上海交通大學

SHANGHAI JIAO TONG UNIVERSITY

# 学士学位论文

BACHELOR'S THESIS


论文题目：基于不透明性的 CPS 信息安全分析与设计

学生姓名：杨烁

学生学号：517021911071

专　　业：自动化

指导教师：殷翔

学院 (系)：电子信息与电气工程学院

Submitted in partial fulfillment of the requirements for the degree of Bachelor in Automation

# Verification and Synthesis of Opacity for Cyber-Physical Systems


Shuo Yang

Advisor

Prof. Xiang Yin

Department of Automation

Shanghai Jiao Tong University

Shanghai, P. R. China


May 25th, 2021

# 基于不透明性的 CPS 信息安全分析与设计

## 摘　要

信息物理系统（CPS）是包含受计算单元控制的物理组件的工程系统。其中网络层涉及感知，控制，决策和计算；物理层主要由物理对象组成，例如工业过程，能源系统和运输。由于计算机和网络已广泛集成到我们的日常生活中，因此对这些系统的物理安全，信息安全，性能和保证的需求变得日益重要。因此，使用基于形式化模型的方法来可验证地验证和合成属性的必要性非常迫切。

本文着重研究离散事件系统（DES）中的不透明性这一安全概念，DES 是 CPS 的重要一类。不透明性是一种信息流属性，用于表征外部入侵者是否可以根据其观察结果确定系统是否处于秘密状态。

本文主要解决了与不透明性有关的两个问题。首先，我们在 DES 中提出了一类新的不透明性，称为预测不透明性，它捕获了系统未来秘密信息的合理可信性。现有的不透明性概念仅将机密视为当前正在访问某些机密状态或过去已访问过某些机密状态。我们在文中提供了预测不透明性的验证算法及其复杂度。我们还将此设定推广到系统的秘密意图被建模为执行特定事件序列而不是访问秘密状态的情况。

其次，我们考虑不透明性在机器人路径规划中的应用。具体来说，我们提出了一种要求安全并且最优的线性时序逻辑路径规划问题。安全约束要求入侵者永远不能推断机器人是从秘密位置出发的，这个安全约束是根据基于初始状态的不透明性而改编的。我们提供了完善而完整的算法程序来解决此问题。我们的方法基于双有权转移系统（twin-WTS）的构建，该系统跟踪具有相同观测值的一对路径。

最后，我们通过两个实际例子分别说明了目的安全性和初始状态安全性在机器人路径规划中的应用。

**关键词：** 信息物理系统, 离散事件系统, 不透明性, 预测, 时序逻辑, 规划

# VERIFICATION AND SYNTHESIS OF OPACITY FOR CYBER-PHYSICAL SYSTEMS

## ABSTRACT

Cyber-physical systems (CPS) are engineering systems that involve physical components which are controlled by computational units. Cyber layer involves perception, control, decision, and computation; physical layer mainly consists of physical objects, such as industry process, energy systems, and transportation. The demand for physical safety, information security, performance, and certification of these systems is becoming significant, since computers and networks are widely integrated into our daily lives. Therefore, the necessity of using formal model-based methods to provably verify and enforce properties is stringent.

This dissertation focuses on a security notion called opacity in Discrete-event systems (DES), an important class of CPS. Opacity is an information-flow property which is used to characterize whether the outside intruder can determine for sure that the system is at a secret or not based on its observations.

This dissertation mainly tackles two problems related to opacity. First, we propose a new class of opacity called pre-opacity in DES, which captures the plausible deniability of the future secret information of the systems. Existing notions of opacity only consider secret either as currently visiting some secret states or as having visited some secret states in the past. The verification algorithm of pre-opacity and its complexity are provided. We also generalize our setting to the case where the secret intention of the system is modeled as executing a particular sequence of events rather than visiting a secret state.

Second, we consider the application of opacity in robot path planing. Specifically, a security-aware optimal linear temporal logic path planning problem is proposed. The security constraint requires that the intruder should never infer that the robot was started from a secret location, which is adapted from the notion of initial-state opacity. We provide a sound and complete algorithmic procedure to solve this problem. Our approach is based on the construction of the twin weighted transition systems (twin-WTS) that tracks a pair of paths having the same observation.

We finally illustrate the applications of intention-security and initial-state-security respectively in robot path planning by two cases.

**KEY WORDS:** cyber-physical system, discrete-event system, opacity, prediction, temporal logic, planning

# Contents

# List of Figures

# List of Abbreviations

| | |
|---|---|
| **CPS** | Cyber-Physical Systems |
| **DES** | Discrete-Event Systems |
| **DFA** | Deterministic Finite-Atate Automaton |
| **LTL** | Linear Temporal Logic |
| **WTS** | Weighted Transition System |
| **NBA** | Nondeterministic Büchi Automaton |

# Chapter 1 Introduction

## 1.1 Background

Cyber-security is an increasingly important issue in cyber-physical systems (CPS). In the study of modern large-scale CPS, two research directions are crucial: verification and synthesis. We are first interested in whether the given system satisfies a property of interest or not; if the verification result is negative, we then would like to synthesize some strategies to enforce this property.

Discrete-event systems (DES) is an important class of CPS. In the context of DES, opacity is an information-flow property which essentially characterizes whether the system's secret information can be revealed by outside malicious intruder or not. Finite State Automaton (FSA) is considered as a DES model. We formally formalize the notions of opacity, analyze opacity, and apply opacity to practical situations.

## 1.2 Literature Review

In the context of DES, the notions of opacity have been well-studied. It was initiated by the computer science community [1, 2]. It then becomes an active topic in DES field, since DES provides a very suitable model for formal analysis of opacity. For instance, finite-state automata is used most commonly, see, e.g., [3-6]. Also, labeled transition systems [1, 7] and Petri nets [2, 8-11] are also widely used. More recently, opacity has been extended to continuous dynamic systems with possibly infinite state spaces and time-driven dynamics [12-14].

### 1.2.1 Opacity Notions

In the past decade, the notion of opacity has drawn a lot of attention in the Discrete-Event Systems (DES) literature as it provides a formal approach towards the verification and design of information-flow security for dynamics systems. Roughly speaking, opacity is a confidentiality property that captures whether or not the information-flow generated by a dynamic system can reveal some "secret behavior" to an outside observer (intruder) that is potentially malicious. In other words, an opaque system should always maintain the plausible deniability for its secret behavior during its execution. In the context of DES, opacity has been extensively studied for

different system models including finite-state automata [3-5, 15], labeled transition systems [1, 7] and Petri nets [2, 8-10]. More recently, opacity has been extended to continuous dynamic systems with possibly infinite state spaces and time-driven dynamics [12-14]. Many enforcement techniques have also been proposed when the original system is not opaque; see, e.g., [16-22]. Opacity has also been applied to certify/enforce security in many real-world systems including mobile robots [23], location-based services [24] and web services [25]. The reader is referred to the survey papers [26, 27] for more details on opacity and its recent developments.

In order to capture different security requirements, different notions of opacity have been proposed in the literature. For example, in language-based opacity [28], the secret is formulated as the executions of some particular secret strings. As shown in [29], this formulation is equivalent to the notion of current-state opacity, where the secret is formulated as a set of secret states and a system is current-state opaque if the intruder cannot determine for sure that the system is currently at a secret state. In some situations, the system may want to hide its initial location or its location at some specific previous instant; such requirements can be captured by initial-state opacity [4] and $K$/infinite-step opacity [3, 5, 30, 31], respectively. More recently, quantitative notions of opacity have been proposed for stochastic DES in order to measure the secret leakage of the system; see, e.g., [32-37].

As we can see from the above discussion, “secret" in opacity analysis is actually a generic concept. Based on what kind of information the user would like to hide, or equivalently, how the intruder can utilize information to infer the secret of the system, existing notions of opacity in the literature as reviewed above can generally be divided into the following two categories:

- Opacity for Current Information: the intruder wants to determine the current behavior of the system based on the current observation. In other words, the user does not want the outsider to know for sure that it *is currently doing* something secret. This category includes, e.g., current-state opacity and language-based opacity.
- Opacity for Delayed Information: the intruder wants to determine the previous secret behavior of the system at some instant based on the current observation. In other words, the user does not want the outsider to know for sure that it *has done* something secret at some previous instant. This category includes, e.g., initial-state opacity, $K$-step opacity and infinite-step opacity. Note that delayed information is involved here as the intruder does not need to specify the visit of a secret state immediately; it can use future information to improve its knowledge about the previous instants.

There are also some works that combine these two types of opacity together, e.g., by combing

current-state opacity and initial-state opacity, one can define the notion of initial-final-state opacity [29].

### 1.2.2 Temporal-Logic-Based Path Planning

Path planning is a fundamental problem in robotics which asks to generate a planned trajectory from an initial location such that such desired requirements can be fulfilled. Classical planning problems usually focus on lower-level tasks such as obstacle avoidance or point-to-point navigation [38, 39]. In the past decades, temporal-logic-based high-level path planning for complex tasks has drawn considerable attention in the literature; see, e.g., [40-45]. In this framework, the planning task is specified by linear temporal logic (LTL) or computation tree logic (CTL). In particular, LTL can be used to represent many important properties such as safety, liveness and priority [46]. By using automata-theoretic approach, algorithmic procedures are developed to automatically generate correct-by-construction plans to achieve the given temporal tasks.

While the temporal-logic-based planning has been extensively investigated for safety requirements, security and privacy are also important concerns in many applications. For instance, in robot data gathering problem, a robot needs to visit different locations in order to gather data and then to transmit collected data to the cloud. However, the data transmission may not be secure in the sense that there may exist an eavesdropper “listening" the communication. Such information leakage may reveal some crucial secret behavior of the robot, e.g., some information the robot does not intend to transmit may be *inferred* by the intruder. Therefore, one also needs to incorporate such a security constraint in the path planning algorithm. Due to its importance, security and privacy concerns have been drawing attentions in the robot path planning literature; see, e.g., [47, 48].

Optimal LTL path planning problem was originally formulated in [49], where the optimization objective is to minimize the worst cost between each satisfying instances. This framework has been extended to the case of multi-robot [50, 51], where each robot may have a local task or a team of robots need to collaborate to achieve a global task. Recently, sampling-based techniques have been applied to improve the scalability of optimal path planning algorithm [52, 53]. Optimal temporal logic path planning problems have also bee studied for stochastic systems modeled as MDPs; see, e.g., [54-57]. However, none of the above mentioned works considers security constraint.

In the context of security-aware path planning, our work is most related to [58]. The differences between our work and [58] are as follows. First, the planning task considered in our work is a general LTL formula which is satisfied over infinite path, while [58] considers a simple reachability task which can be satisfied within a finite horizon. Second, no optimality is consider in [58]. Finally, the security requirement considered in our work is different with that in [58]. Specifically, [58] considers to protect the *current* secret location of the robot, while we consider to protect the *initial* secret location of the robot. We show that initial-type secret has a nice property that it suffices to track a pair of observational equivalent states in the system. Therefore, the complexity of our planning algorithm is *independent* from the number of secret states and is always quadratic in the number of system states. However, the complexity of the planning algorithm in [58] is based on the structure of $K$-detector, whose size grows exponentially as the number of secret states increases.

In the computer science literature, the concept of *hyper-properties* [59] has drawn many attentions in the past years, e.g., HyperLTL [60]. In particular, hyper-properties are closely related to security requirements as it allows to specify the relationships among multiple paths. Very recently, the authors of [61] show that initial-state opacity planning problem can be specified as an instant of the HyperLTL planning problem; symbolic algorithms for finite synthesis is also provided therein. This result is closely related to our results. However, initial-state opacity considered in [61] is based on the equivalence of atomic propositions. In our setting, atomic propositions are only used to specify the desired internal temporal task, while the observation equivalence is specified by a new output function. This setting is more general as the atomic propositions and the outputs can be different. Furthermore, our planning algorithm is customized to initial-state security, which avoids the higher general complexity in HyperLTL synthesis.

Finally, our work is also related to opacity-enforcing supervisory control in the context of discrete-event systems [17, 62-64]. However, the opacity control problem is essentially a reactive synthesis problem whose complexity is exponential in the size of the system. Here, we consider a security-aware path planning problem that can be solved more efficiently. Furthermore, no LTL specification and optimality is considered in the opacity control problem.

## 1.3 Organization and Main Contributions

The organization and main contributions of this dissertation are summarized as follows.

**Chapter 2: System Models**

In this chapter, we provide two system models used in this dissertation. Specifically, we model discrete-event systems by deterministic finite-state automaton; the mobility of the robot in workspace is modeled as a weighted transition system.

**Chapter 3: Secure Your Intention: On Notions of Pre-Opacity in Discrete-Event Systems ([65])**

In this chapter, we investigates an important information-flow security property called opacity in partially-observed discrete-event systems. We consider the presence of a passive intruder (eavesdropper) that knows the dynamic model of the system and can use the generated information-flow to infer some "secret" of the system. A system is said to be opaque if it always holds the plausible deniability for its secret. Existing notions of opacity only consider secret either as currently visiting some secret states or as having visited some secret states in the past. In this chapter, we investigate information-flow security from a new angle by considering the secret of the system as the intention to execute some particular behavior of importance in the future. To this end, we propose a new class of opacity called pre-opacity that characterizes whether or not the intruder can predict the visit of secret states a certain number of steps ahead before the system actually does so. Depending the prediction task of the intruder, we propose two specific kinds of pre-opacity called $K$-step instant pre-opacity and $K$-step trajectory pre-opacity to specify this concept. For each notion of pre-opacity, we provide a necessary and sufficient condition as well as an effective verification algorithm. The complexity for the verification of preopacity is exponential in the size of the system as we show that pre-opacity is inherently PSPACE-hard. Finally, we generalize our setting to the case where the secret intention of the system is modeled as executing a particular sequence of events rather than visiting a secret state.

**Chapter 4: Secure-by-Construction Optimal Path Planning for Linear Temporal Logic Tasks ([66])**

In this chapter, we investigate the problem of planning an optimal infinite path for a single robot to achieve a linear temporal logic (LTL) task with security guarantee. We assume that the external behavior of the robot, specified by an output function, can be accessed by a passive intruder (eavesdropper). The security constraint requires that the intruder should never infer that the robot was started from a secret location. We provide a sound and complete algorithmic procedure to solve this problem. Our approach is based on the construction of the twin weighted

transition systems (twin-WTS) that tracks a pair of paths having the same observation. We show that the security-aware path planning problem can be effectively solved based on graph search techniques in the product of the twin-WTS and the Büchi automaton representing the LTL formula. The complexity of the proposed planning algorithm is polynomial in the size of the system model.

**Chapter 5: Case Studies**

In this chapter, two cases on security-aware robot path planning are presented. The first one is protecting robot's intention, and this type of security is naturally captured by the notion of pre-opacity. The second case aims at securing robot's starting point and it illustrates the algorithm proposed in Chapter 4.

**Chapter 6: Summary**

In this chapter, We conclude this dissertation and discuss several interesting future research directions.

# Chapter 2 System Models

In this chapter, we present two system models used in this dissertation. Specifically, we model a discrete-event system by *deterministic finite-state automaton*; the mobility of the robot in workspace is modeled as a *weighted transition system*.

## 2.1 Deterministic Finite-State Automaton

Let $E$ be a finite set of events. A *string* is a finite sequence of events and we denote by $E^*$ the set of all strings over $E$ including the empty string $\epsilon$. For any string $s \in E^*$, we denote by $|s|$ the length of $s$ with $|\epsilon| = 0$. A language $L \subseteq E^*$ is a set of strings, and $\bar{L}$ denotes the prefix-closure of $L$, i.e., $\bar{L} = \{u \in E^* : \exists v \in E^* \text{ s.t. } uv \in L\}$.

We consider a discrete-event system modeled by a deterministic finite-state automaton (DFA)

$$G = (X, E, f, X_0),$$

where $X$ is the finite set of states, $E$ is the finite set of events, $f : X \times E \to X$ is the partial deterministic transition function such that $f(x, \sigma) = x'$ means that there exists a transition from $x$ to $x'$ with event label $\sigma$, and $X_0 \subseteq X$ is the set of initial states. The transition function $f$ is also extended to $f : X \times E^* \to X$ recursively by: for any $x \in X, s \in E^*, \sigma \in E$, we have $f(x, s\sigma) = f(f(x, s), \sigma)$ with $f(x, \epsilon) = x$.

The language generated by $G$ from state $x \in X$ is defined by $\mathcal{L}(G, x) = \{s \in E^* : f(x, s)!\}$, where "!" means "is defined". Also, we define $\mathcal{L}(G, Q) := \bigcup_{x \in Q} \mathcal{L}(G, x)$ as the language generated from a set of states $Q \subseteq X$. Therefore, the language generated by $G$ is $\mathcal{L}(G) := \mathcal{L}(G, X_0)$. For the sake of simplicity, hereafter, we assume that the system $G$ is live, i.e., for any $x \in X$, there exists $\sigma \in \Sigma$ such that $f(x, \sigma)!$. In some situations, a DFA is also equipped with a set of *marked states* $X_m \subseteq X$ and we write a DFA with marked states as $G = (X, E, f, X_0, X_m)$. Then the marked language of $G$ is $\mathcal{L}_m(G) = \{s \in E^* : \exists x_0 \in X_0 \text{ s.t. } f(x, s) \in X_m\}$.

## 2.2 Weighted Transition Systems

We consider a scenario where single mobile robot works in a workspace $\mathcal{W} \subseteq \mathbb{R}^2$. The workspace is partitioned as $n$ disjoint regions of interest denoted by $r_1, \ldots, r_n$ and we denote

by $\mathcal{I} := \{1, \cdots, n\}$ the index set. In general, workspace regions can be of any arbitrary shape partitioned based on the task properties and the dynamic of the robot; see, e.g., [67, 68] for details on region partition. In this work, we focus on the task planning problem; hence, we model the mobility of the robot in workspace as a *Weighted Transition System* (WTS) defined as follows.

**Definition 1.** *(Weighted Transition System) A weighted transition system is a 6-tuple*

$$T = (Q, Q_0, \rightarrow, w, \mathcal{AP}, L),$$

*where*

- $Q = \{q_i : i \in \mathcal{I}\}$ *is the set of states and each state* $q_i$ *indicates that the robot is at location* $r_i$*;*
- $Q_0 \subseteq Q$ *is the set of initial states representing all possible starting locations of the robot;*
- $\rightarrow \subseteq Q \times Q$ *is the transition relation such that* $(q_i, q_j) \in \rightarrow$ *means that there exists a controller that can drive robot from region* $r_i$ *to region* $r_j$ *without going through any other regions;*
- $w : Q \times Q \rightarrow \mathbb{R}_+$ *is a cost function that assigns each transition* $(q_i, q_j) \in \rightarrow$ *a positive weight* $w(q_i, q_j)$ *representing the cost driving the robot from region* $r_i$ *to region* $r_j$*, e.g., the distance between* $r_i$ *and* $r_j$*;*
- $\mathcal{AP}$ *is the set of atomic propositions representing some basic properties of interest;*
- $L : Q \rightarrow 2^{\mathcal{AP}}$ *is the labeling function that assigns each state a set of atomic propositions.*

Given a WTS $T$, an *infinite internal path* is an infinite sequence of states $\tau = \tau(1)\tau(2)\tau(3)\cdots \in Q^\omega$ such that $\tau(1) \in Q_0$ and $(\tau(i), \tau(i+1)) \in \rightarrow, \forall i \in \mathbb{N}_+$. A *finite internal path* of a WTS is defined analogously. Hereafter, an internal path will just be referred as path for the sake of simplicity. We denote by $\mathrm{Path}^\omega(T)$ and $\mathrm{Path}^*(T)$ the set of all infinite paths and the set of all finite paths in $T$, respectively. The cost function $w$ is considered to be additive; therefore, the cost of a finite path $\tau \in \mathrm{Path}^*(T)$, denoted by $J(\tau)$, is defined by

$$J(\tau) = \sum_{i=1,\ldots,|\tau|-1} w(\tau(i), \tau(i+1)), \tag{2–1}$$

where $|\tau|$ is the length of the path. In words, the cost $J(\tau)$ captures the total cost incurred by during the execution of finite path $\tau$.

# Chapter 3 Secure Your Intention: On Notions of Pre-Opacity in Discrete-Event Systems

## 3.1 Introduction

In this chapter, we investigate opacity from a new angle by considering the system's *intention* of executing some particular behavior as the secret. Then we propose a new type of opacity, called *pre-opacity*, to characterize whether or not the secret intention of the system can be revealed. We follow the standard setting of opacity by considering a passive intruder modeled as an eavesdropper that knows the model of the system. Then we propose two notions of pre-opacity called *$K$-step instant pre-opacity* and *$K$-step trajectory pre-opacity*; the former requires that the intruder cannot determine $K$-step ahead for sure that the system will be at secret states *for some specific instant*, while the latter requires that the intruder cannot determine $K$-step ahead for sure that the system will visit secret states in the future without the need of specifying the instant of being secret. Properties of these two notions of pre-opacity are investigated and we show that instant pre-opacity is strictly weaker than trajectory pre-opacity. Furthermore, for each pre-opacity, we provide necessary and sufficient condition as well as effective verification algorithm. We show that both properties are PSPACE-hard; hence the exponential verification complexity is unavoidable. Also, we discuss the case where "secrets" are modeled as a *sequence pattern* rather secret states.

In the systems theory, there are three fundamental types of estimations problems: filtering, smoothing and prediction. Essentially, current-state opacity can be viewed as the plausible deniability for secret under filtering and infinite/$K$-step opacity can be viewed as the plausible deniability for secret under smoothing. Analogously, the proposed notion of pre-opacity can also be interpreted as the plausible deniability for secret under prediction. Therefore, our new notion also generalizes the framework of opacity from the systems theory point of view.

The proposed notion of pre-opacity, in particular, trajectory pre-opacity, is closely related to the notion of fault predictability (or prognosability) in the literature; see, e.g., [69-74]. However, predictability requires that any fault can be predicted before its occurrence, but our notion of pre-opacity requires that any secret cannot be predicted before it actually happens. Furthermore, our notion of instant pre-opacity is much more different since it requires to determine the precise

instant of being secret, which is not required in predictability analysis. Also, in fault prediction problems, once the system becomes faulty, it is faulty forever. However, in pre-opacity analysis, the system's behavior can become secret/non-secrete intermittently in the sense that, even when the intruder fails to predict the first secret behavior, it may still has chance to predict some future secret so that the security of the system can still be threatened. Therefore, although predictability is conceptually related to our notion of pre-opacity, these two properties are technically very different.

The rest of the chapter is organized as follows. In Section 3.2, we describe the system model and review the existing notions of opacity. Section 3.3 introduces the two new notions of pre-opacity and discusses their properties. In Section 3.4, we provide effective algorithms for the verification of notions of pre-opacity. The proposed pre-opacity is further generalized to the case of sequence pattern in Section 3.5. Finally, we conclude this chapter by Section 3.6.

## 3.2 Preliminaries

Following the standard setting of opacity, we assume that the intruder is modeled as a *passive observer* (eavesdropper), which has the full knowledge of the system's structure. By "passive", we mean that the intruder can only observe some behavior generated by the system, but it cannot actively affect the behavior of the system. Formally, we assume that the event set $E$ is partitioned as:

$$E = E_o \dot{\cup} E_{uo},$$

where $E_o$ and $E_{uo}$ are the set of observable events and the set of unobservable events, respectively. The natural projection from $E$ to $E_o$ is a mapping $P : E^* \to E_o^*$ defined recursively by:

$$P(\epsilon) = \epsilon \text{ and } P(s\sigma) = \begin{cases} P(s)\sigma & \text{if } \sigma \in E_o \\ P(s) & \text{if } \sigma \notin E_{uo} \end{cases} \tag{3–1}$$

The natural projection is also extended to $P : 2^{E^*} \to 2^{E_o^*}$, i.e., $P(L) = \{t \in E_o^* : \exists s \in L \text{ s.t. } P(s) = t\}$ for any $L \subseteq E^*$.

When string $s \in \mathcal{L}(G)$ is generated by the system, the intruder observes $P(s)$ and it can use this observation together with the dynamic model of the system to infer which state the system could be in at some specific instant. In opacity analysis, it is assumed that the system has a set of secret states, denoted by $X_S \subseteq X$. Roughly speaking, a system is said to be opaque if the intruder can never determine for sure that the system is/was at a secret state based on

its observation. Here, we review the notion of $K$-step opacity which can be used to define current-state opacity and infinite-step opacity.

**Definition 2.** *($K$-Step Opacity) Given system $G$, set of observable events $E_o$, set of secret states $X_S$, and non-negative integer $K \in \mathbb{N}$, system $G$ is said to be* $K$-step opaque *(w.r.t. $E_o$ and $X_S$) if*

$$\begin{aligned}&(\forall x_0 \in X_0, \forall st \in \mathcal{L}(G, x_0) : f(x_0, s) \in X_S \wedge |P(t)| \leq K)\\&(\exists x_0' \in X_0)(\exists s't' \in \mathcal{L}(G, x_0')) \text{ s.t.}\\&[P(s) = P(s')] \wedge [P(t) = P(t')] \wedge [f(x_0', s') \notin X_S].\end{aligned} \tag{3–2}$$

*Furthermore, system $G$ is said to be*

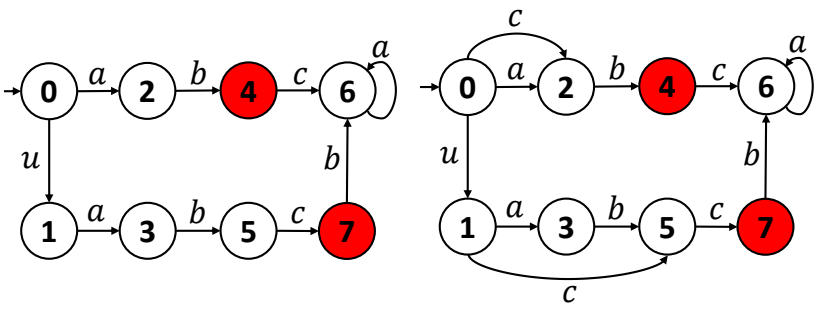
- current-state opaque *if it is* 0*-step opaque;*
- infinite-step opaque *if it is $K$-step opaque for any $K \geq 0$.*

Intuitively, $K$-step opacity says that, whenever the system visits a secret state, it should be able to keep this secret unrevealed within the next $K$ steps. In other words, the intruder should never be able to determine that the system was at a state secret for any instant in the past $K$ steps. Note that current-state opacity can be viewed as a special case of $K$-step opacity ($K = 0$) as it essentially requires that the intruder cannot determine for sure that the system is currently at a secret state. To verify current-state opacity, one can construct the current-state estimator (or the observer) and check whether or not there exists a reachable estimator state that only contains secret states. The verification of $K$-step opacity and infinite-step opacity are more involved as they require the computation of delayed state estimate, which can be done by constructing the two-way observer[5].

**Example 1.** *Let us consider system $G_1$ shown in Figure 3–1(a), where $X_S = \{4, 7\}$ and $E_o = \{a, b, c\}$. Clearly, this system is current-state opaque. For example, by observing $ab$, the intruder cannot determine whether the system is at secret state* 4 *or at non-secret state* 5 *since $P(ab) = P(uab) = ab$. Similarly, when secret state* 7 *is reached via $uabc$, the intruder still cannot distinguish this state from non-secret state* 6*. On the other hand, this system is not* 1*-step opaque. This is because, by observing $abcb$, the intruder can determine for sure that the system was at secret state* 7 *one step ago. Therefore, $G_1$ is also not infinite-step opaque.*

## 3.3 Notions of Pre-Opacity

In this section, we first provide the definitions of $K$-step instant pre-opacity and $K$-step trajectory pre-opacity for DES. Then we discuss properties of the proposed notions of pre-opacity.




(a) $G_1$ (b) $G_2$

Figure 3–1 For both systems, we have $X_0 = \{0\}$, $X_S = \{4, 7\}$ and $E_o = \{a, b, c\}$.

.

### 3.3.1 Definitions of $K$-step Instant/Trajectory Pre-Opacity

In the existing notions of opacity, secret is either characterized by whether the system is doing something secret (current-state opacity) or characterized by whether the system has done something secret ($K$-step and infinite-step opacity). These settings essentially assume that the system is operating against an intruder whose functionality is a current-state estimator or a delayed state-estimator.

However, in some applications, what the system wants to hide might be its *intention*, i.e., maintain the plausible deniability for its willing to do something secret in the future. In this setting, the system is essentially operating against an intruder that can be interpreted as a *predictor*. More specifically, the user may require that the intruder should never be able to determine its intention of visiting a secret state too early, which is characterized by a parameter $K$. To this end, we first propose the notion of *$K$-step instant pre-opacity* as follows; the reason why we use terminology “instant" here will be clear soon.

**Definition 3.** *($K$-Step Instant Pre-Opacity) Given system $G$, set of observable events $E_o$, set of secret states $X_S$, and non-negative integer $K \in \mathbb{N}$, system $G$ is said to be $K$-step instant pre-opaque (w.r.t. $E_o$ and $X_S$) if*

$$\begin{aligned}&(\forall x_0 \in X_0, \forall s \in \mathcal{L}_o(G, x_0))(\forall n \geq K)\\ &(\exists x_0' \in X_0, \exists s' \in \mathcal{L}_o(G, x_0'), \exists t \in \mathcal{L}(G, f(x_0', s'))\ s.t.\\ &[P(s) = P(s')] \wedge [|t| = n] \wedge [f(x_0', s't) \notin X_S]\end{aligned} \tag{3–3}$$

*where*

$$\mathcal{L}_o(G, x) := (\mathcal{L}(G, x) \cap E^* E_o) \cup \{\epsilon\}$$

*is the set of strings generated from $x$ that end up with observable events including the empty string.*

Intuitively, $K$-step instant pre-opacity requires that, for any string $s$ generated from some initial state $x_0$ and any future instant $n \geq K$, there exists another observation-equivalent string $s'$ generated from some initial state $x_0'$ such that $s'$ can reach a non-secret state in exact $n$ steps. In other words, the intruder can never determine more than $K$ steps ahead, based on its current observation, that the system will visit a secret state at some future instant. Therefore, $K$ can be viewed as a parameter that determines *how early* the user does not want to reveal its intention. For instance, if $K = 2$, then the user may allow the intruder to determine just one step ahead that it will visit a secret system. We use the following example to illustrate this notion.

**Example 2.** *First, let us consider again system $G_1$ in Figure 3–1(a). One can easily check that this system is 1-step instant pre-opaque. For example, for string $a \in \mathcal{L}_o(G)$, the intruder cannot predict for sure that the system will be at a secret in one step since there exist another string $ua$ and its one-step extension $b$ such that $P(ua) = P(a)$ but $f(0, uab) = 5 \notin X_S$. Similarly, the intruder also cannot predict for sure that the system will be at a secret after 2 steps. For example, when observing $\epsilon$, the system may reach non-secret state 3 in two steps, which protects the possible secret intention of executing $ab$; when observing $a$, the system may reach non-secret state 6 in two step, which protects the possible secret intention of executing $uabc$.*

*However, for system $G_2$ in Figure 3–1(b), where $X_S = \{4, 7\}$ and $E_o = \{a, b, c\}$, one can check that this system is not 1-step instant pre-opaque. This is because, by observing $c$, the intruder can determine for sure that the system is either at state 2 or at state 5. However, from either state 2 or 5, the system will reach a secret state in the next step. Therefore, its intention of visiting secret states will be revealed one step before it actually happens.*

**Remark 1.** *In Definition 3, "step" is counted by the number of occurrences of* actual events *rather than the occurrences of observable events, i.e., we consider $|t| = n$ rather than $|P(t)| = n$. We believe this setting is more natural for predicting future instants. Furthermore, we consider string $s$ in $\mathcal{L}_o(G, x_0)$ rather than $\mathcal{L}(G, x_0)$. This implicitly assumes that the intruder will make a prediction immediately after observing a new observable event. Hereafter, we will introduce the main developments based on this setting.*

Note that $K$-step instant pre-opacity requires that the intruder cannot predict $K$-step ahead that the system will visit a secret state at some *specific instant*. This is also why we call it "instant" pre-opacity. However, in some situations, the intruder may just want to know whether

or not the system will visit a secret state in the future without the need of telling the specific instant. For instance, for $G_2$ in Figure 3–1(a), after observing $a$, although the intruder cannot determine for sure the specific instant when the secret state will be reached (the system will visit a secret state in one step or in two steps), it can still tell that the system will visit a secret state within the next two steps and at least one step before the occurrence of the first secret state. To capture this scenario, we propose the notion of $K$-step trajectory pre-opacity.

**Definition 4.** *($K$-Step Trajectory Pre-Opacity) Given system $G$, set of observable events $E_o$, set of secret states $X_S$, and non-negative integer $K \in \mathbb{N}$, system $G$ is said to be $K$-step trajectory pre-opaque (w.r.t. $E_o$ and $X_S$) if*

$$
\begin{aligned}
&(\forall x_0 \in X_0, \forall s \in \mathcal{L}_o(G, x_0))(\forall n \geq K) \\
&(\exists x_0' \in X_0, \exists s' \in \mathcal{L}_o(G, x_0'), \exists t_1 t_2 \in \mathcal{L}(G, f(x_0', s'))\ s.t. \\
&[P(s) = P(s')] \wedge [|t_1| = K] \wedge [|t_1 t_2| = n] \wedge \\
&\quad [\forall w \in \overline{\{t_2\}} : f(x_0', s' t_1 w) \notin X_S]
\end{aligned}
$$

Intuitively, $K$-step trajectory pre-opaque says that the intruder will never be able to determine $K$-step ahead for sure that the system will visit a secret state. More specifically, if a system is not $K$-step trajectory pre-opaque, then according to Definition 4, it means that there exist a string $s$ and an integer $n \geq K$ such that any observation equivalent string $s'$ must pass a secret state between the next $K$th instant and the next $n$th instant in the future. In other words, the intruder can determine the system's intention of visiting a secret state more than $K$-step ahead. We use the following example to illustrate this notion.

**Example 3.** *Let us consider again $G_1$ shown in Figure 3–1(a) and we have shown in Example 2 that this system is 1-step instant pre-opaque. However, it is not 1-step trajectory pre-opaque. For example, let us consider $\epsilon \in \mathcal{L}_o(G)$ and $n = 4 \geq 1 = K$. Note that $\epsilon$ itself is the only observation equivalent string in $\mathcal{L}_o(G)$. However, any 4-step extension of $\epsilon$, either $abca$ or $uabc$, will necessarily pass a secret state between the first instant and the forth instant. On the other hand, this system is 3-step trajectory pre-opaque. This is because the only instant to predict the visit of a secret state 3-step ahead is when observing $\epsilon$. However, with this observation, it is possible that the system will be at state 6, from which no secret state will be visited, after three steps. Therefore, the intruder can never determine 3-step ahead for sure that a secret state will be visited.*

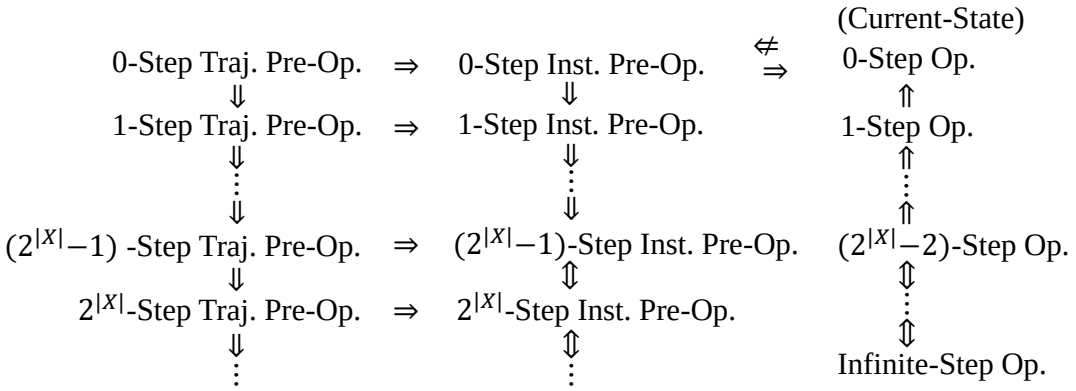



Figure 3–2 Relationships among different notions of opacity and pre-opacity.

**Remark 2.** *Similar to the interpretations of current-state opacity and $K$-step opacity, where the system is operating against the current-state estimator and delay-state estimator, respectively, here one can image that the system is operating against an intruder working as a* predictor *(for its secret intention). Roughly speaking, both $K$-step instant pre-opacity and $K$-step trajectory pre-opacity require that the intruder can never predict its secret $K$-steps ahead. However, the specific prediction tasks of the "virtual predictor" in these two notions are different: in instant pre-opacity, the predictor also needs to identify the precise future instant at which the system will be at a secret state, while in trajectory pre-opacity, the predictor just needs to identify the inevitability of passing through a secret state without specifying the visiting instant.*

### 3.3.2 Properties of Pre-Opacity

Now, we discuss properties of the proposed notions of pre-opacity and their relationships with other notions of opacity in the literature. First, we show that, for any $K$, $K$-step instant pre-opacity is weaker than $K$-step trajectory pre-opacity.

**Proposition 1.** *If $G$ is $K$-step trajectory pre-opaque, then it is $K$-step instant pre-opaque.*

***Proof.*** This result follows directly from the definitions. If the system is $K$-step trajectory pre-opaque, then by setting $t$ in Definition 3 as $t_1t_2$ in Definition 4, we know that the system is $K$-step instant pre-opacity. □

The intuition of the above result can also be interpreted as follows. For the case of instant pre-opacity, the prediction task of intruder is more challenging than that for the case of trajectory

opacity due to the need of determining the specific secret instant. Therefore, from the system's point of view, the underlying security property becomes weaker.

Also, by definitions, we note that $K$-step instant pre-opacity becomes weaker when $K$ increases, i.e., $K$-step instant pre-opacity always implies $(K+1)$-step instant pre-opacity. However, the following result shows that there is an upper bound for $K$ in instant pre-opacity, i.e., pre-opacity will not keep getting strictly weaker when $K$ increases.

To present our result, we introduce two necessary concepts. First, for each state $x \in X$, the set of states that can be reached from $x$ in exactly $K$ steps is define by

$$R_K(x) = \{x' \in X : \exists s \in \mathcal{L}(G,x) \text{ s.t. } f(x,s) = x' \wedge |s| = K\}. \tag{3–4}$$

For a set of states $q \subseteq X$, we also define $R_K(q) := \bigcup_{x\in q} R_K(x)$ as the set of states that can be reached from set $q$ in exactly $K$ steps.

Also, let $\alpha \in P(\mathcal{L}(G))$ be an observed string. Then the current-state estimate upon the occurrence of $\alpha$ without the unobservable tail is defined by

$$\hat{\mathcal{E}}(\alpha) = \{f(x_0,s) \in X : \exists x_0 \in X_0, s \in \mathcal{L}_o(G,x_0) \text{ s.t. } P(s) = \alpha\}. \tag{3–5}$$

Then we have the following the theorem showing the upper bound of $K$ in instant pre-opacity.

**Theorem 1.** *For any $K' > K \geq 2^{|X|} - 1$, system $G$ is $K'$-step instant pre-opaque, if and only if, $G$ is $K$-step instant pre-opaque.*

***Proof.*** It is trivial that $K$-step instant pre-opacity implies $K'$-step instant pre-opacity. Hereafter, we show that $K'$-step instant pre-opacity also implies $K$-step instant pre-opacity. Without loss of generality, we assume that $K' = K + 1$ as the argument can be applied inductively.

Now we assume, for the sake of contradiction, that $G$ is not $K$-step instant pre-opaque but $G$ is $(K+1)$-step instant pre-opaque, where $K \geq 2^{|X|} - 1$. This implies that there exists an initial state $x_0 \in X_0$ and a string $s \in \mathcal{L}_o(G,x_0)$ such that

$$\begin{aligned}&(\forall x_0' \in X_0)(\forall s' \in \mathcal{L}_o(G,x_0'), s't \in \mathcal{L}(G,x_0'))\\ &[P(s) = P(s') \wedge |t| = K] \Rightarrow [f(x_0', s't) \in X_S].\end{aligned}$$

Equivalently, we have $R_K(\hat{\mathcal{E}}(P(s))) \subseteq X_S$. Since for any $i \in \mathbb{N}$, $R_i(\hat{\mathcal{E}}(P(s)))$ is non-empty and it has at most $|X|$ elements, there are only $(2^{|X|} - 1)$ choices for $R_i(\hat{\mathcal{E}}(P(s)))$. Moreover, since the cardinality of multi-set $\{R_j(\hat{\mathcal{E}}(P(s))) : j = 0, 1, \cdots, K\}$ is $K + 1 \geq 2^{|X|} > 2^{|X|} - 1$,

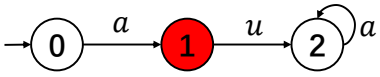



Figure 3–3 A system that is current-state opaque but is not 0-step instant pre-opaque, where $X_S = \{1\}$ and $E_{uo} = \{u\}$.

we know that there exist two integers $0 \leq m < n \leq K$, such that $R_m(\hat{\mathcal{E}}(P(s))) = R_n(\hat{\mathcal{E}}(P(s)))$. Then we know that

$$\begin{aligned} R_{K+n-m}(\hat{\mathcal{E}}(P(s))) &= R_{K-m}(R_n(\hat{\mathcal{E}}(P(s)))) \\ &= R_{K-m}(R_m(\hat{\mathcal{E}}(P(s)))) = R_K(\hat{\mathcal{E}}(P(s))) \subseteq X_S \end{aligned} \tag{3–6}$$

i.e., for initial state $x_0 \in X_0$ and string $s \in \mathcal{L}_o(G, x_0)$, we also have that

$$\begin{aligned} &(\forall x_0' \in X_0)(\forall s' \in \mathcal{L}_o(G, x_0'), s't \in \mathcal{L}(G, x_0')) \\ &[P(s) = P(s') \wedge |t| = K + n - m] \Rightarrow [f(x_0', s't) \in X_S]. \end{aligned}$$

Since $K + n - m \geq K + 1$, we know that $(K + 1)$-step instant pre-opacity is violated, which is a contradiction. □

One may conjecture that 0-step instant pre-opacity is equivalent to current-state opacity. However, it is not exactly the case. For example, let us consider system $G_3$ shown in Figure 3–3. This system is current-state opaque as the intruder cannot distinguish states 1 and 2 after observing $a$ due to unobservable event $u$. On the other hand, it is not 0-step instant pre-opaque according to our definition since the intruder can predict one step ahead for sure that the system will reach the secret state when observing nothing. This difference is due to the fact that we consider instant in terms of actual event steps rather than the observation steps. The only conclusion we can draw is that current-state opacity is weaker than 0-step instant pre-opacity, which is stated as follows.

**Proposition 2.** *If $G$ is 0-step instant/trajectory pre-opaque, then $G$ is current-state opaque.*

***Proof.*** It suffices to show that 0-step instant pre-opacity implies current-state opacity since 0-step trajectory pre-opacity is stronger than 0-step instant pre-opacity. Suppose $G$ is 0-step instant pre-opaque. Let us consider arbitrary initial state $x_0 \in X_0$ and string $s \in \mathcal{L}(G, x_0)$. Note that for string $s$, we can always find $\hat{s} \in \mathcal{L}_o(G, x_0)$, by removing the unobservable tail (if any) of $s$ such that $P(s) = P(\hat{s})$. Since $G$ is 0-step instant pre-opaque, by setting $n$ in Definition 3 as

$n = 0$, we know that there exist $x_0' \in X_0$ and $s' \in \mathcal{L}_o(G, x_0'))$ such that $P(s') = P(\hat{s}) = P(s)$ and $f(x_0', s') \notin X_S$. This means that the system is current-state opaque. □

Based on the above discussion, we summarize the relationships among the proposed notions of pre-opacity and existing notions of opacity in Figure 3–2.

**Remark 3.** *Finally, we note that the proposed concept of pre-opacity is also related to the notion of fault predictability or fault prognosability in the literature [69-71], which captures whether or not a fault event can always be predicted unambiguously a certain number of steps ahead before it actually occurs. Conceptually, by considering the visit of secret states as fault, trajectory pre-opacity can be viewed as a dual problem of predictability. However, trajectory pre-opacity is not exactly non-predictability. The former requires that* all *secret paths cannot be predicted, while the latter says* some *fault path cannot be predicted. Furthermore, our notion of instant opacity is quite different from predictability as we need to determine the specific instant of being secret; this issue does not occur in predictability analysis.*

## 3.4 Verification of Pre-Opacity

In this section, we show how to verify the proposed notions of pre-opacity. Specifically, we present two state-based necessary and sufficient conditions for $K$-step instant pre-opacity and $K$-step trajectory pre-opacity, respectively, that can be checked using the observer structure. Then we discuss the complexity of the verification problems.

### 3.4.1 Necessary and Sufficient Condition for Instant Pre-Opacity

Recall that a system is not $K$-step instant pre-opaque if after some observation, each possible state (immediately after the observation) will visit a secret state in exactly $n$ steps for some $n \geq K$. This suggests that $K$-step instant pre-opacity can be checked by combining the current-state estimation together with the reachability analysis. To this end, we further introduce some necessary notions.

We say that a state $x \in X$ is a *$K$-step indicator* state if it will reach a secret state inevitably in exactly $K$ steps, i.e.,

$$R_K(x) \subseteq X_S.$$

For any $K \in \mathbb{N}$, we define

$$\Im_K := \{x \in X : R_K(x) \subseteq X_S\} \subseteq X$$

as the set of $K$-step indicator states.

Then the following theorem shows that $K$-step instant pre-opacity can be simply characterized in terms of current-state estimate and $K$-step indicator states.

**Theorem 2.** *System $G$ is $K$-step instant pre-opaque if and only if*

$$\forall \alpha \in P(\mathcal{L}(G)), \forall n \geq K : \hat{\mathcal{E}}(\alpha) \not\subseteq \mathfrak{I}_n.$$

***Proof.*** ($\Rightarrow$) By contraposition. Suppose that there exists a string $\alpha \in P(\mathcal{L}(G))$ and an integer $n \geq K$ such that $\hat{\mathcal{E}}(\alpha) \subseteq \mathfrak{I}_n$. Let us consider an initial state $x_0 \in X_0$ and a string $s \in \mathcal{L}_o(G, x_0)$ such that $P(s) = \alpha$. Since $\hat{\mathcal{E}}(\alpha) \subseteq \mathfrak{I}_n$, for any initial state $x_0' \in X_0$ and string $s' \in \mathcal{L}_o(G, x_0)$ such that $P(s') = \alpha$, we have $f(x_0', s') \in \mathfrak{I}_n$, i.e., $R_n(f(x_0', s')) \subseteq X_S$ This means that for for any $t \in \mathcal{L}(G, f(x_0', s'))$ and $|t| = n$, we have $f(x_0', s't) \in X_S$. This means that system $G$ is not $K$-step instant pre-opaque.

($\Leftarrow$) Still by contraposition. Suppose that $G$ is not $K$-step instant pre-opaque, which means that there exists an initial state $x_0 \in X_0$, a string $s \in \mathcal{L}_o(G, x_0)$ and an integer $n \geq K$ such that

$$(\forall x_0' \in X_0)(\forall s' \in \mathcal{L}_o(G, x_0'), s't \in \mathcal{L}(G, x_0'))$$
$$[P(s) = P(s') \wedge |t| = n \Rightarrow f(x_0', s't) \in X_S]$$

Then let $\alpha = P(s)$. Clearly, we have $R_n(\hat{\mathcal{E}}(\alpha)) \subseteq X_S$, i.e., $\hat{\mathcal{E}}(\alpha) \subseteq \mathfrak{I}_n$. This violates the condition in the theorem. □

Theorem 2 essentially provides a state-based characterization of the language-based definition of $K$-step instant pre-opacity. However, it still cannot be directly used for the verification of instant pre-opacity. The main issue is that we need to check whether or not $\hat{\mathcal{E}}(\alpha) \not\subseteq \mathfrak{I}_n$ for any $n \geq K$, which has infinite number of instants. The following result further generalizes Theorem 2 and shows that it suffices to check $\hat{\mathcal{E}}(\alpha) \not\subseteq \mathfrak{I}_n$ for a bounded number of instants.

**Proposition 3.** *For any $\alpha \in P(\mathcal{L}(G))$, the following two statements are equivalent:*

*(i)* $\forall n \geq K : \hat{\mathcal{E}}(\alpha) \not\subseteq \mathfrak{I}_n$;

*(ii)* $\forall n \in \{K, K+1, \ldots, K + 2^{|X|} - 1\} : \hat{\mathcal{E}}(\alpha) \not\subseteq \mathfrak{I}_n$.

***Proof.*** (i)$\Rightarrow$(ii) is trivial. Hereafter, we show that (ii)$\Rightarrow$(i). Let $q := \hat{\mathcal{E}}(\alpha)$ and we consider the reachable set of $q$ for each instant between $K$ and $K + 2^{|X|} - 1$, i.e., $R_K(q), R_{K+1}(q), \ldots, R_{K+2^{|X|}-1}(q)$. For any $n \in \{K, \ldots, K + 2^{|X|} - 1\}$, since $q \not\subseteq \mathfrak{I}_n$, we

know that there exists $x \in q$ such that $x \notin \mathfrak{I}_n$, i.e., $R_n(x) \not\subseteq X_S$. Since $R_n(q) = \bigcup_{x\in q} R_n(x)$, we know that $R_n(q) \not\subseteq X_S$ for any $n \in \{K, \ldots, K + 2^{|X|} - 1\}$.

Now we note that set $\{R_i(q) : K \leq i \leq K + 2^{|X|} - 1\} \subseteq 2^X$ is non-empty, so it contains at most $2^{|X|}$ elements. Therefore, there must exist two instants $K \leq i < j \leq K + 2^{|X|} - 1$ such that $R_i(q) = R_j(q)$. Furthermore, by the definition of $K$-step reachable set, we have

$$R_{n+k}(q) = R_n(R_k(q)) = R_k(R_n(q))$$

Then for any instant $n' > K + 2^{|X|} - 1$, we can always write it in the form of

$$n' = i + (j - i) \times k + m$$

where $1 \leq k, 0 \leq m < (j - i)$ are two integers. Furthermore, since $R_i(q) = R_j(q)$, we have $R_i(q) = R_{i+(j-i)\times k}(q)$ for any $k \geq 0$, so

$$R_{n'}(q) = R_m(R_{i+(j-i)\times k}(q)) = R_{m+i}(q).$$

However, since $m < j - i$, we have $m + i < j$. Therefore,

$$\forall n' > K + 2^{|X|} - 1 : R_{n'}(q) = R_{m+i}(q) \not\subseteq X_S.$$

This further implies that

$$\forall n' > K + 2^{|X|} - 1 : q \notin \mathfrak{I}_{n'},$$

which completes the proof. □

One may ask why we need to search for the entire next $2^{|X|}$ instants to obtain the upper bound in Proposition 3. However, this upper bound seems to be unavoidable. To see this, let us consider the system shown in Figure 3–4, where all events are unobservable and red states denote secret states. This system is not $K$-step instant pre-opaque for any $K$ since one can determine for sure (by observing nothing) that the system will be at a secret state for instants $k \cdot 30, k = 1, 2 \ldots$, where 30 is the least common multiple of cycle lengths 2, 3 and 5. Therefore, the first violation of $\hat{\mathcal{E}}(\alpha) \not\subseteq \mathfrak{I}_n$ occurs at $n = 30$. Similarly, one could add more states to create more such cycles and the upper bound for searching $\mathfrak{I}_n$ will grow exponentially. However, this exponentially searching bound is only needed when the system contains unobservable events. In the following result, we show that such an upper bounded search can be avoided for the extreme case when there is no unobservable event in the system.

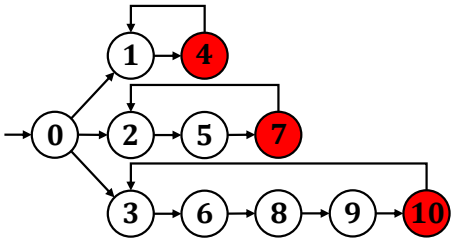



Figure 3–4 A system where all events are unobservable and red states denote secret states.

**Proposition 4.** *Under the assumption that all events in $G$ are observable, then $G$ is $K$-step instant pre-opaque if and only if*

$$\forall \alpha \in P(\mathcal{L}(G)) : \hat{\mathcal{E}}(\alpha) \not\subseteq \mathfrak{I}_K.$$

***Proof.*** The necessity follows directly from Theorem 2. To show the sufficiency, suppose that $\forall \alpha \in P(\mathcal{L}(G)) : \hat{\mathcal{E}}(\alpha) \not\subseteq \mathfrak{I}_K$ and assume that $G$ is not $K$-step instant pre-opaque. Then by Theorem 2, we know that there exist $\alpha \in P(\mathcal{L}(G))$ and $n > K$ such that $\hat{\mathcal{E}}(\alpha) \subseteq \mathfrak{I}_n$. In other words, we have that

$$\begin{aligned}&(\forall x_0 \in X_0)(\forall s \in \mathcal{L}_o(G, x_0), st \in \mathcal{L}(G, x_0))\\ &[P(s) = \alpha \wedge |t| = n] \Rightarrow [f(x_0, st) \in X_S]\end{aligned}$$

For any $t$ satisfying above condition, we let $t = t_1 t_2$, where $|t_1| = n - K$ and $|t_2| = K$. Then we know that $R_K(\hat{\mathcal{E}}(P(st_1))) \subseteq X_S$, i.e., $\hat{\mathcal{E}}(P(st_1)) \subseteq \mathfrak{I}_K$, which is a contradiction. □

### 3.4.2 Necessary and Sufficient Condition for Trajectory Pre-Opacity

Now we discuss the case of $K$-step trajectory pre-opacity. First, we say that a state $x \in X$ is a *non-indicator state* if there exists an infinitely long string defined from this state along which no secret state is visited. Formally, we define the set of non-indicator states by

$$\mathcal{N} := \left\{ x \in X : \begin{array}{c}(\forall n \geq 0)(\exists s \in \mathcal{L}(G, x) : |s| > n)\\ (\forall t \in \overline{\{s\}})[f(x, t) \notin X_S]\end{array} \right\} \tag{3–7}$$

Since the number of states in $G$ is finite, a state is a non-indicator state if and only it can reach a cycle, in which all states are non-secret, via a sequence of non-secret states. In other words, if a state is not in $\mathcal{N}$, then it is an indicator state in the sense that a secret state will be visited inevitably from this state.

**Remark 4.** *Note that a state is not a non-indicator state does not necessarily imply that it is a K-step indicator state for some K since the latter requires the system to be at a secret state for some specific instant while the former does not require this information. Furthermore, a state is an indicator state does not imply that any state reachable from this state is an indicator state. This is because after passing through a secret state, the status of indicating may become non-indicating.*

Therefore, if the system is at a state whose $K$-step reachable set is a subset of indicator state, then based on this state information, one can predict $K$-step ahead that a secret state will be visited. We define

$$\mathcal{N}_K := \{x \in X : R_K(x) \cap \mathcal{N} \neq \emptyset\} \subseteq X$$

as the set of states the intruder cannot make such a prediction. Then we have the following theorem.

**Theorem 3.** *System $G$ is $K$-step trajectory pre-opaque if and only if*

$$\forall \alpha \in P(\mathcal{L}(G)) : \hat{\mathcal{E}}(\alpha) \cap \mathcal{N}_K \neq \emptyset.$$

***Proof.*** ($\Rightarrow$) By contradiction. Suppose that $G$ is $K$-step trajectory pre-opaque and assume that there exists a string $\alpha \in P(\mathcal{L}(G))$ such that $\hat{\mathcal{E}}(\alpha) \cap \mathcal{N}_K = \emptyset$ holds. According to Definition 4, for any $n \geq K$, there exists $x_0' \in X_0, s' \in \mathcal{L}_o(G, x_0'), s't_1t_2 \in \mathcal{L}(G, x_0')$ such that $P(s') = P(s) = \alpha, |t_1| = K$, $|t_2| = n - K$ and for any $w \in \overline{\{t_2\}}$, we have $f(x_0', s't_1w) \notin X_S$. Now, let us choose $n$ such that $n \geq |X| + K$, i.e., $|t_2| \geq |X|$. Since $f(x_0', s't_1t_2)$ can pass through at most $|X|$ states, there are at least two repeated states that forms a cycle along the path of $t_2$ starting from $f(x_0', s't_1)$. This immediately implies that $f(x_0', s't_1) \in \mathcal{N}$. Furthermore, we have $f(x_0', s't_1) \in R_K(f(x_0', s'))$ since $|t_1| = K$. Therefore, we have $R_K(f(x_0', s')) \cap \mathcal{N} \neq \emptyset$, i.e., $f(x_0', s') \in \mathcal{N}_K$. Since $P(s') = \alpha$ and $s' \in E^*E_o \cup \{\epsilon\}$, we have $f(x_0', s') \in \hat{\mathcal{E}}(\alpha)$, which implies that $\hat{\mathcal{E}}(\alpha) \cap \mathcal{N}_K \neq \emptyset$. This, however, contradicts to our assumption.

($\Leftarrow$) By contradiction. Suppose that for any $\alpha \in P(\mathcal{L}(G))$, we have $\hat{\mathcal{E}}(\alpha) \cap \mathcal{N}_K \neq \emptyset$ and assume that $G$ is not $K$-step trajectory pre-opaque, i.e., there exist a state $x_0 \in X_0$, a string $s \in \mathcal{L}_o(G, x_0)$ and an integer $n \geq K$ such that

$$\begin{aligned}
&(\forall x_0' \in X_0)(\forall s' \in \mathcal{L}_o(G, x_0'), \forall t_1t_2 \in \mathcal{L}(G, f(x_0', s')) \text{ s.t.} \\
&[P(s) = P(s') \wedge |t_1| = K \wedge [|t_1t_2| = n] \\
&\Rightarrow [\exists w \in \overline{\{t_2\}} : f(x_0', s't_1w) \in X_S]
\end{aligned} \tag{3–8}$$

Let us consider an arbitrary state $x$ in $\hat{\mathcal{E}}(P(s))$. This means that there exist a state $x_0' \in X_0$ and a string $s' \in \mathcal{L}_o(G, x_0')$ such that $f(x_0', s') = x$ and $P(s') = P(s)$. However, according to Equation (3–8), any string of length $n$ from state $x$ must pass through a secret state between its $K$th instant and its $n$th instant. This means that $R_K(x) \cap \mathcal{N} = \emptyset$, i.e., $x \notin \mathcal{N}_K$. Note that $x$ is an arbitrary state in $\hat{\mathcal{E}}(P(s))$. Therefore, we have $\hat{\mathcal{E}}(P(s)) \cap \mathcal{N}_K = \emptyset$. However, this contradicts to our assumption that $\hat{\mathcal{E}}(\alpha) \cap \mathcal{N}_K \neq \emptyset$ for any $\alpha \in P(\mathcal{L}(G))$. □

### 3.4.3 Verification Algorithms

Now, let us discuss how to use the derived necessary and sufficient conditions to verify $K$-step instant or trajectory pre-opacity. To this end, we need to compute

- All possible state estimates, i.e., $\{\hat{\mathcal{E}}(\alpha) : \alpha \in P(\mathcal{L}(G))\}$;
- A set of $n$-step indicator states for $K \leq n \leq K + 2^{|X|} - 1$, i.e., $\{\mathfrak{I}_K, \ldots, \mathfrak{I}_{K+2^{|X|}-1}\}$ (for instant pre-opacity);
- The set of states whose $K$-step reachable set contains at least a non-indicator state, i.e., $\mathcal{N}_K$ (for trajectory pre-opacity).

#### 3.4.3.1 Computation of $\hat{\mathcal{E}}(\alpha)$

Note that, compared with the standard current-state estimate, the state estimate considered here does not contain the unobservable tail. This can be computed by a slightly modified version of the standard observer automaton (we still call it observer here for the sake of simplicity). Formally, the observer of $G$ is a new DFA

$$Obs(G) = (Q_{obs}, E_o, f_{obs}, q_{obs,0}),$$

where $Q_{obs} \subseteq 2^X \setminus \emptyset$ is the set of states, $E_o$ is the set of events, $q_{obs,0} = X_0$ is the initial state, and $f_{obs} : Q_{obs} \times E_o \to Q_{obs}$ is the deterministic transition function defined by: for any $q \in Q_{obs}, \sigma \in E_o$, we have

$$f_{obs}(q, \sigma) = \{x \in X : \exists x' \in q, w \in E_{uo}^* \text{ s.t. } f(x', w\sigma) = x\} \tag{3–9}$$

For the sake of simplicity, we only consider the reachable part of the observer. Then we have

$$\forall \alpha \in P(\mathcal{L}(G)) : f(q_{obs,0}, \alpha) = \hat{\mathcal{E}}(\alpha).$$

Therefore, all possible state estimate $\hat{\mathcal{E}}(\alpha)$ can be computed with complexity $O(|E_o|2^{|X|})$.

#### 3.4.3.2 Computation of $\mathfrak{I}_n$

For any give $n \geq 0$, one can compute $\mathfrak{I}_n$ by backtracking $n$ steps from the set of all secret states. Formally, one can define an operator $F : 2^X \to 2^X$ by: for any $q \in 2^X$, we have

$$F(q) = \{x \in X : \forall \sigma \in E, f(x, \sigma)! \text{ s.t. } f(x, \sigma) \in q\}. \tag{3–10}$$

Then one can easily check that

$$\mathfrak{I}_n = F^n(\mathfrak{I}_0) \text{ with } \mathfrak{I}_0 = X_S$$

which can be computed with complexity $O(n|E_o||X|)$.

#### 3.4.3.3 Computation of $\mathcal{N}_K$

To compute $\mathcal{N}_K$, first we need to compute the set of non-indicator states $\mathcal{N}$. To this end, we can remove all secret states in $G$ and compute all strongly connected components, i.e., cycles; this can be done by, e.g., Kosaraju's algorithm with a linear complexity in the size of $G$ [75]. Then those states that can reach a non-secret cycle are the set of non-indicator states. Therefore, computing set $\mathcal{N}$ can be done in $O(|E||X|)$. In order to compute $\mathcal{N}_K$, one can backtrack from $\mathcal{N}$ using another operator $W : 2^X \to 2^X$ defined by: for any $q \in 2^X$, we have

$$W(q) = \{x \in X : \exists \sigma \in E \text{ s.t. } f(x, \sigma) \in q\}. \tag{3–11}$$

Then one can easily check that

$$\mathcal{N}_K = W^K(\mathcal{N}_0) \text{ with } \mathcal{N}_0 = \mathcal{N}$$

which can be computed with complexity $O(K|E_o||X|)$. Therefore, the overall complexity for computing set $\mathcal{N}_K$ is $O(K|E_o||X|)$.

Based on the above discussions, we summarize the algorithms for the verification of $K$-step instant pre-opacity and $K$-step trajectory pre-opacity by Algorithm Ins-Pre-Opa-Ver and Algorithm Traj-Pre-Opa-Ver, respectively. The complexity of Algorithm Ins-Pre-Opa-Ver is $O(|E_o|2^{|X|}[K + (K+1) + \cdots + (K + 2^{|X|} - 1)]|E_o||X|) = O(|E_o|^2|X|(K + 2^{|X|-1})2^{2|X|})$ for the general case and is $O(K|E_o|^2|X|2^{|X|})$ under the assumption that there is no unobservable event. The complexity of Algorithm Traj-Pre-Opa-Ver is simply $O(K|E_o|^2|X|2^{|X|})$, which is dominated by the size of the observer. We illustrate the verification algorithms by the following examples.

**Algorithm 3–1** INS-PRE-OPA-VER

**Input:** System $G$ with $X_S$, $E_o$ and $K$

**Output:** YES or NO

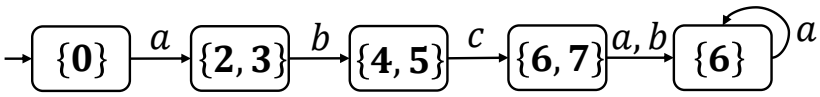
Construct the observer $obs(G)$
    there is no unobservable event in $G$ $M \leftarrow 0$ $M \leftarrow 2^{|X|} - 1$
**if** there is no unobservable event in $G$ **then**
    $M \leftarrow 0$
**else**
    $M \leftarrow 2^{|X|} - 1$
**end if**
**for** $q \in Q_{obs}$ **do**
    **for** $n \in \{K, K+1, \ldots, K+M\}$ **do**
        **if** $q \subseteq \Im_n$ **then**
            **return** NO
        **end if**
    **end for**
**end for**
**return** YES

**Algorithm 3–2** TRAJ-PRE-OPA-VER

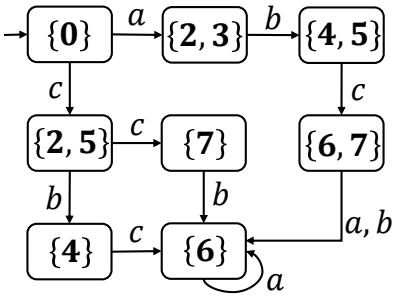
**Input:** System $G$ with $X_S$, $E_o$ and $K$

**Output:** YES or NO

Construct the observer $obs(G)$
**for** $q \in Q_{obs}$ **do**
    **if** $q \cap \mathcal{N}_K = \emptyset$ **then**
        **return** NO
    **end if**
**end for**
**return** YES




(a) $Obs(G_1)$




(b) $Obs(G_2)$

Figure 3–5 Observers for $G_1$ and $G_2$, respectively.

**Example 4.** *Let us consider again system $G_1$ shown in Figure 3–1(a) and we verify whether or not it is $K$-step trajectory pre-opaque. First, we build its observer $Obs(G_1)$ as shown in Figure 3–5(a). The only non-indicator state is 6, i.e., $\mathcal{N} = \{6\}$. For $K = 2$, we have $\mathcal{N}_2 = W^2(\{6\}) = \{2, 4, 5, 6, 7\}$. Since $\{0\} \cap \{2, 4, 5, 6, 7\} = \emptyset$, we know that $G_1$ is not 2-step trajectory pre-opaque. However, for $K = 3$, we have $\mathcal{N}_3 = W^3(\{6\}) = \{0, 2, 3, 4, 5, 6, 7\}$ and each observer state has a common element with $\mathcal{N}_3$. Therefore, $G_1$ is 3-step trajectory pre-opaque. These results are also consistent with our previous intuitive analysis.*

*However, this system is $K$-step instant pre-opaque for any $K \geq 0$. To see this, it suffices to consider the case of $K = 0$. In this case, we have*

$$\mathfrak{I}_0 = \{4, 7\}, \mathfrak{I}_1 = \{2, 5\}, \mathfrak{I}_2 = \{3\}, \mathfrak{I}_3 = \{1\},$$
$$\mathfrak{I}_4 = \mathfrak{I}_5 = \cdots = \emptyset$$

*Therefore, no observer state is a subset of any $\mathfrak{I}_i$, which implies 0-step instant pre-opacity.*

**Example 5.** *For system $G_2$ shown in Figure 3–1(b), its observer is shown in Figure 3–5(a). Then we have*

$$\mathfrak{I}_0 = \{4, 7\}, \mathfrak{I}_1 = \{2, 5\}, \mathfrak{I}_2 = \{3\}, \mathfrak{I}_3 = \mathfrak{I}_4 = \cdots = \emptyset$$

*However, for observer state* $\{2,5\}$*, we have* $\{2,5\} \subseteq \Im_1$*, which means that* $G_2$ *is not* 1*-step instant pre-opaque. On the other hand,* $G_2$ *is* $K$*-step instant pre-opaque for any* $K \geq 2$ *as no state in* $Obs(G_2)$ *is a subset of any* $\Im_n, n \geq 2$*.*

### 3.4.4 The Complexity of $K$-Step Pre-Opacity

Note that the complexity of Algorithm Ins-Pre-Opa-Ver and Algorithm Traj-Pre-Opa-Ver are both exponential in the number of states in $G$. Next, we show that both properties are essentially PSPACE-hard; therefore, the exponential complexity seems to be unavoidable.

**Theorem 4.** *Deciding whether or not* $G$ *is* $K$*-step instant (or trajectory) pre-opaque is PSPACE-hard even when* $G$ *is deterministic.*

***Proof.*** Given two non-deterministic automata (NFAs) $G_1 = (X_1, E, f_1, X_{1,0})$ and $G_2 = (X_2, E, f_2, X_{2,0})$, the problem of language containment asks to decide whether or not $\mathcal{L}(G_1) \subseteq \mathcal{L}(G_2)$. This problem is known to be PSPACE-hard. Hereafter, we show that checking $K$-step instant/trajectory pre-opacity is also PSPACE-hard by reducing the language containment problem to the pre-opacity verification problem.

Let $G_1 = (X_1, E, f_1, X_{1,0})$ and $G_2 = (X_2, E, f_2, X_{2,0})$ be two NFAs with initial states $X_{1,0}$ and $X_{2,0}$, respectively. Without loss of generality, we assume $G_1$ and $G_2$ are live; otherwise, we can add a self-loop with a new event at each state in $G_1$ and $G_2$. Note that, in the analysis of pre-opacity, we assume that the transition function is deterministic; this gap can be bridged by using unobservable events to mimic non-determinism. Formally, let $E_u = \{u_1, u_2, \ldots, u_k\}$ be a set of new unobservable events. Then for each NFA $G_i$, we construct a new DFA $\tilde{G}_i = (\tilde{X}_i, \tilde{E}, \tilde{f}_i, \tilde{X}_{i,0})$ by: $\tilde{X}_i = X_i \cup \{(x,\sigma) \in X_i \times E : f_i(x,\sigma)!\}$, $\tilde{E} = E \cup E_u$, $X_{i,0} = \tilde{X}_{i,0}$, and $\tilde{f}_i : \tilde{X}_i \times \tilde{E} \to \tilde{X}_i$ is the deterministic transition function defined by: for any $f_i(x,\sigma)!$, we have $\tilde{f}_i(x,\sigma) = (x,\sigma)$ and $f_i(x,\sigma) = \{\tilde{f}_i((x,\sigma),u) : u \in E_u\}$. The construction of $\tilde{G}_i$ is illustrated by Figure 3–6. Clearly, one has $\mathcal{L}(G_1) \subseteq \mathcal{L}(G_2)$ iff $P(\mathcal{L}(\tilde{G}_1)) \subseteq P(\mathcal{L}(\tilde{G}_2))$.

Now we construct a new DFA $\tilde{G} = (\tilde{X}, \tilde{E}, \tilde{f}, \tilde{X}_0)$ by taking the union of $\tilde{G}_1$ and $\tilde{G}_2$, i.e., $\tilde{X} = \tilde{X}_1 \cup \tilde{X}_2$, $\tilde{f}$ is consistent with $\tilde{f}_1$ and $\tilde{f}_2$, and $\tilde{X}_0 = \tilde{X}_{1,0} \cup \tilde{X}_{2,0}$. Then, for system $\tilde{G}$, we let $X_S = X_1$ and $E_u$ be the set of unobservable events. We show that $\tilde{G}$ is 0-step instant (or trajectory) pre-opaque if and only if $\mathcal{L}(G_1) \subseteq \mathcal{L}(G_2)$.

($\Rightarrow$) To see this, we suppose that $\mathcal{L}(G_1) \not\subseteq \mathcal{L}(G_2)$, then we know that there exists a string $s \in \mathcal{L}(G_1) \setminus \mathcal{L}(G_2)$, i.e., there exists a string $t \in P(\mathcal{L}(\tilde{G}_1)) \setminus P(\mathcal{L}(\tilde{G}_2))$, since $\mathcal{L}(G_1) \subseteq \mathcal{L}(G_2)$ is equivalent to $P(\mathcal{L}(\tilde{G}_1)) \subseteq P(\mathcal{L}(\tilde{G}_2))$. Therefore, after observing $t$ in $\tilde{G}$, since $X_S = X_1$, we

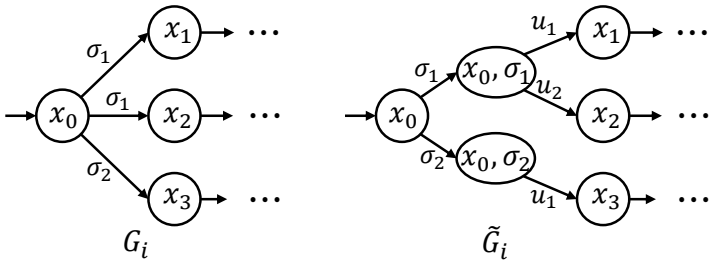



Figure 3–6 Conceptual illustration of how to construct $\tilde{G}_i$ from $G_i$

.

know for sure that the system now is at a secret state and will be at secret states for any future instant. Hence, $\tilde{G}$ is not 0-step instant (or trajectory) pre-opaque.

($\Leftarrow$) Suppose that $\mathcal{L}(G_1) \subseteq \mathcal{L}(G_2)$ and we assume that, for the sake of contradiction, $\tilde{G}$ is not 0-step trajectory pre-opaque, which means it is also not 0-step instant pre-opaque. Then we know that there exists a string $s \in P(\mathcal{L}(\tilde{G}))$ such that $\hat{\mathcal{E}}(s) \cap \mathcal{N}_0 = \emptyset$. Note that we have $P(\mathcal{L}(\tilde{G}_1)) \subseteq P(\mathcal{L}(\tilde{G}_2))$. Since $\mathcal{L}(G_1) \subseteq \mathcal{L}(G_2)$, this also implies that $P(\mathcal{L}(\tilde{G})) = P(\mathcal{L}(\tilde{G}_2))$ and $s \in P(\mathcal{L}(\tilde{G}_2))$. However, since every state in $\tilde{G}_2$ is non-secret and $\tilde{G}_2$ is live, we have $\tilde{X}_2 \subseteq \mathcal{N}_0$. Therefore, it is not possible that $\hat{\mathcal{E}}(s) \cap \mathcal{N}_0 = \emptyset$, which is a contradiction.

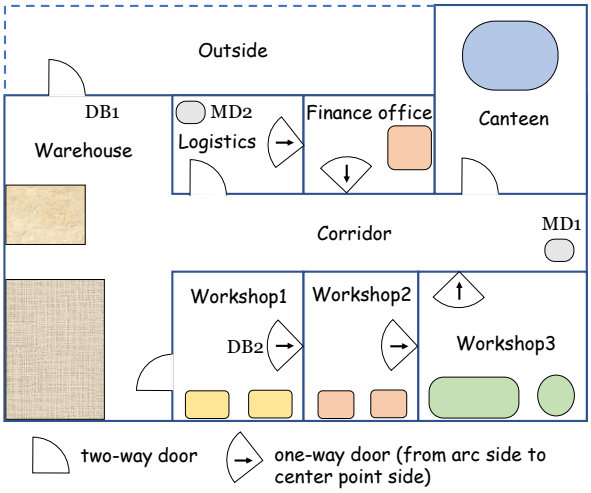
Overall, we conclude that deciding whether or not $G$ is $K$-step instant/trajectory pre-opacity is PSPACE-hard. □

## 3.5 Secret Intention as a Sequence Pattern

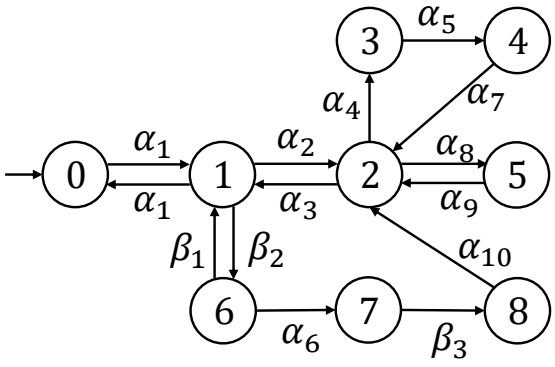
In the previous sections, we model the secret intention of the system as the willing to reach some secret states. In this section, we further generalize this setting by considering the secret intention as the willing to execute some particular sequences of events, which we call a *sequence pattern*. We present an illustrative example that motivates the definition of pattern pre-opacity and show how it can be reduced to state-based pre-opacity.

### 3.5.1 Illustrative Example of Pattern Pre-Opacity

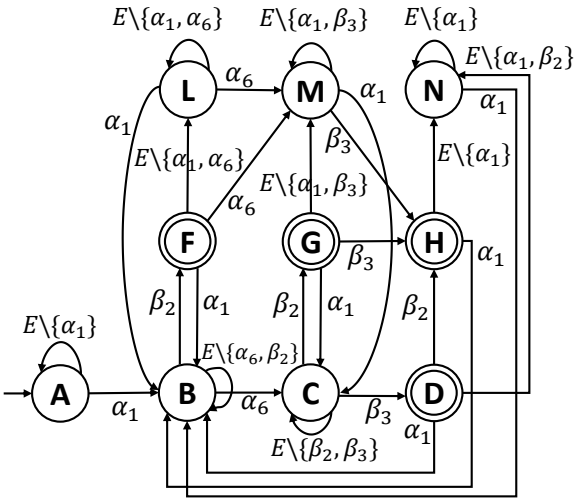
We consider a location-tracking/prediction type problem in a smart factory building equipped with sensors as shown in Figure 3–7(a). The factory has eight regions of interest: a *warehouse*, a *logistics*, a *finance office*, a *canteen*, a *corridor* and three *workshops*. We assume there is a person in the factory that can move from one region to another by passing through a door;




(a) The topology of a factory




(b) The specification automaton $G_3$




(c) $G_\Omega$

Figure 3–7 An illustrative case of pattern pre-opacity.

some doors are one-way and some are two-way as depicted in the figure. In particular, there are two doors $DB_1$ and $DB_2$ secured by door barrier sensors, which allow to observe if a person enters the corresponding rooms. Furthermore, there are two additional motion detector sensors ($MD_1$ and $MD_2$) at corridor and logistics, respectively; they can detect if a person moves to the corridor (or logistics). The building monitor is able to use these sensors to track and predict the behavior of the person.

According to the structure of factory and different types of doors, the overall system, which is the mobility of the person, can be modeled as a DES as shown in Figure 3–7(b), where states 0 to 8 represent, respectively, regions *outside*, *warehouse*, *corridor*, *logistics*, *finance office*, *canteen*, *workshop 1*, *workshop 2* and *workshop 3*. Based on the distribution of motion detector sensors and door barrier sensors, we know that the set of observable events is

$$E_o = \{\alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5, \alpha_6, \alpha_7, \alpha_8, \alpha_9, \alpha_{10}\}.$$

Now we assume that the person wants to move in the factory to complete two tasks "secretly": (*task 1*) first goes to *warehouse* and then goes to *workshop 1*; (*task 2*) first goes to *warehouse*, and then enters *workshop 2*, and finally gets to *workshop 3*. Furthermore, the person wants to hide its intention for executing the above sequences against the monitor before they are completed. In this setting, "secret intention" is no longer visiting a secret state in the future. Instead, completing any sequence containing string $\alpha_1\beta_2$ or $\alpha_1\alpha_6\beta_3$ can be viewed as a secret behavior. One can check that the person may not be able to hide its intention to complete task 2 more than one step before its completion. This is because once motion detector sensor $DB_2$ is triggered, the building monitor can determine for sure that the person was from *warehouse*, and is currently at *workshop 2* and will go to *workshop 3* in one step to complete the task. To formally describe this scenario, we propose $K$-step instant/trajectory pattern pre-opacity in the next part.

### 3.5.2 Definitions of Pattern Pre-Opacity

Now, we formally formulate the notion of pattern pre-opacity. Specifically, we consider a *sequence pattern* modeled as a regular language $\Omega \subseteq E^*$ in order to describe the secret behaviors of the system. Then we say that a system is $K$-step pattern pre-opaque if any completion of a string in the pattern can be predicted $K$-step ahead. Depending on whether or not the intruder needs to determine the specific instant of the completion, pattern pre-opacity can also be categorized as instant pre-opacity and trajectory pre-opacity.

**Definition 5.** *(K-Step Instant Pattern Pre-Opacity) Given system G, set of observable events $E_o$, sequence pattern $\Omega$, and non-negative integer $K \in \mathbb{N}$, system G is said to be K-step instant pattern pre-opaque (w.r.t. $E_o$ and $\Omega$) if*

$$\begin{aligned}&(\forall x_0 \in X_0)(\forall s \in \mathcal{L}_o(G, x_0))(\forall n \geq K)\\&(\exists x_0' \in X_0)(\exists s' \in \mathcal{L}_o(G, x_0'), t \in \mathcal{L}(G, f(x_0', s'))\\&[P(s) = P(s') \wedge |t| = n \wedge s't \notin \Omega]\end{aligned} \tag{3–12}$$

**Definition 6.** *(K-Step Trajectory Pattern Pre-Opacity) Given system G, set of observable events $E_o$, a sequence pattern $\Omega$, and non-negative integer $K \in \mathbb{N}$, system G is said to be K-step trajectory pattern pre-opaque (w.r.t. $E_o$ and $\Omega$) if*

$$\begin{aligned}&(\forall x_0 \in X_0, \forall s \in \mathcal{L}_o(G, x_0))(\forall n \geq K)\\&(\exists x_0' \in X_0, \exists s' \in \mathcal{L}_o(G, x_0'), \exists t_1 t_2 \in \mathcal{L}(G, f(x_0', s'))\ s.t.\\&[P(s) = P(s')] \wedge [|t_1| = K] \wedge [|t_1 t_2| = n] \wedge\\&\quad [\forall w \in \overline{\{t_2\}} : s' t_1 w \notin \Omega]\end{aligned}$$

Intuitively, $K$-step trajectory pattern pre-opacity says that, for any observation, the intruder cannot predict $K$-step ahead that a secret sequence will be completed. The definition of instant pattern pre-opacity is similar; the only difference is that it also requires to specify the specific instant of the completion. Clearly, pre-opacity is a special case of pattern pre-opacity as we can define all sequences reaching secret states as the sequence pattern. Hereafter, we will show that pattern pre-opacity can also be transformed to standard pre-opacity by refining the state-space and suitably defining secret states.

**Example 6.** *Consider again the example shown in Figure 3–7(b). The secret sequence pattern can be described by the regular language*

$$\begin{aligned}\Omega =&((E\backslash\{\alpha_1\})^*\{\alpha_1\}(E\backslash\{\alpha_6\})^*\{\alpha_6\}(E\backslash\{\beta_3\})^*\{\beta_3\}\\&\cup (E\backslash\{\alpha_1\})^*\{\alpha_1\}(E\backslash\{\beta_2\})^*\{\beta_2\})^*\end{aligned} \tag{3–13}$$

*Essentially, regular language $\Omega$ includes all strings that contain $\alpha_1\alpha_6\beta_3$ or $\alpha_1\beta_2$. This language can be marked by DFA $G_\Omega$ shown in Figure 3–7(c). Obviously, $G_3$ is 2-step instant pattern pre-opaque, since based on any observation, the intruder cannot know for sure the system will finish a sequence pattern $\alpha_1\beta_2$ or $\alpha_1\alpha_6\beta_3$ 2-step ahead. However, as we discussed early, it is not 1-step instant pattern pre-opaque; this is because, once string $\alpha_1\alpha_6$ is observed, the monitor*

*knows for sure that sequence $\alpha_1\alpha_6\beta_3$ will be completed in 1-step. Also, we can check that $G_3$ is 2-step trajectory pattern pre-opaque but not 1-step trajectory pattern pre-opaque.*

*Note that when string $\alpha_1\alpha_6$ is observed, we know that sequence $\alpha_1\beta_2$ has been finished one step ago. Although the monitor fails to detect pattern $\alpha_1\beta_2$ before its completion, it still can predict secret sequence pattern $\alpha_1\alpha_6\beta_3$.*

**Remark 5.** *The concept of* sequence pattern *was first proposed in the literature for the purpose of fault diagnosis [76] and fault prognosis [69]. Specifically, a sequence pattern is used to model the set of behaviors considered as fault. Our notion of sequence pattern is more general than that in the context of fault diagnosis/prognosis. In particular, in the context of fault diagnosis/prognosis, the sequence pattern is assumed to be* stable *in the sense that any continuation of a sequence in the pattern is still in the pattern. This is motivated by the setting of permanent fault. However, our definition of sequence pattern does not necessarily be stable as the system can be secret/non-secret intermittently. In other words, even if the intruder miss the predication of the first pattern, it may still be able to predict some future pattern, and in this case, the system is also not pre-opaque.*

### 3.5.3 Verifications of Pattern Pre-Opacity

We show how to verify pattern pre-opacity in this part. To this end, we assume that the secret sequence pattern $\Omega$ is a regular language and it is recognized by a DFA $G_\Omega = (X_\Omega, E, f_\Omega, x_{0,\Omega}, X_{m,\Omega})$, i.e., $\mathcal{L}_m(G_\Omega) = \Omega$, where $x_{0,\Omega}$ is the unique initial state. Without loss of generality, we assume that $G_\Omega$ is total, i.e., $\mathcal{L}(G_\Omega) = E^*$; otherwise, we can add a new unmarked "dump" state and complete the transition function.

Then let $G = (X, E, f, X_0)$ be the system and $G_\Omega = (X_\Omega, E, f_\Omega, x_{0,\Omega}, X_{m,\Omega})$ be the DFA recognizing the sequence pattern. We define the product of $G$ and $G_\Omega$ as

$$G_\times = (X', E', f', X_0'),$$

where $X' \subseteq X \times X_\Omega, E' = E, X_0' = X_0 \times \{x_{0,\Omega}\}$ and $f' : X' \times E \to X'$ is the transition function defined by $f((x_1, x_2), \sigma) = (f(x_1, \sigma), f_\Omega(x_2, \sigma))$, if $f(x_1, \sigma)$ and $f_\Omega(x_2, \sigma)$ are defined, and undefined otherwise. Then we define

$$X_S' = \{(q_1, q_2) : q_2 \in X_{m,\Omega}\}$$

as the set of secret states in $G_\times$. Then the following result shows that pattern pre-opacity can be transformed to state-based pre-opacity.

**Theorem 5.** *System $G$ is $K$-step instant (respectively, trajectory) pattern pre-opaque w.r.t. $\Omega$ if and only if $G \times G_\Omega$ is $K$-step instant (respectively, trajectory) pre-opaque w.r.t. $X'_S$.*

***Proof.*** We only show the case of instant pre-opacity; the case of trajectory pre-opacity is similar.

($\Rightarrow$) Suppose that $G \times G_\Omega$ is not $K$-step instant pre-opaque, which implies that

$$\begin{aligned}
&(\exists (x_0, x_{0,\Omega}) \in X'_0)(\exists s \in \mathcal{L}_o(G \times G_\Omega, (x_0, x_{0,\Omega})))(\exists n_0 \geq K) \\
&(\forall (x'_0, x'_{0,\Omega}) \in X'_0) \\
&(\forall s' \in \mathcal{L}_o(G \times G_\Omega, (x'_0, x'_{0,\Omega})), s't \in \mathcal{L}(G \times G_\Omega, (x'_0, x'_{0,\Omega}))) \\
&[P(s) = P(s') \wedge |t| = n_0] \Rightarrow [f'((x'_0, x'_{0,\Omega}), s't) \in X'_S]
\end{aligned}$$

Since $G_\Omega$ is complete, we have $\mathcal{L}(G) \subseteq \mathcal{L}(G_\Omega)$, then we know that for any $x'_0 \in X_0$, any $s' \in \mathcal{L}_o(G, x'_0), s't \in \mathcal{L}(G, x'_0)$ such that $P(s) = P(s')$ and $|t| = n_0 \geq K$, we have that $f_\Omega(x'_0, s't) \in X_{m,\Omega}$, i.e., $s't \in \mathcal{L}_m(G_\Omega) = \Omega$. This implies that $G$ is not $K$-step instant pattern pre-opaque

($\Leftarrow$) Assume that $G$ is not $K$-step instant pattern pre-opaque, i.e.,

$$\begin{aligned}
&(\exists x_0 \in X_0)(\exists s \in \mathcal{L}_o(G, x_0))(\exists n_0 \geq K) \\
&(\forall x'_0 \in X_0)(\forall s' \in \mathcal{L}_o(G, x'_0), s't \in \mathcal{L}(G, x'_0)) \\
&[P(s) = P(s') \wedge |t| = n_0 \wedge s't \in \Omega]
\end{aligned}$$

Since $\mathcal{L}(G \times G_\Omega) \subseteq \mathcal{L}(G)$, we know that for any $(x'_0, x_{0,\Omega}) \in X'_0, s' \in \mathcal{L}_o(G \times G_\Omega, (x'_0, x_{0,\Omega}))$ and $s't \in \mathcal{L}(G \times G_\Omega, (x'_0, x_{0,\Omega}))$ such that $P(s') = P(s)$ and $|t| = n_0$, we always have $f'((x'_0, x'_{0,\Omega}), s't) \in X'_S$ and $|t| = n_0 \geq K$, which means that $G \times G_\Omega$ is not $K$-step instant pre-opaque. □

**Example 7.** *Consider again system automaton $G_3$ and pattern automaton $G_\Omega$ in Figure 3–7. To verify the $K$-step instant/trajectory pattern pre-opacity of $G_3$, we fist construct $G_3 \times G_\Omega$, which is omitted here for the sake of brevity. Then the set of secret states in $G_3 \times G_\Omega$ is $X'_S = \{(6, F), (8, H)\}$. One can verify that $G_3 \times G_\Omega$ is 2-step instant pre-opaque but not 1-step instant pre-opaque; also, $G_3 \times G_\Omega$ is 2-step trajectory pre-opaque but not 1-step trajectory pre-opaque. Therefore, based on Theorem 5, for sequence pattern captured by $\Omega$, we know that $G_3$ is 2-step instant pattern pre-opaque but not 1-step instant pattern pre-opaque, and it is 2-step trajectory pattern pre-opaque but not 1-step trajectory pattern pre-opaque, which are consistent with our previous analysis.*

## 3.6 Conclusion

In this chapter, we proposed the notion of pre-opacity to verify the *intention security* of a partially-observed DES. Two notions of pre-opacity called $K$-step instant pre-opacity and $K$-step trajectory pre-opacity are proposed. For each notion of pre-opacity, we provide a verifiable necessary and sufficient condition as well as an effective verification algorithm. We also generalize the notions of pre-opacity to the case where the secret behavior is captured by a sequence pattern. Our work extends the theory of opacity to a new class where secret is related to the intention of the system. We believe there are many interesting future directions related to the concept of pre-opacity. One interesting direction is to *synthesize* a supervisor to enforce pre-opacity when the verification result is negative. Also, we would like to extend the notion of pre-opacity to the stochastic setting to quantitatively evaluate the information leakage.

# Chapter 4 Secure-by-Construction Optimal Path Planning for Linear Temporal Logic Tasks

## 4.1 Introduction

Motivated by the security concerns in robotic systems, in this chapter, we formulate and solve a security-aware optimal LTL path planning problem. Specifically, we consider a single robot whose mobility is modeled as a weighted transition system (WTS). We consider an intruder modeled as an outside observer (or eavesdropper) who accesses the external behaviors of the system specified by an output function. We consider the planning problem of achieving a task specified by a general LTL formula, while hiding the secret initial location of the robot. To capture this security requirement, we adopt the notion of an information-flow security property called *initial-state opacity* [4]. Specifically, a planed path from a secret initial-state is said to be *secure* if there exists another path from a non-secret initial-state such that those two paths are observationally equivalent from the intruder's point of view.

Our approach is different from the standard initial-state opacity verification procedure [4], which requires to build the initial-state estimator whose size is exponential in the number of system states. Instead, we propose a computationally more efficient approach by constructing the twin-WTS structure which synchronizes the system with its copy based on the observation. Similar structures have been used in the literature for the purpose of property verification, e.g., diagnosability, observability and prognosability. Here, we show that the security-aware path planning problem can be effectively solved by graph search in the product of the twin-WTS and the Büchi automaton that accepts the desired LTL task. Furthermore, we show that the constructed product system also preserves optimality. Hence, we provide a sound and complete solution to the security-aware optimal LTL planning problem.

The rest of the chapter is organized as follows. In Sections 4.2 and 4.3, a motivating example and some necessary preliminaries are presented, respectively. The security-aware LTL planning problem is formally formulated in Section 4.4. In Section 4.5, we discuss how to solve this problem based on the twin-WTS. Finally, we conclude the chapter by Section 4.6.

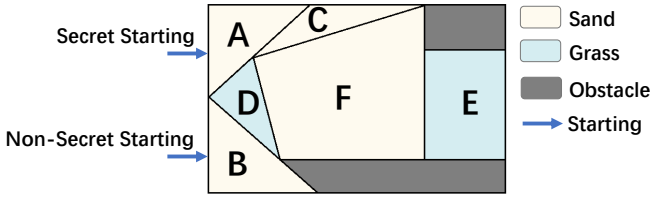



Figure 4–1 Work space of the single robot.

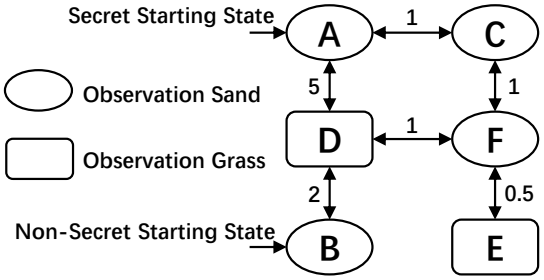



Figure 4–2 The specification automaton of the motivating example. The intruder has two observations on the robot: the robot is in sand land or grass land. The robot could start from A (secret) or B (non-secret). Bidirectional transition means that robot could move in both directions; numbers beside the transition represent the cost of this transition.

## 4.2 Motivating Example

Before we formally formulate the problem, we first consider a motivating example. Suppose that a single mobile robot moves in a workspace with grass lands and sand lands as shown in Figure 4–1. The workspace is partitioned as six regions of interest and black regions denote obstacles. At each instant, the robot can only move to regions that are adjacent to its current region. We assume that the robot always knows exactly its current location. On the other hand, we assume that there is an *outside observer* that knows whether the robot is current at a grass land or at a sand land. The mobility model of the robot can be represented by the transition system shown in Figure 4–2. Furthermore, we assume that there is a cost moving from one region to another, which is specified by the number associated to each bidirectional transition in Figure 4–2.

The task of the robot is to deliver goods between region $F$ (representing, e.g., a factory) and region $E$ (representing, e.g., a warehouse), i.e, visit $F$ and $E$ infinitely often. The robot may initially start from regions $A$ or $B$. However, it does not want the outside observer to know that it started from region $A$ (if so). This may because, for example, starting from different locations

means different robot-types are used, which may further reveal what kind of goods the factory is delivering.

Now, suppose that the robot is starting from region $A$. Clearly, the optimal plan to achieve the task is

$$A \rightarrow C \rightarrow (F \rightarrow E)^{\omega},$$

where notation $\omega$ over parentheses means the infinite repetition of the finite execution inside them. However, this plan is not secure in the sense that the observer will know for sure that the robot started from region $A$ after observing two consecutive $Grass$. This is because there is no feasible path from region $B$ that can generate the same observation. On the other hand, the robot may take another plan

$$A \rightarrow D \rightarrow (F \rightarrow E)^{\omega}.$$

This plan is costlier as the robot will incur higher cost when moving from $A$ to $D$. However, this plan is secure in the sense that there exists another path

$$B \rightarrow D \rightarrow (F \rightarrow E)^{\omega},$$

from region $B$ that generates the same observation. Therefore, although higher cost is paid, the robot is able to hide the secret about its initial location.

## 4.3 Temporal Logic Task Planning

In this section, we define basic notations that we use in the rest of the chapter and introduce some necessary preliminaries. For a set $A$, we denote by $|A|$ and $2^A$ its cardinality and its power set, respectively. A finite sequence over $A$ is a sequence in the form of $a_1 \cdots a_n$, where $a_i \in A$; we denote by $A^*$ the set of all finite sequences over $A$. Similarly, we denote by $A^{\omega}$ the set of all infinite sequences over $A$.

The *trace* of an infinite path $\tau \in Q^{\omega}$ denoted by $\mathrm{trace}(\tau)$ is an infinite sequence over $2^{\mathcal{AP}}$ such that $\mathrm{trace}(\tau) = L(\tau(1))L(\tau(2))L(\tau(3))\cdots$. Given a set of states $Q' \subseteq Q$, we denote by $\mathrm{Reach}(Q')$ the set of states reachable from $Q'$. We say a state $q \in Q$ is in a cycle of $T$ if there exists a sequence $q_1 q_2 \ldots q_k \in Q^*$ such that $q_1 = q_k = q$ and $(q_i, q_{i+1}) \in \rightarrow, \forall i \in \mathbb{N}_+$. We denote by $\mathrm{cycle}(T)$ the set of all states that are in some cycles of $T$.

### 4.3.1 Linear Temporal Logic and Büchi Automata

Let $\mathcal{AP}$ be the set of atomic propositions. A Linear Temporal Logic (LTL) formula is constructed based on atomic propositions, Boolean operators, and temporal operators. Specifically,

an LTL formula $\phi$ is recursively defined by

$$\phi ::= true \mid p \mid \phi_1 \wedge \phi_2 \mid \neg\varphi \mid \bigcirc\phi \mid \phi_1 U \phi_2,$$

where $p \in \mathcal{AP}$ is an atomic proposition; $\bigcirc$ and $U$ denote, respectively, "next" and "until". The above syntax also induces temporal operators $\Diamond$ ("eventually") and $\Box$ ("always"), where $\Diamond\phi := trueU\phi$ and $\Box\phi := \neg\Diamond\neg\phi$.

LTL formulas are used to evaluate whether or not *infinite words* satisfy some properties. Formally, an infinite word $\sigma \in (2^{\mathcal{AP}})^\omega$ is an infinite sequence over alphabet $2^{\mathcal{AP}}$. We denote by $\sigma \models \phi$ if $\sigma$ satisfies the LTL formula $\phi$. For example, $\Box\Diamond\mathbf{p}$ means that property $\mathbf{p}$ should be satisfied infinitely often. The reader is referred to [46] for more details about the syntax and the semantics of LTL, which are omitted here. We define $\mathrm{Words}(\phi) = \{\sigma \in (2^{\mathcal{AP}})^\omega : \sigma \models \phi\}$ as the set of all words satisfying LTL formula $\phi$.

**Definition 7.** *(Nondeterministic Büchi Automaton) A Nondeterministic Büchi Automaton (NBA) is a 5-tuple $B = (Q_B, Q_{0,B}, \Sigma, \rightarrow_B, F_B)$, where $Q_B$ is the set of states, $Q_{0,B} \subseteq Q_B$ is the set of initial states, $\Sigma$ is an alphabet, $\rightarrow_B \subseteq Q_B \times \Sigma \times Q_B$ is the transition relation and $F_B \subseteq Q_B$ is the set of accepting states.*

Given an infinite word $\sigma = \pi_0\pi_1\pi_2\cdots \in \Sigma^\omega$, an infinite *run* of $B$ over $\sigma$ is an infinite sequence $\rho = q_0q_1q_2\cdots \in Q_B^\omega$ such that $q_0 \in Q_{0,B}$ and $(q_i, \pi_i, q_{i+i}) \in \rightarrow_B$ for any $i \in \mathbb{N}$. An infinite run $\rho \in Q_B^\omega$ is said to be *accepted* by $B$ if $\mathrm{Inf}(\rho) \cap F_B \neq \emptyset$, where $\mathrm{Inf}(\rho)$ denotes the set of states that appears infinite number of times in $\rho$. Then an infinite word $\sigma$ is said to be accepted by $B$ if it induces an infinite run accepted by $B$. We denote by $\mathcal{L}_B \subseteq \Sigma^\omega$ the set of all accepted words in NBA $B$.

For any LTL formula $\phi$, it is well-known [77] that there always exists an NBA over $\Sigma = 2^{\mathcal{AP}}$ that accepts exactly all infinite words satisfying $\phi$, i.e., $\mathcal{L}_B = \mathrm{Words}(\phi)$. Throughout this chapter, $B = (Q_B, Q_{0,B}, 2^{\mathcal{AP}}, \rightarrow_B, F_B)$ is used to denote the NBA translated from the LTL formula $\phi$ of interest.

### 4.3.2 Temporal Logic Path Planning

The standard LTL path planning problem asks to find an infinite path $\tau \in \mathrm{Path}^\omega(T)$ in system $T$ such that $\mathrm{trace}(\tau) \models \phi$, i.e., the corresponding temporal property along the infinite path satisfies the LTL formula. Due to the structure of the accepting condition in Büchi automaton, it suffices to find an infinite path with the following *prefix-suffix structure*

$$\tau = q_1 \cdots q_k [q_{k+1} \cdots q_{k+m}]^\omega \in \mathrm{Path}^\omega(T)$$

such that $\mathrm{trace}(\tau) \in \mathcal{L}_B$. Intuitively, $q_{k+1} \cdots q_{k+m}$ is the suffix that forms a cycle such that the robot should execute infinitely often, while $q_1 \cdots q_k$ is the prefix representing that transient path that leads to the cyclic path. Such a prefix-suffix structure is also referred to a *plan*. In this work, we consider the cost of a plan, which is an infinite path, as the cost of its prefix and suffix, i.e.,

$$\hat{J}(\tau) = J(q_1 \cdots q_k q_{k+1} \cdots q_{k+m} q_{k+1}). \tag{4–1}$$

In order to find an optimal plan with least cost, one can perform modified shortest path search in the *product system* composed by $T$ and $B$; see, e.g., [51].

**Remark 6.** *The cost function defined in Equation* (4–1) *essentially treats the transient cost* $J_{pre} = J(q_1 \cdots q_k q_{k+1})$ *and the steady-state cost* $J_{suf} = J(q_{k+1} \ldots q_{k+m} q_{k+1})$ *equivalently. In general, we can define the cost function as* $\hat{J}(\tau) = \alpha J_{pre} + (1-\alpha) J_{suf}$*, where* $\alpha \in [0, 1]$ *is a parameter adjusting the weight of each part. Our work considers the case of* $\alpha = 0.5$ *for the sake of simplicity; all results can be easily extended to the general case.*

**Remark 7.** *Given an infinite path (plan), depending on how we decompose prefix and suffix, the plan may have different costs. For example,* $q_1(q_2q_3)^\omega$ *and* $q_1q_2(q_3q_2)^\omega$ *are the same path but have different costs. Hereafter, for an plan* $\tau$*,* $\hat{J}(\tau)$ *is always considered as the cost for the prefix-suffix structure of* $\tau$ *having the minimum cost.*

## 4.4 Security-Aware Path Planning Problem

As we discussed in the motivating example, the solution to the standard LTL path planning problem does not necessarily provide security guarantees. In this section, we present the considered information-flow security model and formulate the security-aware path planning problem.

Given WTS $T = (Q, Q_0, \to, w, \mathcal{AP}, L)$, we assume that the internal state of the system is not available to the intruder (malicious observer) directly. Instead, the intruder can only infer the behavior of the system via its outputs. Formally, we model the intruder's observation of the system as an output function

$$H : Q \to Y,$$

where $Y$ is the set of outputs. The execution of any infinite internal path $\tau = \tau(1)\tau(2)\tau(3) \cdots \in \mathrm{Path}^\omega(T)$ will generate an infinite *external path* $H(\tau(1))H(\tau(2))H(\tau(3)) \cdots \in Y^\omega$; we also denote this external path by $H(\tau)$ with a slight abuse of notation. A finite external path is defined analogously.

In this work, we consider the problem of protecting *secret initial location* of the robot. To this end, we assume that $Q_S \subset Q_0$ is the set of *secret initial states*. Hereafter, a WTS $T$ equipped with output function $H$ and secret initial states $Q_S$ is also written as $T = (Q, Q_0, \rightarrow, w, \mathcal{AP}, L, H, Y, Q_S)$ for simplicity. To guarantee security, we want to make sure that the intruder is not able to infer confidentially that the robot started from a secret location. This requirement is formalized as follows.

**Definition 8.** *(Security) Let $T = (Q, Q_0, \rightarrow, w, \mathcal{AP}, L, H, Y, Q_S)$ be a WTS. An infinite path $\tau \in \mathrm{Path}^\omega(T)$ is said to be* secure *if there exists an infinite path $\tau' \in \mathrm{Path}^\omega(T)$ such that $\tau'(1) \notin Q_S$ and $H(\tau) = H(\tau')$.*

**Remark 8.** *The above definition of security is related to the notion of initial-state opacity proposed in [4]. Essentially, initial-state opacity is a system property such that all paths generated by the system in secure is our sense. However, as we are considering path planning problem, security is defined only for a specific path rather than the entire system.*

**Problem 1.** *(Security-Aware Optimal LTL Path Planning Problem) Given a WTS $T$, secret states $Q_S \subset Q$, output function $H : Q \rightarrow Y$ and LTL formula $\phi$, for each possible initial-state $q_0 \in Q_0$, determine a plan $\tau \in \mathrm{Path}^\omega(T)$ with $\tau(1) = q_0$ such that the following conditions hold:*

1. $\mathrm{trace}(\tau) \models \phi$;
2. *$\tau$ is secure;*
3. *For any other plan $\tilde{\tau} \in \mathrm{Path}^\omega(T)$ satisfying the above requirements, we have $\hat{J}(\tau) \leq \hat{J}(\tilde{\tau})$.*

**Remark 9.** *In the above problem formulation, it essentially assumes that the intruder knows the followings:*

1. *the mobility model of the robot, i.e., WTS $T$; and*
2. *the external path generated by the robot, i.e., $H(\tau)$.*

*However, it does not know the exact internal state of the robot, which has to be inferred by observing outputs. On the other hand, the robot is assumed to know exactly its initial and current state; therefore, this is still a planning problem under perfect information from the robot's point of view. This setting is reasonable in many applications because: (i) the system usually has more ability to acquire information about itself than the intruder; and (ii) the intruder's information sometimes comes from eavesdropping the information transmission which is a partial information of the robot's knowledge.*

**Remark 10.** *According to Definition 8, if a path $\tau$ is started from a non-secret initial state $q_0 \in Q_0 \setminus Q_S$, then it is always secure as we can choose $\tau' = \tau$. Therefore, for non-secret initial states, we just need to solve the standard optimal LTL path planning problem; see, e.g., [49]. However, for those secret initial states, the security constraint has to be taken into account. This issue will be addressed in the following section.*

## 4.5 Planning Algorithm

In this section, we present the security-aware path planning algorithm. Our approach is based on constructing a new transition system that effectively captures the security constraint.

### 4.5.1 Twin-WTS

In order to handle the security constraint, one needs to track the information of the outside observer based on the external path. Such an information-tracking task can be achieved by constructing the *initial-state estimator* [4]. However, the size of the initial-state estimator grows exponentially as the number of states in the system increases due to the subset construction.

Here, we present a computationally more efficient approach that does not rely on the construction of the initial-state estimator. Instead, we propose a new structure called the *twin-WTS*, which is used to track all current states pairs of two paths that have the same external path from the intruder's point view. This structure is formally defined as follows.

**Definition 9.** *(Twin-WTS) Given a WTS $T = (Q, Q_0, \rightarrow, w, \mathcal{AP}, L, H, Y, Q_S)$, its twin-WTS is a new WTS*

$$V = (X, X_0, \rightarrow_V, w_V, \mathcal{AP}, L_V),$$

*where*

- *$X \subseteq Q \times Q$ is the set of states;*
- *$X_0 = \{(q_1, q_2) \in Q_0 \times Q_0 : H(q_1) = H(q_2)\}$ is the set of initial-states;*
- *$\rightarrow_V \subseteq X \times X$ is the transition relation defined by: for any $x = (q_1, q_2) \in X$ and $x' = (q_1', q_2') \in X$, we have $(x, x') \in \rightarrow_V$ if the followings hold:*
  - *$(q_1, q_1') \in \rightarrow$;*
  - *$(q_2, q_2') \in \rightarrow$;*
  - *$H(q_1') = H(q_2')$.*
- *$w_V : X \times X \rightarrow \mathbb{R}_+$ is the cost function defined by: for any $x = (q_1, q_2) \in X$ and $x' = (q_1', q_2') \in X$, we have $w_V(x, x') = w(q_1, q_1')$;*

- $L_V : X \to 2^{\mathcal{AP}}$ *is the labeling function defined by: for any* $x = (q_1, q_2) \in X$, *we have* $L_V(x) = L(q_1)$.

**Remark 11.** *Intuitively, the twin-WTS tracks two internal paths that generate the same external path. Specifically, the first component is used to represent the trajectory in the real system, while the second component is used to represent a copy that mimics the real system in the sense of output equivalence. Therefore, for any path* $(\tau_1(1), \tau_2(1))(\tau_1(2), \tau_2(2)) \cdots$ *in* $V$, *we have* $H(\tau_1(1))H(\tau_1(2)) \cdots = H(\tau_2(1))H(\tau_2(2)) \cdots$. *On the other hand, for any two paths* $\tau_1, \tau_2$ *in* $T$ *such that* $H(\tau_1) = H(\tau_2)$, *we can find a path* $\tau$ *in* $V$ *such that its first component is* $\tau_1$ *and its second component is* $\tau_2$. *Also, we note that the cost function* $w_V$ *and the labeling function* $L_V$ *are all defined based on the states in the first component, which is the part for the real system. Finally, the size of* $V$ *is polynomial in the size of* $T$ *as it contains at most* $|Q|^2$ *states.*

### 4.5.2 Planning Algorithm

The twin-WTS can be used to capture the security constraint based on the following observation. For any secure path for a secret initial state $q_{s,0}$, it must have an observation-equivalent path from a non-secret initial state $q_{ns,0}$. Furthermore, such a path-pair should exist in the twin-WTS $V$ from state $(q_{s,0}, q_{us,0})$. Therefore, to perform security-aware path planing, it suffices to perform planning from an initial-state in $V$ in which the first component is the real (secret) initial-state and the second component is a non-secret state. Furthermore, in order to incorporate the temporal task, we need to synchronize the twin-WTS with the NBA $B$ that accepts $\phi$; this is defined as the product system.

**Definition 10.** *(Product System) Given twin-WTS* $V = (X, X_0, \to_V, w_V, \mathcal{AP}, L)$ *and NBA* $B = (Q_B, Q_{0,B}, \Sigma, \to_B, F_B)$, *the product of* $V$ *and* $B$ *is a new (unlabeled) WTS*

$$T_\otimes = (Q_\otimes, Q_{0,\otimes}, \to_\otimes, w_\otimes),$$

*where*

- $Q_\otimes \subseteq X \times Q_B$ *is the set of states;*
- $Q_{0,\otimes} = X_0 \times Q_{0,B}$ *is the set of initial states;*
- $\to_\otimes \subseteq Q_\otimes \times Q_\otimes$ *is the transition relation defined by: for any* $q_\otimes = (x, q_B) \in Q_\otimes$ *and* $q'_\otimes = (x', q'_B) \in Q_\otimes$, *we have* $(q_\otimes, q'_\otimes) \in \to_\otimes$ *if the followings hold:*
  - $(x, x') \in \to_V$; *and*
  - $(q_B, L_V(x), q'_B) \in \to_B$.

- $w_\otimes : Q_\otimes \times Q_\otimes \to \mathbb{R}_+$ *is the cost function defined by: for any* $q_\otimes = (x, q_B) \in Q_\otimes$ *and* $q'_\otimes = (x', q'_B) \in Q_\otimes$, *we have* $w_\otimes(q_\otimes, q'_\otimes) = w(x, x')$.

Essentially, the product system further restricts the dynamic of $V$ such that each movement should satisfy the LTL task $\phi$, i.e., $(q_B, L_V(x), q'_B) \in \to_B$. Note that the original WTS $T$ is not synchronized with $B$ as the dynamic of $T$ has already been encoded in the first component of $V$. For each state $((q, q'), q_B) \in Q_\otimes$, we denote by $\Pi[((q, q'), q_B)] = q$ the restriction to the state space of $T$; we also write $\Pi[((q_0, q'_0), q_{0,B}) \cdots ((q_n, q'_n), q_{n,B})] = q_0 \cdots q_n$.

For each initial-state $q_0 \in Q_0$ in $T$, we denote by $\textsc{Int}_{q_0}(T_\otimes) \subseteq Q_{0,\otimes}$ the set of initial-states in $T_\otimes$ whose first components is $q_0$ while the second component is a non-secret state in $T$, i.e.,

$$\textsc{Int}_{q_0}(T_\otimes) = \{((q_0, q'_0), q_B) \in Q_{0,\otimes} : q'_0 \notin Q_S\}.$$

Also, we define $\textsc{Goal}(T_\otimes) \subseteq Q_\otimes$ as the set of states in $T_\otimes$ whose last components are in $F_B$ and they are in some cycles of $T_\otimes$, i.e.,

$$\textsc{Goal}(T_\otimes) = \{((q, q'), q_B) \in Q_\otimes : q_B \in F_B \wedge ((q, q'), q_B) \in \mathrm{cycle}(T_\otimes)\}.$$

In order to find an optimal path from initial state $q_0$ in $T$, it suffices to find an optimal path in the form of

$$\textsc{Int}_{q_0}(T_\otimes) \to (\textsc{Goal}(T_\otimes) \to \textsc{Goal}(T_\otimes))^\omega$$

in $T_\otimes$. Note that both sets $\textsc{Int}_{q_0}(T_\otimes)$ and $\textsc{Goal}(T_\otimes)$ are non-singleton in general. Therefore, we need to consider all possible combinations in order to determine an optimal path. This idea is formalized by Algorithm 1.

Specifically, lines 1-3 construct the NBA $B$, the twin-WTS $V$ and the product system $T_\otimes$. Line 4 aims to determine if there is a feasible path from $q_0$ satisfying both the LTL constraint and the security constraint. In particular, if $\textsc{Int}_{q_0}(T_\otimes)$ cannot reach any goal state in cycle, then this means that there does not exist an infinite path accepted by $\phi$ that has an observation equivalent path from a non-secret initial state, i.e., there exists no feasible path starting from $q_0$. Otherwise, we consider, in lines 7 and 8, each combination of state $q_I$ in $\textsc{Int}_{q_0}(T_\otimes)$ and state $q_G$ in $\mathrm{Reach}(\{q_I\}) \cap \textsc{Goal}(T_\otimes)$, which is a goal state reachable from $q_1$ and in some cycles. In lines 9-10, we determine the shortest path from $q_I$ to $q_G$ and the shortest path from $q_G$ back to itself; the projection onto $T$ by $\Pi$ then gives us an infinite path satisfying both the LTL and the security constraint. Then among all such feasible combinations, we determine the optimal pair $(q_I^*, q_G^*)$

**Algorithm 4–1** Security-Aware Optimal LTL Plan

**Input:** LTL formula $\phi$, WTS $T$ with $H$ and $Q_S$, initial state $q_0$

**Output:** Optimal plan $\tau$ from $q_0 \in Q_0$

Convert $\phi$ to NBA $B = (Q_B, Q_{0,B}, \Sigma, \rightarrow_B, F_B)$
Construct twin-WTS $V = (X, X_0, \rightarrow_V, w_V, \mathcal{AP}, L_V)$
Construct the product of $V$ and $B$ $T_\otimes = (Q_\otimes, Q_{0,\otimes}, \rightarrow_\otimes, w_\otimes)$
**if** $\mathrm{Reach}(\textsc{Int}_{q_0}(T_\otimes)) \cap \textsc{Goal}(T_\otimes) = \emptyset$ **then**
    **return** "no feasible plan from $q_0$"
**else**
    **for** $q_I \in \textsc{Int}_{q_0}(T_\otimes)$ **do**
        **for** $q_G \in \mathrm{Reach}(\{q_I\}) \cap \textsc{Goal}(T_\otimes)$ **do**
            $\tau^{q_I,q_G} = \Pi[\mathrm{Shortpath}(q_I, q_G)]$
            $\tau^{q_G,q_G} = \Pi[\mathrm{Shortpath}(q_G, q_G)]$
        **end for**
    **end for**
    $(q_I^*, q_G^*) =_{(q_I,q_G)} \hat{J}(\tau^{q_I,q_G}[\tau^{q_G,q_G}]^\omega)$
    **return** optimal plan $\tau = \tau^{q_I^*,q_G^*}[\tau^{q_G^*,q_G^*}]^\omega$ for $q_0$
**end if**

that minimizes the path cost function defined in (4–1) and the optimal plan $\tau = \tau^{q_I^*,q_G^*}[\tau^{q_G^*,q_G^*}]^\omega$ is returned.

**Remark 12.** *Let us discuss the complexity of Algorithm 4–1. First, we note that the product system $T_\otimes$ contains at most $|Q|^2|Q_B|$ states, where $|Q|$ is the number of states in the WTS model and $|Q_B|$ is the number of states in the Büchi automaton. Algorithm 4–1 involves at most $|Q|^4|Q_B|^2$ (very roughly estimated) shortest path problems which can be solved in polynomial-times in the number of states in $T_\otimes$. Therefore, the overall planning complexity is polynomial in both the number of states in the plant and the number of states in the Büchi automaton. Note that, in general, $|Q_B|$ is the length of $\phi$. However, in practice, the size of the LTL formula $\phi$ is usually very small and $Q$, which represents the state-space, is usually the main factor for scalability.*

### 4.5.3 Correctness of the Planning Algorithm

Now, we prove the correctness of the proposed planning algorithm. Hereafter, we assume that the robot is starting from initial state $q_0$ and $\tau$ is the optimal plan from $q_0$ returned by Algorithm 1. First, we show that the resulting plan satisfies the LTL task $\phi$.

**Proposition 5.** *Assume that the robot is starting from initial state $q_0$ and $\tau$ is the optimal plan from $q_0$ returned by Algorithm 1, then* $\mathrm{trace}(\tau) \models \phi$.

***Proof.*** We assume that optimal path is obtained from the following projection $\tau = \Pi[p^{pre}(p^{suf})^{\omega}]$, where

$$p^{pre} = ((q_0, q'_0), q_{0,B}) \cdots ((q_n, q'_n), q_{n,B})$$
$$p^{suf} = ((q_{n+1}, q'_{n+1}), q_{n+1,B}) \cdots ((q_{n+m}, q'_{n+m}), q_{n+m,B})$$

and $q_{n+1,B} \in F_B$. That is, $\tau = q_0 \cdots q_n (q_{n+1} \cdots q_{n+m})^{\omega}$. According to the transition rule of $T_{\otimes}$, $\rho = q_{0,B} \cdots q_{n,B} (q_{n+1,B} \cdots q_{n+m,B})^{\omega}$ is an infinite run induced by infinite word $\mathrm{trace}(\tau) = L(q_0) \cdots L(q_n)(L(q_{n+1}) \cdots L(q_{n+m}))^{\omega}$. Since $q_{n+1,B} \in F_B$, we know that $\mathrm{Inf}(\rho) \cap F_B \neq \emptyset$, which means $\mathrm{trace}(\tau) \in \mathcal{L}_B = \mathrm{Word}(\phi)$, i.e., $\mathrm{trace}(\tau) \models \phi$. □

Second, we show that the planned path is secure.

**Proposition 6.** *Assume that the robot is starting from initial state $q_0$ and $\tau$ is the optimal plan from $q_0$ returned by Algorithm 1, $\tau$ is secure.*

***Proof.*** Without loss of generality, we assume that $q_0 \in Q_S$; otherwise, $\tau$ is secure trivially. Still, we assume that optimal path is obtained by $\tau = \Pi[p^{pre}(p^{suf})^{\omega}]$, where

$$p^{pre} = ((q_0, q'_0), q_{0,B}) \cdots ((q_n, q'_n), q_{n,B})$$
$$p^{suf} = ((q_{n+1}, q'_{n+1}), q_{n+1,B}) \cdots ((q_{n+m}, q'_{n+m}), q_{n+m,B})$$

Then we know that $(q_0, q'_0) \cdots (q_n, q'_n)((q_{n+1}, q'_{n+1}) \cdots (q_{n+m}, q'_{n+m}))^{\omega} \in \mathrm{Path}^{\omega}(V)$. According to the transition rule of $V$, we have $H(\tau) = H(q'_0 \cdots q'_n (q'_{n+1} \cdots q'_{n+m})^{\omega})$. Finally, since $((q_0, q'_0), q_{0,B}) \in \mathrm{Inf}_{q_0}(T_{\otimes})$, we know that $q'_0 \notin Q_S$. Therefore, $\tau' = q'_0 \cdots q'_n (q'_{n+1} \cdots q'_{n+m})^{\omega}$ is an internal path from a non-secret initial state having the same observation with $\tau$, i.e., $\tau$ is secure. □

Finally, we show that the planned path is optimal.

**Proposition 7.** *Assume that the robot is starting from initial state $q_0$ and $\tau$ is the optimal plan from $q_0$ returned by Algorithm 1, for any other secure path $\tilde{\tau} = \tilde{\tau}^{pre}[\tilde{\tau}^{suf}]^{\omega}$ such that* $\mathrm{trace}(\tilde{\tau}) \models \phi$, *we have* $\hat{J}(\tau) \leq \hat{J}(\tilde{\tau})$.

***Proof.*** We prove by contradiction. Suppose that there exists a secure path $\tilde{\tau} \in \mathrm{Path}^{\omega}(T)$ such that $\mathrm{trace}(\tilde{\tau}) \models \phi$ and $\hat{J}(\tilde{\tau}) < \hat{J}(\tau)$. Since $\tilde{\tau}$ is secure, we know that there exists another path $\tilde{\tau}' \in \mathrm{Path}^{\omega}(T)$ such that $H(\tilde{\tau}) = H(\tilde{\tau}')$ and $\tilde{\tau}'(1) \notin Q_S$. According to the definition of $V$, we know that there exists a path $\tau_V \in \mathrm{Path}^{\omega}(V)$ in which the first component is $\tilde{\tau}$ and the second component is $\tilde{\tau}'$. Furthermore, since $\mathrm{trace}(\tilde{\tau}) \models \phi$, by the definition of $T_{\otimes}$, there exists a path $\tau_{\otimes} \in \mathrm{Path}^{\omega}(T_{\otimes})$ in which the first component is $\tau_V$ and the second component is $\tau_B$ such that $\mathrm{Inf}(\tau_B) \cap F_B \neq \emptyset$. Without loss of generality, we write $\tau_{\otimes} = p^{pre}(p^{suf})^{\omega}$ in the prefix-suffix structure, where

$$p^{pre} = ((q_0, q_0'), q_{0,B}) \cdots ((q_n, q_n'), q_{n,B})$$
$$p^{suf} = ((q_{n+1}, q_{n+1}'), q_{n+1,B}) \cdots ((q_{n+m}, q_{n+m}'), q_{n+m,B})$$

and $q_{n+1,B} \in F_B$. Since $q_0' = \tilde{\tau}'(1) \notin Q_S$, we know that $\tilde{q}_I := ((q_0, q_0'), q_{0,B}) \in \textsc{Int}_{q_0}(T_{\otimes})$. Furthermore, we have $\tilde{q}_G := ((q_{n+1}, q_{n+1}'), q_{n+1,B}) \in \mathrm{Reach}(\{\tilde{q}_I\}) \cap \textsc{Goal}(T_{\otimes})$. However, $\hat{J}(\tau^{\tilde{q},\tilde{q}_G}[\tau^{\tilde{q}_G,\tilde{q}_G}]^{\omega}) = \hat{J}(\tilde{\tau}) < \hat{J}(\tau)$. This means that Algorithm 1 should at least output $\tau^{\tilde{q},\tilde{q}_G}[\tau^{\tilde{q}_G,\tilde{q}_G}]^{\omega}$ rather than $\tau$, which is a contradiction. □

The above three propositions show that the proposed algorithm is sound in the sense that the solution is correct if it finds one. Note that Algorithm 1 may return "no feasible plan from $q_0$". Next we show that the proposed algorithm is also complete.

**Proposition 8.** *If Algorithm 4–1 returns "no feasible plan from $q_0$", then no solution to Problem 1 exists.*

***Proof.*** The proof is similar to the proof of Proposition 7. Suppose, for the sake of contraposition, that there exists a secure path $\tau \in \mathrm{Path}^{\omega}(T)$ such that $\mathrm{trace}(\tau) \models \phi$. Following the same argument in the proof of Proposition 7, there exists a path $\tau_{\otimes} \in \mathrm{Path}^{\omega}(T_{\otimes})$, which is in the form of $\tau_{\otimes} = ((\tau, \tau'), \tau_B)$ such that $\tau'(1) \notin Q_S$ and $\mathrm{Inf}(\tau_B) \cap F_B \neq \emptyset$. Therefore, we have $\tilde{q}_I := ((\tau(1), \tau'(1)), \tau_B(1)) \in \textsc{Int}_{q_0}(T_{\otimes})$. Furthermore, $\mathrm{Reach}(\{\tilde{q}_I\}) \cap \textsc{Goal}(T_{\otimes}) \neq \emptyset$ since $\mathrm{Inf}(\tau_B) \cap F_B \neq \emptyset$. Therefore, Algorithm 1 will not return "no feasible plan from $q_0$". □

Finally, we summarize Propositions 5, 6, 7 and 8 by the following theorem.

**Theorem 6.** *For any WTS $T = (Q, Q_0, \rightarrow, w, \mathcal{AP}, L)$ with output function $H : Q \rightarrow Y$, secret states $Q_S$ and LTL formula $\phi$, Algorithm 4–1 correctly solves the optimal security-aware LTL planning problem defined in Problem 1.*

***Proof.*** The soundness of the algorithm is established by Propositions 5, 6 and 7, and Proposition 8 shows the completeness of the algorithm. □

## 4.6 Conclusion

In this chapter, we solved a security-aware optimal path planning problem for linear temporal logic tasks. A polynomial-time algorithm was proposed based on the product of the twin-system and the Büchi automaton. The synthesized solution is *secure-by-construction* in the sense that it provides provably security guarantees for the designed systems. Note that, in this work, we consider security requirement for protecting the initial secret of the system. In the future, we would like to extend the proposed algorithm to other types of security, e.g., infinite-step opacity. Also, we are interested in investigating optimal LTL path planning for multi-robot systems with security guarantees.

# Chapter 5 Case Studies

In this chapter, we present two case studies on security-aware path planning. One is related with protecting robot's intention. Specifically, we model its intention by utilizing the notion of $K$-step pre-opacity proposed in Chapter 3. The other one aims at protecting robot's starting point. The sound and complete algorithm developed in Chapter 4 is illustrated in this case.

## 5.1 Intention-Security-Aware Path Planning

In some applications, the "secret" one wants to hide can be its *intention* to do something of particular importance in the future. As a simple example, let us consider a single robot moving in a region whose mobility is described by a DES shown in Figure 5–1, where each state represents a location and each transition represents an action. Some actions are assumed to be observable by outsider; $E_o = \{o_1, o_2, o_3\}$ are observable actions.

The robot may choose to attack state 9 by reaching it. However, it does not want to reveal its intention to attack state 9 too early; otherwise, e.g., some defense strategy can be implemented in advance. Clearly, the shortest path to reach state 9 is $0 \xrightarrow{o_1} 3 \xrightarrow{o_2} 6 \xrightarrow{o_3} 9$. However, by doing so, the outsider will know the robot's intention of attack two steps ahead just by observing the first action $o_1$. On the other hand, the robot can choose to attack state 9 via path $0 \xrightarrow{u_2} 2 \xrightarrow{o_2} 5 \xrightarrow{o_1} 8 \xrightarrow{o_2} 11 \xrightarrow{o_3} 9$, which is longer but allows the robot to hide its intention of visiting state 9 until it actually reaches it. This is because this path has the same observation of $0 \xrightarrow{u_1} 1 \xrightarrow{o_2} 4 \xrightarrow{o_1} 7 \xrightarrow{o_2} 10$ whose continuation may not necessarily be secret.

Existing notions of opacity in the literature cannot capture this scenario as this problem essentially requires another type of opacity for *future information*: the user does not want the outsider to know too early for sure that it *will do* something secret at some future instant. We may characterize this scenario by the notion of pre-opacity. Specifically, based on the verification theorems developed in Chapter 3, we can verifiablely check that this system is 3-step instant pre-opaque but not 2-step instant pre-opaque; it is 3-step trajectory pre-opaque but not 2-step trajectory pre-opaque.

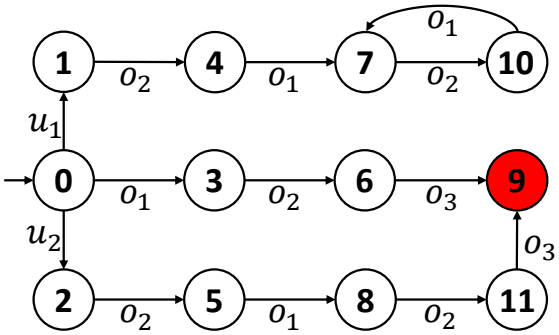



Figure 5–1 A motivating example with $E_o = \{a, b, c\}$ and $E_{uo} = \{u\}$. State 6 is the target (secret) state.

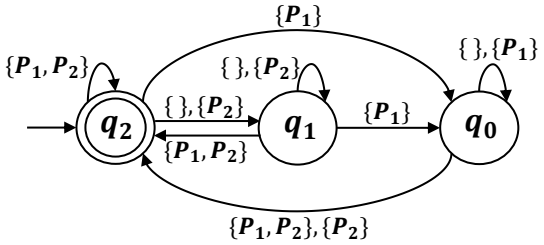



Figure 5–2 An NBA translated from $\phi = \Box\Diamond P_1 \wedge \Box\Diamond P_2$.

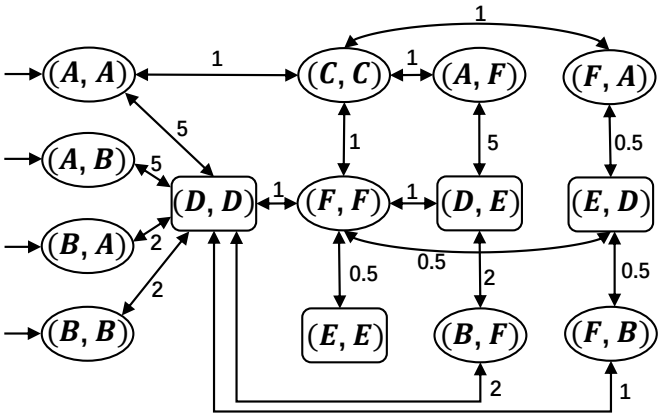



Figure 5–3 Twin-WTS $V$ of $T$ in Figure 4–2.

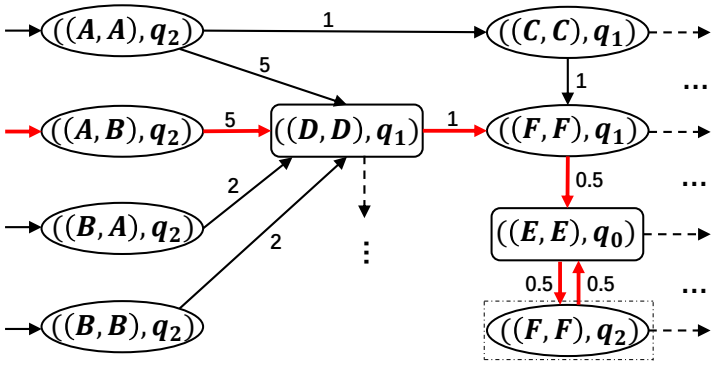



Figure 5–4 Example of the construction of the $T_{\otimes}$. Red transitions represent the optimal feasible path. Due to limited space, some states and transitions are omitted and part of the product system is shown.

## 5.2 Initial-State-Security-Aware Path Planning

We go back to the motivating example in Section 4.2 to illustrate the proposed planning algorithm. Consider again the WTS in Figure 4–2. To formalize the LTL task, we consider two atomic propositions $\mathcal{AP} = \{P_1, P_2\}$ with labeling function $L : Q \to 2^{\mathcal{AP}}$ defined by $L(F) = \{P_1\}$, $L(E) = \{P_2\}$ and $L(q) = \emptyset$ for other states. Then the task of the robot is captured by

$$\phi = \square\lozenge P_1 \wedge \square\lozenge P_2.$$

The observation mapping is $H : Q \to \{Grass, Sand\}$ as specified in Figure 4–2. We define $Q_S = \{A\} \subseteq Q$, i.e., state $A$ is the unique secret initial state.

To achieve the planning task, first we convert $\phi$ to NBA $B = (Q_B, Q_{0,B}, \Sigma, \to_B, F_B)$, which is shown in Figure 5–2; such an conversion can be done by, e.g., the tool developed in [78]. Then we construct the corresponding twin-WTS $V$, which is shown in Figure 5–3. Specifically, $V$ contains four initial states $(A, A), (B, B), (A, B)$ and $(B, A)$ since $H(A) = H(B) = Sand$ and four combination are all valid initial states. Then, for example, starting from $(A, B)$, only state $(D, D)$ can be reached as $A \to D, B \to D$ and $H(D) = H(D) = Grass$. Also, from state $(C, C)$, we can reach $(A, F)$ as $C \to A, C \to F$ and $H(A) = H(F) = Sand$. Finally, we need to construct the product system $T_{\otimes}$; for the sake of simplicity, we just show part of $T_{\otimes}$ in Figure 5–4, which is sufficient for the purpose of planning.

Now, we assume that the robot is starting from secret initial state $A$. Then we have $\textsc{Int}_A(T_{\otimes}) = \{((A, B), q_2)\}$, which is a singleton. Also, we have $((F, F), q_2) \in \mathrm{Reach}(\{((A, B), q_2)\}) \cap \textsc{Goal}(T_{\otimes})$. One can check that such a state pair is indeed the one that minimizes the cost function if we draw the complete product system. There-

fore, we obtain an optimal plan $\tau = \Pi[((A,B),q_2)((D,D),q_1)((F,F),q_1)(((E,E),q_0)((F,F),q_2))^\omega] = AD(FE)^\omega$, which is highlighted by red transitions in Figure 5–4.

# Chapter 6 Summary

## 6.1 Conclusion

In this dissertation, we focus on the notion of opacity in DES and solve two problems: pre-opacity verification problem and security-aware planning problem.

First, we investigate opacity from a new angle by considering the system' s intention of executing some particular behavior as the secret. Then we propose a new type of opacity, called pre-opacity, to characterize whether or not the secret intention of the system can be revealed. Two notions of pre-opacity called $K$-step instant pre-opacity and $K$-step trajectory pre-opacity are proposed. For each notion of pre-opacity, a verifiable necessary and sufficient condition as well as an effective verification algorithm is provided. We show that both properties are PSPACE-hard. We also generalize the notions of pre-opacity to the case where the secret behavior is captured by a sequence pattern. Our work extends the theory of opacity to a new class where secret is related to the intention of the system.

Second, we solve a security-aware optimal path planning problem for linear temporal logic tasks. A polynomial-time algorithm is proposed based on the product of the twin-system and the Büchi automaton. The synthesized solution is secure-by-construction in the sense that it provides provably security guarantees for the designed systems against temporal logic tasks.

## 6.2 Future Work

We believe there are many interesting future directions related to the concept of preopacity. One interesting direction is to synthesize a supervisor to enforce pre-opacity when the verification result is negative. Also, we would like to extend the notion of pre-opacity to the stochastic setting to quantitatively evaluate the information leakage.

Note that, in Chapter 4, we consider security requirement for protecting the initial secret location of the system. In the future, we would like to extend the proposed algorithm to other types of security, e.g., infinite-step opacity. Also, we are interested in investigating optimal LTL path planning for multi-robot systems with security guarantees.

# Bibliography

[1] Bryans J, Koutny M, Mazaré L, Ryan P. Opacity generalised to transition systems[J]. Internationa Journal of Information Security, 2008, 7(6): 421-435.

[2] Bryans J, Koutny M, Ryan P. Modelling opacity using Petri nets[J]. Electronic Notes in Theoretical Computer Science, 2005, 121: 101-115.

[3] 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.

[4] Saboori A, Hadjicostis C N. Verification of initial-state opacity in security applications of discrete event systems[J]. Information Sciences, 2013, 246: 115-132.

[5] Yin X, Lafortune S. A new approach for the verification of infinite-step and $K$-step opacity using two-way observers[J]. Automatica, 2017, 80: 162-171.

[6] 戴维, 刘富春, 赵锐, 邓秀勤, 崔洪刚. 基于状态估计的分布式离散事件系统可诊断性研究[J]. 工业工程, 2021, 24(1): 123.

[7] Zhang K, Zamani M. Infinite-step opacity of nondeterministic finite transition systems: A bisimulation relation approach[C]. in: 56th IEEE Conference on Decision and Control. 2017: 5615-5619.

[8] Tong Y, Li Z, Seatzu C, Giua A. Verification of state-based opacity using Petri nets[J]. IEEE Trans. Automatic Control, 2017, 62(6): 2823-2837.

[9] Tong Y, Li Z, Seatzu C, Giua A. Decidability of opacity verification problems in labeled Petri net systems[J]. Automatica, 2017, 80: 48-53.

[10] Cong X, Fanti M, Mangini A, Li Z. On-line verification of current-state opacity by Petri nets and integer linear programming[J]. Automatica, 2018, 94: 205-213.

[11] 阙蔡雄, 刘富春, 赵锐, 邓秀勤, 崔洪刚. 基于 Petri 网诊断器的离散事件系统模式故障的在线诊断[J]. 控制理论与应用, 2020.

[12] Ramasubramanian B, Cleaveland W, Marcus S. Notions of Centralized and Decentralized Opacity in Linear Systems[J]. IEEE Transactions on Automatic Control, 2020, 265(4): 1442-1455.

[13] An L, Yang G H. Opacity Enforcement for Confidential Robust Control in Linear Cyber-Physical Systems[J]. IEEE Transactions on Automatic Control, 2020, 65(3): 1234-1241.

[14] Yin X, Zamani M, Liu S. On approximate opacity of cyber-physical system[J]. IEEE Transactions on Automatic Control, 2020.

[15] 刘富春, 张旭, 赵锐, 等. 不完备离散事件系统的当前状态不透明性[J]. 控制理论与应用, 2019(2019 年 07): 1067-1071.

[16] Takai S, Oka Y. A formula for the supremal controllable and opaque sublanguage arising in supervisory control[J]. SICE J. Control, Measu. & Syst. Integration, 2008, 1(4): 307-311.

[17] Dubreil J, Darondeau P, Marchand H. Supervisory control for opacity[J]. IEEE Trans. Automatic Control, 2010, 55(5): 1089-1100.

[18] Cassez F, Dubreil J, Marchand H. Synthesis of opaque systems with static and dynamic masks[J]. Formal Methods in System Design, 2012, 40(1): 88-115.

[19] Darondeau P, Marchand H, Ricker L. Enforcing opacity of regular predicates on modal transition systems[J]. Discrete Event Dyn. Sys.: Theory & Apl., 2014, 25(1-2): 251-270.

[20] Zhang B, Shu S, Lin F. Maximum Information Release While Ensuring Opacity in Discrete Event Systems[J]. IEEE Trans. Automation Science and Engineering, 2015, 12(4): 1067-1079.

[21] Ji Y, Wu Y C, Lafortune S. Enforcement of opacity by public and private insertion functions[J]. Automatica, 2018, 93: 369-378.

[22] Behinaein B, Lin F, Rudie K. Optimal Information Release for Mixed Opacity in Discrete-Event Systems[J]. IEEE Transactions on Automation Science and Engineering, 2019, 16(4): 1960-1970.

[23] Saboori A, Hadjicostis C. Coverage analysis of mobile agent trajectory via state-based opacity formulations[J]. Control Engineering Practice, 2011, 19(9): 967-977.

[24] Wu Y C, Sankararaman K, Lafortune S. Ensuring privacy in location-based services: An approach based on opacity enforcement[C]. in: 12th International Workshop on Discrete Event Systems. 2014: 33-38.

[25] Bourouis A, Klai K, Ben Hadj-Alouane N, El Touati Y. On the Verification of Opacity in Web Services and Their Composition[J]. IEEE Transactions on Services Computing, 2017, 10(1): 66-79.

[26] Jacob R, Lesage J J, Faure J M. Opacity of discrete event systems: models, validation and quantification[J]. IFAC-PapersOnLine, 2015, 48(7): 174-181.

[27] Lafortune S, Lin F, Hadjicostis C. On the history of diagnosability and opacity in discrete event systems[J]. Annual Reviews in Control, 2018, 45: 257-266.

[28] Lin F. Opacity of discrete event systems and its applications[J]. Automatica, 2011, 47(3): 496-503.

[29] Wu Y C, Lafortune S. Comparative analysis of related notions of opacity in centralized and coordinated architectures[J]. Discrete Event Dynamic Systems, 2013, 23(3): 307-339.

[30] Saboori A, Hadjicostis C N. Verification of infinite-step opacity and complexity considerations[J]. IEEE Transactions on Automatic Control, 2011, 57(5): 1265-1269.

[31] Falcone Y, Marchand H. Enforcement and validation (at runtime) of various notions of opacity[J]. Discrete Event Dynamic Systems, 2015, 25(4): 531-570.

[32] Saboori A, Hadjicostis C. Current-state opacity formulations in probabilistic finite automata[J]. IEEE Transactions on Automatic Control, 2014, 59(1): 120-133.

[33] Bérard B, Chatterjee K, Sznajder N. Probabilistic opacity for Markov decision processes[J]. Information Processing Letters, 2015, 115(1): 52-59.

[34] Keroglou C, Hadjicostis C. Probabilistic system opacity in discrete event systems[J]. Discrete Event Dynamic Systems, 2017: 1-26.

[35] Chen J, Ibrahim M, Kumar R. Quantification of Secrecy in Partially Observed Stochastic Discrete Event Systems.[J]. IEEE Trans. Automation Science and Engineering, 2017, 14(1): 185-195.

[36] Wu B, Lin H. Privacy Verification and Enforcement via Belief Abstraction[J]. IEEE Control Sys. Letters, 2018, 2(4): 815-820.

[37] Yin X, Li Z, Wang W, Li S. Infinite-step opacity and $K$-step opacity of stochastic discrete-event systems[J]. Automatica, 2019, 99: 266-274.

[38] LaValle S. Planning algorithms[M]. Cambridge university press, 2006.

[39] Karaman S, Frazzoli E. Sampling-based algorithms for optimal motion planning[J]. The international Journal of Robotics Research, 2011, 30(7): 846-894.

[40] Fainekos G, Girard A, Kress-Gazit H, Pappas G. Temporal logic motion planning for dynamic robots[J]. Automatica, 2009, 45(2): 343-352.

[41] Wongpiromsarn T, Topcu U, Murray R. Receding horizon temporal logic planning[J]. IEEE Transactions on Automatic Control, 2012, 57(11): 2817-2830.

[42] Kress-Gazit H, Lahijanian M, Raman V. Synthesis for robots: Guarantees and feedback for robot behavior[J]. Annual Review of Control, Robotics, and Autonomous Systems, 2018, 1: 211-236.

[43] Kloetzer M, Mahulea C. Path planning for robotic teams based on LTL specifications and Petri net models[J]. Discrete Event Dynamic Systems, 2020, 30(1): 55-79.

[44] 肖云涛, 欧林林, 俞立. 基于线性时序逻辑的最优巡回路径规划[J]. 自动化学报, 2014, 40(10): 2126-2133.

[45] 焦梦甜, 宋运忠. 线性时序逻辑约束下的滚动时域控制路径规划[J]. 智能系统学报, 2020, 15(2): 281-288.

[46] Baier C, Katoen J. Principles of model checking[M]. MIT press, 2008.

[47] Li L, Bayuelo A, Bobadilla L, Alam T, Shell D. Coordinated multi-robot planning while preserving individual privacy[C]. in: International Conference on Robotics and Automation (ICRA). 2019: 2188-2194.

[48] Zheng H, Panerati J, Beltrame G, Prorok A. An Adversarial Approach to Private Flocking in Mobile Robot Teams[J]. IEEE Robotics and Automation Letters, 2020, 5(2): 1009-1016.

[49] Smith S, Tumova J, Belta C, Rus D. Optimal path planning for surveillance with temporal-logic constraints[J]. The International Journal of Robotics Research, 2011, 30(14): 1695-1708.

[50] Guo M, Dimarogonas D. Multi-agent plan reconfiguration under local LTL specifications[J]. The International Journal of Robotics Research, 2015, 34(2): 218-235.

[51] Ulusoy A, Smith S, Ding X, Belta C, Rus D. Optimality and robustness in multi-robot path planning with temporal logic constraints[J]. The International Journal of Robotics Research, 2013, 32(8): 889-911.

[52] Li L, Fu J. Sampling-based approximate optimal temporal logic planning[C]. in: IEEE International Conference on Robotics and Automation (ICRA). 2017: 1328-1335.

[53] Kantaros Y, Zavlanos M. Sampling-based optimal control synthesis for multirobot systems under global temporal tasks[J]. IEEE Transactions on Automatic Control, 2018, 64(5): 1916-1931.

[54] Wolff E, Topcu U, Murray R. Robust control of uncertain Markov decision processes with temporal logic specifications[C]. in: 51st IEEE Conference on Decision and Control (CDC). 2012: 3372-3379.

[55] Ding X, Smith S, Belta C, Rus D. Optimal control of Markov decision processes with linear temporal logic constraints[J]. IEEE Transactions on Automatic Control, 2014, 59(5): 1244-1257.

[56] Deng K, Chen Y, Belta C. An approximate dynamic programming approach to multiagent persistent monitoring in stochastic environments with temporal logic constraints[J]. IEEE Transactions on Automatic Control, 2017, 62(9): 4549-4563.

[57] Guo M, Zavlanos M. Probabilistic motion planning under temporal tasks and soft constraints[J]. IEEE Transactions on Automatic Control, 2018, 63(12): 4051-4066.

[58] Hadjicostis C. Trajectory Planning under Current-State Opacity Constraints[C]. in: 14th IFAC Workshop on Discrete Event Systems (WODES). 2018: 337-342.

[59] Clarkson M, Schneider F. Hyperproperties[J]. Journal of Computer Security, 2010, 18(6): 1157-1210.

[60] Clarkson M, Finkbeiner B, Koleini M, Micinski K, Rabe M, Sánchez C. Temporal logics for hyperproperties[C]. in: International Conference on Principles of Security and Trust. 2014: 265-284.

[61] Wang Y, Nalluri S, Pajic M. Hyperproperties for Robotics: Motion Planning via HyperLTL[C]. in: IEEE International Conference on Robotics and Automation (ICRA). 2020.

[62] Saboori A, Hadjicostis C. Opacity-enforcing supervisory strategies via state estimator constructions[J]. IEEE Trans. Automatic Control, 2011, 57(5): 1155-1165.

[63] Yin X, Lafortune S. A uniform approach for synthesizing property-enforcing supervisors for partially-observed discrete-event systems[J]. IEEE Trans. Automatic Control, 2016, 61(8): 2140-2154.

[64] Tong Y, Li Z, Seatzu C, Giua A. Current-state opacity enforcement in discrete event systems under incomparable observations[J]. Discrete Event Dynamic Systems: Theory & Appllications, 2018, 28(2): 161-182.

[65] Yang S, Yin X. Secure Your Intention: On Notions of Pre-Opacity in Discrete-Event Systems[J]. ArXiv preprint arXiv:2010.14120, 2020.

[66] Yang S, Yin X, Li S, Zamani M. Secure-by-Construction Optimal Path Planning for Linear Temporal Logic Tasks[C]. in: 2020 59th IEEE Conference on Decision and Control (CDC). 2020: 4460-4466.

[67] Kloetzer M, Belta C. Temporal logic planning and control of robotic swarms by hierarchical abstractions[J]. IEEE Transactions on Robotics, 2007, 23(2): 320-330.

[68] Belta C, Yordanov B, Gol E. Formal methods for discrete-time dynamical systems[M]. Springer, 2017.

[69] Jéron T, Marchand H, Genc S, Lafortune S. Predictability of sequence patterns in discrete event systems[C]. in: Proc. 17th IFAC World Congress. 2008: 537-543.

[70] Genc S, Lafortune S. Predictability of event occurrences in partially-observed discrete-event systems[J]. Automatica, 2009, 45(2): 301-311.

[71] Kumar R, Takai S. Decentralized prognosis of failures in discrete event systems[J]. IEEE Trans. Autom. Contr., 2010, 55(1): 48-59.

[72] Takai S. Robust prognosability for a set of partially observed discrete event systems[J]. Automatica, 2015, 51: 123-130.

[73] Chen J, Kumar R. Stochastic Failure Prognosability of Discrete Event Systems[J]. IEEE Trans. Autom. Contr., 2015, 60(6): 1570-1581.

[74] Yin X, Li Z J. Decentralized fault prognosis of discrete event systems with guaranteed performance bound[J]. Automatica, 2016, 69: 375-379.

[75] Sedgewick R, Wayne K. Algorithms[M]. 4th. Addison-Wesley Professional, 2011.

[76] Jéron T, Marchand H, Pinchinat S, Cordier M. Supervision patterns in discrete event systems diagnosis[C]. in: 8th International Workshop on Discrete Event Systems. 2006: 262-268.

[77] Vardi M, Wolper P. An automata-theoretic approach to automatic program verification[C]. in: Proceedings of the First Symposium on Logic in Computer Science. 1986: 322-331.

[78] Gastin P, Oddoux D. Fast LTL to Büchi automata translation[C]. in: International Conference on Computer Aided Verification (CAV). 2001: 53-65.

# Acknowledgements

In retrospect, I had never expected my undergraduate journey to be such a wonderful experience. This four-year time is unforgettable. I am very grateful for receiving a lot of help from my professors, friends, and family.

First and foremost, I would like to express my sincere gratitude to my advisor, Professor Xiang Yin, for his guidance and support over the last two years. Xiang gave me so much freedom to explore any project I am interested in. His creativity, enthusiasm, and perfectionism about research helped shape my personality as a researcher. None of my achievements would be possible without his help.

I would like to express my sincere thanks to Professor Shaoyuan Li for his advice and guidance on research. I have learned a lot from him and I got to know how a great researcher think and solve problems through collaboration with him. In addition, His course *Automatic Control Principle* is very interesting and impressive to me!

I am very grateful to have my dissertation committee: Professor Cailian Chen, Professor Yv Qiao, Professor Liang Gong, and Professor Zhichen Li. They have provided invaluable suggestions and feedback for refining this dissertation. I got to learn how to solve problems better and present results better through communication with them.

Many thanks to Professor Michael M. Zavlanos at Duke University, who gave me so much insightful instructions while working with him last summer. He is a mature researcher and is always a model for me. I am also very grateful for his invaluable help for my PhD application. I also would like to thank Professor Majid Zamani at University of Colorado Boulder for his support and encouragements for my PhD application. I had a great time with him during our research collaboration.

I am very thankful to Dr. Xusheng Luo and Junyao Hou for their discussions on formal methods and other research topics. They help me have a deeper understanding of my research. I also would like to thank Zihao Li, who is my partner in many course projects. I sincerely appreciate his kind help when I have some trouble in those projects.

Last but not least, I sincerely thank my parents and my sister for their unconditional love and support. I love them.

# Publications

[1] YANG S AND YIN X. Secure your intention: On notions of pre-opacity in discrete-event systems[J]. IEEE Transactions on Automatic Control, Conditionally Accepted, 2021.

[2] YANG S, HOU J, YIN X, AND LI S. Opacity of networked supervisory control systems over insecure communication channels[J]. IEEE Transactions on Control of Network Systems, Accepted, 2021.

[3] YANG S, YIN X, LI S, AND ZAMANI M. Secure-by-construction optimal path planning for linear temporal logic tasks[C]. in: 59th IEEE Conference on Decision and Control (CDC), 4460-4466, 2020.