell}\,\mathcal{M}$（否则语言 $C^{\mathbb{T},\ell}\,\mathcal{M}$ 就没有定义）。

▶ **引理21**。 $C^{\mathbb{T},\ell}\,\mathcal{M} \cap \mathsf{Reg}^{\mathbb{T},\ell}\,\mathcal{M} \cap A^{\mathbb{T},\ell}\,\mathcal{M} \cap B^{\mathbb{T},\ell}\,\mathcal{M}$ ∈ 非确定性定时自动机$_1$。

证明。设 w 是一个定时词。我们要用非确定性定时自动机$_1$检验何时 $w \in C^{\mathbb{T},\ell}\,\mathsf{M} \cap \mathsf{Reg}^{\mathbb{T},\ell}\,\mathsf{M} \cap A^{\mathbb{T},\ell}\,\mathsf{M} \cap B^{\mathbb{T},\ell}\,\mathsf{M}$。由于我们可以用确定性定时自动机$_1$, 判断 w 是否属于 $\mathsf{Reg}^{\mathbb{T},\ell}\,\mathsf{M}$、$A^{\mathbb{T},\ell}\,\mathsf{M}$ 和 $B^{\mathbb{T},\ell}\,\mathsf{M}$，因此不失一般性，我们假设 $w \in \mathsf{Reg}^{\mathbb{T},\ell}\,\mathsf{M} \cap A^{\mathbb{T},\ell}\,\mathsf{M} \cap B^{\mathbb{T},\ell}\,\mathsf{M}$，并希望用非确定性定时自动机$_1$检查 $w \in C^{\mathbb{T},\ell}\,\mathsf{M}$ 是否成立。

这里的关键在于，我们只根据 w 的最后一条指令来检验 w 的最后一个字母是否有效。由于我们使用的是有损计数器机，因此无需检验在对局中打出 w 的最后一个字母时是否缺少某个字母。否则，一个时钟就不够用了。

令 $w \in \mathsf{Reg}^{\mathbb{T},\ell}\,\mathcal{M} \cap A^{\mathbb{T},\ell}\,\mathcal{M} \cap B^{\mathbb{T},\ell}\,\mathcal{M}$，我们要检查它是否属于 $C^{\mathbb{T},\ell}\,\mathcal{M}$，且可由非确定性定时自动机$_1$识别。由于我们假设处于 $B^{\mathbb{T},\ell}\,\mathcal{M}$ 中，若 w的最后一个字母属于 $\mathcal{I}$，则 $C^{\mathbb{T},\ell}\,\mathcal{M}$ 显然成立，因此在这种情况下我们直接接受 w。否则，由于特别有 $w \in \mathsf{Reg}^{\mathbb{T},\ell}\,\mathcal{M}$，我们可得

$$\mathrm{untime}(w) = \mathrm{enc}(\nu_0)\, I_1 \cdots \mathrm{enc}(\nu)\, I\, \mathrm{enc}(\nu')$$

[2] 如果发生此类错误，它会出现在计时器打出的某个定时词的前缀上。

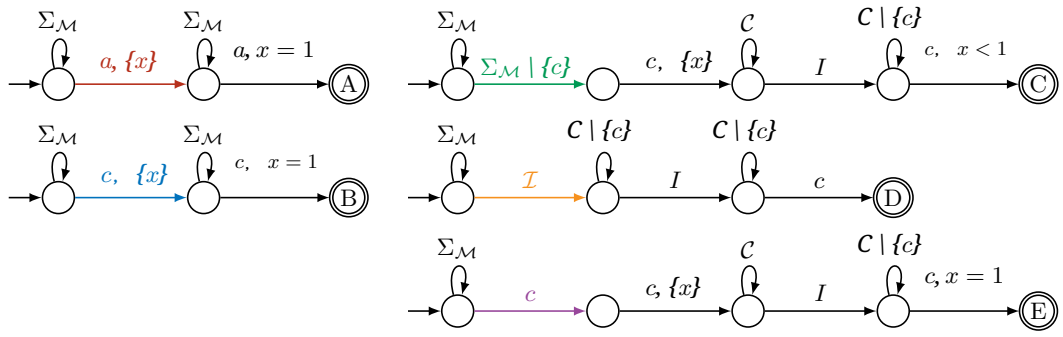


图 4 用于通过公式 (A)、(B)、(C)、(D)、(E) 识别 $C^{\mathbb{T},\ell}_{\mathcal{M}}$ 的装置。

其中 ν 是最后一条指令 I 之前的完整赋值，而 ν' 是执行 I之后可能不完整的赋值。令 $c \in \mathcal{C}$ 为满足 $I \in \mathcal{I}^c$的计数器。根据 I的操作类型，我们区分三种情况，并为每种情况构造一个相应的非确定性定时自动机$_1$ 。

若 $I \in \mathcal{I}^c_{\mathbf{zt}}$:

请注意， I 的操作不会改变赋值，也就是说，由于有损语义，我们只有 $\nu' \geq \nu$ 。因此，要使 w 满足 $C^{\mathbb{T}}_{\mathcal{M}}$ 的局部版本，只需检查 enc(ν) 中是否存在一个与 w 的最后一个字母具有相同小数部分的出现次数。形式化地，w 必须满足以下公式:

$$w = v \cdot (a,t) \ \wedge \ (a,t-1) \in w \tag{A}$$

特别是，可以通过带有图 4 所示红色分支的非确定性定时自动机$_1$ 进行测试。

如果 $I \in \mathcal{I}^c_{\mathbf{inc}}$:

请注意， I 的操作可能会增加计数器 c 的赋值（其他计数器的赋值保持不变或减少）。特别是，当 c首次出现时，它是新出现的，我们需要测试它是否在 enc(ν) 中 c首次出现之后一个时间单位内出现（如果该首次出现存在）。否则，我们需要测试最后一个字母是否在此前一个时间单位处出现（与前一种情况相同）。

设 $(a,\ t)$ 是 w 中最后一个字母。在这种情况下，当满足下文定义的公式 $A \vee B \vee C \vee D$ 时，w 属于 $C^{\mathbb{T},\ell}_{\mathcal{M}}$ 。

情况 A: **a 不是 c 的出现。** 我们假设 $a \neq c$。直观地说，这与零测试的情况相同，因为 I 的操作不会增加计数器 a 的值。因此，我们需要检查 w 是否满足 (A)，这可以通过带有图 4 所示红色分支的非确定性定时自动机$_1$ 进行测试。

情况 B: **a 是 c的重复出现。** 这是 $a = c$ 并非新鲜的情况，即 $(c,\ t-1) \in w$。形式上，w 具有以下形式

$$w = u\,(c,t-1)\,v\,(c,t) \tag{B}$$

可由非确定性定时自动机$_1$ 通过图 4 所示的蓝色分支进行测试。

23:20 单时钟综合问题

情况 C： a 是 c 和 $\nu(c) \geq 1$。 这是指 $a = c$ 在 c 之后首次出现 I的情况，并且递增操作是在赋值为正的计数器上进行的（即， $c \in$enc(ν)）。具体而言， a 是在 enc(ν) 中首次出现后一个时间单位内、 I 之后 c 的首次出现。形式上， w 满足以下公式：

$$w = v_1 \cdot (I, t_I) \cdot v_2 \cdot (c, t) \ \wedge \ c \notin \mathrm{untime}(v_2) \ \wedge \ \exists (\sigma, t_1) \cdot (c, t_2) \subseteq v_1 . (t < t_2 + 1 \wedge \sigma \neq c) \quad \text{(C)}$$

可由带有图 4 所示绿色分支的非确定性定时自动机$_1$ 进行测试。

情况 D： a 是 c 和 $\nu(c) = 0$的一个新出现。 这里的情况是， $a = c$ 在 I之后首次出现 c ，并且 c 首次递增（即，$c \notin$enc(ν)）。此时，我们需要检查： c 不会出现在 enc(ν) 中，也不会在 I 之后出现，直到最后一个字母。此外，由于我们假设 $w \in \mathrm{Reg}^{\mathbb{T},\ell}\mathcal{M}$，我们知道这个 c 将被正确放置（例如， $c = c_0$ 将位于一条指令与 c_1 序列之间）。形式化地，w满足以下公式：

$$w = v_1 \cdot \mathcal{I} \cdot v_2 \cdot (I, t_I) \cdot v_3 \cdot (c, t) \ \wedge \ c \notin \mathrm{untime}(v_2) \ \wedge \ c \notin \mathrm{untime}(v_3) \quad \text{(D)}$$

可由带有图 4 所示橙色分支的非确定性定时自动机$_1$ 进行测试。

如果 $I \in \mathcal{I}^c_{\mathsf{dec}}$：

当 I 的操作使计数器 c递减时，我们需要检查 enc(ν) 中 c 的首次出现是否被“删除”，即在一个时间单位之后不会在 enc(ν') 中再次出现。同样，由于$w \in \mathrm{Reg}^{\mathbb{T},\ell}\mathcal{M} \cap A^{\mathbb{T},\ell}\mathcal{M}$，该出现确实存在且是唯一的。形式化地，令（$a$， t）为 w中的最后一个字母。若 w满足下文定义的公式 $A \vee E$，则 w属于 $C^{\mathbb{T},\ell}\mathcal{M}$。

情况 A： a 不是 c 的出现次数。 如果 $a \neq c$，那么我们再次处于与零测试相同的情况，因为 I 的操作不会增加计数器 a的值。因此，我们需要检查 w 是否满足性质 (A)；可以使用图 4 所示带有红色分支的非确定性定时自动机$_1$ 进行测试。

情况 E： a 是 c 的一次出现。 如果 $a = c$，它就不可能是新出现的（因为不可能递增），即 $(a, t-1) \in w$。此外，$(c, t-1)$ 不能是 c 在 enc(ν) 中的首次出现。形式化地说， w 满足以下公式：

$$w = v \cdot (c, t) \ \wedge \ \exists (c, \tau_1) \colon (c, \tau_1) \cdot (c, \tau - 1) \subseteq w \quad \text{(E)}$$

可以使用图 4 所示带有紫色分支的非确定性定时自动机$_1$ 进行测试。

最后，我们构造用于 $C^{\mathbb{T},\ell}\mathcal{M}$ 的非确定性定时自动机$_1$ ： 首先猜测最后一条指令 I 满足三种情况中的哪一种，然后运行相应非确定性定时自动机$_1$ 的并集，以检查该词是否满足相应的公式。◀

我们已经证明， $A^{\mathbb{T},\ell}\mathcal{M}$、 $B^{\mathbb{T},\ell}\mathcal{M}$ 和 $C^{\mathbb{T},\ell}\mathcal{M}$ 均可由非确定性定时自动机$_1$识别。因此，$W_{\mathcal{M}}$也同样如此。引理18的证明到此结束。◀

归约的正确性。 我们证明以下引理：

▶**引理 22.** 计时器 拥有一个获胜控制器，当且仅当 M 会反复到达 s；否则，监视器 拥有一个获胜策略。

证明。 由于配置上的良拟序以及有损语义，如果 $\mathcal{M}$ 反复到达 s，那么 $\mathcal{M}$ 就有一个 套索运行，能够反复到达 s [33]：该运行从某一点起便呈周期性。因此，该运行受某个 $k \in \mathbb{N}$ 界定，且 计时器 拥有一个获胜控制器，可生成该运行的编码，其粒度至多为 $\delta = 1/4(k+1)$。事实上，采用这一粒度，计时器 可以通过在每个片段的长度为 $1/4$的区间中使用 k 个“槽位”，对所有潜在赋值进行编码。

反之，如果 $\mathcal{M}$ 不会反复到达 s，那么每当计时器产生一次正确的运行时，它必然只会有限次访问 s。于是，监视器可以采用一种获胜策略，记录全部历史，并在计时器没有作弊时继续对局；如果计时器作弊，也能够检测出计时器产生的所有类型的错误。◀

6 计时器是所有者，但不是代理

本节将证明最后一个尚未解决情形下的不可判定性：

▶ **定理 23.** 当所有者 = 计时器且代理 =监视器时，定时 *Church* 综合问题是不可判定的。

（注意定理15和定理23的相似之处：二者交换了计时器和监视器的角色。）

我们在自由测试语义下，从LCM有界性问题归约。固定一个四计数器LCM $\mathcal{M} = \langle \mathcal{C}, S, s_0, \mathcal{I} \rangle$，其中$\mathcal{C} = \{c_1, c_2, c_3, c_4\}$。令 $\mathsf{T} = \mathcal{I} \cup \{\square\}$ 和 $\mathsf{M} = \{\ ,\ \}$。我们定义一个定时博弈，其获胜条件为 $W_{\mathcal{M}}$，使得监视器存在获胜控制器，当且仅当$\mathcal{M}$有界。

1-重置非确定性定时自动机$_1$。 在这种情况下，我们需要对非确定性定时自动机作轻微扩展，以便定义获胜条件和控制器：具有一个时钟以及形式极为受限的 ε-迁移的非确定性定时自动机，这些迁移每当时钟等于1时就会重置时钟。令 $\mathbb{F} = \mathbb{Q}_{\geq 0} \cap [0, 1)$表示小数部分的集合。对于任意时间值 $t \in \mathbb{R}_{\geq 0}$，令 frac($t$)表示其小数部分，即 $\mathbb{F}$中满足以下条件的唯一值：对于某个 $n \in \mathbb{N}$，有 $t = n +$frac(t)。若 ν 是一个赋值，则 frac(ν)表示赋值$\{x \mapsto \text{frac}(\nu(x))\}$。

1-重置非确定性定时自动机$_1$（非确定性定时自动机$_1^{\text{res}}$）是第 2 节中定义的非确定性定时自动机$_1$，仅作如下修改：$(\ell, \nu) \xrightarrow{\delta, \tau} (\ell', \nu')$（带有 $a \in \Sigma$）当且仅当 $\delta = (\ell, a, g, Y, \ell') \in \Delta$ 是 A 的一条迁移，满足 frac($\nu + \tau$) $\models g$，且 $\nu' =$frac($\nu + \tau$)$[Y := 0]$。1-重置非确定性定时自动机$_1$ 可以用带有 ε 转移的非确定性定时自动机$_1$ 进行模拟：在每个位置 $\ell \in L$ 添加一个自环 $(\ell, \varepsilon, x = 1, \{x\}\ \ell)$。

归约的思路。直观地说，计时器负责模拟 $\mathcal{M}$ 的一次运行，而监视器可以在认为计时器在模拟过程中出错时指出错误。具体而言，计时器将执行 $\mathcal{M}$ 的指令，并使用时间值对计数器的赋值进行编码，如下所述（此处使用的编码与上一节中的编码不同）。计时器每走一步后，监视器可以选择表示目前为止模拟正确，或者指出计时器上一步犯了错误。若选择前者，如果计时器确实犯了错误，则计时器获胜（监视器必须指出任何错误）；如果没有错误，则博弈继续（如果博弈就这样永远继续下去，则监视器获胜）。若选择后者，博弈实际上会立即结束：要么监视器判断正确，计时器确实犯了错误，此时监视器获胜；要么监视器判断错误，计时器上一步走得正确，此时计时器获胜。

时间用于编码计数器赋值的方式如下。时间戳的小数部分充当标识符，使递增操作之后能够与相应的递减操作匹配。例如，时间 2.314 发生的递增操作，之后可以与

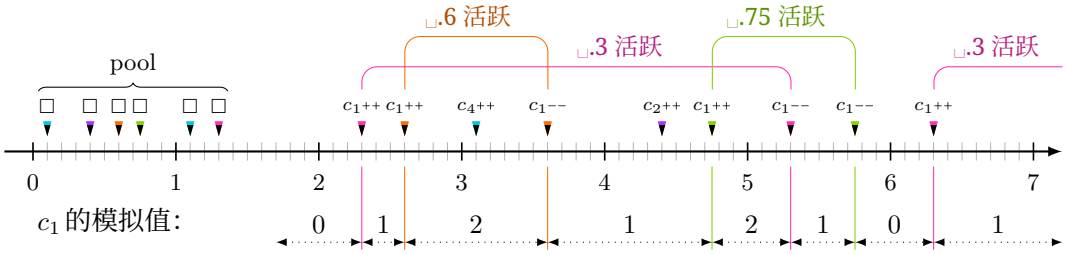


图 5 运行编码示例。对于每条指令（s，op s'），只显示 op。

时间 7.314 发生的递减操作匹配。这样，计数器的赋值就是尚未与小数部分相同的后续递减操作匹配的递增操作／小数部分的数量。示意图见图 5。

显然，要实现这一想法，如果某个小数部分首次用于递增指令，就需要通过不受限制的“猜测” ε-迁移来检查它是否是新的（也就是说，在第一条指令之前猜出这个小数部分，然后检查它在最后一个位置之前从未出现）。为避免这种情况，我们在运行编码的开头引入一个小数部分池，用符号“□”表示。其思路是，计时器必须在开头包含若干个相关的小数部分，这些部分将在之后的运行编码中使用。这样，我们就可以通过非确定性地选择池中的一个位置 来消除时间猜测，在该位置重置时钟，并在之后每当时钟等于 1 时重置它。如果词的最后一个位置是唯一一个时钟恰好为零的位置，那么我们就知道此前从未使用过这个小数部分。

运行编码$\mathcal{M}$。我们将 $\mathcal{M}$ 的运行编码定义为三部分。首先，我们刻画其在 T 上投影的正则性质。令 $\mathrm{proj_T}\colon (\mathrm{T} \times \mathrm{M} \times \mathbb{Q}_{\geq 0})^* \to \mathrm{T}^*$ 为自然投影。定义 $\mathrm{Reg}_{\mathcal{M}} \subseteq (\mathrm{T} \times \mathrm{M} \times \mathbb{Q}_{\geq 0})^*$ 为所有有限定时词 w 的集合，其中 $\mathrm{proj_T}(w) = \square^k I_1 I_2 \cdots I_n$，指令 I_1 从初始位置开始，并且每个 I_i 都与 I_{i+1} 兼容，兼容性的定义见前文。

接下来需要说明，在带有符号 $\mathcal{I}$的位置上，时间戳如何维持 $\mathcal{M}$的赋值。考虑一个有限定时词 $w \in (\mathrm{T} \times \mathrm{M} \times \mathbb{Q}_{\geq 0})^*$。如果最后一个定时字母 (I, t) 满足 $I \in \mathcal{I}^c$且 $\mathrm{frac}(t) = f$ 递增 c $(I \in \mathcal{I}^c_{\mathrm{inc}})$，并且该字母出现在 c最近一次零测试之后，那么我们称小数部分 $f \in \mathbb{F}$ 对于计数器 c 在 w 中是活跃的。反之，如果自 c 最近一次零测试以来，没有出现过包含涉及 c 的指令 $I \in \mathcal{I}^c$ 的 f，或者最后一次此类出现位于 $\mathcal{I}^c_{\mathrm{dec}}$中，那么我们称 f 对于 c 在 w 中是非活跃的。对于运行编码的任意前缀 w，计数器 c的对应赋值等于对其活跃的小数部分进行计数所得的不同小数部分的数量。令 $\mathrm{val}(w)$表示该赋值。

回顾 $\mathcal{M}$ 的运行形式：$\rho = (s_0, \nu_0) \xrightarrow{I_1} (s_1, \nu_1) \xrightarrow{I_2} \cdots$，它是配置与指令交替组成的序列，其中 $\nu_0 = 0^C$。我们归纳定义 $\rho(w)$，这是与任意 $w \in \mathrm{Reg}_{\mathcal{M}}$ 相关联的有限配置与指令交替序列：

$$\rho(w) = \begin{cases} (s_0, \nu_0) & \text{if } \mathrm{proj_T}(w) \in \square^*, \\ \rho(w') \xrightarrow{I} (s, \mathrm{val}(w)) & \text{if } \mathrm{proj_T}(w) = w' \cdot I \text{ and } I = (s', \mathrm{op}, s). \end{cases}$$

注意，$\rho(w)$ 不一定是$\mathcal{M}$ 的运行，原因有二。首先，赋值不一定与指令相关联（例如，一次运行可能包含递增指令，而相应的赋值保持不变或递减）。其次，即使相应计数器的赋值为零，也可能出现递减指令。反之，如果

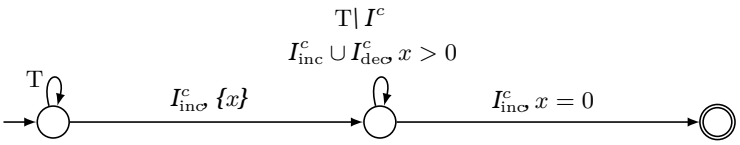


图6 用于 $\text{Err}^{\text{R1}}_{\mathcal{M}}$ 的非确定性定时自动机$_1^{\text{res}}$由上述自动机的四个副本组成，每个副本对应一个 $c \in \mathcal{C}$。

赋值均得到正确更新，且递减指令仅在赋值非零时出现，那么 $\text{Reg}_{\mathcal{M}}$ 可确保 $\rho(w)$ 是$\mathcal{M}$的一次有效运行。为确保有效性，我们引入了以下规则： w中出现的每条指令 I 的时间戳 t 必须满足：

规则1： 如果 $I \in \mathcal{I}^c_{\text{inc}}$，那么 t 的小数部分在 c时必须处于非活跃状态。
规则2： 如果 $I \in \mathcal{I}^c_{\text{dec}}$，那么 t 的小数部分在 c时必须处于活跃状态。

不难看出，同时遵循这两条规则即可确保正确维护计数器赋值。此外，这也能防止在计数器赋值为零时添加递减指令，因为此时没有活跃的小数部分可供选择。请注意，零测试指令不受任何约束，其时间戳无关紧要。

▷ 断言24. 如果 $w \in \text{Reg}_{\mathcal{M}}$ 中的所有指令都满足规则1和规则2，那么 $\rho(w)$ 是 $\mathcal{M}$ 的一次运行。

定时博弈。 对于这两条规则中的每一条，我们现在定义一种有限定时词的语言，要求最后一个字母满足该规则；另定义一种语言，要求最后一个字母违反该规则。令 $\text{Err}^{\text{R1}}_{\mathcal{M}}$ 表示这样一种有限定时词的语言：其中最后一个字母是一条违反规则1的递增指令。此外，令 $\text{Ok}^{\text{R1}}_{\mathcal{M}}$ 表示这样一种词的语言：其中最后一个字母满足规则1，且最后一个小数部分位于池中。请注意，以递减指令或零测试指令结尾的任何定时词都属于 $\text{Ok}^{\text{R1}}_{\mathcal{M}}$，而不属于 $\text{Err}^{\text{R1}}_{\mathcal{M}}$。$\text{Err}^{\text{R1}}_{\mathcal{M}}$ 可由图6所示的 1-重置可达性非确定性定时自动机$_1$ 识别，而 $\text{Ok}^{\text{R1}}_{\mathcal{M}}$ 可由图7所示的 1-重置可达性非确定性定时自动机$_1$ 识别。

很容易看出，语言 $\text{Err}^{\text{R1}}_{\mathcal{M}}$ 和 $\text{Ok}^{\text{R1}}_{\mathcal{M}}$ 不相交。但请注意，这些语言并不是补集。事实上，一个满足规则1、但其最后一个小数部分在开头并未出现在池中的词，不属于这两个语言中的任何一个。

$\text{Err}^{\text{R2}}_{\mathcal{M}}$ 和 $\text{Ok}^{\text{R2}}_{\mathcal{M}}$ 的定义及自动机分别与 $\text{Ok}^{\text{R1}}_{\mathcal{M}}$ 和 $\text{Err}^{\text{R1}}_{\mathcal{M}}$ 的定义及自动机相对应，其中，$\text{Err}^{\text{R2}}_{\mathcal{M}}$ 是违反规则2且 最后一个小数部分在池中 的词所构成的语言，而 $\text{Ok}^{\text{R2}}_{\mathcal{M}}$ 是最后一个字母满足规则2的词所构成的语言。同样，以递减结尾的词自动属于 $\text{Ok}^{\text{R2}}_{\mathcal{M}}$，而不属于 $\text{Err}^{\text{R2}}_{\mathcal{M}}$。

设 $\text{Err}_{\mathcal{M}} = \text{Reg}_{\mathcal{M}} \cap (\text{Err}^{\text{R1}}_{\mathcal{M}} \cup \text{Err}^{\text{R2}}_{\mathcal{M}})$ 和 $\text{Ok}_{\mathcal{M}} = \text{Reg}_{\mathcal{M}} \cap \text{Ok}^{\text{R1}}_{\mathcal{M}} \cap \text{Ok}^{\text{R2}}_{\mathcal{M}}$。我们将计时器的获胜条件定义为以下无限定时词语言：

$$W_{\mathcal{M}} = \text{Reach}\left\{w \mid (\text{proj}_{\text{M}}(w) \in \nu^*\chi \wedge w \in \text{Ok}_{\mathcal{M}}) \vee (\text{proj}_{\text{M}}(w) \in \nu^*\nu \wedge w \in \text{Err}_{\mathcal{M}})\right\}.$$

直观而言，只要在任何时刻监视器的判断出现错误——也就是在计时器给出的序列实际上 χ 正确时出手，或在存在错误时出手——计时器就会获胜。不过，在此之后对局仍会继续，而计时器将获胜。另一方面，如果监视器在对局不属于 $\text{Ok}_{\mathcal{M}}$ 时出手，那么此后的任何延续都将不属于 $W_{\mathcal{M}}$；或者，如果监视器一直出手，而对局始终不属于 $\text{Err}_{\mathcal{M}}$，监视器就会获胜。请注意，计时器在 $\text{Reg}_{\mathcal{M}}$ 之外出手的情况，既属于不在 $\text{Ok}_{\mathcal{M}}$中的情形，因此监视器可以在这种情况下出手并获胜；也属于不在 $\text{Err}_{\mathcal{M}}$中的情形，因此监视器同样可以在这种情况下出手并获胜。

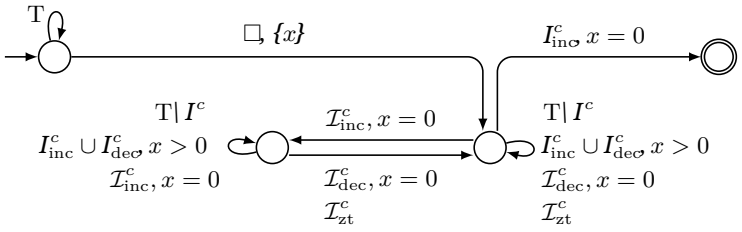


图7 一个能够识别 $\text{Ok}^{\text{R1}}_{\mathcal{M}}$ 的非确定性定时自动机$_1^{\text{res}}$由上述自动机的四个副本组成，每个副本对应一个$\mathcal{C}$，另有一个分支用于检查最后一次迁移是否属于$\mathcal{I} \setminus \mathcal{I}_{\text{inc}}$（省略）。

►**引理25**。 $W_{\mathcal{M}}$ 可由 *reach-NTA*$_1^{res}$识别。

证明。 $\text{Err}_{\mathcal{M}}$ 和$_{\mathcal{M}}$ $\text{Ok}_{\mathcal{M}}$ 都可以由重置一次的可达性非确定性定时自动机$_1$识别。对于$\text{Err}_{\mathcal{M}}$，这是因为可达性非确定性定时自动机$_1^{\text{res}}$在语言并集运算下封闭，并且与$\text{Reg}_{\mathcal{M}}$ 求交不会增加时钟数。至于$\text{Ok}_{\mathcal{M}}$，虽然求交通常会使时钟数相加，但我们可以根据最后一个字母进行分支，模拟$\text{Ok}^{\text{R1}}_{\mathcal{M}}$ 或$\text{Ok}^{\text{R2}}_{\mathcal{M}}$，只需要一个时钟。◄

正确性。 在陈述归约的正确性之前，我们先来直观地说明对局开始时池的作用。计时器最有利的做法，是准确列出之后会用到的尽可能多的小数部分，但数量必须有限。有限这一点显而易见：如果计时器只永远执行 □，那么监视器只需永远执行，便能获胜，因此计时器必须在某个时刻开始真实运行编码。计时器之所以有动力加入许多小数部分，源于它的获胜条件：如果监视器犯错，它就获胜。例如，如果计时器执行一次递增迁移，使用某个非活跃小数部分 f，而监视器以作答，那么当且仅当对局处于 $\text{Ok}^{\text{R1}}_{\mathcal{M}}$时，计时器获胜。但 $\text{Ok}^{\text{R1}}_{\mathcal{M}}$ 只有在最后一个小数部分出现在池中时才接受。因此，如果 f 不在池中，那么该对局不属于 $\text{Ok}_{\mathcal{M}}$ （尽管它是运行的正确编码），监视器通过执行获胜。另一方面，向池中添加更多小数部分并不会带来任何不利，即使其中一些之后永远不会用到；因此，计时器有动力尽可能填满池。

► 引理 *26*。设 $w \in \mathit{Reg}_{\mathcal{M}}$。我们有以下蕴含关系：*1.* $\rho(w) \in$运行$_{\mathcal{M}} \Rightarrow$ 的所有前缀 w' *of* w 都满足 $w' \notin \mathit{Err}_{\mathcal{M}}$。*2.* $\rho(w) \notin$ 运行$_{\mathcal{M}} \Rightarrow$ 的某个前缀 $w' = w'' \cdot (I, m, t)$ *of* w 满足 $\rho(w'') \in$ 运行$_{\mathcal{M}}$, $\rho(w') \notin$运行$_{\mathcal{M}}$，且 $w' \notin \mathit{Ok}_{\mathcal{M}}$。

证明。1. 对 w 的前缀作归纳。基础情况是，对于计时器目前只进行过 □ 个字母的那些前缀，结论显然成立。设 w' 是 w 的一个前缀。由于运行$_{\mathcal{M}}$ 是前缀封闭的，且 $\rho(w')$ 是 $\rho(w)$ 的一个前缀，因此 $\rho(w') \in$ 运行$_{\mathcal{M}}$。假设根据归纳假设， w' 的所有严格前缀都不属于 $\text{Err}_{\mathcal{M}}$。令 $w' = w'' \cdot (I, m, t)$，其中 $I \in \text{I}$、 $m \in \text{M}$，且令$f = \text{frac}(t) \in \mathbb{F}$。我们考察 I 的不同情形，以证明 $w' \notin \text{Err}_{\mathcal{M}}$。

- 假设 $I \in \text{I}^c_{\text{inc}}$。我们有 $\rho(w') \in$运行$_{\mathcal{M}}$、 $\rho(w'') \in$运行$_{\mathcal{M}}$，以及 $\rho(w') = \rho(w'') \xrightarrow{I}(s, \text{val}(w'))$。因此 $\text{val}(w') = \text{val}(w'')[c{+}{+}]$，由此可知 f 对于 w''中的 c是非活跃的。这并不意味着 $w' \in \text{Ok}^{\text{R1}}_{\mathcal{M}}$，因为 f 可能不在池中，但这确实意味着 $w' \notin \text{Err}^{\text{R1}}_{\mathcal{M}}$，因为 $\text{Err}^{\text{R1}}_{\mathcal{M}}$ 只有在最后一个小数部分已经活跃时才能接受。因此 $w' \notin \text{Err}_{\mathcal{M}}$。假设 $I \in \text{I}^c_{\text{dec}}$。由于该运行是正确的，我们有 $\text{val}(w'')(c) > 0$ 且 $\text{val}(w') = \text{val}(w'')[c{-}{-}]$，所以 f 对于 w'' 中的 c 必须是活跃的。于是 $w' \in \text{Ok}^{\text{R2}}_{\mathcal{M}}$，这意味着 $w' \notin \text{Err}_{\mathcal{M}}$。

若 $I \in \mathcal{I}^c_{\mathrm{zt}}$，则 w' 可被 $\mathrm{Ok}^{\mathrm{R1}}_{\mathcal{M}}$ 和 $\mathrm{Ok}^{\mathrm{R2}}_{\mathcal{M}}$ 平凡接受，因此 $w' \notin$ 不属于 $\mathrm{Err}_{\mathcal{M}}$。

⇒ **2.** 如果 $\rho(w) \notin$ 运行$_{\mathcal{M}}$，则存在 w 的某个前缀 $w' = w'' \cdot (I,\ m,\ t)$，使得 $\rho(w') \notin$ 运行$_{\mathcal{M}}$且 $\rho(w'') \in$ 运行$_{\mathcal{M}}$。这是因为至少所有形如 $(\square,\ m,\ t)^*$ 的前缀都编码了空运行，而空运行属于运行$_{\mathcal{M}}$。

我们考察 I 的不同情形，以证明 $w' \notin \mathrm{Ok}_{\mathcal{M}}$。同样，令 $f = \mathrm{frac}(t) \in \mathbb{F}$。假设 $I \in \mathrm{I}^c_{\mathrm{inc}}$。我们有 $\rho(w') \notin$运行$_{\mathcal{M}}$、$\rho(w'') \in$运行$_{\mathcal{M}}$，以及 $\rho(w') = \rho(w'') \xrightarrow{I} (s, \mathrm{val}(w'))$。这意味着 $\mathrm{val}(w')$ 不是正确的赋值，即 $\mathrm{val}(w') \neq \mathrm{val}(w'')[c{+}{+}]$。由此可推知，对于 c，f 在 w'' 中是活跃的，否则 $\mathrm{val}(w')$ 就会是正确的。因此 $w' \in \mathrm{Err}^{\mathrm{R1}}_{\mathcal{M}}$，这意味着 $w' \notin \mathrm{Ok}_{\mathcal{M}}$。假设 $I \in \mathrm{I}^c_{\mathrm{dec}}$。$\rho(w')$ 不属于 $\mathrm{Runs}_{\mathcal{M}}$ 有两种可能的原因：$\mathrm{val}(w')$ 有误，或者 $\mathrm{val}(w'')(c) = 0$。第一种情形下，我们可以像处理递增情形那样推知，对于 c，f 在 w'' 中是非活跃的。第二种情形意味着 c 在 w'' 中没有活跃的小数部分，因此 f 也只能是非活跃的。在这两种情形下，f 是非活跃的并不意味着 $w' \in Err_2$，因为池中可能缺少 f。然而，这仍然意味着 $w' \notin \mathrm{Ok}^{\mathrm{R2}}_{\mathcal{M}}$，因为 $\mathrm{Ok}^{\mathrm{R2}}_{\mathcal{M}}$ 只在最后一个小数部分处于活跃状态时才接受。因此我们有 $w' \notin \mathrm{Ok}_{\mathcal{M}}$。若 $I \in \mathrm{I}^c_{\mathrm{zt}}$，则立即得到矛盾，因为根据定义，$\mathrm{val}(w')$ 将是正确的，所以 $\rho(w')$ 必须属于 $\mathrm{Runs}_{\mathcal{M}}$。◀

请注意，我们无法对第一部分表述为 $w' \in \mathrm{Ok}_{\mathcal{M}}$，对第二部分表述为 $w' \in \mathrm{Err}_{\mathcal{M}}$，原因仍然是池中可能缺少为此所需的关键小数部分。不过，这个引理足以证明该归约的正确性：

▶**定理 27.** *M 有界，当且仅当 监视器 有获胜控制器。*

证明。 ⇒ 假设 $\mathcal{M}$ 的界为 k。我们构造一个控制器$\mathcal{A}$，用于 监视器，它能够正确检测运行编码中首次发生错误的时刻。此控制器假设在此之前，给定计数器的活跃小数部分数量始终不超过 k。这是一个 1-resetting 确定性定时自动机，它使用 $4k$ 个时钟，处理字母表为 T、输出属于M的输入定时词。

令 $\mathcal{X} = \{x^c_i \mid 1 \leq i \leq k,\ c \in \mathcal{C}\}$ 为 $\mathcal{A}$ 的时钟集合。直观来说，时钟 x^c_1，……，x^c_k 被分配给计数器 c，并将“存储”对 c而言处于活跃状态的小数部分。所谓“存储”小数部分 f，是指该时钟恰好在小数部分为 f 的时间戳处取值为 0。$\mathcal{A}$ 的状态空间会跟踪哪些时钟被视为各自计数器的活跃时钟。初始状态为空集。$\mathcal{A}$ 也会忽略初始数据池，并始终在这些数据上输出。

读取 $c \in \mathcal{C}$ 的一次递增时，若状态指示 c 的任一活跃时钟的值恰好为 0，$\mathcal{A}$ 就输出；否则输出 。若它输出，则转到一个始终输出 的汇状态。否则，它将第一个标记为非活跃的时钟 x^c_i 重置，然后将该时钟标记为活跃。若没有这样的空闲时钟，则转到一个输出任意内容的错误状态。

类似地，读取 c 的一次递减且时间为 t时，若没有活跃时钟的值为 0，$\mathcal{A}$ 就输出；否则输出 。根据构造，不会有两个时钟同时取值为 0。若输出为，则转到之前相同的汇状态。否则，它会将该时钟从状态中的活跃时钟集合里移除。

读取计数器 c 的零测试转移时，$\mathcal{A}$ 会输出，并将 c 的活跃时钟集合重置为 $\emptyset$。

首先我们证明，对于由 k 限定的运行，$\mathcal{A}$ 是准确的：

23:26 单时钟综合问题

▶**引理 28.** 对于任意符合 A 的对局 w，若对任何计数器，活跃小数部分的数量始终不超过 k，则 $\rho(w) \in$ 运行$_{\mathcal{M}}$ 当且仅当 $\text{proj}_{\text{M}}(w) \in \checkmark^*$。此外，若$\text{proj}_{\text{M}}(w) \in \checkmark^*$，则某个计数器的赋值恰好等于 A 的状态标记为活跃的时钟数量；这些时钟各自存储一个活跃小数部分，且彼此不同。

证明。 我们通过对 w进行归纳来证明这一点。对于任何形如$(\square、、t)^*$的对局，这同样是显然的。假设上述结论对 w 成立，并且到目前为止$\mathcal{A}$ 始终有输出。现在考虑某个延续 $w' = w\checkmark(I,\ t)$，其中 $f = \text{frac}(t)$。

情况 1a：如果 I 是一个 c 递增指令，并且对于 w中的 c，f 处于非活跃状态，那么 $\text{val}(w') = \text{val}(w)[c{+}{+}]$，因此 $\rho(w') \in$ 运行$_{\mathcal{M}}$。如果 $\text{val}(w')(c) > k$，那么证明就完成了，此后$\mathcal{A}$的行为无关紧要。否则，$\text{val}(w)(c) < k$，根据归纳假设，对于 w 中的 c，恰好有 $\text{val}(w)(c)$ 个不同的小数部分处于活跃状态，并且它们分别位于$\mathcal{A}$ 中恰好同样多的、分配给 c的时钟中。此外，f 不属于其中任何一个。在这种情况下，$\mathcal{A}$ 输出，并将 f 存储在一个分配给 c的空闲时钟中；我们知道至少存在一个这样的时钟。于是，w'满足所有性质。

情况 1b：如果 I 是一条 c 递增指令，且 f 对于 c 在 w 中是活跃的，那么 $\rho(w') \notin$ 运行$_{\mathcal{M}}$。同样根据归纳假设，f 必须位于 $\mathcal{A}$ 分配给 c 且标记为活跃的某个时钟中。因此，$\mathcal{A}$ 输出，我们就此完成。 χ

情况 2a 和 2b 分别对应递减指令且小数部分活跃或非活跃的情形，二者非常相似。

零测试指令的情况 3 也很简单：由于在自由测试语义下，零测试指令始终可用，因此 $w' \in$ 运行$_{\mathcal{M}}$ 是显而易见的，$\mathcal{A}$ 总是输出，且所有时钟都标记为非活跃，这与赋值被设为 0 相一致。◀

回到定理 27 的证明，我们现在证明 $\mathcal{A}$ 是监视器的获胜控制器。设 w 是一局对局。

如果 $\rho(w) \in$ 运行$_{\mathcal{M}}$，那么根据有界性假设，我们知道在 w 的所有前缀中，对于任意计数器，活跃小数部分至多有 k 个，因此根据引理 28，$\text{proj}_{\text{M}}(w) \in \checkmark^*$。此外，根据引理 26，$w$ 的所有前缀均不属于 $\text{Err}_{\mathcal{M}}$。显然，w 不属于 $W_{\mathcal{M}}$。

若 $\rho(w) \notin$运行$_{\mathcal{M}}$，则根据引理 26，存在某个前缀 $w' = w'' \cdot (I,\ m,\ t)$，使得$\rho(w'') \in$运行$_{\mathcal{M}}$，$\rho(w') \notin$运行$_{\mathcal{M}}$，且 $w' \notin \text{Ok}_{\mathcal{M}}$。因此，根据引理 28，$\text{proj}_{\text{M}}(w'') \in \checkmark^*$，并且 $\mathcal{A}$ 在 w'上输出 $m =$ 。由于与之前相同的原因,χ w'' 不属于 $W_{\mathcal{M}}$；而且由于$w' \notin \text{Ok}_{\mathcal{M}}$，$w' \notin W_{\mathcal{M}}$ 也不属于。因此，任何延续（包括 w）的 proj_{M} 都具有如下形式
$\checkmark^* \cdot \chi \not\checkmark$，因此不属于 $W_{\mathcal{M}}$。

$\square \Leftarrow$ 设 ρ 是 $\mathcal{M}$ 的一个无界运行。反设存在某个对监视器而言具有获胜性的控制器 $\mathcal{A}$。令 $G \subset \text{Guard}(\mathcal{X})$ 为 $\mathcal{A}$ 中出现的守卫所构成的有限集合，并令 $k = |G|$。给定某个守卫 $g \in G$ 和一个时钟赋值 ν，令$T(\nu,\ g) = \{t \in \mathbb{R}_{\geq 0} \mid \nu + t \models g\}$。由于 g 是约束的合取，$T(\nu,\ g)$ 要么为空集，要么是单元素集合，要么是一个区间。因此，如果存在两个不同的 t, $t' \in T(\nu,\ g)$，那么任意 $t < t'' < t'$也属于 $T(\nu,\ g)$。

在某个时刻，ρ 从 $k+1$ 变为 $k+2$，这是通过某条递增指令 I对某个计数器 c 的赋值实现的。令 w 为一个定时词，它正确编码了 ρ，直到紧邻 I之前；在 w结束时，小数部分 f_1、.....、f_{k+1}处于活跃状态。为简单起见，我们取$f_1 < \cdots < f_{k+1}$，并假设 w在池中包含这些小数部分。此外，我们还在每一对 f_i 和 f_{i+1}之间放入一个非活跃小数部分。

现在让我们来看看 $\mathcal{A}$ 在这个词上的表现。到目前为止，$\mathcal{A}$ 必定只能有输出，否则计 $\checkmark$ 时器就会获胜。在一个新的动作 (I, t) 上，它必定会输出

χ 如果 frac(t) $\in \{f_1, \ldots, f_{k+1}\}$ 属于活跃小数部分，因此不应将其用于新的递增。但对于任何其他时间戳，如果对应的小数部分在池中，它就必须输出，否则会立即失败。从$\mathcal{A}$的当前状态出发，输出 . 的转移少于 $\aleph_0$ 个。因此，其中至少有一个可以通过至少两个不同的活跃小数部分触发。但正如我们先前所见，这意味着这两个时间戳之间的任何时间戳也能触发该转移。两个活跃小数部分之间的池中至少有一个非活跃小数部分，并且存在一个具有该小数部分的时间戳，可以触发该转移并输出 .。因此，我们得出矛盾。◀

χ

请注意，在这里控制器的存在性与策略的存在性并不等价：监视器始终拥有获胜策略，但仅当$\mathcal{M}$ 有界时才拥有获胜控制器。

7 结论

本文的主要贡献是解决了 [13, 31]中遗留的问题，证明对于已经由NTA_1指定获胜集合的情形，定时综合问题的所有变体都是不可判定的。唯一的例外是第 6 节中的定理 23，在该处我们需要通过一种非常有限形式的 ε-转移来扩展（可达性）NTA_1 。尽管我们相信可以消除这些ε-转移，但我们认为当前结果已经令人满意地解决了这一开放问题：我们的所有不可判定性结果都可从定时场景转化到数据场景，在数据场景中，获胜集合使用非确定性单寄存器自动机（NRA_1）指定，并且既不需要 ε-转移，也不需要猜测。

研究 [13, 31] 中的定时/数据综合，动机之一是它可能有助于解决**确定性可分性** 问题：给定两个定义在有限定时词（分别为数据词）上的非确定性 NTA（分别为 NRA），其语言不相交 L_1，L_2，是否存在一个 DTA（DRA），使其语言将 L_1 与 L_2分开，也就是说，包含其中一个语言且与另一个语言不相交。在 [13, 31] 中，资源受限定时/数据综合的可判定性被用于证明资源受限确定性可分性的可判定性；在后者中，分离自动机中的时钟/寄存器数量预先受到限制。无约束的确定性可分性是否可判定仍未解决。此外，还有一个相关问题——确定性成员资格：给定一个非确定性 NTA，询问其语言是否能被某个确定性 DTA 接受。即使对于 NTA_1 [12]，这个问题也是不可判定的。另一方面，如果预先限制确定性定时自动机中的时钟数量，该问题便是可判定的 [12]。

鉴于 NTA_1 获胜集合会导致综合问题普遍不可判定，一个自然的后续问题是：当我们将范围限制在 NTA_1,的子类（例如，可达性 NTA_1 获胜集合）时，情况是否会有所不同？由于有限词上的确定性可分性可归约为具有可达性 NTA 获胜条件的定时博弈，这或许有助于解决 NTA_1 语言的确定性可分性问题。

最后，我们提醒读者，我们尚不清楚本文研究的博弈是否是确定的，也就是说，是否总有一方玩家拥有获胜策略。

参考文献

1 Parosh Aziz Abdulla、Pavel Krcal 和 Wang Yi。定时自动机的采样普遍性。载于《软件科学与计算结构基础》，第 2–16 页，柏林、海德堡，2007年。Springer Berlin Heidelberg。2 Rajeev Alur 和 David L. Dill。定时自动机理论。《理论计算机科学》，126(2):183–235，1994。URL: https://www.sciencedirect.com/science/article/pii/0304397594900108, doi:10.1016/0304-3975(94)90010-8。3 Eugene Asarin 和 Oded Maler。尽可能快：定时自动机的时间最优控制。载于 HSCC'99 论文集，HSCC '99，第 19–30 页，英国伦敦，1999年。施普林格出版社。URL:http://dl.acm.org/citation.cfm?id=646879.710314。4 Eugene Asarin、Oded Maler、Amir Pnueli 和 Joseph Sifakis。定时自动机的控制器综合。载于 SSSC'98 论文集，第 31 卷《第 5 届 IFAC系统结构与控制会议》，第 447–452 页，1998年。URL: http://www.sciencedirect.com/science/article/pii/S1474667017420325, doi:10.1016/S1474-6670(17)42032-5。5 Gerd Behrmann、Alexandre David、Kim G. Larsen、John Hakansson、Paul Petterson、Wang Yi 和 Martijn Hendriks。Uppaal 4.0。载于第 3 届《国际系统定量评估会议》论文集，QEST '06，第 125–126 页，美国华盛顿特区，2006年。IEEE计算机学会。 doi:10.1109/QEST.2006.59。6 Béatrice Bérard、Antoine Petit、Volker Diekert 和 Paul Gastin。定时自动机中静默迁移表达能力的刻画。《基础信息学》，36(2–3):145–182，1998年11月。7 Sougata Bose、Thomas A. Henzinger、Karoliina Lehtinen、Sven Schewe 和 Patrick Totzke。历史确定性定时自动机。《逻辑方法与计算机科学》，20(4)，2024年。URL:https://doi.org/10.46298/lmcs-20(4:1)2024, doi:10.46298/LMCS-20(4:1)2024。8 Patricia Bouyer、Fabrice Chevalier 和 Deepak D'Souza。使用定时自动机进行故障诊断。载于 FOSSACS'05 论文集，FOSSACS'05，第 219–233 页，柏林、海德堡，2005年。施普林格出版社。 doi:10.1007/978-3-540-31982-5_14。9 Thomas Brihaye、Thomas A. Henzinger、Vinayak S. Prabhu 和 Jean-François Raskin。定时博弈中的最短时间可达性。Lars Arge、Christian Cachin、Tomasz Jurdziński 和 Andrzej Tarlecki 编。载于 ICALP'07 论文集，第 825–837 页，柏林、海德堡，2007年。Springer Berlin Heidelberg。10 J. Richard Büchi 和 Lawrence H. Landweber。通过有限状态策略求解序列条件。《美国数学会汇刊》，138:295–311，1969年。URL:http://www.jstor.org/stable/1994916。11 Franck Cassez、Alexandre David、Emmanuel Fleury、Kim G. Larsen 和 Didier Lime。用于定时博弈分析的高效即时算法。Martín Abadi 和 Luca de Alfaro 编。载于 CONCUR'05 论文集，第 66–80 页，柏林、海德堡，2005年。Springer Berlin Heidelberg。12 Lorenzo Clemente、Sławomir Lasota 和 Radosław Piórkowski。单时钟定时自动机的可确定化性。Igor Konnov 和 Laura Kovács 编。载于第 31 届国际并发理论会议（CONCUR 2020），第 171 卷《莱布尼茨国际信息学论文集》（LIPIcs），第 42:1–42:17 页，德国达格施图尔，2020年。达格施图尔城堡—莱布尼茨信息学中心。URL: https://drops.dagstuhl.de/entities/document/10.4230/LIPIcs.CONCUR.2020.42, doi:10.4230/LIPIcs.CONCUR.2020.42。13 Lorenzo Clemente、Sławomir Lasota 和 Radosław Piórkowski。定时博弈与确定性可分性。Artur Czumaj、Anuj Dawar 和 Emanuela Merelli 编。载于第 47 届国际自动机、语言与编程研讨会（ICALP 2020），2020年7月8日至11日，德国萨尔布吕肯（线上会议），第 168 卷 LIPIcs，第 121:1–121:16 页。达格施图尔城堡—莱布尼茨信息学中心，2020年。URL: https://doi.org/10.4230/LIPIcs.ICALP.2020.121, doi:10.4230/LIPICS.ICALP.2020.121。

14 Luca de Alfaro、Marco Faella、Thomas A. Henzinger、Rupak Majumdar 和 Mariëlle Stoelinga。定时博弈中的惊奇元素。载于 Roberto Amadio 和 Denis Lugiez（编），CONCUR'03 会议论文集，第 144–158 页，柏林、海德堡，2003年。Springer Berlin Heidelberg。15 Stéphane Demri 和 Ranko Lazic。带冻结量词的线性时序逻辑与寄存器自动机。《ACM计算逻辑汇刊》，10(3):16:1–16:30，2009年。 doi:10.1145/1507244.1507246。16 Deepak D'souza 和 P. Madhusudan。面向外部规范的定时控制综合。载于 Helmut Alt 和 Afonso Ferreira（编），STACS'02 会议论文集，第 571–582 页，柏林、海德堡，2002年。Springer Berlin Heidelberg。17 Léo Exibard。《带数据系统的自动综合》（Synthèse Automatique de Systèmes avec Données）。博士论文，艾克斯-马赛大学，法国，2021年。网址： https://tel.archives-ouvertes.fr/tel-03409602。18 Léo Exibard、Emmanuel Filiot 和 Ayrat Khalimov。线性有序数据域上寄存器自动机的丘奇综合。《形式化方法与系统设计》，61(2):290–337，2022年。网址：https://doi.org/10.1007/s10703-023-00435-w，doi:10.1007/S10703-023-00435-W。19 Léo Exibard、Emmanuel Filiot 和 Pierre-Alain Reynier。数据词转换器的综合。载于 Wan J. Fokkink 和 Rob van Glabbeek（编），第 30 届国际并发理论会议 CONCUR 2019，2019年8月27日至30日，阿姆斯特丹，荷兰，第 140 卷 LIPIcs，第 24:1–24:15 页。Schloss Dagstuhl - Leibniz-Zentrum für Informatik，2019年。网址：https://doi.org/10.4230/LIPIcs.CONCUR.2019.24，doi:10.4230/LIPICS.CONCUR.2019.24。20 Diego Figueira、Piotr Hofman 和 Sławomir Lasota。定时自动机与寄存器自动机的关联。《数学结构与计算机科学》，26(6):993–1021，2016年。 doi:10.1017/S0960129514000322。21 Mark Jenkins、Joël Ouaknine、Alexander Rabinovich 和 James Worrell。带度量的丘奇综合问题。载于《计算机科学逻辑》（CSL'11）——EACSL 第 25 届国际研讨会/第 20 届年会（2011），第 307–321 页。Schloss Dagstuhl–Leibniz-Zentrum für Informatik，2011年。22 Marcin Jurdziński 和 Ashutosh Trivedi。定时自动机上的可达时间博弈。载于 ICALP'07 会议论文集，第 838–849 页，柏林、海德堡，2007年。施普林格出版社。网址：http://dl.acm.org/citation.cfm?id=2394539.2394637。23 Michael Kaminski 和 Nissim Francez。有限记忆自动机。《理论计算机科学》，134(2):329–363，1994年。doi:10.1016/0304-3975(94)90242-9。24 M. Kwiatkowska、G. Norman 和 D. Parker。PRISM 4.0：概率实时系统的验证。载于 G. Gopalakrishnan 和 S. Qadeer（编），CAV'11 会议论文集，《计算机科学讲义》第 6806 卷，第 585–591 页。Springer，2011年。25 Sławomir Lasota 和 Igor Walukiewicz。交替定时自动机。《ACM计算逻辑汇刊》，9(2，第 10 篇文章)，2008年。26 Oded Maler 和 Amir Pnueli。论可识别定时语言。载于 Igor Walukiewicz（编），FOSSACS'04 会议论文集，《计算机科学讲义》第 2987 卷，第 348–362 页。Springer Berlin Heidelberg，2004年。网址： http://dx.doi.org/10.1007/978-3-540-24727-2_25，doi:10.1007/978-3-540-24727-2_25。27 Oded Maler、Amir Pnueli 和 Joseph Sifakis。定时系统离散控制器的综合。载于 Ernst W. Mayr 和 Claude Puech（编），STACS'95 会议论文集，第 229–242 页，柏林、海德堡，1995年。Springer Berlin Heidelberg。28 Richard Mayr。不可靠计算中的不可判定问题。《理论计算机科学》，297(1-3):337–354，2003年。doi:10.1016/S0304-3975(02)00646-1。29 Brian Nielsen 和 Arne Skou。基于定时自动机的自动化测试生成。《软件工具与技术转移国际期刊》，5(1):59–77，2003年11月。 doi:10.1007/s10009-002-0094-1。30 Joël Ouaknine 和 James Worrell。定时自动机的语言包含问题：弥合可判定性缺口。载于第 19 届 IEEE计算机科学逻辑研讨会（LICS 2004），2004年7月14日至17日，图尔库，芬兰，论文集，第 54–63 页。IEEE计算机学会，2004年。 doi:10.1109/LICS.2004.1319600。

23:30 单时钟综合问题

31 Radosław Piórkowski。无限状态系统的简化问题。博士论文，华沙大学，2022年。32 Alexander Rabinovich 和 Daniel Fattal。连续时间上的丘奇综合问题。《计算机科学逻辑方法》，21(3)，2025年。网址： https://doi.org/10.46298/lmcs-21(3:8)2025, doi:10.46298/LMCS-21(3:8)2025。33 Philippe Schnoebelen。有损计数器机可判定性速查表。载于 Antonín Kučera 和 Igor Potapov（编），《可达性问题》，第 51–75 页，2010年。 doi:10.1007/978-3-642-15349-5_4。34 Luc Segoufin。无限字母表上的词与树的自动机和逻辑。载于 Zoltán Ésik（编），《计算机科学逻辑》，第 20 届国际研讨会、EACSL 第 15 届年会 CSL 2006，匈牙利塞格德，2006年9月25日至29日，论文集，《计算机科学讲义》第 4207 卷，第 41–57 页。Springer，2006年。 doi:10.1007/11874683_3。35 Martin Tappler、Bernhard K. Aichernig、Kim Guldstrand Larsen 和 Florian Lorber。学习的时机——从测试中学习定时自动机。载于 Étienne André 和 Mariëlle Stoelinga（编），FORMATS'19 会议论文集，第 216–235 页，Cham，2019年。Springer International Publishing。36 Sicco Verwer、Mathijs de Weerdt 和 Cees Witteveen。学习实时自动机的算法。载于比利时-荷兰机器学习会议年会论文集（Benelearn'078），2007年。37 H. Wong-Toi 和 G. Hoffmann。稠密实时离散事件系统的控制。载于 CDC'91 会议论文集，《第 30 届 IEEE 决策与控制会议论文集》第 2 卷，第 1527–1528 页，1991年12月。 doi:10.1109/CDC.1991.261658。