

时间自动机主动学习算法研究进展^①

曹舒, 涂键, 刘芳

(东北石油大学 计算机与信息技术学院, 大庆 163318)

通信作者: 曹舒, E-mail: stucaoshu@163.com



摘要: 时间自动机 (timed automata, TA) 是描述实时系统时间约束行为的重要形式化工具, 广泛应用于嵌入式系统、通信协议等领域. 传统手动构建实时系统模型的方式耗时且易出错, 自动推断模型成为研究热点. 本文聚焦时间自动机主动学习算法, 按照数据存储结构以及等价查询方法进行梳理, 总结了当前时间自动机领域中主动学习算法的最新研究现状, 梳理其核心思想、技术框架, 同时分析当前研究面临的挑战. 通过对比各种方法的优势与局限性, 本文希望为研究者提供一个清晰的参考框架, 并提出未来可能的研究思路, 旨在推动 TA 自动化建模理论与实践发展.

关键词: 形式化方法; 时间自动机; 主动学习; 模型推断

引用格式: 曹舒,涂键,刘芳.时间自动机主动学习算法研究进展.计算机系统应用,2026,35(1):39-51. <http://www.c-s-a.org.cn/1003-3254/10055.html>

Advances in Active Learning Algorithms for Timed Automata

CAO Shu, TU Jian, LIU Fang

(School of Computer and Information Technology, Northeast Petroleum University, Daqing 163318, China)

Abstract: As an important formal tool for describing the time-constrained behavior of real-time systems, timed automata are widely employed in fields such as embedded systems and communication protocols. The traditional way of manually building real-time system models is time-consuming and prone to errors, and automatic inference models have become a research hotspot. This study focuses on the active learning algorithms of time automata, sorts them out according to the data storage structure and equivalent query method, summarizes both the latest research status of active learning algorithms in the current field of time automata, and their core ideas and technical frameworks, with the challenges faced by the current research analyzed at the same time. By comparing the advantages and limitations of various methods, this study hopes to provide researchers with a clear reference framework and propose possible future research ideas, aiming to promote the development of the theory and practice of TA automated modeling.

Key words: formal method; timed automata (TA); active learning; model inference

在安全攸关的领域 (高铁、航空航天、医疗设备), 系统中的任何错误都可能导致灾难性后果^[1]. 近年来, 形式化技术在软件开发中的成功, 展现了其提升复杂软件可靠性的潜力. 各大著名的研究机构都投入了大量人力和物力从事这方面的研究. 时间自动机 (timed automata, TA)^[2,3] 作为一类典型的形式化模型, 它扩展

了自动机理论, 用于实时系统的建模和验证. 由于其强大的形式化建模能力和高表达能力, TA 在理论和实践中都得到了广泛研究, 诸如最优规划、调度或受控合成问题^[4]. 此外, 它们已被证明在实时系统领域之外也具有重要作用, 例如为隐私问题从真实数据生成合成数据^[5], 污水处理厂中的异常检测^[6].

① 收稿时间: 2025-06-23; 修改时间: 2025-07-14, 2025-08-13; 采用时间: 2025-08-19; csa 在线出版时间: 2025-10-29
CNKI 网络首发时间: 2025-10-30

然而,经典方法对于理解复杂的物理或软件系统是通过手动分析,以生成尽可能精确的模型.当系统规模庞大且状态众多时,这一过程对执行的人来说既繁琐又耗时.面对日益复杂的系统行为和海量数据,手动构建模型的局限性愈发明显,例如,需要领域专家知识,耗时长、易引入人为偏差且难以适应动态变化.通过算法自动推断这类模型的方法为这一问题提供了解决方案.从系统行为观察中构建模型可以被视为一个学习问题.对于有限状态反应系统,这可以表述为20世纪70年代的正规语法推理问题,又被称为自动机学习.

自动机学习算法主要分为两种类型:主动学习和被动学习.被动学习通过采样待学模型(system under learning, SUL)的输入输出轨迹,使用状态合并构建与样本一致的自动机模型.然而,被动算法学习到的模型精度完全依赖给定的样本数据,算法所需样本数据量大,且数据常伴有噪音,所以,被动学习难以学到真正的SUL模型.而主动学习则是通过不断与现实系统交互,不断地猜测、查询构建与SUL相同的模型.主动学习的优势在于其高效性和适应性,相比被动学习,主动学习能够动态调整查询策略以适应SUL的行为变化.

目前,关于TA的综述研究已有不少成果^[7-13],但现有研究大多聚焦于TA的语义、分类、相关工具以及在安全领域应用的介绍.本文重点关注TA自动化生成的主动学习算法发展,基于算法演进的主线,系统梳理TA自动生成学习方法,总结不同技术路线的最新研究进展,并深入探讨当前面临的挑战、未来发展趋势及潜在研究方向.

本文第1节介绍时间自动机的基础,第2节详细介绍时间自动机的主动学习算法,第3节探讨未来发展趋势、挑战及优势,第4节总结全文.

1 时间自动机

“Timed automata”这一概念由Alur等人^[3]于1994年提出,TA通过在传统自动机的状态转换中加入时间维度,能够精确地描述和验证具有实时要求的复杂系统行为.相比于传统的离散事件系统模型,TA不仅可以捕获状态间的逻辑关系,还可以对事件发生的时间间隔和时间约束进行建模.为了更好地理解TA的核心思想,可以将其视为一种结合“离散”状态与“连续”过程的混合模型.在TA中,系统的状态由离散状态和连

续变化的时钟变量共同定义,而状态之间的转换则受到时钟变量的约束条件的控制.在本文中,我们将通过一个灯泡控制系统示例,介绍TA的基本概念及其建模方法.

1.1 基础概念介绍

从结构上来看,TA模型是一种带有实值时钟集的有限图.图的顶点称为位置,边是位置之间的转换.每条边都标有时钟守卫、一个动作和一组待重置的时钟.时钟可以在任何转换时重置为0.所有时钟以相同的速率前进,并测量自启动或重置以来经过的时间.

本文使用TA对一个灯的行为建模,如图1所示.一个传感器在开关被按下时发送一个按下事件.灯泡可以处于关闭、开启状态,并且具有低亮度和高亮度.当灯泡关闭时,按下开关一次会打开低亮度,而双击(在2s内)会使灯变亮.如果第1次和第2次按下之间的延迟过长,灯会关闭.

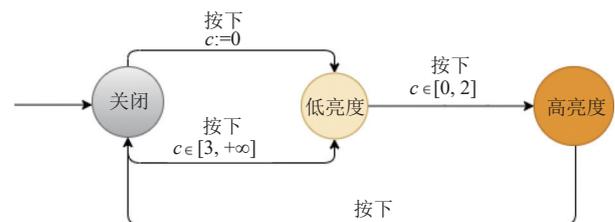


图1 灯的时间自动机

TA通过3个状态(关闭、低亮、高亮)定义了灯的行为,按下事件标记了状态之间的转换.一个时钟 c 测量每次事件之间的时间.在TA中,转换是瞬时的,并允许重置时钟.当按下事件发生时,状态之间的转换被触发(从关闭到开启),并且时钟 c 被重置.如果下一次按下事件在3个时间单位后发生,系统会回到关闭状态.

以下是时间自动机的形式化定义.

一个时间自动机(timed automata, TA)可以表示为六元组 $M = (Q, q_0, \Sigma, F, C, \Delta)$.其中 Q 是有限的位位置集合; $q_0 \in Q$ 是起始位置; Σ 是有限字符集; $F \subseteq Q$ 是接受位置集合; C 是有限时钟集; $\Delta \subseteq Q \times \Sigma \times \Phi(C) \times 2^C \times Q$ 是转移关系的有限集合.

转移关系 $\delta \in \Delta$ 是一个5元组 (q, a, g, γ, q') ,其中 $q, q' \in Q$ 分别是源位置和目标位置, $a \in \Sigma$ 是一个动作, g 是时钟 C 上的时钟约束,被称为转换的守卫,指定当转移条件在源状态中被满足时启用转换,而集合通过

这个转移给出重置时钟. 因此, δ 允许通过执行动作 a 从 q 跳转到 q' , 即 $g(v) = T$.

1.2 时间自动机分类

研究者根据 TA 的结构和行为特点对其进行分类, 以便针对不同应用场景选择合适的模型并优化分析效率. 从时钟个数和同一状态后继个数两个维度对 TA 进行分类是较为常见的分类标准. 前者基于模型中时钟变量的数量, 以适应不同时间约束的复杂性; 后者基于状态转换的确定性, 旨在平衡建模能力与计算复杂性. 其具体分类如图 2 所示.

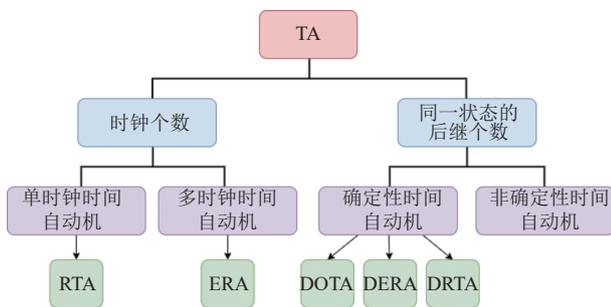


图 2 TA 分类

(1) 按时钟个数分类

1) 单时钟时间自动机: 模型变量组成中仅使用一个时钟变量的自动机. 例如, 实时自动机 (real-time automata, RTA), 仅有一个时钟, 时钟在每次转换时重置.

2) 多时钟时间自动机: 模型变量组成中使用多个时钟变量的自动机. 这类模型适用于对更复杂的时间约束进行建模. 事件记录时间自动机 (event recording timed automata, ERA) 就是多时钟时间自动机的一种. ERA 中的每个动作 (或事件) 都有一个事件记录时钟, 所有时钟都与动作相关联, 且动作数量固定, 事件记录时钟记录关联动作上次发生的时间.

(2) 按同一状态后继个数分类

1) 确定性时间自动机 (deterministic timed automata, DTA): 对于任意给定的状态、时钟估值和输入, 在满足守卫条件时, 只有一个可能的后续状态的自动机. DTA 因其确定性在可达性分析和验证中具有较低的计算复杂度. 本文研究的 TA 均是确定性的, 如确定性单时钟时间自动机 (DOTA)、确定性实时自动机 (DRTA)、确定性事件记录自动机 (DERA). 其中 DERA 又分为时间确定的 ERA (timed deterministic ERA, TDERA) 和事件确定的 ERA (event deterministic ERA,

EDERA). TDERA 可以理解为时钟重叠的情况下, 目标位置必须唯一; 时钟互斥的情况下, 同一事件 (动作) 有不同的后继状态. EDERA 则是指每个状态在每个事件 (动作) 下最多只有一个输出转换 (即移除时钟约束后得到的自动机仍是确定性的).

2) 非确定性时间自动机: 允许同一状态下存在多个可能的后续状态的自动机. 这类自动机分析复杂性较高. 当前研究主要聚焦于 DTA 及其子类, 以规避非确定性带来的困难.

2 时间自动机主动学习算法

TA 主动学习算法通过与系统交互, 自动推断复杂时序模型, 革新了传统建模方式. 本节围绕最小充分教师框架, 系统探讨主动学习方法分类, 剖析其核心技术与效率瓶颈.

2.1 MAT 架构

当前高效的模型主动学习算法多遵循 Angluin^[14] 提出的最小充分教师 (minimally adequate teacher, MAT) 原则. MAT 假设存在一位掌握 SUL 全部信息的教师, 能够回答两类查询: 成员查询 (membership query, MQ) 和等价查询 (equivalence query, EQ). 其基本流程基于“学习-修正”的迭代机制: 算法首先提出一个假设模型 (hypothesis), 通过 MQ 询问 SUL 的特定输入序列是否被接受, 通过 EQ 验证假设模型与 SUL 是否一致. 若发现不一致 (即反例), 算法利用反例信息精炼模型, 直至生成与 SUL 等价的模型. 其框架如图 3 所示.

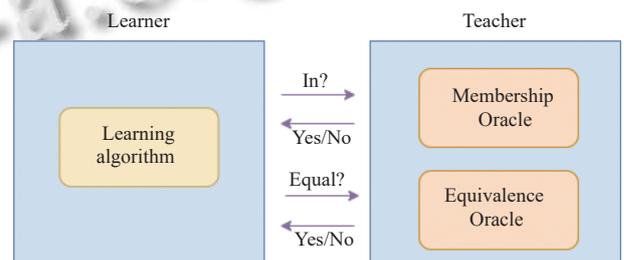


图 3 MAT 框架

2.2 TA 主动学习算法分类

由于 TA 涉及连续时间变量和时钟约束, 学习过程面临状态空间爆炸、时序约束推断和查询效率等挑战. 为此, 研究者提出了多种优化策略, 围绕不同的数据结构和查询方法展开探索.

2.2.1 基于观察表结构

TA 学习旨在推断一个与目标自动机 A 等价的自

动机 H, 即接受相同语言的过程. 其核心方法多源于 Angluin 针对确定性有限状态自动机 (deterministic finite automata, DFA) 提出的 L* 算法. 该算法引入观察表 (observation table, OT) 存储查询结果, 当 OT 满足特

定性时, 可生成与目标系统一致的自动机. 本文按照时间顺序介绍了近 20 年来基于 OT 的 TA 主动学习算法, 如表 1 所示. 对于没有具体名称的改进算法, 使用原型工具名称或作者名代替具体方法名称.

表 1 基于 OT 的主动学习算法

方法	TA类型	时钟个数	数据结构	反例处理	OT查询内容	算法效率
LSDERA	TDERA	多时钟	OT	将反例的所有前缀加入OT	守卫时间序列 $w = (a_1, g_1) \cdots (a_n, g_n)$	$N(M) = O(k^2 n^2 h K)$ $N(E) = O(n)$
TL_{sg}^*	EDERA	多时钟	OT	将反例的所有前缀加入OT	守卫时间序列 $w = (a_1, g_1) \cdots (a_n, g_n)$	$N(M) = O\left(\frac{pn^2 hl}{ \Sigma } \binom{ \Sigma + K}{ \Sigma }\right)$ $N(E) = O(n)$
TL^*	ERA	单时钟	OT	将反例加入OT	守卫时间序列 $w = (a_1, g_1) \cdots (a_n, g_n)$	$N(M) = O(k G_A n^2 + n \log h)$ $N(E) = O(n + k G_A)$
OTALearning	DOTA	单时钟	OT	将反例的所有前缀加入OT	重置本地时间序列 $w = (a_1, \tau_1, b_1) \cdots (a_n, \tau_n, b_n)$	智慧教师: $N(M) = O(kn^5 K^4)$ $N(E) = O(kn^2(2K + 2)^3)$ 普通教师: 最坏情况下需要探索 $2^{(S + R) \times (1 + \sum_{e_i \in E, e_i } (e_i - 1))}$ 个 OT 实例
OneSMT	DOTA&DTMM	单时钟	OT	将反例的所有前缀加入OT	延迟时间序列 $w = (a_1, g_1) \cdots (a_n, g_n) b_w$	查询次数与目标模型大小呈多项式关系, 但 SMT 求解可能是指数级的 (最坏情况)
RTALearning	DRTA	单时钟	OT	将反例的所有前缀加入OT	本地时间序列 $w = (a_1, \tau_1) \cdots (a_n, \tau_n)$	$N(M) = O(kn^2 + kmn^2 + n^3)$ $N(E) = O(kmn)$
LearnTA	DTA	多时钟	OT	将能增加表区分度的反例的前缀和后缀加入OT	初等时间序列 $w = \{t_0 a t_1 t_0 \in (0, 1), t_1 \in (0, 1), t_0 + t_1 \in (0, 1)\}$	智慧教师: $N(M + E) = O(N \text{ 和 } J \text{ 的多项式})$ 普通教师: $N(M + E) = O(2^N)$
DTAL	DTA	多时钟	OT	将反例的所有前缀加入OT	重置时间序列 $w = (a_1, t_1, b_1) \cdots (a_n, t_n, b_n)$ (t_i 表示执行 a_i 之前等待的时间; b_i 表示时钟; t_i 是否重置)	智慧教师: $N(M) = O(kn\Lambda^2)$ $N(R) = O((E - 1)(S + R) E + k S)$ $N(E) = O((n\Lambda = hkn\Lambda^2 + kn\Lambda + (n\Lambda)^2)n\Lambda)$ 普通教师: 最坏情况下需要探索 $2^{(S + R) \times C ^2 \times (\sum_{e_i \in E, e_i } e_i)}$ 个 OT 实例, $N(R)$ 代表重置信息查询的数量

注: $|\Sigma| = k$ 表示字母表的大小; $|Q| = n$ 为状态数; h 为最长反例长度; K 为时钟约束中的最大常数; p 是 $\Sigma \times C$, 表示每个事件与所有时钟守卫的组合; l 是最长守卫时间序列被查询的次数; $|G_A|$ 为两个时钟值确定的时钟区域个数; m 表示由时钟值划分的时钟区域个数; J 为反例分析的最大迭代次数; $|C|$ 是时钟变量数量; N 表示等价类的数量, $N \leq |C|! \times 2^{|C|} \times \prod_{c \in C} (2K + 2) \times n$; $\Lambda = |C|! \times 2^{|C|} \times \prod_{c \in C} (2K + 2)$, 代表时钟区域数量的界限; $|S|$ 是重置时间序列的前缀集合的大小; $|R|$ 是边界集合大小; $|E|$ 是区分 $S \cup R$ 中的不同序列的后缀的大小; g_i 是施加在 a 上的守卫条件; τ_i 表示全局时钟的值; b_i 表示执行完 a_n 后是否重置时钟; t_i 表示执行动作 a_i 的驻留时间; b_w 表示最后一次重置发生的位置

LSDERA 是一种用于学习 TDERA 的算法^[15]. 该方法中每个 DERA 都可以通过区域图构造转化为一个等价的简单 DERA, 且简单 DERA 在等价性上与对应的确定性有限状态自动机等价. 这种符号表示可以复用 L* 算法对正则语言的学习技术, 同时通过一个助手将符号查询转换为定时查询, 从而与时序系统的实际输入兼容. 但是, 它构建了一个区域图, 从而导致了状态和转换数量的激增. 针对 EDERA 则出现了算法 TL_{sg}^* 和 TL_{sng}^* ^[16]. 它们将时间信息嵌入到状态和转换的

定义中, 通过划分时间区域描述状态转移条件, 处理事件之间的时间依赖关系. TL_{sng}^* 在 TL_{sg}^* 的基础上进一步尝试构造状态数更少的自动机. 为了避免每个无时间单词的所有可能时间约束进行主动猜测的过程, 减少所需时间成员查询数量, Lin 等人^[17]提出了一种高效的多项式时间学习算法 TL^* . TL^* 首先学习一个接受无时钟版本时间语言的 DFA, 然后通过添加时间约束被动地细化该自动机.

An 等人^[18]提出了首个针对 DOTA 在连续时间语

义下的主动学习算法. 学习 TA 的核心挑战在于获取时钟重置信息, 为此 An 等人设计了智慧教师 (提供重置信息) 和普通教师 (不提供重置信息) 两种方案. 智慧教师下, 学习器通过 OT 构造 DFA, 转换为 COTA (complete DOTA) 表示的假设 H, 等价查询直至找到与目标自动机等价的 H. 普通教师下, 学习器猜测时间序列的重置信息生成多个候选表, 用广度优先搜索优先探索最少猜测的候选表, 最终构造正确的假设. 对于小规模实时系统, 算法复杂度为多项式的, 具有较高效率. 但是, 算法直接应用分支定界算法, 采用暴力搜索的方式, 从模型跳转到模型, 来处理重置猜测, 这可能会导致较高的计算成本. An 等人还将这种方法拓展到了 DRTA 学习上^[19], 算法成功学习了连续时间语义和离散时间语义的正确模型.

为了应对因猜测重置导致的状态爆炸问题出现了通过约束求解器学习 DOTA 的算法^[20], 该算法在 OT 中记录时间序列间的区别, 不固定重置选择, 用布尔变量表示重置信息和位置分配. 随后, 将闭合性、一致性等条件编码为逻辑公式, 用 SMT 求解器 (如 Z3) 解这些公式, 找到一组可行的重置和位置分配. 若找不到解, 则放松条件或增加位置数重试求解. 算法用单一 OT 和 SMT 求解器避免了指数级重置搜索, 效率得到提高, 适用于较大模型. 作者将这种算法成功扩展到确定性时间 Mealy 自动机 (deterministic timed Mealy machine, DTMM) 上.

上述算法中, 主动 TA 学习算法主要研究的是有限 TA 的子类, 其中时钟变量的数量或在每条边重置的时钟的数量是固定的, 为了学到更一般的 TA, 即不限时钟数量和重置时钟个数, Waga^[21]提出了一种基于 Myhill-Nerode 风格的学习确定性时间自动机 (DTA) 的算法 LearnTA. Myhill-Nerode 定理通过定义语言的等价关系 (Nerode 同余) 来刻画正则语言的可识别性. 这种等价关系下等价类的数量是有限的, 且对应于最小 DFA 的状态数. 这一性质, 可以使得 L* 算法通过学习有限地区分后缀集 S 和前缀集 P, 构造 OT 来推断最小的 DFA. LearnTA 算法引入了“初等语言”“重命名方程”“符号化成员查询”的概念来应对 TA 推断中的时间连续性处理.

为了处理时钟变量在不同时间序列中的重置位置差异, LearnTA 引入了“重命名方程”. 这些方程通过将不同初等语言的时间变量映射到相同的时钟值, 抽象

了时钟重置的语义差异, 这允许算法动态地关联不同初等语言中的时间约束, 而无需预先假设时钟数量或重置位置. 这一举措为多时钟的 TA 学习提供了参考. “符号化成员查询”区别于传统的成员查询 (每次只查询一个序列是否属于目标语言), 符号化成员查询允许一次性查询一个初等语言的所有成员关系, 返回的是时间约束条件, 表示该初等语言中的哪些时间序列属于目标语言. 这种查询方式大大减少了需要查询的时间序列数量, 有效提高了学习速度.

通过将 Nerode 等价关系从离散的字符串扩展到连续的初等语言, Myhill-Nerode 框架为多时钟 DTA 的学习提供了一个通用的理论基础. 这种扩展不仅保留了正则语言学习的结构化方法, 还能处理时间约束的连续性, 从而适用于任意数量的时钟变量. 初等语言、符号化成员查询和重命名方程的引入, 能够处理任意 DTA 的学习, 不限制时钟数量和重置模式, 极大地扩展了适用范围.

尽管 Waga^[21]提出了有效的多时钟主动 TA 学习算法, 但是算法学到的 TA 与原始自动机相比, 最终状态数量大大增加, 缺乏可解释性. 且当目标 DTA 中的时钟守卫具有较大的最大常数 K 时, 该算法在学习期间所需的查询次数和学习到的 DTA 中的位置数量显著增加.

Teng 等人^[22]扩展了 An 等人^[18]的单时钟方法, 提出了具有多个时钟的 DTA 的算法. 算法假定事先知道目标自动机的时钟数目. 给定一个 DTA, 它重置时钟语言存在有限个等价类, 从而将学习问题转化为学习目标自动机的相应重置时钟语言. 由于重置时钟语言包括不可观察的时钟重置信息, 该方法包含两种方案: 智慧教师 (回答重置信息查询) 和普通教师 (不回答重置信息查询). 对于重置信息查询, 学习器沿着一个有效的时间序列的运行过程询问重置信息. 在普通教师场景下, 由于需要猜测重置信息, 算法在目标自动机的大小上仍然具有指数级复杂性.

综上, 基于 OT 的 TA 主动学习算法通过 L* 框架进行扩展, 有效推断 TA 子类的模型. LSDERA、 TL_{sg}^* 、 TL_{sng}^* 等利用区域图或时间嵌入处理时序约束, TL^* 通过分阶段学习降低查询需求, An 等人^[18]和 Teng 等人^[22]的算法灵活处理重置信息, OneSMT 方法和 LearnTA 进一步提升效率和通用性. 这些算法在小规模系统或智慧教师场景下表现出多项式复杂度, 但在区域图构造、

重置猜测或大常数 K 时面临状态爆炸和查询次数激增的挑战。此外, OT 在冗余信息存储问题上的局限, 限制了其在复杂系统中的应用。而树结构可以通过层次化存储、动态精化和反例复用, 克服这些不足, 以更小的空间和更高的查询效率优化学习过程, 故引入树结构存储查询信息成为必然。

2.2.2 基于观察树结构

OT 中存在大量冗余的成员查询信息。为缓解状态爆炸问题, 衍生出了以树结构替代表结构存储信息的策略。有学者进一步将该结构扩展至 TA 领域, 优化了

查询与假设构建过程。表 2 展示了以树结构存储查询信息的 TA 主动学习算法。

为了避免构造区域图, 有效减少状态爆炸, 于是出现了使用时间决策树和观察结构作为中间结构来存储查询结果的方法^[23]。该方法首先通过成员查询收集目标系统的行为信息, 构建时间决策树。同一节点同时包含接受和拒绝的时钟序列时, 算法识别一对“临近”时钟序列, 推断区分它们的守卫条件, 并分裂节点。决策树满足特定条件后, 根据未来行为的等价性将其折叠为假设的 TDERA 模型 H 。

表 2 基于树结构的 OT 主动学习算法

方法	TA类型	时钟个数	数据结构	反例处理	特点	算法效率
文献[23]	TDERA	多时钟	时间决策树	将反例加入观察结构中, 通过二分搜索细化守卫	高复杂度	$N(M) = O\left(\left(\frac{(k+2K+h)e}{\Sigma}\right)^{k(h+1)}\right)$ $N(E) = O\left(\left(\frac{(k+2K+h)e}{\Sigma}\right)^{k(h+1)}\right)$
文献[24]	ERA	多时钟	时间观察图	将反例加入时间观察图中	能处理多时钟系统不固定重置的限制	—
文献[25]	RTA	单时钟	实时分类树	定位错误点, 守卫错误则修正时间守卫, 节点缺失则更新树结构	成员查询数少	—
文献[26]	DOTA	单时钟	逻辑时间分类树	反例前缀分析发现错误, 则增加叶节点迁移; 反例后缀分析发现错误则分裂新节点或细化时间约束	内存空间占用少	$N(M) = O(kn^3K^3h)$ $N(E) = O(kn^2K)$

针对黑盒系统中时间行为的推断, 衍生出了一种针对不可观测重置的定时自动机 (RERA) 的主动学习方法^[24], 该方法通过时间决策图和观察图动态检测和修剪错误重置假设。在线检测不一致性和无效性, 避免存储所有可能假设, 有效减少查询次数。

存储同样的信息, 树结构所需的空间更小, 且反例处理时, 不必再保存反例的所有前缀, 相比基于 OT 的方法, 这种处理方式可以减少大量冗余信息的存储, 故出现了使用实时分类树结构代替 OT 来存放成员查询信息的方法^[25], 该方法根据分类树构建设假设 H 进行等价查询, 直至找到与目标自动机 S 一致的假设。针对 DOTA, 米钧日等人^[26]又在分类树中扩展了时钟信息, 提出了逻辑时间分类树的概念。其提出的算法在反例处理过程中, 对反例中出现的第 1 个分歧下标进行处理。然而, 当反例中存在多个错误下标, 则会丢失一些关键信息, 从而导致等价查询次数的增加。因此, 作者提出了“反例复用”的策略, 对反例进行处理并构建出一个新的假设 DOTA 之后, 再次判断这个反例是否能作为新构建的假设 DOTA 的反例, 以此来充分利用反例中的有效信息。如此一来, 算法避免了对相同序列的

重复成员查询, 显著降低了查询开销。实验数据表明米钧日等人提出的改进算法减少了 60% 左右的成员查询和 5% 左右的等价查询。整个反例的复用使得算法能够更快地构建和优化时间分类树, 从而加速假设自动机的构造过程, 极大地提高了学习的速度。实验中, 改进算法的学习速度最高可提高 45 倍以上。

同样是树结构, Grinchtein 等人^[23]在反例处理时, 并不会将先前的反例再与假设自动机进行比较。这跟具体的反例处理策略有关, 当然也跟两种算法要处理的目标不同有关。Grinchtein 等人提出的算法面向的对象是具有多时钟的 ERA, 这意味他们面临着多时钟的区域数量随事件和时钟数量呈指数增长问题, 导致反例处理需要区分复杂的时序约束。反例处理时, 他们先将反例存放在观察结构中, 通过二分搜索寻找临界对, 推导分离守卫, 以分裂不一致节点, 随后重构子树以更新决策树。而米钧日等人^[26]的方法专注于只有一个时钟的 DOTA, 时钟约束可以通过节点的分裂更简单的被引入, 学习效率更高。

综上, 基于观察树结构存储主动查询信息, 相比传统 OT 显著减少了冗余信息和存储空间, 有效缓解状

态爆炸问题. 这些方法通过动态分裂、折叠、修剪或反例复用优化查询效率, 尤其适合黑盒系统和复杂时序行为的建模. 时间决策树和图通过避免区域图降低复杂度, 实时分类树和逻辑时间分类树则通过反例分析和时钟信息扩展提高学习速度. 然而, 树结构的构建和维护需平衡计算开销, 且对复杂系统的适应性仍需进一步优化. 未来可结合启发式或机器学习方法, 进一步提升算法在工业级场景中的适用性.

2.2.3 非精确等价查询

实现等价 Oracle 被认为是“自动机学习中的真正瓶颈”^[27]. 等价查询在现实中的实现较为困难, 因为没有知道黑盒系统所有信息的强大教师. 用大量成员查

询替代等价查询的一致性测试^[28-30]是等价查询的主流研究方法之一. 这一方法同样可以应用于 TA 的学习, 表 3 展示了两种非精确等价查询的 TA 主动学习算法.

基于 Shen 等人^[31]的工作引入 PAC 学习理论, 通过测试样本集近似回答等价查询, 放宽精确查询需求, 其目标是学习在给定误差和置信度下与目标系统一致的 DOTA 模型. 该算法放弃均匀分布, 采用基于假设结构的随机采样, 并引入“比较器”, 通过成员查询找到区分当前假设 H 与前一假设 H' 的最短时间序列, 使更新后的假设更接近目标. 有效放宽了对精确等价查询的要求, 算法更适用于现实黑盒系统学习场景.

表 3 非精确等价查询算法

方法	TA类型	时钟个数	数据结构	反例处理	OT查询内容
learning OTA by testing	DOTA	单时钟	OT	将最小化反例加入OT	重置延迟(本地) 时间序列 $w = (a_1, \mu_1, b_1) \cdots (a_n, \mu_n, b_n)$
mut-learn-DOTAs	DOTA	单时钟	OT	将反例加入OT	重置延迟(本地) 时间序列 $w = (a_1, \mu_1, b_1) \cdots (a_n, \mu_n, b_n)$

Tang 等人^[32]则提出了一种结合随机测试和突变测试的一致性测试方法来替代精确等价查询, 专门用于特定类别——DOTA 的主动学习. 该方法首先通过启发式方法生成大量候选测试用例; 对当前假设 H 应用基于时间和分裂位置的两种变异操作符, 生成一系列变异模型; 根据变异覆盖率和评分机制, 从候选测试用例中筛选出能够高效检测差异的测试子集; 执行筛选后的测试子集, 若发现假设 H 与目标系统行为不一致, 则返回反例; 否则确认 H 正确. 当前参数(如切片步长、评分权重)依赖经验设置.

综上, 非精确等价查询方法通过近似或启发式方式有效降低了计算复杂度和对系统访问的需求, 适合黑盒实时系统的建模. PAC 学习策略通过随机采样和比较器机制实现了高效的近似等价查询, 适合资源受限的黑盒系统; Tang 等人^[32]的方法通过随机与突变测试结合, 强调测试覆盖和差异检测, 适用于复杂 DOTA 模型的学习. 现有方法仍存在局限: PAC 学习依赖随机采样的质量, 可能遗漏关键行为; Tang 等人^[32]的方法中参数依赖经验调整, 缺乏自适应性. 一个可行的未来研究方向是, 融合 PAC 学习的理论保证与突变测试的启发式优势, 设计多阶段测试策略, 比如, 在早期利用随机采样快速逼近目标模型, 后期通过变异操作精炼高覆盖率测试用例. 针对大规模黑盒系统, 探索增量式测试方法大有可为, 通过动态更新测试子集, 以减少每次等价查询的计算开销.

2.2.4 其他方法

除了上述分类外, 研究者还探索了描述实时系统的另一种状态机——带有时钟的 Mealy 机, 其与 TA 具有一定的相似性. 它通过显式定义输入动作到输出动作的映射, 结合时钟变量描述时间约束行为. 相较于 TA, 它的输出直接依赖于当前状态和输入, 而 TA 更强调状态转移的时钟守卫和不变量约束. 尽管两者在形式化表达上有差异, 但实践中, 研究人员将带有时钟的 Mealy 机视为 TA 的一种变体, 可以转换成 TA.

Caldwell 等人^[33]提出了一种算法, 用于从可编程逻辑控制器(PLC)中学习带有计时器的 Mealy 机, 称之为 TDMM. 算法使用 LearnLib 库的直接假设构建算法, 结合自定义的 PLCSUL 适配器, 主动学习 TDMM, 从工业示例-废水处理厂的阀门功能块中学习模型并分析其执行效率. 实验结果表明, 随着软件规模的增加, 学习所需的时间迅速增加, 这是因为阀门功能模块中自动/手动模式切换、限位开关反馈等功能增加了行为复杂性, 使得学习算法需要更多时间来构建准确的模型. 此外, TDMM 模型需要精确捕获时间延迟, 而 PLC 软件中的时间延迟可能因输入条件或操作模式而异. 对于较大的功能模块, 涉及更复杂的时序逻辑, 修剪不确定延迟守卫的过程会变得更加耗时, 因为需要多次与 SUL 交互以验证行为一致性.

现有主动学习工具(如 LearnLib)主要支持 DFA 或 Mealy 机, 难以捕获时间行为. 已有针对 TA 的学习算

法需要推断时钟守卫和重置而面临组合爆炸问题,效率低下。鉴于此,有学者提出单定时器 Mealy 机 (MMIT) 以及相应的推断算法 $L_{\#}$ ^[34], 利用随着时间递减的定时器表示时间变化, MMIT 没有守卫条件和不变量, 这一举措简化了学习过程。算法在多个工业基准 (AKM, TCP, CAS, PC...) 上进行了测试, 结果显示 $L_{\#}$ 在 SUL 重置、输入数量以及查询效率上优于 OTALearning。但是, MMIT 的表达力弱于 TA, 无法表示时间区域 $[t1, t2]$ 的行为, 仅适用于系统时间为倒计时且仅有单一时钟的场景, 即动作变迁应该是在几秒内执行而非持续多长时间执行某个动作。

Kogel 等人^[35,36]提出学习带有本地定时器的 Mealy 机 (MMLT) 及相关的算法。MMLT 允许多个定时器, 但施加严格的约束 (如周期性定时器), MMLT 可以等价转换为 MMIT。最新提出的优化方法 MMLT/ik, 利用不精确的先验知识, 引入符号过滤器, 保证模型准确性的同时减少了学习时间。算法在 9 个现实实时系统的基准上 (HVAC, WM, Oven, WSN, ...) 进行了测试, 与不使用符号过滤器的 MMLT 算法相比, MMLT/ik 显著减少了运行时间。这一结论展示出了算法优化的巨大潜力, 为大规模实时系统自动建模提供了方案。然而, MMLT 的严格约束使其难以应对需要灵活定时器配置的复杂系统。

现有带有时钟的 Mealy 机学习算法 (如 MMIT 和 MMLT) 受限于单一定时器和严格约束, 难以应对复杂系统 (如 FDDI 协议)。此外, 定时行为的推断需处理定时器超时事件。Bruyère 等人^[37]则提出包含多个定时器的 Mealy 机——MMT 及对应算法 $L_{\#}^{MMT}$ 。他们将符号查询方法应用在时钟推理中, 扩展了 $L_{\#}$ 在定时环境下

的应用。算法首先根据主动查询构建观察树并通过符号输出查询、符号等待查询和等价查询逐步推断模型结构。其中观察树包含基础状态和前沿状态, 通过函数仿真将状态和定时器映射到目标模型中。最终算法返回与目标自动机符号等价的 MMT, 复杂度在状态数和输入上为多项式, 在定时器数为阶乘复杂度。造成这种现象主要是因为需要处理时钟重命名以及状态-时钟组合的潜在指数增长。 $L_{\#}^{MMT}$ 也指出可以通过优化符号输出查询与符号等待查询的数量来缓解这种复杂性, 关键优化在于确保对于可行的运行序列没有两个时钟同时超时, 从而使学习器能够通过较少的查询唯一确定目标自动机的行为。另外, 为了避免从观察树构建 MMT 时状态空间的阶乘膨胀, 算法构建了一个 gMMT, 在 gMMT 中, 转换更新是重命名计时器或分配常量的函数, 这一操作允许更灵活的计时器处理。同时为了减少前沿状态兼容集的大小, 算法定义了弱共传递性来减少生成的假设数量。算法在 FDDI 协议上验证了可行性, 相比于更为通用的 LearnTA 算法, 该算法无需推断守卫条件和重置, 更易实现。但算法在固定定时器数量时表现出多项式复杂度, 适合中等规模的系统推断, 即定时器的数量不宜过多。且算法假设系统无时间竞争, 因此不适合存在复杂时间竞争的系统。表 4 展示了多时钟 TA 推断算法在同一基准上的实验结果, 通过对比查询总数可以发现, $L_{\#}^{MMT}$ 比 LearnTA 所需要的查询总数要更少。但是 MMT 的表达力要低于 DTA, 相较之下, LearnTA 更为通用, 能够学习具有复杂时间约束的 DTA。然而, 这种通用性带来了更高的计算复杂度, 使得它在大型系统中缺乏可扩展性。比如学习 FDDI 协议时, 耗费了 50 min 的时间^[21]。

表 4 工业基准上的对比实验

Model	$L_{\#}^{MMT}$					LearnTA			
	Q	I	WQ ^S	OQ ^S	EQ ^S	Time (ms)	MQ	EQ	Time (ms)
AKM	4	5	22	35	2	684	12 263	11	585
CAS	8	4	60	89	3	1 344	66 067	17	4 650
Light	4	2	10	13	2	302	3 057	7	33
PC	8	9	75	183	4	2 696	245 134	23	64 900
TCP	11	8	123	366	8	3 182	11 300	15	382
Train	6	3	32	28	3	1 559	—	—	—
Oven	12	5	907	317	3	9 542	—	—	—
WSN	9	4	105	108	4	3 291	—	—	—
FDDI 1-station	9	2	32	20	1	1 105	118 193	8	—

注: |I|为输入表的大小; |WQ^S|为符号等待查询的数量; |OQ^S|为符号输出查询的数量; |EQ^S|为符号等价查询的数量

综上, 带有时钟约束的 Mealy 机为实时系统的建模提供了多样化的方法, 相比于经典的 TA 推断算法,

这些方法或简化了时钟约束推断, 或优化了查询效率, 甚至在多种现实基准上进行了可行性验证。然而, 这些

算法在表达能力和扩展性方面仍面临挑战: TDMM 等算法在大型系统上因学习时间激增而效率低下; MMIT 和 MMLT 受限于单定时器以及严格约束, 难以表示复杂时间行为; MMT 虽支持多定时器, 但其阶乘复杂度限制了在大规模系统中的应用. 未来可以开发自适应符号查询和假设折叠策略, 降低对定时器数量的阶乘依赖, 尤其针对 MMT 的大规模应用. 结合并行计算是一个不错的思路.

3 未来发展趋势、挑战与优势

TA 的主动学习方法存在众多的优化方向, 有着巨大的发展潜力. 本节主要介绍主动学习方法的未来发展趋势、挑战与优势.

3.1 未来发展趋势

TA 为各行各业包含时间属性的智能型系统提供可靠性、安全性的验证和保障, 近年来越来越多的专家学者, 利用 TA 建模现实中的实时系统, 具体涉及网络协议^[38-42]、信息物理系统^[43-48]、铁路交通系统^[49-51]、航空航天系统^[52-54]、生物和医疗系统^[55-57]等领域. 纵观全局, 其未来发展趋势将在技术革新与应用扩展的交织中持续深化, 为智能化、实时化系统的设计与验证提供更高效、精确的解决方案.

从技术革新的角度看, TA 主动学习算法的未来将聚焦于算法效率与适用性的提升, 以应对日益复杂的系统需求. 首先是数据结构与查询策略优化, 其中包含了 OT 的高效存储与检索, 旨在提升闭合性与一致性检查时的匹配效率; 索引结构的优化, 旨在加速假设模型构建时的状态等价性判断. 其次, 当前神经网络已在各大领域大放光彩, 已有一些文献^[58,59]研究从神经网络 (RNN、Transformer) 中提取 DFA. 我们认为这一应用也为 TA 学习提供了思路. 神经网络擅长从大量 MQ 的数据中提取时间模式和行为特征, 适合处理 TA 的复杂时间约束. 在非精确等价查询时加入神经网络也是一个不错的方向. 具体结合如下: 我们使用神经网络学习黑盒系统的输入-输出映射, 利用启发式方法生成高概率覆盖关键行为的测试序列, 以提升测试用例的质量, 减少查询次数的同时尽可能完善表征目标 TA. 此外, 大语言模型 (large language model, LLM) 的广泛应用极大加快了技术更迭. 文献^[60]提出 LLM 充当 MAT, 遵从主动学习框架进行 DFA 学习. 我们认为在 TA 学习中也可以借鉴这种思路. 例如, LLM 与仿真工

具 (simulink) 结合, 在 simulink 中建模实时系统, 将仿真得到的时间序列数据用于训练 LLM, 使 LLM 以较高的概率接近真实的黑盒实时系统. LLM 的上下文学习能力使其能够基于少量示例快速适应 TA 推断任务.

计算架构的创新也是技术革新的一大关键. 随着实时系统规模的增长, 单节点计算难以满足 TA 学习的高复杂度需求. 分布式与并行计算的引入将成为关键, 特别是在大规模系统的学习场景中, 通过将学习任务分解到多节点计算框架, 可以大幅提升算法的可扩展性与效率. 例如, 我们可以将反例分析的过程并行化, 将反例分解为子序列或时间片段, 使用分布式 SMT 求解器 (如 Z3 的并行版本) 分解时间约束求解任务. 研究初期使用轻量级的分布式框架 Ray 进行快速原型开发, 使用 Ray 的共享内存 Plasma 存储 OT, 主控节点通过 Ray Actor 协调任务, 合并最终学习结果. 除此之外, 主动学习与被动学习的结合也会碰撞出不一样的火花. 文献^[61]中首先根据被动学习的思想, 使用高斯过程从一组时间序列中拟合一个 TA, 再通过主动学习检查假设自动机与目标自动机的一致性, 若不一致则通过反例修正假设, 直至学到与目标一致的 TA. 这为我们提供了启发, 我们可以利用主动学习发现模型与目标系统之间的时序差异, 利用被动学习从已有迹数据细化时间约束. 比如, 可以设计一个双层学习架构: 底层通过被动学习从迹数据中提取局部时间自动机, 顶层通过主动学习整合局部模型并验证全局一致性. 这种框架能够提升学习效率, 但面临的主要挑战是局部模型的合并问题.

从应用场景的角度看, 未来, AI 浪潮推动的智能系统还有其他各类服务型机器人都将是 TA 主动学习算法的应用对象. 首先, GTC 2025 大会提出, 下一代 AI 需要理解物理规律 (如摩擦、惯性) 以实现高级机器人交互, 而将 TA 与物理引擎 (如 NVIDIA Omniverse) 结合可构建出可模拟真实物理约束的数字孪生体, 主动学习算法在智能机器人的感知算法中扮演着动态环境认知优化与安全决策验证的双重核心作用, 确保系统在不确定性下的可靠性. 其次, Gartner (全球最权威的 IT 研究和咨询公司) 预测到 2027 年, 50% 的业务决策将通过用于决策智能的 AI 智能体得到增强或实现自动化. TA 在建模和验证复杂时序行为方面的独特优势, 与决策智能 AI 智能体对实时性、可解释性、动态适应性和可靠性需求高度契合. 主动学习算法可以通

过与业务系统的交互(如查询历史决策日志或实时数据流),精确捕捉决策的时序逻辑,但是多模态数据的噪声与不一致性可能影响模型精度。

AI浪潮与智能实体系统将为TA主动学习算法的发展带来新的机遇。TA主动学习伴随AI相关主流产业的发展,未来的增长与发展是必然的,但也必然是一个漫长的成长过程。随着技术的不断进步和创新,它将不断演化和发展,为智能化系统、智能产品带来更加丰富、多样化的保障路径,从而保障趋向于通过智能系统实现万物互联的人类世界。

3.2 当前挑战

TA的主动学习算法作为一种高效的自动化建模方法为TA在不同场景的实际应用开辟了道路,指明了方向。尽管TA的主动学习在理论和实践上均取得了显著进展,但仍面临一些亟待解决的关键挑战。

1) 目前TA的主动学习算法大多只能在理论层面或者简单的受控实验中实现。具体表现为:目前的绝大多数实验并非直接与真实的黑盒系统交互,而是与事先规定好结构的状态机交互。且当前的算法难以直接应用在分布式网络的复杂系统中,这类系统往往存在多个并发的时间约束和事件,时间行为往往超出了简单TA的表达能力。例如,对于支持多时钟算法的LearnTA,它们学到的仅是具有2个进程的FDDI协议。

2) 主动学习要求与目标系统交互,但查询复杂实时系统可能代价高昂,特别是在硬件或嵌入式系统中。每次查询涉及运行测试用例以及模拟,这需要耗费大量的时间和资源。此外,EQ需要验证假设模型与目标系统的一致性,难以在有限时间内完成。

3) TA学习的主要挑战仍然是识别多个时钟及其重置信息^[62]。如前文所述,现有的方法大多集中于学习TA的一个子类,通过限制时钟的数量或通过限制时钟变量可以被重置的边,简化学习算法中对时间约束的推断,降低任务的复杂性。单一时钟的模式使TA在现实中的应用受到一定限制。

4) TA学习的复杂度高度依赖于时钟数量,过多的时钟会导致算法复杂度增加。尽管Waga^[21]、Teng等人^[22]提出了可以不限时钟数量以及重置位置的DTA学习算法,但仍存在局限性。LearnTA在多时钟场景下,普通教师的总查询次数双指数级于时钟个数,单指数级于状态数,实际应用受限。且算法在每个转换时重置时钟,并使用基于区域的守卫,通常产生复杂的自动机,

并不能保证结果自动机的最小性。DTAL虽然学到了较小的自动机,但是算法预设时钟的个数,这一点限制了应用场景,且在普通教师场景下,计算复杂度仍为指数级。

5) 现有自动机主动学习工具(LearnLib)主要是推断针对离散状态和事件的有限状态机(DFA、Mealy机、Moore机),难以捕获时间行为。而TA需要处理连续时间变量和时钟重置逻辑,这显著增加了学习复杂性,故现有工具对TA推断的支持较为有限。截止到2025年,现有工具(包括主动学习工具的轻量级框架AALpy)的最新版本仍未完全集成对TA的原生支持。利用LearnLib学习TA通常需要额外的处理或抽象机制,例如前文所述加入“定时器”的概念。且LearnLib中的算法在处理TA学习时,对时间区域进行离散化处理,这可能导致状态空间爆炸,尤其是在多时钟或复杂时间约束的场景中。

3.3 主动学习算法优势

使用主动学习算法,经过与系统交互后能自动生成时间自动机模型,相比于手动创建状态机,无需额外的人工就可以学习到其时序、状态等信息,避免出现人为误差。而自动化的特性又能节约大量建模时间和专家资源。主动学习算法相较于被动学习算法,增量式是一个显著优势,无需考虑样本数据数量与质量,支持逐步完善模型。当系统行为或功能发生转变时,算法可以灵活地追加查询以更新模型,适应动态变化。此外,主动查询系统行为,能够发现隐藏的边界情况和异常行为,生成更鲁棒的模型。在测试和调试复杂系统(如嵌入式软件)时,主动学习能发现传统测试方法难以捕捉的时序错误。

4 总结

TA作为实时系统形式化建模与验证的关键技术,其模型的自动推断已成为重要的研究方向。本文对TA的主动学习算法领域进行了系统的梳理与回顾。归纳了当前主流TA主动学习算法的技术特点与演进脉络,分析了其在时钟重置以及查询效率等方面的挑战,展望了其在算法效率提升以及应用场景扩展方面的未来发展方向。此外,我们还发现,尚无学者在定量的标准下比较上述算法在相同基准中的表现。这也是一个大有可为的方向。本文期望通过对现有工作的总结与分析,为后续研究提供有益的参考与启示。

参考文献

- 1 王戟, 詹乃军, 冯新宇, 等. 形式化方法概貌. 软件学报, 2019, 30(1): 33–61. [doi: [10.13328/j.cnki.jos.005652](https://doi.org/10.13328/j.cnki.jos.005652)]
- 2 Alur R, Dill DL. Automata for modeling real-time systems. Proceedings of the 17th International Colloquium on Automata, Languages and Programming. Coventry: Springer, 1990. 322–335.
- 3 Alur R, Dill DL. A theory of timed automata. Theoretical Computer Science, 1994, 126(2): 183–235. [doi: [10.1016/0304-3975\(94\)90010-8](https://doi.org/10.1016/0304-3975(94)90010-8)]
- 4 Clarke EM, Henzinger TA, Veith H, *et al.* Handbook of Model Checking. Cham: Springer, 2018. 978–973.
- 5 Connes V, De La Higuera C, Le Capitaine H. Using grammatical inference to build privacy preserving data-sets of user logs. Proceedings of the 15th International Conference on Grammatical Inference. Nantes: PMLR, 2021. 176–190.
- 6 Xu QH, Ali S, Yue T. Digital twin-based anomaly detection in cyber-physical systems. Proceedings of the 14th IEEE Conference on Software Testing, Verification and Validation (ICST). Porto de Galinhas: IEEE, 2021. 205–216.
- 7 Bengtsson J, Yi W. Timed automata: Semantics, algorithms and tools. In: Desel J, Reisig W, Rozenberg G, eds. Lectures on Concurrency and Petri Nets. Berlin: Springer, 2004. 87–124.
- 8 Alur R, Madhusudan P. Decision problems for timed automata: A survey. International School on Formal Methods for the Design of Computer, Communication, and Software Systems. Bertinoro: Springer, 2004. 1–24.
- 9 Waez MTB, Dingel J, Rudie K. A survey of timed automata for the development of real-time systems. Computer Science Review, 2013, 9: 1–26. [doi: [10.1016/j.cosrev.2013.05.001](https://doi.org/10.1016/j.cosrev.2013.05.001)]
- 10 Fontana P, Cleaveland R. A menagerie of timed automata. ACM Computing Surveys, 2014, 46(3): 40.
- 11 Keiren JJA, Fontana P, Cleaveland R. Corrections to “A menagerie of timed automata”. ACM Computing Surveys, 2018, 50(3): 42.
- 12 André É. What’s decidable about parametric timed automata? International Journal on Software Tools for Technology Transfer, 2019, 21(2): 203–219.
- 13 Arcile J, André É. Timed automata as a formalism for expressing security: A survey on theory and practice. ACM Computing Surveys, 2023, 55(6): 127.
- 14 Angluin D. Learning regular sets from queries and counterexamples. Information and Computation, 1987, 75(2): 87–106. [doi: [10.1016/0890-5401\(87\)90052-6](https://doi.org/10.1016/0890-5401(87)90052-6)]
- 15 Grinchtein O, Jonsson B, Leucker M. Inference of timed transition systems. Electronic Notes in Theoretical Computer Science, 2005, 138(3): 87–99. [doi: [10.1016/j.entcs.2005.02.062](https://doi.org/10.1016/j.entcs.2005.02.062)]
- 16 Grinchtein O, Jonsson B, Leucker M. Learning of event-recording automata. Theoretical Computer Science, 2010, 411(47): 4029–4054. [doi: [10.1016/j.tcs.2010.07.008](https://doi.org/10.1016/j.tcs.2010.07.008)]
- 17 Lin SW, André É, Dong JS, *et al.* An efficient algorithm for learning event-recording automata. Proceedings of the 9th International Symposium on Automated Technology for Verification and Analysis (ATVA). Taipei: Springer, 2011. 463–472.
- 18 An J, Chen MS, Zhan BH, *et al.* Learning one-clock timed automata. Proceedings of the 26th International Conference on Tools and Algorithms for the Construction and Analysis of Systems. Dublin: Springer, 2020. 444–462.
- 19 An J, Wang LT, Zhan BH, *et al.* Learning real-time automata. Science China Information Sciences, 2021, 64(9): 192103. [doi: [10.1007/s11432-019-2767-4](https://doi.org/10.1007/s11432-019-2767-4)]
- 20 Xu RQ, An J, Zhan BH. Active learning of one-clock timed automata using constraint solving. Proceedings of the 20th International Symposium on Automated Technology for Verification and Analysis. Berlin: Springer, 2022. 249–265.
- 21 Waga M. Active learning of deterministic timed automata with Myhill-Nerode style characterization. Proceedings of the 35th International Conference on Computer Aided Verification. Paris: Springer, 2023. 3–26.
- 22 Teng Y, Zhang MM, An J. Learning deterministic multi-clock timed automata. Proceedings of the 27th ACM International Conference on Hybrid Systems: Computation and Control. Hong Kong: ACM, 2024. 6.
- 23 Grinchtein O, Jonsson B, Pettersson P. Inference of event-recording automata using timed decision trees. Proceedings of the 17th International Conference on Concurrency Theory (CONCUR). Bonn: Springer, 2006. 435–449.
- 24 Henry L, Jérón T, Markey N. Active learning of timed automata with unobservable resets. Proceedings of the 18th International Conference on Formal Modeling and Analysis of Timed Systems (FORMATS). Vienna: Springer, 2020. 144–160.
- 25 Mi JR, Xu J. The learning algorithm of real-time automata. Proceedings of the 6th IEEE International Conference on Computer and Communications (ICCC). Chengdu: IEEE, 2020. 2146–2150.
- 26 米钧日, 张苗苗, 安杰, 等. 运用时间分类树的确定单时钟时间自动机学习. 软件学报, 2022, 33(8): 2797–2814. [doi: [10.13328/j.cnki.jos.005652](https://doi.org/10.13328/j.cnki.jos.005652)]

- 10.13328/j.cnki.jos.006599]
- 27 Berg T, Grinchtein O, Jonsson B, *et al.* On the correspondence between conformance testing and regular inference. Proceedings of the 8th International Conference on Fundamental Approaches to Software Engineering (FASE). Edinburgh: Springer, 2005. 175–189.
- 28 Aichernig BK, Tappler M. Efficient active automata learning via mutation testing. Journal of Automated Reasoning, 2019, 63(4): 1103–1134. [doi: 10.1007/s10817-018-9486-0]
- 29 Chen YF, Hsieh C, Lengál O, *et al.* PAC learning-based verification and model synthesis. Proceedings of the 38th International Conference on Software Engineering. Austin: IEEE, 2016. 714–724.
- 30 Maler O, Mens IE. A generic algorithm for learning symbolic automata from membership queries. In: Aceto L, Bacci G, Bacci G, *et al.*, eds. Models, Algorithms, Logics and Tools: Essays Dedicated to Kim Guldstrand Larsen on the Occasion of His 60th Birthday. Cham: Springer, 2017. 146–169.
- 31 Shen W, An J, Zhan BH, *et al.* PAC learning of deterministic one-clock timed automata. Proceedings of the 22nd International Conference on Formal Methods and Software Engineering. Singapore: Springer, 2020. 129–146.
- 32 Tang XC, Shen W, Zhang MM, *et al.* Learning deterministic one-clock timed automata via mutation testing. Proceedings of the 20th International Symposium on Automated Technology for Verification and Analysis. Beijing: Springer, 2022. 233–248.
- 33 Caldwell B, Cardell-Oliver R, French T. Learning time delay Mealy machines from programmable logic controllers. IEEE Transactions on Automation Science and Engineering, 2016, 13(2): 1155–1164. [doi: 10.1109/TASE.2015.2496242]
- 34 Vaandrager F, Ebrahimi M, Bloem R. Learning Mealy machines with one timer. Information and Computation, 2023, 295: 105013. [doi: 10.1016/j.ic.2023.105013]
- 35 Kogel P, Klös V, Glesner S. Learning Mealy machines with local timers. Proceedings of the 24th International Conference on Formal Methods and Software Engineering. Brisbane: Springer, 2023. 47–64.
- 36 Kogel P, Schwabe W, Glesner S. MMLT/ik: Efficiently learning Mealy machines with local timers by using imprecise symbol filters. Proceedings of the 1st International Conference on Quantitative Evaluation of Systems and Formal Modeling and Analysis of Timed Systems. Calgary: Springer, 2024. 143–159.
- 37 Bruyère V, Garhewal B, Pérez GA, *et al.* Active learning of Mealy machines with timers. arXiv:2403.02019, 2024.
- 38 Wang J, Chen ZC, Hou G, *et al.* A modeling and verification method of Modbus TCP/IP protocol. Proceedings of the 21st International Conference on Algorithms and Architectures for Parallel Processing. Berlin: Springer, 2022. 527–539.
- 39 Lv J, Zhao YX, Wu X, *et al.* Formal analysis of TSN scheduler for real-time communications. IEEE Transactions on Reliability, 2021, 70(3): 1286–1294. [doi: 10.1109/TR.2020.3026689]
- 40 Hmidi Z, Kahloul L, Benharzallah S. A new mobility and energy harvesting aware medium access control (MEH-MAC) protocol: Modelling and performance evaluation. Ad Hoc Networks, 2023, 142: 103108. [doi: 10.1016/j.adhoc.2023.103108]
- 41 Guo XY, Aoki T, Lin HH. Model checking of in-vehicle networking systems with CAN and FlexRay. Journal of Systems and Software, 2020, 161: 110461. [doi: 10.1016/j.jss.2019.110461]
- 42 Roumane A, Kechar B, Kouninef B. Formal verification of a radio network random access protocol. International Journal of Communication Systems, 2017, 30(18): e3447. [doi: 10.1002/dac.3447]
- 43 Jawad A, Jaskolka J. Analyzing the impact of cyberattacks on industrial control systems using timed automata. Proceedings of the 21st IEEE International Conference on Software Quality, Reliability and Security. Hainan: IEEE, 2021. 966–977.
- 44 Gao HH, Zhang YD, Miao HK, *et al.* SDTIOA: Modeling the timed privacy requirements of IoT service composition: A user interaction perspective for automatic transformation from BPEL to timed automata. Mobile Networks and Applications, 2021, 26(6): 2272–2297. [doi: 10.1007/s11036-021-01846-x]
- 45 Canadas N, Machado J, Soares F, *et al.* Simulation of cyber physical systems behaviour using timed plant models. Mechatronics, 2018, 54: 175–185. [doi: 10.1016/j.mechatronics.2017.10.009]
- 46 Seceleanu C, Johansson M, Suryadevara J, *et al.* Analyzing a wind turbine system: From simulation to formal verification. Science of Computer Programming, 2017, 133: 216–242. [doi: 10.1016/j.scico.2016.09.007]
- 47 Machado J, Galvão J, Fernandes A. Formal verification considering a systematic modeling approach for function blocks. Journal of the Brazilian Society of Mechanical Sciences and Engineering, 2017, 39(10): 4107–4113. [doi: 10.1007/s40430-017-0893-7]
- 48 安冬冬, 刘静, 陈小红, 等. 不确定环境下 hCPS 系统的形式

- 化建模与动态验证. 软件学报, 2021, 32(7): 1999–2015. [doi: 10.13328/j.cnki.jos.006272]
- 49 Basile D, Fantechi A, Rucher L, *et al.* Analysing an autonomous tramway positioning system with the UPPAAL Statistical Model Checker. *Formal Aspects of Computing*, 2021, 33(6): 957–987. [doi: 10.1007/s00165-021-00556-1]
- 50 Lv JD, Ahmad E, Tang T. Non-deterministic delay behavior testing of Chinese train control system using UPPAAL-TRON. *IEEE Intelligent Transportation Systems Magazine*, 2021, 13(3): 58–82. [doi: 10.1109/IMITS.2019.2953536]
- 51 Himrane O, Beugin J, Ghazel M. Implementation of a model-oriented approach for supporting safe integration of GNSS-based virtual balises in ERTMS/ETCS level 3. *IEEE Open Journal of Intelligent Transportation Systems*, 2023, 4: 294–310. [doi: 10.1109/OJITS.2023.3267142]
- 52 de Moraes RS, Nadjm-Tehrani S. Abstraction models for verifying resource adequacy of IMA systems at concept level. *Science of Computer Programming*, 2021, 208: 102654. [doi: 10.1016/j.scico.2021.102654]
- 53 肖思慧, 刘琦, 黄滢鸿, 等. 基于 SysML 的机载软件分层精细化建模与验证方法. 软件学报, 2022, 33(8): 2851–2874. [doi: 10.13328/j.cnki.jos.006602]
- 54 Liu HY, Liu J, Sun HY, *et al.* Uncertainty-aware behavior modeling and quantitative safety evaluation for automatic flight control systems. *Proceedings of the 22nd IEEE International Conference on Software Quality, Reliability and Security*. Guangzhou: IEEE, 2022. 549–560.
- 55 Wetselaar P, Lobbezoo F, de Jong P, *et al.* A methodology for evaluating tooth wear monitoring using timed automata modelling. *Journal of Oral Rehabilitation*, 2020, 47(3): 353–360. [doi: 10.1111/joor.12908]
- 56 Newaz AI, Aris A, Sikder AK, *et al.* Systematic threat analysis of modern unified healthcare communication systems. *Proceedings of the 2022 IEEE Global Communications Conference*. Rio de Janeiro: IEEE, 2022. 1404–1410.
- 57 Alshalalfah AL, Hamad GB, Mohamed OA. Towards system level security analysis of artificial pancreas via UPPAAL-SMC. *Proceedings of the 2019 IEEE International Symposium on Circuits and Systems*. Sapporo: IEEE, 2019. 1–5.
- 58 Umili E, Capobianco R. DeepDFA: Automata learning through neural probabilistic relaxations. *Proceedings of the 27th European Conference on Artificial Intelligence*. Santiago de Compostela: IOS Press, 2024. 1051–1058.
- 59 Zhang YH, Wei ZM, Sun M. Automata extraction from Transformers. arXiv: 2406.05564, 2024.
- 60 Chen LK, Trivedi A, Velasquez A. LLMs as probabilistic minimally adequate teachers for DFA learning. arXiv: 2408.02999, 2024.
- 61 Aichernig BK, Pferscher A, Tappler M. From passive to active: Learning timed automata efficiently. *Proceedings of the 12th International Symposium on NASA Formal Methods*. Moffett Field: Springer, 2020. 1–19.
- 62 Cornanguer L. Timed automata learning from time series [Ph.D. Thesis]. Rennes: Université de Rennes, 2023.

(校对责编: 李慧鑫)