第 40 卷 第 3 期
2025 年 6 月

天津科技大学学报
Journal of Tianjin University of Science and Technology

Vol. 40 No. 3
Jun. 2025

DOI:10.13364/j.issn.1672-6510.20240024
网络首发日期：2024–11–05；网络首发地址：http://link.cnki.net/urlid/12.1355.n.20241105.1347.001

基于 HyperLTL 模型检测技术的 *K* 步不透明性验证

任　翔[1]，王子佩[1]，张佳慧[2]，韩晓光[1]
（1. 天津科技大学电子信息与自动化学院，天津 300222；2. 澳门科技大学澳门系统工程研究所，澳门 999078）

摘　要：不透明性是一种重要的信息流安全属性，用于表征不同动态系统中各种安全和隐私需求。在离散事件系统中，*K* 步不透明性刻画的是入侵者无法根据当前的输出观测序列确定系统是否在 *K* 个观测步之内曾访问过秘密状态。HyperLTL 作为一种在形式化验证中表达信息流属性的工具，已被证明可以采用一个统一的 Kripke 结构分别验证初始状态不透明性、当前状态不透明性和无限步不透明性，但并未涉及 *K* 步不透明性。针对 HyperLTL 模型的 *K* 步不透明性验证问题，通过添加 Sink 状态释放了对原系统的限制性条件，进一步构造改进的 Kripke 结构并给出 *K* 步不透明性在 HyperLTL 中的判据，即验证原系统在 HyperLTL 语义下是否满足 *K* 步不透明性。

关键词：离散事件系统；*K* 步不透明性；HyperLTL；改进的 Kripke 结构；Sink 状态
中图分类号：TP273　　**文献标志码**：A　　**文章编号**：1672-6510(2025)03-0065-07

K-Step Opacity Verification Based on HyperLTL Model Checking Technology

REN Xiang[1]，WANG Zipei[1]，ZHANG Jiahui[2]，HAN Xiaoguang[1]
（1. College of Electronic Information and Automation, Tianjin University of Science and Technology, Tianjin 300222, China；
2. Macao Institute of Systems Engineering, Macau University of Science and Technology, Macao 999078, China）

Abstract：Opacity is an important information-flow security property used to characterize various security and privacy requirements in different dynamic systems. In discrete-event systems, *K*-step opacity describes the inability of an intruder to determine whether the system has accessed the secret state within *K* observation steps based on the current output observation sequence. As a suitable tool for expressing the properties of information flows, HyperLTL has been proven to use a unified Kripke structure to separately verify initial-state opacity, current-state opacity and infinite-step opacity. However, it does not involve *K*-step opacity. To solve the verification of the *K*-step opacity in HyperLTL, we added the Sink state to release the restriction condition of the original system. Then, we constructed the modified Kripke structure. Furthermore, we provided the HyperLTL formula of *K*-step opacity over the constructed modified Kripke structure for the purpose of verification.

Key words：discrete-event system；*K*-step opacity；HyperLTL；modified Kripke structure；Sink state

引文格式：
任翔，王子佩，张佳慧，等. 基于 HyperLTL 模型检测技术的 *K* 步不透明性验证[J]. 天津科技大学学报，2025，40(3)：65–71.
REN X，WANG Z P，ZHANG J H，et al. *K*-step opacity verification based on HyperLTL model checking technology[J]. Journal of Tianjin university of science and technology，2025，40(3)：65–71.

通信技术的飞速发展和数据处理能力的不断提高引发了诸多安全和隐私问题，这对实际系统在运行过程中保护秘密信息的能力提出了更高的要求。离散事件系统（discrete-event systems，DES）是一类具有

收稿日期：2024–02–21；**修回日期**：2024–05–08
基金项目：国家自然科学基金项目(61903274)；天津市教委科研计划一般项目(2019KJ212)
作者简介：任　翔(1998—)，男，天津人，硕士研究生；**通信作者**：韩晓光，副教授，hxg-allen@163.com

离散状态空间和事件触发动态变化特征的重要信息物理系统，它凭借强大的建模能力，被广泛应用于分析制造系统、自主机器人等领域的高级逻辑行为。近来，在计算机科学相关文献中发展出一种新的时序逻辑，被称为 HyperLTL[1]，其作为线性时序逻辑（LTL）的推广，广泛应用于对信息流属性的形式化验证。本文研究了在 HyperLTL 框架下，DES 中 K 步不透明性的验证问题。

不透明性[2]表征系统的秘密行为对潜在的恶意入侵者隐藏其机密信息的能力，具体可描述为：如果对系统的每个秘密行为，都存在一个使入侵者产生混淆的非秘密行为，则系统是不透明的。不透明性概念起源于计算机科学领域中对加密协议的分析[3]。此后，在有限状态自动机建模的 DES 背景下，发展出了多种类型的不透明性概念，这些概念包括：当前状态不透明性[4]、初始状态不透明性[5]、无限步不透明性[6]、K 步不透明性[7]及基于语言的不透明性[8-9]。K 步不透明性描述的是入侵者永远无法确定系统在 K 个观测步之前是否处于秘密状态。在实际应用中，由于标准不透明性存在一定的局限性，研究人员相继提出了强 K 步不透明性[10]、强无限步不透明性[11]、强当前状态不透明性[12]、强初始状态不透明性[12]，4 种强版本不透明性概念并设计有效算法对其进行验证[12-13]。此外，作者证明了各类不透明性之间可以相互转换，即对归约问题进行了研究[14]。文献[15]在 HyperLTL 框架下，通过构造 Kripke 结构，为初始状态不透明性、当前状态不透明性、无限步不透明性概念的验证提供了一种统一方法。然而，未对 K 步不透明性进行研究。

本文研究了 HyperLTL 模型下 K 步不透明性的验证问题。受文献[15]启发，通过构造改进的 Kripke 结构，提出了一种基于 HyperLTL 语义的 K 步不透明性判据，进而对系统的 K 步不透明性进行验证。此外，通过添加 Sink 状态[16]，将死锁系统转换成非死锁系统，释放了文献[15]中对原系统的两个限制条件，使适用场景更加广泛。最后，通过实例说明了本文所设计验证方法的可行性和有效性。

1　预备知识

1.1　系统模型

设 Σ 为一个有限事件集，有限（无穷）字符串 $s=\sigma_1\cdots\sigma_n(\cdots)$ 是一个有限（无穷）事件序列，其中 $\sigma_i\in\Sigma$，分别用 Σ^* 和 Σ^ω 表示 Σ 上所有有限（包括空序列）和无穷序列的集合。一个有限序列 $s\in\Sigma^*$ 的长度被定义为 $|s|$（$|\epsilon|=0$），$*$-语言（ω-语言）是一个有限（无穷）序列集。给定一个语言 $L\subseteq\Sigma^*\cup\Sigma^\omega$，语言 L 的前缀闭包表示为 $\overline{L}$，其被定义为它所有有限前缀的集合，即

$$\overline{L}=\{s\in\Sigma^*:\exists\omega\in\Sigma^*\cup\Sigma^\omega \text{ s.t. } s\omega\in L\} \tag{1}$$

如果一个 $*$-语言 $L=\overline{L}$（$L\subseteq\Sigma^*$），那么它是前缀闭包的。

本文将一个离散事件系统（下文简称系统）建模为有限状态自动机

$$G=(X,\Sigma,\delta,X_0) \tag{2}$$

其中 X 是有限状态集，Σ 是有限事件集，$X_0\subseteq X$ 是初始状态集，$\delta:X\times\Sigma\to 2^X$ 是跃迁函数，描述了系统动态行为：给定状态 $x,y\in X$ 和一个事件 $\sigma\in\Sigma$，$y\in\delta(x,\sigma)$ 表示存在一个 σ 标记的事件从状态 x 到状态 y 的跃迁。通过递归可将跃迁函数 δ 扩展为 $\delta:X\times\Sigma^*\to 2^X$。$\mathcal{L}(G,x)$ 表示系统 G 从状态 x 出发生成的 $*$-语言，即 $\mathcal{L}(G,x)=\{s\in\Sigma^*:\delta(x,s)!\}$，其中 $!$ 表示该跃迁有定义。为方便起见，定义 $\mathcal{L}(G)=\bigcup_{x_0\in X_0}L(G,x_0)$ 为系统 G 生成的 $*$-语言。同样，定义 $\mathcal{L}^\omega(G,x)=\{s\in\Sigma^\omega:\forall t\in\overline{\{s\}} \text{ s.t. } \delta(x,t)!\}$ 为系统 G 从状态 x 出发生成的 ω-语言并定义为 $\mathcal{L}^\omega(G)=\bigcup_{x_0\in X_0}\mathcal{L}^\omega(G,x_0)$。

不失一般性，假设入侵者只能看到系统的局部行为。为此，考虑一种观测掩码函数，为

$$M:\Sigma\to\mathcal{O}\cup\{\epsilon\} \tag{3}$$

其中 $\mathcal{O}$ 是观测符号的集合。如果 $M(\sigma)=\epsilon$，那么称事件 $\sigma\in\Sigma$ 是不可观测的；否则事件 σ 是可观测的。Σ_o 和 Σ_{uo} 分别表示可观测事件集和不可观测事件集。如果 $M(\sigma)=M(\sigma')$，则称事件 $\sigma,\sigma'\in\Sigma$ 是不可区分的。掩码函数 M 可扩展为 $M:\Sigma^*\cup\Sigma^\omega\to\mathcal{O}^*\cup\mathcal{O}^\omega$，使得对于任意 $s\in\mathcal{L}(G)$，有：(a) $M(\epsilon)=\epsilon$；(b) 对于任意 $s\in\Sigma^*,\sigma\in\Sigma$，$M(s\sigma)=M(s)M(\sigma)$。对于任意 $L\subseteq\Sigma^*\cup\Sigma^\omega:M(L)=\{M(s):s\in L\}$，掩码可扩展为 $M:2^{\Sigma^*\cup\Sigma^\omega}\to 2^{\mathcal{O}^*\cup\mathcal{O}^\omega}$。因此，$M(\mathcal{L}(G))$ 和 $M(\mathcal{L}^\omega(G))$ 分别表示由系统 G 产生并观测到的 $*$-语言和 ω-语言。

由于系统是部分可观测的，系统用户需要根据观测序列确定系统的状态，此问题被称为状态估计。考虑延迟状态估计，将使用延迟状态估计表示 K 步不透明性定义。

定义 1（延迟状态估计[17]）：设 $\alpha,\alpha\beta\in M(\mathcal{L}(G))$，为一个观测序列，其中 α 是观测序列 $\alpha\beta$ 的前缀，观测 $\alpha\beta$ 时 α 时刻的延迟状态估计表示为：观测到 $\alpha\beta$

时系统在 $|\beta|$ 步前的状态集，即

$$\hat{X}_G(\alpha|\alpha\beta)=\left\{\delta(x_0,s)\in X:\begin{matrix}\exists x_0\in X_0, s\omega\in\mathcal{L}(G,x_0)\ \text{s.t.}\\ M(s)=\alpha\wedge M(s\omega)=\alpha\beta\end{matrix}\right\} \tag{4}$$

1.2　LTL 与 HyperLTL

设 $\mathcal{AP}$ 为用户所感兴趣的基本属性原子命题的集合。轨迹 $\pi=\pi_0\pi_1\cdots\in(2^{\mathcal{AP}})^\omega$ 是 $2^{\mathcal{AP}}$ 上的一条无穷序列。$\pi[i]=\pi_i$ 表示轨迹 π 上的第 i 个元素并用 $\pi[i,\infty]=\pi_i\pi_{i+1}\cdots\in(2^{\mathcal{AP}})^\omega$ 表示轨迹 π 从第 i 时刻开始的后缀。线性时序逻辑（LTL）是一种广泛用于评估轨迹 π 是否满足某些属性的方法。LTL 公式的语法为

$$\psi::=a\,|\,\neg\psi\,|\,\psi\vee\psi\,|\,\bigcirc\psi\,|\,\psi\mathcal{U}\psi \tag{5}$$

其中：$a\in\mathcal{AP}$ 是一个原子命题，$\neg$ 和 $\vee$ 分别表示布尔运算符“否定”和“或”，$\bigcirc$ 和 $\mathcal{U}$ 分别表示时序运算符“下一个”和“直到”。通过上述符号，更多的布尔运算符被定义，例如：用 $\psi_1\to\psi_2\equiv\neg\psi_1\wedge\psi_2$ 表示布尔运算符“蕴含”，用 $\psi_1\wedge\psi_2\equiv\neg(\neg\psi_1\vee\neg\psi_2)$ 表示布尔运算符“与”。可以进一步通过 $\Diamond\psi\equiv\top\mathcal{U}\psi$ 和 $\Box\psi\equiv\neg\Diamond\neg\psi$ 分别定义时序运算符“最终”和“总是”。LTL 的语义可以在文献[18]中查阅。当轨迹 π 满足 LTL 公式 ψ 时，用 $\pi\vDash\psi$ 表示。

需要注意的是，LTL 只能用于评估单一轨迹的正确定，然而实际系统的属性验证问题，往往需要在多条轨迹的范围内进行评估。例如在不透明性的分析中，需要检测是否存在多条轨迹具有相同的观测。HyperLTL 通过支持轨迹量词对 LTL 进行了推广。形式上，设 $\mathcal{V}=\{\pi_1,\pi_2,\cdots\}$ 为轨迹变量的集合，其中每一个 π_i 代表一个单独的轨迹。HyperLTL 公式的语法[1]为

$$\phi::=\exists\pi.\ \phi\,|\,\forall\pi.\ \phi\,|\,\psi \tag{6}$$

$$\psi::=a^\pi\,|\,\neg\psi\,|\,\psi\vee\psi\,|\,\bigcirc\psi\,|\,\psi\mathcal{U}\psi \tag{7}$$

其中 $\exists$ 和 $\forall$ 分别是轨迹量词，表示“存在”和“任意”。除了原子命题可以引用不同的轨迹变量之外，ψ 仅是一个 LTL 公式。由于 HyperLTL 公式可以引用多个轨迹，用 a^π 表示应检查轨迹 π 是否满足原子命题 $a\in\mathcal{AP}$。

HyperLTL 的语义定义在轨迹集合 $T\subseteq(2^{\mathcal{AP}})^\omega$ 和映射 $\Pi:\mathcal{V}\to(2^{\mathcal{AP}})^\omega$ 上。用 $\prod_\varnothing$ 表示定义域为空集的空分配。用 $\prod[\pi\mapsto\xi]$ 表示与 $\prod$ 相同的轨迹分配，期望 π 映射到 ξ。轨迹分配后缀 $\prod[i,\infty]$ 表示所有 π 的轨迹分配 $\prod'(\pi)=\prod(\pi)[i,\infty]$。用 $(T,\prod)\vDash\phi$ 表示 HyperLTL ϕ 在轨迹集 $T\subseteq(2^{\mathcal{AP}})^\omega$ 与轨迹分配 $\Pi:\mathcal{V}\to(2^{\mathcal{AP}})^\omega$ 上满足 ϕ，其定义如下（iff 表示当且仅当）：

$(T,\prod)\vDash\exists\pi.\phi$ iff $\exists\xi\in T:(T,\prod[\pi\mapsto\xi])\vDash\phi$

$(T,\prod)\vDash\forall\pi.\phi$ iff $\forall\xi\in T:(T,\prod[\pi\mapsto\xi])\vDash\phi$

$(T,\prod)\vDash a^\pi$ iff $a\in\prod(\pi)[0]$

$(T,\prod)\vDash\neg\psi$ iff $(T,\prod)\nvDash\psi$

$(T,\prod)\vDash\psi_1\vee\psi_2$ iff $(T,\prod)\vDash\psi_1$ or $(T,\prod)\vDash\psi_2$

$(T,\prod)\vDash\bigcirc\psi$ iff $(T,\prod[1,\infty])\vDash\psi$

$(T,\prod)\vDash\psi_1\mathcal{U}\psi_2$ iff $(\exists i\geq0:(T,\prod[i,\infty])\vDash\psi_2)\wedge(\forall0\leqslant j<i:(T,\prod[j,\infty])\vDash\psi_1)$

1.3　Kripke 结构

在 HyperLTL 的模型检测中，轨迹集 $T\subseteq(2^{\mathcal{AP}})^\omega$ 通常由一个 Kripke 结构生成。形式上，Kripke 结构是一个五元组 $K_S=(Q,Q_0,\Delta,\mathcal{AP},L)$，其中 Q 是状态集，$Q_0\subseteq Q$ 是初始状态集，$\Delta\subseteq Q\times Q$ 是转换函数，$\mathcal{AP}$ 是原子命题集，$L:Q\to2^{\mathcal{AP}}$ 是标记函数。

如果 $\rho_0\in Q_0$ 并且 $\langle\rho_i,\rho_{i+1}\rangle\in\Delta,\forall i\geq0$，那么 $\rho=\rho_0\rho_1\cdots\in Q^\omega$ 是 K_S 中的一条运行。如果存在一条运行 $\rho=\rho_0\rho_1\cdots\in Q^\omega$ 使 $\pi_i=L(\rho_i),\forall i\geq0$，那么 $\pi=\pi_0\pi_1\cdots\in Q^\omega$ 是 K_S 中的一条轨迹。用 $\mathrm{Run}(K_S)$ 和 $\mathrm{Trace}(K_S)$ 分别表示由 K_S 生成的所有运行和轨迹的集合。继而，如果 $\mathrm{Trace}(K_S)\vDash\phi$，那么 Kripke 结构 K_S 满足 HyperLTL 公式 ϕ，表示为 $K_S\vDash\phi$。

1.4　K 步不透明性

在本小节中，回顾 K 步不透明性的形式化定义。假设系统中存在某种“秘密”且存在一个可以通过观测掩码观测事件发生的入侵者。不透明性对“秘密”可能会经过信息流泄露给入侵者的机密性进行捕捉。在 DES 背景下，系统中的“秘密”通常被建模为秘密状态集 $X_{\mathrm{S}}\subseteq X$，同时自然地将状态空间划分为 $X=X_{\mathrm{S}}\dot{\cup}X_{\mathrm{NS}}$，其中 $X_{\mathrm{NS}}=X\setminus X_{\mathrm{S}}$ 为非秘密状态集。

定义 2（K 步不透明性[7]）：给定系统 $G=(X,\Sigma,\delta,X_0)$，一个非负整数 $K\in\mathbf{N}$（$\mathbf{N}$ 为非负整数集合），观测掩码 $M:\Sigma\to\mathcal{O}\cup\{\epsilon\}$，秘密状态集 $X_{\mathrm{S}}\subseteq X$。如果有

$$\left(\forall\alpha\beta\in M(\mathcal{L}(G)):|\beta|\leqslant K\right)\left[\hat{X}_G(\alpha|\alpha\beta)\not\subseteq X_{\mathrm{S}}\right] \tag{8}$$

则称系统 G 是关于 M 和 X_{S} K 步不透明的。

例 1　考虑图 1 所示系统 $G=(X,\Sigma,\delta,X_0)$，其中 $X=\{x_0,x_1,x_2,x_3,x_4,x_5\}$，$X_{\mathrm{S}}=\{x_3\}$，$\Sigma_{\mathrm{o}}=\{a,b,c,d,e\}$，$\Sigma_{\mathrm{uo}}=\{u\}$，观测掩码 M：$M(a)=M(b)=o_1$，$M(c)=M(d)=o_2$，$M(e)=o_3$，$M(u)=\epsilon$，$\mathcal{O}=\{o_1,o_2,o_3\}$。对序列 $st=bd$：$s=b$，$t=d$，使 $\delta(x_1,s)=\{x_3\}$，存在一个

与之观测等价的序列 $s't'=ac:s'=a$，$t'=c$，使 $\delta(x_0,s')=\{x_2\}$，根据定义 2 可知，系统 G 是 K 步不透明的。

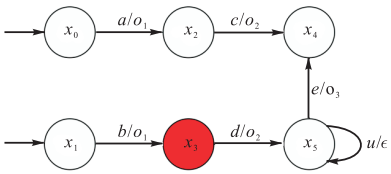

图 1　系统 G
Fig. 1　System G

2　Kripke 结构建模的离散事件系统

2.1　用于 DES 的 Kripke 结构

在这一小节中，讨论在 DES 中如何运用 Sink 状态将死锁系统转换为非死锁系统并对其构造 Kripke 结构，以验证系统的 K 步不透明性。

HyperLTL 模型检测技术虽然为多种不透明性的验证提供了一种统一方法，但其在本质上存在一些缺陷。由于 HyperLTL 模型检测技术适用于无死锁的转换系统，因此与大量相关文献一样，文献[15]在分析系统时做出如下标准假设：(a) 系统是活的，即 $\forall x\in X,\exists\sigma\in\Sigma:\delta(x,\sigma)!$；(b) 系统不包含不可观测的循环，即 $\forall x\in X,\forall s\in\Sigma^*\setminus\{\epsilon\}:x=\delta(x,s)!\Rightarrow M(s)\neq\epsilon$。此假设制约了系统在工程实际中的应用范围。因此，通过向死锁系统中添加 Sink 状态，将原系统转换为非死锁系统，以释放上述限制性假设并有效提升了其在工程中的适用度。具体转换定义如下。

转换 1　考虑一个死锁系统 G（如图 1 所示），在系统 G 中添加一个新的 Sink 状态 $\Diamond$，令 $\Diamond\in X$，在每个死锁状态及不可观测事件自循环中的每个状态 x'，添加一个跃迁 $x'\xrightarrow{\Diamond_g/\Diamond_g}\Diamond$，令 $M(\Diamond_g)=\Diamond_g\in\Sigma$，在 Sink 状态 $\Diamond$ 上添加一个自循环跃迁 $\Diamond\xrightarrow{\Diamond_s/\Diamond_s}\Diamond$，令 $M(\Diamond_s)=\Diamond_s\in\Sigma$，转换后的系统表示为 $G^\Diamond$。

备注 1　由于在原系统 G 中添加 Sink 状态可能会改变该系统原本所具有的性质，因此为方便起见，本文只考虑添加 Sink 状态后不改变其 K 步不透明性的这样一类系统。若原系统为非死锁系统，则转换后的系统 $G^\Diamond$ 与原系统 G 结构相同，即 $G^\Diamond=G$。

图 2 为说明性示例，在死锁状态 x_4 与不可观测事件自循环中的状态 x_5 之后添加 Sink 状态 $\Diamond$，将图 1 中死锁系统 G 转换为非死锁系统 $G^\Diamond$。

为得到系统 $G^\Diamond$ 中的所有无穷轨迹，需要构造 K_G 结构。在 HyperLTL 模型检测中，原子命题通常分配给系统模型中的每个状态或转换。然而，在 DES 中，对于任意内部序列 $s=\sigma_1\sigma_2\cdots\in\Sigma^\omega$，它的观测序列为 $M(\alpha)=o_1o_2\cdots\in\mathcal{O}^\omega$，其中对于每一个 $i\geqslant 1$，o_i 不一定是事件 σ_i 的观测，因为中间可能存在不可观测的序列，内部序列的时间标识与其信息流可能不匹配。

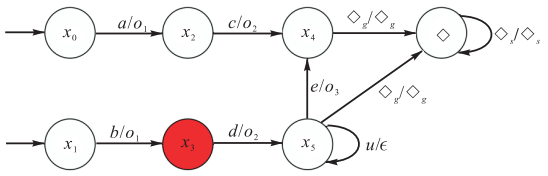

图 2　系统 G 经转换 1 后得到的系统 $G^\Diamond$
Fig. 2　System $G^\Diamond$ obtained after transformation 1 of system G

为解决不可观测问题，设 $x,x'\in X$ 是 G 中的两个状态，$o\in\Delta\cup\{\epsilon\}$ 是一个包括空观测 ϵ 的观测符号。如果 x 能经过观测为 o 的序列到达 x'，那么用 $x\overset{o}{\rightsquigarrow}x'$ 对此过程进行定义。

定义 3（DES 中的 Kripke 结构）：给定经过转换且带有观测掩码 $M:\Sigma\to\mathcal{O}\cup\{\epsilon\}$ 的非死锁离散事件系统 $G^\Diamond$，其相关的 Kripke 结构定义为

$$K_G^\Diamond=(Q,Q_0,\Delta,\mathcal{AP},L)\tag{9}$$

其中：$Q\subseteq X\times(\mathcal{O}\cup\{\epsilon\})$ 表示状态集；$Q_0=\{(x,\epsilon)\in X\times\{\epsilon\}:\exists x_0\in X_0\ \text{s.t.}\ x_0\overset{o}{\rightsquigarrow}x\}$ 表示初始状态集；$\Delta\subseteq Q\times Q$ 表示转换函数，对于两个任意状态 $q=(x,o)$，$q'=(x',o')\in Q$，当且仅当 $x\overset{o'}{\rightsquigarrow}x'\wedge o'\in\mathcal{O}$，都有 $\langle(x,o),(x',o')\rangle\in\Delta$；$\mathcal{AP}=X\cup\mathcal{O}$ 为原子命题集；$L:Q\to 2^{\mathcal{AP}}$ 为标记函数，对每一个 $q=(x,o)\in Q$，都有

$$L(q)=\begin{cases}\{x\} & q\in Q_0\\ \{x,o\} & q\notin Q_0\end{cases}\tag{10}$$

对于 $K_G^\Diamond$ 中的每一个状态 (x,o)，第 1 个分量 x 捕捉了 $G^\Diamond$ 中的状态，第 2 个分量 o 捕获了 $G^\Diamond$ 中最近发生的观测。在每一个状态 (x,o)，当一个使得 $M(s)=o'$ 发生的可行序列 $s\in\mathcal{L}(G^\Diamond,x)$ 出现时，Kripke 结构到达新的状态 $(\delta(x,s),o')$。其中，第 1 个分量由 $G^\Diamond$ 中的跃迁函数决定，第 2 个分量记录了观测符号，如图 3 中状态 (x_2,o_1) 表示系统在观测到 o_1 后，转换到 x_2 状态。由于最初没有观测到事件，因此初始状态的第 2 个分量记为 ϵ，如图 3 中初始状态 (x_0,ϵ) 与 (x_1,ϵ) 所示。标记函数将状态符号 x 和观测符号 o 指定为每个 (x,o) 的原子命题，如图 3 中状态 (x_0,ϵ) 与 (x_2,o_1) 的标记函数分别为 $\{x_0\}$ 与 $\{x_2,o_1\}$。因此，$K_G^\Diamond$

中的轨迹信息已经包含了状态序列信息和观测序列信息，足以验证 $G^{\Diamond}$ 的观测性质。为正式地看到 $G^{\Diamond}$ 和 $K_G^{\Diamond}$ 之间的联系，考虑 $G^{\Diamond}$ 中一个任意的无穷序列 $s=\omega_0\sigma_1\omega_1\sigma_2\omega_2\cdots\in\Sigma^{\omega}$，$s\in\mathcal{L}^{\omega}(G^{\Diamond},x_0)$，其中每个 $\omega_i\in\Sigma_{\mathrm{uo}}^{*}$ 是一个不可观测序列，每个 $\sigma_i\in\Sigma_{\mathrm{o}}$ 是一个可观测事件，其中 $M(\sigma_i)=o_i$。假设：

$$\underbrace{x_0^0\cdots x_0^{|\omega_0|}}_{\text{沿}\omega_0\text{访问}}\ \underbrace{x_1^0\cdots x_1^{|\omega_1|}}_{\text{沿}\sigma_1\omega_1\text{访问}}\ \underbrace{x_2^0\cdots x_2^{|\omega_2|}}_{\text{沿}\sigma_2\omega_2\text{访问}}\cdots\in X^{\omega} \tag{11}$$

是从 x_0 开始沿 s 访问的无穷状态序列，其中 $x_0^0=x_0$。注意，在 $K_G^{\Diamond}$ 的构造中，每次观测都将直接“跳转”到一个状态，其间不考虑不可观测序列所访问的状态。因此可知，对任意指标 $k_0,k_1,\cdots$，其中 $k_i\in\{0,\cdots,|\omega_i|\}$，$K_G^{\Diamond}$ 存在以下运行：

$$\rho=(x_0^{k_0},\epsilon)(x_1^{k_1},o_1)(x_2^{k_2},o_2)\cdots\in\mathrm{Run}(K_G^{\Diamond}) \tag{12}$$

称这样的运行与从初始状态 x_0 开始的序列 s 相兼容。由于指标 $k_0,k_1,\cdots$ 的选择不是唯一的，使用 $\mathrm{Run}(s,x_0)\subseteq\mathrm{Run}(K_G^{\Diamond})$ 作为所有与 s 和 x_0 相兼容运行的集合。

另一方面，对于任意运行

$$\rho=(x_0,\epsilon)(x_1,o_1)(x_2,o_2)\cdots\in\mathrm{Run}(K_G^{\Diamond}) \tag{13}$$

根据构造，有 $x_i\overset{o_{i+1}}{\rightsquigarrow}x_{i+1}$。因此，总能找到一个初始状态 $\hat{x}_0\in X_0$ 和一条无穷序列 $s\in\mathcal{L}^{\omega}(G^{\Diamond},\hat{x}_0)$，使 $M(s)=o_1o_2\cdots$，并且每个 x_i 均能通过 s 的一个观测为 $o_1o_2\cdots o_i$ 的前缀到达，即 $\rho\in\mathrm{Run}(s,\hat{x}_0)$。

图 3 为一个说明性示例，根据图 2 中系统 $G^{\Diamond}$ 构造出其 Kripke 结构 $K_G^{\Diamond}$。

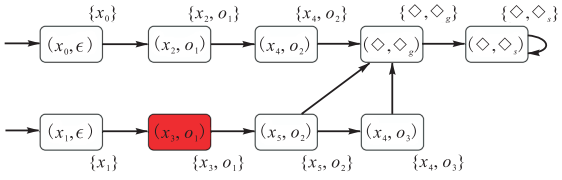

图 3　Kripke 结构 $K_G^{\Diamond}$

Fig. 3　Kripke structure $K_G^{\Diamond}$

2.2　改进的 Kripke 结构

在这一小节中，对如何构造改进的 Kripke 结构进行讨论。如果直接在 Kripke 结构 $K_G^{\Diamond}$ 中检测 K 步不透明性的 HyperLTL 公式，就会导致以下问题的出现。K 步不透明性需要确定任意序列上某一状态的机密性，然而 HyperLTL 的语义是定义在无穷轨迹上的。为在 HyperLTL 语义下能够刻画上述需求，需要制定相应机制对两条分别处于无穷轨迹中同一时刻的秘密和非秘密状态进行指示。为建立 K 步不透明性的有限要求与 HyperLTL 的无穷语义之间的联系，进一步改进 Kripke 结构 $K_G^{\Diamond}$，允许其进程在任意有限的时刻指示。

定义 4（改进的 Kripke 结构）：给定系统 $G^{\Diamond}$ 及其 Kripke 结构 $K_G^{\Diamond}=(Q,Q_0,\Delta,\mathcal{AP},L)$，改进的 Kripke 结构定义为

$$\tilde{K}_G^{\Diamond}=(\tilde{Q},\tilde{Q}_0,\tilde{\Delta},\widetilde{\mathcal{AP}},\tilde{L}) \tag{14}$$

其中：$\tilde{Q}=Q\cup Q^{\mathrm{c}}$ 为状态集，$Q^{\mathrm{c}}=\{(x^{\mathrm{c}},o^{\mathrm{c}}):(x,o)\in Q\}$ 是原始状态集 Q 的一个复制；$\tilde{Q}_0=Q_0$ 为初始状态集；$\tilde{\Delta}\subseteq\tilde{Q}\times\tilde{Q}$ 为转换函数，对于任意状态 $q,q'\in Q$：$\langle q,q'\rangle\in\tilde{\Delta}$，$\langle q,q'\rangle\in\tilde{\Delta}$ 成立；对于任意状态 $q=(x,o)\in Q$，有

$$\big\langle(x,o),(x^{\mathrm{c}},o^{\mathrm{c}})\big\rangle,\big\langle(x^{\mathrm{c}},o^{\mathrm{c}}),(x,o)\big\rangle\in\tilde{\Delta} \tag{15}$$

$\widetilde{\mathcal{AP}}=X\cup\mathcal{O}\cup\{\tau\}$ 为原子命题集，τ 是一个指示符号；$\tilde{L}:\tilde{Q}\to 2^{\widetilde{\mathcal{AP}}}$ 为标记函数，对于任意状态 $q\in Q$，$\tilde{L}(q)=L(q)$ 成立，对于任意复制状态 $q^{\mathrm{c}}=(x^{\mathrm{c}},o^{\mathrm{c}})\in Q^{\mathrm{c}}$，$\tilde{L}(q^{\mathrm{c}})=\{x,\tau\}$ 成立。

从 $K_G^{\Diamond}$ 构造 $\tilde{K}_G^{\Diamond}$，只需为 $K_G^{\Diamond}$ 中的每个状态 (x,o) 添加一个新的复制状态 $(x^{\mathrm{c}},o^{\mathrm{c}})$，如图 4 中状态 (x_2,o_1) 的复制状态为 $(x_2^{\mathrm{c}},o_1^{\mathrm{c}})$。除了 $K_G^{\Diamond}$ 中的原始转换之外，每个状态 (x,o) 及其复制状态 $(x^{\mathrm{c}},o^{\mathrm{c}})$ 形成一个循环，如图 4 中状态 (x_2,o_1) 及其复制状态 $(x_2^{\mathrm{c}},o_1^{\mathrm{c}})$ 所示。此外，对于每个复制状态，赋予它一个新的原子命题 τ。直观地说，τ 将被用作对特定时刻进行定位的指标，以检查秘密状态。

考虑 $K_G^{\Diamond}$ 中的一个任意运行 $\rho=(x_0,\epsilon)(x_1,o_1)\cdots\in\mathrm{Run}(K_G^{\Diamond})$。对于感兴趣的任意时刻 $k\geqslant 0$，基于 $\tilde{K}_G^{\Diamond}$ 结构检查当前的秘密状态，其存在以下运行：

$$\tilde{\rho}=(x_0,\epsilon)(x_1,o_1)\cdots\underbrace{(x_k,o_k)(x_k^{\mathrm{c}},o_k^{\mathrm{c}})(x_k,o_k)}_{\text{循环到第}k\text{ 时刻的复制状态}}(x_{k+1},o_{k+1})\cdots \tag{16}$$

其轨迹由 $\tilde{L}(\tilde{\rho})=\{x_0\}\{x_1,o_1\}\cdots\{x_k,o_k\}\{x_k,\tau\}\{x_k,o_k\}\cdots$ 给出。因此，借助原子命题 τ，可以容易地定位到 x_k，以检测是否 $x_k\in X_{\mathrm{S}}$ 并截断 τ 后的无穷序列。

注意，在上面的无穷轨迹中，只想在感兴趣的特定时刻到复制状态上循环一次。然而，时序运算符 $\Diamond\tau$ 不足以表达这一点，因为 τ 可能发生多次。为此，将运算符 $\Diamond_1$ 定义为“最终且仅一次”，即

$$\Diamond_1\tau\equiv\Diamond\tau\wedge\Box(\tau\to\bigcirc\Box\neg\tau) \tag{17}$$

τ 最终会发生，一旦发生，未来将不再发生。

图 4 为一个说明性示例，根据图 3 中 Kripke 结构 $K_G^{\Diamond}$ 构造出其改进的 Kripke 结构 $\tilde{K}_G^{\Diamond}$。

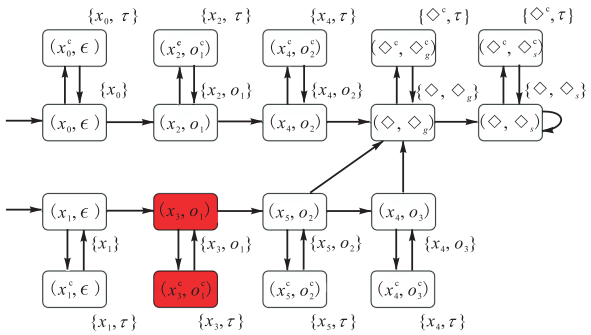

图 4　改进的 Kripke 结构 $K_G^{\Diamond}$

Fig. 4　Modified Kripke structure $K_G^{\Diamond}$

3　基于 HyperLTL 的 K 步不透明性

3.1　HyperLTL 语义下的 K 步不透明性

在这一小节中，讨论 HyperLTL 模型下的 K 步不透明性判据。本质上 K 步不透明性要求，对于系统中任意到达秘密状态的序列，均存在另一个与之观测等价的序列到达非秘密状态，并且对于任意从上述秘密状态开始的序列，均存在另一个从上述非秘密状态开始并与之观测等价的序列，以确保外部观测者无法推断出系统是否在 K 步之前曾经过秘密状态。此需求能够被基于改进的 Kripke 结构 $\tilde{K}_G^{\Diamond}$ 的 HyperLTL 公式捕捉。

备注 2　为简单起见，分别用 S^{π} 和 NS^{π} 表示轨迹处于秘密状态和非秘密状态的命题，即 $S^{\pi}\equiv \mathrm{V}_{x\in X_{\mathrm{S}}}x^{\pi}$，$NS^{\pi}\equiv \mathrm{V}_{x\in X_{\mathrm{NS}}}x^{\pi}$。用 $\square^{\leqslant K}a$ 表示轨迹上从第 0 个位置到第 K–1 个位置依次满足性质 a，即

$$\square^{\leqslant K}a\equiv \bigwedge_{i=0}^{K-1}a^{\pi[i]} \tag{18}$$

其中 $\pi^{[i]}$ 表示轨迹上的第 i 个位置。

定理 1　系统 G 是 K 步不透明的，当且仅当 $\tilde{K}_G^{\Diamond}\vDash\phi_{KSO}$，其中

$$\phi_{KSO}=\forall\pi_1\exists\pi_2.,\left[\begin{array}{c}\left[\Diamond_1\tau^{\pi_1}\wedge\square(\tau^{\pi_1}\rightarrow S^{\pi_1})\right]\rightarrow\\ [(o^{\pi_1}=o^{\pi_2})\mathcal{U}\tau^{\pi_1}\wedge\square(\tau^{\pi_1}\rightarrow(\tau^{\pi_2}\wedge NS^{\pi_2}\wedge\square^{\leqslant K+1}(o^{\pi_1}=o^{\pi_2})))]\end{array}\right]$$

证明　**充分性**：采用反证法，假设 $\tilde{K}_G^{\Diamond}\nvDash\phi_{KSO}$，即存在一个运行

$$\rho=(x_0,\epsilon)(x_1,o_1)\cdots(x_k,o_k)(x_k^{\mathrm{c}},o_k^{\mathrm{c}})(x_k,o_k)\cdots\in\mathrm{Run}(\tilde{K}_G^{\Diamond}) \tag{19}$$

使 $\tilde{L}((x_k^{\mathrm{c}},o_k^{\mathrm{c}}))=\{x_k,\tau\}$ 且 $x_k\in X_{\mathrm{S}}$，即 $\Diamond_1\tau^{\pi_1}\wedge\square(\tau^{\pi_1}\rightarrow S^{\pi_1})$ 适用于轨迹 $\pi=\tilde{L}(\rho)$。对任意的其他运行

$$\rho'=(x_0',\epsilon)(x_1',o_1)\cdots(x_k',o_k)(x_k'^{\mathrm{c}},o_k^{\mathrm{c}})(x_k',o_k)\cdots\in\mathrm{Run}(\tilde{K}_G^{\Diamond}) \tag{20}$$

即 $[(o^{\pi}=o^{\pi'})\mathcal{U}\tau^{\pi}]\wedge[\square(\tau^{\pi}\rightarrow\tau^{\pi'}\wedge(\square^{\leqslant K+1}(o^{\pi}=o^{\pi'})))]$ 适用于轨迹 $\pi'=\tilde{L}(\rho')$，且 $x_k'\in X_{\mathrm{S}}$。现在，对任意 $k\leqslant j\leqslant k+K$，定义 $\hat{q}_j^k$ 作为序列 $o_1\cdots o_j$ 中关于观测 $o_1\cdots o_k$ 的延迟状态估计，即

$$\hat{q}_j^k=\left\{\begin{array}{c}x_k'\in X:(x_0',\epsilon)(x_1',o_1')\cdots\in\mathrm{Run}(\tilde{K}_G^{\Diamond})\\ \text{s.t. } o_i'=o_i,\forall 1\leqslant i\leqslant j\end{array}\right\} \tag{21}$$

根据改进的 Kripke 结构 $\tilde{K}_G^{\Diamond}$ 的构造可知：$\hat{q}_j^k=\hat{X}_G(o_1\cdots o_k\,|o_1\cdots o_j)$，$\hat{q}_{j+1}^k\subseteq\hat{q}_j^k$，则存在 $\hat{q}_{k+K}^k\subseteq X_{\mathrm{S}}$，否则将存在一个运行 ρ' 使 $o_i'=o_i$，$\forall i\geqslant 1$ 而 $x_k'\notin X_{\mathrm{S}}$，进一步考虑观测 $\alpha=o_1\cdots o_k$ 及 $\beta=\alpha o_{k+1}\cdots o_{k+K}$，能够得到 $\hat{X}_G(\alpha|\beta)=\hat{q}_{k+K}^k\subseteq X_{\mathrm{S}}$，故根据定义 2 可知，$G$ 不是 K 步不透明的。

必要性：仍采用反证法，假设系统 G 不是 K 步不透明的，则表明存在观测序列 $\alpha=:o_1\cdots o_k$ 和 $\beta=:\alpha o_{k+1}\cdots o_{k+K}\in M(\mathcal{L}(G))$，使 $\hat{X}_G(\alpha|\beta)\subseteq X_{\mathrm{S}}$ 成立，基于 $\tilde{K}_G^{\Diamond}$ 的构造可知，存在一个运行

$$\rho=(x_0,\epsilon)(x_1,o_1)\cdots(x_k,o_k)(x_k^{\mathrm{c}},o_k^{\mathrm{c}})(x_k,o_k)\cdots\in\mathrm{Run}(\tilde{K}_G^{\Diamond}) \tag{22}$$

其中 $\tilde{L}((x_k^{\mathrm{c}},o_k^{\mathrm{c}}))=\{x_k,\tau\}$ 且 $x_k\in X_{\mathrm{S}}$，即 $\Diamond_1\tau^{\pi_1}\wedge\square(\tau^{\pi_1}\rightarrow S^{\pi_1})$ 适用于轨迹 $\pi=\tilde{L}(\rho)$。进一步有 $\hat{q}_{k+K}^k=\hat{X}_G(\alpha|\beta)\subseteq X_{\mathrm{S}}$，其中 $\hat{q}_{k+K}^k$ 由式(6)定义。称公式

$$\forall\pi'\in\mathrm{Trace}(\tilde{K}_G^{\Diamond})\cdot\left[\begin{array}{c}[(o^{\pi}=o^{\pi'})\mathcal{U}\tau^{\pi}\wedge\square(\tau^{\pi}\rightarrow\tau^{\pi'})]\\ \rightarrow[\square(\tau^{\pi}\rightarrow(\tau^{\pi'}\wedge S^{\pi'}\wedge\square^{\leqslant k+1}(o^{\pi}=o^{\pi'})))]\end{array}\right] \tag{23}$$

成立。否则，将存在一个运行 $\rho'=(x_0',\epsilon)(x_1',o_1)\cdots(x_k',o_k)(x_k'^{\mathrm{c}},o_k^{\mathrm{c}})(x_k',o_k)\cdots\in\mathrm{Run}(\tilde{K}_G^{\Diamond})$，其中 $x_k'\in X_{\mathrm{NS}}$。则对任意 $k\leqslant j\leqslant k+K$ 有 $\{x_k,x_k'\}\subseteq\hat{q}_j^k\not\subseteq X_{\mathrm{S}}$，这与 $\hat{q}_j^k\subseteq X_{\mathrm{S}}$，$k\leqslant j\leqslant k+K$ 相矛盾。因此式(7)成立，即 $\tilde{K}_G^{\Diamond}\nvDash\phi_{KSO}$，证毕。

3.2　HyperLTL 语义下 K 步不透明性的验证

在这一小节中，讨论 K 步不透明性的验证问题。使用以下示例说明 K 步不透明性的验证。

例 2　考虑图 1 所示系统 $G=(X,\Sigma,\delta,X_0)$，经转换后的系统 $G^{\Diamond}$ 如图 2 所示，其中 $X=\{x_0,x_1,x_2,x_3,x_4,x_5,\Diamond\}$，$X_{\mathrm{S}}=\{x_3\}$，$\Sigma_{\mathrm{o}}=\{a,b,c,d,e,\Diamond_g,\Diamond_s\}$，$\Sigma_{\mathrm{uo}}=\{u\}$，观测掩码 M：$M(a)=M(b)=o_1$，$M(c)=M(d)=o_2$，$M(e)=o_3$，$M(u)=\epsilon$，$M(\Diamond_g)=\Diamond_g$，$M(\Diamond_s)=\Diamond_s$，$\mathcal{O}=\{o_1,o_2,o_3,\Diamond_g,\Diamond_s\}$。改进的 Kripke 结构 $\tilde{K}_G^{\Diamond}$ 如图 4 所示。考虑如下轨迹：

$$\pi_1=\{x_1\}\{x_3,o_1\}\{x_3,\tau\}\{x_3,o_1\}\{x_5,o_2\}\{x_4,o_3\}\{\Diamond,\Diamond_g\}(\{\Diamond,\Diamond_s\})^{\omega}\in\mathrm{Trace}(\tilde{K}_G^{\Diamond}) \tag{24}$$

对于 π_1，有 $\Diamond_1\tau^{\pi_1} \wedge \Box(\tau^{\pi_1} \to S^{\pi_1})$，其中 x_3 是一个秘密状态。存在轨迹 π_2 为

$$\begin{aligned}\pi_2 = &\{x_0\}\{x_2,o_1\}\{x_2,\tau\}\{x_2,o_1\}\{x_4,o_2\}\\ &\{\Diamond,\Diamond_g\}(\{\Diamond,\Diamond_s\})^{\omega} \in \mathrm{Trace}(\tilde{K}_G^{\Diamond})\end{aligned} \tag{25}$$

满足

$$(o^{\pi_1} = o^{\pi_2})\mathcal{U}\tau^{\pi_1} \wedge \Box(\tau^{\pi_1} \to (\tau^{\pi_2} \wedge NS^{\pi_2} \wedge \Box^{\leqslant K+1}(o^{\pi_1} = o^{\pi_2}))) \tag{26}$$

因此，有 $\tilde{K}_G^{\Diamond} \vDash \phi_{KSO}$，根据定理 1，系统 G 是 K 步不透明的，与例 1 结论一致。

4　结　语

本文在 HyperLTL 框架下研究了 DES 的 K 步不透明性验证问题。在原系统 G 的基础上，通过添加 Sink 状态，构造系统 $G^{\Diamond}$ 释放了对原系统的限制性条件，根据转换后得到的系统 $G^{\Diamond}$ 构造改进的 Kripke 结构 $\tilde{K}_G^{\Diamond}$，并给出 HyperLTL 语义下 K 步不透明性判据，进而对原系统的 K 步不透明性进行验证。未来会尝试推广本文的验证方法讨论强 K 步不透明性验证问题。

参考文献：

[1] CLARKSON M R，FINKBEINER B，KOLEINI M，et al. Temporal logics for hyperproperties[M]//ABADI M，KREMER S. Principles of security and trust. POST 2014. lecture notes in computer science：vol 8414. Heidelberg：Springer，2014：265–284.

[2] LAFORTUNE S，LIN F，HADJICOSTIS C N. On the history of diagnosability and opacity in discrete event systems[J]. Annual reviews in control，2018，45：257–266.

[3] MAZARÉ L. Using unification for opacity properties[R/OL]. (2004–10–28)[2024–01–25]. http://www.eva.imag.fr/bib/EVA-TR16.pdf.

[4] SABOORI A，HADJICOSTIS C N. Notions of security and opacity in discrete event systems[C]//IEEE. Proceedings of 46th IEEE Conference on Decision and Control. New Orleans：IEEE，2007：5056–5061.

[5] SABOORI A，HADJICOSTIS C N. Verification of initial-state opacity in security applications of discrete event systems[J]. Information sciences，2013，246：115–132.

[6] SABOORI A，HADJICOSTIS C N. Verification of infinite-step opacity and complexity considerations[J]. IEEE Transactions on automatic control，2011，57(5)：1265–1269.

[7] SABOORI A，HADJICOSTIS C N. Verification of K-step opacity and analysis of its complexity[J]. IEEE Transactions on automation science and engineering，2011，8(3)：549–559.

[8] BRYANS J W，KOUTNY M，MAZARÉ L，et al. Opacity generalised to transition systems[J]. International journal of information security，2008，7(6)：421–435.

[9] LIN F. Opacity of discrete event systems and its applications[J]. Automatica，2011，47(3)：496–503.

[10] FALCONE Y，MARCHAND H. Enforcement and validation(at runtime)of various notions of opacity[J]. Discrete event dynamic systems，2015，25：531–570.

[11] MA Z Y，YIN X，LI Z W. Verification and enforcement of strong infinite-and k-step opacity using state recognizers[J]. Automatica，2021，133：109838.

[12] HAN X G，ZHANG K Z，ZHANG J H，et al. Strong current-state and initial-state opacity of discrete-event systems[J]. Automatica，2023，148：110756.

[13] HAN X G，ZHANG K Z，LI Z W. Verification of strong K-step opacity for discrete-event systems[C]//IEEE. Proceedings of 2022 IEEE 61st Conference on Decision and Control(CDC). Cancun：IEEE，2022：4250–4255.

[14] 魏佳唯，赵骥，陈晓艳，等. 标准当前状态不透明性与强当前状态不透明性的归约[J]. 天津科技大学学报，2023，38(2)：70–74.

[15] ZHAO J N，LI S Y，YIN X. A unified framework for verification of observational properties for partially-observed discrete-event systems[J]. IEEE Transactions on automatic control，2024，69(7)：4710–4717.

[16] ZHANG K Z. A unified concurrent-composition method to state/event inference and concealment in labeled finite-state automata as discrete-event systems[J]. Annual reviews in control，2023，56：100902.

[17] YIN X. Estimation and verification of partially-observed discrete-event systems[EB/OL]. (2019–03–27)[2024–01–28]. https://arxiv.org/pdf/1903.11413v1.pdf.

[18] BAIER C，KATOEN J P. Principles of model checking[M]. Cambridge：MIT Press，2008.

责任编辑：周建军