Proceedings of the 3rd International Conference on Signal Processing and Machine Learning
DOI: 10.54254/2755-2721/6/20230783

Opacity verification in discrete event system

Qingxian Liu

Faculty of Science, University of British Columbia, Vancouver, British Columbia, V6T1K2, Canada

qliu20@student.ubc.ca

Abstract. The increasing data frequency and size of information flow have aroused people's concern about their private information safety. For a system, it is necessary to verify whether the system is opaque to external invaders. A plant is said to be opaque if any secret behaviour in the plant cannot be detected by an external invader. This paper focuses on illustrating a method to decide whether a given system is opaque. To solve the question, the system is first abstracted as the model of finite state automata. Then this paper studies the safety property from the perspective of current-state opacity of a given automaton.

Keywords: Opacity, Discrete event systems.

1. Introduction

With the increasing popularity of online services such as e-commercial, cryptocurrency, search engines, and social media, a massive amount of private or protected information is collected and switched from end to end during the process of this interaction. Being used to indicate this process of transferring information from one variable to the other, the terminology information flow is introduced and defined as the information transmission from a high-level user to a low-level user [1]. This paper focuses on the opacity of information flow where two main aspects are included and widely explored in recent years: anonymity and secrecy [2].

The anonymity of information flow is studied by Dingledineet et al. [3] and Lin [2]. Dingledine et al. [3] explore the relationship between reputation and anonymity. It especially focuses on exploring the anonymous uploading and anonymous forwarder. Moreover, it gives a potential explanation about why these will affect the reputation of the system. In Lin [2], anonymity is defined as the ability to conceal an action from all actions where strong anonymity and weak anonymity are introduced. The system can be imagined as being covered by an opaque black box. Strong anonymity is described in a way that, for all the actions executed by the system, the observer cannot deduce which action is triggered in detail, while weak anonymity is described in a way that, based on some/partial actions executed by the system, the observer can deduce which action is triggered.

The second aspect is secrecy which has been studied by many researchers. In the research of Badouel et al. [4], "secret" is defined as a subset of system's trajectories. They test whether the secrecy is preserved by checking whether an external observer can find out any "secret". In Lin [2], similar to the previous definitions of strong anonymity and weak anonymity, the research proposed the concepts of weak secrecy and strong secrecy.

DOI: 10.54254/2755-2721/6/20230783

In Jacob et al. [5], both anonymity and secrecy problems can be included in opacity issues. Similar to their opinions, in this paper, the opacity of a discrete system is described as whether an external observer can deduce the hidden behaviors without knowing the detailed internal operation of this system. The observer can only estimate system behaviors through limited observations. The system will be defined as strong opaque if and only if, any existent hidden behaviors, there always exists at least one non-hidden behavior that encounters and outputs the same observations with it. On the other hand, the system will be defined as weak-opaque if, for some hidden behaviors, there exist non-hidden behaviors that can output the same observation.

Opacity is a comparatively new research region. It is first introduced and explored in the region of computer science in Mazaré's work [6], where it is used to assist in analysing cryptographic protocols. After that, more and more relevant regions have been continuously explored. Later, Bryans et al. [7] first prove that, for bounded labelled Petri nets, it is decidable to verify the state-based opacity. Based on that, Tong et al. [8] explore an efficient way to tackle current-state and initial-state opacity problems in bounded labelled Petri nets by using the basis reachability graph (BRG) which is a condensed model of the reachability graph. Except for using the Petri nets, finite state automaton is also a popular model to deal with state-based opacity. Saboori et al. [9] study the verification of k-step opacity, where two verification methods based on two different state elimination methods are analysed, and the corresponding complexity is calculated as well. Moreover, Saboori et al. [10] focus on the current-state opacity, where they extended the formulation of current-state opacity to the model of probabilistic finite automata. Moreover, three probabilistic notions of opacity were also introduced in [10].

This work mainly focuses on exploring the verification method of the opacity by using the model of finite state automata. System opacity is defined as a measurement to decide whether or not the system is opaque. The system is described as opaque if the system's entry into a collection of secret states is opaque (uncertain) to outsiders. The paper is organized as the following sections. In the section of preliminaries, relevant notions used in the paper is recalled. In the section of method, the problem formulation is introduced, and opacity verification approach is proposed. Eventually, the summary and future development will be discussed in Section 4.

2. Preliminaries

E is defined as the alphabet of the events. E^* is a set containing composite string with limited length that is composed of the elements in the alphabet. The empty string represented by symbol ε is also included. A language $L \subseteq E^*$ is denoted as the language consists of stings with finite length in E. As for the string s, symbol | s | depicts the length of s. A finite-state automaton is defined as G = (Q, E, f, q_0). It contains an initial state set Q_0, a transition function $f : Q \times E \to Q$, the finite events set E and the finite states set Q = {0, 1, . . .}. The transition function f is able to be generalized to the domain $Q \times E^* \to Q$ which can be denoted as $f^*: Q \times E^* \to Q$. The language accepted by the system G defined by $L(G, Q_0)$ describes the internal behaviors of system where $L(G, Q_0) = \{v \in E^* | \exists\, q \in Q_0, f(q, v)\ is\ defined\}$. As the number of initial states Q_0 in describing the opacity problem is unknown, in that case, Q_0 is defined as a set of initial states here instead of just a single state. Moreover, the system that is discuss in this paper is limited observable, which means that only part of events can be directly observed from the perspective of the external observer. The event set can be divided into two sets: the observable set E_o and the unobservable set E_{uo}. For any given input string $s \in E^*$, the corresponding observation of s ought to be the output of the projection function $P: E^* \to E_o^*$, where P (ne) = P (n) P (e), $n \in E^*$ and $e \in E$. If $e \in E_o$, the output of projection function of any events P (e) = e, while P (e) = ε if $e \in E_{uo} \cup \{\varepsilon\}$. Ultimately, for the state estimate function $C(\omega) = \{q \mid P(\sigma) \in \omega, f(q_0, \sigma) = q\}$ where $q \in Q$, $q_0 \in Q_0$, $\sigma \in L(G, Q_0)$, $\omega \in E_o^*$. A simple example can be taken to illustrate all those terminologies in a more intuitive way.

Figure 1 is an automaton G = (Q, E, f, q_0), where Q is the set $\{q_0, q_1, q_2, q_3, q_4, q_5\}$ contains all the states, E is the event set{a, b, c, d, e} and the initial state is state q_0. As for the state transition function f, starting from the initial state, there are $f(q_0, a) = q_1$. Starting from q_1, there are $f(q_1, c) = q_3$ and

DOI: 10.54254/2755-2721/6/20230783

$f(q_1, b) = q_2$. Starting from q_1, there are $f(q_2, e) = q_5$. Moreover, starting from state q_3, there are $f(q_3, d) = q_4$.

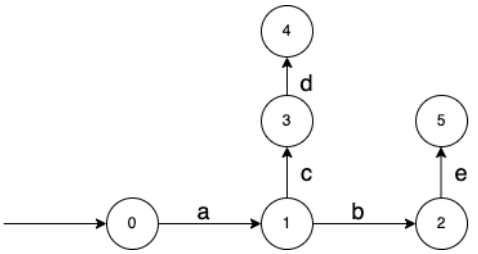

Figure 1. An automaton model.

3. Opacity verification

In our lives, our personal data are collected and exchanged in various ways by all kinds of systems, ranging from mobile app operation systems to banking deposit systems. It is important and essential to make sure that those systems are robust enough to keep our data safe. In that case, with the increasing amount of information flow that people share, the worries about data safety especially system reliability has raised as well. The opacity of the system becomes a significant factor in measuring and verifying the safety of given systems. One intuitive way to describe opacity is whether an external observer with all the necessary knowledge can find the secret actions just from limited observations. Most of the systems will hide corresponding internal steps when they are processing secret information to avoid them be observed from external observers. Thus, in order to better test the opacity of a system, it is assumed that the external observer can only observe limited behaviors.

Besides, as there are various kinds of design logic behind each system that contain a lot of details and they are quite different from each other, it is necessary to find an appropriate model to generalize the basic characters of those systems and mitigate the weights of their unique designs. Based on that, the finite state automata become an ideal choice. Symbolling all kinds of complex middleware as the set of states and representing the transmission of information flow by using the transition function, the finite automata can simulate the basic operation theorem of most systems. Therefore, this paper aims at illustrating a method to decide whether a given system is opaque based on the model of finite state automata.

Based on the notations mentioned in preliminary, let us initiate a finite-state automata $G = (Q, E, f, q_0)$, a secrete states set $X_S \subseteq X$, a non-secrete states set $X_{NS} \subseteq X$, and a projection P. Then the opacity of G can be depicted as whether for any $q \in q_0$ and for any $t \in L(G, q)$ so that $f(q, t) \in X_S$, for some $q' \in q_0$ and for some $t' \in L(G, q')$ so that $f(q, t) \in X_{NS}$ and $P(t) = P(t')$.

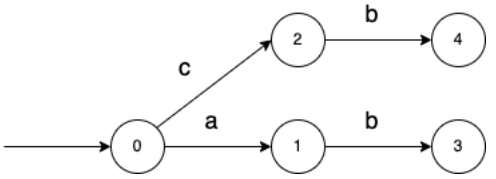

Figure 2. An automaton model.

An intuitive example is used in Figure 2 to better illustrate the previous description. Assume that the secret states set $X_S = \{3\}$, the observable events set $E_O = \{b\}$, and the set of non-secret states $X_{NS} = \{0, 1, 2, 4\}$. For the external observer, since one cannot observe either a or c, it is impossible to distinguish whether the actual event is ab or cb when the behavior b is observed. In that case, the system automaton G is current-state opaque as the external observer is not able to distinguish which state (state 3 or 4) is the current state.

DOI: 10.54254/2755-2721/6/20230783

However, when the observable events $E_O = \{a, b\}\ or\ \{c, b\}$. Take $E_O = \{c, b\}$ as an example, since one now has the views of both c and a, one can easily tell the difference between ab and cb. When the behaviour cb is observed, it is known to the observer that system is currently in state 3, otherwise, it is in state 4. In that case, the system automaton G is not tested as current-state opaque as the external observer can distinguish the current state of G just by those observable events.

Let's generalize it to the level of language. Based on the previous assumption, let's assume that the secrete language $L_S \subseteq L(G, q_0)$ and the non-secrete language $L_{NS} \subseteq L(G, q_0)$. G is defined as language-based opaque when for any string $s \in L_S$, there always exist other strings $s' \in L_{NS}$ so that $P(s) = P(s')$.

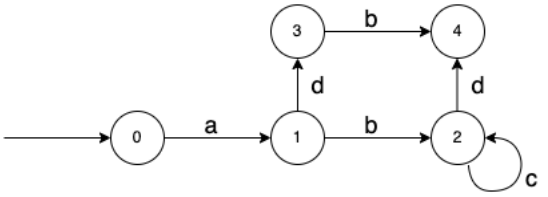

Figure 3. An automaton model.

Another intuitive example in Figure 3 to better illustrate the previous description. Assume the observable events set $E_O = \{a, b, c\}$. When $L_S = \{abd\}$ and $L_{NS} = \{abc^*d\}$, the external observer cannot tell whether the string is abd or adb when $P(L_S) = \{ab\}$ is observed. Thus, it is language based opaque. However, if it is assumed that $L_S = \{abcd\}$ and $L_{NS} = \{adb, abd\}$, it is easy to find out that there does not exist any string in L_S has the same output as the one in L_{NS}. In that case, it is not language-based opaque.

4. Conclusion

With the increasing complexity of current system, an abstract model that is equivalent but easier for opacity verification is increasingly needed. Based on implementing the previous model of finite automata, a method is developed to test the opacity of the given system. If a plant is opaque to an intruder, then the secret behavior in the system cannot be leaked and the private data can be properly protected.

With the significant development of network packet transfer, big data, transaction transfer and so on, more and more private data flow are switched and also exposed to the intended external observers. Based on that, the needs of more secured and robust systems have been increasingly arisen. The approach to test and then improves the system security has gradually become a widely discussed topic. Opacity could be the potential answer in assisting solving the system security issue, which could help designers analyse the system in a lucid and efficient way.

References

[1] Lowe G 2002 Quantifying information flow In *Proceedings 15th IEEE Computer* Security *Foundations Workshop. CSFW-15* pp 18-31
[2] Lin F 2011 Opacity of discrete event systems and its applications *Automatica* **47** 496-503
[3] Dingledine R, Mathewson N and Syverson P 2003 Reputation in p2p anonymity systems In *Workshop on economics of peer-to-peer systems* pp 1-5
[4] Badouel E, Bednarczyk M, Borzyszkowski A, Caillaud B and Darondeau P 2007 Concurrent secrets. *Discrete Event Dynamic Systems* **17** 425-446
[5] Jacob R, Lesage J and Faure J M 2016 Overview of discrete event systems opacity: Models, validation, and quantification *Annual reviews in control* **41** 135-146
[6] Mazaré L 2004 Using unification for opacity properties In *Proceedings of the 4th IFIP WG1* vol 7 pp 165-176
[7] Bryans J W, Koutny M, Mazaré L and Ryan P Y 2008 Opacity generalised to transition systems *International Journal of Information Security* **7** 421-435

DOI: 10.54254/2755-2721/6/20230783

[8] Tong Y, Li Z, Seatzu C and Giua A 2016 Verification of state-based opacity using Petri nets *IEEE Transactions on Automatic Control* **62** 2823-2837
[9] Saboori A and Hadjicostis C N 2011 Verification of K-step opacity and analysis of its complexity *IEEE Transactions on Automation Science and Engineering* **8** 549-559
[10] Saboori A and Hadjicostis C N 2013 Current-state opacity formulations in probabilistic finite automata *IEEE Transactions on automatic control* **59** 120-133