基于隐蔽性裕度的离散事件系统监控

王　飞[1], 戴茵茵[1†], 金福江[2]

(1. 华侨大学 信息科学与工程学院, 福建 厦门 361021; 2. 华侨大学 机电及自动化学院, 福建 厦门 361021)

摘要: 本文研究了离散事件系统基于隐蔽性裕度的k–隐蔽性验证及监控综合问题. 首先, 文章分别给出基于语言和基于(估计)状态的隐蔽性裕度的概念, 通过提出两者之间的等价关系, 给出基于状态的隐蔽性裕度与基于语言的隐蔽性裕度等价的性质. 之后, 提出系统满足k–隐蔽性的充分性条件可以通过状态的隐蔽性裕度获得. 并以此条件, 分别给出计算状态的隐蔽性裕度以及验证k–隐蔽性的算法. 再后, 如果系统不满足k–隐蔽性时, 又给出一种算法来获取监控器使闭环系统是k–隐蔽的, 并且给出定理说明获得的监控器不仅可以保持受控系统的k–隐蔽性, 而且表明其是最大允许的. 最后, 利用实例说明验证系统的k–隐蔽性以及获取保持k–隐蔽性的最大允许监控器构造方法的有效性.

关键词: 隐蔽性裕度; k–隐蔽的; 监控器; 离散事件系统

引用格式: 王飞, 戴茵茵, 金福江. 基于隐蔽性裕度的离散事件系统监控. 控制理论与应用, 2025, 42(3): 618 – 626

DOI: 10.7641/CTA.2023.30186

Supervisory control on opacity-margin of discrete event systems

WANG Fei[1], DAI Yin-yin[1†], JIN Fu-jiang[2]

(1. College of Information Science and Engineering, Huaqiao University, Xiamen Fujian 361021, China;
2. College of Mechanical Engineering and Automation, Huaqiao University, Xiamen Fujian 361021, China)

Abstract: The synthesis problem of supervisory control and verification of k-opacity on opacity-margin is formulated. Firstly, two definitions of opacity-margin based on languages and based on (estimated) states are given respectively. By presenting the equivalent relations about the two definitions, we show that opacity-margin on states is equivalent to opacity-margin on languages. Secondly, a sufficient condition that secret is k-opaque with respect to the plant can be obtained by counting of opacity-margin on states. Based on the sufficient condition, algorithms to get opacity-margin and verify k-opacity are given. Thirdly, if the plant is not k-opaque, an algorithm is given to design a supervisor to assure the k-opacity. And, a conclusion is shown that the supervisor obtained by the algorithm is not only to preserve the k-opacity, but also to be maximal permissive. Finally, an illustration example is formulated to show the validity of the k-opacity's verification and method of achieving maximal permissive supervisor to keep the k-opacity.

Key words: opacity-margin; k-opaque; supervisors; discrete event systems

Citation: WANF Fei, DAI Yinyin, JIN Fujiang. Supervisory control on opacity-margin of discrete event systems. *Control Theory & Applications*, 2025, 42(3): 618 – 626

1　引言

离散事件系统(discrete event systems, DES)[1–2]是20世纪80年代兴起的用来研究事件驱动的人造系统的一门科学. 自1987年, Ramadge与Wonham[3]提出了监控理论(supervisory control theory, SCT), 之后, SCT发展成为了控制理论的一门新兴分支, 并在之后的40年间, 在制造系统、通信系统以及军事系统等方面得到了越来越多的应用. 近些年, 随着智能网络的发展, 各种在线服务提供了大量的信息, 其中有些信息不适合暴漏到外界, 如企业或个人数据库中需保护的商业机密或个人隐私等, 这些信息的安全性逐渐引起了DES领域的关注. 在信息安全方面, 研究者更多关注系统本身的隐蔽性、隐蔽性如何验证(验证问题)以及保证系统隐蔽性的方法(综合问题)等.

2004年, 文献[4]在计算机科学中为了分析加密协议, 首次提出隐蔽性的概念. 2005年, 文献[5]在Petri网模型中引入了隐蔽性来保证信息的安全, 这也是隐蔽性的概念首次出现在DES领域. 而在自动机模型中,

收稿日期: 2023−04−04; 录用日期: 2023−12−06.

†通信作者. E-mail: crystle@hqu.edu.cn; Tel.: +86 13459227556.

本文责任编委: 胡跃明.

国家自然科学基金项目(61203040), 福建省自然科学基金项目(2022J01295, 2023J05045), 泉州市科协项目资助.

Supported by the the National Natural Science Foundation of China (61203040), the National Natural Science Foundation of Fujian Province (2022J01295, 2023J05045) and the Quanzhou Association for Science and Technology.

文献[6]基于语言给出了隐蔽性的概念, 并区分了强隐蔽、弱隐蔽和无隐蔽等, 这种定义比文献[5]的定义更通用; 文献[7–9]又基于状态定义了3种不同的定义, 即初始状态(initial state)隐蔽[7–8]、k–步(k-step)隐蔽[9]和无限步(infinite step)隐蔽[8]等; 而文献[10]也整合了4种概念, 即基于语言的隐蔽性、初始状态隐蔽性、当前状态(current-state)隐蔽性和初始–结束状态(initial-and-final-state)隐蔽性等, 并提出不同的定义在相互转换时, 具有多项式的计算复杂度, 之后, 这些概念又进一步被推广到逼近隐蔽性[11]、k–步强隐蔽性[12]、当前状态强隐蔽[13]和初始状态强隐蔽[13]等. 基于这些概念, 近些年验证和综合问题的相关研究也丰富了很多. 文献[14]提出动态信息释放模型, 给出了新的当前状态隐蔽性概念以及观测器的结构, 并提出了验证该隐蔽性的一种有效算法; 文献[15]利用粗糙集作为知识提取工具给出了一种可用来验证系统基于语言的(强或弱)隐蔽性的结论与相应的算法; 文献[16]基于代数状态空间方法, 对于部分可观的DES提出验证和综合当前状态隐蔽性的方法; 文献[17]使用双向观测器提出了验证无限步与k–步的隐蔽性的方法; 文献[18]对于部分可观的系统提出了识别器, 且基于该识别器给出了无限步强隐蔽性与k–步强隐蔽性的验证方法, 并在系统不满足隐蔽性的条件下, 基于SCT给出了监控综合问题的解; 文献[19]基于精简的状态估计(condensed state estimates)的方法, 采用不动点理论在每一步的迭代中验证系统的隐蔽性, 并获得保证隐蔽性的最大允许监控器; 文献[20]利用子观测器方法, 用系统与其观测器(observers, obs)的同步积代替文献[19]的精简状态估计, 迭代获得保证隐蔽性的最大允许监控器; 文献[21]利用插入函数混淆秘密的方法, 迫使系统满足无限步隐蔽性与k–步隐蔽性; 文献[22]则利用插入虚拟事件来改变对手观测函数的方式迫使系统保持隐蔽性, 这种插入事件的方法也被应用到文献[23–24]中, 其中, 文献[23]在插入函数中考虑了一些约束条件(如能量限制), 文献[24]在插入事件时也考虑了删除事件的情况, 并且文献[23–24]都提出了相应的迫使系统保持隐蔽性的方法; 文献[25]考虑了具有动态可观函数的系统的无限步隐蔽性的综合问题(dynamic mask). 除此之外, 隐蔽性的验证与综合问题也被推广到了随机系统[26]、模系统[27]、不确定系统[28]和网络系统[29–30]等. 本文推广了文献[6]提出的隐蔽性概念, 通过计算隐蔽性裕度提出验证k–隐蔽性的方法, 并基于文献[20]的方法解决了k–隐蔽性的综合问题.

本文基于控制理论中稳定裕度的思想, 提出了基于语言的k–隐蔽的定义, 该定义是隐蔽性在隐蔽能力方面的推广. 由于基于语言验证隐蔽性的过程中存储能力的限制, 本文又讨论了基于语言的隐蔽性裕度和基于状态的隐蔽性裕度的关系, 提出通过计算有限状态的隐蔽性裕度是可以验证系统的k–隐蔽性的, 并基于SCT提出可以设计监控器来保证受控系统的k–隐蔽性. 在日常的生活中, k–隐蔽性的概念也是常见的, 因为在保护某些秘密信息时, 可能不仅仅关注秘密本身, 有时可能更关注保护的程度. 如在现代战争中, 己方目标为了迷惑对方雷达, 常需要一些伪装目标来保护真实目标, 则伪装的数目可以用来表示保护真实目标的程度, 如果伪装较少, 虽然能起到保护真实目标的意图, 使得对方雷达不能辨别哪个是真的, 但此时的隐蔽性较弱, 对方可能通过试错的手段, 对所有目标进行攻击, 此时真实目标被攻击到的概率较大; 反之, 则隐蔽性较强, 试错成本增加, 真实目标被攻击的概率则会大幅降低. 再如在室内使用手机, 为了避免定位技术造成个人位置的泄露, 对个人与定位服务商造成人身安全与经济上的重要影响, 在利用匿名器向定位服务器发送定位请求信息时, 添加哑元信息一起发送给定位服务器, 使定位服务商无法区分真实位置信息与哑元信息, 添加的哑元信息越多, 则真实位置信息被保护的程度就越强, 个人位置泄露的概率越小.

本文提出的基于状态的隐蔽性裕度的概念与文献[31]中的隐蔽性的度的概念相比, 放松了权函数的单调性的限制, 利用估计状态中可以混淆秘密的非秘密状态的数量定义了隐蔽性裕度, 该定义可以直观地描述混淆秘密之外的非秘密的数量, 也可以用来表示秘密被泄漏前允许禁止发生的非秘密串的数量. 本文提出的验证k–隐蔽性和设计监控器的方法, 是一般的隐蔽性验证和综合问题的推广.

2　预备知识

2.1　离散事件系统的监控理论

给定DES, 其模型为4–元组的自动机$G=(Q, \Sigma, \delta, q_0)$, 其中: Q为有限状态集, Σ为有限事件集, $\delta: Q\times\Sigma\to Q$为状态转移函数, q_0为初始状态. 记$L(G)=\{s\in\Sigma^*|\delta(q_0,s)!\}$为$G$的生成语言. 为了限制系统的行为, 事件集被分为可控事件集$\Sigma_{\rm c}$与不可控事件集$\Sigma_{\rm u}$, 且$\Sigma=\Sigma_{\rm c}\dot\cup\Sigma_{\rm u}$. 记$\Gamma=\{\gamma|\Sigma_{\rm u}\subseteq\gamma\subseteq\Sigma\}$为控制模式集. 称映射$f: L(G)\to\Gamma$为系统$G$的监控器, 在$f$的控制下, 闭环系统$L(f/G)$可按如下递推方式获得:

1) $\varepsilon\in L(f/G)$;

2) $s\sigma\in L(f/G)\Leftrightarrow s\in L(f/G), s\sigma\in L(G), \sigma\in f(s)$.

如果4–元组自动机G包含不可观事件, 则在G中, 用空串ε替代不可观事件, 此时, 确定自动机可变为非确定自动机, 进而, 利用文献[32], 将非确定自动机转为确定自动机, 即可得系统G的观测器$\mathrm{obs}(G)=(Q_{\rm a}, \Sigma_{\rm a}, \delta_{\rm a}, q_{0\rm a})$, 具体算法见文献[1].

对于两个自动机, 引入如下概念表示自动机的同步积运算[1].

定义 1 给定两个自动机$G_1=(Q_1,\Sigma_1,\delta_1,q_{01})$和$G_2=(Q_2,\Sigma_2,\delta_2,q_{02})$, 称$G_1\|G_2=Ac(Q_1\times Q_2,\Sigma_1\cup\Sigma_2,\delta_{12},(q_{01},q_{02}))$为同步积, 其中: $Ac(\cdot)$为·的可达运算,

$$\delta_{12}((q_1,q_2),\sigma)=\begin{cases}(\delta_1(q_1,\sigma),\delta_2(q_2,\sigma)), & \sigma\in\Sigma_1\cap\Sigma_2;\\ (\delta_1(q_1,\sigma),q_2), & \sigma\in\Sigma_1\backslash\Sigma_2;\\ (q_1,\delta_2(q_2,\sigma)), & \sigma\in\Sigma_2\backslash\Sigma_1;\\ \text{无定义}, & \text{其他}.\end{cases}$$

2.2 离散事件系统的隐蔽性

假设对手完全了解系统的结构, 但对手仅能观测到系统的部分事件集, 记为$\Sigma_{\rm a}$. 设$\theta:\Sigma^*\to\Sigma_{\rm a}^*$为对手的观测函数, 对手可通过$\theta$了解系统的信息, 并记其视野为$G$关于$\theta$的观测器, 即$\mathrm{obs}(G)$. 对手根据观测到的信息, 构筑估计行为. 如果对手不能识别秘密信息, 则称该秘密是隐蔽的, 具体定义[6,19]如下.

定义 2 给定非空语言K, 对于任意的$s\in K\cap L(G)$, 如果存在$s'\in L(G)\backslash K$使得$\theta(s)=\theta(s')$成立, 则称K关于$L(G)$与$\Sigma_{\rm a}$是(强)隐蔽的(opaque), 简称K是(强)隐蔽的.

从对手的角度, 隐蔽性意味着对手无法分辨秘密信息与非秘密的信息, 即对手通过传感器"看到"的内容是一样的. 为了方便, 称被对手"看到"一样的串是等价的, 即s与s'是等价的当且仅当$\theta(s)=\theta(s')$. 如果不满足上述定义的秘密称其是非隐蔽的.

由隐蔽性对于并运算的封闭性知, 系统中存在能保持隐蔽性的最大的可控闭的子语言[6,19].

2.3 离散事件系统隐蔽性的度

为了推广隐蔽性的定义, 文献[31]通过权函数, 利用估计状态的个数定义了状态的隐蔽性的度(degree of opacity), 该定义描述了对手能"看到"的所有"相同"状态的最小个数, 具体内容如下.

定义 3 给定系统G, 称单调递增函数$\omega:2^Q\to\mathbb{R}^+$为隐蔽性的权函数. 特例: 给定状态集$A$, 称$\omega(A)=|A|$为计数函数.

显然, 在上述定义中, 集合中元素越多, 则权值越大. 利用上述计数函数的定义, 文献[31]给出了基于状态的隐蔽性的度和系统的隐蔽性的度的概念.

定义 4 给定系统G和其观测器$\mathrm{obs}(G)$, 称$\Theta(q)=\min\limits_{\substack{E(q)\in Q_{\rm a}\\ q\in Q}}\omega(E(q))$为状态$q$的隐蔽性的度.

定义 5 给定系统G和其观测器$\mathrm{obs}(G)$, 称$\Theta(G)=\min\limits_{q\in Q}\Theta(q)$为系统$G$的隐蔽性的度.

由定义4和定义5知, 度的定义是状态集的绝对计数, 其需结合每个估计状态以及秘密状态集才可以验证与综合系统的隐蔽性[31].

3 k–隐蔽性的验证方法

给定系统$G=(Q,\Sigma,\delta,q_0)$, G的观测器为$\mathrm{obs}(G)=(Q_{\rm a},\Sigma_{\rm a},\delta_{\rm a},q_{0{\rm a}})$. 在系统$G$中, 设秘密$K$是正则语言, 且$\varepsilon\notin K$, 记$K$可以被自动机$G$中的状态集$Q_{\rm s}\subseteq Q$辨识, 即$K=L_{\rm s}(G)=\{s|\delta(q_0,s)\in Q_{\rm s}\}$. 为了研究系统的$k$–隐蔽性, 本文做如下假设:

假设 1 系统G中不存在对手看不到的非秘密循环串, 即$\forall s=s_1s_2s_3\in L(G)$, 且$s_1s_2\in L(G)\backslash K$, 如果$\forall m\geqslant 0, s_1s_2^ms_3\in L(G)$, 则$s_2\notin(\Sigma\backslash\Sigma_{\rm a})^*$, 其中$s_2\neq\varepsilon$.

上述假设1说明, 对手看到相同的非秘密串一定不在同一个循环内.

在假设条件下, 利用定义2, 给出如下定义来表示系统的隐蔽性程度:

定义 6 对于任意的$s\in K\cap L(G)$, 如果至少存在k个不同的$s'\in L(G)\backslash K$使得$\theta(s)=\theta(s')$成立, 即$\forall s\in K\cap L(G),\exists s_1,s_2,\cdots,s_{k'}\in L(G)\backslash K$, 其中$k'\geqslant k$, 对于任意的$i\in\{1,2,\cdots,k'\}$均有$\theta(s)=\theta(s_i)$, 则称秘密$K$关于$L(G)$和$\Sigma_{\rm a}$是$k$–隐蔽的, 简称秘密$K$是$k$–隐蔽的($k$-opaque).

对于秘密事件串s, k–隐蔽的意味着至少存在着k个可以混淆s的非秘密事件串. 特殊地, 如果$k=1$, 则1–隐蔽的就是隐蔽的(见定义2).

根据语言的k–隐蔽性定义, 易得如下性质.

定理 1 如果K关于$L_i\subseteq L(G)$和$\Sigma_{\rm a}$是k–隐蔽的, 则K关于$\bigcup\limits_{i=1}^{n}L_i$和$\Sigma_{\rm a}$是$k$–隐蔽的, 其中$i=1,2,\cdots,n$.

证 $\forall s\in\bigcup\limits_{i=1}^{n}L_i\Rightarrow\exists i\in\{i=1,2,\cdots,n\}$, 使得$s\in L_i\Rightarrow$总存在至少$k$个$s'\in L_i\backslash K$, 使得$\theta(s)=\theta(s')\Rightarrow$至少$k$个$s'\in\bigcup\limits_{i=1}^{n}L_i\backslash K$, 使得$\theta(s)=\theta(s')\Rightarrow K$关于$\bigcup\limits_{i=1}^{n}L_i$和$\Sigma_{\rm a}$是$k$–隐蔽. 证毕.

由定理1中k–隐蔽性关于并运算的封闭性可知, 系统中存在能保持k–隐蔽的最大的子语言. 再由闭的可控语言对并运算的封闭性[1]可知, 系统中存在能保持k–隐蔽的最大的可控闭的子语言.

为了描述混淆秘密串的非秘密串的数目, 本文先基于事件串, 给出如下概念表示系统的隐蔽性程度:

定义 7 对于串$s\in L(G)$, 称$o_G(s)$为串s的隐蔽性裕度, 其中映射$o_G:L(G)\to N$为权函数, 其值

可按如下方式获得:

1) 如果$s \in K$, 记$o_G(s) = |[s]_\theta^{\neg K}|$, 其中: $|\cdot|$表示$\cdot$的基数, $[s]_\theta^{\neg K} = \{s' \in L(G)\backslash K | \theta(s) = \theta(s')\}$;

2) 如果$s \in L(G)\backslash K$, 且存在秘密串$s' \in K$使得$\theta(s) = \theta(s')$, 记$o_G(s) = o_G(s')$;

3) 如果$s \in L(G)\backslash K$, 若对于任意的秘密串$s' \in K$使得$\theta(s) \neq \theta(s')$成立, 记$o_G(s) = +\infty$.

在定义7中, 串的隐蔽性裕度表示某秘密信息在被泄漏之前可以禁止用来混淆该秘密串的所有非秘密串的数目. 显然, 秘密串的隐蔽性裕度越大, 该串越隐蔽, 泄漏的机会越小.

由上述定义7, 可做如下说明:

注 1　设$o_G(s) = k$, 记与s等价的秘密串为s'. 如果$0 < k < +\infty$, 表示只要禁止[1]可以混淆s'的非秘密串的数量小于k, 则秘密s'就不会被泄露; 如果$k = 0$, 则s就是一个秘密串, 且系统中不存在可以混淆秘密串s的非秘密串, 即该秘密串s以及所有与s等价的秘密串都会被泄漏; 如果$k = +\infty$, 表示s是非秘密串, 且所有与s等价的串都是非秘密串, 这些非秘密串不能混淆任意的秘密串, 如果这些非秘密串都被禁止了, 系统的隐蔽性也不会受影响.

由定义7以及假设易得如下定理:

定理 2　如果某一秘密串$s \in K$, 则对于任意的等价串$s' \in L(G)$均有$o_G(s') < +\infty$, 即秘密串以及与其等价的串的隐蔽性裕度是有限的.

综合定义7中的3种情况, 串的隐蔽性裕度可以推广到如下概念:

定义 8　称所有串$s \in L$的隐蔽性裕度中的最小值为语言L在系统G的隐蔽性裕度, 记为$o_G(L) = \min\{o_G(s) | s \in L\}$.

由定义7、定义8以及$L(G) = (L(G)\backslash K) \cup K$可知, $o_G(L(G)) = o_G(K)$成立.

定义 9　称$o_G(K)$或$o_G(L(G))$为系统G的隐蔽性裕度.

上述定义的系统隐蔽性裕度表示系统的秘密信息在被泄漏之前可以禁止用来混淆秘密的非秘密串的数目. 显然, 系统的隐蔽性裕度越大, 系统的秘密越隐蔽, 泄漏的机会越小.

由定义6与定义9可知, k–隐蔽性和系统的隐蔽性裕度之间存在着如下的等价关系:

定理 3　给定系统G、秘密K和容许度k, 则K是k–隐蔽的, 当且仅当系统G的隐蔽性裕度$o_G(K) \geqslant k$.

证　K是k–隐蔽的$\Leftrightarrow (\forall s \in K)$, 则至少存在$k$个不同$s' \in L(G)\backslash K$, 使得 $\theta(s) = \theta(s')$ 成立$\Leftrightarrow |\{s' \in L(G)\backslash K | \theta(s) = \theta(s'), s \in K\}| \geqslant k \Leftrightarrow (\forall s \in K) \times o_G(s) \geqslant k \Leftrightarrow o_G(K) \geqslant k$.　证毕.

再由定理3和定义9, 可得如下推论:

推论 1　给定系统G、秘密K和容许度k, K是k–隐蔽的, 当且仅当$(\forall s \in L(G))o_G(s) \geqslant k$成立.

特殊地, 当$k = 1$时, 定理3可简化如下:

推论 2　给定系统G、秘密K和容许度k, K是隐蔽的, 当且仅当系统G的隐蔽性裕度$o_G(K) \geqslant 1$.

由定理3知, 欲说明秘密K是隐蔽的, 则需计算$o_G(K)$. 如果系统中存在Kleen闭包时, 语言K与$L(G)\backslash K$的存储量会很大, 不方便数据存储. 基于有限状态自动机, 本文又定义了基于状态的隐蔽性裕度, 通过研究基于串和基于状态的隐蔽性裕度之间的关系, 计算有限状态的隐蔽性裕度不仅方便存储数据, 而且可以判别k–隐蔽性, 具体定义和相关结论如下:

定义 10　给定系统G的观测器$\mathrm{obs}(G)$, 任取$E \in Q_{\mathrm{a}}$, 则在G中, 总存在$q \in Q$使得$E(q) = E$, 其中$E(q)$是状态q的一个估计状态. 称$M(E)$(或$M(E(q))$为E的隐蔽性裕度, 其中映射$M : Q_{\mathrm{a}} \to N$为权函数, 其值可按如下方式获得:

1) 如果$E \cap Q_{\mathrm{s}} \neq \varnothing$, 记$M(E) = |E\backslash Q_{\mathrm{s}}|$;

2) 如果$E \cap Q_{\mathrm{s}} = \varnothing$, 记$M(E) = +\infty$.

从对手的角度来看, 如果原系统中到达不同状态的事件序列被对手"看到"的是相同的, 则在观测器中这些状态被合并为一个状态, 即估计状态. 隐蔽性裕度$M(E(q))$表示在状态q处, 估计状态$E(q)$的隐蔽性裕度, 表示秘密状态被泄露之前, 允许删除用来混淆该秘密状态的所有非秘密状态的数目.

对于定义10, 可做如下说明:

注 2　设$M(E) = k$. 如果$0 < k < +\infty$, 表示在E中删除非秘密状态的个数小于k, 则E中的秘密状态(或进入该秘密状态的秘密串)就不会被泄露; 如果$k = 0$, 表示E中只有秘密状态, 无非秘密状态, E中的所有秘密都会被泄漏; 如果$k = +\infty$, 表示E中只有非秘密状态, 这些非秘密状态完全被删除也不会影响系统的隐蔽性.

假设 2　不在同一个循环内的用来混淆密串两个等价的非秘密串会转移到不同的状态.

假设2成立说明对手看到用来混淆秘密信息的等价的非秘密串, 转移到的状态不同.

由定义7和定义10可知, 在假设2成立下, 基于串与基于状态的隐蔽性裕度是等价的, 具体定理如下:

定理 4　给定系统G、观测器$\mathrm{obs}(G)$和容许度k, 则对于任意的$q \in Q_{\mathrm{s}}$, 均有$M(E(q)) \geqslant k$成立, 当且仅当对于任意的$s \in K$, 均有$o_G(s) \geqslant k$成立, 即$o_G(K) \geqslant k$成立.

[1]禁止串是指禁止串的发生, 即在该串中寻找到某一可控事件, 禁止该可控事件的发生就可以禁止该串的发生.

证 1) 充分性.

$\forall q \in Q_{\rm s} \Rightarrow \exists s \in K$ s.t. $q=\delta(q_0,s) \Rightarrow o_G(s) \geqslant k \Rightarrow |\{s' \in L(G)\backslash K | \theta(s)=\theta(s'), s \in K\}| \geqslant k \Rightarrow \forall s_i \in \{s' \in L(G)\backslash K | \theta(s)=\theta(s'), s \in K\}$, 记$q_i = \delta(q_0,s_i) \in Q\backslash Q_{\rm s}$. 若$s_i \neq s_j, i \neq j$, 则$q_i \neq q_j$, 其中$i = 1,2,\cdots,k,\cdots. \Rightarrow q_i \in E(q) = \delta_{\rm a}(q_{0{\rm a}},\theta(s_i)) \Rightarrow M(E(q)) = |E(q)\backslash Q_{\rm s}| = |\{q_1,q_2,\cdots,q_k,\cdots\}| \geqslant k$.

2) 必要性.

$\forall s \in K \Rightarrow \exists q \in Q_{\rm s},\ q=\delta(q_0,s) \Rightarrow E(q) = \delta_{\rm a}(q_{0{\rm a}},\theta(s)) \Rightarrow E(q) \cap Q_{\rm s} \neq \varnothing \Rightarrow E(q)$中至少包含$k$个相异的非秘密状态$\Rightarrow E(q)\backslash Q_{\rm s} \neq \varnothing \Rightarrow (\forall q' \in E(q)\backslash Q_{\rm s}) \exists s' \in L(G)\backslash K$ s.t. $\delta(q_0,s')=q', \theta(s)=\theta(s') \Rightarrow$ (至少有k个相异的$q' \in E(q)\backslash Q_{\rm s} \Rightarrow$至少有$k$个相异的$s' \in L(G)\backslash K$) s.t. $\delta(q_0,s')=q'$,$\theta(s)=\theta(s') \Rightarrow o_G(s) \geqslant k$. 证毕.

类似于定义8, 利用最小值定义如下的系统隐蔽性裕度的概念:

定义 11 在系统G的观测器obs(G)中, 对于任意估计状态E的隐蔽性裕度$M(E)$, 称其最小值为系统G的隐蔽性裕度, 记为$M(G)=\min\{M(E)|E \in Q_{\rm a}\}$.

上述定义的系统隐蔽性裕度表示系统的秘密在泄露前允许删除可以混淆该秘密的非秘密状态的数目. 显然, 系统的隐蔽性裕度越大, 系统的秘密越隐蔽, 泄露的机会越小. 根据定义11, 给出如算法1(见表1)的计算系统的隐蔽性裕度$M(G)$.

表 1 算法1: 基于对手视野, 计算系统隐蔽性裕度$M(G)$的算法

Table 1 Algorithm 1: For calculating the opacity-margin $M(G)$ of the system on adversarys view

输入: 自动机G、状态子集$Q_{\rm s}$;
输出: 系统隐蔽性裕度$M(G)$.
1 $M(G)=+\infty$;
2 计算自动机G的观测器obs(G),
3 **for each** $E \in Q_{\rm a}$ **do**
4 **if** $E \cap Q_{\rm s} \neq \varnothing$ **then**
5 $M(E)=|E\backslash Q_{\rm s}|$;
6 **else**
7 $M(E)=+\infty$;
8 **end if**
9 $M(G)=\min\{M(G),M(E)\}$;
10 **end for**

由定理3与定理4中的等价关系可知, 基于状态定义的系统隐蔽性裕度(定义11)也可以用来判别k–隐蔽性, 具体内容如下:

定理 5 给定系统G, 则$M(G) \geqslant k$, 当且仅当K关于$L(G)$和$\Sigma_{\rm a}$是k–隐蔽的.

证 $M(G) \geqslant k \Leftrightarrow (\forall q \in Q_{\rm s})\ M(E(q)) \geqslant k \Leftrightarrow (\forall s \in K)\ o_G(s) \geqslant k \Leftrightarrow o_G(K) \geqslant k \Leftrightarrow K$关于$L(G)$和$\Sigma_{\rm a}$是$k$–隐蔽的. 证毕.

由上述定理的证明可得.

推论 3 给定系统G, 则$M(G) \geqslant k$当且仅当$o_G(K) \geqslant k$.

特殊地, 当$k=1$时, 定理5可以简化为隐蔽性的判别定理, 具体结论如下:

推论 4 给定系统G, 则$M(G) \geqslant 1$, 当且仅当K关于$L(G)$和$\Sigma_{\rm a}$是隐蔽的.

例 1 给定一个系统$G=(Q,\Sigma,\delta,q_0)$, 如图1, 其中$\Sigma_{\rm u}=\{f,t\}$. 假设对手了解整个系统, 但其"看到"的事件集为$\Sigma_{\rm a}=\{a,b,d,f,g,t\}$. 设$Q_{\rm s}=\{2,4,7,12,14\}$, 其可以辨识秘密$K=\{aebg(t)^+, a\overline{(bg(fc)^*d)^*}\}$. 为了判别该系统是否具有2–隐蔽性, 则构造观测器obs(G), 如图2. 根据算法1, 计算每一个估计状态的隐蔽性裕度, 见表2. 由表2知, 系统的隐蔽性裕度$M(G)=0$. 再由定理5知, 该系统不是2–隐蔽性的.

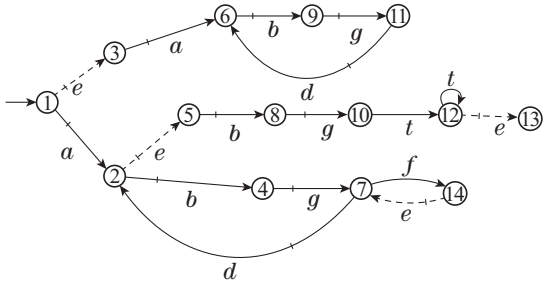

图 1 系统G

Fig. 1 System G

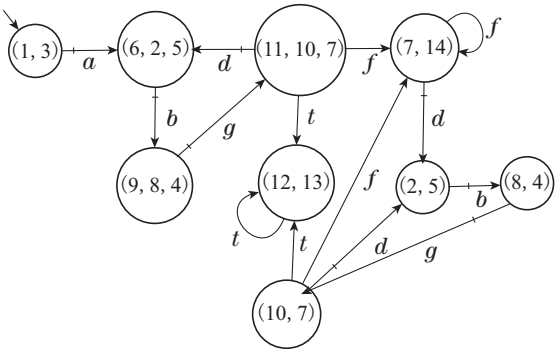

图 2 系统G的观测器

Fig. 2 Observer of system G

4 k–隐蔽的监控综合问题

为了保证系统有足够的隐蔽程度, 可以考虑采用SCT的方法来约束系统的合法行为, 使闭环行为不仅

满足隐蔽性, 而且隐蔽性达到一定的程度, 本文称这种问题为k–隐蔽的监控综合问题, 具体内容如下.

表 2 计算估计状态的隐蔽性裕度

Table 2 Compute opacity-margin of the estimated states

估计状态	隐蔽性裕度
$(1,3)$	∞
$(6,2,5),(9,8,4),(11,10,7)$	2
$(12,13),(10,7),(2,5),(8,4)$	1
$(7,14)$	0

k–隐蔽的监控综合问题: 给定系统G以及容许度k, 设计一个监控器f, 使得秘密语言K关于闭环系统行为$L(f/G)$是k–隐蔽的.

为了解决上述综合问题, 修改算法1获得算法2, 见表3, 使其不仅可以通过计算隐蔽裕度来判别闭环行为的k–隐蔽性, 而且知道哪些秘密行为会被泄漏或隐蔽程度较低.

表 3 算法2: 基于同步积, 计算隐蔽裕度不满足容许度k的状态集

Table 3 Algorithm 2: On parallel composition, calculate the state set where the opacity-margin is not sufficient for the tolerance k

输入: 同步积$G\|\mathrm{obs}(G)$, 状态子集Q_{s}, 容许度k;
输出: 系统的隐蔽性裕度$M(G)$和相应的状态集Δ.

```
初始化Δ = ∅;
初始化M(G) = +∞;
for each (q, E(q)) ∈ Q_{G||obs(G)} do
    if E(q) ∩ Q_s = ∅ then
        M(E(q)) = +∞
    else
        M(E(q)) = |E(q)\Q_s|
    end if
    if M(E(q)) ⩽ k − 1 and q ∈ Q_s then
        Δ = Δ ∪ {(q, E(q))};
    end if
    M(G) = min{M(G), M(E(q))};
end for
```

注 3　由于算法2中只有一个循环, 需要检测$G\|\mathrm{obs}(G)$的每一个状态, 故该算法的计算复杂度为$O(|Q|\times 2^{|Q|})$.

对比算法1和算法2, 虽然两者都可以计算系统的隐蔽性裕度, 但面对的模型不一样. 算法1是基于对手的视野, 对于任一估计状态E给出的算法; 而算法2则是基于系统和其观测器的同步积, 对于任一状态$(q,E(q))$给出的算法. 对比两者的搜索状态数, 由于$|Q_{\mathrm{a}}|\leqslant|Q_{G\|\mathrm{obs}(G)}|$, 则算法1搜索的状态数不会超过算法2搜索的状态数, 故算法1的计算复杂度不会超过算法2的计算复杂度. 而对比两者的隐蔽性裕度的定义, 两者又是一样的, 具体见如下定理6和定理7:

定理 6　任取$(q,E(q)),(q',E(q'))\in Q_{G\|\mathrm{obs}(G)}$ $(q\neq q')$, 如果$E(q)=E(q')$成立, 则有$M(E(q))=M(E(q'))$.

证　由算法2知, $M(E(q))$仅与估计状态$E(q)$有关, 而与状态q无关, 故上述定理成立.　证毕.

定理 7　记算法1中获得的系统隐蔽性裕度为$M_1(G)$, 算法2中获得的系统隐蔽性裕度为$M_2(G)$, 则$M_1(G)=M_2(G)$.

证　欲证$M_1(G)=M_2(G)$, 则根据两种算法中隐蔽性裕度的构造方法, 仅需证$\mathrm{obs}(G)$中的状态集Q_a和$G\|\mathrm{obs}(G)$中状态集的估计状态形成的集合$\{E(q)|\forall(q,E(q))\in Q_{G\|\mathrm{obs}(G)}\}$相等即可, 证明如下:

$\forall E\in Q_{\mathrm{a}}\Rightarrow\exists q\in Q,\mathrm{s.t.}\ (q,E)\in Q_{G\|\mathrm{obs}(G)}\Rightarrow E\in\{E(q)|\forall(q,E(q))\in Q_{G\|\mathrm{obs}(G)}\}\Rightarrow Q_{\mathrm{a}}\subseteq\{E(q)|\forall(q,E(q))\in Q_{G\|\mathrm{obs}(G)}\}$, 而$\forall E\in\{E(q)|\forall(q,E(q))\in Q_{G\|\mathrm{obs}(G)}\}\Rightarrow\exists q\in Q,\mathrm{s.t.}\ E=E(q),(q,E)\in Q_{G\|\mathrm{obs}(G)}\Rightarrow E\in Q_{\mathrm{a}}\Rightarrow\{E(q)|\forall(q,E(q))\in Q_{G\|\mathrm{obs}(G)}\}\subseteq Q_{\mathrm{a}}$.

由上述证明可知, $M_1(G)=M_2(G)$成立.　证毕.

为了解决k–隐蔽监控综合问题, 本文基于文献[20]提出的REFINE算法, 可以直观地给出如下获取k–隐蔽监控器的实现算法3, 见表4.

表 4 算法3: 获得监控器f使得秘密是k–隐蔽的

Table 4 Algorithm 3: Obtain the supervisor f to make the secnet k-opaque

输入: 系统G, 状态子集Q_{s}, 容许度k;
输出: 闭环系统行为$L(f/G)$.

```
1 计算观测器obs(G);
2 计算M = G||obs(G);
3 利用算法2, 获得Δ;
4   if Δ ≠ ∅ then
5       M = REFINE(M, Δ), 算法REFINE见文献[20];
6       Goto 3;
7   end if
8 L(f/G) = L(M).
```

上述算法3获得的闭环系统行为是前述k–隐蔽监控综合问题的解, 具体定理如下:

定理 8　对于算法3中获得的$L(f/G)$, 可得如下结论:

1) K关于$L(f/G)$和Σ_{a}是k–隐蔽的;

2) f是使结论1成立的最大允许的监控器.

为了证明上述定理, 引入文献[20]中关于REFINE算法的一条引理.

引理 1 给定同步积$G\|\mathrm{obs}(G)$和其中的一个状态集Δ, 如果存在一个监控器f使得Δ在$G\|\mathrm{obs}(G)$中不可达, 记$G'=f/G\sqsubseteq G$(即G'是G的子自动机[1]), 则REFINE算法生成$G'\|\mathrm{obs}(G')$.

定理8的证明.

证 先证结论1: K关于$L(f/G)$和Σ_{a}是k–隐蔽的.

在执行算法过程中, 如果算法3第3行得到的$\Delta\neq\varnothing$, 首先考虑第1次循环过程.

在算法3第5行中, 利用REFINE算法中的第4–10行可知, 在$G\|\mathrm{obs}(G)$中删去$\Delta_Q(\neq\varnothing)$时[2], $G\|\mathrm{obs}(G)$中的状态数(即$|Q_{G\|\mathrm{obs}(G)}|$)会减少. 根据算法REFINE的第11–15行, 因为删去$\Delta_Q(\neq\varnothing)$时, 会导致$\mathrm{obs}(G)$中的估计状态被更新, 故在第16行中利用可达运算$Ac(\cdot)$可得新的同步积$G'\|\mathrm{obs}(G')$(见引理1). 在算法3中第5行得到$M=G'\|\mathrm{obs}(G')$. 根据前述分析, 可得$|Q_{G'\|\mathrm{obs}(G')}|<|Q_{G\|\mathrm{obs}(G)}|$, 但$Q_{G'\|\mathrm{obs}(G')}\nsubseteq Q_{G\|\mathrm{obs}(G)}$.

经过有限次循环后, 考虑下列两种情况:

1) 如果在算法3第3行有$\Delta=\varnothing$时, 则由算法2可知$M(G)\geqslant k$, 即K关于$L(M)$和Σ_{a}是k–隐蔽的;

2) 如果在算法3第3行有$\Delta\neq\varnothing$时, 则经过有限次循环后, $G'\|\mathrm{obs}(G')$的状态中的第1元素中不包含秘密状态, 并且更新后的估计状态也不包含秘密状态(因为估计状态更新, 见REFINE算法[20]), 由定义10知, 对于任意$(q,E)\in Q_M$, 可得$M(E)=+\infty$, 即K关于$L(M)$和Σ_{a}是k–隐蔽的.

再证结论2: f是最大允许的监控器. 利用反证法证明如下:

假设f不是最大允许的监控器, 由前述定理1知系统存在最大允许的监控器, 设其为f'. 显然, K关于$L(f'/G)$和Σ_{a}是k–隐蔽的, 并且$L(f/G)\subset L(f'/G)$. 在算法3中, 对于自动机M, 总存在一个状态转移过程可以被监控器f'允许, 但不会被监控器f允许. 由算法3中的REFINE算法知, 该状态转移过程后续的状态转移过程(到达不可达状态集Δ)都是不可控事件. 由假设知, 该状态转移过程被监控器f'允许, 则由闭环系统$L(f'/G)$的可控性知, Δ对于闭环系统是f'/G可达的. 而在算法3中, Δ中的状态的隐蔽性裕度均不超过$k-1$, 这与K关于$L(f'/G)$和Σ_{a}是k–隐蔽的结论矛盾. 故算法3获得的f是使K关于$L(f/G)$和Σ_{a}是k–隐蔽的最大允许的监控器. 证毕.

如果$k=1$, 则算法3可综合隐蔽性, 具体如下:

推论 5 如果$k=1$, 对于算法3中获得的$L(f/G)$, 可使得K关于$L(f/G)$和Σ_{a}是隐蔽的, 并且监控器f是最大允许的.

例 2 在前例中, 系统的秘密K不是2–隐蔽的. 此时需要利用算法3来设计监控器f使得秘密K关于闭环系统$L(f/G)$是2–隐蔽的.

首先, 在算法3中, 获得自动机$M=G\|\mathrm{obs}(G)$, 如图3所示. 在算法3第3行, 利用算法2获得隐蔽性裕度为$k=2$的状态集Δ, 如表5所示.

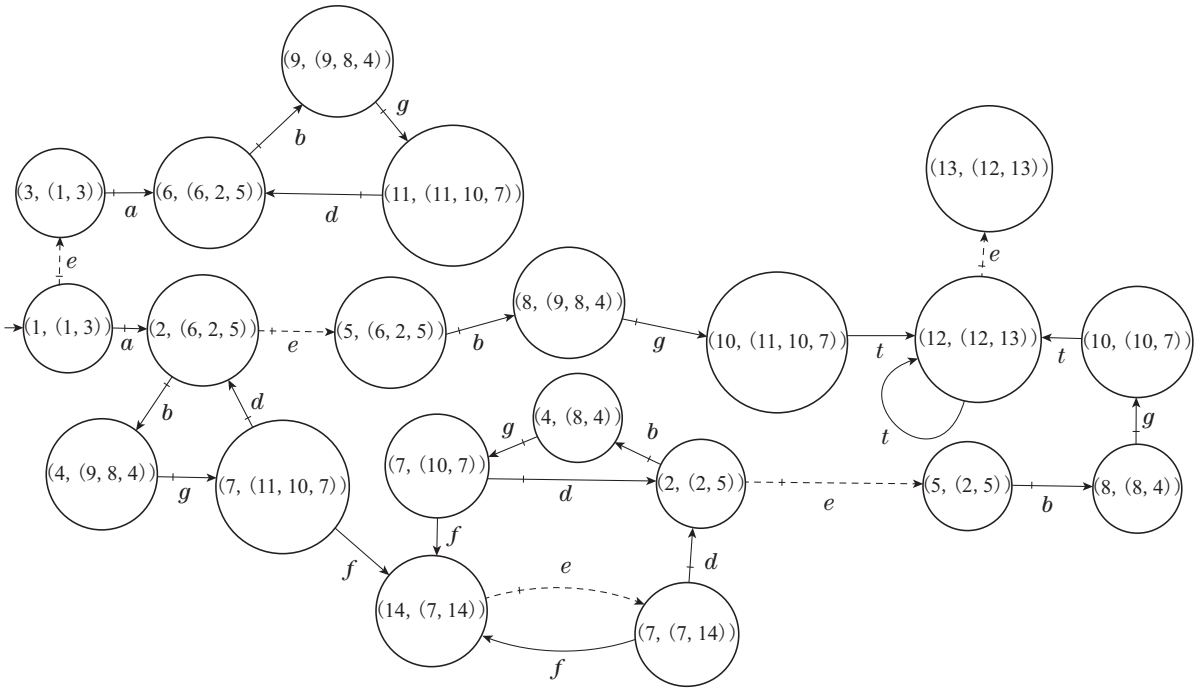

图 3 系统G与观测器$\mathrm{obs}(G)$的同步积

Fig. 3 Parallel composition between system G and its observer $\mathrm{obs}(G)$

[2] Δ_Q见文献[20]的REFINE的算法.

表 5 基于算法2, 计算Δ

Table 5 Compute Δ based on algorithm 2

同步积状态	隐蔽性裕度	Δ $(k=2)$
$(1,(1,3)),(3,(1,3))$	$+\infty$	—
$(6,(6,2,5)),(2,(6,2,5)),(5,(6,2,5)),(9,(9,8,4)),(8,(9,8,4)),(4,(9,8,4))$	2	—
$(11,(11,10,7)),(10,(11,10,7)),(7,(11,10,7))$	2	—
$(12,(12,13)),(13,(12,13))$	1	$(12,(12,13))$
$(10,(10,7)),(7,(10,7))$	1	$(7,(10,7))$
$(2,(2,5)),(5,(2,5))$	1	$(2,(2,5))$
$(8,(8,4)),(4,(8,4))$	1	$(4,(8,4))$
$(7,(7,14)),(14,(7,14))$	0	$(7,(7,14)),(14,(7,14))$

基于Δ, 利用算法3, 构造闭环系统$L(f/G)$, 如图4, 即K关于$L(f/G)$和$\Sigma_{\rm a}$是2–隐蔽的.

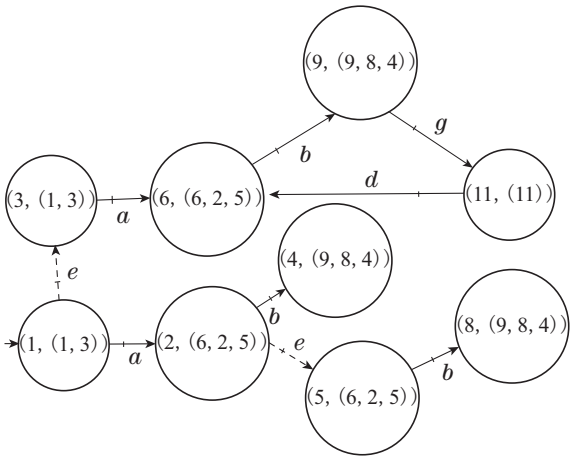

图 4 当容许度$k=2$时的闭环行为$L(f/G)$

Fig. 4 Closed-loop behavior $L(f/G)$ when tolerance $k=2$

由算法3的第4行可得, 闭环系统$L(f/G)$中Δ见表6. 再由表3和定理5易知, K关于$L(f/G)$和$\Sigma_{\rm a}$是2–隐蔽的.

表 6 闭环系统中的Δ

Table 6 Δ of the closed-loop system

同步积状态	隐蔽性裕度	Δ
$(1,(1,3)),(3,(1,3)),(11,(11))$	$+\infty$	—
$(6,(6,2,5)),(2,(6,2,5)),(5,(6,2,5))$	2	—
$(9,(9,8,4)),(8,(9,8,4)),(4,(9,8,4))$	2	—

注 4　算法3的前两行, 计算了观测器和同步积, 其计算复杂度为$\mathrm{O}(|Q|\times 2^{|Q|})$; 算法3中只有一个循环, 中间调用了算法2和REFINE算法, 其中循环次数取决于Δ是否为空, 其最坏的情况为$|Q_{G\|\mathrm{obs}(G)}|$, 算法2的计算复杂度见注3, REFINE算法的计算复杂度为$\mathrm{O}(|Q|\times 2^{|Q|}\times|\Sigma_{G\|\mathrm{obs}(G)}|)$[20], 即$\mathrm{O}(|Q|\times 2^{|Q|}\times|\Sigma|)$. 综上, 算法3的计算复杂度为$\mathrm{O}((|Q|\times 2^{|Q|})^2\times|\Sigma|)$. 经对比, 算法3的计算复杂度与文献[20]的算法2(SYNTHESIZE算法)的计算复杂度相同, 但复杂度高于传统方法, 结合文献[20]的算法3, 在其中镶嵌算法2与REFINE算法来降低计算复杂度, 具体如算法4所示, 见表7.

表 7 算法 4: 获得监控器f使得秘密是k–隐蔽的

Table 7 Algorithm 4: Obtain the supervisorf to make the secnet k-opaque

输入: 系统G, 状态子集$Q_{\rm s}$, 容许度k;
输出: 闭环系统$L(f/G)$.
1 计算观测器$\mathrm{obs}(G)$;
2 计算$M=G\|\mathrm{obs}(G)$;
3 生成一个哈希表H, 其中以$\mathrm{obs}(G)$中的估计状态为关键字, 以M中的状态为映射值;
4 利用算法2, 获得Δ;
5 **for each** $(q,E(q))\in\Delta$;
6 　**for each** $(q',E(q'))\in Q_M$, 如果存在$\sigma\in\Sigma$使得$\delta_M(\sigma,(E(q'),q'))=(q,E(q))$成立;
7 　　**if** $\sigma\notin\Sigma_{\rm c}$ **then**
8 　　$\Delta=\Delta\cup\{(q',E(q'))\}$;
9 　　**end if**
10 　**end for**
11 　$\Delta=\Delta\backslash\{(q,E(q))\}$
12 　在M中删去状态$\{(q,E(q))\}$
13 　在哈希表H中, 找到关键字$E(q)$, 用$E(q)\backslash\{q\}$替换$E(q)$
14 　**if** $q\in Q\backslash Q_{\rm s}$ **then**
15 　　$M(E(q)\backslash\{q\})=M(E(q))-1$
16 　　**if** $M(E(q)\backslash\{q\})<k$ **then**
17 　　$\Delta=\Delta\cup(Q_{\rm s}\cap\{E(q)\backslash\{q\}\})$
18 　　**end if**
19 　**end if**
20 　在M中计算可达状态$Q_{M_{\rm ac}}$
21 　$\Delta=\Delta\cup(Q_M-Q_{M_{\rm ac}})$
22 **end for**
23 $L(f/G)=L(M)$

显然, 算法4将算法3中重复执行REFINE算法的过程变为在哈希表关键字和其映射值的更新, 执行算法2的过程变为更新哈希表中关键字映射的隐蔽性裕度的过程(见第13–18行). 算法4的计算过程与算法3相同. 但其计算复杂度却大大的降低了, 在第3 行生成

哈希表的过程中, 其计算复杂度为$\mathrm{O}(|Q|\times 2^{|Q|})$; 在算法的内外两层循环中, 遍历了整个同步积$M$的全部状态转移, 其计算复杂度为$\mathrm{O}(|Q|\times 2^{|Q|}\times|\Sigma|)$. 综上可知, 算法3的计算复杂度为$\mathrm{O}(|Q|\times 2^{|Q|}\times|\Sigma|)$, 与文献[20]的算法3相同.

5 结论

本文借鉴控制理论中稳定裕度的思想, 定义了基于语言和状态的隐蔽性裕度的概念, 并以此定义衡量非秘密串(或状态)混淆秘密信息的程度. 通过推广隐蔽性的概念为k–隐蔽的概念, 提出验证系统的k–隐蔽性和保证k–隐蔽性的监控综合问题. 由于基于语言隐蔽性裕度验证中需考虑语言的维数, 如果系统中出现循环导致语言是无限集时, 在验证时容易出现存储量过大的情况, 故通过讨论基于语言与基于状态的k–隐蔽性的关系, 将基于语言的k–隐蔽性的验证和监控综合问题转化为基于状态的隐蔽性裕度的计算与判别, 并给出了相应的算法.

参考文献:

[1] CASSANDRAS C, LAFORTUNE S. *Introduction to Discrete Event Systems*. Berlin, Heidelberg, Germany: Springer, 2008.

[2] WONHAM W, CAI K, RUDIE K. Supervisory control of discrete event systems: A brief history. *Annual Reviews in Control*, 2018, 45: 250 – 256.

[3] RAMADGE P, WONHAM W. Supervisory control of a class of discrete-event processes. *SIAM Journal on Control and Optimization*, 1987, 25(1): 206 – 230.

[4] MAZARÉ L. Using unification for opacity properties. *Proceedings of the Workshop on Issues in the Theory of Security (WITS)*. Las Vegas, LV, USA: Association for Computing Machinery, 2004, 4: 165 – 176.

[5] BRYANS J, KOUTNY M, RYAN P. Modelling opacity using Petri net. *Electronic notes in Theoretical Computer Science*, 2005, 121: 101 – 115.

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

[7] SABOORI A, HADJICOSTIS C. Verification of initial-state opacity in security applications of DES. *Proceedings of the 9th International Workshop on Discrete Event Systems*. Gothenburg, Sweden: IEEE, 2008: 328 – 333.

[8] SABOORI A, HADJICOSTIS C. Opacity-enforcing supervisory strategies via state estimator constructions. *IEEE Transactions on Automatic Control*, 2012, 57(5): 1155 – 1165.

[9] SABOORI A, HADJICOSTIS C. Verification of k-step opacity and analysis of its complexity. *IEEE Transactions on Automatic Control*, 2011, 8(3): 549 – 559.

[10] WU Y, LAFORTUNE S. Comparative analysis of related notions of opacity in centralized and coordinated architectures. *Discrete Event Dynamic Systems*, 2013, 23(3): 307 – 339.

[11] YIN X, ZAMANI M, LIU A. On approximate opacity of cyber-physical systems. *IEEE Transactions on Automatic Control*, 2021, 66(4): 1630 – 1645.

[12] FALCONE Y, MARCHAND H. Enforcement and validation (at runtime) of various notions of opacity. *Discrete Event Dynamic Systems*, 2015, 25(4): 531 – 570.

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

[14] HOU J, YIN X, LI S. A framework for current-state opacity under dynamic information release mechanism. *Automatica*, 2022, 140: 110238.

[15] LIU Fuchun, ZHAO Yipeng, ZHAO Rui. Verification algorithm for opacity of discrete-event systems with rough set theory. *Control Theory & Applications*, 2019, 36(8): 1259 – 1264.
(刘富春, 赵毅澎, 赵锐. 基于粗糙集理论的离散事件系统不透明性的验证算法. 控制理论与应用, 2019, 36(8): 1259 – 1264.)

[16] ZHOU Y, CHEN Z, LIU Z. Verification and enforcement of current-state opacity based on a state space approach. *European Journal of Control*, 2023, 71: 100795.

[17] YIN X, LAFORTUNE S. A new approach for the verification of infinite-step and K-step opacity using two-way observers. *Automatica*, 2017, 80: 162 – 171.

[18] MA Z, YIN X, LI Z. Verification and enforcement of strong infinite- and K-step opacity using state recognizers. *Automatica*, 2021, 133: 109838.

[19] DUBREIL J, DARONDEAU P, MARCHAND H. Supervisory control for Opacity. *IEEE Transactions on Automatic Control*, 2010, 55(5): 1089 – 1100.

[20] MOULTON R, HAMGINI B, KHOUZANI Z, et al. Using subobservers to synthesize opacity enforcing supervisors. *Discrete Event Dynamic Systems*, 2022, 32(4): 611 – 640.

[21] LIU R, LU J. Enforcement for infinite-step opacity and K-step opacity via insertion mechanism. *Automatica*, 2022, 140: 110212.

[22] JI Y, WU Y, LAFORTUNE S. Enforcement of opacity by public and private insertion functions. *Automatica*, 2018, 93: 369 – 378.

[23] JI Y, YIN X, LAFORTUNE S. Enforcing opacity by insertion functions under multiple energy constraints. *Automatica*, 2019, 108: 108476.

[24] JI Y, YIN X, LAFORTUNE S. Opacity enforcement using nondeterministic publicly-known edit functions. *IEEE Transactions on Automatic Control*, 2019, 64(10): 4369 – 4376.

[25] YIN X, LI S. Synthesis of dynamic masks for infinite-step opacity. *IEEE Transactions on Automatic Control*, 2020, 65(4): 1429 – 1441.

[26] YIN X, LI Z, WANG W, et al. Infinite-step opacity and K-step opacity of stochastic discrete-event systems. *Automatica*, 2019, 99: 266 – 274.

[27] YANG J, DENG W, QIU D. Current-state opacity and initial-state opacity of modular discrete event systems. *International Journal of Control*, 2022, 95(11): 3037 – 3049.

[28] XIE Y, YIN X, LI S. Opacity enforcing supervisory control using non-deterministic supervisors. *IEEE Transactions on Automatic Control*, 2022, 67(12): 6567 – 6582.

[29] YANG J, DENG W, QIU D, et al. Opacity of networked discrete event systems. *Information Sciences*, 2021, 543: 328 – 344.

[30] LIN F, WANG L, CHEN W, et al. Information control in networked discrete event systems and its application to battery management systems. *Discrete Event Dynamic Systems*, 2020, 30(2): 243 – 268.

[31] SCHONEWILLE B. *Enforcing security on autonomous vehicle searches through the quantification of opacity*. Kingston: Queen's University, 2021.

[32] HOPCROFT J, MOTWANI R. *Introduction to Automata Theory, Languages, and Computation*. 3rd Edition. Boston, MA, USA: Addison Wesley, 2007.

[33] BEN-KALEFA M, LIN F. Opaque superlanguages and sublanguages in discrete event systems. *Cybernetics and Systems*, 2016, 47(5): 392 – 426.

作者简介:

王 飞 副教授, 博士, 目前研究方向为离散事件系统的监控理论, E-mail: feiw545@163.com;

戴茵茵 讲师, 博士研究生, 目前研究方向为离散事件系统控制, E-mail: crystle@hqu.edu.cn;

金福江 教授, 博士生导师, 目前研究方向为复杂系统建模与控制, E-mail: jinfujiang@163.com.