OPACITY OF DISCRETE EVENT SYSTEMS: TRANSFORMATIONS AND ALGORITHMS

Jiří Balun

Dissertation Thesis

Department of Computer Science
Faculty of Science
Palacký University Olomouc
2023

Author
Jiří Balun
Department of Computer Science
Faculty of Science
Palacký University Olomouc
17. listopadu 12
CZ–779 00 Olomouc
Czech Republic
jiri.balun@gmail.com

Supervisor
doc. RNDr. Tomáš Masopust, Ph.D., DSc.

Keywords
Discrete event system, finite automaton, opacity, transformation, complexity, algorithm, verification

Declaration
Hereby I declare that the thesis is my original work.

Most parts of this thesis are based on outcomes of the joint scientific work with Tomáš Masopust. All authors have even share in the results.

Jiří Balun

Abstract – Opacity is a security property of discrete-event systems that asks whether, at any point of a computation, the secret is revealed to a passive intruder. The literature has introduced several notions of opacity, including language-based opacity, trace opacity, current-state opacity, weak k-step opacity, weak ∞-step opacity, strong k-step opacity, initial-state opacity, and initial-and-final-state opacity. In this work, we provide a complete and improved complexity picture of verifying the discussed opacity notions within the finite automata model. First, we focus on the complexity of deciding current-state opacity in systems with a restricted set of events and a restricted structure. Second, we present polynomial-time transformations among the notions that preserve determinism and the number of observable events, allowing the generalization of results across different notions of opacity. Third, we propose three new algorithms for verifying language-based opacity, trace opacity, weak k-step opacity, weak ∞-step opacity, and strong k-step opacity that improve their respective algorithmic complexity.

Acknowledgements

I sincerely thank my supervisor, Tomáš Masopust, for all his help and excellent guidance. I am also grateful to my family for their support.

This thesis was supported by the Ministry of Education, Youth and Sports under INTER-EXCELLENCE project LTAUSA19098 and by grants IGA PrF 2020 019, IGA PrF 2021 022, IGA PrF 2022 018, and IGA PrF 2023 026 of Palacký University Olomouc.

Contents

Preface

The focus of this thesis is on opacity of discrete-event systems, examining three key areas: the complexity of deciding opacity, the design of verification algorithms, and the relationships among various notions of opacity. The results presented in this thesis are mostly based on following articles:

[5] J. Balun and T. Masopust. On opacity verification for discrete-event systems. *IFAC-PapersOnLine*, 53(2):2075–2080, 2020.

[7] J. Balun and T. Masopust. Comparing the notions of opacity for discrete-event systems. *Discrete Event Dynamic Systems*, 31:553–582, 2021.

[9] J. Balun and T. Masopust. On transformations among opacity notions. *2022 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 3012–3017, 2022.

[10] J. Balun and T. Masopust. On verification of weak and strong k-step opacity for discrete-event systems. *IFAC-PapersOnLine*, 55(28):108–113, 2022. 16th IFAC Workshop on Discrete Event Systems WODES 2022.

In [5], we mainly focus on the complexity of deciding current-state opacity in systems with a restricted set of events and a restricted structure. Most of the results from this paper are presented in Chapter 4.

In [7], we introduce transformations between weak k-step opacity and current-state opacity, and between language-based opacity and initial-state opacity. Selected transformations from this article are presented in Sections 5.1, 5.5, and 5.6. We also design new algorithms for verifying language-based opacity, weak k-step opacity and weak ∞-step opacity, the first of which is presented in Section 6.1.

In [9], we have further improved the previously presented transformations

from weak k-step opacity to current-state opacity, which were initially introduced in [7]. The updated transformations are polynomial in terms of the parameter k. These transformations, along with others from this paper, are discussed in Sections 5.4 and 5.6. An extended version of this paper, under review in *Discrete Event Dynamic Systems* at the time of writing this thesis, is available on `arXiv`.

In [10], we design a transformation from strong k-step opacity to weak k-step opacity, as well as algorithms to verify both strong and weak k-step opacity. As the algorithm for verifying weak k-step opacity is an updated version of the algorithm presented in [7], I have included only this improved variant in this thesis. The transformation can be found in Section 5.7, while algorithms are presented in Sections 6.2 and 6.3. An extended version of this paper, accepted for publication in *Automatica* at the time of writing this thesis, is available on `arXiv`.

Furthermore, some of the transformations from Sections 5.2, 5.3, and 5.4 are not yet included in any article. I decided to include them in this thesis to give a complete picture of the transformations among the discussed notions.

In addition to the articles listed above, I have made contributions to the following publications:

[6] J. Balun and T. Masopust. On verification of strong periodic D-detectability for discrete event systems. *IFAC-PapersOnLine*, 53(4):263–268, 2020. 15th IFAC Workshop on Discrete Event Systems WODES 2020.

[8] J. Balun and T. Masopust. On verification of D-detectability for discrete event systems. *Automatica*, 133:109884, 2021.

[31] J. Komenda, D. Zorzenon, and J. Balun. Modeling of safe timed petri nets by two-level (max,+) automata. *IFAC-PapersOnLine*, 55(28):212–219, 2022. 16th IFAC Workshop on Discrete Event Systems WODES 2022.

[53] D. Zorzenon, J. Balun, and J. Raisch. Weak consistency of P-time event graphs. *IFAC-PapersOnLine*, 55(40):19–24, 2022. 1st IFAC Workshop on Control of Complex Systems COSY 2022.

No results from these articles have been included in this thesis as they do not focus on opacity and due to space reasons.

Jiří Balun
Olomouc, June 2023

Chapter 1

Introduction

With the development of digitalization, the security is becoming an increasingly important topic. Since many properties of the systems can be deduced from their discrete abstraction, several cybersecurity notions have been introduced for the discrete-event systems. Namely, such properties include *anonymity* of Schneider and Sidiropoulos [43], *noninterference* of Hadj-Alouane et al. [11], *secrecy* of Alur et al. [1], *security* of Focardi and Gorrieri [22], and *opacity* of Mazaré [36].

This thesis focuses on the opacity property, which guarantees that a system prevents an intruder from revealing its secret. In the opacity setting, the intruder is a passive observer that knows the structure of the system but has only limited capability to observe its behavior. Therefore, if the intruder wants to reveal the secret, he must estimate the current state of the system based on his observations. Intuitively, the system is opaque if for every secret behavior, there is a nonsecret behavior that looks the same to the intruder. Therefore, at no point during the computation can the intruder be certain whether or not secret behavior has occurred. The secret itself is usually modeled as either a set of secret behaviors or a set of secret states. The former option leads to *language-based opacity*, while the latter leads to *state-based opacity*. Several notions of language-based and state-based opacity have been discussed in the literature, from which we selected, in our opinion, the most important and practical ones.

Defining the secret as a behavior results in two notions, that is, language-based opacity (LBO) and trace opacity (TO). In the case of language-based opacity, which was introduced by Badouel et al. [4] and Dubreil et al. [20], the secret is defined as a subset of system's behavior. This subset is known as a

secret language and it contains compromising sequences of the system. Such a sequence may, for example, represent the initiation of a system reboot. Trace opacity, as introduced by Bryans et al. [12], is a special case of language-based opacity. In trace opacity, the secret language comprises only those behaviors that contain one of the secret events that represent the occurence of some critical action of the system.

Regarding state-based opacity, we consider the following six notions: current-state opacity (CSO), weak k-step opacity (k-SO), weak ∞-step opacity (∞-SO), strong k-step opacity (k-SSO), initial-state opacity (ISO), and initial-and-final-state opacity (IFO). In this case, each secret state represents a vulnerable condition of the system, such as a particular part of the system undergoing maintenance.

The most basic state-based notion is current-state opacity of Bryans et al. [13] that prevents the intruder from revealing whether the system is currently in a secret state. However, in the future, the intruder may realize that the system was in a secret state at some earlier point of the computation. For example, if the intruder estimates that the system could be in one of two possible states, and then in the following step, the system proceeds via an observable event that is only possible from one of those states, the intruder can deduce the state in which the system was one step ago. This issue has been considered in the literature and led to the introduction of weak k-step opacity and weak ∞-step opacity by Saboori and Hadjicostis [38, 42]. While weak k-step opacity requires that the intruder cannot ascertain the secret in the current state and k subsequent observable steps, weak ∞-step opacity requires that the intruder can never ascertain that the system was in a secret state. Note that weak 0-step opacity coincides with current-state opacity by definition, and that an n-state automaton is weakly ∞-step opaque if and only if it is weakly $(2^n - 2)$-step opaque [52].

Falcone and Marchand [21] have suggested that weak k-step opacity is not as secure as it may seem. Although it may seem sufficiently confidential, the intruder can still deduce that the system was previously in a secret state, even if the intruder cannot determine the exact time at which the system entered that state. To address this issue, they introduced a stronger version of k-step opacity called strong k-step opacity, which provides a higher level of confidentiality.

Bryans et al. [13] introduced initial-state opacity, which prevents the intruder from revealing, at any time instant, whether the system started in a

Model	$\lvert\Sigma_o\rvert = 1$	$\lvert\Sigma_o\rvert \geq 2$
NFA	coNP-complete	PSPACE-complete [17]
DFA	coNP-complete	PSPACE-complete
partially ordered NFA	NL-complete	PSPACE-complete
partially ordered DFA	NL-complete	PSPACE-complete
acyclic NFA	NL-complete	coNP-complete
acyclic DFA	NL-complete	coNP-complete

Table 1.1: Complexity of verifying current-state opacity for different models with Σ_o being the set of observable events.

secret state. Initial-and-final-state opacity of Wu and Lafortune [50] is a generalization of both current-state opacity and initial-state opacity, where the secret is represented as a pair of an initial and a marked state. Therefore, the intruder can never reveal both starting and ending point of the computation at the same time.

This thesis focuses solely on the theoretical aspects of opacity. However, there have been successful implementations of opacity in various applications, such as concealment of vehicle positions by Saboori and Hadjicostis [40], and ensuring privacy of location-based services by Wu et al. [51]. For a comprehensive overview of opacity and its applications, we recommend the reader the work of Jacob et al. [27].

Most of the mentioned notions have been studied within the framework of many different models, such as finite automata [38], Petri nets [13], timed automata [16], and stochastic automata [30]. In this thesis, we model the system as a finite automaton with partially observable behavior. In some cases, we also consider structurally simpler variants such as partially ordered automata or acyclic automata. In Chapter 2, we introduce relevant concepts of automata theory and we formalize the model itself. Chapter 3 provides an overview of all the opacity notions considered in this work together with illustrative examples.

One of the key areas in opacity research is the complexity of deciding whether a system satisfies a given notion of opacity. Since the verification is often based on the observer construction, the problem belongs to PSPACE. In fact, most of the notions are PSPACE-complete in the general case, and thus there is no polynomial-time verification algorithm unless P = PSPACE. This raises the question of whether the problem is easier to solve if we somehow

Notion	$\lvert\Sigma_o\rvert = 1$	$\lvert\Sigma_o\rvert \geq 2$	Order
LBO	coNP-complete	PSPACE-complete	$O((n+m)2^n)$
TO	NL-complete	PSPACE-complete	$O((n+m)2^n)$
CSO	coNP-complete	PSPACE-complete	$O(\ell 2^n)$ [37]
k-SO	coNP-complete	PSPACE-complete	$O((n+m)2^n)$
∞-SO	coNP-complete	PSPACE-complete	$O((n+m)2^n)$
k-SSO	coNP-complete	PSPACE-complete	$O((n+m)2^n)$
ISO	NL-complete	PSPACE-complete	$O(\ell 2^n)$ [50]
IFO	coNP-complete	PSPACE-complete	$O(\ell 2^{n^2})$ [50]

Table 1.2: Complexity of verifying the notions of opacity for DESs following from the transformations, algorithms, and known results; Σ_o stands for the set of observable events, n for the number of states of the input automaton, ℓ for the number of observable events of the input automaton, and $m \leq \ell n^2$ for the number of transitions in the projected automaton of the input automaton.

restrict the structure of the system. Therefore, in Chapter 4, we investigate the problem of deciding current-state opacity for systems that have a limited number of observable events and that are represented by partially ordered or acyclic automata. However, despite these restrictions, the problem remains hard in almost all practical cases, as indicated in Table 1.1, where we summarize our findings and existing results.

Transformations are another useful tool for analysing the complexity of decision problems. If we can, for example, transform an instance of the current-state opacity problem to an instance of the language-based opacity problem in polynomial time and vice versa, we can derive PSPACE-completeness of language-based opacity from the PSPACE-completeness of current-state opacity. Such transformations were first provided by Wu and Lafortune [50] between language-based opacity, current-state opacity, initial-state opacity, and initial-and-final-state opacity. In Chapter 5, we extend their results and provide transformations for trace opacity, weak k-step opacity, weak ∞-step opacity, and strong k-step opacity. Thus, by combining these transformations, we show how to transform between any two notions, allowing the generalization of results across different notions of opacity. In particular, we show that for systems with two or more observable events, the decision problem of any of the considered notions is PSPACE-complete. On the other hand, if the system has only one observable event, then the

problem is coNP-complete for all notions, except for initial-state opacity and trace opacity, which are NL-complete. We summarize results following from transformations, together with the existing results, in Table 1.2.

In addition to the new complexity results, the transformations also enabled us to design three new algorihms, which we introduce in Chapter 6. Through the analysis of existing algorithms [34, 50, 41, 42, 52, 21, 35, 49], we demonstrate that our algorithms improve the algorithmic complexity of verifying language-based opacity, trace opacity, weak k-step opacity, weak ∞-step opacity, and strong k-step opacity. The right-most column of Table 1.2 provides a summary of the complexities of the best-known algorithms for all of the discussed notions. Note that we have not compared the algorithms experimentally, and therefore in practical cases our algorithms might be outperformed.

Chapter 2

Preliminaries

In this chapter, we formalize the notation and model of a discrete-event system based on finite automata. For more details on these topics see [24, 15].

For a set S, $|S|$ denotes the cardinality of S, and 2^S denotes the power set of S. We define $\mathbb{N}$ to be the set of all non-negative integers, and we extend it with its limit to $\mathbb{N}_\infty = \mathbb{N} \cup \{\infty\}$.

2.1 Languages and automata

An alphabet Σ is a finite nonempty set of events. A string over Σ is a sequence of events from Σ; the empty string is denoted by ε. The set of all finite strings over Σ is denoted by Σ^*. A language L over Σ is a subset of Σ^*. The set of prefixes of strings of L is the set $\overline{L} = \{u \mid \exists v \in \Sigma^*,\, uv \in L\}$. For a string $u \in \Sigma^*$, $|u|$ denotes the length of u, and $\overline{u}$ denotes the set of all prefixes of u.

Definition 2.1. A *nondeterministic finite automaton* (NFA) over an alphabet Σ is a structure $\mathcal{A} = (Q, \Sigma, \delta, I, F)$, where Q is a finite set of states, $\delta\colon Q \times \Sigma \to 2^Q$ is a transition function, $I \subseteq Q$ is a set of initial states, and $F \subseteq Q$ is a set of marked states.

The transition function can be extended to the domain $2^Q \times \Sigma^*$ by induction. Equivalently, the transition function is a relation $\delta \subseteq Q \times \Sigma \times Q$, where, e.g., $\delta(q, a) = \{s, t\}$ denotes two transitions (q, a, s) and (q, a, t). To simplify our proofs, we use the notation $\delta(Q, S) = \cup_{s \in S} \delta(Q, s)$, where $S \subseteq \Sigma^*$.

For a set $Q_0 \subseteq Q$, the set $L_m(\mathcal{A}, Q_0) = \{w \in \Sigma^* \mid \delta(Q_0, w) \cap F \neq \emptyset\}$ is the language marked by $\mathcal{A}$ from the states of Q_0, and $L(\mathcal{A}, Q_0) = \{w \in \Sigma^* \mid$

$\delta(Q_0, w) \neq \emptyset\}$ is the language generated by $\mathcal{A}$ from the states of Q_0. The languages *marked* and *generated* by $\mathcal{A}$ are defined as $L_m(\mathcal{A}) = L_m(\mathcal{A}, I)$ and $L(\mathcal{A}) = L(\mathcal{A}, I)$, respectively. If $\overline{L_m(\mathcal{A})} = L(\mathcal{A})$ holds, then $\mathcal{A}$ is *non-blocking* and every string generated by $\mathcal{A}$ can be extended to a marked string.

The NFA $\mathcal{A}$ is *deterministic* (DFA) if $|I| = 1$ and $|\delta(q, a)| \leq 1$ for every $q \in Q$ and $a \in \Sigma$. In this case, we identify the singletons with their elements, and simply write $\mathcal{A} = (Q, \Sigma, \delta, q_0, F)$ if $I = \{q_0\}$ and $\delta(q, a) = q'$ instead of $\delta(q, a) = \{q'\}$.

Let $\leq$ be the reachability relation on the state set Q defined as $p \leq q$ if there is $w \in \Sigma^*$ such that $q \in \delta(p, w)$. Then, the NFA $\mathcal{A}$ is *partially ordered* (poNFA) if its reachability relation $\leq$ is a partial order. If $\mathcal{A}$ is a partially ordered DFA, we use the notation *poDFA*. The automaton is *acyclic*, if $q \notin \delta(q, w)$ for every $q \in Q$ and $w \in \Sigma^* - \{\varepsilon\}$.

Let $\mathcal{A}_i = (Q_i, \Sigma_i, \delta_i, I_i, F_i)$, where $i \in \{1, 2\}$, be two NFAs. For $\mathcal{A}_1$ and $\mathcal{A}_2$ over common alphabet $\Sigma = \Sigma_1 = \Sigma_2$, the *product automaton* of $\mathcal{A}_1$ and $\mathcal{A}_2$ is defined as the automaton $\mathcal{A}_1 \times \mathcal{A}_2 = (Q_1 \times Q_2, \Sigma, \delta, I_1 \times I_2, F_1 \times F_2)$, where $\delta((q_1, q_2), a) = \delta_1(q_1, a) \times \delta_2(q_2, a)$ for every pair of states $(q_1, q_2) \in Q_1 \times Q_2$ and every event $a \in \Sigma$. Notice that the definition does not restrict the state space of the product automaton to its reachable part. In case where $\Sigma_1 \neq \Sigma_2$, we use the *synchronous product* of $\mathcal{A}_1$ and $\mathcal{A}_2$, which is defined as the automaton $\mathcal{A}_1 \parallel \mathcal{A}_2 = (Q_1 \times Q_2, \Sigma_1 \cup \Sigma_2, \delta, I_1 \times I_2, F_1 \times F_2)$ where

$$\delta((q_1, q_2), a) = \begin{cases} (\delta_1(q_1, a), \delta_2(q_2, a)) & \text{if } a \in \Sigma_1 \cap \Sigma_2, \delta_1(q_1, a)! \text{ and } \delta_2(q_2, a)! \\ (\delta_1(q_1, a), q_2) & \text{if } a \in \Sigma_1 - \Sigma_2 \text{ and } \delta_1(q_1, a)! \\ (q_1, \delta_2(q_2, a)) & \text{if } a \in \Sigma_2 - \Sigma_1 \text{ and } \delta_2(q_2, a)! \\ \text{undefined} & \text{otherwise} \end{cases}$$

for $(q_1, q_2) \in Q_1 \times Q_2$ and $a \in \Sigma_1 \cup \Sigma_2$, and $\delta_i(q_i, a)!$ denotes the fact that there is a transition under a defined at q_i in $\mathcal{A}_i$.

2.2 Discrete-event systems

In this section, we recall the standard definition of a discrete-event system. Intuitively, we model the system as a non-deterministic finite automaton with partially observable behavior.

Definition 2.2. A *discrete-event system* (DES) G over Σ is an NFA over Σ together with the partition of Σ into Σ_o and Σ_{uo} of *observable* and *unobservable events*, respectively.

If we want to specify that the DES is modeled by a DFA, we talk about *deterministic* DES. If the marked states are irrelevant, we omit them and simply write $G = (Q, \Sigma, \delta, I)$.

Definition 2.3. Let Σ be an alphabet and $\Sigma_o, \Sigma_{uo} \subseteq \Sigma$ be its partition into observable and unobservable events. The *observation projection* $P\colon \Sigma^* \to \Sigma_o^*$ is a morphism for concatenation defined by $P(a) = \varepsilon$ if $a \in \Sigma_{uo}$, and $P(a) = a$ if $a \in \Sigma_o$.

The action of P on a string $a_1 a_2 \cdots a_n$, with $a_i \in \Sigma$ for $1 \leq i \leq n$, is to erase all unobservable events, that is, $P(a_1 a_2 \cdots a_n) = P(a_1)P(a_2) \cdots P(a_n)$. The definition can be readily extended to languages.

Definition 2.4. A *projected automaton* of a DES G over Σ with respect to the projection $P\colon \Sigma^* \to \Sigma_o^*$ is the NFA $P(G)$ obtained from G by replacing every transition (p, a, q) by $(p, P(a), q)$, followed by the standard elimination of the ε-transitions.

Equivalently, the transition function $\gamma\colon Q \times \Sigma_o \to 2^Q$ of $P(G)$ can be defined as $\gamma(q, a) = \delta(q, P^{-1}(a))$. Note that $P(G)$ is an NFA over Σ_o with the same states as G that recognizes the language $P(L_m(G))$ and can be constructed in polynomial time, see [24] for more details.

Definition 2.5. An *observer* of a DES G is the accessible part of the DFA constructed from $P(G)$ by the standard subset construction.

We call the DFA constructed from $P(G)$ by the standard subset construction a *full observer* of G. The full observer has exponentially many states compared with G, and in the worst case, the same holds for the observer as well, see [28] for more details.

For DESs with a single observable event we define a function φ_k that assigns, to every state q, the maximal number $i \in \{0, \ldots, k\}$ of observable steps that are possible from state q.

Definition 2.6. Let $G^a = (Q, \Sigma, \delta, I)$ be a DES with $\Sigma_o = \{a\}$ and $P\colon \Sigma^* \to \{a\}^*$ be the observation projection. The function $\varphi_k\colon Q \to \{0, \ldots, k\}$ with respect to P is defined as $\varphi_k(q) = \max\{i \in \{0, \ldots, k\} \mid \delta(q, P^{-1}(a^i)) \neq \emptyset\}$.

Evidently, if $\varphi_k(q) \geq |Q|$ for a state $q \in Q$, then $\varphi_k(q) = k$, since there must be a cycle containing an observable event that is reachable from q. Therefore, we can assume that k is never greater than the number of states of the system G^a, i.e., $k \leq |Q|$.

Chapter 3

Notions of opacity

In this chapter, we present the formal definitions of all considered opacity notions within the finite automata model. For more details about opacity, we refer the reader to the overview by Jacob et al. [27].

The opacity notions studied in this thesis can be divided into two types, namely language-based opacity and state-based opacity. The difference between the two types is the way the secret is modeled. If the secret is modeled as a set of behaviors, then opacity notion is referred to as language-based. In the second case, the secret is modeled as a set of states, giving the state-based opacity notion.

In the first two sections, we introduce the language-based notions, namely language-based opacity and trace opacity. The rest of the chapter is dedicated to the notions of state-based opacity, namely current-state opacity, weak k-step opacity, strong k-step opacity, initial-state opacity, and initial-and-final-state opacity. Aside from strong k-step opacity, which is defined only for deterministic DESs, we define all other notions for nondeterministic systems.

3.1 Language-based opacity (LBO)

Language-based opacity was introduced by Badouel et al. [4] and Dubreil et al. [20]. We recall the most general definition by Lin [34]. Intuitively, a system is language-based opaque if for every string w in the secret language, there exists a string w' in the non-secret language with the same observation $P(w) = P(w')$. In this case, the intruder cannot conclude whether the secret string w or the non-secret string w' has occurred.

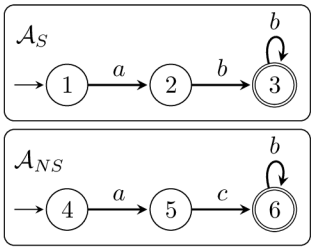


Figure 3.1: Example of language-based opacity.

Definition 3.1. Given a DES $G = (Q, \Sigma, \delta, I)$, a projection $P\colon \Sigma^* \to \Sigma_o^*$, a secret language $L_S \subseteq L(G)$, and a non-secret language $L_{NS} \subseteq L(G)$. System G is *language-based opaque* (LBO) if $L_S \subseteq P^{-1}P(L_{NS})$.

We assume that the languages $L_S = L(\mathcal{A}_S)$ and $L_{NS} = L(\mathcal{A}_{NS})$ are represented by the non-blocking automata $\mathcal{A}_S = (Q_S, \Sigma, \delta_S, I_S, F_S)$ and $\mathcal{A}_{NS} = (Q_{NS}, \Sigma, \delta_{NS}, I_{NS}, F_{NS})$, respectively. Without loss of generality, we may assume that their sets of states are disjoint, that is, $Q_S \cap Q_{NS} = \emptyset$. It is worth mentioning that the secret and non-secret languages are often considered to be regular; and we consider it as well. The reason is that, for non-regular languages, the inclusion problem is undecidable; see Asveld and Nijholt [3] for more details.

Another notion studied in the literature is *weak language-based opacity* [34], which should not be confused with (strong) language-based opacity defined above. In comparison, the weak notion holds if the intruder confuses at least one secret string, formally $L_S \cap P^{-1}P(L_{NS}) \neq \emptyset$. We do not consider the weak notion in this thesis.

Example 3.2. Let G over $\Sigma = \{a, b, c\}$ depicted in Figure 3.1 be an instance of the language-based opacity problem with the secret language $L_S = L(\mathcal{A}_S) = abb^*$ and the non-secret language $L_{NS} = L(\mathcal{A}_{NS}) = acb^*$. We distinguish two cases depending on whether event c is observable or not.

In the first case, we assume that event c is unobservable. In this case, G is language-based opaque, because $P(L_S) = abb^*$ and $P(L_{NS}) = ab^*$, and the reader can see that $P(L_S) \subseteq P(L_{NS})$.

In the second case, we assume that event c is observable. In this case, G is not language-based opaque, because $ab \in P(L_S)$ whereas $ab \notin P(L_{NS})$. $\diamond$

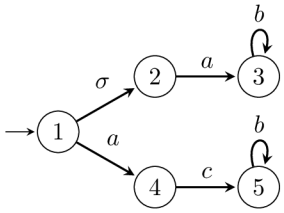


Figure 3.2: Example of trace opacity.

3.2 Trace opacity (TO)

Trace opacity was introduced by Bryans et al. [12]. A trace $w \in \Sigma^*$ is secret if it contains any event from a specified set of secret events, otherwise w is non-secret. In accordance with [12], we consider all secret events to be unobservable. A system is trace opaque if for every secret trace, there is a non-secret trace that looks the same to the intruder.

Definition 3.3. Given a DES $G = (Q, \Sigma, \delta, I)$, a projection $P\colon \Sigma^* \to \Sigma_o^*$, and a set of unobservable secret events $S \subseteq \Sigma_{uo}$. System G is *trace opaque* (TO) if $P(Sec(G)) \subseteq P(Pub(G))$, where $Sec(G) = L(G) \cap \Sigma^* S \Sigma^*$ is the set of secret traces and $Pub(G) = L(G) \cap (\Sigma - S)^*$ is the set of non-secret traces.

Intuitively, trace opacity is a special case of language-based opacity, where the secret language of trace opacity is strictly defined as a set of strings containing at least one secret event, and the non-secret language is defined as any other behavior of the system. In Section 5.3, we present a way to construct automata $\mathcal{A}_S$ and $\mathcal{A}_{NS}$ from a trace opacity problem instance G such that $L(\mathcal{A}_S) = Sec(G)$ and $L(\mathcal{A}_{NS}) = Pub(G)$.

Example 3.4. Let G over $\Sigma = \{a, b, c, \sigma\}$ depicted in Figure 3.2 be an instance of the trace opacity problem with the set of secret events $S = \{\sigma\}$. We distinguish two cases depending on whether event c is observable or not.

If event c is unobservable, then G is trace opaque, because for every secret trace $w \in Sec(G) = \sigma\overline{ab^*}$ there is a non-secret trace $w' \in Pub(G) = \overline{acb^*}$ with the same observation $P(w) = P(w')$, since we have $P(Sec(G)) = P(Pub(G))$.

If event c is observable, then the reader can see that G is not trace opaque. There are non-secret traces ε and a with the same observation as secret traces σ and σa, respectively, but there is no non-secret trace for the secret trace σab with observation $P(\sigma ab) = ab$. ◇

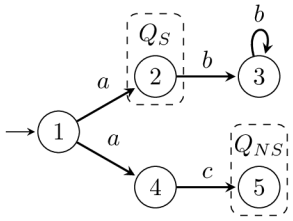


Figure 3.3: Example of current-state opacity.

3.3 Current-state opacity (CSO)

Bryans et al. [13] introduced current-state opacity for systems modeled by Petri nets and Bryans et al. [12] generalized it to transition systems. Current-state opacity asks whether the intruder cannot decide, at any instance of time, whether the system is currently in a secret state. Therefore, the system is current-state opaque if, for every string leading to a secret state, there exists another string with the same observation that leads to a non-secret state.

Definition 3.5. Given a DES $G = (Q, \Sigma, \delta, I)$, a projection $P\colon \Sigma^* \to \Sigma_o^*$, a set of secret states $Q_S \subseteq Q$, and a set of non-secret states $Q_{NS} \subseteq Q$. System G is *current-state opaque* if for every string w such that $\delta(I, w) \cap Q_S \neq \emptyset$, there exists a string w' such that $P(w) = P(w')$ and $\delta(I, w') \cap Q_{NS} \neq \emptyset$.

Note that the definition of current-state opacity does not require $Q_{NS} = Q - Q_S$, and thus the systems we consider can contain states that are neither secret nor non-secret. We call these states *neutral* and we cannot simply handle them as non-secret states.

Example 3.6. Let G over $\Sigma = \{a, b, c\}$ depicted in Figure 3.3 be an instance of the current-state opacity problem with the set of secret states $Q_S = \{2\}$ and the set of non-secret states $Q_{NS} = \{5\}$. We distinguish two cases depending on whether event c is observable or not.

If event c is unobservable, then G is current-state opaque, because the only string leading to the secret state, state 2, is the string a, for which the string ac leading to the non-secret state, state 5, satisfies that $P(a) = P(ac)$.

If event c is observable, then G is not current-state opaque, because the only string leading to a non-secret state, string ac, has a different observation than the string a leading to the secret state, that is, $P(ac) \neq P(a)$. ◇

3.4 Weak k-step opacity (k-SO)

The notion of weak k-step opacity, which was introduced by Saboori and Hadjicostis [38, 42], is a generalization of current-state opacity requiring that the intruder cannot reveal the secret in the current state and k subsequent observable steps.

Definition 3.7. Given a DES $G = (Q, \Sigma, \delta, I)$, a projection $P\colon \Sigma^* \to \Sigma_o^*$, a set of secret states $Q_S \subseteq Q$, a set of non-secret states $Q_{NS} \subseteq Q$, and a parameter $k \in \mathbb{N}_\infty$. System G is *weakly k-step opaque* (k-SO) if for every string $st \in L(G)$ with $|P(t)| \leq k$ and $\delta(\delta(I, s) \cap Q_S, t) \neq \emptyset$, there exists a string $s't' \in L(G)$ such that $P(s) = P(s')$, $P(t) = P(t')$, and $\delta(\delta(I, s') \cap Q_{NS}, t') \neq \emptyset$.

We distinguish two special cases for $k = 0$ and $k = \infty$. By definition, weak 0-step opacity is equivalent to current-state opacity. In the case of weak ∞-step opacity, Yin and Lafortune [52] have shown that an n-state DES is weakly ∞-step opaque if and only if it is weakly $(2^n - 2)$-step opaque.

Below we present a separate definition of weak ∞-step opacity, since this notion is often studied independently in the literature. In addition, the transformations of weak ∞-step opacity are simpler than those of weak k-step opacity, and so in Section 5.6 we use Transformation 5.34 from weak ∞-step opacity to current-state opacity as an intermediate step before introducing a general transformation for any $k \in \mathbb{N}_\infty$.

Definition 3.8. Given a DES $G = (Q, \Sigma, \delta, I)$, a projection $P\colon \Sigma^* \to \Sigma_o^*$, a set of secret states $Q_S \subseteq Q$, and a set of non-secret states $Q_{NS} \subseteq Q$. System G is *weakly ∞-step opaque* (∞-SO) if for every string $st \in L(G)$ such that $\delta(\delta(I, s) \cap Q_S, t) \neq \emptyset$, there exists a string $s't' \in L(G)$ such that $P(s) = P(s')$, $P(t) = P(t')$, and $\delta(\delta(I, s') \cap Q_{NS}, t') \neq \emptyset$.

Example 3.9. Let G over $\Sigma = \{a, b, c\}$ depicted in Figure 3.4 be an instance of the weak k-step opacity problem with the set of secret states $Q_S = \{2\}$ and the set of non-secret states $Q_{NS} = \{4\}$. We consider two cases based on the observability status of event c.

If event c is unobservable, then G is weakly k-step opaque for any $k \in \mathbb{N}_\infty$. Indeed, the only string leading to the unique secret state, state 2, is the string a. The same string leads to the unique non-secret state, state 4. Then, any possible extension of the string a from the secret state 2 is the string b^i, for

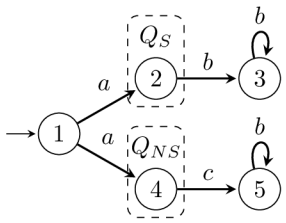


Figure 3.4: Example of weak k-step opacity.

$i \in \mathbb{N}$, which reaches state 3. However, for any such extension, there is the extension cb^i from the non-secret state 4 with $P(ab^i) = P(acb^i)$.

If c is observable, then the reader can see that G is weakly 0-step opaque, or in other words, current-state opaque. However, G is not weakly k-step opaque for any $k > 0$, because after observing the string ab, the intruder can deduce that the system was in the secret state 2 one step ago. ◇

3.5 Strong k-step opacity (k-SSO)

While weak k-step opacity is considered to be relatively confidential, Falcone and Marchand [21] have shown that it is not as confidential as it may seem. The intruder may still be able to determine that the system was previously in a secret state, but not the exact time when this occurred. Therefore, they introduced a stronger notion of opacity called strong k-step opacity, which provides a higher level of confidentiality.

In accordance with Falcone and Marchand [21], we consider strong k-step opacity only for deterministic DESs where all states that are not secret are non-secret, that is, $Q_{NS} = Q - Q_S$. It means that every state has its own secret/non-secret status and there are no neutral states.

Definition 3.10. Given a deterministic DES $G = (Q, \Sigma, \delta, q_0)$, a projection $P\colon \Sigma^* \to \Sigma_o^*$, a set of secret states $Q_S \subseteq Q$, and a parameter $k \in \mathbb{N}_\infty$. System G is *strongly k-step opaque* (k-SSO) if for every string $s \in L(G)$, there exists a string $w \in L(G)$ such that $P(s) = P(w)$ and for every prefix w' of w, if $|P(w)| - |P(w')| \leq k$, then $\delta(q_0, w') \notin Q_S$.

Note that strong 0-step opacity is not equivalent to current-state opacity as in the case of weak 0-step opacity. In Theorem 5.59, we show that unobservable transitions from secret states to non-secret states, like transition

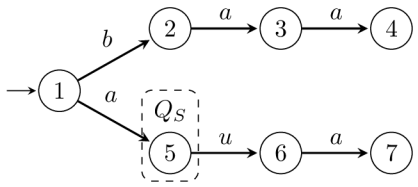


Figure 3.5: Example of strong k-step opacity.

$(5, u, 6)$ in Example 3.11, are the only issues making the difference between strong 0-step opacity and weak 0-step (current-state) opacity. However, as pointed out by Wintenberg et al. [49], strong k-step opacity implies weak k-step opacity as long as no neutral states are considered.

Example 3.11. Let G over $\Sigma = \{a, b, u\}$ depicted in Figure 3.5 be an instance of the strong k-step opacity problem with unobservable event u, the set of secret states $Q_S = \{5\}$, and the set of non-secret states $Q_{NS} = Q - Q_S$. We consider two cases based on the observability status of event b.

If event b is unobservable, then G is strongly k-step opaque for any $k \in \mathbb{N}_\infty$. Indeed, the only string leading to the unique secret state, state 5, is the string a, while the string ba with the same observation leads to the non-secret state, state 3, without going through any secret state. Then, any possible extensions of the string a from the secret state 5 are the strings u and ua, which reach states 6 and 7, respectively. However, for these extensions there are ε and a extensions of the string ba from state 3 such that $P(au) = P(ba)$ and $P(aua) = P(baa)$, respectively, that do not go through a secret state.

If b is observable, then G is weakly k-step opaque for any $k \in \mathbb{N}_\infty$, but not strongly 1-step opaque, because for $s = aua$, the only string with the same observation as s is $w = aua$, and hence the prefixes w' for which $|P(w)| - |P(w')| \leq 1$ are the strings $w' = a$, $w' = au$, and $w' = aua$. However, for $w' = a$, we obtain that $\delta(1, a) = 5 \in Q_S$, which violates the definition of strong 1-step opacity. In fact, the system G is neither strongly 0-step opaque, because for $s = au$, the only strings w with the same observation as s are the strings au and a, both with prefix $w' = a$ such that $|P(w)| - |P(w')| \leq 0$ and $\delta(1, a) = 5 \in Q_S$, which violates the definition of strong 0-step opacity. On the other hand, the system is obviously current-state opaque. Consequently, the notions of strong 0-step opacity and current-state opacity do not coincide. ◇

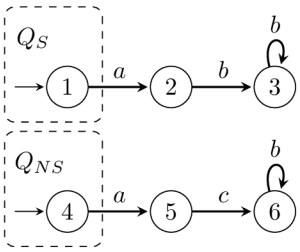


Figure 3.6: Example of initial-state opacity.

3.6 Initial-state opacity (ISO)

Initial-state opacity was first introduced by Bryans et al. [13] for systems modeled by Petri nets and then Bryans et al. [12] generalized it to transition systems. Intuitively, initial-state opacity asks whether the intruder can never reveal whether the computation started in a secret state.

Definition 3.12. Given a DES $G = (Q, \Sigma, \delta, I)$, a projection $P\colon \Sigma^* \to \Sigma_o^*$, a set of secret initial states $Q_S \subseteq I$, and a set of non-secret initial states $Q_{NS} \subseteq I$. System G is *initial-state opaque* (ISO) if for every $w \in L(G, Q_S)$, there exists $w' \in L(G, Q_{NS})$ such that $P(w) = P(w')$.

We consider all states that are neither secret nor non-secret to be neutral. In particular, the secrecy status of the non-initial states do not play any role in initial-state opacity.

Example 3.13. Let G over $\Sigma = \{a, b, c\}$ depicted in Figure 3.6 be an instance of the initial-state opacity problem with the set of secret initial states $Q_S = \{1\}$ and the set of non-secret initial states $Q_{NS} = \{4\}$. We distinguish two cases depending on whether event c is observable or not.

In the first case, we assume that event c is unobservable. In this case, G is initial-state opaque, because $P(L(G, 1)) = \overline{abb^*}$ and $P(L(G, 4)) = \overline{ab^*}$, and the reader can see that $P(L(G, Q_S)) \subseteq P(L(G, Q_{NS}))$.

In the second case, we assume that event c is observable. In this case, G is not initial-state opaque, because $ab \in P(L(G, 1))$ whereas $ab \notin P(L(G, 4))$, and hence $P(L(G, Q_S)) \not\subseteq P(L(G, Q_{NS}))$. ◇

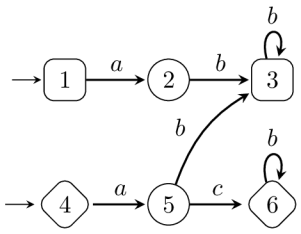


Figure 3.7: Example of initial-and-final-state opacity; the states of secret pair are square-shaped and the states of non-secret pair are diamond-shaped.

3.7 Initial-and-final-state opacity (IFO)

The last notion we consider is initial-and-final-state opacity of Wu and Lafortune [50]. Initial-and-final-state opacity is a generalization of both current-state opacity and initial-state opacity, where the secret is represented as a pair of an initial and a marked state. Consequently, initial-state opacity is a special case of initial-and-final-state opacity where the marked states do not play a role, and current-state opacity is a special case where the initial states do not play a role.

Definition 3.14. Given a DES $G = (Q, \Sigma, \delta, I)$, a projection $P\colon \Sigma^* \to \Sigma_o^*$, a set of secret state pairs $Q_S \subseteq I \times Q$, and a set of non-secret state pairs $Q_{NS} \subseteq I \times Q$. System G is *initial-and-final-state opaque* (IFO) if for every secret pair $(q_0, q_f) \in Q_S$ and every $w \in L(G, q_0)$ such that $q_f \in \delta(q_0, w)$, there exists a non-secret pair $(q_0', q_f') \in Q_{NS}$ and $w' \in L(G, q_0')$ such that $q_f' \in \delta(q_0', w')$ and $P(w) = P(w')$.

Wu and Lafortune [50] also consider a special case with the sets of secret and non-secret pairs of the form $Q_S = I_S \times F_S$ and $Q_{NS} = I_{NS} \times F_{NS}$, where $I_S, I_{NS} \subseteq I$ and $F_S, F_{NS} \subseteq Q$. In this case, however, the instance of initial-and-final-state opacity corresponds to an instance of language-based opacity, where $\mathcal{A}_S = (Q, \Sigma, \delta, I_S, F_S)$ and $\mathcal{A}_{NS} = (Q, \Sigma, \delta, I_{NS}, F_{NS})$ are automata for the secret and non-secret languages, respectively.

Example 3.15. Let G over $\Sigma = \{a, b, c\}$ depicted in Figure 3.7 be an instance of the initial-and-final-state opacity problem with the set of secret pairs $Q_S = \{(1, 3)\}$ and the set of non-secret pairs $Q_{NS} = \{(4, 6)\}$. We distinguish two cases depending on whether event c is observable or not.

In the first case, we assume that event c is unobservable. In this case, G is initial-and-final-state opaque, because the language of sequences that start and end in single secret pair $(1, 3)$ equals to $L_{(1,3)} = abb^*$, while the language of sequences of the non-secret pair $(4, 6)$ equals to $L_{(4,6)} = acb^*$, and the reader can see that $P(L_{(1,3)}) \subseteq P(L_{(4,6)})$.

In the second case, we assume that event c is observable. In this case, G is not initial-and-final-state opaque, because $ab \in P(L_{(1,3)})$ whereas $ab \notin P(L_{(4,6)})$, and hence $P(L_{(1,3)}) \not\subseteq P(L_{(4,6)})$. Notice that sequences that start in state 4 and end in state 3 do not affect initial-and-final-state opacity, since pair $(4, 3)$ belongs neither to Q_S nor Q_{NS}. $\diamond$

Chapter 4

Properties of current-state opacity

This chapter focuses on analyzing the complexity of verifying current-state opacity in systems with a restricted set of events and a restricted structure. We show that in most cases these restrictions do not make the verification tractable, and therefore the problem remains hard in essentially all practical cases.

The complexity of opacity verification has widely been investigated in the literature and is often based on the computation of observer. Thus the problem belongs to PSpace. It is actually PSpace-complete for most of the discussed notions. Indeed, Cassez et al. [17] showed that the verification of current-state opacity is at least as hard as deciding universality, which is PSpace-complete for nondeterministic automata as well as for deterministic automata with partial observation.

Remark 4.1. *By Cassez et al. [17], the verification of current-state opacity is at least as hard as deciding universality. Indeed, for a DES* $G = (Q, \Sigma, \delta, I, F)$*, we have* $L(G) = \Sigma^*$ *if and only if* G *is current-state opaque with respect to* $Q_S = Q - F$*,* $Q_{NS} = F$*, and* $P\colon \Sigma \to \Sigma$*.*

However, PSpace-completeness of universality problem requires a non-trivial structure of the model and the ability to express all possible strings. This give rise to a question whether there are structurally simpler systems for which the verification of opacity is tractable. We investigate the problem for, in our opinion, structurally the simplest systems: for acyclic automata (that do not have the ability to express all strings, and actually express only

a finite number of strings) and for automata where all cycles are in the form of self-loops (which may still seem trivial in the structure, because as soon as the system leaves a state, it can never return to that state).

To simplify the proofs, we first reduce current-state opacity to the language inclusion problem. This reduction is similar to that of Wu and Lafortune [50] reducing current-state opacity to language-based opacity.

Lemma 4.2. *Let $G = (Q, \Sigma, \delta, I)$ be a DES, $P\colon \Sigma^* \to \Sigma_o^*$ a projection, and $Q_S, Q_{NS} \subseteq Q$ sets of secret and non-secret states, respectively. Let L_S denote the marked language of the automaton $G_S = (Q, \Sigma, \delta, I, Q_S)$ and L_{NS} denote the marked language of the automaton $G_{NS} = (Q, \Sigma, \delta, I, Q_{NS})$. Then G is current-state opaque if and only if $P(L_S) \subseteq P(L_{NS})$.*

Proof. Assume that w is such that $\delta(I, w) \cap Q_S \neq \emptyset$. This is if and only if $P(w) \in P(L_S)$. Then, by definition, there is a string w' such that $P(w) = P(w')$ and $\delta(I, w') \cap Q_{NS} \neq \emptyset$, which is if and only if $P(w) \in P(L_{NS})$. □

The observations from Remark 4.1 and Lemma 4.2, together with the results on the complexity of deciding universality and inclusion give us strong tools to show lower and upper complexity bounds for deciding (current-state) opacity. We summarized results from this chapter, together with the existing results, in Table 1.1.

4.1 Simplification of the system

In this section we provide two useful transformations that can simplify any system without affecting its property of being current-state opaque. As a result, any instance of current-state opacity decision problem can be transformed in polynomial time into a deterministic system that has at most two observable events. Later, these simplifications will allow us to generalize some of the results from this chapter to other opacity notions.

The following transformation reduces the number of observable events in DESs with at least three observable events. The main idea is to encode the transition labels in binary. In Theorems 4.4, 5.6, and 5.12, we show that this transformation does not affect the system's status of current-state opacity, initial-state opacity, and trace opacity. This way we preserve the number of observable events in transformations in Chapter 5 that introduce new observable events.

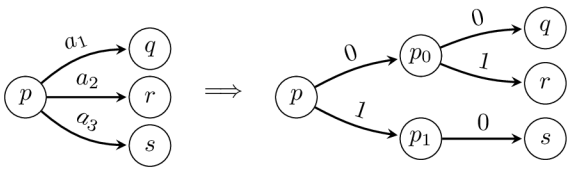


Figure 4.1: The replacement of three observable events $\{a_1, a_2, a_3\}$ with the encoding $e(a_1) = 00$, $e(a_2) = 01$, and $e(a_3) = 10$, and new states p_0 and p_1.

Transformation 4.3. Let $G = (Q, \Sigma, \delta, I)$ be a DES modeled by an NFA, $P\colon \Sigma^* \to \Sigma_o$ be the observation projection, $\Gamma_o \subseteq \Sigma_o$ be an alphabet with at least three events, and $e\colon \Gamma_o \to \{0,1\}^k$ be a binary encoding (that is, an injective function), where $k \leq \lceil \log_2(|\Gamma_o|) \rceil$. We construct a DES

$$r(G) = (Q', (\Sigma - \Gamma_o) \cup \{0,1\}, \delta', I)$$

so that we start with the system G and replace every transition (p, a, q) with $a \in \Gamma_o$ and $e(a) = b_1 b_2 \cdots b_k \in \{0,1\}^k$ by k transitions

$$(p, b_1, p_{b_1}), (p_{b_1}, b_2, p_{b_1 b_2}), \ldots, (p_{b_1 \cdots b_{k-1}}, b_k, q)$$

where the states $p_{b_1}, \ldots, p_{b_1 \cdots b_{k-1}}$ are added to the set of states Q' of the system $r(G)$. These states are created when needed for the first time, and reused later during the replacements, cf. Figure 4.1 illustrating a replacement of three observable events $\{a_1, a_2, a_3\}$ with the encoding $e(a_1) = 00$, $e(a_2) = 01$, and $e(a_3) = 10$. Finally, we define projection $P'\colon [(\Sigma - \Gamma_o) \cup \{0,1\}]^* \to [(\Sigma_o - \Gamma_o) \cup \{0,1\}]^*$. ◇

Notice that the Transformation 4.3 preserves the number of unobservable events and determinism, and that it can be done in polynomial time. On the other hand, it does not preserve partial order because the encoding of a self-loop transition results in a cycle over two or more states.

The following theorem shows that the transformation does not affect the property of the system to be current-state opaque, and therefore we can reduce the number of observable events of any current-state opacity instance to just two.

Theorem 4.4. *A DES G is current-state opaque with respect to Q_S, Q_{NS}, and P if and only if the DES $r(G)$ obtained by Transformation 4.3 is current-state opaque with respect to $Q'_S = Q_S$, $Q'_{NS} = Q_{NS} \cup (Q' - Q)$, and P'.*

Proof. By Lemma 4.2, to show that the system G is current-state opaque if and only if the system $r(G)$ is current-state opaque, we need to show that $P(L_S) \subseteq P(L_{NS})$ if and only if $P'(L'_S) \subseteq P'(L'_{NS})$, where

- $L_S = L_m(\mathcal{A}_S)$, where $\mathcal{A}_S = (Q, \Sigma, \delta, I, Q_S)$,
- $L_{NS} = L_m(\mathcal{A}_{NS})$, where $\mathcal{A}_{NS} = (Q, \Sigma, \delta, I, Q_{NS})$,
- $L'_S = L_m(\mathcal{A}'_S)$, where $\mathcal{A}'_S = (Q', (\Sigma - \Gamma_o) \cup \{0, 1\}, \delta', I, Q'_S)$, and
- $L'_{NS} = L_m(\mathcal{A}'_{NS})$, where $\mathcal{A}'_{NS} = (Q', (\Sigma - \Gamma_o) \cup \{0, 1\}, \delta', I, Q'_{NS})$.

We define a morphism $f\colon \Sigma^* \to ((\Sigma - \Gamma_o) \cup \{0, 1\})^*$ such that $f(a) = e(a)$ for $a \in \Gamma_o$, and $f(a) = a$ for $a \in \Sigma - \Gamma_o$. By the definition of e and the construction of the system $r(G)$, any string $w \in L(G)$ if and only if the string $f(w) \in L(r(G))$. In particular, $P(w) \in P(L_S)$ if and only if $P'(f(w)) \in P'(L'_S)$, and $P(w) \in P(L_{NS})$ if and only if $P'(f(w)) \in P'(L'_{NS})$. Therefore, if $P'(L'_S) \subseteq P'(L'_{NS})$ then $P(L_S) \subseteq P(L_{NS})$. On the other hand, we assume that $P(L_S) \subseteq P(L_{NS})$, and we consider any $P'(x) \in P'(L'_S)$. Then, $P'(x)$ is of the form $P'(f(y))$ for some string $y \in L_S$, and $P(y) \in P(L_S) \subseteq P(L_{NS})$ implies that $P'(x) = P'(f(y)) \in P'(L'_{NS})$. □

In the second transformation, we show how to transform a system modeled by an NFA to a system modeled by a DFA without affecting the system's properties of being current-state opaque, acyclic, and partially ordered.

Transformation 4.5. Let $G = (Q, \Sigma, \delta, I)$ be a DES modeled by an NFA with the secret states Q_S, the non-secret states Q_{NS}, and the corresponding projection $P\colon \Sigma^* \to \Sigma_o^*$. We construct a deterministic DES G_{det} in two steps.

1. First, we ensure that the system has a unique initial state. From G we construct a DES $G' = (Q', \Sigma, \delta', \{q_0\})$, where $Q' = Q \cup \{q_0\}$ contains a new non-secret initial state q_0. Further, for each $q \in I$ we add a new transition (q_0, a, q) to δ', where $a \in \Sigma_o$ is an arbitrary observable event.
2. In the second step, we determinize the transition function of the system. From G' we construct a DES $G_{det} = (Q'', \Sigma \cup \{u\}, \delta'', (q_0, q_0))$ modeled by a DFA, where $Q'' = Q' \times Q'$ is the set of pairs of states, u is a new unobservable event, and the pair $(q_0, q_0) \in Q''$ is a new initial state. We define the transition function δ'' as follows.
   (a) For every transition (p, a, q) in δ', where $p, q \in Q'$ and $a \in \Sigma$, we add a transition $((p, q), a, (q, q))$ to δ''.

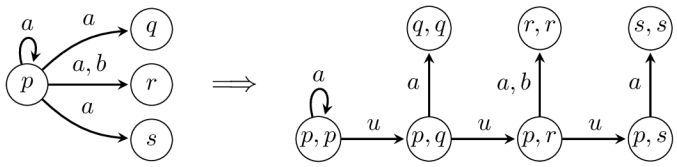


Figure 4.2: Determinization of a DES.

(b) For every state $p \in Q'$ we define the set $R_p = \cup_{a\in\Sigma} \delta'(p,a) - \{p\} = \{p_1, p_2, \ldots, p_\ell\}$ of states different from p that can be reached from p by a single transition. We then add transitions of the form $((p,p), u, (p,p_1))$ and $((p,p_i), u, (p,p_{i+1}))$ for $i = 1, \ldots, \ell - 1$, as shown in Figure 4.2, to create a chain of states from R_p connected to state (p,p). Note that the order in which we connect states from R_p does not affect the resulting system.

We remove unreachable states from G_{det}. Finally, we define the projection $P'\colon (\Sigma^* \cup \{u\}) \to \Sigma_o^*$, and the sets of secret states $Q'_S = \{(p,q) \mid p \in Q_S\}$ and of non-secret states $Q'_{NS} = \{(p,q) \mid p \in Q_{NS} \cup \{q_0\}\}$. ◇

Notice that Transformation 4.5 can be done in polynomial time using at most $(n+1)^2$ states, where n is the number of states in G. In fact, if we omit removing unreachable states at the end of the tranformation, then G_{det} can computed in deterministic logarithmic space. Additionally, this transformation does not introduce any new neutral states and it preserves the number of observable events, acyclicity, and partial order.

Theorem 4.6. *A DES G is current-state opaque with respect to Q_S, Q_{NS}, and P if and only if the deterministic DES G_{det} obtained by Transformation 4.5 is current-state opaque with respect to Q'_S, Q'_{NS}, and P'.*

Proof. The first step of Transformation 4.5 just creates a new non-secret initial state q_0 that is unreachable from any other state and that is connected to the original initial states by an observable event a, and so we have $\delta'(q_0, a) = I$. Therefore, G is clearly current-state opaque with respect to Q_S, Q_{NS}, and P if and only if G' is current-state opaque with respect to Q_S, $Q_{NS} \cup \{q_0\}$, and P.

In the second step, the number of observable steps from a state of the system G' is preserved in the system G_{det}. By the construction of G_{det}, every state $(p,q) \in R_p$ is reachable from the state (p,p) by a sequence consisting

only of unobservable event u, and hence (p, q) is contained in every state of the observer that contains state (p, p); and vice versa, because every path to the state (p, q) goes through the state (p, p) in the system G_{det}. Therefore, if a state of the observer contains a secret state (p, q) and a non-secret state (p', q'), then it also contains the original secret state (p, p) and the original non-secret state (p', p'). That is, the system G is current-state opaque with respect to Q_S, Q_{NS}, and P if and only if the system G_{det} is current-state opaque with respect to Q'_S, Q'_{NS}, and P'. □

4.2 Restriction on structure of the system

Our first restriction concerns the number of observable and unobservable events in the system. The following result thus improves the general case in two ways: (i) compared to the general settings we keep the system deterministic, and, mainly, (ii) we restrict the number of observable events to two and the number of unobservable events to one.

Theorem 4.7. *Deciding current-state opacity of a DES modeled by a DFA with three events, one of which is unobservable, is* PSPACE*-complete.*

Proof. Membership in PSPACE was shown by Saboori [37], and also follows directly from Lemma 4.2.

To show hardness, we reduce the current-state opacity problem for a DES modeled by an NFA with just two observable events, which is PSPACE-complete by Remark 4.1 and Saboori [37]. This can be done by Transformation 4.5 which, for a DES modeled by an NFA with just two observable events, constructs a deterministic DES with three events, one of which is unobservable, without affecting the property of current-state opacity. □

Notice that an unobservable event in the previous theorem is unavoidable because any DFA with all events observable is always in a unique state, and therefore never opaque. However, the reader may wonder what happens if we further restrict the number of observable events to just one. We now show that having only one observable event makes the problem computationally easier unless coNP = PSPACE. This result holds even without any restriction on the number of unobservable events, and for nondeterministic automata.

Theorem 4.8. *Deciding current-state opacity of a DES modeled by an NFA with a single observable event is* coNP*-complete.*

Proof. Membership in coNP follows from Lemma 4.2 and the fact that inclusion for unary NFAs is coNP-complete, and hardness follows from the complexity of deciding universality for unary NFAs. For both claims used here, the reader is referred to Stockmeyer and Meyer [47]. □

We obtain the following result for DFAs by applying Transformation 4.5 which, for a DES modeled by an NFA with a single observable event, constructs a deterministic DES with two events, one of which is unobservable, without affecting the property of current-state opacity.

Corollary 4.9. *Deciding current-state opacity of a DES modeled by a DFA with two events, one of which is unobservable, is* coNP*-complete.*

Previous results show that only restricting the number of events does not lead to tractable complexity. But it gives rise to another question whether there are structurally simpler systems for which the opacity verification problem is tractable.

Structurally the simplest systems we could think of are acyclic DFAs with full observation, recognizing only finite languages. However, these systems are never opaque, since they are deterministic and fully observed. Nontrivial structures to be considered could thus be acyclic NFAs that still recognize only finite languages, and hence do not possess the ability to express all strings over the alphabet. We combine this restriction with the restriction on the number of events.

Theorem 4.10. *Deciding current-state opacity of a DES modeled by an acyclic NFA with two or more observable events is* coNP*-complete.*

Proof. Assume that the acyclic NFA has n states. Then any string from its language is of length at most $n - 1$. Thus, to show that the system is not opaque, an NP algorithm guesses a subset of secret states and a string of length at most $n - 1$ and verifies, in polynomial time, that the guessed subset is reachable by the guessed string. This shows that verifying opacity is in coNP. Notice that membership in coNP can also be directly derived

from Lemma 4.2 and the complexity of inclusion for so-called rpoNFAs[1] of Krötzsch et al. [32] that are more general than acyclic NFAs.

To show coNP-hardness, we reduce the complement of CNF satisfiability. The proof is based on the construction showing that non-equivalence for regular expressions with operations union and concatenation is NP-complete even if one of them is of the form Σ^n for some fixed n, see [25, 47].

Let $\{x_1, \ldots, x_n\}$ be a set of variables and $\varphi = \varphi_1 \wedge \cdots \wedge \varphi_m$ be a formula in CNF, where every φ_i is a disjunction of literals. Without loss of generality, we may assume that no clause φ_i contains both x and $\neg x$. Let $\neg\varphi$ be the negation of φ obtained by de Morgan's laws. Then $\neg\varphi = \neg\varphi_1 \vee \cdots \vee \neg\varphi_m$ is in disjunctive normal form.

For every $i = 1, \ldots, m$, we define a regular expression $\beta_i = \beta_{i,1}\beta_{i,2}\cdots\beta_{i,n}$, where

$$\beta_{i,j} = \begin{cases} (0+1) & \text{if neither } x_j \text{ nor } \neg x_j \text{ appear in } \neg\varphi_i \\ 0 & \text{if } \neg x_j \text{ appears in } \neg\varphi_i \\ 1 & \text{if } x_j \text{ appears in } \neg\varphi_i \end{cases}$$

for $j = 1, \ldots, n$. Let $\beta = \bigcup_{i=1}^{m} L(\beta_i)$ be the union of languages defined by expressions β_i. Then we have that $w \in L(\beta)$ if and only if w satisfies some $\neg\varphi_i$. That is, we have that $L(\beta) = \{0,1\}^n$ if and only if $\neg\varphi$ is a tautology, which is if and only if φ is not satisfiable. Notice that the length of every string recognized by β_i is exactly n.

Let G be an NFA consisting of m paths of length n, each corresponding to the language of β_i, and make the last state of each of these paths non-secret, that it, it is placed to Q_{NS}. In addition, add a path consisting of $n+1$ states $\{\alpha_0, \alpha_1, \ldots, \alpha_n\}$ and transitions $(\alpha_\ell, a, \alpha_{\ell+1})$, for $0 \leq \ell < n$, where $a \in \{0,1\}$. Let α_n be the sole secret state, i.e., $Q_S = \{\alpha_n\}$. Notice that the language of G marked by the states in Q_S is $\{0,1\}^n$, whereas the language marked by the states in Q_{NS} is $L(\beta)$. By Lemma 4.2, G is current-state opaque if and only if $\{0,1\}^n \subseteq L(\beta)$, which is if and only if φ is not satisfiable. This completes the proof of coNP-completeness. □

Again, we can show that the situation is computationally simpler if only one observable event is allowed.

[1]The NFA $\mathcal{A}$ is *restricted partially ordered (rpoNFA)* if the reachability relation $\leq$ is a partial order and $\mathcal{A}$ is self-loop deterministic, i.e. for every state q and every event a, if $q \in \delta(q,a)$ then $\delta(q,a) = \{q\}$.

Theorem 4.11. *Deciding current-state opacity of a DES modeled by an acyclic NFA with a single observable event is* NL*-complete, and hence solvable in polynomial time.*

Proof. Membership in NL follows from Lemma 4.2 and the complexity of inclusion for unary languages, see Krötzsch et al. [32].

To prove NL-hardness, we reduce the DAG-reachability problem. Let $\mathcal{G}$ be a directed acyclic graph with n vertices, and let s and t be two vertices of $\mathcal{G}$. We define an acyclic NFA $\mathcal{A}$ as follows. With each node of $\mathcal{G}$, we associate a state in $\mathcal{A}$. Whenever there is an edge from i to j in $\mathcal{G}$, we add a transition (i, a, j) to $\mathcal{A}$. The resulting automaton $\mathcal{A}$ is an acyclic NFA. Let t be the sole secret state, i.e., $Q_S = \{t\}$, and let Q_{NS} be empty. Obviously, $\mathcal{A}$ is not current-state opaque if and only if there is a string $w \in \{a\}^*$ such that $\delta(s, w) \cap Q_S \neq \emptyset$. Hence $\mathcal{A}$ is not current-state opaque if and only if t is reachable from s in $\mathcal{G}$. □

Since Transformation 4.5 preserves acyclicity and can be computed in deterministic logarithmic space, we can apply it to the systems of Theorems 4.10 and 4.11 to obtain hardness part of following results. Membership then follows from Lemma 4.2 and the corresponding results on the complexity of inclusion.

Corollary 4.12. *Deciding current-state opacity of a DES is*

1. coNP*-complete if the system is modeled by an acyclic DFA with three events, one of which is unobservable, and*
2. NL*-complete if the system is modeled by an acyclic DFA with two events, one of which is unobservable.*

Above, we considered systems generating only finitely many behaviors. However, real-world systems are usually not that simple and often require additional properties, such as deadlock freeness. Therefore, we now consider partially ordered automata, a kind of automata where all cycles are only in the form of self-loops. Such automata are, in our opinion, structurally the simplest DES where deadlock freeness can be ensured (by adding a self-loop). Their mark languages form a subclass of regular languages strictly included in *star-free languages*, see [14, 44]. Star-free languages are languages definable by *linear temporal logic* that is often used as a specification language in automated verification.

We then immediately obtain the following result for nondeterministic partially ordered automata.

Theorem 4.13. *Deciding current-state opacity of a DES modeled by a poNFA with only two events, both of which are observable, is* PSPACE*-complete.*

Proof. Membership in PSPACE follows from Lemma 4.2 and the results on the complexity of inclusion for poNFAs, and hardness from the fact that deciding universality for poNFAs with only two events is PSPACE-complete. For both claims see Krötzsch et al. [32]. □

The situation is again easier if the model has only a single observable event.

Theorem 4.14. *Deciding current-state opacity of a DES modeled by a poNFA with a single observable event is* NL*-complete.*

Proof. Membership in NL follows from Lemma 4.2 and the corresponding complexity of inclusion, and hardness from the fact that deciding universality for unary poNFAs is NL-complete, see Krötzsch et al. [32]. □

Again, we use Transformation 4.5, which preserves partial order and can be computed in deterministic logarithmic space, and apply it to the systems of Theorems 4.13 and 4.14 to obtain the hardness part of the following results. Membership then follows from Lemma 4.2 and the corresponding results on the complexity of inclusion.

Corollary 4.15. *Deciding current-state opacity of a DES is*

1. PSPACE*-complete if the system is modeled by a poDFA with three events, one of which is unobservable, and*
2. NL*-complete if the system is modeled by a poDFA with two events, one of which is unobservable.*

Chapter 5

Transformations among opacity notions

In this chapter, we introduce new trasnformations among the considered opacity decision problems. In other words, for an instance of one opacity notion that consists of a DES, an observation projection, and a secret description, we transform it into an instance of another opacity notion.

Comparing different notions of opacity for automata models, Saboori and Hadjicostis [39] provided a language-based definition of initial-state opacity, Cassez et al. [17] transformed trace opacity to current-state opacity, and Wu and Lafortune [50] showed that current-state opacity, initial-and-final-state opacity, and language-based opacity can be transformed to each other. They further provided transformations of initial-state opacity to language-based opacity and to initial-and-final-state opacity, and, for prefix-closed languages, a transformation of language-based opacity to initial-state opacity.

In this thesis, we extend these results by showing that, for automata models, all the discussed notions of opacity are transformable to each other. As well as the existing transformations, our transformations are computable in polynomial time and preserve the number of observable events and determinism (whenever it is meaningful). In the case of state-based opacity notions, our goal was to design transformations that do not introduce any new neutral states into the system, since their existence may not be practically justified. However, in some cases, we may need to give a separate transformation for systems that already contain neutral states. The meaning of neutral states is not yet clear in the literature. They are fundamental in language-based opacity, but questionable in state-based opacity. In any case, we cannot sim-

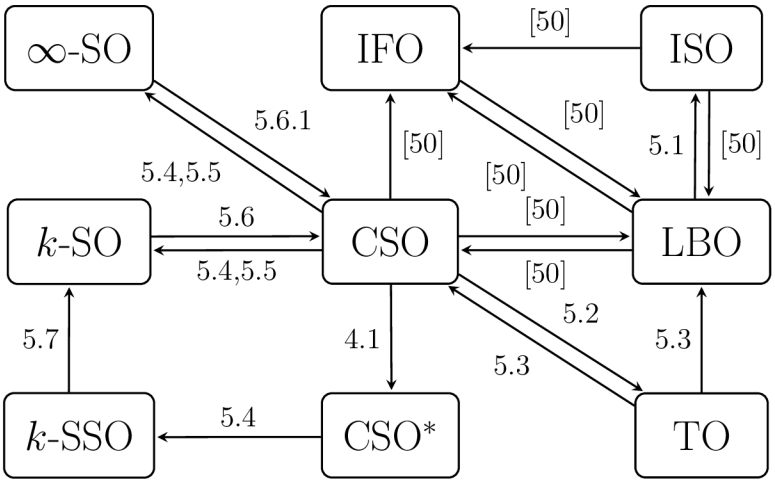


Figure 5.1: Overview of the transformations among the notions of opacity for automata models. The node CSO* denotes a simplified instance of current-state opacity modeled by a DFA with at most two observable events.

ply handle neutral states as non-secret states. We summarize our results, together with the existing results, in Figure 5.1.

There are two immediate applications of the transformations. First, the transformations provide a deeper understanding of the differences among the opacity notions from the structural point of view. For instance, the reader may deduce from the transformations that, for prefix-closed languages, the notions of language-based opacity, initial-state opacity, and current-state opacity coincide, or that to transform current-state opacity to weak ∞-step opacity means to add only a single state and a few transitions.

Second, the transformations provide a tool to obtain the complexity results for all the discussed opacity notions by studying just one of the notions. To illustrate, consider for example the result of Theorem 4.7 showing that deciding current-state opacity for systems modeled by DFAs with three events, one of which is unobservable, is PSPACE-complete. Since we can transform the problems of deciding current-state opacity and of deciding weak k-step opacity to each other in polynomial time, preserving determinism and the number of observable events, we obtain that deciding weak k-step opacity for systems modeled by DFAs with three events, one of which is unobservable, is PSPACE-complete as well. In particular, combining the transformations with known results from Jacob et al. [27] and results from Chapter 4, we obtain a

complete complexity picture of verifying the discussed notions of opacity, as summarized in Table 1.2.

Thus, by combining the transformations of Wu and Lafortune [50] with Theorems 4.7 and 4.8, we immedietly obtain new results for language-based opacity and initial-and-final-state opacity. In more detail, the transformations of Wu and Lafortune [50] preserve the determinism of transitions, but result in automata with a set of initial states. This issue can, however, be easily fixed by adding a new initial state, connecting it to the original initial states by new unobservable events, and making the original initial states non-initial.

Corollary 5.1. *The problems of deciding whether a DES satisfies language-based opacity and initial-and-final-state opacity are* PSPACE*-complete. The problems remain* PSPACE*-complete even if the system is a DFA with three events, one of which is unobservable.*

Corollary 5.2. *The problems of deciding whether a DES with a single observable event satisfies language-based opacity and initial-and-final-state opacity are* coNP*-complete.*

Moreover, the transformations of Wu and Lafortune [50] preserve both acyclicity and partial order, and hence we can generalize the results from Chapter 4 for acyclic and partially ordered automata in the same way. On the other hand, the majority of our transformations do not preserve either partial order, due to the utilization of Transformation 4.3, or acyclicity. Consequently, our transformations do not extend these results to the remaining notions discussed.

5.1 LBO to ISO

In this section, we discuss the transformations from language-based opacity to initial-state opacity. The transformation for the case where both the secret and non-secret languages of the language-based opacity problem are prefix closed has been provided by Wu and Lafortune [50]. We now extend this transformation to the general case. We further show that the initial-state opacity decision problem with a single observable event is NL-complete. Consequently, there exists no polynomial-time transformation for this case that preserves the number of observable events, unless P = NP.

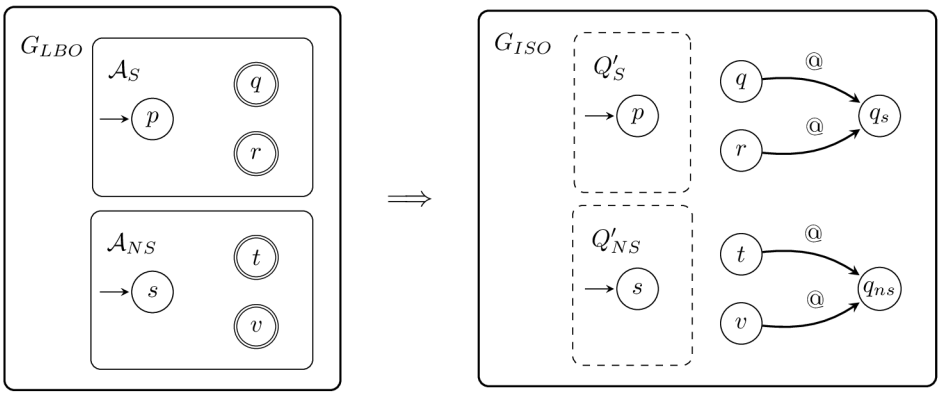


Figure 5.2: Transforming language-based opacity to initial-state opacity.

5.1.1 The general case

Let the language-based opacity problem be represented by a DES G_{LBO}. We transform it to a DES G_{ISO} in such a way that G_{LBO} is language-based opaque if and only if G_{ISO} is initial-state opaque. Our transformation proceeds in two steps:

1. We construct a DES G_{ISO} with one additional observable event @ using Transformation 5.3.
2. We use Transformation 4.3 to reduce the number of observable events of G_{ISO} by one.

Since the second step follows from Transformation 4.3, we only describe the first step, that is, the construction of G_{ISO} over $\Sigma \cup \{@\}$.

Transformation 5.3. Let $G_{LBO} = (Q, \Sigma, \delta, I)$ be a DES with the corresponding projection $P\colon \Sigma^* \to \Sigma_o^*$, a secret language $L_S \subseteq L(G_{LBO})$ given by the non-blocking automaton $\mathcal{A}_S = (Q_S, \Sigma, \delta_S, I_S, F_S)$, and a non-secret language $L_{NS} \subseteq L(G_{LBO})$ given by the non-blocking automaton $\mathcal{A}_{NS} = (Q_{NS}, \Sigma, \delta_{NS}, I_{NS}, F_{NS})$. We construct a DES

$$G_{ISO} = (Q_S \cup Q_{NS} \cup \{q_s, q_{ns}\}, \Sigma \cup \{@\}, \delta', I_S \cup I_{NS})$$

where G_{ISO} is a disjoint union of the automata $\mathcal{A}_S$ and $\mathcal{A}_{NS}$ together with two new states and a new observable event @. The transition function δ' is initialized as $\delta' := \delta_S \cup \delta_{NS}$ and further extended as follows, see Figure 5.2 for an illustration:

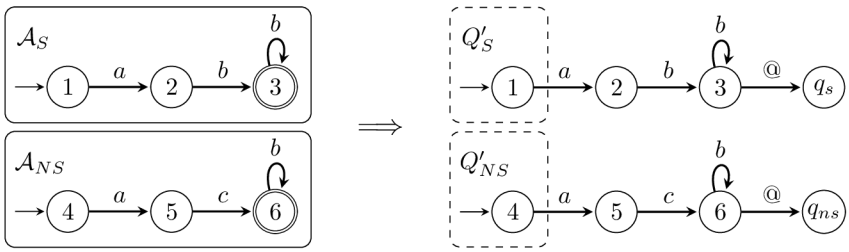


Figure 5.3: An example of the transformation of the LBO problem (left) to the ISO problem (right).

1. for every state $q \in F_S$, we add a new transition $(q, @, q_s)$ to δ';
2. for every state $q \in F_{NS}$, we add a new transition $(q, @, q_{ns})$ to δ'.

Finally, let $Q'_S = I_S$ denote the set of secret initial states of G_{ISO}, and let $Q'_{NS} = I_{NS}$ denote the set of non-secret initial states of G_{ISO}. We extend the projection P to P': $(\Sigma \cup \{@\})^* \to (\Sigma_o \cup \{@\})^*$. ◇

Notice that Transformation 5.3 can be done in polynomial time and that it preserves determinism of transitions.

Example 5.4. Let G over $\Sigma = \{a, b, c\}$ depicted in Figure 5.3 (left) be an instance of the LBO problem from Example 3.2 with the secret language $L_S = abb^*$ and the non-secret language $L_{NS} = acb^*$. Transformation 5.3 of LBO to ISO then results in the DES G' depicted in Figure 5.3 (right) with a new observable event @, a single secret initial state 1, and a single non-secret initial state 4. We distinguish two cases depending on whether event c is observable or not.

In the first case, we assume that event c is unobservable. In this case, G is language-based opaque, because $P(L_S) \subseteq P(L_{NS})$, and the reader can see that $P'(L(G', 1)) = \overline{abb^*@} \subseteq \overline{ab^*@} = P'(L(G', 4))$. Therefore, G' is initial-state opaque.

In the second case, we assume that event c is observable. In this case, G is not language-based opaque, because $ab \in P(L_S)$ whereas $ab \notin P(L_{NS})$, and we can see that $ab \in P'(L(G', 1))$ and $ab \notin P'(L(G', 4))$. Therefore, G' is not initial-state opaque. ◇

The following theorem justifies the correctness of Transformation 5.3.

Theorem 5.5. *A DES G_{LBO} is language-based opaque with respect to L_S, L_{NS}, and P if and only if the DES G_{ISO} obtained by Transformation 5.3 is initial-state opaque with respect to Q'_S, Q'_{NS}, and P'.*

Proof. We show that $P(L_S) \subseteq P(L_{NS})$ if and only if $P'(L(G_{ISO}, Q'_S)) \subseteq P'(L(G_{ISO}, Q'_{NS}))$. However, by construction, $L(G_{ISO}, Q'_S) = \overline{L_S} \cup L_S@$ and $L(G_{ISO}, Q'_{NS}) = \overline{L_{NS}} \cup L_{NS}@$, and hence $P(L_S) \subseteq P(L_{NS})$ if and only if $P'(L(G_{ISO}, Q'_S)) \subseteq P'(L(G_{ISO}, Q'_{NS}))$, which is if and only if G_{ISO} is initial-state opaque. □

We now show that reducing the number of observable events by using Transformation 4.3 does not affect initial-state opacity of any DES.

Theorem 5.6. *A DES $G = (Q, \Sigma, \delta, I)$ is initial-state opaque with respect to Q_S, Q_{NS}, and P if and only if the DES $r(G) = (Q', (\Sigma - \Gamma_o) \cup \{0, 1\}, \delta', I)$ obtained by Transformation 4.3 is initial-state opaque with respect to Q_S, Q_{NS}, and P'.*

Proof. To show that G is initial-state opaque if and only if $r(G)$ is initial-state opaque, we define the languages

- $L_S = L_m(\mathcal{A}_S)$, where $\mathcal{A}_S = (Q, \Sigma, \delta, Q_S, Q)$,
- $L_{NS} = L_m(\mathcal{A}_{NS})$, where $\mathcal{A}_{NS} = (Q, \Sigma, \delta, Q_{NS}, Q)$,
- $L'_S = L_m(\mathcal{A}'_S)$, where $\mathcal{A}'_S = (Q', (\Sigma - \Gamma_o) \cup \{0, 1\}, \delta', Q_S, Q')$, and
- $L'_{NS} = L_m(\mathcal{A}'_{NS})$, where $\mathcal{A}'_{NS} = (Q', (\Sigma - \Gamma_o) \cup \{0, 1\}, \delta', Q_{NS}, Q')$.

Since this transforms initial-state opacity to language-based opacity [50], it is sufficient to show that $P(L_S) \subseteq P(L_{NS})$ if and only if $P'(L'_S) \subseteq P'(L'_{NS})$. However, this can be shown analogously as in the proof of Theorem 4.4. □

Since we need at least two initial states for initial-state opacity to be non-trivial, we generalize the weaker form of Theorem 4.7 to initial-state opacity. Therefore, using Transformations 5.3 and 4.3, and taking into account the fact that the problem of verifying initial-state opacity is in PSPACE [37], we can state the following result for NFAs with deterministic transition function.

Corollary 5.7. *The problem of deciding whether a DES satisfies initial-state opacity is* PSPACE*-complete. The problem remains* PSPACE*-complete even if the system is an NFA with deterministic transition function and three events, one of which is unobservable.*

5.1.2 The case of $|\Sigma_o| = 1$

To preserve the number of observable events, the general transformation relies on the binary encoding of events by Transformation 4.3. However, the encoding requires at least two observable events in G_{LBO}, and hence it is not applicable to systems with a single observable event. In fact, we show that there does not exist such a transformation unless P = NP, which is a longstanding open problem of computer science. Deciding language-based opacity for systems with a single observable event is coNP-complete [23, 47]. We show that deciding initial-state opacity for systems with a single observable event is NL-complete, and hence efficiently solvable on a parallel computer [2]. In particular, the problem can be solved in polynomial time.

Theorem 5.8. *Deciding initial-state opacity of a DES with a single observable event is* NL*-complete.*

Proof. Deciding initial-state opacity is equivalent to checking the inclusion of two prefix-closed languages. Namely, a DES G with $\Sigma_o = \{a\}$ is initial-state opaque with respect to the secret states Q_S and the non-secret states Q_{NS} if and only if $K_S \subseteq K_{NS}$ for $K_S = P(L(G, Q_S))$ and $K_{NS} = P(L(G, Q_{NS}))$. Since the languages K_S and K_{NS} are prefix-closed, they are either finite, consisting of at most $|Q|$ strings, or equal to $\{a\}^*$.

To show that the problem belongs to NL, we show how to verify $K_S \not\subseteq K_{NS}$ in nondeterministic logarithmic space. Then, since NL is closed under complement [26, 48], $K_S \subseteq K_{NS}$ belongs to NL. Thus, to check that $K_S \not\subseteq K_{NS}$ in nondeterministic logarithmic space, we guess $k \in \{0, \ldots, |Q|\}$ in binary, store it in logarithmic space, and verify that $a^k \in K_S$ and $a^k \notin K_{NS}$. To verify $a^k \in K_S$, we guess a path in G step by step, storing only the current state, and counting the number of steps by decreasing k by one in each step; logarithmic space is sufficient for this. Since $a^k \notin K_{NS}$ belongs to the complement of NL, which coincides with NL, we can check $a^k \notin K_{NS}$ in nondeterministic logarithmic space as well.

To show that deciding initial-state opacity for DESs with a single observable event is NL-hard, we reduce the DAG reachability problem [29]: given a DAG $\mathcal{G} = (V, E)$ and nodes $s, t \in V$, the problem asks whether t is reachable from s. From $\mathcal{G}$, we construct a DES $\mathcal{A} = (V \cup \{i\}, \{a\}, \delta, \{s, i\})$, where i is a new initial state and a is an observable event, as follows. With each node of $\mathcal{G}$, we associate a state in $\mathcal{A}$. Whenever there is an edge from j to k in $\mathcal{G}$, we add a transition (j, a, k) to $\mathcal{A}$. We add a self-loop by a to state t and to

state i. The set of secret initial states is $Q_S = \{i\}$ and the set of non-secret initial states $Q_{NS} = \{s\}$. Then, $\mathcal{A}$ is initial-state opaque if and only if there is a path from s to t in $\mathcal{G}$. Indeed, $L(\mathcal{A}, i) = \{a\}^*$ is included in $L(\mathcal{A}, s)$ if and only if $L(\mathcal{A}, s) = \{a\}^*$, which is if and only if t is reachable from s. □

5.2 CSO to TO

In this section, we discuss the transformations from current-state opacity to trace opacity. The transformation we provide results in a system with at least two observable events. Similar to initial-state opacity, we show that the trace opacity decision problem with a single observable event is NL-complete. Consequently, there exists no polynomial-time transformation for this case that preserves the number of observable events, unless P = NP.

5.2.1 The general case

Let the current-state opacity problem be represented by a DES G_{CSO}. We transform it to a DES G_{TO} in such a way that G_{CSO} is current-state opaque if and only if G_{TO} is trace opaque. Our transformation proceeds in two steps:

1. We construct a DES G_{TO} with one additional observable event @ using Transformation 5.9.
2. We use Transformation 4.3 to reduce the number of observable events of G_{TO} by one.

Since the second step follows from Transformation 4.3, we only describe the first step, that is, the construction of G_{CSO} over $\Sigma \cup \{@\}$.

Transformation 5.9. Let $G_{CSO} = (Q, \Sigma, \delta, I)$ be a DES with the secret states Q_S, the non-secret states Q_{NS}, and the corresponding projection $P\colon \Sigma^* \to \Sigma_o^*$. We construct a DES

$$G_{TO} = (Q \cup \{q_s, q_{ns}\}, \Sigma \cup \{@, \sigma\}, \delta', I)$$

where q_s and q_{ns} are new states, @ is a new observable event, and σ is a new unobservable secret event. The transition function δ' is initialized as the transition function δ of the system G_{CSO} and further extended as follows, see Figure 5.4 for an illustration:

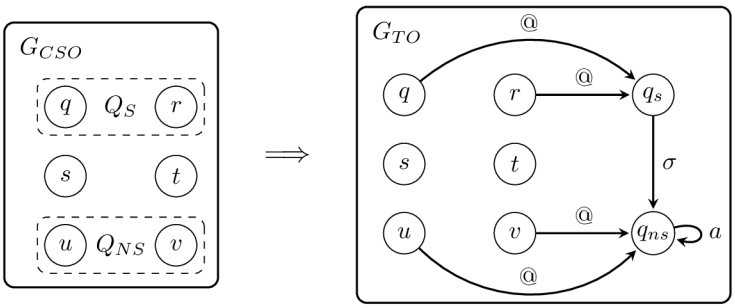


Figure 5.4: Transforming current-state opacity to trace opacity.

1. for every secret state $q \in Q_S$, we add the transition $(q, @, q_s)$ to δ',
2. for every non-secret state $q \in Q_{NS}$, we add the transition $(q, @, q_{ns})$ to δ',
3. we add the secret transition (q_s, σ, q_{ns}) to δ', and
4. we add one self-loop transition (q_{ns}, a, q_{ns}) to δ', where $a \in \Sigma_o$ is an arbitrary observable event.

We define the projection $P' \colon (\Sigma \cup \{@, \sigma\})^* \to (\Sigma_o \cup \{@\})^*$ and the set of secret events $S = \{\sigma\}$. ◇

Notice that Transformation 5.9 can be done in polynomial-time and that it preserves determinism.

Example 5.10. Let G over $\Sigma = \{a, b, c\}$ depicted in Figure 5.5 (left) be an instance of the CSO problem from Example 3.6 with the secret states $Q_S = \{2\}$ and the non-secret states $Q_{NS} = \{5\}$. Transformation 5.9 of CSO to TO then results in the DES G' depicted in Figure 5.5 (right) with a new observable event @ and a new unobservable secret event σ. We distinguish two cases depending on whether event c is observable or not.

If event c is unobservable, then G is current-state opaque, because the only string leading to the secret state, state 2, is the string a, for which the string ac leading to the non-secret state, state 5, satisfies that $P(a) = P(ac)$. Then, the reader can see that G' is trace opaque, because all possible secret traces are of the form $a@\sigma a^i \in Sec(G')$, for $i \in \mathbb{N}$, and for every such trace there is a non-secret trace $ac@a^i \in Pub(G')$ such that $P'(a@\sigma a^i) = P'(ac@a^i)$.

If event c is observable, then G is not current-state opaque, because the only string leading to the non-secret state, string ac, has a different obser-

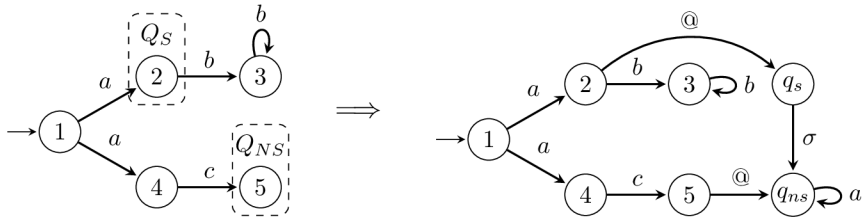


Figure 5.5: An example of the transformation of the CSO problem (left) to the TO problem (right).

vation than the string a leading to the secret state, that is, $P(ac) \neq P(a)$. Consequently, the reader can verify that G' is not trace opaque, since now $P'(a@\sigma a) = a@a \in P'(Sec(G'))$ while $a@a \notin P'(Pub(G'))$. ◇

The following theorem justifies the correctness of Transformation 5.9.

Theorem 5.11. *A DES G_{CSO} is current-state opaque with respect to Q_S, Q_{NS}, and P if and only if the DES G_{TO} obtained by Transformation 5.9 is trace opaque with respect to S and P'.*

Proof. Assume that the DES G_{CSO} is not current-state opaque. Then, there exists a string $w \in \Sigma^*$ that leads the system G_{CSO} to a secret state q, while every string that looks the same as the string w leads G_{CSO} out of non-secret states. In G_{TO} we have that $\delta'(I, w@) = \delta'(q, @) = \{q_s\}$ and $w@\sigma a \in Sec(G_{TO})$ is a secret trace. Since generating any string that looks the same as the string w leads the system G_{CSO} to a state out of non-secret states, then in G_{TO} we have that $\delta'(I, P'^{-1}P'(w)@) = \{q_s\}$. Evidently, every extension of a trace from q_s makes the trace secret, and hence we have $P'(w@\sigma a) \notin P'(Pub(G_{TO}))$. Therefore, the system G_{TO} is not trace opaque.

On the other hand, assume that the system G_{CSO} is current-state opaque, and let $w = w_1\sigma w_2 \in Sec(G_{TO})$ be a secret trace. Then, the string w_1 is of the form $w_1 = v@$ where v contains neither @ nor σ, and $w_2 = a^k$ for $k \in \mathbb{N}$. By construction, generating the string v in G_{CSO} ends up in a secret state. Since the system G_{CSO} is current-state opaque, there is a string $v' \in P^{-1}P(v)$ such that generating v' in G_{CSO} ends up in a non-secret state. Then, generating the trace $v'@$ in G_{TO} ends up in a state q_{ns}, and hence taking the strings $w_1' = v'@$ and $w_2' = w_2 = a^k$ results in trace $w' = w_1'w_2' \in Pub(G_{TO})$ such that $P'(w') = P'(w_1'w_2') = P'(w_1\sigma w_2) = P'(w)$ and $\delta'(I, w') \neq \emptyset$, showing that the system G_{TO} is trace opaque. □

We now show that reducing the number of observable events by using Transformation 4.3 does not affect trace opacity of any DES.

Theorem 5.12. *A DES G is trace opaque with respect to S and P if and only if the DES $r(G)$ obtained by Transformation 4.3 is trace opaque with respect to S and P'.*

Proof. To show that G is trace opaque if and only if the system $r(G)$ is trace opaque, it is sufficient to show that $P(Sec(G)) \subseteq P(Pub(G))$ if and only if $P'(Sec(r(G))) \subseteq P'(Pub(r(G)))$. However, this can be shown analogously to Theorem 4.4. □

Since Transformation 5.9 introduces a new unobservable secret event, we cannot directly generalize Theorem 4.7 to trace opacity. However, by utilizing Theorem 5.12 and the membership of trace opacity in PSPACE [19], we can state a weaker result as follows.

Corollary 5.13. *The problem of deciding whether a DES satisfies trace opacity is* PSPACE*-complete. The problem remains* PSPACE*-complete even if the system is a DFA with four events, two of which are unobservable.*

5.2.2 The case of $|\Sigma_o| = 1$

The second step of our transformation, that is the binary encoding represented by Transformation 4.3, requires that G_{CSO} has at least three observable events or, equivalently, that G_{TO} has at least two observable events. Consequently, our transformation does not preserve the number of observable events if G_{TO} has a single observable event. We show that the trace opacity decision problem with a single observable event is NL-complete, and hence efficiently solvable on a parallel computer [2]. In particular, the problem can be solved in polynomial time.

Theorem 5.14. *Deciding trace opacity of a DES with a single observable event is* NL*-complete.*

Proof. Deciding trace opacity is equivalent to checking the inclusion of two languages. Namely, a DES G with $\Sigma_o = \{a\}$ is trace opaque with respect to the set of secret events S if and only if $K_S \subseteq K_{NS}$ for $K_S = P(Sec(G))$ and $K_{NS} = P(Pub(G))$. Since the language K_{NS} is prefix-closed, it is either finite, consisting of at most $|Q|$ strings, or equal to $\{a\}^*$. Similarly, K_S is

either finite or $K_S = \{a\}^* - L_{fin}$, where $L_{fin} \subseteq \{a^i \mid i < |Q|\}$ is finite, because for any secret trace $u\sigma v \in Sec(G)$, where $\sigma \in S$ and $u, v \in \Sigma^*$, we have that $u\sigma v' \in Sec(G)$ for any $v' \in \overline{v}$. Therefore, we can use the same algorithm to verify $K_S \not\subseteq K_{NS}$ in nondeterministic logarithmic space as in the proof of Theorem 5.8.

To show that deciding trace opacity for DESs with a single observable event is NL-hard, we again reduce the DAG reachability problem [29]: given a DAG $\mathcal{G} = (V, E)$ and nodes $s, t \in V$, the problem asks whether t is reachable from s. From $\mathcal{G}$, we construct a DES $\mathcal{A} = (V \cup \{q_1, q_2\}, \{a, \sigma\}, \delta, \{s, q_1\})$, where q_1 and q_2 are new states, a is an observable event, and σ is an unobservable secret event. With each node of $\mathcal{G}$, we associate a state in $\mathcal{A}$. Whenever there is an edge from j to k in $\mathcal{G}$, we add the transition (j, a, k) to δ. Further, we add the secret transition (q_1, σ, q_2) and two self-loops (t, a, t) and (q_2, a, q_2) to δ. Then, $\mathcal{A}$ is trace opaque if and only if there is a path from s to t in $\mathcal{G}$. Indeed, we have $Sec(\mathcal{A}) = L(\mathcal{A}, q_1) = \sigma a^*$, and hence $P(Sec(\mathcal{A})) = \{a\}^*$ is included in $P(Pub(\mathcal{A})) = L(\mathcal{A}, s)$ if and only if $Pub(\mathcal{A}) = \{a\}^*$, which is if and only if t is reachable from s. □

5.3 TO to CSO

In this section, we show how to transform trace opacity to current-state opacity. Previously, such a transformation was provided by Cassez et al. [17], but they assumed that a deterministic automaton $\mathcal{A}_S$ for the language of secret traces was given as input. Additionally, for a nondeterministic $\mathcal{A}_S$ their transformation is not polynomial. We improve this result by providing a transformation from trace opacity to current-state opacity that is always polynomial. Further, our transformation enables us to construct automata $\mathcal{A}_S$ and $\mathcal{A}_{NS}$ representing the secret and non-secret trace languages, thus transforming the problem also to language-based opacity problem.

Let the trace opacity problem be represented by a DES G_{TO}. We transform it to a DES G_{CSO} in such a way that G_{TO} is trace opaque if and only if G_{CSO} is current-state opaque.

Transformation 5.15. Let $G_{TO} = (Q, \Sigma, \delta, I)$ be a DES with the set of secret events $S \subseteq \Sigma_{uo}$ and the corresponding projection $P\colon \Sigma^* \to \Sigma_o^*$. We construct a DES

$$G_{CSO} = (Q \cup Q_S, \Sigma, \delta', I)$$

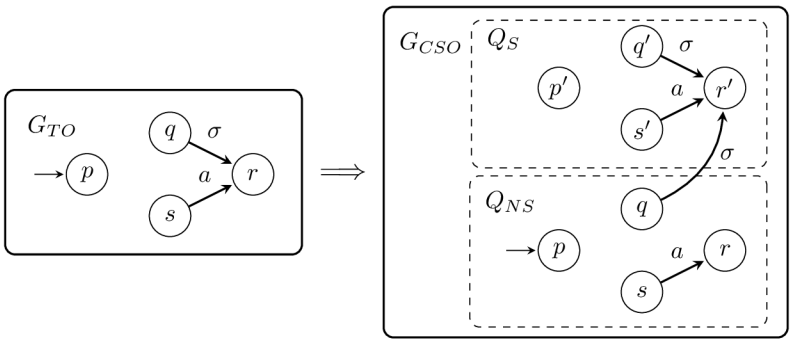


Figure 5.6: Transforming trace opacity to current-state opacity.

as a disjoint union of G and $G_s = (Q_S, \Sigma, \delta_s, I_s)$, where G_s is copy of G and $Q_S = \{q' \mid q \in Q\}$ is a disjoint copy of Q. We initialize $\delta' := \delta \cup \delta_s$ and further modify it by replacing every secret transition (q, σ, r) originally in δ by transition (q, σ, r') in δ', where $\sigma \in S$ and $r' \in Q_S$, cf. Figure 5.6 for an illustration. The states of Q_S are the secret states of G_{CSO}, while the rest of the states are non-secret, i.e., $Q_{NS} = Q$. Finally, we remove unreachable states and corresponding transitions. $\diamond$

Notice that Transformation 5.15 can be done in polynomial time and that it preserves determinism and the number of observable and unobservable events.

Remark 5.16. *To reduce G_{TO} to language-based opacity, we set $L_S = L(\mathcal{A}_S)$ and $L_{NS} = L(\mathcal{A}_{NS})$, where $\mathcal{A}_S = (Q \cup Q_S, \Sigma, \delta', I, Q_S)$ is identical to the DES G_{CSO} from Transformation 5.15, except for the set of marked states, and $\mathcal{A}_{NS} = (Q, \Sigma, \delta'', I, Q)$ is an automaton that corresponds to the original system G_{TO} with all states marked and with all secret transitions removed, that is, $\delta'' = \delta \cap Q \times (\Sigma - S) \times Q$.*

We now provide an illustrative example.

Example 5.17. Let G over $\Sigma = \{a, b, c, \sigma\}$ depicted in Figure 5.7 (left) be an instance of the TO problem from Example 3.4 with the set of secret events $S = \{\sigma\}$. Transformation 5.15 of TO to CSO then results in the DES G' depicted in Figure 5.7 (right) with the set of secret states $Q_S = \{2', 3'\}$ and the set of non-secret states $Q_{NS} = \{1, 4, 5\}$. Note that states 2, 3, 1′, 4′, and 5′ were unreachable in G', and therefore were removed at the end of the

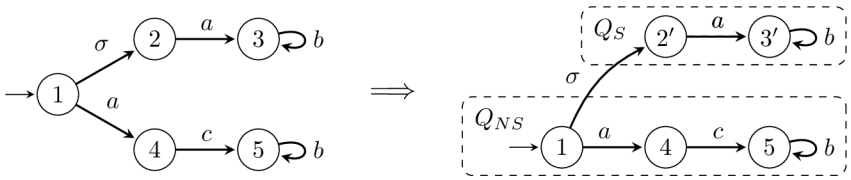


Figure 5.7: An example of the transformation of the TO problem (left) to the CSO problem (right).

transformation. We distinguish two cases depending on whether event c is observable or not.

First, we assume that event c is unobservable. In this case, G is trace opaque because $P(Sec(G)) = P(Pub(G))$. In G', the initial state 1 is non-secret, and therefore, for a string consisting of only the event σ, the empty string ε is such that $P(\sigma) = P(\varepsilon)$ and $\delta(1, \varepsilon) \cap Q_{NS} = \{1\}$. Similarly, if a string of the form σab^* leads G' to the secret state $3'$, then there is a string of the form acb^* with $P(\sigma ab^*) = P(acb^*)$ that leads G' to the non-secret state 5. Thus, G' is current-state opaque.

In the second case, we assume that event c is observable. In this case, G is not trace opaque because $ab \in P(Sec(G))$ while $ab \notin P(Pub(G))$, and the reader can see that $\delta'(1, \sigma ab) \cap Q_S \neq \emptyset$ while $\delta'(1, P^{-1}P(\sigma ab)) \cap Q_{NS} = \emptyset$. Therefore, G' is not current-state opaque. ◇

The following theorem justifies the correctness of Transformation 5.15.

Theorem 5.18. *A DES G_{TO} is trace opaque with respect to S and P if and only if the DES G_{CSO} obtained by Transformation 5.15 is current-state opaque with respect to Q_S, Q_{NS}, and P.*

Proof. Assume that the system G_{TO} is trace opaque. To show that the system G_{CSO} is current-state opaque, we consider a string w such that $\delta'(I, w) \cap Q_S \neq \emptyset$, and show that there is a string w' such that $P(w) = P(w')$ and $\delta'(I, w') \cap Q_{NS} \neq \emptyset$. To reach the set of secret states Q_S, the string w must be of the form $w_1 \sigma w_2$. By construction, there is a state $q \in \delta(I, w_1\sigma)$ in G_{TO}, such that there is a state $q' \in \delta'(I, w_1\sigma) \cap Q_S$ in the system G_{CSO}, and the string w_2 can be generated from the state q'. Therefore, we can generate the string w_2 from the state q in G_{TO} and $\delta(I, w_1\sigma w_2) \neq \emptyset$, that is, $w_1\sigma w_2 \in Sec(G_{TO})$ is a secret trace of G_{TO}. However, trace opacity of G_{TO} implies that there is a string $w' \in Pub(G_{TO})$ such that $P(w_1\sigma w_2) = P(w')$

and $\delta(I, w') \neq \emptyset$. In particular, w' does not contain σ, and thus we obtain $\delta'(I, w') \cap Q_{NS} \neq \emptyset$, which completes this part of the proof.

If the system G_{TO} is not trace opaque, then there exists a secret trace $w = w_1\sigma w_2 \in Sec(G_{TO})$ such that $P(w) \notin P(Pub(G_{TO}))$. In G_{CSO}, after generating σ we can only reach secret states, and therefore $\emptyset \neq \delta'(\delta'(I, w_1\sigma) \cap Q_S, w_2) = \delta'(I, w_1\sigma w_2) \subseteq Q_S$. Since the language marked by the set Q_{NS} in G_{CSO} equals to $Pub(G_{TO})$, then for every string $w' \in L(G_{CSO})$ with $P(w) = P(w_1\sigma w_2) = P(w')$, we have that $\delta'(I, w') \cap Q_{NS} = \emptyset$. Therefore, the system G_{CSO} is not current-state opaque. □

5.4 CSO to k-SSO

In this section, we show how to transform current-state opacity to strong k-step opacity. For systems without neutral states, strong k-step opacity implies weak k-step opacity [49], and thus the following transformations are also applicable to weak k-step opacity. Again, the general transformation uses Transformation 4.3 to preserve the number of observable events, and therefore we provide a separate transformation for systems with a single observable event.

5.4.1 The general case

Let the current-state opacity problem be represented by a DES G_{CSO}. We transform it to a deterministic DES $G_{k\text{-}SSO}$ in such a way that G_{CSO} is current-state opaque if and only if $G_{k\text{-}SSO}$ is strongly k-step opaque.

Our transformation proceeds in three steps:

1. If G_{CSO} is not deterministic, we determinize it by Transformation 4.5.
2. We construct a DES $G_{k\text{-}SSO}$ with one additional observable event @ using Transformation 5.19.
3. We use Transformation 4.3 to reduce the number of observable events of $G_{k\text{-}SSO}$ by one.

Since the first and third step follow from Transformations 4.5 and 4.3, we only describe the second step, that is, the construction of $G_{k\text{-}SSO}$ over $\Sigma \cup \{@\}$.

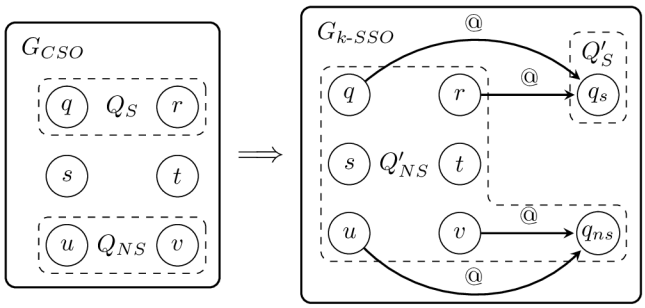


Figure 5.8: Transforming current-state opacity to strong k-step opacity, for an arbitrary parameter $k \in \mathbb{N}_\infty$.

Transformation 5.19. Let $G_{CSO} = (Q, \Sigma, \delta, q_0)$ be a deterministic DES with the secret states Q_S, the non-secret states Q_{NS}, and the corresponding projection $P\colon \Sigma^* \to \Sigma_o^*$. We construct a DES

$$G_{k\text{-}SSO} = (Q \cup \{q_s, q_{ns}\}, \Sigma \cup \{@\}, \delta', q_0)$$

where q_s and q_{ns} are new states and @ is a new observable event. The transition function δ' is initialized as the transition function δ of the system G_{CSO} and further extended as follows, see Figure 5.8 for an illustration:

1. for every secret state $q \in Q_S$, we add the transition $(q, @, q_s)$ to δ', and
2. for every non-secret state $q \in Q_{NS}$, we add the transition $(q, @, q_{ns})$ to δ'.

We define the projection $P'\colon (\Sigma \cup \{@\})^* \to (\Sigma_o \cup \{@\})^*$, and the sets of secret states $Q'_S = \{q_s\}$ and of non-secret states $Q'_{NS} = Q \cup \{q_{ns}\}$. ◇

Notice that Transformation 5.19 can be done in polynomial time and that it preserves determinism. It is also independent of the parameter k, and therefore works for any $k \in \mathbb{N}_\infty$ without affecting the size of the resulting system $G_{k\text{-}SSO}$.

Intuitively, since there is no extension from the unique secret state q_s, there is always a corresponding (trivial) extension from every non-secret state. Consequently, we can apply Transformation 4.3 to $G_{k\text{-}SSO}$ and encode new event @ in binary without affecting strong k-step opacity of the system $G_{k\text{-}SSO}$.

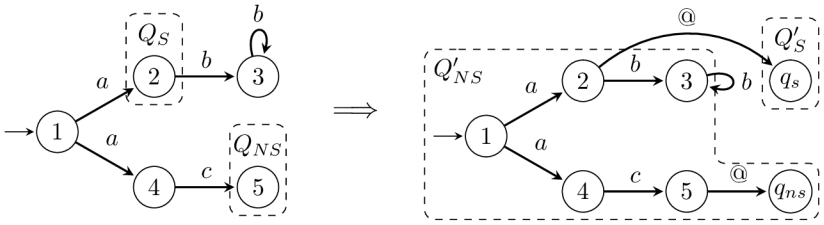


Figure 5.9: An example of the transformation of the CSO problem (left) to the k-SSO problem (right).

Remark 5.20. *Transformation 5.19 can also be used to remove neutral states from the system, so can we think of it as a transformation from current-state opacity with neutral states to current-state opacity without neutral states.*

We now provide an illustrative example.

Example 5.21. Let G over $\Sigma = \{a, b, c\}$ depicted in Figure 5.9 (left) be an instance of the CSO problem from Example 3.6 with the secret states $Q_S = \{2\}$ and the non-secret states $Q_{NS} = \{5\}$. Transformation 5.19 of CSO to k-SSO then results in the DES G' depicted in Figure 5.9 (right) with a new observable event @ and two new states q_s and q_{ns}, where q_s is the unique secret state of G'. We distinguish two cases depending on whether event c is observable or not.

If event c is unobservable, then G is current-state opaque, because the only string leading to the secret state, state 2, is the string a, for which the string ac leading to the non-secret state, state 5, satisfies that $P(a) = P(ac)$. Then, the reader can see that G' is strongly k-step opaque, for any $k \in \mathbb{N}_\infty$, because the only possible string leading to the secret state, state q_s, is the string $a@$, for which there is the string $ac@$ such that $P'(a@) = P'(ac@)$ and G' never enters a secret state by generating $\delta'(1, ac@)$.

If event c is observable, then G is not current-state opaque, since now we have $P(ac) \neq P(a)$. Consequently, the reader can verify that G' is not current-state opaque since $\delta'(1, P'^{-1}P'(a@)) = \{q_s\}$, and hence G' is neither strongly k-step opaque, for any parameter $k \in \mathbb{N}_\infty$. ◇

The following theorem justifies the correctness of Transformation 5.19.

Theorem 5.22. *A DES G_{CSO} is current-state opaque with respect to Q_S, Q_{NS}, and P if and only if the DES $G_{k\text{-}SSO}$ obtained by Transformation 5.19 is strongly k-step opaque, for any parameter $k \in \mathbb{N}_\infty$, with respect to Q'_S and P'.*

Proof. Assume that the DES G_{CSO} is not current-state opaque. Then, there exists a string $w \in \Sigma^*$ that leads the system G_{CSO} to a secret state, while every string that looks the same as the string w leads the system G_{CSO} out of non-secret states. Then, in the system $G_{k\text{-}SSO}$, generating the string $w@$ ends up in the secret state $q_s \in \delta'(q_0, w@) \cap Q'_S$. Since generating any string that looks the same as the string w leads the system G_{CSO} to a state out of non-secret states, we have that $\delta'(q_0, P'^{-1}P'(w@)) \cap Q'_{NS} = \emptyset$. Therefore, the system $G_{k\text{-}SSO}$ is not current-state opaque, and hence neither strongly k-step opaque, for any parameter $k \in \mathbb{N}_\infty$.

On the other hand, assume that the system G_{CSO} is current-state opaque, and let the string $st \in L(G_{k\text{-}SSO})$ be such that the string s leads the system $G_{k\text{-}SSO}$ to a secret state and the string t may be generated from this secret state in $G_{k\text{-}SSO}$, formally $\delta'(\delta'(q, s) \cap Q'_S, t) \neq \emptyset$. Since Q'_S contains a single secret state with no outgoing transition, then the string s is of the form $s = s_1@$, where s_1 does not contain $@$, and $t = \varepsilon$. By construction, generating the string s_1 in G_{CSO} ends up in a secret state. Since the system G_{CSO} is current-state opaque, there is a string $s'_1 \in P^{-1}P(s_1)$ such that generating s'_1 in G_{CSO} ends up in a non-secret state. Then, by generating the string $s'_1@$, $G_{k\text{-}SSO}$ ends up in the non-secret state q_{ns} and for every prefix $w \in \overline{s'_1@}$ we have $\delta'(q_0, w) \notin Q'_S$. Therefore, if we take the string $s' = s'_1@$, then $G_{k\text{-}SSO}$ never enters a secret state and $P'(s') = P'(s) = P'(st)$, showing that the system $G_{k\text{-}SSO}$ is strongly k-step opaque, for any parameter $k \in \mathbb{N}_\infty$. □

In Theorem 4.7 we showed that the problem of deciding current-state opacity of a DES modeled by a DFA with three events, one of which is unobservable, is PSPACE-complete. Transformations 5.19 and 4.3 allow us to transform instance of this problem to the problems of deciding weak and strong k-step opacity while preserving determinism and the number of observable events. Thus, we can state the following result.

Corollary 5.23. *Given a natural number k represented by $O(\log(k))$ bits and a DES G. The problems of deciding whether the system G satisfies weak k-step opacity and strong k-step opacity are* PSPACE*-hard. The problems remain* PSPACE*-hard even if the system G is a DFA with three events, one of which is unobservable.*

Since weak ∞-step opacity is a special case of weak k-step opacity, the previous corollary also implies PSPACE-hardness for weak ∞-step opacity.

5.4.2 The case of $|\Sigma_o| = 1$

To preserve the number of observable events, the general transformation relies on the binary encoding of events by Transformation 4.3. However, the encoding requires at least two observable events in G_{CSO}, and hence it is not applicable to systems with a single observable event. For these systems, we provide a separate transformation that requires to add $k+1$ new states, and therefore the size of the resulting system is linear with respect to the parameter $k \in \mathbb{N}$.

Let the current-state opacity problem with a single observable event be represented by a DES G^a_{CSO} without neutral states. We transform it to a DES $G^a_{k\text{-}SSO}$ in such a way that G^a_{CSO} is current-state opaque if and only if $G^a_{k\text{-}SSO}$ is strongly k-step opaque.

Without loss of generality, we assume that G^a_{CSO} is deterministic, as we can always use Transformation 4.5 to determinize it. We further assume that in G^a_{CSO}, there are no non-secret states that can be reached from a secret state by any sequence of unobservable events, formally $\delta(Q_S, P^{-1}(\varepsilon)) \cap Q_{NS} = \emptyset$. We describe this property with respect to current-state opacity of the system in the following lemma.

Lemma 5.24. *A DES G is current-state opaque with respect to Q_S, Q_{NS}, and P if and only if G is current-state opaque with respect to $Q'_S = Q_S - R$, $Q'_{NS} = Q_{NS} \cup R$, and P, where $R = \{q_s \in Q_S \mid \delta(q_s, P^{-1}(\varepsilon)) \cap Q_{NS} \neq \emptyset\}$.*

Proof. We show that any state $X \subseteq Q$ in the observer G^{obs} of G contains a non-secret state from Q_{NS} if and only if X contains a non-secret state from Q'_{NS}. Evidently, if X contains a non-secret state from Q_{NS}, then X also contains a non-secret state from Q'_{NS}, since $Q_{NS} \subseteq Q'_{NS}$. On the other hand, let $q \in X \cap R$ be a newly added state to Q'_{NS}, then there is another state $p \in Q_{NS}$ such that $p \in \delta(q, P^{-1}(\varepsilon))$, and therefore $p \in X \cap Q_{NS}$. □

Transformation 5.25. Let $G^a_{CSO} = (Q, \Sigma, \delta, q_0)$ be a deterministic DES with a single observable event $\Sigma_o = \{a\}$, the secret states Q_S, the non-secret states $Q_{NS} = Q - Q_S$, and the corresponding projection $P\colon \Sigma^* \to \{a\}^*$. By Lemma 5.24, we assume that $\delta(Q_S, P^{-1}(\varepsilon)) \cap Q_{NS} = \emptyset$. We construct a DES

$$G^a_{k\text{-}SSO} = (Q \cup \{q^\star_0, \ldots, q^\star_k\}, \Sigma \cup \{u\}, \delta', q_0)$$

by adding $k+1$ new non-secret states and a new unobservable event u. The transition function δ' is initialized as the transition function δ of the system G^a_{CSO} and further extended as follows, see Figure 5.10 for an illustration:

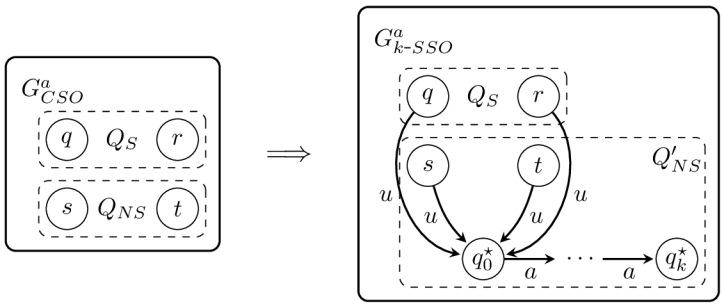


Figure 5.10: Transforming current-state opacity with a single observable event to strong k-step opacity.

1. for every state $q \in Q$, we add a new transition $(q, u, q^\star_0)$ to δ';
2. for every state $q^\star_i$, where $i \in \{0, \ldots, k-1\}$, we add a new transition $(q^\star_i, a, q^\star_{i+1})$ to δ'.

The set of secret states Q_S remains unchanged in $G_{k\text{-}SSO}$, while all other states are non-secret. We extend the projection P to $P'\colon (\Sigma \cup \{u\})^* \to \{a\}$. $\diamond$

Notice that Transformation 5.25 can be done in polynomial time and that it preserves determinism and the number of observable events.

Remark 5.26. *It seems that adding k new states to the system cannot be avoided, since for $k \geq |Q|$ the problem of deciding strong k-step opacity of a system with a single observable event can be solved in polynomial time. First, we search the system for a cycle containing only non-secret states and at least one observable transition. Then, we verify if the system is strongly k-step opaque in the first $|Q|$ observable steps before the cycle is reached (if it exists). Clearly, both conditions can verified in polynomial time.*

We now provide an illustrative example.

Example 5.27. Let G over $\Sigma = \{a, u_1\}$ depicted in Figure 5.11 (left) be an instance of the CSO problem with a single observable event $\Sigma_o = \{a\}$, the set of secret states $Q_S = \{2\}$, and the set of non-secret states $Q_{NS} = \{1, 3\}$. Transformation 5.25 of CSO to 2-SSO results in the DES G' depicted in Figure 5.11 (right) with a new unobservable event u_2, the set of secret states Q_S, and the set of non-secret states $Q'_{NS} = Q_{NS} \cup \{q^\star_0, q^\star_1, q^\star_2\}$. We consider two cases based on the presence of the unobservable transition $(1, u_1, 2)$ in G.

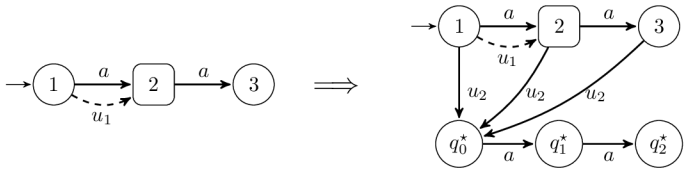


Figure 5.11: An example of the transformation of the CSO problem with a single observable event (left) to the k-SSO problem (right); the secret states are squared and all other states are non-secret.

We first assume that the transition $(1, u_1, 2)$ exists in G. Then, G is current-state opaque because the only string leading to the secret state, state 2, is the string a, for which the string u_1a leading to the non-secret state, state 3, satisfies that $P(a) = P(au_1)$. The reader can verify that G' is strongly 2-step opaque, because for every string $s \in L(G')$ there is a string with the same observation that does not go through a secret state in last 2 observable steps. If $|P'(s)| \leq 2$, then there is a string u_2aa and its prefixes such that G' never enters a secret state. If $|P'(s)| > 2$, then there are strings $w_1 = u_1au_2aa$ and $w_2 = aau_2aa$ such that each prefix $w_i' \in \overline{w}_i$ with $|P'(w_i)| - |P'(w_i')| \leq 2$, where $i \in \{1, 2\}$, leads G' to one of the non-secret states 3, $q_0^\star$, $q_1^\star$, or $q_2^\star$.

If the transition $(1, u_1, 2)$ is not present in G, then G is not current-state opaque, and therefore G' is not strongly 2-step opaque. Indeed, by observing a string $aaa \in P(L(G'))$ the intruder knows that G' has visited the secret state 2 during last two steps, since for every string with the same observation, such as $w_1 = au_2aa$ and $w_2 = aau_2a$, there exists prefix $w' = a$ such that $|aaa| - |P'(w')| \leq 2$ and $\delta'(1, w') = 2 \in Q_S$. ◇

The following theorem justifies the correctness of Transformation 5.25.

Theorem 5.28. *A DES G^a_{CSO} with a single observable event $\Sigma_o = \{a\}$ is current-state opaque with respect to Q_S, Q_{NS}, and P if and only if the DES $G^a_{k\text{-}SSO}$ obtained by Transformation 5.25 is strongly k-step opaque with respect to Q_S and P'.*

Proof. Assume that the DES G^a_{CSO} is not current-state opaque. Then, there exists a string $w \in \Sigma^*$ that leads the system G^a_{CSO} to a secret state, while every string that looks the same as the string w leads the system G^a_{CSO} out of non-secret states. Since there are no neutral states in G^a_{CSO}, we denote $Z = \delta(q_0, P^{-1}P(w)) \subseteq Q_S$ the set of secret states under observation of $P(w)$.

In $G^a_{k\text{-}SSO}$, we have that $\delta'(q_0, P'^{-1}P'(w)) \cap Q = Z$ and string ua^k can be generated from every state in Z. Let $s = wua^k \in L(G^a_{k\text{-}SSO})$ be a string that can be generated in $G^a_{k\text{-}SSO}$. We show that by generating any $v \in L(G^a_{k\text{-}SSO})$ with $P'(s) = P'(v)$ the system $G^a_{k\text{-}SSO}$ must have visited a secret state in last k observable steps. If $\delta'(q_0, v) \in Q$, then there is a prefix $v' \in \overline{v}$ such that $\delta'(q_0, v') \in Z$ and $|P'(v')| = |P'(v)| - k = |P(w)|$. On the other hand, if $\delta'(q_0, v) \in \{q^\star_0, \ldots, q^\star_k\}$, then v is of the form $v = v_1uv_2$, where $|uv_2| \leq k$. Thus, there is a prefix $v'_1 \in \overline{v}_1$ such that $|P'(v'_1)| = |P'(v)| - k = |P(w)|$ and $\delta'(q_0, v'_1) \in Z$. Since $Z \subseteq Q_S$, the system $G^a_{k\text{-}SSO}$ is not strongly k-step opaque.

On the other hand, assume that the system G^a_{CSO} is current-state opaque and let $s \in L(G^a_{k\text{-}SSO})$. We show that there is $w \in L(G^a_{k\text{-}SSO})$ such that $P'(s) = P'(w)$ and for every $w' \in \overline{w}$, if $|P'(w)| - |P'(w')| \leq k$, then $\delta'(q_0, w') \notin Q_S$. We consider two cases depending on the length of string s. If $\ell = |s| \leq k$, then for $w = ua^\ell$ we have that $P'(s) = P'(w)$ and $G^a_{k\text{-}SSO}$ does not go through a secret state by generating $\delta'(q_0, w)$. Indeed, $q^\star_0, \ldots, q^\star_k$ are non-secret by the construction of $G^a_{k\text{-}SSO}$ and q_0 is non-secret by Lemma 5.24, current-state opacity of G^a_{CSO}, and by the fact that $\delta(q_0, P^{-1}(\varepsilon)) \cap Q_{NS} \neq \emptyset$. If $\ell = |s| > k$, then by current-state opacity of G^a_{CSO} there is $v \in L(G^a_{CSO})$ such that $\delta(q_0, v) \in Q_{NS}$ and $|P(v)| = |P'(s)| - k$. By Lemma 5.24, we have that $\delta(q_0, v') \in Q_{NS}$ for every prefix $v' \in \overline{v}$ with $P(v) = P(v')$. In $G^a_{k\text{-}SSO}$, the sequence v can be extended by ua^k such that $P'(s) = P'(vua^k)$. Therefore, the string $w = vua^k$ is such that $P'(s) = P'(w)$ and $G^a_{k\text{-}SSO}$ does not go through a secret state in last k observable steps by generating w. Altogether, $G^a_{k\text{-}SSO}$ is strongly k-step opaque. □

In Theorem 4.8 we showed that the problem of deciding current-state opacity of a DES with a single observable event is coNP-complete. Transformation 5.25 allows us to generalize the hardness part of this result to strong k-step opacity. However, the transformation is linear with respect to the parameter k, and therefore we consider k to be encoded in unary in the following corollary.

Corollary 5.29. *Given a natural number k represented in unary and a DES G with a single observable event. The problem of deciding whether the system G satisfies strong k-step opacity is* coNP*-hard.*

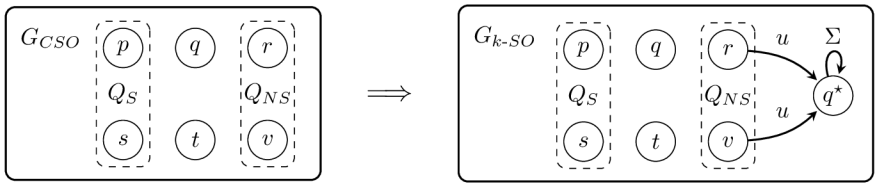


Figure 5.12: Transforming current-state opacity to weak k-step opacity, for an arbitrary parameter $k \in \mathbb{N}_\infty$.

5.5 CSO to k-SO

In this section, we describe the general transformation from current-state opacity to weak k-step opacity that uses neutral states to preserve the number of observable events without the help of Transformation 4.3. Notably, unlike the transformations discussed in the previous section, Transformation 5.30 is applicable to systems that have both neutral states and a single observable event, and the resulting system will still have a single observable event.

Let the current-state opacity problem be represented by a DES G_{CSO}. We transform it to a DES $G_{k\text{-}SO}$ in such a way that G_{CSO} is current-state opaque if and only if $G_{k\text{-}SO}$ is weakly k-step opaque.

Transformation 5.30. Let $G_{CSO} = (Q, \Sigma, \delta, I)$ be a DES with the secret states Q_S, the non-secret states Q_{NS}, and the corresponding projection $P\colon \Sigma^* \to \Sigma_o^*$. We construct a DES

$$G_{k\text{-}SO} = (Q \cup \{q^\star\}, \Sigma \cup \{u\}, \delta', I)$$

where u is a new unobservable event and $q^\star$ is a new neutral state. The transition function δ' is initialized as the transition function δ of the system G_{CSO} and further extended as follows, see Figure 5.12 for an illustration:

1. for each state $q \in Q_{NS}$, we add a transition $(q, u, q^\star)$ to δ';
2. for each $a \in \Sigma$, we add a self-loop $(q^\star, a, q^\star)$ to δ'.

We extend the projection P to the projection $P'\colon (\Sigma \cup \{u\})^* \to \Sigma_o^*$. The sets Q_S and Q_{NS} remain unchanged. ◇

Notice that Transformation 5.30 can be done in polynomial time and that it preserves determinism and the number of observable events. It is also independent of the parameter k, and hence it works for any parameter $k \in \mathbb{N}_\infty$ without affecting the size of the resulting system $G_{k\text{-}SO}$.

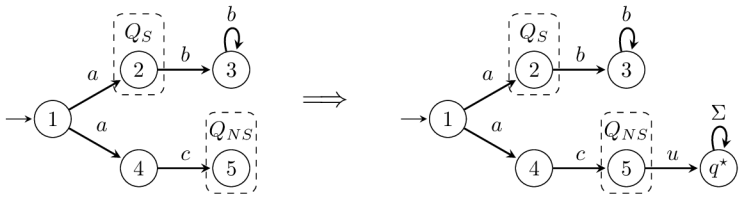


Figure 5.13: An example of the transformation of the CSO problem (left) to the k-SO problem (right).

Example 5.31. Let G over $\Sigma = \{a, b, c\}$ depicted in Figure 5.13 (left) be an instance of the CSO problem from Example 3.6 with the set of secret states $Q_S = \{2\}$ and the set of non-secret states $Q_{NS} = \{5\}$. Transformation 5.30 of CSO to k-SO then results in the DES G' depicted in Figure 5.13 (right) with a new neutral state $q^\star$ and a new unobservable event u. We distinguish two cases depending on whether event c is observable or not.

If event c is unobservable, then G is current-state opaque, because the only string leading to the secret state, state 2, is the string a, for which the string ac leading to the non-secret state, state 5, satisfies that $P(a) = P(ac)$. Then, the reader can see that G' is weakly k-step opaque, for any $k \in \mathbb{N}_\infty$, because the only possible extensions of the string a from the secret state 2 are of the form b^i, for $i \in \mathbb{N}$, and for every such extension there is an extension ub^i of the string ac from the non-secret state 5 such that $P'(ab^i) = P'(acub^i)$.

If event c is observable, then G is not current-state opaque, because the only string leading to a non-secret state, string ac, has a different observation than the string a leading to the secret state, that is, $P(ac) \neq P(a)$. Consequently, the reader can verify that G' is not current-state opaque, and hence neither weakly k-step opaque, for any parameter $k \in \mathbb{N}_\infty$. $\diamond$

The following theorem justifies the correctness of Transformation 5.30.

Theorem 5.32. *A DES G_{CSO} is current-state opaque with respect to Q_S, Q_{NS}, and P if and only if the DES $G_{k\text{-}SO}$ obtained by Transformation 5.30 is weakly k-step opaque, for any parameter $k \in \mathbb{N}_\infty$, with respect to Q_S, Q_{NS}, and P'.*

Proof. Assume first that G_{CSO} is not current-state opaque. Since the new state $q^\star$ is neither secret nor non-secret, we have that $G_{k\text{-}SO}$ is not current-state opaque either. Therefore, $G_{k\text{-}SO}$ is not weakly k-step opaque, for any parameter $k \in \mathbb{N}_\infty$.

On the other hand, assume that G_{CSO} is current-state opaque. Since the new state $q^\star$ is neither secret nor non-secret, we have that $G_{k\text{-}SO}$ is current-state opaque as well. Let $st \in L(G_{k\text{-}SO})$ be such that $\delta'(\delta'(I,s) \cap Q_S, t) \neq \emptyset$; in particular, $\delta'(I,s) \cap Q_S \neq \emptyset$. Then, since $G_{k\text{-}SO}$ is current-state opaque, there exists $s' \in L(G_{k\text{-}SO})$ such that $P'(s') = P'(s)$ and $\delta'(I,s') \cap Q_{NS} \neq \emptyset$. By construction, s' can be extended by the string ut using the transitions to state $q^\star$ followed by self-loops in state $q^\star$. Therefore, $\delta'(\delta'(I,s') \cap Q_{NS}, ut) \neq \emptyset$ and $P'(st) = P'(s'ut)$, which shows that $G_{k\text{-}SO}$ is weakly k-step opaque, for any parameter $k \in \mathbb{N}_\infty$. □

In Theorem 4.8, we showed that the problem of deciding current-state opacity of a DES with a single observable event is coNP-complete. Transformation 5.30 allows us to generalize the hardness part of this result to weak k-step opacity. Unlike strong k-step opacity, the weak notion remains coNP-hard even for instances with the parameter $k \geq |Q|$, and therefore we can consider k to encoded in binary in the following corollary.

Corollary 5.33. *Given a natural number k represented by $O(\log(k))$ bits and a DES G with a single observable event. The problem of deciding whether the system G satisfies weak k-step opacity is* coNP*-hard.*

5.6 k-SO to CSO

In this section, we discuss the transformations from weak k-step opacity to current-state opacity. The general transformation takes place in four steps, each of which is described in a separate subsection. Initially, we show how to transform weak ∞-step opacity to current-state opacity in Subsection 5.6.1. The construction of a k-step counter automaton of size polynomial in the logarithm of k is described in Subsection 5.6.2. The general transformation from weak k-step opacity to current-state opacity for systems that allow neutral states is presented in Subsection 5.6.3. In Subsection 5.6.4, we further modify the previous transformation so that the resulting system does not use neutral states. Since the general transformation relies on binary encoding of observable events by Transformation 4.3, we provide separate transformations for systems with a single observable event in Subsections 5.6.5 and 5.6.6. Again, we distinquish two cases depending on the presence of neutral states in the system.

5.6.1 ∞-SO to CSO

Let the weak ∞-step opacity problem be represented by a DES $G_{\infty\text{-}SO}$. We transform it to a DES G_{CSO} in such a way that $G_{\infty\text{-}SO}$ is weakly ∞-step opaque if and only if G_{CSO} is current-state opaque. Our transformation proceeds in two steps:

1. We construct a DES G_{CSO} with one additional observable event @ using Transformation 5.34.
2. We use Transformation 4.3 to reduce the number of observable events of G_{CSO} by one.

Since the second step follows from Transformation 4.3, we only describe the first step, that is, the construction of G_{CSO} over $\Sigma \cup \{@\}$.

Transformation 5.34. Let $G_{\infty\text{-}SO} = (Q, \Sigma, \delta, I)$ be a DES with the secret states Q_S, the non-secret states Q_{NS}, and the corresponding projection $P\colon \Sigma^* \to \Sigma_o^*$. We construct a DES

$$G_{CSO} = (Q \cup Q^+ \cup Q^-, \Sigma \cup \{@\}, \delta', I)$$

by creating two disjoint copies of the system $G_{\infty\text{-}SO}$, denoted by G^+ and G^-, with disjoint state sets $Q^+ = \{q^+ \mid q \in Q\}$ and $Q^- = \{q^- \mid q \in Q\}$, and with an additional observable event @ that connects the system $G_{\infty\text{-}SO}$ to the copies G^+ and G^- by transitions $(p, @, p^+)$, for every secret state $p \in Q_S$, and $(q, @, q^-)$, for every non-secret state $q \in Q_{NS}$, see Figure 5.14.

We define the projection $P'\colon (\Sigma \cup \{@\})^* \to (\Sigma_o \cup \{@\})^*$, and the sets of secret states $Q'_S = Q^+$ and of non-secret states $Q'_{NS} = Q \cup Q^-$. ◇

Notice that Transformation 5.34 can be done in polynomial time using no neutral states and that it preserves determinism.

Example 5.35. Let G over $\Sigma = \{a, b, c\}$ depicted in Figure 5.15 (left) be an instance of the weak ∞-step opacity problem from Example 3.9 with the set of secret states $Q_S = \{2\}$ and the set of non-secret states $Q_{NS} = \{4\}$. Transformation 5.34 of ∞-SO to CSO then results in the DES G' depicted in Figure 5.15 (right) with a new observable event @, the set of secret states Q'_S, and the set of non-secret states Q'_{NS}. We again consider two cases based on the observability status of event c.

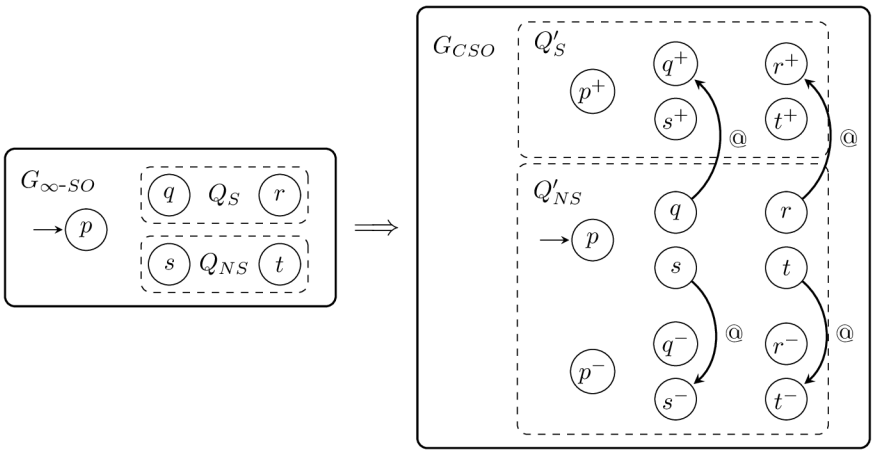


Figure 5.14: Transforming weak ∞-step opacity to current-state opacity.

If event c is unobservable, then G is weakly ∞-step opaque. Indeed, the only string leading to the single secret state, state 2, is the string a. The same string leads to the single non-secret state, state 4. Then, any possible extension of the string a from the secret state 2 is the string b^i, for $i \in \mathbb{N}$, which reaches state 3. However, for any such extension, there is an extension cb^i from the non-secret state 4 with $P(ab^i) = P(acb^i)$. The reader can further see that G' is current-state opaque, because it can enter a secret state only after generating a string of the form $a@b^i$, $i \in \mathbb{N}$, in which case $\delta'(1, P'^{-1}(a@)) = \{2^+, 4^-, 5^-\}$ and $\delta'(1, P'^{-1}(a@b^i)) = \{3^+, 5^-\}$ for $i \geq 1$, where states 4^- and 5^- are non-secret.

If event c is observable, then G is not weakly ∞-step opaque, because after generating string ab, the intruder can deduce that the system was in the secret state 2 one step ago. Similarly, after observing string $a@b \in P'(L(G'))$, the intruder knows that G' is in the secret state 3^+, and hence the system G' is not current-state opaque. $\diamond$

The following theorem justifies the correctness of Transformation 5.34.

Theorem 5.36. *A DES $G_{\infty\text{-}SO}$ is weakly ∞-step opaque with respect to Q_S, Q_{NS}, and P if and only if the DES G_{CSO} obtained by Transformation 5.34 is current-state opaque with respect to Q'_S, Q'_{NS}, and P'.*

Proof. Assume that the system $G_{\infty\text{-}SO}$ is weakly ∞-step opaque. To show that the system G_{CSO} is current-state opaque, we consider a string w such

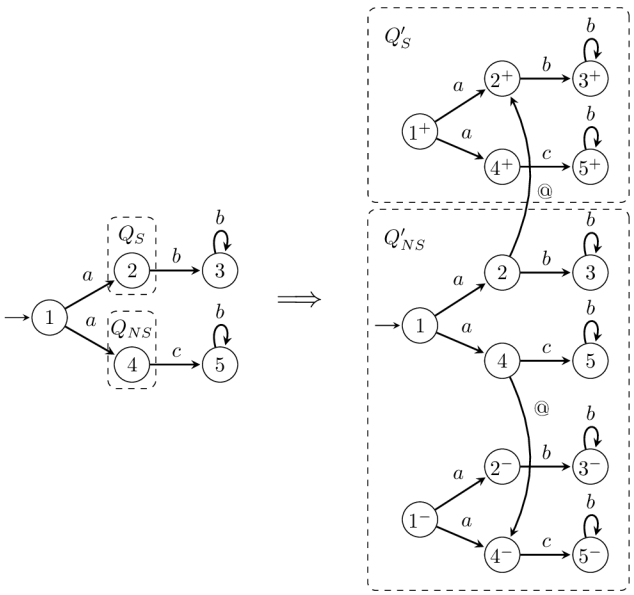


Figure 5.15: An example of the transformation of the ∞-SO problem (left) to the CSO problem (right).

that $\delta'(I, w) \cap Q'_S \neq \emptyset$, and show that there is a string w' such that $P'(w) = P'(w')$ and $\delta'(I, w') \cap Q'_{NS} \neq \emptyset$. Since the set of secret states is $Q'_S = Q^+$, the string w must be of the form $w_1@w_2$. By construction, there exists a secret state $q \in \delta(I, w_1) \cap Q_S$ in $G_{\infty\text{-}SO}$ such that the state $q^+ \in \delta'(I, w_1@) \cap Q'_S$ in the system G_{CSO}, and the string w_2 is generated from the state q^+. Therefore, we can generate the string w_2 from the state q in $G_{\infty\text{-}SO}$, that is, $\delta(\delta(I, w_1) \cap Q_S, w_2) \neq \emptyset$, and hence weak ∞-step opacity of $G_{\infty\text{-}SO}$ implies that there is a string $w'_1 w'_2 \in L(G_{\infty\text{-}SO})$ such that $P(w_1) = P(w'_1)$, $P(w_2) = P(w'_2)$, and $\delta(\delta(I, w'_1) \cap Q_{NS}, w'_2) \neq \emptyset$. If we define the string $w' = w'_1@w'_2$, then we have that $P'(w) = P'(w')$ and we obtain that $\emptyset \neq \delta'(\delta'(I, w'_1@) \cap Q'_{NS}, w'_2) \subseteq Q'_{NS}$, which completes this part of the proof.

If the system $G_{\infty\text{-}SO}$ is not weakly ∞-step opaque, then there exists a string $st \in L(G_{\infty\text{-}SO})$ such that $\delta(\delta(I, s) \cap Q_S, t) \neq \emptyset$ and $\delta(\delta(I, s') \cap Q_{NS}, t') = \emptyset$ for every string $s't' \in L(G_{\infty\text{-}SO})$ with $P(s) = P(s')$ and $P(t) = P(t')$. Taking the string $s@t \in L(G_{CSO})$, we obtain that $\emptyset \neq \delta'(\delta'(I, s@) \cap Q'_S, t) = \delta'(I, s@t) \subseteq Q'_S$ and, for every string $s'@t' \in L(G_{CSO})$ with $P'(s@t) = P'(s'@t')$, we have that $\delta'(I, s'@t') \cap Q'_{NS} = \delta'(\delta'(I, s'@) \cap Q'_{NS}, t') = \emptyset$, and hence the system G_{CSO} is not current-state opaque. $\square$

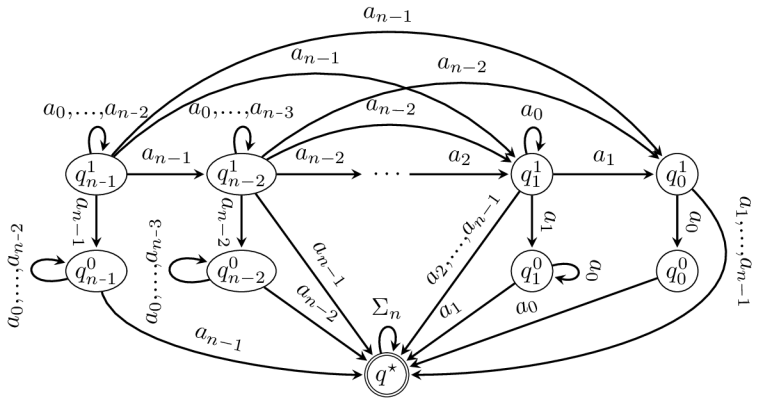


Figure 5.16: The NFA $\mathcal{A}_k$ of Theorem 5.38.

We now apply our transformations to solve the open problem concerning the complexity of deciding weak ∞-step opacity. Transformation 5.34 allows us to transform an instance of weak ∞-step opacity decision problem to a current-state opacity decision problem that can be solved in polynomial space. Combined with the PSPACE-hardness of weak ∞-step opacity from Corollary 5.23, we can generalize Theorem 4.7 for weak ∞-step opacity.

Corollary 5.37. *The problem of deciding whether a DES satisfies weak ∞-step opacity is* PSPACE*-complete. The problem remains* PSPACE*-complete even if the system is a DFA with three events, one of which is unobservable.*

5.6.2 k-step counter

Before proceeding to the general transformation for weak k-step opacity, we define an automaton to serve as a k-step counter. Informally, we construct an NFA $\mathcal{A}_k$ of size polynomial in the logarithm of k such that the observer of the automaton $\mathcal{A}_k$ has a unique path of length k consisting solely of non-marked states, while all the other states are marked. This path plays the role of a k-step counter, which is essential in the following transformations.

Theorem 5.38. *For every integer $k \geq 1$, there is an NFA $\mathcal{A}_k$ with $n = \lceil \log_2(k+1) \rceil$ events and $2n+1$ states, such that the automaton $\mathcal{A}_k$ marks all strings except for the unique string W_k of length k and all its prefixes.*

Proof. Let $k \geq 1$ be given, and let $n = \lceil \log_2(k+1) \rceil$. We recursively define

the string Z_n over the alphabet $\Sigma_n = \{a_0, a_1, \ldots, a_{n-1}\}$ as follows:

$$Z_1 = a_0 \quad \text{and} \quad Z_i = Z_{i-1}a_{i-1}Z_{i-1} \text{ for } 1 < i \leq n\,.$$

For example, $Z_3 = Z_2a_2Z_2 = Z_1a_1Z_1a_2Z_1a_1Z_1 = a_0a_1a_0a_2a_0a_1a_0$. Such strings are known in the literature as *Zimin words*, and it is well-known that the string Z_n is of length $2^n - 1$ [45]. We denote the suffix of the string Z_n of length k by W_k. It is also known that the event on the ℓth position of the string Z_n is a_j, where j is the number of trailing zeros in the binary representation of ℓ [46]. Since the string Z_n is a palindrome, the same event appears on positions ℓ and $2^n - 1 - \ell$. For instance, since 2 is encoded as 10 in binary, the event at the second positions from both sides of the string Z_3 is a_1.

Let $b_{n-1}b_{n-2}\cdots b_0$ be the binary representation of k, that is, $k = b_{n-1} \cdot 2^{n-1} + b_{n-2} \cdot 2^{n-2} + \cdots + b_0 \cdot 2^0$, where the leftmost bit is the most significant bit; in particular, we have that $b_{n-1} = 1$. We construct the NFA

$$\mathcal{A}_k = (Q, \Sigma_n, \delta, I, F)$$

where the set of states $Q = \{q^\star\} \cup \{q_i^1, q_i^0 \mid i = 0, \ldots, n-1\}$ consists of the state $q^\star$ and of two states q_i^1 and q_i^0 for every bit b_i of the binary representation of k; the state $q^\star$ is the only marked state, that is, $F = \{q^\star\}$; and the transition function δ is defined as follows, see Figure 5.16 for an illustration:

1. For every event $a \in \Sigma_n$, the self-loop $(q^\star, a, q^\star) \in \delta$;
2. For every state q_i^1,
   (a) the transition $(q_i^1, a_i, q_i^0) \in \delta$;
   (b) the self-loop $(q_i^1, a_j, q_i^1) \in \delta$, for $0 \leq j \leq i-1$;
   (c) the transition $(q_i^1, a_i, q_j^1) \in \delta$, for $0 \leq j \leq i-1$;
   (d) the transition $(q_i^1, a_j, q^\star) \in \delta$, for $i+1 \leq j \leq n-1$;
3. For every state q_i^0,
   (a) the transition $(q_i^0, a_i, q^\star) \in \delta$;
   (b) the self-loop $(q_i^0, a_j, q_i^0) \in \delta$, for $0 \leq j \leq i-1$;
   (c) the other transitions are undefined.

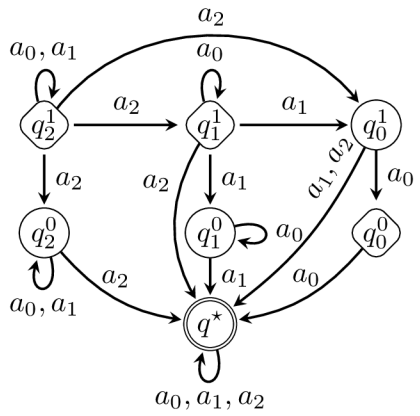


Figure 5.17: The NFA $\mathcal{A}_6$, where the initial states are diamond-shaped.

Finally, the set of initial states is defined as the set

$$I = \{q_{n-1}^{b_{n-1}}, q_{n-2}^{b_{n-2}}, \ldots, q_0^{b_0}\}$$

corresponding to the states encoding k in binary.

Before we show that the automaton $\mathcal{A}_k$ marks all strings over the alphabet Σ_n other than the prefixes of the string W_k, we illustrate the construction. We consider $k = 6$, for which $n = 3$ and the binary encoding of 6 is 110. Since the string $Z_3 = a_0a_1a_0a_2a_0a_1a_0$, its suffix of length 6 is the string $W_6 = a_1a_0a_2a_0a_1a_0$. The automaton $\mathcal{A}_6$ is depicted in Figure 5.17, where the initial states are q_2^1, q_1^1, and q_0^0 corresponding to the bits of 110. For the computation of the automaton $\mathcal{A}_6$ on the string $W_6 = a_1a_0a_2a_0a_1a_0$, see the observer of the automaton $\mathcal{A}_6$ depicted in Figure 5.18. It is clear from the observer that the automaton $\mathcal{A}_6$ does not mark any prefix of the string $W_6 = a_1a_0a_2a_0a_1a_0$, and that it marks all strings different from the string W_6.

It remains to show that the automaton $\mathcal{A}_k$ marks all strings except for the prefixes of the string W_k. We first show that the automaton $\mathcal{A}_k$ does not mark any prefix of the string W_k, and then we show that the automaton $\mathcal{A}_k$ marks all strings that do not form a prefix of the string W_k. To show that the automaton $\mathcal{A}_k$ does not mark any prefix of the string W_k, we prove the following lemma.

Lemma 5.39. *The observer of the automaton $\mathcal{A}_{2^n-1}$ having generated the prefix of the string Z_n of length $\ell \leq 2^n-1$ is in the state $\{q_{n-1}^{r_{n-1}}, q_{n-2}^{r_{n-2}}, \ldots, q_0^{r_0}\}$, where $r_{n-1}r_{n-2}\cdots r_0$ is the number $2^n - 1 - \ell$ in binary.*

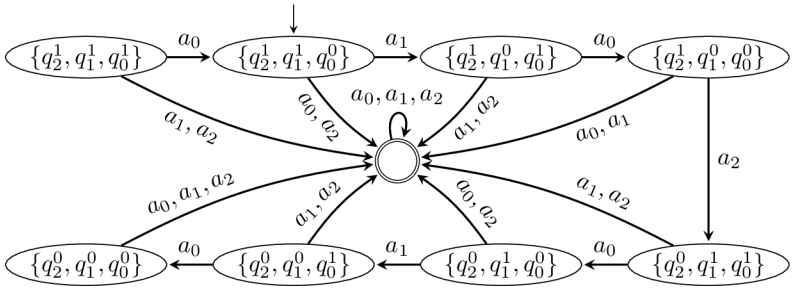


Figure 5.18: The minimized observer of the NFA $\mathcal{A}_6$ showing the behavior of the NFA $\mathcal{A}_6$ on the strings Z_3 and W_6. The initial state of the automaton $\mathcal{A}_6$ is denoted by the little arrow from above.

Proof of Lemma 5.39. We prove the claim by induction on ℓ. There is nothing to prove for $\ell = 0$, since $2^n - 1$ is $11 \cdots 1$ in binary, corresponding to the initial state $\{q_{n-1}^1, q_{n-2}^1, \ldots, q_0^1\}$ of the observer of the automaton $\mathcal{A}_{2^n-1}$. We now assume that the claim holds for $\ell \geq 1$, and we show that it holds for $\ell + 1$. By induction, we have that the observer of the NFA $\mathcal{A}_{2^n-1}$ having generated the prefix of the string Z_n of length $\ell < 2^n - 1$ is in the state $\{q_{n-1}^{r_{n-1}}, q_{n-2}^{r_{n-2}}, \ldots, q_0^{r_0}\}$, where $r_{n-1}r_{n-2}\cdots r_0$ is $2^n-1-\ell$ in binary. Let r_t be the rightmost non-zero bit of $r_{n-1}r_{n-2}\cdots r_0$. Then, there are t trailing zeros, and hence the event of the string Z_n at position $\ell+1$ is a_t. By the definition of the NFA $\mathcal{A}_{2^n-1}$, the transition under a_t is undefined in states $q_{t-1}^0, \ldots, q_0^0$, and it is a self-loop in states $q_{n-1}^{r_{n-1}}, \ldots, q_{t+1}^{r_{t+1}}$. The transitions from the state q_t^1 under the event a_t lead to states $q_{t-1}^1, \ldots, q_0^1$ and to the state q_t^0. Thus, generating the event a_t, the observer of $\mathcal{A}_{2^n-1}$ moves from the state $\{q_{n-1}^{r_{n-1}}, q_{n-2}^{r_{n-2}}, \ldots, q_0^{r_0}\}$ to the state $\{q_{n-1}^{r_{n-1}}, q_{n-2}^{r_{n-2}}, \ldots, q_{t+1}^{r_{t+1}}, q_t^0, q_{t-1}^1, \ldots, q_0^1\}$, which is a non-marked state binary representing the number $2^n - 1 - \ell - 1 = 2^n - 1 - (\ell + 1)$, completing the proof of Lemma 5.39. □

In particular, Lemma 5.39 implies that the NFA $\mathcal{A}_k$ corresponds to the automaton $\mathcal{A}_{2^n-1}$ having generated the prefix of the string Z_n of length $2^n - 1 - k = |Z_n| - |W_k|$ (recall that W_k is the suffix of Z_n of length k), since, in this case, the automaton $\mathcal{A}_{2^n-1}$ is in the states encoding the number $2^n - 1 - (2^n - 1 - k) = k$. Consequently, the observer of the automaton $\mathcal{A}_k$ generating the string W_k event by event goes through the respective states representing the numbers k, $k - 1, \ldots, 0$ in binary, which are not marked, and therefore the automaton $\mathcal{A}_k$ does not mark any prefix of the string W_k.

Finally, to show that the automaton $\mathcal{A}_k$ marks all strings that do not form a prefix of the string W_k, assume that the observer of the automaton $\mathcal{A}_k$ is in a state of the form $\{q_{n-1}^{r_{n-1}}, q_{n-2}^{r_{n-2}}, \ldots, q_0^{r_0}\}$ reached by a prefix w of the string W_k. Then, either $W_k = waw'$ or $W_k = w$. In the former case, if the automaton $\mathcal{A}_k$ generates an event different from a, it reaches the marked state $q^\star$, while in the latter case, generating any event reaches the state $q^\star$. In both cases, the state $q^\star$ then appears in every state of the observer of the automaton $\mathcal{A}_k$ from now on, which makes the state of the observer marked. □

5.6.3 The general case with neutral states

Even though the DES G_{CSO} that results from Transformation 5.34 applied to a system $G_{\infty\text{-}SO}$ can verify weak ∞-step opacity of the system $G_{\infty\text{-}SO}$ by checking current-state opacity of the system G_{CSO}, it is not suitable to verify weak k-step opacity of the system $G_{\infty\text{-}SO}$; indeed, the system G_{CSO} verifies any number of steps from the visited secret state rather than at most k steps. To overcome this issue, we extend Transformation 5.34 by adding a counter that allows us to count up to k observable events from a visited secret state.

However, we cannot simply add k states to model the counter, because adding k states requires k steps in the transformation, which is exponential in the size (the number of bits) of the binary representation of k. Instead, we model the counter with the help of the automaton $\mathcal{A}_k$ from Theorem 5.38 that can be constructed in time $O(\log^2(k))$.

Let the weak k-step opacity problem be represented by a DES $G_{k\text{-}SO}$. We transform it to a DES G_{CSO} in such a way that $G_{k\text{-}SO}$ is weakly k-step opaque if and only if G_{CSO} is current-state opaque.

Transformation 5.40. Let $G_{k\text{-}SO} = (Q, \Sigma, \delta, I)$ be a DES with the secret states Q_S, the non-secret states Q_{NS}, the corresponding projection $P\colon \Sigma^* \to \Sigma_o^*$, and the parameter $k \in \mathbb{N}$. We construct a DES

$$G_{CSO} = (Q', \Sigma', \delta', I)$$

consisting of the original system $G_{k\text{-}SO}$ along with its two modified copies and a k-step counter automaton. In more detail, we consider:

- two disjoint copies G^+ and G^- of the system $G_{k\text{-}SO}$, as in Transformation 5.34, with disjoint state sets $Q^+ = \{q^+ \mid q \in Q\}$ and $Q^- = \{q^- \mid q \in Q\}$, respectively, and

- the k-step counter automaton $\mathcal{A}_k$ constructed in Theorem 5.38.

By construction, $\mathcal{A}_k$ is of size polynomial in the logarithm of k, and its observer has a unique path of length k consisting solely of non-marked states, while all the other states are marked. However, before we connect the automata $G_{k\text{-}SO}$, G^+, G^-, and $\mathcal{A}_k$ together, we note that the automata $G_{k\text{-}SO}$, G^+, and G^- are over the alphabet Σ, while the automaton $\mathcal{A}_k$ is over the alphabet Σ_n, which is disjoint from Σ. Therefore, we change the alphabets of the automata to

$$\tilde{\Sigma} = \Sigma \cup (\Sigma_o \times \Sigma_n)\,.$$

Namely, in G^+ and G^-, we replace every *observable* transition (p, α, q) by $|\Sigma_n|$ transitions $(p, (\alpha, \beta), q)$, for every event $\beta \in \Sigma_n$, and we denote the results by $\tilde{G}^+$ and $\tilde{G}^-$. Similarly, in the automaton $\mathcal{A}_k$, we replace every transition (p, β, q) by $|\Sigma_o|$ transitions $(p, (\alpha, \beta), q)$, for every observable event $\alpha \in \Sigma_o$, and we denote the result by $\tilde{\mathcal{A}}_k$.

Now, we construct a DES

$$G_{CSO} \text{ as a disjoint union of the automata } G_{k\text{-}SO},\ \tilde{G}^+,\ \tilde{G}^-, \text{ and } \tilde{\mathcal{A}}_k,$$

over alphabet $\Sigma' = \tilde{\Sigma} \cup \{@\}$. We connect the parts of G_{CSO} with the transitions $(p, @, p^+)$ and $(p, @, q_0)$, for every secret state $p \in Q_S$ and every initial state $q_0 \in I$ of $\tilde{\mathcal{A}}_k$, and the transitions $(q, @, q^-)$, for every non-secret state $q \in Q_{NS}$, cf. Figure 5.19.

We define the projection $P' \colon (\tilde{\Sigma} \cup \{@\})^* \to (\Sigma_o \cup \{@\} \cup \Sigma_o \times \Sigma_n)^*$, and the sets of secret states $Q'_S = Q^+$ and of non-secret states $Q'_{NS} = Q^- \cup \{q^\star\}$, where $q^\star$ is the unique marked state of $\tilde{\mathcal{A}}_k$. The other states are neutral. $\diamond$

Notice that Transformation 5.40 can be done in polynomial time in the size of the system and in the number of bits of the binary representation of k.

In G_{CSO}, every event after generating the event @ is either unobservable or pair of events of $\Sigma_o \times \Sigma_n$. Therefore, in the sequel we denote strings over $\Sigma_{uo} \cup \Sigma_o \times \Sigma_n$, such as $s = u(a, x)u(b, y)$, simply as a pair of the form $\Sigma^* \times \Sigma_n^*$ of concatenated strings of the corresponding alphabets, such as $s = (uaub, xy)$, where $u \in \Sigma_{uo}$, $a, b \in \Sigma_o$, and $x, y \in \Sigma_n$.

Example 5.41. Let $G = (\{1, \ldots, 8\}, \{a\}, \delta, \{1, 2\})$ in Figure 5.20(a) be an instance of the weak 6-step opacity problem with a single secret state 1 and a single non-secret state 2. We distinguish two cases depending on whether state 8 is reachable or not.

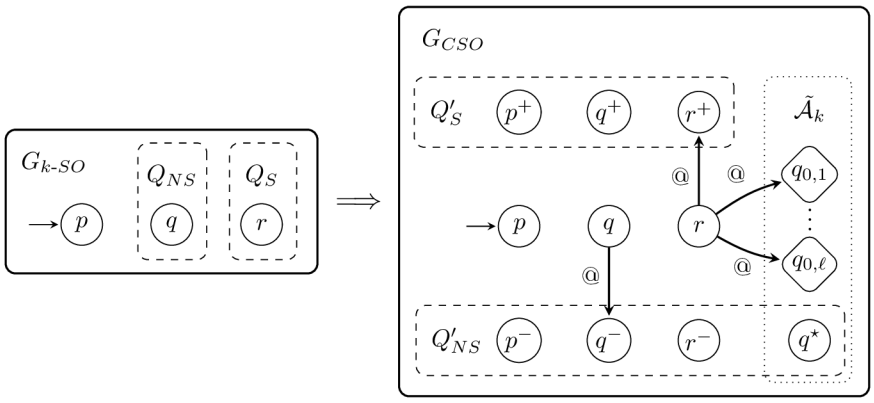


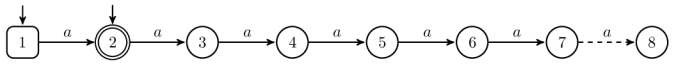
Figure 5.19: Transforming weak k-step opacity to current-state opacity with neutral states; the initial states of $\tilde{\mathcal{A}}_k$ are diamond-shaped.

If state 8 is reachable, then G is weakly 6-step opaque, since we can make six observable steps from both states 1 and 2. To encode $k = 6$, Transformation 5.40 of 6-SO to CSO uses the automaton $\mathcal{A}_6$ (see Theorem 5.38 and Figure 5.17), and results in the system G' of Figure 5.20(b), where the non-secret states are marked. The minimized observer of the system G' is shown in Figure 5.20(c). Since every state of the observer that is reachable by a string containing @ is marked, it contains a non-secret state of the system G', and hence the system G' is current-state opaque.

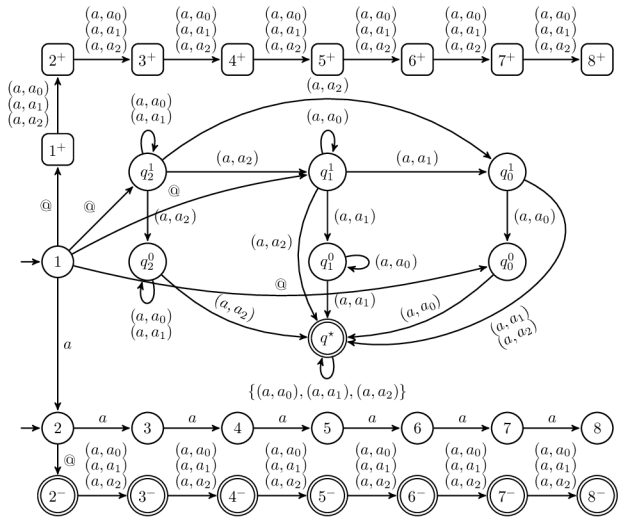
If we remove state 8 and the corresponding transition from the system G, then G is not weakly 6-step opaque, since we can make six observable steps from the secret state 1, but only five steps from the corresponding non-secret state 2. The transformation results in the system G' coinciding with the automaton of Figure 5.20(b) without the states 8, 8^+, 8^-, and the corresponding transitions. The minimized observer is shown in Figure 5.20(d), where the unique secret state (squared) denoting the state $\{7^+, q_2^0, q_1^0, q_0^0\}$ is reachable by the string $@(a,a_1)(a,a_0)(a,a_2)(a,a_0)(a,a_1)(a,a_0)$, that is, the system G' is not current-state opaque. ◇

The following theorem justifies the correctness of Transformation 5.40.

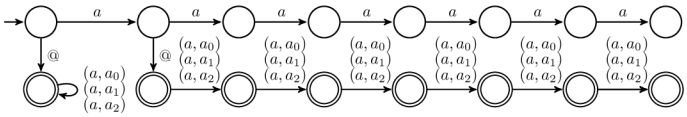
Theorem 5.42. *A DES $G_{k\text{-}SO}$ is weakly k-step opaque with respect to Q_S, Q_{NS}, and P if and only if the DES G_{CSO} obtained by Transformation 5.40 is current-state opaque with respect to Q'_S, Q'_{NS}, and P'.*



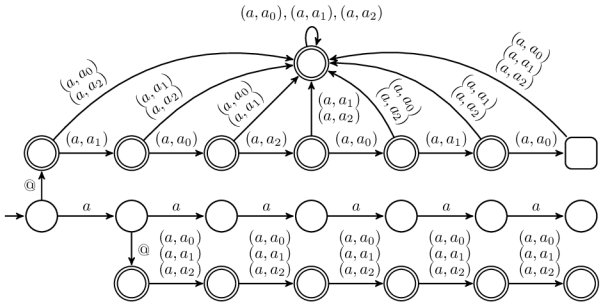
(a) The DES G, which is an instace of 6-SO.



(b) The DES G', which is an instance CSO.



(c) The minimized observer of the system G'.



(d) The minimized observer of the system G' without the states 8, 8^+, 8^-.

Figure 5.20: An example of the transformation of 6-SO to CSO with neutral states; the secret states are squared and the non-secret states are marked.

Proof. Assume that the system $G_{k\text{-}SO}$ is weakly k-step opaque. We show that the system G_{CSO} is current-state opaque. To this end, we consider a string $w \in L(G_{CSO})$ such that $\delta'(I, w) \cap Q'_S \neq \emptyset$, and we show that there is a string $w' \in P'^{-1}P'(w)$ such that $\delta'(I, w') \cap Q'_{NS} \neq \emptyset$. Since the set of secret states is $Q'_S = Q^+$, the string w must be of the form $w_1@(w_2, x)$ and, by construction, $\delta(I, w_1)$ contains a secret state of the system $G_{k\text{-}SO}$ from which the string w_2 can be generated.

If the string x is not the prefix of the unique string not marked by the automaton $\mathcal{A}_k$, then $\tilde{A}_k$ marks the string (w_2, x), and hence G_{CSO} reaches the non-secret state $q^\star$ for $w' = w$, that is, $q^\star \in \delta'(I, w_1@(w_2, x)) \cap Q'_{NS} \neq \emptyset$.

If $|P(w_2)| \leq k$, then weak k-step opacity of the system $G_{k\text{-}SO}$ implies the existence of a string $w'_1 w'_2 \in L(G_{k\text{-}SO})$ such that $P(w'_1) = P(w_1)$, $P(w'_2) = P(w_2)$, and $\delta(\delta(I, w'_1) \cap Q_{NS}, w'_2) \neq \emptyset$; that is, there exists a non-secret state $q \in \delta(I, w'_1)$ from which the string w'_2 can be generated, reaching a state r. Then, for the string $w' = w'_1@(w'_2, x)$, where the string x is the prefix of the unique string not marked by the automaton $\mathcal{A}_k$ of length $|P(w'_2)|$, we obtain that $\delta'(I, w') \cap Q'_{NS} \neq \emptyset$, since the non-secret state $r^- \in Q^-$ is reachable from the state q^- in the system G_{CSO} by the string (w'_2, x).

If $|P(w_2)| > k$, then every string $w'_1@(w'_2, x) \in \Sigma'^*$ is such that the string x is marked by the automaton $\mathcal{A}_k$, because the automaton $\mathcal{A}_k$ marks all strings longer than k, and hence the string (w'_2, x) is marked by the automaton $\tilde{\mathcal{A}}_k$, that is, $\delta'(I, w'_1@(w'_2, x)) \cap Q'_{NS} \neq \emptyset$. Altogether, the system G_{CSO} is current-state opaque.

On the other hand, assume that the system $G_{k\text{-}SO}$ is not weakly k-step opaque, that is, there exists a string $st \in L(G_{k\text{-}SO})$ such that $|P(t)| \leq k$, $\delta(\delta(I, s) \cap Q_S, t) \neq \emptyset$ and, for every string $s' \in P^{-1}P(s)$ and every string $t' \in P^{-1}P(t)$, we have that $\delta(\delta(I, s') \cap Q_{NS}, t') = \emptyset$. Then, in particular, $\delta'(I, s@) \cap Q'_S \neq \emptyset$.

If $\delta(I, s') \cap Q_{NS} = \emptyset$, then $\delta'(I, s'@) \cap Q'_{NS} = \emptyset$, and hence the system G_{CSO} is not current-state opaque.

If $\delta(I, s') \cap Q_{NS} = Z \neq \emptyset$, we consider any string $s'@(t', y) \in L(G_{CSO})$, where the string y is a prefix of the unique string not marked by the automaton $\mathcal{A}_k$, which exists because $|y| = |P(t')| \leq k$. Then, the strings (t, y) and (t', y) are not marked by the automaton $\tilde{\mathcal{A}}_k$, and hence $\delta'(I, s@(t, y)) \cap Q'_S \neq \emptyset$ and $\delta'(I, s'@(t', y)) \cap Q'_{NS} = \delta'([\delta'(I, s'@) \cap Q^-], (t', y)) = \delta'(Z^-, (t', y)) = \emptyset$, where the set $Z^- = \{z^- \mid z \in Z\}$. Note that the string (t', y) is not generated in the system G_{CSO} from a state of the set Z^-, since the string t' cannot be

generated in the system $G_{k\text{-}SO}$ from any state $z \in Z$. Hence, the system G_{CSO} is not current-state opaque. □

5.6.4 The general case without neutral states

Finally, we show how to transform weak k-step opacity to current-state opacity without employing neutral states by modifying Transformation 5.40.

Transformation 5.43. Let $G_{k\text{-}SO} = (Q, \Sigma, \delta, I)$ be a DES with the secret states Q_S, the non-secret states Q_{NS}, the corresponding projection $P\colon \Sigma^* \to \Sigma_o^*$, and the parameter $k \in \mathbb{N}$. We first perform Transformation 5.40 on $G_{k\text{-}SO}$ to obtain automata $\tilde{G}^+$, $\tilde{G}^-$, and $\tilde{\mathcal{A}}_k$ over alphabet $\tilde{\Sigma}$. Now, we make all the states of the automaton $\tilde{G}^+$ initial and marked, and synchronize the computations of the automata $\tilde{G}^+$ and $\tilde{\mathcal{A}}_k$ by their synchronous product $\tilde{G}^+ \| \tilde{\mathcal{A}}_k$. We construct a DES

$$G_{CSO} = (Q', \Sigma', \delta', I) \text{ as a disjoint union of } G_{k\text{-}SO},\ \tilde{G}^-, \text{ and } \tilde{G}^+ \| \tilde{\mathcal{A}}_k,$$

connected together by transitions under a new observable event @ as follows:

1. we add transition $(q, @, (q^+, q_0))$ to δ', for every secret state $q \in Q_S$ and every initial state (q^+, q_0) of $\tilde{G}^+ \| \tilde{\mathcal{A}}_k$, and
2. we add transition $(q, @, q^-)$ to δ', for every non-secret state $q \in Q_{NS}$.

We define the projection $P'\colon (\tilde{\Sigma} \cup \{@\})^* \to (\Sigma_o \cup \{@\} \cup \Sigma_o \times \Sigma_n)^*$. The secret states Q'_S of the system G_{CSO} are defined as the non-marked states of the system $\tilde{G}^+ \| \tilde{\mathcal{A}}_k$, that is, the states of $\tilde{G}^+ \| \tilde{\mathcal{A}}_k$ where second part is not equal to $q^\star$. All the other states are non-secret, that is, $Q'_{NS} = Q' - Q'_S$. ◇

Notice that Transformation 5.43 can be done in polynomial time in the size of the system and in the number of bits of the binary representation of the parameter k. Since this transformation does not preserve determinisn and the number of the observable events, we apply Transformations 4.3 and 4.5 on the resulting system G_{CSO} to reduce its number of observable events and to determinize it.

Example 5.44. We again consider weak 6-step opacity of the DES $G = (\{1, \ldots, 8\}, \{a\}, \delta, \{1, 2\})$ from Example 5.41 with a single secret state 1 and a single non-secret state 2, cf. Figure 5.20(a). Transformation 5.43 of 6-SO to

CSO uses the NFA $\mathcal{A}_6$ and results in the system G' depicted in Figure 5.21(a). We distinguish two cases depending on whether state 8 is reachable or not.

If state 8 is reachable, then G is weakly 6-step opaque, because we can make six observable steps from both states 1 and 2. The minimized observer of the system G' is depicted in Figure 5.21(b). Since every state of the observer is marked, it contains a non-secret state of the system G', that is, the system G' is current-state opaque.

If we remove state 8 from the system G, then the system G is not weakly 6-step opaque. The transformation results in the system G' that coincides with the NFA of Figure 5.21(a) without the states containing 8, 8^+, 8^-, and the corresponding transitions. The minimized observer is shown in Figure 5.21(c), where the secret state $\{(7^+, q_0^0), (7^+, q_1^0), (7^+, q_2^0))\}$ (single circled), containing the secret states of the automaton $\tilde{G}^+ \| \tilde{\mathcal{A}}_k$, is reachable by the string $@(a, a_1)(a, a_0)(a, a_2)(a, a_0)(a, a_1)(a, a_0)$, that is, the system G' is not current-state opaque. ◇

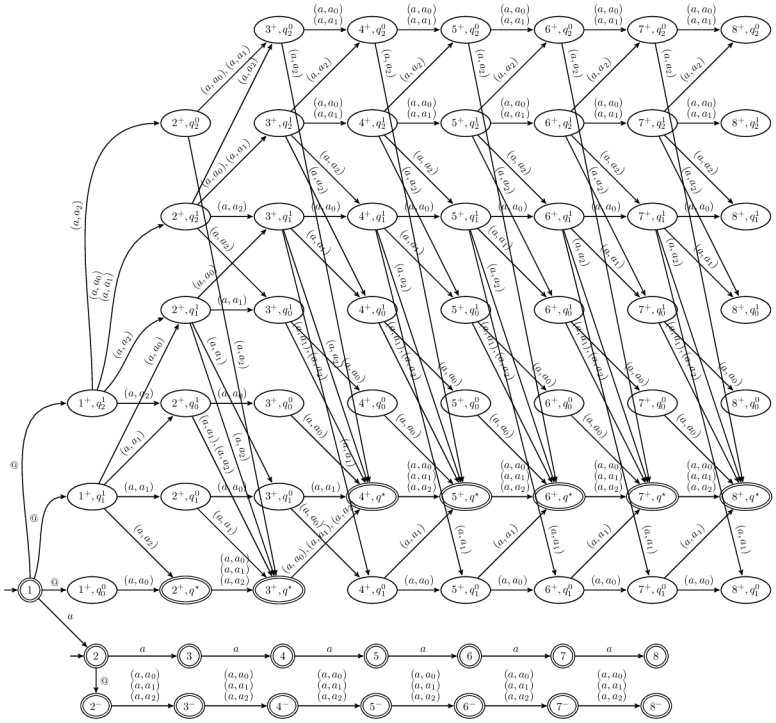
The following theorem justifies the correctness of Transformation 5.43.

Theorem 5.45. *A DES $G_{k\text{-}SO}$ is weakly k-step opaque with respect to Q_S, Q_{NS}, and P if and only if the DES G_{CSO} obtained by Transformation 5.43 is current-state opaque with respect to Q'_S, Q'_{NS}, and P'.*

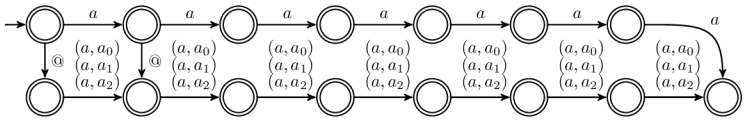
Proof. Assume that the system $G_{k\text{-}SO}$ is weakly k-step opaque. We show that the system G_{CSO} is current-state opaque. To this end, we consider a string $w \in L(G_{CSO})$ such that $\delta'(I, w) \cap Q'_S \neq \emptyset$, and show that there exists a string $w' \in P'^{-1}P'(w)$ such that $\delta'(I, w') \cap Q'_{NS} \neq \emptyset$. Since the set of secret states Q'_S consists of non-marked states of the automaton $\tilde{G}^+ \| \tilde{\mathcal{A}}_k$, the string w must be of the form $w_1@(w_2, x)$ and, by construction, $\delta(I, w_1)$ contains a secret state of the system $G_{k\text{-}SO}$, from which the string w_2 can be generated.

If $|P(w_2)| \leq k$, then weak k-step opacity of the system $G_{k\text{-}SO}$ implies the existence of a string $w'_1 w'_2 \in L(G_{k\text{-}SO})$ such that $P(w'_1) = P(w_1)$, $P(w'_2) = P(w_2)$, and $\delta(\delta(I, w'_1) \cap Q_{NS}, w'_2) \neq \emptyset$; in particular, there is a non-secret state $q \in \delta(I, w'_1)$ from which the string w'_2 can be generated, reaching a state r. Then, for the string $w' = w'_1@(w'_2, x)$, where the string x is the prefix of the unique string not marked by the automaton $\mathcal{A}_k$ of length $|P(w_2)|$, we obtain that $\delta'(I, w') \cap Q'_{NS} \neq \emptyset$, since the non-secret state $r^- \in Q^-$ is reachable from the state q^- in the system G_{CSO} by the string (w'_2, x).

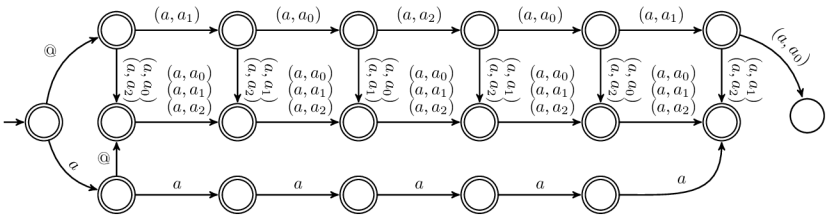
If $|P(w_2)| > k$ or x is not the prefix of the unique string not marked by the automaton $\mathcal{A}_k$, then, for every string $w'_1@(w'_2, x) \in L(G_{CSO})$, the string x is



(a) The DES G' with a relevant part of the automaton $\tilde{G}^{+} \| \tilde{\mathcal{A}}_6$.



(b) The minimized observer of the system G'.



(c) The minimized observer of the system G' without the states 8, 8^{+}, 8^{-}.

Figure 5.21: An example of the transformation of 6-SO to CSO without neutral states; non-secret states are marked, other states are secret.

marked by the automaton $\mathcal{A}_k$, and hence the string (w_2', x) is marked by the automaton $\tilde{G}^+ \| \tilde{\mathcal{A}}_k$ because $L_m(\tilde{G}^+ \| \tilde{\mathcal{A}}_k) = L_m(\tilde{G}^+) \parallel L_m(\tilde{\mathcal{A}}_k)$ and (w_2', x) belongs to both $L_m(\tilde{G}^+)$ and $L_m(\tilde{\mathcal{A}}_k)$. In particular, $\delta'(I, w_1@(w_2, x)) \cap Q'_{NS} \neq \emptyset$, and hence the system G_{CSO} is current-state opaque.

On the other hand, assume that the system $G_{k\text{-}SO}$ is not weakly k-step opaque, that is, there exists a string $st \in L(G_{k\text{-}SO})$ such that $|P(t)| \leq k$, $\delta(\delta(I, s) \cap Q_S, t) \neq \emptyset$ and, for every string $s' \in P^{-1}P(s)$ and every string $t' \in P^{-1}P(t)$, we have that $\delta(\delta(I, s') \cap Q_{NS}, t') = \emptyset$. Then, $\delta'(I, s@) \cap Q'_S \neq \emptyset$.

Now, if $\delta(I, s') \cap Q_{NS} = \emptyset$, then so is $\delta'(I, s'@) \cap Q'_{NS} = \emptyset$, and hence the system G_{CSO} is not current-state opaque.

Otherwise, if $\delta(I, s') \cap Q_{NS} = Z \neq \emptyset$, we take any string $s'@(t', y) \in L(G_{CSO})$, where the string y is a prefix of the unique string not marked by the automaton $\mathcal{A}_k$, which exists because $|y| = |P(t')| \leq k$. Then, the strings (t, y) and (t', y) are not marked by the automaton $\tilde{G}^+ \| \tilde{\mathcal{A}}_k$, and hence $\delta'(I, s@(t, y)) \cap Q'_S \neq \emptyset$ and $\delta'(I, s'@(t', y)) \cap Q'_{NS} = \delta'([\delta'(I, s'@) \cap Q^-], (t', y)) = \delta'(Z^-, (t', y)) = \emptyset$, where the set $Z^- = \{z^- \mid z \in Z\}$. Note that the string (t', y) is not generated in the system G_{CSO} from a state of the set Z^-, since the string t' cannot be generated in the system $G_{k\text{-}SO}$ from any state of $z \in Z$. Therefore, the system G_{CSO} is not current-state opaque. □

We now apply our transformations to solve the open problem concerning the complexity of deciding weak k-step opacity. Transformation 5.43 allows us to transform an instance of weak k-step opacity decision problem to a current-state opacity decision problem that can be solved in polynomial space. Combined with the PSPACE-hardness of weak k-step opacity from Corollary 5.23, we can generalize Theorem 4.7 for weak k-step opacity.

Corollary 5.46. *Given a natural number k represented by $O(\log(k))$ bits and a DES G. The problem of deciding whether the system G satisfies weak k-step opacity is* PSPACE*-complete. The problem remains* PSPACE*-complete even if the system G is a DFA with three events, one of which is unobservable.*

5.6.5 The case of $|\Sigma_o| = 1$ with neutral states

To preserve the number of observable events, our transformation of weak k-step opacity to current-state opacity relies on binary encoding by Transformation 4.3. This transformation requires at least two observable events in $G_{k\text{-}SO}$, and hence it is not applicable to systems with a single observable

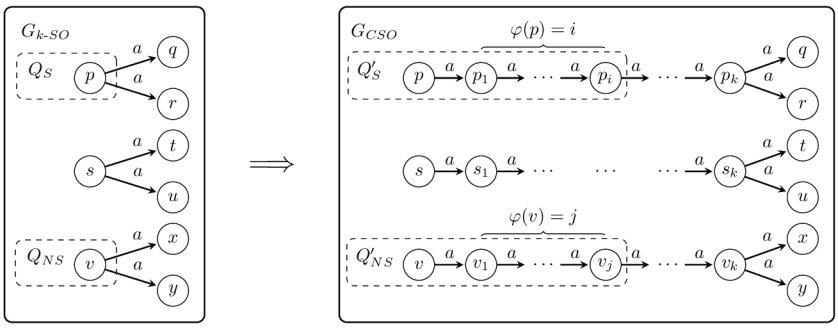


Figure 5.22: Transforming weak k-step opacity with a single observable event to current-state opacity.

event. For these systems, we provide two different transformations. First one, which allows neutral states, requires to add at most a quadratic number of new states.

Let the weak k-step opacity problem with a single observable event be represented by a DES $G^a_{k\text{-}SO}$. We transform it to a DES G^a_{CSO} in such a way that $G^a_{k\text{-}SO}$ is weakly k-step opaque if and only if G^a_{CSO} is current-state opaque.

Transformation 5.47. Let $G^a_{k\text{-}SO} = (Q, \Sigma, \delta, I)$ be a DES with a single observable event $\Sigma_o = \{a\}$, the secret states Q_S, the non-secret states Q_{NS}, the corresponding projection $P\colon \Sigma^* \to \{a\}^*$, and the parameter $k \in \mathbb{N}$. We construct a DES

$$G_{CSO} = (Q', \Sigma, \delta', I)$$

where δ' is initialized as δ and modified as follows using the function φ_k from Definition 2.6. For every state $p \in Q$ with $\varphi_k(p) > 0$, we add k new states $p_1, \ldots, p_k$ to Q' and k new transitions (p, a, p_1) and (p_i, a, p_{i+1}), for $i = 1, \ldots, k-1$, to δ'. Finally, we replace every observable transition (p, a, r) in δ' by the transition (p_k, a, r). We initialize the sets $Q'_S := Q_S$ and $Q'_{NS} := Q_{NS}$. For every state $p \in Q_S$ with $\varphi_k(p) = \ell > 0$, we add the corresponding states $p_1, \ldots, p_\ell$ to Q'_S and, for every $q \in Q_{NS}$ with $\varphi_k(q) = \ell > 0$, we add $q_1, \ldots, q_\ell$ to Q'_{NS}. ◇

Notice that Transformation 5.47 preserves determinism and, by the following remark, requires to add at most n^2 states, and hence it can be done in polynomial time.

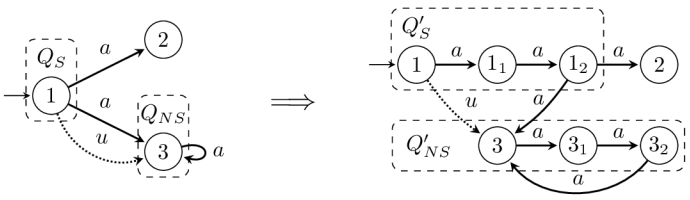


Figure 5.23: An example of the transformation of the k-SO problem with a single observable event (left) to the CSO problem (right).

Remark 5.48. *It follows directly from Definition 2.6 of the function φ_k that if $k \geq |Q|$, then a system with a single observable event is weakly k-step opaque if and only if it is weakly ∞-step opaque. Therefore, we may consider $k \leq |Q|$, which also covers the case of weak ∞-step opacity.*

We now provide an illustrative example.

Example 5.49. Let G over $\Sigma = \{a, u\}$ depicted in Figure 5.23 (left) be an instance of the 2-SO problem with a single observable event $\Sigma_o = \{a\}$, the set of secret states $Q_S = \{1\}$, and the set of non-secret states $Q_{NS} = \{3\}$. Then, $\varphi_2(1) = \varphi_2(3) = 2$, and Transformation 5.47 of 2-SO to CSO results in the DES G' depicted in Figure 5.23 (right) with the set of secret states Q'_S and the set of non-secret states Q'_{NS}. We consider two cases based on the presence of the unobservable transition $(1, u, 3)$ in G.

We first assume that the transition $(1, u, 3)$ exists in G. Then, G is weakly k-step opaque, for any $k \in \mathbb{N}_\infty$, because any string a^i leading from the secret state 1 is indistinguishable from the string ua^i that leads the system to the non-secret state 3. The reader can see that G' is current-state opaque, because a secret state is reachable only under a string of the form a^i, for $i \in \{0, 1, 2\}$, and for any such string there is an indistinguishable string ua^i reaching a non-secret state.

If the transition $(1, u, 3)$ does not exist in G, then G is not weakly 2-step opaque, because it is neither current-state opaque and neither G' is current-state opaque. ◇

The following theorem justifies the correctness of Transformation 5.47.

Theorem 5.50. *A DES $G^a_{k\text{-}SO}$ with a single observable event is weakly k-step opaque with respect to Q_S, Q_{NS}, and P if and only if the DES G^a_{CSO} obtained by Transformation 5.47 is current-state opaque with respect to Q'_S, Q'_{NS}, and P.*

Proof. Assume that $G^a_{k\text{-}SO}$ is not k-step opaque, that is, there is $st \in L(G^a_{k\text{-}SO})$ with $|P(t)| \leq k$ such that $\delta(\delta(I,s) \cap Q_S, t) \neq \emptyset$ and $\delta(\delta(I, P^{-1}P(s)) \cap Q_{NS}, P^{-1}P(t)) = \emptyset$. Let $f\colon \Sigma^* \to \Sigma^*$ be a morphism such that $f(a) = a^{k+1}$ and $f(b) = b$, for $a \neq b \in \Sigma_{uo}$. Then, by construction, $\delta(I,s) = \delta'(I, f(s))$, and hence $\delta'(I, f(s)) \cap Q'_S \neq \emptyset$. If $\delta(I, P^{-1}P(s)) \cap Q_{NS} = \emptyset$, then $\delta'(I, f(P^{-1}P(s))) \cap Q'_{NS} = \emptyset$ because $\delta(I, s') = \delta'(I, f(s'))$ for any $s' \in P^{-1}P(s)$, and G^a_{CSO} is not current-state opaque. Otherwise, we denote by $q_s \in \delta(I,s) \cap Q_S$ and $q_{ns} \in \delta(I, P^{-1}P(s)) \cap Q_{NS}$ the states with maximal $\varphi_k(q_s)$ and $\varphi_k(q_{ns})$. Since $G^a_{k\text{-}SO}$ is not weakly k-step opaque, $\varphi_k(q_s) > \varphi_k(q_{ns})$. Then, in G^a_{CSO}, q_s has exactly one outgoing observable transition and is followed by $\varphi_k(q_s) = \ell$ secret states, while q_{ns} is followed by $\varphi_k(q_{ns}) < \ell$ non-secret states. Therefore, $\delta'(I, f(s)a^\ell) \cap Q'_S \neq \emptyset$ and $\delta'(I, f(s')a^\ell) \cap Q'_{NS} = \emptyset$ for any $s' \in P^{-1}P(s)$, and hence G^a_{CSO} is not current-state opaque.

On the other hand, assume that $G^a_{k\text{-}SO}$ is weakly k-step opaque, and that $\delta'(I,w) \cap Q'_S \neq \emptyset$. We show that $\delta'(I, P^{-1}P(w)) \cap Q'_{NS} \neq \emptyset$. Consider a state $q_s \in \delta'(I,w) \cap Q'_S$ and a path π in G^a_{CSO} leading to q_s under w. Denote by p the last state of π that corresponds to a state of $G^a_{k\text{-}SO}$; that is, p is not a new state added by the construction of G^a_{CSO}. Since $q_s \in Q'_S$, we have, by construction, that $p \in Q_S$. Then the choice of p partitions $w = uv$, where u, read along the path π, leads to state p, and $v = a^\ell$ is a suffix of length $\ell \leq k$. Let u' be a string such that $f(u') = u$. Then $p \in \delta(I, u') \cap Q_S$. Since $\varphi_k(p) \geq \ell$, there exists t such that $P(t) = a^\ell$ and $\delta(\delta(I, u') \cap Q_S, t) \neq \emptyset$ in $G^a_{k\text{-}SO}$. Then weak k-step opacity of $G^a_{k\text{-}SO}$ implies that there exists u'' and t' such that $P(u') = P(u'')$, $P(t) = P(t')$, and $\delta(\delta(I, u'') \cap Q_{NS}, t') \neq \emptyset$. In particular, there is a state $q_{ns} \in \delta(I, u'') \cap Q_{NS}$ with $\varphi_k(q_{ns}) \geq \ell$, and $\delta'(I, f(u'')) \cap Q'_{NS} \neq \emptyset$. Therefore, $\delta'(I, f(u'')a^\ell) \cap Q'_{NS} \neq \emptyset$ and $P(f(u'')a^\ell) = P(uv) = P(w)$, which completes the proof. □

Transformation 5.47 allows us to transform an instance of weak k-step opacity decision problem to a current-state opacity decision problem, while preserving a single observable event. Combined with the coNP-hardness of weak k-step opacity with a single observable event from Corollary 5.33, we can generalize Theorem 4.8 for weak k-step opacity. Additionally, Remark 5.48 allows us to state the same result for weak ∞-step opacity.

Corollary 5.51. *Given a natural number k represented by $O(\log(k))$ bits and a DES G with a single observable event. The problem of deciding whether*

the system G satisfies weak k-step opacity is coNP*-complete. Analogously, the problem of deciding whether the system G satisfies weak ∞-step opacity is* coNP*-complete.*

5.6.6 The case of $|\Sigma_o| = 1$ without neutral states

To avoid introducing new neutral states into the system as in the previous transformation, we provide a separate transformation for cases where such states are not allowed.

Let the weak k-step opacity problem with a single observable event be represented by a DES G^a without neutral states. Since the following transformation does not change the structure of the system, we denote both the original and the resulting system simply by G^a. We transform the sets of secret and non-secret states of G^a in such a way that G^a is weakly k-step opaque with respect to Q_S and Q_{NS} if and only if G^a is current-state opaque with respect to Q'_S and Q'_{NS}.

Transformation 5.52. Let $G^a = (Q, \Sigma, \delta, I)$ be a DES with a single observable event $\Sigma_o = \{a\}$, the secret states Q_S, the non-secret states $Q_{NS} = Q - Q_S$, the corresponding projection $P\colon \Sigma^* \to \{a\}^*$, and the parameter $k \in \mathbb{N}$. We construct sets Q'_S and Q'_{NS} as follows. We determine (in linear time) whether the language $P(L(G^a))$ is finite.

(A) If so, we verify weak k-step opacity of the system G^a in linear time by checking the subsets of states $\delta(I, P^{-1}(a^i))$, for every $i \leq |Q| - 1$.

  (A1) If the system G^a is weakly k-step opaque, and hence also current-state opaque, we define the sets of secret states $Q'_S = Q_S$ and of non-secret states $Q'_{NS} = Q_{NS}$.

  (A2) If the system G^a is not weakly k-step opaque, we define the sets of secret states $Q'_S = Q$ and of non-secret states $Q'_{NS} = \emptyset$.

(B) If the language $P(L(G^a))$ is infinite, we define the set of non-secret states $Q'_{NS} = \{q \in Q_{NS} \mid \varphi_k(q) = k\}$ using the function φ_k from Definition 2.6, which assings to the state q the maximal number of observable steps that are possible from state q. Finally, we define the set of secret states to be $Q'_S = Q - Q'_{NS}$. ◇

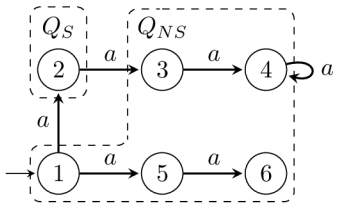


Figure 5.24: The DES G from Example 5.53.

Notice that Transformation 5.52 can be done in polynomial time and that it does not change the structure of the system in any way. Analogously to Transformation 5.47, we can consider $k \leq |Q|$ by Remark 5.48.

Example 5.53. Let G over $\Sigma = \{a\}$ depicted in Figure 5.24 be an instance of the weak k-step opacity problem with a single secret state $Q_S = \{2\}$ and five non-secret states $Q_{NS} = \{1, 3, 4, 5, 6\}$. Notice that the language $P(L(G))$ is infinite, and hence Transformation 5.52 applies option (B) to G.

For the parameter $k = 1$, the system G is weakly 1-step opaque and the transformation results in the set of secret states $Q'_S = \{2, 6\}$ and the set of non-secret states $Q'_{NS} = \{1, 3, 4, 5\}$. For example, the state 6 is now secret, since $\varphi_1(6) = 0 < k = 1$, while for every $q \in Q'_{NS}$ we have that $\varphi_1(q) = 1$. The reader can see that G is current-state opaque with respect to Q'_S and Q'_{NS}, because both strings a and aa that lead G to secret states also lead to non-secret states.

For the parameter $k = 2$, the system G is not weakly 2-step opaque and the transformation results in the set of secret states $Q'_S = \{2, 5, 6\}$ and the set of non-secret states $Q'_{NS} = \{1, 3, 4\}$. Therefore, G is not current-state opaque with respect to Q'_S and Q'_{NS}, since the string a leads G to states 2 and 5, both of which are secret. ◇

The following theorem justifies the correctness of Transformation 5.52.

Theorem 5.54. *A DES G^a with a single observable event $\Sigma_o = \{a\}$ is weakly k-step opaque with respect to Q_S, Q_{NS}, and P if and only if G^a is current-state opaque with respect to Q'_S, Q'_{NS}, and P defined by Transformation 5.52.*

Proof. If the system G^a is weakly k-step opaque with respect to Q_S, Q_{NS}, and P, either the language $P(L(G^a))$ is finite, and hence the system G^a is current-state opaque with respect to Q_S, Q_{NS}, and P, or the language $P(L(G^a))$ is infinite. In this case, for every string $w \in L(G^a)$, there is a

state $q \in \delta(I, P^{-1}P(w))$, such that $\varphi_k(q) = k$. Since the system G^a is weakly k-step opaque with respect to Q_S, Q_{NS}, and P, for every secret state $q_s \in \delta(I, P^{-1}P(w))$, there is a non-secret state $q_{ns} \in \delta(I, P^{-1}P(w))$ such that $\varphi_k(q_{ns}) \geq \varphi_k(q_s)$; in particular, there is a non-secret state $q'_{ns} \in \delta(I, P^{-1}P(w))$ such that $\varphi_k(q'_{ns}) = k$, that is, $q'_{ns} \in Q'_{NS}$, and hence the system G^a is current-state opaque with respect to Q'_{NS}, Q'_S, and P.

If the system G^a is not weakly k-step opaque with respect to Q_S, Q_{NS}, and P, then either the language $P(L(G^a))$ is finite, and hence the system G^a is not current-state opaque with respect to $Q'_S = Q$, $Q'_{NS} = \emptyset$, and P, or the language $P(L(G^a))$ is infinite. In this case, there exists a string $w \in L(G^a)$ and a secret state $q_s \in \delta(I, P^{-1}P(w))$ such that $\varphi_k(q_s) > \varphi_k(q_{ns})$ for every non-secret state $q_{ns} \in \delta(I, P^{-1}P(w))$. Hence, $\varphi_k(q_{ns}) < k$ for every non-secret state $q_{ns} \in \delta(I, P^{-1}P(w)) \cap Q_{NS}$. Thus $\delta(I, P^{-1}P(w)) \cap Q'_{NS} = \emptyset$, which shows that the system G^a is not current-state opaque with respect to Q'_S, Q'_{NS}, and P. □

5.7 k-SSO to k-SO

In this section, we show how to transform strong k-step opacity to weak k-step opacity. Our transformation proceeds in two steps. The first step of the transformation is called *normalization* and we also use it to describe the relationship between strong 0-step opacity and current-state opacity. The second step then transforms the normalized system to the weak k-step opacity instance.

5.7.1 Normalization

In what follows, we call the systems where there are no unobservable transitions from secret states to non-secret states *normal*. For systems that are not normal, we provide a construction to normalize them, that is, we eliminate unobservable transitions from secret states to non-secret states without affecting the property of being strongly k-step opaque.

Transformation 5.55. Let $G = (Q, \Sigma, \delta, q_0)$ be a deterministic DES with the secret states Q_S, the non-secret states $Q_{NS} = Q - Q_S$, the corresponding projection $P\colon \Sigma^* \to \Sigma_o^*$, and the parameter $k \in \mathbb{N}_\infty$. We construct a DES

$$G_{norm} = (Q_n, \Sigma, \delta_n, q_0)$$

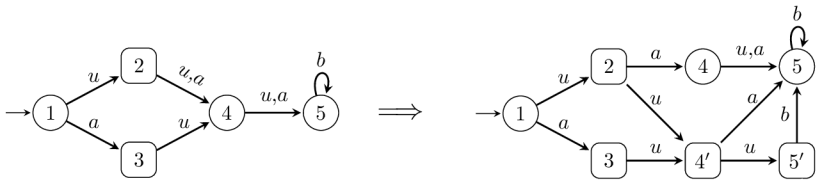


Figure 5.25: A deterministic DES G (left) and its normalization G_{norm} (right); the secret states are squared.

where $Q_n = Q \cup Q'$ for $Q' = \{q' \mid q \in Q\}$ being a disjoint copy of Q, and the transition function δ_n is defined as follows. We initialize $\delta_n := \delta$ and further modify it in the following four steps:

1. For every $p \in Q_S$, $q \in Q_{NS}$, and $u \in \Sigma_{uo}$, we replace the transition (p, u, q) by (p, u, q') in δ_n.
2. For every unobservable transition (p, u, q) in δ, that is, $u \in \Sigma_{uo}$, we add the transition (p', u, q') to δ_n.
3. For every observable transition (q, a, r) in δ, that is, $a \in \Sigma_o$, we add the transition (q', a, r) to δ_n.
4. We remove unreachable states and corresponding transitions.

The set of secret states of G_{norm} is the set $Q_n^S = Q_S \cup Q'$. The set of non-secret states Q_{NS} remains unchanged. ◇

In the sequel, we call G_{norm} the *normalization* of G. If G and G_{norm} coincide, we say that G is *normal*.

Example 5.56. To illustrate Transformation 5.55, consider the system depicted in Figure 5.25 (left). Its normalization G_{norm} is depicted in the same figure (right). States 2 and 3 of G are secret, events a and b are observable, and u is unobservable. The normalization G_{norm} of G initially contains five new secret states $1'$, $2'$, $3'$, $4'$, $5'$. Step (1) of Transformation 5.55 replaces transitions $(2, u, 4)$ and $(3, u, 4)$ by $(2, u, 4')$ and $(3, u, 4')$, respectively, step (2) adds four unobservable transitions $(1', u, 2')$, $(2', u, 4')$, $(3', u, 4')$, and $(4', u, 5')$, and step (3) adds the observable transitions $(1', a, 3)$, $(2', a, 4)$, $(4', a, 5)$ and $(5', b, 5)$. Finally, step (4) eliminates unreachable states $1'$, $2'$, $3'$, and the corresponding transitions. ◇

The following lemma compares the behaviors of the system G and its normalization G_{norm}.

Lemma 5.57. *Let $G = (Q, \Sigma, \delta, q_0)$ be a deterministic DES with the secret states Q_S and the non-secret states $Q_{NS} = Q - Q_S$. Let G_{norm} be the normalization of G obtained by Transformation 5.55. Then, for every $w \in \Sigma^*$ and $a \in \Sigma$, the following holds:*

1. *For $a \in \Sigma_{uo}$, $\delta(q_0, wa) = p$ if and only if $\delta_n(q_0, wa) \in \{p, p'\}$, where $p' \in Q'$ is a copy of $p \in Q$;*
2. *For $a \in \Sigma_o$, $\delta(q_0, wa) = \delta_n(q_0, wa)$;*
3. *$L(G) = L(G_{norm})$.*

Proof. We prove (1) and (2) by induction on the length of w. The induction hypothesis is that either $\delta(q_0, w) = p = \delta_n(q_0, w)$, or $\delta(q_0, w) = p$ and $\delta_n(q_0, w) = p'$.

To prove (1), let a be unobservable. We first consider the case $\delta(q_0, w) = \delta_n(q_0, w) = p$. First, if p is non-secret, Transformation 5.55 adds every transition $(p, a, q) \in \delta$ to δ_n. On the other hand, if p is secret, δ_n contains the transition (p, a, q') for every transition $(p, a, q) \in \delta$ with $q \in Q_{NS}$, and the transition (p, a, q) for every transition $(p, a, q) \in \delta$ with $q \in Q_S$. In both cases, Transformation 5.55 adds no other transition from state p to δ_n, and hence $\delta(q_0, wa) = \delta(p, a) = q$ if and only if $\delta_n(q_0, wa) = \delta_n(p, a) \in \{q, q'\}$. Notice that this case also covers the base case of the induction, since for $w = \varepsilon$ we have $\delta(q_0, w) = \delta_n(q_0, w) = q_0$.

Now, we consider the case $\delta_n(q_0, w) = p'$ and $\delta(q_0, w) = p$. Since Transformation 5.55 adds the transition (p', a, q') to δ_n for every unobservable transition $(p, a, q) \in \delta$, we have that $\delta(q_0, wa) = \delta(p, a) = q$ if and only if $\delta_n(q_0, wa) = \delta_n(p', a) = q'$.

To prove (2), let a be observable. We first consider the case $\delta(q_0, w) = \delta_n(q_0, w) = p$. Then, from the state p, Transformation 5.55 adds to δ_n all and only the observable transitions of δ, and hence $\delta(p, a) = \delta_n(p, a)$.

Now, we consider the case $\delta_n(q_0, w) = p'$ and $\delta(q_0, w) = p$. Then, Transformation 5.55 adds the transition (p', a, q) to δ_n for every observable transition $(p, a, q) \in \delta$, and therefore $\delta(q_0, wa) = \delta(p, a) = \delta_n(p', a) = \delta_n(q_0, wa)$.

Finally, $L(G) = L(G_{norm})$ of (3) follows from (1) and (2), since, for every $w \in \Sigma^*$, $\delta(q_0, w)$ is undefined if and only if $\delta_n(q_0, w)$ is undefined. □

The following lemma describes the meaning of normalization and states the main properties of a normalized DES.

Lemma 5.58. *For a deterministic DES $G = (Q, \Sigma, \delta, q_0)$ with the secret states Q_S, the non-secret states $Q_{NS} = Q - Q_S$, the observation $P\colon \Sigma^* \to \Sigma_o^*$, and the parameter $k \in \mathbb{N}_\infty$, let G_{norm} be the normalization of G obtained by Transformation 5.55. Then, the following holds true:*

1. *G_{norm} is deterministic;*
2. *In G_{norm}, there is no non-secret state reachable from a secret state by a sequence of unobservable events, i.e., $\delta_n(Q_n^S, P^{-1}(\varepsilon)) \cap (Q_n - Q_n^S) = \emptyset$;*
3. *G is strongly k-step opaque with respect to Q_S and P if and only if G_{norm} is strongly k-step opaque with respect to Q_n^S and P.*

Proof. To prove (1), we analyze the steps of Transformation 5.55 creating δ_n. First, δ_n is defined as δ, which is deterministic. Then, step (1) replaces some unobservable transitions, which is an operation that preserves determinism of δ_n. Step (2) adds the transition (p', u, q') for every unobservable transition (p, u, q) in G. Similarly, step (3) adds the transition (q', a, p) for every observable transition (q, a, p) in G. Since G is deterministic, steps (2) and (3) preserve determinism. Altogether, G_{norm} is deterministic.

To prove (2), step (1) of Transformation 5.55 replaces all unobservable transitions from a secret state to a non-secret state by transitions from a secret state to a new secret state. Step (2) adds unobservable transitions only between the new states, which are all secret. Since no unobservable transition is defined from the new states to the old states, there is no non-secret state in G_{norm} reachable from a secret state by a sequence of unobservable events.

To prove the first direction of (3), we assume that G is strongly k-step opaque with respect to Q_S and P, and show that then G_{norm} is strongly k-step opaque with respect to Q_n^S and P. To this end, we show that for every string $s \in L(G_{norm})$, there exists a string $w \in L(G_{norm})$ such that $P(s) = P(w)$ and, for every prefix w' of w, if $|P(w)| - |P(w')| \leq k$, then $\delta_n(q_0, w') \notin Q_n^S$. Thus, let $s \in L(G_{norm})$ be an arbitrary string. Then, by Lemma 5.57, $s \in L(G_{norm}) = L(G)$, and since G is strongly k-step opaque with respect to Q_S and P, there is a string $\tilde{w} \in L(G)$ such that $P(s) = P(\tilde{w})$ and, for every prefix $\tilde{w}'$ of $\tilde{w}$, if $|P(\tilde{w})| - |P(\tilde{w}')| \leq k$, then $\delta(q_0, \tilde{w}') \notin Q_S$. By defining $w = \tilde{w}$, we obtain that the string $w \in L(G) = L(G_{norm})$ and that $P(s) = P(w)$. It remains to show that for every prefix w' of w, if $|P(w)| - |P(w')| \leq k$,

then $\delta_n(q_0, w') \notin Q_n^S$. To this end, let $xy = w$ be the decomposition of w, where x is the shortest prefix of w such that $|P(w)| - |P(x)| \leq k$. Then, x is either empty or ends with an observable event. Hence, by Lemma 5.57, $\delta(q_0, x) = \delta_n(q_0, x) = q \in Q$ in G. However, for every prefix y' of y, the string xy' is a prefix of $\tilde{w}$ satisfying $|P(\tilde{w})| - |P(xy')| \leq k$, and hence $\delta(q_0, xy') \notin Q_S$. In other words, the computation of $\delta(q, y)$ in G does not go through a secret state, and therefore the same sequence of transitions exists in G_{norm}, that is, $\delta(q_0, xy') = \delta_n(q_0, xy') \notin Q_n^S = Q_S \cup Q'$. Since every prefix w' of w satisfying $|P(w)| - |P(w')| \leq k$ is of the form $w' = xy'$, where y' is a prefix of y, we have shown that $\delta_n(q_0, w') \notin Q_n^S$, which was to be shown.

To prove the other direction, we assume that G is not strongly k-step opaque with respect to Q_S and P, and show that neither the G_{norm} is strongly k-step opaque with respect to Q_n^S and P. To this end, let $s \in L(G)$ be a string violating strong k-step opacity in G; that is, for every $w \in L(G)$ such that $P(s) = P(w)$, there exists a prefix w' of w such that $|P(w)| - |P(w')| \leq k$ and $\delta(q_0, w') = q_w \in Q_S$. However, by Lemma 5.57, $w \in L(G_{norm}) = L(G)$ and $\delta_n(q_0, w') \in \{q_w, q'_w\}$. Since both states $q_w, q'_w \in Q_n^S = Q_S \cup Q'$ are secret in G_{norm}, we conclude that G_{norm} is not strongly k-step opaque with respect to Q_n^S and P. □

In the following theorem, we discuss the relationship between strong 0-step opacity and weak 0-step (current-state) opacity for normal deterministic DESs. This result characterizes the relationship between these two notions and fixes the claim of Ma et al. [35] stating that strong 0-step opacity reduces to current-state opacity, which is not the case as shown in Example 3.11.

Theorem 5.59. *A normal deterministic DES $G = (Q, \Sigma, \delta, q_0)$ is strongly 0-step opaque with respect to Q_S and P if and only if G is weakly 0-step opaque with respect to Q_S, $Q_{NS} = Q - Q_S$, and P.*

Proof. We first assume that $G = (Q, \Sigma, \delta, q_0)$ is strongly 0-step opaque with respect to Q_S and P. To show that G is weakly 0-step opaque with respect to Q_S and P, let $st \in L(G)$ be such that $|P(t)| \leq 0$ and $\delta(q_0, s) \in Q_S$. Since $st \in L(G)$ and G is deterministic, $\delta(q_0, st)$ is defined. Therefore, we need to show that there is a string $s't' \in L(G)$ such that $P(s) = P(s')$, $P(t) = P(t')$, and $\delta(q_0, s') \in Q_{NS} = Q - Q_S$. However, since G is strongly k-step opaque with respect to Q_S and P, there is a string $w \in L(G)$ such that $P(w) = P(st)$ and, for every prefix w' of w with $|P(w)| - |P(w')| = 0$, $\delta(q_0, w') \in Q - Q_S$.

Let w' be any, but fixed, such prefix of w. We set $s' = w'$ and $s't' = w$. Then, $P(s') = P(w') = P(w) = P(st) = P(s)$, $P(t') = \varepsilon = P(t)$, and $\delta(q_0, s') = \delta(q_0, w') \in Q - Q_S = Q_{NS}$. Thus, we have shown that G is weakly 0-step opaque with respect to Q_S and P.

For the other direction, we assume that G is not strongly 0-step opaque with respect to Q_S and P. To show that G is neither weakly 0-step opaque with respect to Q_S and P, we need to find a string $st \in L(G)$ with $|P(t)| \leq 0$ and $\delta(q_0, s) \in Q_S$ such that, for every string $s't' \in L(G)$ with $P(s) = P(s')$ and $P(t) = P(t')$, the state $\delta(q_0, s') \in Q_S$. However, from the assumption that G is not strongly 0-step opaque with respect to Q_S and P, we have a string $st \in L(G)$ such that $\delta(q_0, s) \in Q_S$, $|P(t)| \leq 0$, and, for every string $w \in L(G)$ with $P(st) = P(w)$, there is a prefix w' of w such that $|P(w)| - |P(w')| = 0$ and $\delta(q_0, w') \in Q_S$. To complete the proof, we show that for every $s't' \in L(G)$ with $P(s) = P(s')$ and $P(t) = P(t')$, the state $\delta(q_0, s')$ is secret. To this end, let $xy = s't'$ be the decomposition of $s't'$ such that y is the longest suffix of $s't'$ consisting only of unobservable events. Notice that x is a prefix of s', because $P(t') = P(t) = \varepsilon$. Since $P(st) = P(x)$, there must be a prefix x' of x such that $|P(x)| - |P(x')| = 0$ and $\delta(q_0, x') \in Q_S$. However, the last event of x is observable, and hence the only prefix x' of x for which $|P(x)| - |P(x')| = 0$ is $x' = x$, and therefore $\delta(q_0, x) = \delta(q_0, x') \in Q_S$. Since G is normal, there are no non-secret states reachable from the secret state $\delta(q_0, x)$ under a sequence of unobservable events. In particular, $\delta(q_0, s') = \delta(q_0, xy') \in Q_S$, where y' is a prefix of y for which $s' = xy'$; recall that y is the longest suffix of $s't'$ consisting only of unobservable events. Thus, we have shown that G is not weakly 0-step opaque. □

5.7.2 Normalized k-SSO to k-SO

Let the strong k-step opacity problem be represented by a DES $G_{k\text{-}SSO}$. We transform it to a DES $G_{k\text{-}SO}$ in such a way that $G_{k\text{-}SSO}$ is strongly k-step opaque if and only if $G_{k\text{-}SO}$ is weakly k-step opaque. In the construction, we assume that $G_{k\text{-}SSO}$ is a normal deterministic DES. By Lemma 5.58, this assumption is without loss of generality, because if $G_{k\text{-}SSO}$ is not normal, then we can consider its normalization instead.

Transformation 5.60. Let $G_{k\text{-}SSO} = (Q, \Sigma, \delta, q_0)$ be a normal deterministic DES with the secret states Q_S, the non-secret states $Q_{NS} = Q - Q_S$, the

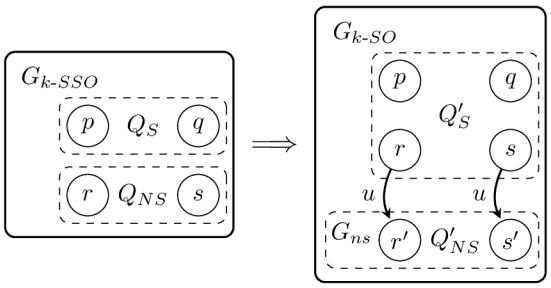


Figure 5.26: Transforming strong k-step opacity to weak k-step opacity.

corresponding projection $P\colon \Sigma^* \to \Sigma_o^*$, and the parameter $k \in \mathbb{N}_\infty$. We construct a DES

$$G_{k\text{-}SO} = (Q \cup Q'_{NS}, \Sigma \cup \{u\}, \delta', q_0)$$

as a disjoint union of $G_{k\text{-}SSO}$ and $G_{ns} = (Q'_{NS}, \Sigma, \delta_{ns}, q'_0)$, where G_{ns} is obtained from $G_{k\text{-}SSO}$ by removing all secret states and corresponding transitions, and $Q'_{NS} = \{q' \mid q \in Q_{NS}\}$ is a copy of Q_{NS} disjoint from Q. We use a new unobservable event u to connect G_{ns} to $G_{k\text{-}SSO}$ so that we initialize $\delta' := \delta \cup \delta_{ns}$ and extend δ' by additional transitions (q, u, q') for every $q \in Q_{NS}$, cf. Figure 5.26 for an illustration. The states of Q'_{NS} are the only non-secret states of $G_{k\text{-}SO}$, that is, the set of secret states of $G_{k\text{-}SO}$ is the set $Q'_S = Q$. Finally, we define the projection $P'\colon (\Sigma \cup \{u\})^* \to \Sigma_o^*$. ◇

Notice that both Transformations 5.55 and 5.60 can be done in polynomial time and that they preserve determinism and the number of observable events. In addition, they are independent of the parameter k, and hence they work for any $k \in \mathbb{N}_\infty$ without affecting the size of the resulting system $G_{k\text{-}SO}$.

Example 5.61. Let G_{norm} over $\Sigma = \{a, b, u_1\}$ depicted in Figure 5.27 (left) be the normalized instance of the strong k-step opacity problem from Example 3.11 with an unobservable event u_1 and two secret states 5 and $6'$. Since the non-secret state 6 was reachable in the original DES G only from the secret state 5 by an unobservable event, we replace it by the secret state $6'$ in G_{norm}. Transformation 5.60 of k-SSO to k-SO then results in the DES G' depicted in Figure 5.27 (right) with a new unobservable event u_2, a set of secret states Q'_S that is equal to the set of states of the original system G_{norm}, and five new non-secret states $Q'_{NS} = \{1', 2', 3', 4', 7'\}$. We distinguish two cases depending on whether event b is observable or not.

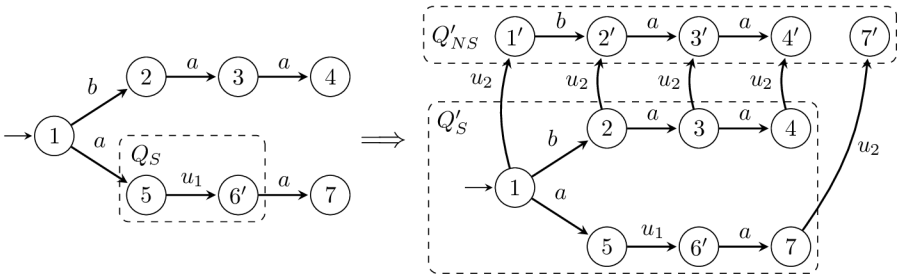


Figure 5.27: An example of the transformation of the k-SSO problem (left) to the k-SO problem (right).

If event b is unobservable, then G_{norm} is strongly k-step opaque for any $k \in \mathbb{N}_\infty$, see Example 3.11 for an explanation. The reader can verify that in G', for every string st such that $\delta'(\delta'(1,s) \cap Q'_S, t) \neq \emptyset$, there is a string $s't' \in \overline{u_2baa}$ such that $P'(s) = P'(s')$, $P'(t) = P'(t')$, and $\delta'(\delta('1,s') \cap Q'_{NS}, t') \neq \emptyset$. Therefore, G' is weakly k-step opaque, for any parameter $k \in \mathbb{N}_\infty$.

In the second case, we assume that event b is observable. In this case, G_{norm} is not even strongly 0-step opaque as we have shown in Example 3.11 for the original DES G. In G', we have $\delta'(1, P^{-1}P(a)) = \{5, 6'\} \subseteq Q_S$, therefore G' is not weakly 0-step opaque. ◇

The following theorem describes the relationship between strong k-step opacity and weak k-step opacity, and justifies the correctness of Algorithm 4 in Section 6.3.

Theorem 5.62. *A normal deterministic DES $G_{k\text{-}SSO}$ is strongly k-step opaque with respect to Q_S and P if and only if the DES $G_{k\text{-}SO}$ obtained by Transformation 5.60 is weakly k-step opaque with respect to Q'_S, Q'_{NS}, and P'.*

Proof. For the first implication, we assume that $G_{k\text{-}SSO}$ is strongly k-step opaque with respect to Q_S and P, and we show that $G_{k\text{-}SO}$ is weakly k-step opaque with respect to Q'_S, Q'_{NS}, and P'. To this end, let $st \in L(G_{k\text{-}SO})$ be such that $|P'(t)| \leq k$ and $\delta'(q_0, s) \in Q'_S$. We need to show that there is a string $s't' \in L(G_{k\text{-}SO})$ such that $P'(s) = P'(s')$, $P'(t) = P'(t')$, and $\delta'(q_0, s') \in Q'_{NS}$.

Let P_u denote the projection that removes every occurrence of event u, that is, $P_u(a) = a$ for $a \in \Sigma$, and $P_u(u) = \varepsilon$. We first show that $P_u(st) \in$

$L(G_{k\text{-}SSO})$. Indeed, if st does not contain u, then $P_u(st) = st \in L(G_{k\text{-}SSO})$. If st contains u, then, by the construction of $G_{k\text{-}SO}$, any string of $L(G_{k\text{-}SO})$ contains at most one occurrence of u. Since $\delta'(q_0, s) \in Q'_S$, we have that u occurs in t. Let $st = st_1ut_2$. Then, there are states $p, r \in Q$ in $G_{k\text{-}SO}$ such that $\delta'(q_0, st_1) = p$, $\delta'(p, u) = p'$, and $\delta'(p', t_2) = r'$. However, by the construction, this means that $\delta(q_0, st_1) = p$ and $\delta(p, t_2) = r$ in $G_{k\text{-}SSO}$, and hence $P_u(st) = st_1t_2 \in L(G_{k\text{-}SSO})$.

Since $G_{k\text{-}SSO}$ is strongly k-step opaque with respect to Q_S and P, there exists a string $w \in L(G_{k\text{-}SSO})$ such that $P(P_u(st)) = P(w)$ and, for every prefix w' of w, if $|P(w)| - |P(w')| \leq k$, then $\delta(q_0, w') \notin Q_S$. Since $P'(st) = P(P_u(st)) = P(w)$, we define $xy = w$ to be a (fixed) decomposition of w such that $P'(s) = P(x)$ and $P'(t) = P(y)$. Then, $|P(w)| - |P(x)| = |P'(st)| - |P'(s)| = |P'(t)| \leq k$, which implies that $\delta(q_0, x) = \delta'(q_0, x) = q$ for some state q that is not secret in $G_{k\text{-}SSO}$. Therefore, the transition $(q, u, q') \in \delta'$, and hence $\delta'(q_0, xu) = q' \in Q'_{NS}$. Since the state $\delta(q_0, xy') \notin Q_S$ for every prefix y' of y, because xy' is a prefix of w with $|P(w)| - |P(xy')| \leq k$, the computation of $\delta(q, y)$ in $G_{k\text{-}SSO}$ does not go through a secret state. Therefore, the same sequence of transitions is enabled in $G_{k\text{-}SO}$ from state q'. Setting now $s' = xu$ and $t' = y$ implies that $P'(s) = P'(s')$, $P'(t) = P'(t')$, $\delta'(q_0, s') \in Q'_{NS}$, and $\delta'(q_0, s't')$ is defined, which proves that $G_{k\text{-}SO}$ is weakly k-step opaque.

To prove the other direction, we assume that $G_{k\text{-}SSO}$ is not strongly k-step opaque with respect to Q_S and P, and we show that $G_{k\text{-}SO}$ is not weakly k-step opaque with respect to Q'_S, Q'_{NS}, and P'. To this end, we need to show that there is a string $st \in L(G_{k\text{-}SO})$ such that $|P'(t)| \leq k$, $\delta'(q_0, s) \in Q'_S$, and for every $s't' \in L(G_{k\text{-}SO})$ such that $P'(s) = P'(s')$ and $P'(t) = P'(t')$, the state $\delta'(q_0, s') \notin Q'_{NS}$.

However, since $G_{k\text{-}SSO}$ is not strongly k-step opaque with respect to Q_S and P, there exists a string $v \in L(G_{k\text{-}SSO})$ such that, for every string $w \in L(G_{k\text{-}SSO})$ with $P(w) = P(v)$, there is a prefix w' of w such that $|P(w)| - |P(w')| \leq k$ and $\delta(q_0, w') \in Q_S$. In particular, there is a prefix v' of v such that $|P(v)| - |P(v')| \leq k$ and $\delta(q_0, v') \in Q_S$.

Let $xy = v$ be the decomposition of v such that y is the longest suffix of v containing at most k observable events. We set $s = x$ and $t = y$, for which we have that $|P(t)| \leq k$, $\delta'(q_0, st)$ is defined, and, since neither s nor t contains the event u, the state $\delta'(q_0, s) \in Q'_S$. It remains to show that for every string $s't' \in L(G_{k\text{-}SO})$ with $P'(s') = P'(s)$ and $P'(t') = P'(t)$, the state

$\delta'(q_0, s') \notin Q'_{NS}$. We distinguish two cases.

In the first case, we assume that $\delta(q_0, P^{-1}P(s)) \cap Q_{NS} = \emptyset$, and we consider any string $s't' \in L(G_{k\text{-}SO})$ such that $P'(s') = P'(s)$ and $P'(t') = P'(t)$. If s' does not contain the event u, then $s' \in P^{-1}P(s)$, and therefore $\delta'(q_0, s') = \delta(q_0, s') \in Q_S \subseteq Q'_S$. If, on the other hand, s' contains the event u, then $s' = s_1 u s_2$ where neither s_1 nor s_2 contains the event u. But then $\delta'(q_0, s') = r'$, where r' is a copy of $r = \delta(q_0, s_1 s_2)$. Since $s_1 s_2 \in P^{-1}P(s)$, the state $r = \delta(q_0, s_1 s_2) \in Q_S$, and hence $r' \notin Q'_{NS}$ by construction. In both cases, $\delta'(q_0, s') \notin Q'_{NS}$, which was to be shown.

In the second case, let $\delta(q_0, P^{-1}P(s)) \cap Q_{NS} = Z \neq \emptyset$, and consider any string $s't' \in L(G_{k\text{-}SO})$ with $P'(s') = P'(s)$ and $P'(t') = P'(t)$. Using the projection P_u removing the event u, we set $z := P_u(s't') \in L(G_{k\text{-}SSO})$. Recall that the string st does not contain the event u, that is, $P'(st) = P(st)$, and therefore $P(z) = P(P_u(s't')) = P'(s't') = P'(st) = P(st) = P(v)$. Since $G_{k\text{-}SSO}$ is not strongly k-step opaque with respect to Q_S and P, there is a prefix z' of z such that $|P(z)| - |P(z')| \leq k$ and $q_s := \delta(q_0, z') \in Q_S$. In particular, by the choice of s, we have that $|P(s)| \leq |P(z')|$. Furthermore, $G_{k\text{-}SSO}$ is normal, and hence there is no non-secret state reachable from the secret state q_s by a sequence of unobservable events.

In particular, the prefix $P_u(s')$ of the string $z = P_u(s')P_u(t')$ satisfies $P(P_u(s')) = P'(s') = P'(s) = P(s)$, where the last equality comes from the fact that s does not contain the event u. Then $P_u(s') \in P^{-1}P(s)$, and hence if $\delta(q_0, P_u(s')) \in Q_{NS}$, then $\delta(q_0, P_u(s')) \in Q_{NS} \cap Z$. Thus, assume that $\delta(q_0, P_u(s')) \in Q_{NS} \cap Z$. Then, the string $P_u(s')$ is a strict prefix of z'; otherwise, if z' was a strict prefix of $P_u(s')$, then we would have that $|P(z')| \leq |P(P_u(s'))| = |P(s)|$, which, together with $|P(s)| \leq |P(z')|$, would give that $|P(z')| = |P(P_u(s'))| = |P(s)|$, and hence the non-secret state $q_{ns} = \delta(q_0, P_u(s'))$ would be reachable from the secret state q_s by a sequence of unobservable events, which is a contradiction with the normality of $G_{k\text{-}SSO}$. Consequently, generating the string $P_u(t')$ from the state q_{ns}, $G_{k\text{-}SSO}$ must go through the secret state q_s. In other words, q_s is reachable from the state q_{ns} by a prefix of $P_u(t')$.

Thus, in $G_{k\text{-}SO}$, state $\delta'(q_0, s') \in \{q_{ns}, q'_{ns}\}$, where $q'_{ns} \in Z' = \{q' \mid q \in Z\} \subseteq Q'_{NS}$. If $\delta'(q_0, s') = q_{ns} \in Q'_S$, we are done. If $\delta'(q_0, s') = q'_{ns} \in Q'_{NS}$, we show that $\delta'(q'_{ns}, t')$ is undefined, which contradicts the assumption that $s't' \in L(G_{k\text{-}SO})$, and hence $\delta'(q_0, s') = q'_{ns} \in Q'_{NS}$ cannot happen. Indeed, if $\delta'(q_0, s') \in Q'_{NS}$, then $P_u(t') = t'$. Since the computation of $\delta(q_{ns}, t') =$

$\delta(q_{ns}, P_u(t'))$ in $G_{k\text{-}SSO}$ goes through the secret state q_s, the computation of $\delta'(q'_{ns}, t')$ in $G_{k\text{-}SO}$ has to go through the state q'_s, which is the primed copy of the state q_s. But the computation $\delta'(q'_{ns}, t')$ is performed in the automaton G_{ns}, which is obtained from $G_{k\text{-}SSO}$ by removing all secret states and corresponding transitions. Since q'_s is a copy of a secret state, it does not exist in G_{ns}, and hence it does not belong to Q'_{NS}. Therefore, $\delta'(q'_{ns}, t')$ is undefined.

We have thus shown that $G_{k\text{-}SO}$ is not weakly k-step opaque. □

We now apply our transformations to solve the open problem concerning the complexity of deciding strong k-step opacity. Transformation 5.60 allows us to transform an instance of strong k-step opacity decision problem to a weak k-step opacity decision problem. Combined with the PSPACE-hardness of strong k-step opacity from Corollary 5.23 and PSPACE-completeness of weak k-step opacity from Corollary 5.46, we can generalize Theorem 4.7 for strong k-step opacity.

Corollary 5.63. *Given a natural number k represented by $O(\log(k))$ bits and a DES G. The problem of deciding whether the system G satisfies strong k-step opacity is* PSPACE*-complete. The problem remains* PSPACE*-complete even if the system G is a DFA with three events, one of which is unobservable.*

Analogously, we generalize Theorem 4.8 for systems with a single observable event using Transformation 5.60 together with coNP-hardness of strong k-step opacity from Corollary 5.29 and coNP-completeness of weak k-step opacity from Corollary 5.33.

Corollary 5.64. *Given a natural number k represented in unary and a DES G with a single observable event. The problem of deciding whether the system G satisfies strong k-step opacity is* coNP*-complete.*

Chapter 6

Verification of opacity

In this chapter, we introduce three new algorithms for verifying language-based opacity and trace opacity (Algorithm 1), weak k-step opacity (Algorithm 2), and strong k-step opacity (Algorithm 4). Note that our algorithms for k-step notions are applicable with the parameter $k = \infty$, and thus can also verify weak and strong ∞-step opacity.

Each section contains an analysis of the complexity of the proposed algorithm, as well as a comparison with previously existing results.

6.1 Verification of LBO and TO

The algorithmic complexity of deciding whether a given DES is language-based opaque with respect to given secret and non-secret languages has been investigated in the literature. Lin [34] suggested an algorithm with the complexity $O(2^{2n})$, where n is the order of the state spaces of the automata representing the secret and non-secret languages. The same complexity has been achieved by Wu and Lafortune [50] using the transformation to current-state opacity. We improve this complexity with Algorithm 1.

The language-based opacity verification problem consists of a DES G, a projection $P\colon \Sigma^* \to \Sigma_o^*$, a secret language $L_S = L(\mathcal{A}_S)$ given by the non-blocking automaton $\mathcal{A}_S$, and a non-secret language $L_{NS} = L(\mathcal{A}_{NS})$ given by the non-blocking automaton $\mathcal{A}_{NS}$. The complexity improvement of Algorithm 1 comes from solving the language inclusion problem $P(L_S) \subseteq P(L_{NS})$ by the intersection of the projected automaton $P(\mathcal{A}_S)$ with the observer co-$\mathcal{A}_{NS}^{obs}$, instead of the intersection of two observer structures as in [34].

Algorithm 1 Verification of language-based opacity

Require: A DES $G = (Q, \Sigma, \delta, I)$, automata $\mathcal{A}_S$ and $\mathcal{A}_{NS}$, and $\Sigma_o \subseteq \Sigma$
Ensure: `true` if and only if G is language-based opaque with respect to $L_S = L(\mathcal{A}_S)$, $L_{NS} = L(\mathcal{A}_{NS})$, and $P\colon \Sigma^* \to \Sigma_o^*$

1: Compute the projected automaton $P(\mathcal{A}_S)$ of $\mathcal{A}_S$
2: Compute the observer $\mathcal{A}_{NS}^{obs}$ of $\mathcal{A}_{NS}$
3: Compute the complement co-$\mathcal{A}_{NS}^{obs}$ of $\mathcal{A}_{NS}^{obs}$
4: Compute the instersection automaton $\mathcal{C} = P(\mathcal{A}_S) \cap \text{co-}\mathcal{A}_{NS}^{obs}$
5: **if** $L_m(\mathcal{C}) = \emptyset$ **then**
6: **return** `true`
7: **else**
8: **return** `false`
9: **end if**

We now prove the correctness of our algorithm.

Theorem 6.1. *A DES G is language-based opaque with respect to L_S, L_{NS}, and P if and only if Algorithm 1 returns* `true`.

Proof. We have $P(L_S) \subseteq P(L_{NS})$ if and only if $P(L_S) \cap \text{co-}P(L_{NS}) = \emptyset$, where co-$P(L_{NS})$ stands for $\Sigma^* - P(L_{NS})$. We represent $P(L_S)$ by the projected automaton $P(\mathcal{A}_S)$ and co-$P(L_{NS})$ by the complement of the observer of $\mathcal{A}_{NS}$, denoted by co-$\mathcal{A}_{NS}^{obs}$. The problem is now equivalent to checking whether the language of $P(\mathcal{A}_S) \cap \text{co-}\mathcal{A}_{NS}^{obs}$ is empty, which means to search the structure for a reachable marked state. □

We now discuss the complexity of our algorithm.

Theorem 6.2. *The space and time complexity of Algorithm 1 is $O(n_1 2^{n_2})$ and $O((n_1 + m)2^{n_2})$, respectively, where n_1 is the number of states of the automaton $\mathcal{A}_S$, n_2 is the number of states of the automaton $\mathcal{A}_{NS}$, and m is the number of transitions of $P(\mathcal{A}_S)$. In particular, $m \leq \ell n_1^2$, where ℓ is the number of observable events.*

Proof. The projected automaton $P(\mathcal{A}_S)$ has n_1 states and m transitions, and co-$\mathcal{A}_{NS}^{obs}$, has at most 2^{n_2} states and $\ell 2^{n_2}$ transitions. Therefore, we search the automaton $P(\mathcal{A}_S) \cap \text{co-}\mathcal{A}_{NS}^{obs}$ that has at most $O(n_1 2^{n_2})$ states and $O(m2^{n_2})$ transitions. Since $m > \ell$, the proof is complete. □

(a) The case of $c \in \Sigma_{uo}$. (b) The case of $c \in \Sigma_o$.

Figure 6.1: The relevant part of the intersection automaton $\mathcal{C}$ for both observability cases of the event c.

Notice that Algorithm 1 can be used to verify trace opacity, since Remark 5.16 provides a procedure for constructing automata $\mathcal{A}_S$ and $\mathcal{A}_{NS}$ from the trace opacity problem instance. Since the size of $\mathcal{A}_S$ is at most twice the size of the original n-state system and $\mathcal{A}_{NS}$ has exactly n states, we obtain the same complexity $O((n+m)2^n)$ also for verifying trace opacity.

Another use of the algorithm is verification of the special case of initial-and-final-state opacity considered in [50]. If the secret and non-secret pairs are of the form $Q_S = I_S \times F_S$ and $Q_{NS} = I_{NS} \times F_{NS}$, where $I_S, I_{NS} \subseteq I$ and $F_S, F_{NS} \subseteq Q$, then we use languages of $\mathcal{A}_S = (Q, \Sigma, \delta, I_S, F_S)$ and $\mathcal{A}_{NS} = (Q, \Sigma, \delta, I_{NS}, F_{NS})$ for the secret and non-secret languages, respectively.

Example 6.3. To illustrate the algorithm, we use the DES G from Example 3.2. Again, we distinguish two cases depending on whether event c is observable or not. Since $\mathcal{A}_S$ does not have any unobservable transition, then $P(\mathcal{A}_S) = \mathcal{A}_S$.

In the case of $c \in \Sigma_{uo}$, the G is language-based opaque, and hence the intersection automaton $\mathcal{C}$ does not contain any marked state, cf. Figure 6.1(a) for an illustration. In particular, $\mathcal{C}$ does not mark any string in the language $P(L_S) \cap \text{co-}P(L_{NS}) = \emptyset$, and therefore Algorithm 1 returns `true`.

On the other hand, if $c \in \Sigma_o$, then G is not language-based opaque and the intersection automaton $\mathcal{C}$ contains marked state $(3, \emptyset)$, see Figure 6.1(b). Therefore, Algorithm 1 returns `false` because observing any string from $P(L_S) \cap \text{co-}P(L_{NS}) = abb^*$ violates language-based opacity. $\diamond$

6.2 Verification of k-SO

This particular version of the algorithm for verifying weak k-step opacity was presented in [10], which was itself a revision of our previous algorithm from [7]. Initially, we provide an overview of the algorithm and its use of

Algorithm 2 Verification of weak k-step opacity

Require: A DES $G = (Q, \Sigma, \delta, I)$, $Q_S, Q_{NS} \subseteq Q$, $\Sigma_o \subseteq \Sigma$, and $k \in \mathbb{N}_\infty$.
Ensure: `true` if and only if G is weakly k-step opaque with respect to Q_S, Q_{NS}, and $P\colon \Sigma^* \to \Sigma_o^*$

1: Set $Y := \emptyset$
2: Compute the observer G^{obs} of G
3: Compute the projected automaton $P(G)$ of G
4: **for** every state X of G^{obs} **do**
5: **for** every state $x \in X \cap Q_S$ **do**
6: add state $(x, X \cap Q_{NS})$ to set Y
7: **end for**
8: **end for**
9: Construct H as the part of the full observer of G accessible from the states of the second components of Y
10: Compute the product automaton $\mathcal{C} = P(G) \times H$
11: Use the Breadth-First Search (BFS) of Algorithm 3 to mark all states of $\mathcal{C}$ reachable from the states of Y in at most k steps
12: **if** $\mathcal{C}$ contains a marked state of the form $(q, \emptyset)$ **then**
13: **return** `false`
14: **else**
15: **return** `true`
16: **end if**

Breadth-First Search. Following that, we analyse the time and space complexity of the algorithm and compare it with previously existing algorithms.

We remind that the weak k-step opacity verification problem consists of a DES G, a projection $P\colon \Sigma^* \to \Sigma_o^*$, a set of secret states $Q_S \subseteq Q$, a set of non-secret states $Q_{NS} \subseteq Q$, and parameter $k \in \mathbb{N}_\infty$. Note that the system may contain neutral states.

In Algorithm 2 we describe our new algorithm verifying weak k-step opacity. The idea of the algorithm is as follows. We first compute the observer of G, denoted by G^{obs}, and the projected automaton of G, denoted by $P(G)$. Then, for every reachable state X of G^{obs}, we add the pairs $(x, X \cap Q_{NS})$ to the set Y, where x is a secret state of X and $X \cap Q_{NS}$ is the set of all non-secret states of X. Intuitively, in these states, the intruder estimates that G may be in the secret state x or in the non-secret states of $X \cap Q_{NS}$.

To verify that the intruder does not reveal the secret state, we need to check that every possible path of length up to k starting in x is accompanied by a path with the same observation starting in a non-secret state of $X \cap Q_{NS}$. To this end, we construct the automaton H as the part of the full observer of G consisting only of states reachable from the states forming the second components of the pairs in Y, and the automaton $\mathcal{C} = P(G) \times H$ as the product automaton of the projected automaton of G and H. In $\mathcal{C}$, all transitions are observable, and every path from a secret state x is synchronized with all the possible paths with the same observation starting in the states of $X \cap Q_{NS}$. Thus, if there is a path from the secret state x of length up to k that is not accompanied by a path with the same observation from a state of $X \cap Q_{NS}$, then this path from the state x in $P(G)$ ends up in a state, say, q, whereas all paths in H with the same observation from the state $X \cap Q_{NS}$ end up in the state $\emptyset$. Here, $X \cap Q_{NS}$ and $\emptyset$ are understood as the states of the full observer of G. Thus, if the DES G is not weakly k-step opaque, there is a state of Y from which a state of the from $(q, \emptyset)$ is reachable in at most k steps. Therefore, we search the automaton $\mathcal{C}$ and mark all its states that are reachable from a state of Y in at most k steps. If a state of the from $(q, \emptyset)$ is marked, then G is not weakly k-step opaque; otherwise, it is.

We prove the correctness of Algorithm 2 and analyze its complexity in detail below. Intuitively, the correctness follows from the fact that the BFS visits all nodes at distance d before visiting any nodes at distance $d + 1$. In other words, all states of $\mathcal{C}$ reachable from the states of Y in at most k steps are visited (and marked) before any state at distance $k + 1$. The implementation of the BFS is, however, the key step to obtain the claimed complexity. Namely, the classical BFS of [18] maintains an array to store the shortest distances (aka the number of hops) of every node to an initial node. Since storing a number less than or equal to k requires $\log(k)$ bits, using the classical BFS requires the space of size $O(\log(k)n2^n)$ to store the shortest distance of every state of $\mathcal{C}$ to a state of Y, because $\mathcal{C}$ has $O(n2^n)$ states.

For our purposes, we do not need to know the shortest distance of every state to a state of Y, but we rather need to keep track of the number of hops from the states of Y made so far.

We can achieve this by modifying the classical BFS so that we do not store the shortest distances for every state of $\mathcal{C}$, but only the current distance. We store the current distance in the queue used by the BFS, see Algorithm 3. In particular, we first push number 0 to the queue, followed by all the states

Algorithm 3 The Breadth-First Search used in Algorithm 2

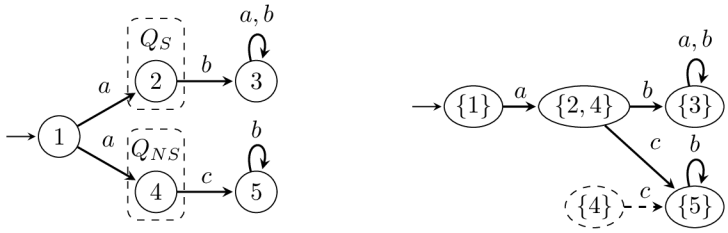
Require: A DES $G = (V, \Sigma, \delta, I)$, a set $S \subseteq V$, $k \in \mathbb{N}_\infty$
Ensure: G with all states at distance at most k from the states of S marked

```
1: Initialize the queue Q := ∅
2: Enqueue number 0 to Q
3: Mark every node s ∈ S and enqueue it to Q
4: Color every node u ∈ V − S white
5: while Q ≠ ∅ do
6:     u := DEQUEUE(Q)
7:     if u ∉ V and u = k then
8:         Terminate, states at distance ≤ k were visited
9:     else if u ∉ V and u < k then
10:        Enqueue u + 1 to Q
11:    else if u ∈ V is a state of G then
12:        for every state v reachable in one step from u do
13:            if the color of v is white then
14:                Mark state v and enqueue it to Q
15:            end if
16:        end for
17:        Color u black
18:    end if
19: end while
```

of Y. Assuming that $k > 0$, number 0 is processed in such a way that it is dequeued from the queue, and number 1 is enqueued. After processing all the states of Y from the queue, that is, having number 1 at the head of the queue, we know that all the elements of the queue after number 1 are the states at distance one from the states of Y and not less. The algorithm proceeds this way until it has either visited all the states of $\mathcal{C}$ or the number stored in the queue is k. The algorithm marks all states of $\mathcal{C}$ that it visits.

This approach requires to store only one $\log(k)$-bit number at a time rather than $n2^n$ such numbers, and hence the complexity of the algorithm then basically follows from the fact that the distance is bounded by the number of states of $\mathcal{C}$, and not by the parameter k.

Since Algorithm 3 is a minor modification of the BFS of Cormen et al. [18], very similar arguments show its correctness and complexity. For this reason,



(a) A DES G. (b) Automata G^{obs} and H.

Figure 6.2: A DES G (a) and the observer G^{obs} (b), the solid part. The automaton H forming the relevant part of the full observer of G is obtained from G^{obs} by adding the dashed part; neither state $\emptyset$ nor the missing transitions to it are depicted in G^{obs} and H.

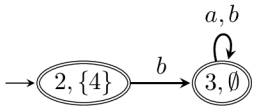


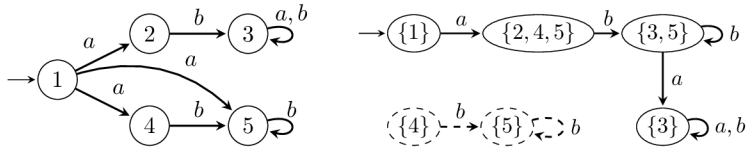
Figure 6.3: The reachable part of $\mathcal{C}_1$, where the single state of Y is denoted by the little arrow.

we do not further discuss the correctness and complexity of Algorithm 3.

Before we prove Theorem 6.6 below showing that G is weakly k-step opaque if and only if no state of the form $(\cdot, \emptyset)$ is marked in $\mathcal{C}$, we illustrate Algorithm 2 in the following two examples.

Example 6.4. We consider weak 1-step opacity of the DES G depicted in Figure 6.2(a) where all events are observable, state 2 is secret, and state 4 is non-secret. The other states are neutral, meaning that they are neither secret nor non-secret. The observer G^{obs} of G is depicted in Figure 6.2(b).

Since G has no unobservable events, the projected automaton $P(G) = G$. Now, only the state $X = \{2, 4\}$ of G^{obs} contains a secret state, and hence intersecting it with Q_S results in the set $Y = \{(2, \{4\})\}$. Notice that state $\{4\}$ is not in the observer G^{obs}, and therefore we need to add it to H together with all the states that are reachable from state $\{4\}$ in the full observer of G. The resulting automaton H is depicted in Figure 6.2(b) and is formed by the observer G^{obs} together with the dashed state $\{4\}$ and the dashed transition from $\{4\}$ to $\{5\}$. Notice that, by the definition of the (full) observer, all the missing transitions in Figure 6.2(b) indeed lead to state $\emptyset$,



(a) The automaton $P(\tilde{G})$. (b) Automata $\tilde{G}^{obs}$ and $\tilde{H}$.

Figure 6.4: The automaton $P(\tilde{G})$ (a) and the observer $\tilde{G}^{obs}$ (b), the solid part. The automaton $\tilde{H}$ forming the relevant part of the full observer of $\tilde{G}$ is obtained from $\tilde{G}^{obs}$ by adding the dashed part; neither state $\emptyset$ nor the missing transitions to it are depicted in G^{obs} and H.

for instance, $\delta(\{1\}, b) = \delta(\{5\}, a) = \emptyset$. However, to keep the figures simple, we do not depict state $\emptyset$ and the transitions to state $\emptyset$. The marked part of the automaton $\mathcal{C}_1 = P(G) \times H$ reachable from the states of Y in at most one step is depicted in Figure 6.3. Since state $(3, \emptyset)$ is marked in $\mathcal{C}_1$, G is not weakly 1-step opaque; indeed, observing the string ab, the intruder reveals that G must have been in the secret state 2 one step ago. ◇

We now illustrate the affirmative case.

Example 6.5. Again, we consider the DES G from Example 6.4, but this time we assume that the event c is unobservable. We denote by $\tilde{G}$ the DES G where events a, b are observable, the event c is unobservable, state 2 is secret, and state 4 is non-secret. The projected automaton $P(\tilde{G})$ and the observer $\tilde{G}^{obs}$ are depicted in Figure 6.4. The only state of $\tilde{G}^{obs}$ containing a secret state is the state $X = \{2, 4, 5\}$, which results in the set $Y = \{(2, \{4\})\}$. Again, state $\{4\}$ is not in $\tilde{G}^{obs}$, and hence we construct the relevant part $\tilde{H}$ of the full observer of $\tilde{G}$ by extending $\tilde{G}^{obs}$ by state $\{4\}$ and all reachable states from it. The result (without state $\emptyset$ and the transitions to state $\emptyset$) is depicted in Figure 6.4(b), both the solid and the dashed part. The marked part of $\mathcal{C}_2 = P(\tilde{G}) \times \tilde{H}$ is depicted in Figure 6.5. Since no state of the form $(\cdot, \emptyset)$ is marked in $\mathcal{C}_2$, $\tilde{G}$ is weakly 1-step opaque. It is worth mentioning that state $(3, \emptyset)$ remains unmarked due to the fact that we need two steps from state $(2, \{4\})$ to reach it. ◇

We now prove the correctness of our algorithm.

Theorem 6.6. *A DES G is weakly k-step opaque with respect to Q_S, Q_{NS}, and P if and only if Algorithm 2 returns* `true`.

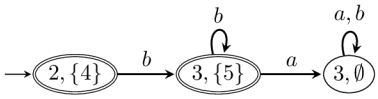


Figure 6.5: The reachable part of $\mathcal{C}_2$, where the single state of Y is denoted by the little arrow.

Proof. If $G = (Q, \Sigma, \delta, I)$ is not weakly k-step opaque, then there is $st \in L(G)$ such that $|P(t)| \leq k$, $\delta(\delta(I, s) \cap Q_S, t) \neq \emptyset$, and $\delta(\delta(I, P^{-1}P(s)) \cap Q_{NS}, P^{-1}P(t)) = \emptyset$. We have two cases.

(i) If $\delta(I, P^{-1}P(s)) \cap Q_{NS} = \emptyset$, then G is not weakly k-step opaque. Algorithm 2 detects this case, because for the state $X = \delta(I, P^{-1}P(s))$ of the observer of G, we have that $X \cap Q_S \supseteq \delta(I, s) \cap Q_S \neq \emptyset$ and $X \cap Q_{NS} = \emptyset$, and hence there is $q \in X \cap Q_S$ resulting in adding the pair $(q, \emptyset)$ to the set Y in line 6.

(ii) If $\delta(I, P^{-1}P(s)) \cap Q_{NS} = Z \neq \emptyset$, then all pairs from $(\delta(I, P^{-1}P(s)) \cap Q_S) \times \{Z\}$ are added to Y. Since $\delta(\delta(I, s) \cap Q_S, t) \neq \emptyset$, there is a pair $(z, Z) \in Y$ such that generating the string $P(t)$ in the automaton $P(G)$ from state z changes the state to a state q. On the other hand, $\delta(Z, P^{-1}P(t)) = \emptyset$ implies that generating $P(t)$ in the full observer of G from state Z changes the state to state $\emptyset$, and hence the pair $(q, \emptyset)$ is reachable in $\mathcal{C}$ from the state $(z, Z) \in Y$ in at most $|P(t)| \leq k$ steps. In both cases, Algorithm 2 marks $(q, \emptyset)$, and returns `false`.

On the other hand, if G is weakly k-step opaque, we show that no pair of the form $(q, \emptyset)$ is reachable in $\mathcal{C}$ from a state of Y in at most k steps. For the sake of contradiction, we assume that a pair $(q, \emptyset)$ is marked in $\mathcal{C}$. However, this means that, in G, there is a string s and a state $z \in Q$ such that $z \in \delta(I, s) \cap Q_S$, the state of the observer of G reached under the string $P(s)$ is $X = \delta(I, P^{-1}P(s))$, and, for $Z = X \cap Q_{NS}$, the pair $(q, \emptyset)$ is reachable in $\mathcal{C}$ from the pair $(z, Z) \in Y$ by a string $w \in \Sigma_o^*$ of length at most k. In particular, there is a string $t \in P^{-1}(w)$ such that when G generates t, it changes its state from z to q. Therefore, $q \in \delta(\delta(I, s) \cap Q_S, t) \neq \emptyset$. However, $\delta(\delta(I, P^{-1}P(s)) \cap Q_{NS}, P^{-1}(w)) = \delta(Z, P^{-1}(w)) = \emptyset$, because generating w in $\mathcal{C}$ changes the pair (z, Z) to $(q, \emptyset)$, and hence the full observer of G changes its state from Z to $\emptyset$ when generating w. This shows that G is not weakly k-step opaque, which is a contradiction. □

We now discuss the complexity of our algorithm.

Theorem 6.7. *The space and time complexity of Algorithm 2 is $O(n2^n)$ and $O((n+m)2^n)$, respectively, where n is the number of states of the input DES G and m is the number of transitions of $P(G)$. In particular, $m \leq \ell n^2$, where ℓ is the number of observable events.*

Proof. Computing the observer and the projected NFA of G, lines 2 and 3, takes time $O(\ell 2^n)$ and $O(m+n)$, respectively. The cycle on lines 4–8 takes time $O(n2^n)$. Constructing the part H of the full observer of G, line 9, takes time $O(\ell 2^n)$. Constructing $\mathcal{C}$, line 10, takes time $O(n2^n+m2^n)$, where $O(n2^n)$ is the number of states and $O(m2^n)$ is the number of transitions of $\mathcal{C}$. The bounds come from the fact that we create at most 2^n copies of the automaton $P(G)$. The BFS takes time linear in the size of $\mathcal{C}$, and the condition of line 11 can be processed during the BFS. Since $m \geq \ell$, the proof is complete. □

We now briefly review the complexity of existing algorithms verifying weak k-step opacity. First, notice that the complexity of existing algorithms is exponential, which seems unavoidable because the problem is PSPACE-complete by Corollary 5.46. In particular, Saboori and Hadjicostis [41] designed an algorithm with complexity $O(\ell(\ell+1)^k 2^n)$, where n is the number of states and ℓ is the number of observable events. Considering the verification of weak ∞-step opacity, Saboori and Hadjicostis [42] designed an algorithm with complexity $O(\ell 2^{n^2+n})$. Yin and Lafortune [52] introduced the notion of a two-way observer and applied it to the verification of weak k-step opacity with complexity $O(\min\{n2^{2n}, n\ell^k 2^n\})$, and to the verification of weak ∞-step opacity with complexity $O(n2^{2n})$; the formulae already include a correction by Lan et al. [33]. In [7] we designed algorithms verifying weak k-step opacity and weak ∞-step opacity with complexities $O((k+1)2^n(n+m\ell^2))$ and $O((n+m\ell)2^n)$, respectively, where $m \leq \ell n^2$ is the number of transitions in the projected automaton. These algorithms outperform the two-way observer if k is polynomial in n or larger than 2^n-2, since weak (2^n-2)-step opacity and weak ∞-step opacity coincide [52]. Wintenberg et al. [49] discussed and experimentally compared four approaches to the verification of weak k-step opacity based on (i) the secret observer, (ii) the reverse comparison, (iii) the state estimator, and (iv) the two-way observer. Their respective state complexities are $O(2^{n(k+3)})$, $O(n(k+1)3^n)$, $O((\ell+1)^k 2^n)$, and $O(\min\{2^n, \ell^k\}2^n)$.[1]

[1]The state complexity of the two-way observer is correct. The correction of Lan et al. [33] consists in adding a time bound to compute the intersection of two sets, and hence it does not influence the number of states.

Notice that these bounds are formulated only in the number of states of the constructed automata, disregarding the number of transitions and the time of the construction. Therefore, the time-complexity bounds differ from the state-complexity bounds at least by the factor of ℓ, if the constructed automata are deterministic, or by a factor of $m \leq \ell n^2$ if the construction of the automaton involves an NFA, such as in the case of the reverse comparison. Namely, the time-complexity bounds are $O(\ell 2^{n(k+3)})$ for the secret observer, where n is the number of states and ℓ is the number of observable events, $O((n+m)(k+1)3^n)$ for the reverse comparison, where $m \leq \ell n^2$ is the number of transitions in an involved NFA, $O(\ell(\ell+1)^k 2^n)$ for the state estimator, and $O(\min\{n2^{2n}, n\ell^k 2^n\})$ for the two-way observer.

As the reader may notice, the above complexities depend on the parameter k. A partial exception is the two-way observer that does not depend on k if $\ell^k \geq 2^n$, that is, if k is larger than the number of states divided by the logarithm of the number of observable events.

Since the complexity of Algorithm 2 is $O((n + m)2^n)$, where n is the number of states of the input DES G and $m \leq \ell n^2$ is the number of transitions of the projected automaton of G, it does not depend on the parameter k and, in general, outperforms the existing algorithms. An exception is the case of a very small parameter k. In particular, if $k < 2\log(n)/\log(\ell)$, the algorithms based on the state estimator and on the two-way observer are, in the worst-case, faster than our algorithm. Notice that this theoretical result agrees with the experimental results of Wintenberg et al. [49].

6.3 Verification of k-SSO

Theorem 5.62 gives us a clue how to verify strong k-step opacity of a given deterministic DES with the help of the verification algorithm for weak k-step opacity from the previous section. Given an instance of strong k-step opacity problem, we first transform it into an instance of weak k-step opacity problem using Transformation 5.60, and then verify the property with Algorithm 2. This idea is formulated as Algorithm 4.

The input of Algorithm 4 is the strong k-step opacity verification problem, which consists of a deterministic DES G, a projection $P\colon \Sigma^* \to \Sigma_o^*$, a set of secret states $Q_S \subseteq Q$, and a parameter $k \in \mathbb{N}_\infty$. We remind that we do not consider neutral states, and therefore $Q_{NS} = Q - Q_S$.

Algorithms verifying strong k-step opacity have been investigated in the

Algorithm 4 Verification of strong k-step opacity

Require: A deterministic DES $G = (Q, \Sigma, \delta, q_0)$, $Q_S \subseteq Q$, $\Sigma_o \subseteq \Sigma$, and $k \in \mathbb{N}_\infty$.

Ensure: `true` if and only if G is strongly k-step opaque with respect to Q_S and $P\colon \Sigma^* \to \Sigma_o^*$

1: Let G_{norm} be the normalization of G by Transformation 5.55
2: Transform G_{norm} to G' by Transformation 5.60
3: Use Algorithm 2 on G' with the set of secret states Q'_S, the set of non-secret states Q'_{NS}, observable events Σ_o, and k
4: **return** the answer of Algorithm 2

literature. In particular, Falcone and Marchand [21] designed an algorithm based on a k-delay trajectory estimation, however, they did not analyze its complexity, and the complexity analyses in the literature are inconsistent. While Ma et al. [35] state that the complexity is $O(\ell 2^{n^2+n})$, where n is the number of states and ℓ is the number of observable events of the verified deterministic DES, Wintenberg et al. [49] state that the state complexity is $O((\ell+1)^k 2^n)$. According to [21, Definition 7], however, the k-delay trajectory estimator has $O(2^{n^{k+1} \cdot 2^k})$ states.

Recently, Ma et al. [35] designed another algorithm with complexity $O(\ell 2^{(k+2)n})$, and even more recently, Wintenberg et al. [49] discussed and experimentally compared algorithms based on (i) the secret observer with complexity $O(\ell(k+3)^n)$, on (ii) the reverse comparison with complexity $O((n+m)(k+1)2^n)$, where $m \leq \ell n^2$ is the number of transitions in the involved projected NFA, and on (iii) the construction of the k-delay trajectory estimator of Falcone and Marchand [21], which they claim to be of complexity $O(\ell(\ell+1)^k 2^n)$.

We now analyze the complexity of Algorithm 4 and show that its worst-case complexity is better than the complexity of existing algorithms. Namely, we show that the space and time complexity of Algorithm 4 is $O(n2^n)$ and $O((n+m)2^n)$, respectively, where n is the number of states of G and m is the number of transitions of $P(G)$. Notice that the complexity does not depend on the parameter k.

Before we prove this result, notice that $m \leq \ell n^2$, where ℓ is the number of observable events. Since ℓn^2 is the maximum number of transitions in an n-state NFA with ℓ events, m is often significantly smaller than ℓn^2.

For a deterministic DES with n states, Transformation 5.55 results in a normalized DES with up to $2n$ states, and hence it may seem that the observer of the normalized DES could have up to 2^{2n} states. The following lemma shows that the observer of the normalized DES has in fact at most 2^n states.

Lemma 6.8. *Let G be an n-state deterministic DES, and let G_{norm} be its normalization obtained by Transformation 5.55. Then, the observer of G_{norm} has at most 2^n states.*

Proof. Let $G = (Q, \Sigma, \delta, q_0)$ be a deterministic DES with n states, and let Σ_{uo} be the set of unobservable events. The application of Transformation 5.55 on G results in the deterministic DES $G_{norm} = (Q_n, \Sigma, \delta_n, q_0)$, where $Q_n \subseteq Q \cup Q'$ and $Q' = \{q' \mid q \in Q\}$ is a disjoint copy of Q. All states of Q_n are reachable in G_{norm} by construction. The observer $G^{obs}_{norm} = (X_{obs}, \Sigma_o, \delta_{obs}, X_0)$ of G_{norm} is defined as follows. The set of states is the subset of the power set of Q_n, namely, $X_{obs} \subseteq 2^{Q_n}$. The initial state is the unobservable reach (UR) of the initial state of the automaton G_{norm}, that is, $X_0 := UR(q_0) = \delta_n(q_0, \Sigma^*_{uo})$. The transition function δ_{obs} is defined for every $X \in X_{obs}$ and every observable event $a \in \Sigma_o$ as the unobservable reach of the states reachable in G_{norm} from the states of X by the event a, that is,

$$\delta_{obs}(X, a) := UR(\delta_n(X, a))$$

where, for every $Y \subseteq Q_n$, $UR(Y) = \delta_n(Y, \Sigma^*_{uo})$. By item (2) of Lemma 5.57,

$$\delta_n(X, a) \subseteq Q\,,$$

and hence every state of the observer of G_{norm} is uniquely determined by a subset of Q. In particular, we define an injective mapping $f\colon X_{obs} \to 2^Q$ assigning subsets of Q to the states of the observer of G_{norm} as follows: $f(X_0) = \{q_0\}$, and for every state $Y \neq X_0$ of the observer of G_{norm}, we pick and fix a state $X \in X_{obs}$ such that $\delta_{obs}(X, a) = Y$, for some observable event $a \in \Sigma_o$, and we define $f(Y) = \delta_n(X, a)$. Such a state X exists because every state of the observer is reachable. Then, $Y = \delta_{obs}(X, a) = UR(\delta_n(X, a)) = UR(f(Y))$, and we have that if $f(Y_1) = f(Y_2)$, then $Y_1 = UR(f(Y_1)) = UR(f(Y_2)) = Y_2$, which shows that the mapping f is injective. Consequently, the number of states of the observer of G_{norm} is bounded by the number of subsets of the set Q, which is 2^n. □

Notice that Lemma 6.8 does not claim that the number of states of the observer of G and of the observer of its normalization G_{norm} coincide. It only provides an upper bound on the worst-case complexity.

Similarly, for a normal deterministic DES G with n states, Transformation 5.60 results in a deterministic DES, denoted by G', with up to $2n$ states. The second lemma shows that the observer of G' has as many states as the observer of G.

Lemma 6.9. *Let G be a normal deterministic DES with n states, and let G' be obtained from G by Transformation 5.60. Then, the numbers of states of the observer of G' and of the observer of G coincide.*

Proof. Let $G = (Q, \Sigma, \delta, q_0)$ be a normal deterministic DES, and let $G' = (Q \cup Q'_{NS}, \Sigma \cup \{u\}, \delta', q_0)$ be the DES obtained from G by Transformation 5.60. Recall that G' is obtained as a disjoint union of G and G_{ns}, where G_{ns} is a copy of G without the secret states and the corresponding transitions, $Q'_{NS} = \{q' \mid q \in Q_{NS}\}$ is a copy of Q_{NS} disjoint from Q, and the event u is unobservable. For every reachable state S of the observer of G', we show that S contains a state $p' \in Q'_{NS}$ if and only if S contains the corresponding state $p \in Q_{NS}$. Consequently, the observer of G' and the observer of G have the same number of states.

To prove one direction, let S be a reachable state of the observer of G'. If S contains a state $p \in Q_{NS}$, then the unobservable transition (p, u, p') of G' implies that S also contains the state $p' \in Q'_{NS}$.

To prove the other direction, let S be a reachable state of the observer of G', and assume that a state $p' \in Q'_{NS}$ belongs to S. Then, for every string $w \in P'(L(G'))$ under which the state S is reachable from the initial state $\{q_0\}$ in the observer of G', there exists a string $w' \in L(G')$ such that $P'(w') = w$ and $\delta'(q_0, w') = p'$. Since $q_0 \in Q$, $p' \in Q'_{NS}$, and every string of $L(G')$ contains at most one occurrence of the event u, we can partition the string $w' = w_1 u w_2$ so that $\delta'(q_0, w_1) = r$, $\delta'(r, u) = r'$, and $\delta'(r', w_2) = p'$, for some state $r \in Q$. However, $\delta'(r', w_2) = p'$ is executed in G_{ns}, which is obtained from G by removing all secret states. Therefore, $\delta(r, w_2) = p$ must be defined in G. Altogether, we have shown that $\delta(q_0, w_1 w_2) = p$ is defined in G, and hence the string $w_1 w_2 \in L(G)$. Since $w = P'(w') = P'(w_1 w_2)$, we have shown that $p \in S$. □

We can now prove the following result analyzing the complexity of Algorithm 4.

Theorem 6.10. *The space and time complexity of Algorithm 4 is $O(n2^n)$ and $O((n+m)2^n)$, respectively, where n is the number of states of G and m is the number of transitions of $P(G)$, that is, $m \leq \ell n^2$, where ℓ is the number of observable events.*

Proof. Let G be an n-state deterministic DES. In the first step, we construct the normalization G_{norm} of G with at most $2n$ states, the observer of which has at most 2^n states by Lemma 6.8. Then, we apply Algorithm 2 to G' obtained from G_{norm} by Transformation 5.60. In particular, by Lemma 6.9, we compute the observer of G' with at most 2^n states, and the projected automaton $P(G')$ with at most $4n$ states. Then, for every reachable state X of G'^{obs}, and for every $x \in X \cap Q'_S$, we add the pair $(x, X \cap Q'_{NS})$ to the set Y. This cycle takes time $O(n2^n)$. Afterwards, we construct the automaton H as the part of the full observer of G' accessible from the states of the second components of Y. Since H consists only of the subsets of Q'_{NS}, of which there is at most 2^n, the automaton H has $O(2^n)$ states. The automaton $\mathcal{C} = P(G') \times H$ thus has $O(n2^n)$ states and $O(m2^n)$ transitions, the sum of which is the time complexity of the BFS applied to mark states of $\mathcal{C}$ reachable from the states of Y in at most k steps. Therefore, the state complexity of Algorithm 4 is $O(n2^n)$ and the time complexity is $O(n2^n + (n+m)2^n) = O((n+m)2^n)$. □

Comparing the complexity $O((n+m)2^n)$ of Algorithm 4 with the complexity of the existing algorithms, the reader may see that (1) the complexity of Algorithm 4 does not depend on the parameter k, and (2) it is better than the complexity of the existing algorithms, because the minimum of the worst-case complexities $O(\ell 2^{n^{k+1} \cdot 2^k})$, $O(\ell 2^{(k+2)n})$, $O(\ell(k+3)^n)$, and $O((n+m)(k+1)2^n)$ of the existing algorithms discussed at the beginning of this subsection is $O((n+m)2^n)$ for $k = 1$, and $O((n+m)(k+1)2^n) = O((n+m)2^{2n})$ for $k \in O(2^n)$. Notice that the minimum worst-case complexity for large k is significantly higher than the complexity $O((n+m)2^n)$ of Algorithm 4. In fact, the complexity of Algorithm 4, and the minimum worst-case complexity of the existing algorithms for very small k, coincide. However, while the existing algorithms can handle only inputs with a very small k with this complexity, our algorithm can handle inputs with k of arbitrary value with this complexity. Consequently, our algorithm improves the complexity of the verification of strong k-step opacity.

Chapter 7

Conclusions

In this thesis, we presented new results in three areas concerning opacity of discrete-event systems modeled by automata: the complexity of deciding opacity, the design of verification algorithms, and the relationships among various notions of opacity. We thus provided a complete and improved complexity picture of verifying the discussed notions of opacity.

In Chapter 4, we study the properties of current-state opacity in systems with a restricted alphabet and a restricted structure. We showed that the problem of deciding current-state opacity remains hard for almost all practial cases, cf. Table 1.1. Most notably, we showed that current-state opacity is:

1. PSpace-complete for systems modeled by DFAs/poDFAs with three events, one of which is unobservable (Theorem 4.7 and Corollary 4.15),
2. coNP-complete for systems modeled by NFAs/DFAs with a single observable event (Theorem 4.8 and Corollary 4.9), and
3. coNP-complete for systems modeled by acyclic NFAs/acyclic DFAs with two observable events (Theorem 4.10 and Corollary 4.12).

Chapter 5 is dedicated to transformations among the considered opacity notions. Our transformations are computable in polynomial time and preserve the number of observable events and determinism (whenever it is meaningful), allowing us to derive new results for corresponding opacity notions, see Table 1.2 for an overview. Below we summarize the results obtained from the transformations.

- **Language-based opacity** and **initial-and-final-state opacity** – By combining Theorems 4.7 and 4.8 with transformations of Wu and Lafor-

tune [50], we can conclude that deciding LBO and IFO is PSPACE-complete for systems with two or more observable events, and coNP-complete for systems with a single observable event.

- **Initial-state opacity** – We show that deciding ISO is PSPACE-complete for systems with two or more observable events. This result is established through Transformation 5.3 (hardness) and the membership result of Saboori [37]. Additionally, Theorem 5.8 shows that deciding ISO is NL-complete in the single observable event case.
- **Trace opacity** – We show that deciding TO is PSPACE-complete for systems with two or more observable events. This result is established through Transformation 5.9 (hardness) and the membership result of Dubreil [19]. Additionally, Theorem 5.14 shows that deciding TO is NL-complete in the single observable event case.
- **Weak k-step opacity** – We show that deciding k-SO is PSPACE-complete for systems with two or more observable events and the parameter $k \in \mathbb{N}_\infty$ encoded in binary. This result is established through Transformations 5.19 (hardness) and 5.43 (membership). In the single observable event case, deciding k-SO is coNP-complete by Transformations 5.30 (hardness) and 5.47 (membership).
- **Strong k-step opacity** – We show that deciding k-SSO is PSPACE-complete for systems with two or more observable events and the parameter $k \in \mathbb{N}_\infty$ encoded in binary. This result is established through Transformations 5.19 (hardness) and 5.60 (membership). In the single observable event case, deciding k-SSO is coNP-complete by Transformations 5.25 (hardness) and 5.60 (membership). Additionally, Theorem 5.59 describes the relationship of 0-SO and 0-SSO.

In Chapter 6, we propose three algorithms for verifying language-based opacity and trace opacity (Algorithm 1), weak k-step opacity (Algorithm 2), and strong k-step opacity (Algorithm 4). We provide an analysis of all mentioned algorithms and we show that their time complexity is $O((n+m)2^n)$, where n stands for the number of states of the input automaton and m for the number of transitions in the projected automaton of the input automaton. In particular, the complexity of algorithms for verifying weak and strong k-step opacity does not depend on the parameter $k \in \mathbb{N}_\infty$. However, it remains an open question how our algorithms would perform if tested experimentally.

Bibliography

[1] R. Alur, P. Černý, and S. Zdancewic. Preserving secrecy under refinement. In *Automata, Languages and Programming*, pages 107–118. Springer, 2006.

[2] S. Arora and B. Barak. *Computational Complexity – A Modern Approach*. Cambridge University Press, 2009.

[3] P. R. Asveld and A. Nijholt. The inclusion problem for some subclasses of context-free languages. *Theoretical Computer Science*, 230(1-2):247–256, 2000.

[4] E. Badouel, M. Bednarczyk, A. Borzyszkowski, B. Caillaud, and P. Darondeau. Concurrent secrets. *Discrete Event Dynamic Systems*, 17:425–446, 2007.

[5] J. Balun and T. Masopust. On opacity verification for discrete-event systems. *IFAC-PapersOnLine*, 53(2):2075–2080, 2020.

[6] J. Balun and T. Masopust. On verification of strong periodic D-detectability for discrete event systems. *IFAC-PapersOnLine*, 53(4):263–268, 2020. 15th IFAC Workshop on Discrete Event Systems WODES 2020.

[7] J. Balun and T. Masopust. Comparing the notions of opacity for discrete-event systems. *Discrete Event Dynamic Systems*, 31:553–582, 2021.

[8] J. Balun and T. Masopust. On verification of D-detectability for discrete event systems. *Automatica*, 133:109884, 2021.

[9] J. Balun and T. Masopust. On transformations among opacity notions. *2022 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 3012–3017, 2022.

[10] J. Balun and T. Masopust. On verification of weak and strong k-step opacity for discrete-event systems. *IFAC-PapersOnLine*, 55(28):108–113, 2022. 16th IFAC Workshop on Discrete Event Systems WODES 2022.

[11] N. BenHadj-Alouane, S. Lafrance, F. Lin, J. Mullins, and M. Yeddes. On the verification of intransitive noninterference in mulitlevel security. *IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics)*, 35(5):948–958, 2005.

[12] J. W. Bryans, M. Koutny, L. Mazaré, and P. Y. A. Ryan. Opacity generalised to transition systems. *International Journal of Information Security*, 7(6):421–435, 2008.

[13] J. W. Bryans, M. Koutny, and P. Y. Ryan. Modelling opacity using Petri nets. *Electronic Notes in Theoretical Computer Science*, 121:101–115, 2005.

[14] J. A. Brzozowski and F. E. Fich. Languages of R-trivial monoids. *Journal of Computer and System Sciences*, 20(1):32–49, 1980.

[15] C. G. Cassandras and S. Lafortune, editors. *Introduction to Discrete Event Systems*. Springer, third edition, 2021.

[16] F. Cassez. The dark side of timed opacity. In *Advances in Information Security and Assurance*, volume 5576, pages 21–30. Springer, 2009.

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

[18] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms*. MIT Press, 2009.

[19] J. Dubreil. Monitoring and supervisory control for opacity properties. 11 2009.

[20] J. Dubreil, P. Darondeau, and H. Marchand. Opacity enforcing control synthesis. In *WODES*, pages 28–35, 2008.

[21] Y. Falcone and H. Marchand. Enforcement and validation (at runtime) of various notions of opacity. *Discrete Event Dynamic Systems*, 25:531–570, 2014.

[22] R. Focardi and R. Gorrieri. A taxonomy of trace-based security properties for CCS. In *The Computer Security Foundations Workshop VII*, pages 126–136. IEEE Comput. Soc. Press, 1994.

[23] M. Holzer and M. Kutrib. Descriptional and computational complexity of finite automata—A survey. *Information and Computation*, 209(3):456–470, 2011.

[24] J. E. Hopcroft, R. Motwani, and J. D. Ullman. *Introduction to Automata Theory, Languages, and Computation.* Addison-Wesley, 2006.

[25] H. B. Hunt III. *On the Time and Tape Complexity of Languages.* PhD thesis, Cornell University, Ithaca, NY, 1973.

[26] N. Immerman. Nondeterministic space is closed under complementation. *SIAM Journal on Computing*, 17:935–938, 1988.

[27] R. Jacob, J.-J. Lesage, and J.-M. Faure. Overview of discrete event systems opacity: Models, validation, and quantification. *Annual Reviews in Control*, 41:135–146, 2016.

[28] G. Jirásková and T. Masopust. On a structural property in the state complexity of projected regular languages. *Theoretical Computer Science*, 449:93–105, 2012.

[29] N. D. Jones. Space-bounded reducibility among combinatorial problems. *Journal of Computer and System Sciences*, 11(1):68–85, 1975.

[30] C. Keroglou and C. N. Hadjicostis. Initial state opacity in stochastic des. In *2013 IEEE 18th Conference on Emerging Technologies & Factory Automation (ETFA)*, pages 1–8, 2013.

[31] J. Komenda, D. Zorzenon, and J. Balun. Modeling of safe timed petri nets by two-level (max,+) automata. *IFAC-PapersOnLine*, 55(28):212–219, 2022. 16th IFAC Workshop on Discrete Event Systems WODES 2022.

[32] M. Krötzsch, T. Masopust, and M. Thomazo. Complexity of universality and related problems for partially ordered NFAs. *Information and Computation*, 255(1):177–192, 2017.

[33] H. Lan, Y. Tong, J. Guo, and A. Giua. Comments on "A new approach for the verification of infinite-step and K-step opacity using two-way observers" [Automatica 80 (2017) 162–171]. *Automatica*, 122:109290, 2020.

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

[35] Z. Ma, X. Yin, and Z. Li. Verification and enforcement of strong infinite- and k-step opacity using state recognizers. *Automatica*, 133:109838, 2021.

[36] L. Mazaré. Decidability of opacity with non-atomic keys. In *Formal Aspects in Security and Trust*, pages 71–84. Springer, 2004.

[37] A. Saboori. *Verification and enforcement of state-based notions of opacity in discrete event systems*. PhD thesis, University of Illinois at Urbana-Champaign, 2011.

[38] A. Saboori and C. N. Hadjicostis. Notions of security and opacity in discrete event systems. In *IEEE CDC*, pages 5056–5061, 2007.

[39] A. Saboori and C. N. Hadjicostis. Opacity-enforcing supervisory strategies for secure discrete event systems. In *Conference on Decision and Control*. IEEE, 2008.

[40] A. Saboori and C. N. Hadjicostis. Coverage analysis of mobile agent trajectory via state-based opacity formulations. *Control Engineering Practice*, 19(9):967–977, 2011. Special Section: DCDS'09 – The 2nd IFAC Workshop on Dependable Control of Discrete Systems.

[41] A. Saboori and C. N. Hadjicostis. Verification of K-step opacity and analysis of its complexity. *IEEE Transactions on Automation Science and Engineering*, 8(3):549–559, 2011.

[42] A. Saboori and C. N. Hadjicostis. Verification of infinite-step opacity and complexity considerations. *IEEE Transactions on Automatic Control*, 57(5):1265–1269, 2012.

[43] S. Schneider and A. Sidiropoulos. CSP and anonymity. In *Computer Security — ESORICS 96*, pages 198–218. Springer, 1996.

[44] T. Schwentick, D. Thérien, and H. Vollmer. Partially-ordered two-way automata: A new characterization of DA. In *Developments in Language Theory (DLT)*, volume 2295 of *LNCS*, pages 239–250, 2001.

[45] N. J. A. Sloan. The on-line encyclopedia of integer sequences (OEIS). A123121.

[46] N. J. A. Sloan. The on-line encyclopedia of integer sequences (OEIS). A001511.

[47] L. J. Stockmeyer and A. R. Meyer. Word problems requiring exponential time: Preliminary report. In *ACM Symposium on Theory of Computing (STOC)*, pages 1–9. ACM Press, 1973.

[48] R. Szelepcsényi. The method of forced enumeration for nondeterministic automata. *Acta Informatica*, 26:279–284, 1988.

[49] A. Wintenberg, M. Blischke, S. Lafortune, and N. Ozay. A general language-based framework for specifying and verifying notions of opacity. *Discrete Event Dynamic Systems*, 32:253–289, 2022.

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

[51] Y.-C. Wu, K. A. Sankararaman, and S. Lafortune. Ensuring privacy in location-based services: An approach based on opacity enforcement. *IFAC Proceedings Volumes*, 47(2):33–38, 2014. 12th IFAC International Workshop on Discrete Event Systems (2014).

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

[53] D. Zorzenon, J. Balun, and J. Raisch. Weak consistency of P-time event graphs. *IFAC-PapersOnLine*, 55(40):19–24, 2022. 1st IFAC Workshop on Control of Complex Systems COSY 2022.

Opacity of discrete event systems: transformations and algorithms

Jiří Balun

Author Paper of Dissertation Thesis

Department of Computer Science
Faculty of Science
Palacký University Olomouc
2023

Uchazeč
Mgr. Jiří Balun
jiri.balun@gmail.com

Školitel
doc. RNDr. Tomáš Masopust, Ph.D., DSc.

Místo a termín obhajoby

Oponenti

S dizertační prací a posudky se bude možné seznámit na katedře informatiky PřF UP, 17.listopadu 12, 779 00 Olomouc.

Abstrakt – Opacity is a security property of discrete-event systems that asks whether, at any point of a computation, the secret is revealed to a passive intruder. The literature has introduced several notions of opacity, including language-based opacity, trace opacity, current-state opacity, weak k-step opacity, weak ∞-step opacity, strong k-step opacity, initial-state opacity, and initial-and-final-state opacity. In this work, we provide a complete and improved complexity picture of verifying the discussed opacity notions within the finite automata model. First, we focus on the complexity of deciding current-state opacity in systems with a restricted set of events and a restricted structure. Second, we present polynomial-time transformations among the notions that preserve determinism and the number of observable events, allowing the generalization of results across different notions of opacity. Third, we propose three new algorithms for verifying language-based opacity, trace opacity, weak k-step opacity, weak ∞-step opacity, and strong k-step opacity that improve their respective algorithmic complexity.

Keywords: Discrete event system, finite automaton, opacity, transformation, complexity, algorithm, verification

Preface

The focus of this paper is on opacity of discrete-event systems, examining three key areas: the complexity of deciding opacity, the design of verification algorithms, and the relationships among various notions of opacity. The results presented in this paper are mostly based on outcomes of the joint scientific work with Tomáš Masopust, which were published in the following articles.

[4] J. Balun and T. Masopust. On opacity verification for discrete-event systems. *IFAC-PapersOnLine*, 53(2):2075–2080, 2020.

[6] J. Balun and T. Masopust. Comparing the notions of opacity for discrete-event systems. *Discrete Event Dynamic Systems*, 31:553–582, 2021.

[8] J. Balun and T. Masopust. On transformations among opacity notions. *2022 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 3012–3017, 2022.

[9] J. Balun and T. Masopust. On verification of weak and strong k-step opacity for discrete-event systems. *IFAC-PapersOnLine*, 55(28):108–113, 2022. 16th IFAC Workshop on Discrete Event Systems WODES 2022.

In [4], we mainly focus on the complexity of deciding current-state opacity in systems with a restricted set of events and a restricted structure. Most of the results from this paper are presented in Chapter 4.

In [6], we introduce transformations between weak k-step opacity and current-state opacity, and between language-based opacity and initial-state opacity. Selected transformations from this article are presented in Sections 5.1, 5.5, and 5.6. We also design new algorithms for verifying language-based opacity, weak k-step opacity and weak ∞-step opacity, the first of which is presented in Section 6.1.

In [8], we have further improved the previously presented transformations from weak k-step opacity to current-state opacity, which were initially introduced in [6]. The updated transformations are polynomial in terms of the parameter k. These transformations, along with others from this paper, are discussed in Sections 5.4 and 5.6. An extended version of this paper, under review in *Discrete Event Dynamic Systems* at the time of writing the thesis, is available on `arXiv`.

In [9], we design a transformation from strong k-step opacity to weak k-step opacity, as well as algorithms to verify both strong and weak k-step opacity. As the algorithm for verifying weak k-step opacity is an updated version of the algorithm presented in [6], I have included only this improved variant in this paper. The transformation can be found in Section 5.7, while algorithms are presented in Sections 6.2 and 6.3. An extended version of this paper, accepted for publication in *Automatica* at the time of writing the thesis, is available on `arXiv`.

Furthermore, some of the transformations from Sections 5.2, 5.3, and 5.4 are not yet included in any article. I decided to include them in the thesis to give a complete picture of the transformations among the discussed notions.

In addition to the articles listed above, I have made contributions to the following publications:

[5] J. Balun and T. Masopust. On verification of strong periodic D-detect-ability for discrete event systems. *IFAC-PapersOnLine*, 53(4):263–268, 2020. 15th IFAC Workshop on Discrete Event Systems WODES 2020.

[7] J. Balun and T. Masopust. On verification of D-detectability for discrete event systems. *Automatica*, 133:109884, 2021.

[26] J. Komenda, D. Zorzenon, and J. Balun. Modeling of safe timed petri nets by two-level (max,+) automata. *IFAC-PapersOnLine*, 55(28):212–219, 2022. 16th IFAC Workshop on Discrete Event Systems WODES 2022.

[45] D. Zorzenon, J. Balun, and J. Raisch. Weak consistency of P-time event graphs. *IFAC-PapersOnLine*, 55(40):19–24, 2022. 1st IFAC Workshop on Control of Complex Systems COSY 2022.

No results from these articles have been included in the thesis as they do not focus on opacity and due to space reasons.

The thesis was supported by grants:

- INTER-EXCELLENCE project LTAUSA19098 of Ministry of Education, Youth and Sports, and
- IGA PrF 2020 019, IGA PrF 2021 022, IGA PrF 2022 018, and IGA PrF 2023 026 of Palacký University Olomouc.

Chapter 1

Introduction

With the development of digitalization, the security is becoming an increasingly important topic. Since many properties of the systems can be deduced from their discrete abstraction, several cybersecurity notions have been introduced for the discrete-event systems. Namely, such properties include *anonymity* of Schneider and Sidiropoulos [38], *noninterference* of Hadj-Alouane et al. [10], *secrecy* of Alur et al. [1], *security* of Focardi and Gorrieri [21], and *opacity* of Mazaré [31].

This paper focuses on the opacity property, which guarantees that a system prevents an intruder from revealing its secret. In the opacity setting, the intruder is a passive observer that knows the structure of the system but has only limited capability to observe its behavior. Therefore, if the intruder wants to reveal the secret, he must estimate the current state of the system based on his observations. Intuitively, the system is opaque if for every secret behavior, there is a nonsecret behavior that looks the same to the intruder. Therefore, at no point during the computation can the intruder be certain whether or not secret behavior has occurred. The secret itself is usually modeled as either a set of secret behaviors or a set of secret states. The former option leads to *language-based opacity*, while the latter leads to *state-based opacity*. Several notions of language-based and state-based opacity have been discussed in the literature, from which we selected, in our opinion, the most important and practical ones.

Defining the secret as a behavior results in two notions, that is, language-based opacity (LBO) and trace opacity (TO). In the case of language-based opacity, which was introduced by Badouel et al. [3] and Dubreil et al. [19], the secret is defined as a subset of system's behavior. This subset is known as a secret language and it contains compromising sequences of the system. Such a sequence may, for example, represent the initiation of a system reboot. Trace opacity, as introduced by Bryans et al. [11], is a special case of language-based opacity. In trace opacity, the secret language comprises only those behaviors that contain one of the secret events that represent the occurence of some critical action of the system.

Regarding state-based opacity, we consider the following six notions: current-state opacity (CSO), weak k-step opacity (k-SO), weak ∞-step opacity (∞-SO), strong k-step opacity (k-SSO), initial-state opacity (ISO), and initial-and-final-state opacity (IFO). In this case, each secret state represents a vulnerable condition of the system, such as a particular part of the system undergoing maintenance.

The most basic state-based notion is current-state opacity of Bryans et al. [12] that prevents the intruder from revealing whether the system is currently in a secret state. However, in the future, the intruder may realize that the system was in a secret state

Model	$\lvert\Sigma_o\rvert = 1$	$\lvert\Sigma_o\rvert \geq 2$
NFA	coNP-complete	PSpace-complete [16]
DFA	coNP-complete	PSpace-complete
partially ordered NFA	NL-complete	PSpace-complete
partially ordered DFA	NL-complete	PSpace-complete
acyclic NFA	NL-complete	coNP-complete
acyclic DFA	NL-complete	coNP-complete

Table 1.1: Complexity of verifying current-state opacity for different models with Σ_o being the set of observable events.

at some earlier point of the computation. For example, if the intruder estimates that the system could be in one of two possible states, and then in the following step, the system proceeds via an observable event that is only possible from one of those states, the intruder can deduce the state in which the system was one step ago. This issue has been considered in the literature and led to the introduction of weak k-step opacity and weak ∞-step opacity by Saboori and Hadjicostis [33, 37]. While weak k-step opacity requires that the intruder cannot ascertain the secret in the current state and k subsequent observable steps, weak ∞-step opacity requires that the intruder can never ascertain that the system was in a secret state. Note that weak 0-step opacity coincides with current-state opacity by definition, and that an n-state automaton is weakly ∞-step opaque if and only if it is weakly $(2^n - 2)$-step opaque [44].

Falcone and Marchand [20] have suggested that weak k-step opacity is not as secure as it may seem. Although it may seem sufficiently confidential, the intruder can still deduce that the system was previously in a secret state, even if the intruder cannot determine the exact time at which the system entered that state. To address this issue, they introduced a stronger version of k-step opacity called strong k-step opacity, which provides a higher level of confidentiality.

Bryans et al. [12] introduced initial-state opacity, which prevents the intruder from revealing, at any time instant, whether the system started in a secret state. Initial-and-final-state opacity of Wu and Lafortune [42] is a generalization of both current-state opacity and initial-state opacity, where the secret is represented as a pair of an initial and a marked state. Therefore, the intruder can never reveal both starting and ending point of the computation at the same time.

This paper focuses solely on the theoretical aspects of opacity. However, there have been successful implementations of opacity in various applications, such as concealment of vehicle positions by Saboori and Hadjicostis [35], and ensuring privacy of location-based services by Wu et al. [43]. For a comprehensive overview of opacity and its applications, we recommend the reader the work of Jacob et al. [23].

Most of the mentioned notions have been studied within the framework of many different models, such as finite automata [33], Petri nets [12], timed automata [15], and stochastic automata [25]. In this paper, we model the system as a finite automaton with partially observable behavior. In some cases, we also consider structurally simpler variants such as partially ordered automata or acyclic automata. In Chapter 2, we introduce relevant concepts of automata theory and we formalize the model itself. Chapter 3 provides an overview of all the opacity notions considered in this work.

One of the key areas in opacity research is the complexity of deciding whether a system satisfies a given notion of opacity. Since the verification is often based on the observer

Notion	$\lvert\Sigma_o\rvert = 1$	$\lvert\Sigma_o\rvert \geq 2$	Order
LBO	coNP-complete	PSPACE-complete	$O((n+m)2^n)$
TO	NL-complete	PSPACE-complete	$O((n+m)2^n)$
CSO	coNP-complete	PSPACE-complete	$O(\ell 2^n)$ [32]
k-SO	coNP-complete	PSPACE-complete	$O((n+m)2^n)$
∞-SO	coNP-complete	PSPACE-complete	$O((n+m)2^n)$
k-SSO	coNP-complete	PSPACE-complete	$O((n+m)2^n)$
ISO	NL-complete	PSPACE-complete	$O(\ell 2^n)$ [42]
IFO	coNP-complete	PSPACE-complete	$O(\ell 2^{n^2})$ [42]

Table 1.2: Complexity of verifying the notions of opacity for DESs following from the transformations, algorithms, and known results; Σ_o stands for the set of observable events, n for the number of states of the input automaton, ℓ for the number of observable events of the input automaton, and $m \leq \ell n^2$ for the number of transitions in the projected automaton of the input automaton.

construction, the problem belongs to PSPACE. In fact, most of the notions are PSPACE-complete in the general case, and thus there is no polynomial-time verification algorithm unless P = PSPACE. This raises the question of whether the problem is easier to solve if we somehow restrict the structure of the system. Therefore, in Chapter 4, we investigate the problem of deciding current-state opacity for systems that have a limited number of observable events and that are represented by partially ordered or acyclic automata. However, despite these restrictions, the problem remains hard in almost all practical cases, as indicated in Table 1.1, where we summarize our findings and existing results.

Transformations are another useful tool for analysing the complexity of decision problems. If we can, for example, transform an instance of the current-state opacity problem to an instance of the language-based opacity problem in polynomial time and vice versa, we can derive PSPACE-completeness of language-based opacity from the PSPACE-completeness of current-state opacity. Such transformations were first provided by Wu and Lafortune [42] between language-based opacity, current-state opacity, initial-state opacity, and initial-and-final-state opacity. In Chapter 5, we extend their results and provide transformations for trace opacity, weak k-step opacity, and strong k-step opacity. Thus, by combining these transformations, we show how to transform between any two notions, allowing the generalization of results across different notions of opacity. In particular, we show that for systems with two or more observable events, the decision problem of any of the considered notions is PSPACE-complete. On the other hand, if the system has only one observable event, then the problem is coNP-complete for all notions, except for initial-state opacity and trace opacity, which are NL-complete. We summarize results following from transformations, together with the existing results, in Table 1.2.

In addition to the new complexity results, the transformations also enabled us to design three new algorihms, which we introduce in Chapter 6. Through the analysis of existing algorithms [29, 42, 36, 37, 44, 20, 30, 41], we demonstrate that our algorithms improve the algorithmic complexity of verifying language-based opacity, trace opacity, weak k-step opacity, weak ∞-step opacity, and strong k-step opacity. The right-most column of Table 1.2 provides a summary of the complexities of the best-known algorithms for all of the discussed notions. Note that we have not compared the algorithms experimentally, and therefore in practical cases our algorithms might be outperformed.

Chapter 2

Preliminaries

In this chapter, we formalize the notation and model of a discrete-event system based on finite automata. For more details on these topics see [14].

For a set S, $|S|$ denotes the cardinality of S, and 2^S denotes the power set of S. We define $\mathbb{N}$ to be the set of all non-negative integers, and we extend it with its limit to $\mathbb{N}_\infty = \mathbb{N} \cup \{\infty\}$.

2.1 Languages and automata

An alphabet Σ is a finite nonempty set of events. A string over Σ is a sequence of events from Σ; the empty string is denoted by ε. The set of all finite strings over Σ is denoted by Σ^*. A language L over Σ is a subset of Σ^*. The set of prefixes of strings of L is the set $\overline{L} = \{u \mid \exists v \in \Sigma^*,\, uv \in L\}$. For a string $u \in \Sigma^*$, $|u|$ denotes the length of u, and $\overline{u}$ denotes the set of all prefixes of u.

Definition 2.1. A *nondeterministic finite automaton* (NFA) over an alphabet Σ is a structure $\mathcal{A} = (Q, \Sigma, \delta, I, F)$, where Q is a finite set of states, $\delta\colon Q \times \Sigma \to 2^Q$ is a transition function, $I \subseteq Q$ is a set of initial states, and $F \subseteq Q$ is a set of marked states.

The transition function can be extended to the domain $2^Q \times \Sigma^*$ by induction. Equivalently, the transition function is a relation $\delta \subseteq Q \times \Sigma \times Q$, where, e.g., $\delta(q, a) = \{s, t\}$ denotes two transitions (q, a, s) and (q, a, t).

For a set $Q_0 \subseteq Q$, the set $L_m(\mathcal{A}, Q_0) = \{w \in \Sigma^* \mid \delta(Q_0, w) \cap F \neq \emptyset\}$ is the language marked by $\mathcal{A}$ from the states of Q_0, and $L(\mathcal{A}, Q_0) = \{w \in \Sigma^* \mid \delta(Q_0, w) \neq \emptyset\}$ is the language generated by $\mathcal{A}$ from the states of Q_0. The languages *marked* and *generated* by $\mathcal{A}$ are defined as $L_m(\mathcal{A}) = L_m(\mathcal{A}, I)$ and $L(\mathcal{A}) = L(\mathcal{A}, I)$, respectively. If $\overline{L_m(\mathcal{A})} = L(\mathcal{A})$ holds, then $\mathcal{A}$ is *non-blocking* and every string generated by $\mathcal{A}$ can be extended to a marked string.

The NFA $\mathcal{A}$ is *deterministic* (DFA) if $|I| = 1$ and $|\delta(q, a)| \leq 1$ for every $q \in Q$ and $a \in \Sigma$. In this case, we identify the singletons with their elements, and simply write $\mathcal{A} = (Q, \Sigma, \delta, q_0, F)$ if $I = \{q_0\}$ and $\delta(q, a) = q'$ instead of $\delta(q, a) = \{q'\}$.

Let $\leq$ be the reachability relation on the state set Q defined as $p \leq q$ if there is $w \in \Sigma^*$ such that $q \in \delta(p, w)$. Then, the NFA $\mathcal{A}$ is *partially ordered* (poNFA) if its reachability relation $\leq$ is a partial order. If $\mathcal{A}$ is a partially ordered DFA, we use the notation *poDFA*. The automaton is *acyclic*, if $q \notin \delta(q, w)$ for every $q \in Q$ and $w \in \Sigma^* - \{\varepsilon\}$.

Let $\mathcal{A}_i = (Q_i, \Sigma_i, \delta_i, I_i, F_i)$, where $i \in \{1, 2\}$, be two NFAs. For $\mathcal{A}_1$ and $\mathcal{A}_2$ over common alphabet $\Sigma = \Sigma_1 = \Sigma_2$, the *product automaton* of $\mathcal{A}_1$ and $\mathcal{A}_2$ is defined as the

automaton $\mathcal{A}_1 \times \mathcal{A}_2 = (Q_1 \times Q_2, \Sigma, \delta, I_1 \times I_2, F_1 \times F_2)$, where $\delta((q_1, q_2), a) = \delta_1(q_1, a) \times \delta_2(q_2, a)$ for every pair of states $(q_1, q_2) \in Q_1 \times Q_2$ and every event $a \in \Sigma$. Notice that the definition does not restrict the state space of the product automaton to its reachable part. In case where $\Sigma_1 \neq \Sigma_2$, we use the *synchronous product* of $\mathcal{A}_1$ and $\mathcal{A}_2$, which is defined as the automaton $\mathcal{A}_1 \parallel \mathcal{A}_2 = (Q_1 \times Q_2, \Sigma_1 \cup \Sigma_2, \delta, I_1 \times I_2, F_1 \times F_2)$ where

$$\delta((q_1, q_2), a) = \begin{cases} (\delta_1(q_1, a), \delta_2(q_2, a)) & \text{if } a \in \Sigma_1 \cap \Sigma_2, \delta_1(q_1, a)! \text{ and } \delta_2(q_2, a)! \\ (\delta_1(q_1, a), q_2) & \text{if } a \in \Sigma_1 - \Sigma_2 \text{ and } \delta_1(q_1, a)! \\ (q_1, \delta_2(q_2, a)) & \text{if } a \in \Sigma_2 - \Sigma_1 \text{ and } \delta_2(q_2, a)! \\ \text{undefined} & \text{otherwise} \end{cases}$$

for $(q_1, q_2) \in Q_1 \times Q_2$ and $a \in \Sigma_1 \cup \Sigma_2$, and $\delta_i(q_i, a)!$ denotes the fact that there is a transition under a defined at q_i in $\mathcal{A}_i$.

2.2 Discrete-event systems

In this section, we recall the standard definition of a discrete-event system. Intuitively, we model the system as a non-deterministic finite automaton with partially observable behavior.

Definition 2.2. A *discrete-event system* (DES) G over Σ is an NFA over Σ together with the partition of Σ into Σ_o and Σ_{uo} of *observable* and *unobservable events*, respectively.

If we want to specify that the DES is modeled by a DFA, we talk about *deterministic* DES. If the marked states are irrelevant, we omit them and simply write $G = (Q, \Sigma, \delta, I)$.

The *observation projection* $P\colon \Sigma^* \to \Sigma_o^*$ is a morphism for concatenation defined by $P(a) = \varepsilon$ if $a \in \Sigma_{uo}$, and $P(a) = a$ if $a \in \Sigma_o$. The action of P on a string $a_1 a_2 \cdots a_n$, with $a_i \in \Sigma$ for $1 \leq i \leq n$, is to erase all unobservable events, that is, $P(a_1 a_2 \cdots a_n) = P(a_1)P(a_2) \cdots P(a_n)$. The definition can be readily extended to languages.

Definition 2.3. A *projected automaton* of a DES G over Σ with respect to the projection $P\colon \Sigma^* \to \Sigma_o^*$ is the NFA $P(G)$ obtained from G by replacing every transition (p, a, q) by $(p, P(a), q)$, followed by the standard elimination of the ε-transitions.

Equivalently, the transition function $\gamma\colon Q \times \Sigma_o \to 2^Q$ of $P(G)$ can be defined as $\gamma(q, a) = \delta(q, P^{-1}(a))$. Note that $P(G)$ is an NFA over Σ_o with the same states as G that recognizes the language $P(L_m(G))$ and can be constructed in polynomial time.

Definition 2.4. An *observer* of a DES G is the accessible part of the DFA constructed from $P(G)$ by the standard subset construction.

We call the DFA constructed from $P(G)$ by the standard subset construction a *full observer* of G. The full observer has exponentially many states compared with G, and in the worst case, the same holds for the observer as well, see [24] for more details.

For DESs with a single observable event we define a function φ_k that assigns, to every state q, the maximal number $i \in \{0, \ldots, k\}$ of observable steps that are possible from q.

Definition 2.5. Let $G^a = (Q, \Sigma, \delta, I)$ be a DES with $\Sigma_o = \{a\}$ and $P\colon \Sigma^* \to \{a\}^*$ be the observation projection. The function $\varphi_k\colon Q \to \{0, \ldots, k\}$ with respect to P is defined as $\varphi_k(q) = \max\{i \in \{0, \ldots, k\} \mid \delta(q, P^{-1}(a^i)) \neq \emptyset\}$.

Evidently, if $\varphi_k(q) \geq |Q|$ for a state $q \in Q$, then $\varphi_k(q) = k$, since there must be a cycle containing an observable event that is reachable from q. Therefore, we can assume that k is never greater than the number of states of the system G^a, i.e., $k \leq |Q|$.

Chapter 3

Notions of opacity

In this chapter, we present the formal definitions of all considered opacity notions within the finite automata model. For more details about opacity, we refer the reader to the overview by Jacob et al. [23].

In the first two sections, we introduce the language-based notions, namely language-based opacity and trace opacity. The rest of the chapter is dedicated to the notions of state-based opacity, namely current-state opacity, weak k-step opacity, strong k-step opacity, initial-state opacity, and initial-and-final-state opacity. Aside from strong k-step opacity, which is defined only for deterministic DESs, we define all other notions for nondeterministic systems.

Language-based opacity (LBO)

Language-based opacity was introduced by Badouel et al. [3] and Dubreil et al. [19]. We recall the most general definition by Lin [29]. Intuitively, a system is language-based opaque if for every string w in the secret language, there exists a string w' in the non-secret language with the same observation $P(w) = P(w')$. In this case, the intruder cannot conclude whether the secret string w or the non-secret string w' has occurred.

Definition 3.1. Given a DES $G = (Q, \Sigma, \delta, I)$, a projection $P\colon \Sigma^* \to \Sigma_o^*$, a secret language $L_S \subseteq L(G)$, and a non-secret language $L_{NS} \subseteq L(G)$. System G is *language-based opaque* (LBO) if $L_S \subseteq P^{-1}P(L_{NS})$.

We assume that the languages $L_S = L(\mathcal{A}_S)$ and $L_{NS} = L(\mathcal{A}_{NS})$ are represented by the non-blocking automata $\mathcal{A}_S = (Q_S, \Sigma, \delta_S, I_S, F_S)$ and $\mathcal{A}_{NS} = (Q_{NS}, \Sigma, \delta_{NS}, I_{NS}, F_{NS})$, respectively. Without loss of generality, we may assume that their sets of states are disjoint, that is, $Q_S \cap Q_{NS} = \emptyset$. It is worth mentioning that the secret and non-secret languages are often considered to be regular; and we consider it as well. The reason is that, for non-regular languages, the inclusion problem is undecidable; see Asveld and Nijholt [2] for more details.

Another notion studied in the literature is *weak language-based opacity* [29], which should not be confused with (strong) language-based opacity defined above. In comparison, the weak notion holds if the intruder confuses at least one secret string, formally $L_S \cap P^{-1}P(L_{NS}) \neq \emptyset$. We do not consider the weak notion in this paper.

Trace opacity (TO)

Trace opacity was introduced by Bryans et al. [11]. A trace $w \in \Sigma^*$ is secret if it contains any event from a specified set of secret events, otherwise w is non-secret. In accordance with [11], we consider all secret events to be unobservable. A system is trace opaque if for every secret trace, there is a non-secret trace that looks the same to the intruder.

Definition 3.2. Given a DES $G = (Q, \Sigma, \delta, I)$, a projection $P\colon \Sigma^* \to \Sigma_o^*$, and a set of unobservable secret events $S \subseteq \Sigma_{uo}$. System G is *trace opaque* (TO) if $P(Sec(G)) \subseteq P(Pub(G))$, where $Sec(G) = L(G) \cap \Sigma^* S \Sigma^*$ is the set of secret traces and $Pub(G) = L(G) \cap (\Sigma - S)^*$ is the set of non-secret traces.

Intuitively, trace opacity is a special case of language-based opacity, where the secret language of trace opacity is strictly defined as a set of strings containing at least one secret event, and the non-secret language is defined as any other behavior of the system. In Section 5.3, we present a way to construct automata $\mathcal{A}_S$ and $\mathcal{A}_{NS}$ from a trace opacity problem instance G such that $L(\mathcal{A}_S) = Sec(G)$ and $L(\mathcal{A}_{NS}) = Pub(G)$.

Current-state opacity (CSO)

Bryans et al. [12] introduced current-state opacity for systems modeled by Petri nets and Bryans et al. [11] generalized it to transition systems. Current-state opacity asks whether the intruder cannot decide, at any instance of time, whether the system is currently in a secret state. Therefore, the system is current-state opaque if, for every string leading to a secret state, there exists another string with the same observation that leads to a non-secret state.

Definition 3.3. Given a DES $G = (Q, \Sigma, \delta, I)$, a projection $P\colon \Sigma^* \to \Sigma_o^*$, a set of secret states $Q_S \subseteq Q$, and a set of non-secret states $Q_{NS} \subseteq Q$. System G is *current-state opaque* if for every string w such that $\delta(I, w) \cap Q_S \neq \emptyset$, there exists a string w' such that $P(w) = P(w')$ and $\delta(I, w') \cap Q_{NS} \neq \emptyset$.

Note that the definition of current-state opacity does not require $Q_{NS} = Q - Q_S$, and thus the systems we consider can contain states that are neither secret nor non-secret. We call these states *neutral* and we cannot simply handle them as non-secret states.

Weak k-step opacity (k-SO)

The notion of weak k-step opacity, which was introduced by Saboori and Hadjicostis [33, 37], is a generalization of current-state opacity requiring that the intruder cannot reveal the secret in the current state and k subsequent observable steps.

Definition 3.4. Given a DES $G = (Q, \Sigma, \delta, I)$, a projection $P\colon \Sigma^* \to \Sigma_o^*$, a set of secret states $Q_S \subseteq Q$, a set of non-secret states $Q_{NS} \subseteq Q$, and a parameter $k \in \mathbb{N}_\infty$. System G is *weakly k-step opaque* (k-SO) if for every string $st \in L(G)$ with $|P(t)| \leq k$ and $\delta(\delta(I, s) \cap Q_S, t) \neq \emptyset$, there exists a string $s't' \in L(G)$ such that $P(s) = P(s')$, $P(t) = P(t')$, and $\delta(\delta(I, s') \cap Q_{NS}, t') \neq \emptyset$.

We distinguish two special cases for $k = 0$ and $k = \infty$. By definition, weak 0-step opacity is equivalent to current-state opacity. In the case of weak ∞-step opacity, Yin and Lafortune [44] have shown that an n-state DES is weakly ∞-step opaque if and only if it is weakly $(2^n - 2)$-step opaque.

Strong k-step opacity (k-SSO)

While weak k-step opacity is considered to be relatively confidential, Falcone and Marchand [20] have shown that it is not as confidential as it may seem. The intruder may still be able to determine that the system was previously in a secret state, but not the exact time when this occurred. Therefore, they introduced a stronger notion of opacity called strong k-step opacity, which provides a higher level of confidentiality.

In accordance with [20], we consider strong k-step opacity only for deterministic DESs where all states that are not secret are non-secret, that is, $Q_{NS} = Q - Q_S$.

Definition 3.5. Given a deterministic DES $G = (Q, \Sigma, \delta, q_0)$, a projection $P\colon \Sigma^* \to \Sigma_o^*$, a set of secret states $Q_S \subseteq Q$, and a parameter $k \in \mathbb{N}_\infty$. System G is *strongly k-step opaque* (k-SSO) if for every string $s \in L(G)$, there exists a string $w \in L(G)$ such that $P(s) = P(w)$ and for every prefix w' of w, if $|P(w)| - |P(w')| \leq k$, then $\delta(q_0, w') \notin Q_S$.

Note that strong 0-step opacity is not equivalent to current-state opacity as in the case of weak 0-step opacity. In Theorem 5.47, we show that unobservable transitions from secret states to non-secret states are the only issues making the difference between strong 0-step opacity and weak 0-step (current-state) opacity. However, as pointed out by Wintenberg et al. [41], strong k-step opacity implies weak k-step opacity as long as no neutral states are considered.

Initial-state opacity (ISO)

Initial-state opacity was first introduced by Bryans et al. [12] for systems modeled by Petri nets and then Bryans et al. [11] generalized it to transition systems. Intuitively, initial-state opacity asks whether the intruder can never reveal whether the computation started in a secret state.

Definition 3.6. Given a DES $G = (Q, \Sigma, \delta, I)$, a projection $P\colon \Sigma^* \to \Sigma_o^*$, a set of secret initial states $Q_S \subseteq I$, and a set of non-secret initial states $Q_{NS} \subseteq I$. System G is *initial-state opaque* (ISO) if for every $w \in L(G, Q_S)$, there exists $w' \in L(G, Q_{NS})$ such that $P(w) = P(w')$.

We consider all states that are neither secret nor non-secret to be neutral. In particular, the secrecy status of the non-initial states do not play any role in initial-state opacity.

Initial-and-final-state opacity (IFO)

The last notion we consider is initial-and-final-state opacity of Wu and Lafortune [42]. Initial-and-final-state opacity is a generalization of both current-state opacity and initial-state opacity, where the secret is represented as a pair of an initial and a marked state. Consequently, initial-state opacity is a special case of initial-and-final-state opacity where the marked states do not play a role, and current-state opacity is a special case where the initial states do not play a role.

Definition 3.7. Given a DES $G = (Q, \Sigma, \delta, I)$, a projection $P\colon \Sigma^* \to \Sigma_o^*$, a set of secret state pairs $Q_S \subseteq I \times Q$, and a set of non-secret state pairs $Q_{NS} \subseteq I \times Q$. System G is *initial-and-final-state opaque* (IFO) if for every secret pair $(q_0, q_f) \in Q_S$ and every $w \in L(G, q_0)$ such that $q_f \in \delta(q_0, w)$, there exists a non-secret pair $(q_0', q_f') \in Q_{NS}$ and $w' \in L(G, q_0')$ such that $q_f' \in \delta(q_0', w')$ and $P(w) = P(w')$.

Chapter 4

Properties of current-state opacity

This chapter focuses on analyzing the complexity of verifying current-state opacity in systems with a restricted set of events and a restricted structure. We show that in most cases these restrictions do not make the verification tractable, and therefore the problem remains hard in essentially all practical cases.

The complexity of opacity verification has widely been investigated in the literature and is often based on the computation of observer. Thus the problem belongs to PSPACE. It is actually PSPACE-complete for most of the discussed notions. Indeed, Cassez et al. [16] showed that the verification of current-state opacity is at least as hard as deciding universality, which is PSPACE-complete for nondeterministic automata as well as for deterministic automata with partial observation.

Remark 4.1. *By Cassez et al. [16], the verification of current-state opacity is at least as hard as deciding universality. Indeed, for a DES $G = (Q, \Sigma, \delta, I, F)$, we have $L(G) = \Sigma^*$ if and only if G is current-state opaque with respect to $Q_S = Q - F$, $Q_{NS} = F$, and $P\colon \Sigma \to \Sigma$.*

However, PSPACE-completeness of universality problem requires a nontrivial structure of the model and the ability to express all possible strings. This give rise to a question whether there are structurally simpler systems for which the verification of opacity is tractable. We investigate the problem for, in our opinion, structurally the simplest systems: for acyclic automata (that do not have the ability to express all strings, and actually express only a finite number of strings) and for automata where all cycles are in the form of self-loops (which may still seem trivial in the structure, because as soon as the system leaves a state, it can never return to that state).

First, we reduce current-state opacity to the language inclusion problem. This reduction is similar to that of Wu and Lafortune [42] reducing current-state opacity to language-based opacity.

Lemma 4.2. *Let $G = (Q, \Sigma, \delta, I)$ be a DES, $P\colon \Sigma^* \to \Sigma_o^*$ a projection, and $Q_S, Q_{NS} \subseteq Q$ sets of secret and non-secret states. Let L_S denote the marked language of the automaton $G_S = (Q, \Sigma, \delta, I, Q_S)$ and L_{NS} denote the marked language of the automaton $G_{NS} = (Q, \Sigma, \delta, I, Q_{NS})$. Then G is current-state opaque if and only if $P(L_S) \subseteq P(L_{NS})$.* □

The observations from Remark 4.1 and Lemma 4.2, together with the results on the complexity of deciding universality and inclusion give us strong tools to show lower and upper complexity bounds for deciding (current-state) opacity. We summarized results from this chapter, together with the existing results, in Table 1.1.

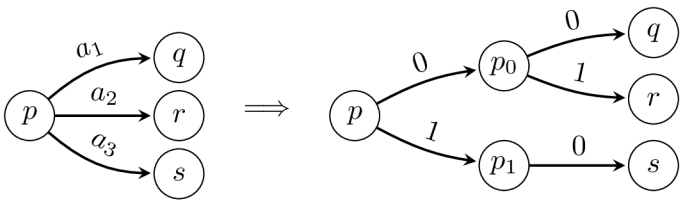


Figure 4.1: The replacement of three observable events $\{a_1, a_2, a_3\}$ with the encoding $e(a_1) = 00$, $e(a_2) = 01$, and $e(a_3) = 10$, and new states p_0 and p_1.

4.1 Simplification of the system

In this section we provide two useful transformations that can simplify any system without affecting its property of being current-state opaque. As a result, any instance of current-state opacity decision problem can be transformed in polynomial time into a deterministic system that has at most two observable events. Later, these simplifications will allow us to generalize some of the results from this chapter to other opacity notions.

The following transformation reduces the number of observable events in DESs with at least three observable events. The main idea is to encode the transition labels in binary. In Theorems 4.4, 5.5, and 5.10, we show that this transformation does not affect the system's status of current-state opacity, initial-state opacity, and trace opacity. This way we preserve the number of observable events in transformations in Chapter 5 that introduce new observable events.

Transformation 4.3. Let $G = (Q, \Sigma, \delta, I)$ be a DES modeled by an NFA, $P\colon \Sigma^* \to \Sigma_o$ be the observation projection, $\Gamma_o \subseteq \Sigma_o$ be an alphabet with at least three events, and $e\colon \Gamma_o \to \{0,1\}^k$ be a binary encoding (that is, an injective function), where $k \leq \lceil \log_2(|\Gamma_o|) \rceil$. We construct a DES

$$r(G) = (Q', (\Sigma - \Gamma_o) \cup \{0,1\}, \delta', I)$$

so that we start with the system G and replace every transition (p, a, q) with $a \in \Gamma_o$ and $e(a) = b_1 b_2 \cdots b_k \in \{0,1\}^k$ by k transitions

$$(p, b_1, p_{b_1}), (p_{b_1}, b_2, p_{b_1 b_2}), \ldots, (p_{b_1 \cdots b_{k-1}}, b_k, q)$$

where the states $p_{b_1}, \ldots, p_{b_1 \cdots b_{k-1}}$ are added to the set of states Q' of the system $r(G)$. These states are created when needed for the first time, and reused later during the replacements, cf. Figure 4.1 illustrating a replacement of three observable events $\{a_1, a_2, a_3\}$ with the encoding $e(a_1) = 00$, $e(a_2) = 01$, and $e(a_3) = 10$. Finally, we define projection $P'\colon [(\Sigma - \Gamma_o) \cup \{0,1\}]^* \to [(\Sigma_o - \Gamma_o) \cup \{0,1\}]^*$. ◇

Notice that the Transformation 4.3 preserves the number of unobservable events and determinism, and that it can be done in polynomial time. On the other hand, it does not preserve partial order because the encoding of a self-loop transition results in a cycle over two or more states.

The following theorem states that the transformation does not affect the property of the system to be current-state opaque, and therefore we can reduce the number of observable events of any current-state opacity instance to just two.

Theorem 4.4. *A DES G is current-state opaque with respect to Q_S, Q_{NS}, and P if and only if the DES $r(G)$ obtained by Transformation 4.3 is current-state opaque with respect to $Q'_S = Q_S$, $Q'_{NS} = Q_{NS} \cup (Q' - Q)$, and P'.* □

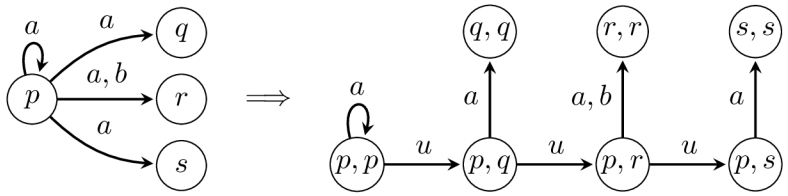


Figure 4.2: Determinization of a DES.

In the second transformation, we show how to transform a system modeled by an NFA to a system modeled by a DFA without affecting the system's properties of being current-state opaque, acyclic, and partially ordered.

Transformation 4.5. Let $G = (Q, \Sigma, \delta, I)$ be a DES modeled by an NFA with the secret states Q_S, the non-secret states Q_{NS}, and the corresponding projection $P\colon \Sigma^* \to \Sigma_o^*$. We construct a deterministic DES G_{det} in two steps.

1. First, we ensure that the system has a unique initial state. From G we construct a DES $G' = (Q', \Sigma, \delta', \{q_0\})$, where $Q' = Q \cup \{q_0\}$ contains a new non-secret initial state q_0. Further, for each $q \in I$ we add a new transition (q_0, a, q) to δ', where $a \in \Sigma_o$ is an arbitrary observable event.
2. In the second step, we determinize the transition function of the system. From G' we construct a DES $G_{det} = (Q'', \Sigma \cup \{u\}, \delta'', (q_0, q_0))$ modeled by a DFA, where $Q'' = Q' \times Q'$ is the set of pairs of states, u is a new unobservable event, and the pair $(q_0, q_0) \in Q''$ is a new initial state. We define the transition function δ'' as follows.
   (a) For every transition (p, a, q) in δ', where $p, q \in Q'$ and $a \in \Sigma$, we add a transition $((p, q), a, (q, q))$ to δ''.
   (b) For every state $p \in Q'$ we define the set $R_p = \cup_{a\in\Sigma}\, \delta'(p, a) - \{p\} = \{p_1, p_2, \ldots, p_\ell\}$ of states different from p that can be reached from p by a single transition. We then add transitions of the form $((p, p), u, (p, p_1))$ and $((p, p_i), u, (p, p_{i+1}))$ for $i = 1, \ldots, \ell - 1$, as shown in Figure 4.2, to create a chain of states from R_p connected to state (p, p). Note that the order in which we connect states from R_p does not affect the resulting system.

We remove unreachable states from G_{det}. Finally, we define the projection $P'\colon (\Sigma^* \cup \{u\}) \to \Sigma_o^*$, and the sets of secret states $Q'_S = \{(p, q) \mid p \in Q_S\}$ and of non-secret states $Q'_{NS} = \{(p, q) \mid p \in Q_{NS} \cup \{q_0\}\}$. ◇

Notice that Transformation 4.5 can be done in polynomial time using at most $(n+1)^2$ states, where n is the number of states in G. In fact, if we omit removing unreachable states at the end of the tranformation, then G_{det} can computed in deterministic logarithmic space. Additionally, this transformation does not introduce any new neutral states and it preserves the number of observable events, acyclicity, and partial order.

Theorem 4.6. *A DES G is current-state opaque with respect to Q_S, Q_{NS}, and P if and only if the deterministic DES G_{det} obtained by Transformation 4.5 is current-state opaque with respect to Q'_S, Q'_{NS}, and P'.* □

4.2 Restriction on structure of the system

Our first restriction concerns the number of observable and unobservable events in the system. The following result thus improves the general case in two ways: (i) compared to the general settings we keep the system deterministic, and, mainly, (ii) we restrict the number of observable events to two and the number of unobservable events to one.

Theorem 4.7. *Deciding current-state opacity of a DES modeled by a DFA with three events, one of which is unobservable, is* PSpace*-complete.*

Proof. Membership in PSpace was shown by Saboori [32], and also follows directly from Lemma 4.2.

To show hardness, we reduce the current-state opacity problem for a DES modeled by an NFA with just two observable events, which is PSpace-complete by Remark 4.1 and Saboori [32]. This can be done by Transformation 4.5 which, for a DES modeled by an NFA with just two observable events, constructs a deterministic DES with three events, one of which is unobservable, without affecting the property of current-state opacity. □

Notice that an unobservable event in the previous theorem is unavoidable because any DFA with all events observable is always in a unique state, and therefore never opaque. However, the reader may wonder what happens if we further restrict the number of observable events to just one. We now show that having only one observable event makes the problem computationally easier unless coNP = PSpace. This result holds even without any restriction on the number of unobservable events, and for nondeterministic automata.

Theorem 4.8. *Deciding current-state opacity of a DES modeled by an NFA with a single observable event is* coNP*-complete.*

Proof. Membership in coNP follows from Lemma 4.2 and the fact that inclusion for unary NFAs is coNP-complete, and hardness follows from the complexity of deciding universality for unary NFAs. For both claims used here, the reader is referred to Stockmeyer and Meyer [40]. □

We obtain the following result for DFAs by applying Transformation 4.5 which, for a DES modeled by an NFA with a single observable event, constructs a deterministic DES with two events, one of which is unobservable, without affecting the property of current-state opacity.

Corollary 4.9. *Deciding current-state opacity of a DES modeled by a DFA with two events, one of which is unobservable, is* coNP*-complete.*

Previous results show that only restricting the number of events does not lead to tractable complexity. But it gives rise to another question whether there are structurally simpler systems for which the opacity verification problem is tractable.

Structurally the simplest systems we could think of are acyclic DFAs with full observation, recognizing only finite languages. However, these systems are never opaque, since they are deterministic and fully observed. Nontrivial structures to be considered could thus be acyclic NFAs that still recognize only finite languages, and hence do not possess the ability to express all strings over the alphabet. We combine this restriction with the restriction on the number of events.

Theorem 4.10. *Deciding current-state opacity of a DES modeled by an acyclic NFA with two or more observable events is* coNP*-complete.* □

Again, we can show that the situation is computationally simpler if only one observable event is allowed.

Theorem 4.11. *Deciding current-state opacity of a DES modeled by an acyclic NFA with a single observable event is* NL*-complete, and hence solvable in polynomial time.* □

Since Transformation 4.5 preserves acyclicity and can be computed in deterministic logarithmic space, we can apply it to the systems of Theorems 4.10 and 4.11 to obtain hardness part of following results. Membership then follows from Lemma 4.2 and the corresponding results on the complexity of inclusion.

Corollary 4.12. *Deciding current-state opacity of a DES is*

1. coNP*-complete if the system is modeled by an acyclic DFA with three events, one of which is unobservable, and*
2. NL*-complete if the system is modeled by an acyclic DFA with two events, one of which is unobservable.*

Above, we considered systems generating only finitely many behaviors. However, real-world systems are usually not that simple and often require additional properties, such as deadlock freeness. Therefore, we now consider partially ordered automata, a kind of automata where all cycles are only in the form of self-loops. Such automata are, in our opinion, structurally the simplest DES where deadlock freeness can be ensured (by adding a self-loop). Their mark languages form a subclass of regular languages strictly included in *star-free languages*, see [13, 39]. Star-free languages are languages definable by *linear temporal logic* that is often used as a specification language in automated verification.

We then immediately obtain the following result for nondeterministic partially ordered automata.

Theorem 4.13. *Deciding current-state opacity of a DES modeled by a poNFA with only two events, both of which are observable, is* PSPACE*-complete.*

Proof. Membership in PSPACE follows from Lemma 4.2 and the results on the complexity of inclusion for poNFAs, and hardness from the fact that deciding universality for poNFAs with only two events is PSPACE-complete. For both claims see Krötzsch et al. [27]. □

The situation is again easier if the model has only a single observable event.

Theorem 4.14. *Deciding current-state opacity of a DES modeled by a poNFA with a single observable event is* NL*-complete.* □

Again, we use Transformation 4.5, which preserves partial order and can be computed in deterministic logarithmic space, and apply it to the systems of Theorems 4.13 and 4.14 to obtain the hardness part of the following results. Membership then follows from Lemma 4.2 and the corresponding results on the complexity of inclusion.

Corollary 4.15. *Deciding current-state opacity of a DES is*

1. PSPACE*-complete if the system is modeled by a poDFA with three events, one of which is unobservable, and*
2. NL*-complete if the system is modeled by a poDFA with two events, one of which is unobservable.*

Chapter 5

Transformations among opacity notions

In this chapter, we introduce new trasnformations among the considered opacity decision problems. In other words, for an instance of one opacity notion that consists of a DES, an observation projection, and a secret description, we transform it into an instance of another opacity notion.

Comparing different notions of opacity for automata models, Saboori and Hadjicostis [34] provided a language-based definition of initial-state opacity, Cassez et al. [16] transformed trace opacity to current-state opacity, and Wu and Lafortune [42] showed that current-state opacity, initial-and-final-state opacity, and language-based opacity can be transformed to each other. They further provided transformations of initial-state opacity to language-based opacity and to initial-and-final-state opacity, and, for prefix-closed languages, a transformation of language-based opacity to initial-state opacity.

In this work, we extend these results by showing that, for automata models, all the discussed notions of opacity are transformable to each other. As well as the existing transformations, our transformations are computable in polynomial time and preserve the number of observable events and determinism (whenever it is meaningful). In the case of state-based opacity notions, our goal was to design transformations that do not introduce any new neutral states into the system, since their existence may not be practically justified. However, in some cases, we may need to give a separate transformation for systems that already contain neutral states. The meaning of neutral states is not yet clear in the literature. They are fundamental in language-based opacity, but questionable in state-based opacity. In any case, we cannot simply handle neutral states as non-secret states. We summarize our results, together with the existing results, in Figure 5.1.

There are two immediate applications of the transformations. First, the transformations provide a deeper understanding of the differences among the opacity notions from the structural point of view. For instance, the reader may deduce from the transformations that, for prefix-closed languages, the notions of language-based opacity, initial-state opacity, and current-state opacity coincide, or that to transform current-state opacity to weak ∞-step opacity means to add only a single state and a few transitions.

Second, the transformations provide a tool to obtain the complexity results for all the discussed opacity notions by studying just one of the notions. To illustrate, consider for example the result of Theorem 4.7 showing that deciding current-state opacity for systems modeled by DFAs with three events, one of which is unobservable, is PSPACE-complete. Since we can transform the problems of deciding current-state opacity and of deciding

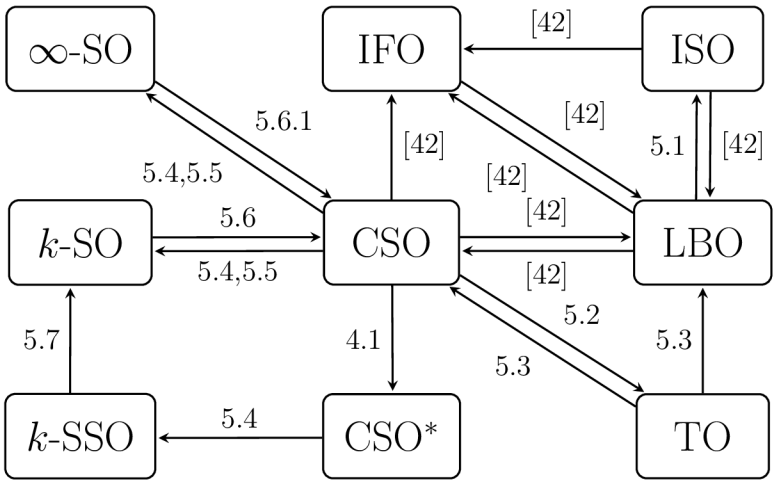


Figure 5.1: Overview of the transformations among the notions of opacity for automata models. The node CSO* denotes a simplified instance of current-state opacity modeled by a DFA with at most two observable events.

weak k-step opacity to each other in polynomial time, preserving determinism and the number of observable events, we obtain that deciding weak k-step opacity for systems modeled by DFAs with three events, one of which is unobservable, is PSPACE-complete as well. In particular, combining the transformations with known results from Jacob et al. [23] and results from Chapter 4, we obtain a complete complexity picture of verifying the discussed notions of opacity, as summarized in Table 1.2.

Thus, by combining the transformations of Wu and Lafortune [42] with Theorems 4.7 and 4.8, we immedietly obtain new results for language-based opacity and initial-and-final-state opacity. In more detail, the transformations of Wu and Lafortune [42] preserve the determinism of transitions, but result in automata with a set of initial states. This issue can, however, be easily fixed by adding a new initial state, connecting it to the original initial states by new unobservable events, and making the original initial states non-initial.

Corollary 5.1. *The problems of deciding whether a DES satisfies language-based opacity and initial-and-final-state opacity are* PSPACE*-complete. The problems remain* PSPACE*-complete even if the system is a DFA with three events, one of which is unobservable.*

Corollary 5.2. *The problems of deciding whether a DES with a single observable event satisfies language-based opacity and initial-and-final-state opacity are* coNP*-complete.*

Moreover, the transformations of Wu and Lafortune [42] preserve both acyclicity and partial order, and hence we can generalize the results from Chapter 4 for acyclic and partially ordered automata in the same way. On the other hand, the majority of our transformations do not preserve either partial order, due to the utilization of Transformation 4.3, or acyclicity. Consequently, our transformations do not extend these results to the remaining notions discussed.

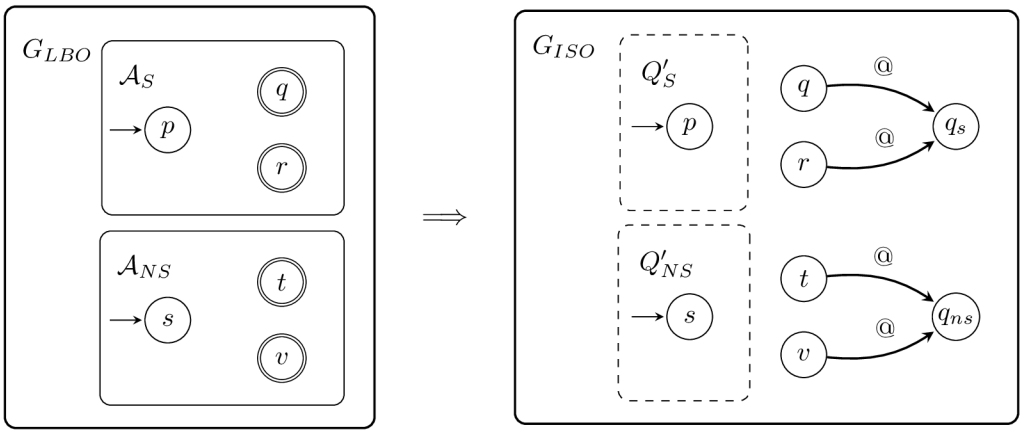


Figure 5.2: Transforming language-based opacity to initial-state opacity.

5.1 LBO to ISO

In this section, we discuss the transformations from language-based opacity to initial-state opacity. The transformation for the case where both the secret and non-secret languages of the language-based opacity problem are prefix closed has been provided by Wu and Lafortune [42]. We now extend this transformation to the general case. We further show that the initial-state opacity decision problem with a single observable event is NL-complete. Consequently, there exists no polynomial-time transformation for this case that preserves the number of observable events, unless P = NP.

Let the language-based opacity problem be represented by a DES G_{LBO}. We transform it to a DES G_{ISO} in such a way that G_{LBO} is language-based opaque if and only if G_{ISO} is initial-state opaque. Our transformation proceeds in two steps:

1. We construct a DES G_{ISO} with one additional observable event @ using Transformation 5.3.
2. We use Transformation 4.3 to reduce the number of observable events of G_{ISO} by one.

Since the second step follows from Transformation 4.3, we only describe the first step, that is, the construction of G_{ISO} over $\Sigma \cup \{@\}$.

Transformation 5.3. Let $G_{LBO} = (Q, \Sigma, \delta, I)$ be a DES with the corresponding projection $P\colon \Sigma^* \to \Sigma_o^*$, a secret language $L_S \subseteq L(G_{LBO})$ given by the non-blocking automaton $\mathcal{A}_S = (Q_S, \Sigma, \delta_S, I_S, F_S)$, and a non-secret language $L_{NS} \subseteq L(G_{LBO})$ given by the non-blocking automaton $\mathcal{A}_{NS} = (Q_{NS}, \Sigma, \delta_{NS}, I_{NS}, F_{NS})$. We construct a DES

$$G_{ISO} = (Q_S \cup Q_{NS} \cup \{q_s, q_{ns}\}, \Sigma \cup \{@\}, \delta', I_S \cup I_{NS})$$

where G_{ISO} is a disjoint union of the automata $\mathcal{A}_S$ and $\mathcal{A}_{NS}$ together with two new states and a new observable event @. The transition function δ' is initialized as $\delta' := \delta_S \cup \delta_{NS}$ and further extended as follows, see Figure 5.2 for an illustration:

1. for every state $q \in F_S$, we add a new transition $(q, @, q_s)$ to δ';
2. for every state $q \in F_{NS}$, we add a new transition $(q, @, q_{ns})$ to δ'.

Finally, let $Q'_S = I_S$ denote the set of secret initial states of G_{ISO}, and let $Q'_{NS} = I_{NS}$ denote the set of non-secret initial states of G_{ISO}. We extend the projection P to $P'\colon (\Sigma \cup \{@\})^* \to (\Sigma_o \cup \{@\})^*$. ◇

Notice that Transformation 5.3 can be done in polynomial time and that it preserves determinism of transitions.

The following theorem justifies the correctness of Transformation 5.3.

Theorem 5.4. *A DES G_{LBO} is language-based opaque with respect to L_S, L_{NS}, and P if and only if the DES G_{ISO} obtained by Transformation 5.3 is initial-state opaque with respect to Q'_S, Q'_{NS}, and P'.* □

By the following theorem, reducing the number of observable events by using Transformation 4.3 does not affect initial-state opacity of any DES.

Theorem 5.5. *A DES $G = (Q, \Sigma, \delta, I)$ is initial-state opaque with respect to Q_S, Q_{NS}, and P if and only if the DES $r(G) = (Q', (\Sigma - \Gamma_o) \cup \{0,1\}, \delta', I)$ obtained by Transformation 4.3 is initial-state opaque with respect to Q_S, Q_{NS}, and P'.* □

Since we need at least two initial states for initial-state opacity to be non-trivial, we generalize the weaker form of Theorem 4.7 to initial-state opacity. Therefore, using Transformations 5.3 and 4.3, and taking into account the fact that the problem of verifying initial-state opacity is in PSPACE [32], we can state the following result for NFAs with deterministic transition function.

Corollary 5.6. *The problem of deciding whether a DES satisfies initial-state opacity is* PSPACE*-complete. The problem remains* PSPACE*-complete even if the system is an NFA with deterministic transition function and three events, one of which is unobservable.*

To preserve the number of observable events, the general transformation relies on the binary encoding of events by Transformation 4.3. However, the encoding requires at least two observable events in G_{LBO}, and hence it is not applicable to systems with a single observable event. In fact, we show that there does not exist such a transformation unless P = NP, which is a longstanding open problem of computer science. Deciding language-based opacity for systems with a single observable event is coNP-complete [22, 40]. We show that deciding initial-state opacity for systems with a single observable event is NL-complete. In particular, the problem can be solved in polynomial time.

Theorem 5.7. *Deciding initial-state opacity of a DES with a single observable event is* NL*-complete.* □

5.2 CSO to TO

In this section, we discuss the transformations from current-state opacity to trace opacity. The transformation we provide results in a system with at least two observable events. Similar to initial-state opacity, we show that the trace opacity decision problem with a single observable event is NL-complete. Consequently, there is no polynomial-time transformation for this case that preserves the number of observable events, unless P = NP.

Let the current-state opacity problem be represented by a DES G_{CSO}. We transform it to a DES G_{TO} in such a way that G_{CSO} is current-state opaque if and only if G_{TO} is trace opaque. Our transformation proceeds in two steps:

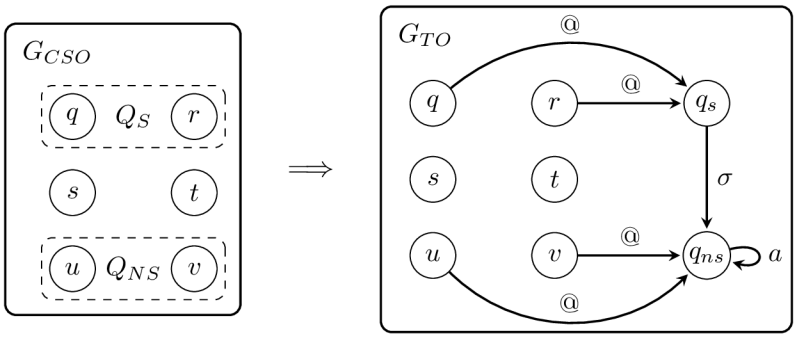


Figure 5.3: Transforming current-state opacity to trace opacity.

1. We construct a DES G_{TO} with one additional observable event @ using Transformation 5.8.
2. We use Transformation 4.3 to reduce the number of observable events of G_{TO} by one.

Since the second step follows from Transformation 4.3, we only describe the first step, that is, the construction of G_{CSO} over $\Sigma \cup \{@\}$.

Transformation 5.8. Let $G_{CSO} = (Q, \Sigma, \delta, I)$ be a DES with the secret states Q_S, the non-secret states Q_{NS}, and the corresponding projection $P\colon \Sigma^* \to \Sigma_o^*$. We construct a DES

$$G_{TO} = (Q \cup \{q_s, q_{ns}\}, \Sigma \cup \{@, \sigma\}, \delta', I)$$

where q_s and q_{ns} are new states, @ is a new observable event, and σ is a new unobservable secret event. The transition function δ' is initialized as the transition function δ of the system G_{CSO} and further extended as follows, see Figure 5.3 for an illustration:

1. for every secret state $q \in Q_S$, we add the transition $(q, @, q_s)$ to δ',
2. for every non-secret state $q \in Q_{NS}$, we add the transition $(q, @, q_{ns})$ to δ',
3. we add the secret transition (q_s, σ, q_{ns}) to δ', and
4. we add one self-loop transition (q_{ns}, a, q_{ns}) to δ', where $a \in \Sigma_o$ is an arbitrary observable event.

We define the projection $P'\colon (\Sigma \cup \{@, \sigma\})^* \to (\Sigma_o \cup \{@\})^*$ and the set of secret events $S = \{\sigma\}$. ◇

Notice that Transformation 5.8 can be done in polynomial-time and that it preserves determinism.

The following theorem justifies the correctness of Transformation 5.8.

Theorem 5.9. *A DES G_{CSO} is current-state opaque with respect to Q_S, Q_{NS}, and P if and only if the DES G_{TO} obtained by Transformation 5.8 is trace opaque with respect to S and P'.* □

By the following theorem, reducing the number of observable events by using Transformation 4.3 does not affect trace opacity of any DES.

Theorem 5.10. *A DES G is trace opaque with respect to S and P if and only if the DES $r(G)$ obtained by Transformation 4.3 is trace opaque with respect to S and P'.* □

Since Transformation 5.8 introduces a new unobservable secret event, we cannot directly generalize Theorem 4.7 to trace opacity. However, by utilizing Theorem 5.10 and the membership of trace opacity in PSPACE [18], we can state a weaker result as follows.

Corollary 5.11. *The problem of deciding whether a DES satisfies trace opacity is* PSPACE*-complete. The problem remains* PSPACE*-complete even if the system is a DFA with four events, two of which are unobservable.*

The second step of our transformation, that is the binary encoding represented by Transformation 4.3, requires that G_{CSO} has at least three observable events or, equivalently, that G_{TO} has at least two observable events. Consequently, our transformation does not preserve the number of observable events if G_{TO} has a single observable event. We show that the trace opacity decision problem with a single observable event is NL-complete. In particular, the problem can be solved in polynomial time.

Theorem 5.12. *Deciding trace opacity of a DES with a single observable event is* NL*-complete.* □

5.3 TO to CSO

In this section, we show how to transform trace opacity to current-state opacity. Previously, such a transformation was provided by Cassez et al. [16], but they assumed that a deterministic automaton $\mathcal{A}_S$ for the language of secret traces was given as input. Additionally, for a nondeterministic $\mathcal{A}_S$ their transformation is not polynomial. We improve this result by providing a transformation from trace opacity to current-state opacity that is always polynomial. Further, our transformation enables us to construct automata $\mathcal{A}_S$ and $\mathcal{A}_{NS}$ representing the secret and non-secret trace languages, thus transforming the problem also to language-based opacity problem.

Let the trace opacity problem be represented by a DES G_{TO}. We transform it to a DES G_{CSO} in such a way that G_{TO} is trace opaque if and only if G_{CSO} is current-state opaque.

Transformation 5.13. Let $G_{TO} = (Q, \Sigma, \delta, I)$ be a DES with the set of secret events $S \subseteq \Sigma_{uo}$ and the corresponding projection $P\colon \Sigma^* \to \Sigma_o^*$. We construct a DES

$$G_{CSO} = (Q \cup Q_S, \Sigma, \delta', I)$$

as a disjoint union of G and $G_s = (Q_S, \Sigma, \delta_s, I_s)$, where G_s is copy of G and $Q_S = \{q' \mid q \in Q\}$ is a disjoint copy of Q. We initialize $\delta' := \delta \cup \delta_s$ and further modify it by replacing every secret transition (q, σ, r) originally in δ by transition (q, σ, r') in δ', where $\sigma \in S$ and $r' \in Q_S$, cf. Figure 5.4 for an illustration. The states of Q_S are the secret states of G_{CSO}, while the rest of the states are non-secret, i.e., $Q_{NS} = Q$. Finally, we remove unreachable states and corresponding transitions. ◇

Notice that Transformation 5.13 can be done in polynomial time and that it preserves determinism and the number of observable and unobservable events.

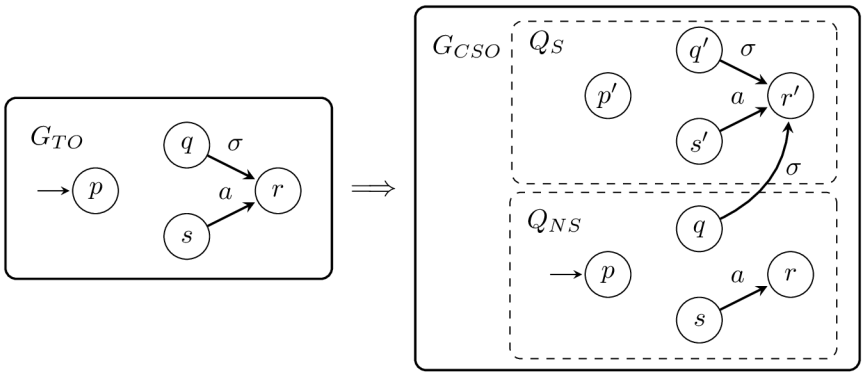


Figure 5.4: Transforming trace opacity to current-state opacity.

Remark 5.14. *To reduce G_{TO} to language-based opacity, we set $L_S = L(\mathcal{A}_S)$ and $L_{NS} = L(\mathcal{A}_{NS})$, where $\mathcal{A}_S = (Q \cup Q_S, \Sigma, \delta', I, Q_S)$ is identical to the DES G_{CSO} from Transformation 5.13, except for the set of marked states, and $\mathcal{A}_{NS} = (Q, \Sigma, \delta'', I, Q)$ is an automaton that corresponds to the original system G_{TO} with all states marked and with all secret transitions removed, that is, $\delta'' = \delta \cap Q \times (\Sigma - S) \times Q$.*

The following theorem justifies the correctness of Transformation 5.13.

Theorem 5.15. *A DES G_{TO} is trace opaque with respect to S and P if and only if the DES G_{CSO} obtained by Transformation 5.13 is current-state opaque with respect to Q_S, Q_{NS}, and P.* □

5.4 CSO to k-SSO

In this section, we show how to transform current-state opacity to strong k-step opacity. For systems without neutral states, strong k-step opacity implies weak k-step opacity [41], and thus the following transformations are also applicable to weak k-step opacity. Again, the general transformation uses Transformation 4.3 to preserve the number of observable events, and therefore we provide a separate transformation for systems with a single observable event.

5.4.1 The general case

Let the current-state opacity problem be represented by a DES G_{CSO}. We transform it to a deterministic DES $G_{k\text{-}SSO}$ in such a way that G_{CSO} is current-state opaque if and only if $G_{k\text{-}SSO}$ is strongly k-step opaque.

Our transformation proceeds in three steps:

1. If G_{CSO} is not deterministic, we determinize it by Transformation 4.5.
2. We construct a DES $G_{k\text{-}SSO}$ with one additional observable event @ using Transformation 5.16.
3. We use Transformation 4.3 to reduce the number of observable events of $G_{k\text{-}SSO}$ by one.

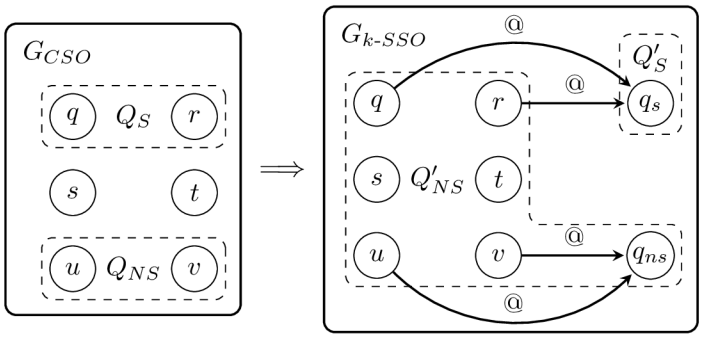


Figure 5.5: Transforming current-state opacity to strong k-step opacity, for an arbitrary parameter $k \in \mathbb{N}_\infty$.

Since the first and third step follow from Transformations 4.5 and 4.3, we only describe the second step, that is, the construction of $G_{k\text{-}SSO}$ over $\Sigma \cup \{@\}$.

Transformation 5.16. Let $G_{CSO} = (Q, \Sigma, \delta, q_0)$ be a deterministic DES with the secret states Q_S, the non-secret states Q_{NS}, and the corresponding projection $P\colon \Sigma^* \to \Sigma_o^*$. We construct a DES

$$G_{k\text{-}SSO} = (Q \cup \{q_s, q_{ns}\}, \Sigma \cup \{@\}, \delta', q_0)$$

where q_s and q_{ns} are new states and @ is a new observable event. The transition function δ' is initialized as the transition function δ of the system G_{CSO} and further extended as follows, see Figure 5.5 for an illustration:

1. for every secret state $q \in Q_S$, we add the transition $(q, @, q_s)$ to δ', and
2. for every non-secret state $q \in Q_{NS}$, we add the transition $(q, @, q_{ns})$ to δ'.

We define the projection $P'\colon (\Sigma \cup \{@\})^* \to (\Sigma_o \cup \{@\})^*$, and the sets of secret states $Q'_S = \{q_s\}$ and of non-secret states $Q'_{NS} = Q \cup \{q_{ns}\}$. ◇

Notice that Transformation 5.16 can be done in polynomial time and that it preserves determinism. It is also independent of the parameter k, and therefore works for any $k \in \mathbb{N}_\infty$ without affecting the size of the resulting system $G_{k\text{-}SSO}$.

Intuitively, since there is no extension from the unique secret state q_s, there is always a corresponding (trivial) extension from every non-secret state. Consequently, we can apply Transformation 4.3 to $G_{k\text{-}SSO}$ and encode new event @ in binary without affecting strong k-step opacity of the system $G_{k\text{-}SSO}$.

Remark 5.17. *Transformation 5.16 can also be used to remove neutral states from the system, so can we think of it as a transformation from current-state opacity with neutral states to current-state opacity without neutral states.*

The following theorem justifies the correctness of Transformation 5.16.

Theorem 5.18. *A DES G_{CSO} is current-state opaque with respect to Q_S, Q_{NS}, and P if and only if the DES $G_{k\text{-}SSO}$ obtained by Transformation 5.16 is strongly k-step opaque, for any parameter $k \in \mathbb{N}_\infty$, with respect to Q'_S and P'.* □

In Theorem 4.7 we showed that the problem of deciding current-state opacity of a DES modeled by a DFA with three events, one of which is unobservable, is PSpace-complete. Transformations 5.16 and 4.3 allow us to transform instance of this problem to the problems of deciding weak and strong k-step opacity while preserving determinism and the number of observable events. Thus, we can state the following result.

Corollary 5.19. *Given a natural number k represented by $O(\log(k))$ bits and a DES G. The problems of deciding whether the system G satisfies weak k-step opacity and strong k-step opacity are* PSpace*-hard. The problems remain* PSpace*-hard even if the system G is a DFA with three events, one of which is unobservable.*

Since weak ∞-step opacity is a special case of weak k-step opacity, the previous corollary also implies PSpace-hardness for weak ∞-step opacity.

5.4.2 The case of $|\Sigma_o| = 1$

To preserve the number of observable events, the general transformation relies on the binary encoding of events by Transformation 4.3. However, the encoding requires at least two observable events in G_{CSO}, and hence it is not applicable to systems with a single observable event. For these systems, we provide a separate transformation that requires to add $k+1$ new states, and therefore the size of the resulting system is linear with respect to the parameter $k \in \mathbb{N}$.

Let the current-state opacity problem with a single observable event be represented by a DES G^a_{CSO} without neutral states. We transform it to a DES $G^a_{k\text{-}SSO}$ in such a way that G^a_{CSO} is current-state opaque if and only if $G^a_{k\text{-}SSO}$ is strongly k-step opaque.

Without loss of generality, we assume that G^a_{CSO} is deterministic, as we can always use Transformation 4.5 to determinize it. We further assume that in G^a_{CSO}, there are no non-secret states that can be reached from a secret state by any sequence of unobservable events, formally $\delta(Q_S, P^{-1}(\varepsilon)) \cap Q_{NS} = \emptyset$. We describe this property with respect to current-state opacity of the system in the following lemma.

Lemma 5.20. *A DES G is current-state opaque with respect to Q_S, Q_{NS}, and P if and only if G is current-state opaque with respect to $Q'_S = Q_S - R$, $Q'_{NS} = Q_{NS} \cup R$, and P, where $R = \{q_s \in Q_S \mid \delta(q_s, P^{-1}(\varepsilon)) \cap Q_{NS} \neq \emptyset\}$.* □

Transformation 5.21. Let $G^a_{CSO} = (Q, \Sigma, \delta, q_0)$ be a deterministic DES with a single observable event $\Sigma_o = \{a\}$, the secret states Q_S, the non-secret states $Q_{NS} = Q - Q_S$, and the corresponding projection $P\colon \Sigma^* \to \{a\}^*$. By Lemma 5.20, we assume that $\delta(Q_S, P^{-1}(\varepsilon)) \cap Q_{NS} = \emptyset$. We construct a DES

$$G^a_{k\text{-}SSO} = (Q \cup \{q^\star_0, \ldots, q^\star_k\}, \Sigma \cup \{u\}, \delta', q_0)$$

by adding $k+1$ new non-secret states and a new unobservable event u. The transition function δ' is initialized as the transition function δ of the system G^a_{CSO} and further extended as follows, see Figure 5.6 for an illustration:

1. for every state $q \in Q$, we add a new transition $(q, u, q^\star_0)$ to δ';
2. for every state $q^\star_i$, where $i \in \{0, \ldots, k-1\}$, we add a new transition $(q^\star_i, a, q^\star_{i+1})$ to δ'.

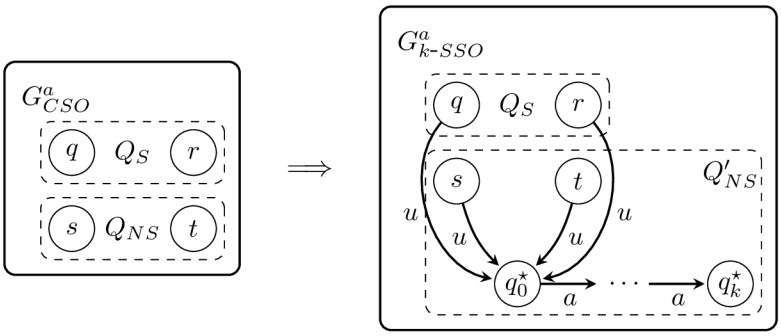


Figure 5.6: Transforming current-state opacity with a single observable event to strong k-step opacity.

The set of secret states Q_S remains unchanged in $G_{k\text{-}SSO}$, while all other states are non-secret. We extend the projection P to $P'\colon (\Sigma \cup \{u\})^* \to \{a\}$. ◇

Notice that Transformation 5.21 can be done in polynomial time and that it preserves determinism and the number of observable events.

Remark 5.22. *It seems that adding k new states to the system cannot be avoided, since for $k \geq |Q|$ the problem of deciding strong k-step opacity of a system with a single observable event can be solved in polynomial time. First, we search the system for a cycle containing only non-secret states and at least one observable transition. Then, we verify if the system is strongly k-step opaque in the first $|Q|$ observable steps before the cycle is reached (if it exists). Clearly, both conditions can verified in polynomial time.*

The following theorem justifies the correctness of Transformation 5.21.

Theorem 5.23. *A DES G^a_{CSO} with a single observable event $\Sigma_o = \{a\}$ is current-state opaque with respect to Q_S, Q_{NS}, and P if and only if the DES $G^a_{k\text{-}SSO}$ obtained by Transformation 5.21 is strongly k-step opaque with respect to Q_S and P'.* □

In Theorem 4.8 we showed that the problem of deciding current-state opacity of a DES with a single observable event is coNP-complete. Transformation 5.21 allows us to generalize the hardness part of this result to strong k-step opacity. However, the transformation is linear with respect to the parameter k, and therefore we consider k to be encoded in unary in the following corollary.

Corollary 5.24. *Given a natural number k represented in unary and a DES G with a single observable event. The problem of deciding whether the system G satisfies strong k-step opacity is* coNP*-hard.*

5.5 CSO to k-SO

In this section, we describe the general transformation from current-state opacity to weak k-step opacity that uses neutral states to preserve the number of observable events without the help of Transformation 4.3. Notably, unlike the transformations discussed in the previous section, Transformation 5.25 is applicable to systems that have both neutral states and a single observable event, and the resulting system will still have a single observable event.

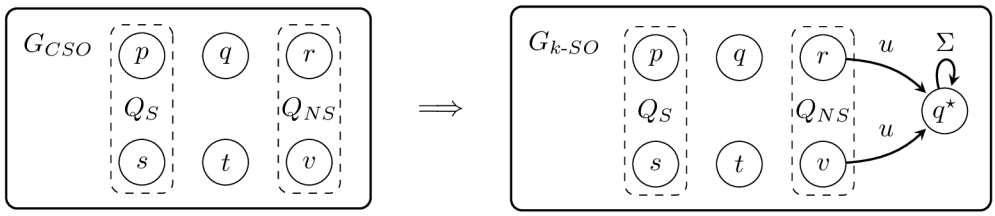


Figure 5.7: Transforming current-state opacity to weak k-step opacity, for an arbitrary parameter $k \in \mathbb{N}_\infty$.

Let the current-state opacity problem be represented by a DES G_{CSO}. We transform it to a DES $G_{k\text{-}SO}$ in such a way that G_{CSO} is current-state opaque if and only if $G_{k\text{-}SO}$ is weakly k-step opaque.

Transformation 5.25. Let $G_{CSO} = (Q, \Sigma, \delta, I)$ be a DES with the secret states Q_S, the non-secret states Q_{NS}, and the corresponding projection $P\colon \Sigma^* \to \Sigma_o^*$. We construct a DES

$$G_{k\text{-}SO} = (Q \cup \{q^\star\}, \Sigma \cup \{u\}, \delta', I)$$

where u is a new unobservable event and $q^\star$ is a new neutral state. The transition function δ' is initialized as the transition function δ of the system G_{CSO} and further extended as follows, see Figure 5.7 for an illustration:

1. for each state $q \in Q_{NS}$, we add a transition $(q, u, q^\star)$ to δ';
2. for each $a \in \Sigma$, we add a self-loop $(q^\star, a, q^\star)$ to δ'.

We extend the projection P to the projection $P'\colon (\Sigma \cup \{u\})^* \to \Sigma_o^*$. The sets Q_S and Q_{NS} remain unchanged. ◇

Notice that Transformation 5.25 can be done in polynomial time and that it preserves determinism and the number of observable events. It is also independent of the parameter k, and hence it works for any parameter $k \in \mathbb{N}_\infty$ without affecting the size of the resulting system $G_{k\text{-}SO}$.

Theorem 5.26. *A DES G_{CSO} is current-state opaque with respect to Q_S, Q_{NS}, and P if and only if the DES $G_{k\text{-}SO}$ obtained by Transformation 5.25 is weakly k-step opaque, for any parameter $k \in \mathbb{N}_\infty$, with respect to Q_S, Q_{NS}, and P'.* □

In Theorem 4.8, we showed that the problem of deciding current-state opacity of a DES with a single observable event is coNP-complete. Transformation 5.25 allows us to generalize the hardness part of this result to weak k-step opacity. Unlike strong k-step opacity, the weak notion remains coNP-hard even for instances with the parameter $k \geq |Q|$, and therefore we can consider k to encoded in binary in the following corollary.

Corollary 5.27. *Given a natural number k represented by $O(\log(k))$ bits and a DES G with a single observable event. The problem of deciding whether the system G satisfies weak k-step opacity is* coNP*-hard.*

5.6 k-SO to CSO

In this section, we discuss the transformations from weak k-step opacity to current-state opacity. The general transformation takes place in four steps, each of which is described in a separate subsection. Initially, we show how to transform weak ∞-step opacity to current-state opacity in Subsection 5.6.1. The construction of a k-step counter automaton of size polynomial in the logarithm of k is described in Subsection 5.6.2. The general transformation from weak k-step opacity to current-state opacity for systems that allow neutral states is presented in Subsection 5.6.3. In Subsection 5.6.4, we further modify the previous transformation so that the resulting system does not use neutral states. Since the general transformation relies on binary encoding of observable events by Transformation 4.3, we provide separate transformations for systems with a single observable event in Subsections 5.6.5 and 5.6.6. Again, we distinquish two cases depending on the presence of neutral states in the system.

5.6.1 ∞-SO to CSO

Let the weak ∞-step opacity problem be represented by a DES $G_{\infty\text{-}SO}$. We transform it to a DES G_{CSO} in such a way that $G_{\infty\text{-}SO}$ is weakly ∞-step opaque if and only if G_{CSO} is current-state opaque. Our transformation proceeds in two steps:

1. We construct a DES G_{CSO} with one additional observable event @ using Transformation 5.28.
2. We use Transformation 4.3 to reduce the number of observable events of G_{CSO} by one.

Since the second step follows from Transformation 4.3, we only describe the first step, that is, the construction of G_{CSO} over $\Sigma \cup \{@\}$.

Transformation 5.28. Let $G_{\infty\text{-}SO} = (Q, \Sigma, \delta, I)$ be a DES with the secret states Q_S, the non-secret states Q_{NS}, and the corresponding projection $P\colon \Sigma^* \to \Sigma_o^*$. We construct a DES

$$G_{CSO} = (Q \cup Q^+ \cup Q^-, \Sigma \cup \{@\}, \delta', I)$$

by creating two disjoint copies of the system $G_{\infty\text{-}SO}$, denoted by G^+ and G^-, with disjoint state sets $Q^+ = \{q^+ \mid q \in Q\}$ and $Q^- = \{q^- \mid q \in Q\}$, and with an additional observable event @ that connects the system $G_{\infty\text{-}SO}$ to the copies G^+ and G^- by transitions $(p, @, p^+)$, for every secret state $p \in Q_S$, and $(q, @, q^-)$, for every non-secret state $q \in Q_{NS}$, see Figure 5.8.

We define the projection $P'\colon (\Sigma \cup \{@\})^* \to (\Sigma_o \cup \{@\})^*$, and the sets of secret states $Q'_S = Q^+$ and of non-secret states $Q'_{NS} = Q \cup Q^-$. ◇

Notice that Transformation 5.28 can be done in polynomial time using no neutral states and that it preserves determinism.

The following theorem justifies the correctness of Transformation 5.28.

Theorem 5.29. *A DES $G_{\infty\text{-}SO}$ is weakly ∞-step opaque with respect to Q_S, Q_{NS}, and P if and only if the DES G_{CSO} obtained by Transformation 5.28 is current-state opaque with respect to Q'_S, Q'_{NS}, and P'.* □

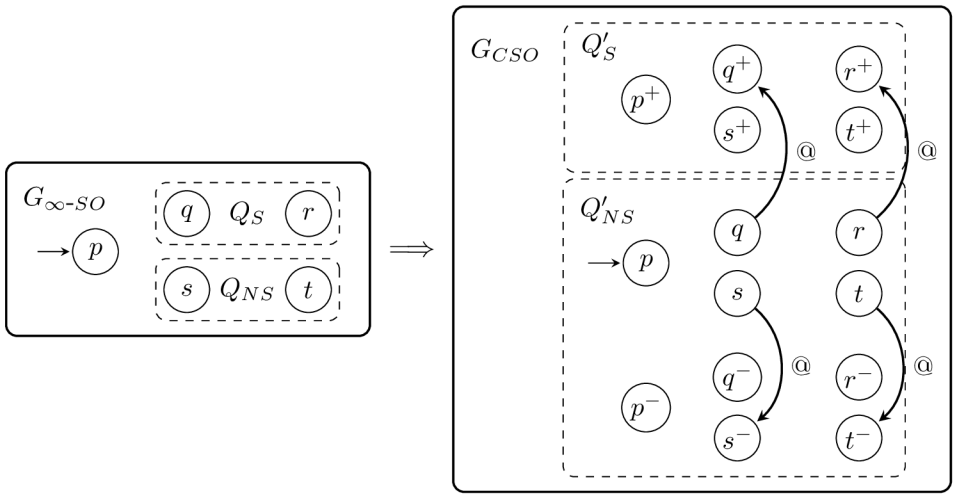


Figure 5.8: Transforming weak ∞-step opacity to current-state opacity.

We now apply our transformations to solve the open problem concerning the complexity of deciding weak ∞-step opacity. Transformation 5.28 allows us to transform an instance of weak ∞-step opacity decision problem to a current-state opacity decision problem that can be solved in polynomial space. Combined with the PSPACE-hardness of weak ∞-step opacity from Corollary 5.19, we can generalize Theorem 4.7 for weak ∞-step opacity.

Corollary 5.30. *The problem of deciding whether a DES satisfies weak ∞-step opacity is* PSPACE*-complete. The problem remains* PSPACE*-complete even if the system is a DFA with three events, one of which is unobservable.*

5.6.2 k-step counter

Before proceeding to the general transformation for weak k-step opacity, we define an automaton to serve as a k-step counter. Informally, we construct an NFA $\mathcal{A}_k$ of size polynomial in the logarithm of k such that the observer of the automaton $\mathcal{A}_k$ has a unique path of length k consisting solely of non-marked states, while all the other states are marked. This path plays the role of a k-step counter, which is essential in the following transformations.

Theorem 5.31. *For every integer $k \geq 1$, there is an NFA $\mathcal{A}_k$ with $n = \lceil \log_2(k+1) \rceil$ events and $2n+1$ states, such that the automaton $\mathcal{A}_k$ marks all strings except for the unique string W_k of length k and all its prefixes.* □

We now describe the construction of the automaton $\mathcal{A}_k$ from Theorem 5.31. Let $k \geq 1$ be given, and let $n = \lceil \log_2(k+1) \rceil$. We recursively define the string Z_n over the alphabet $\Sigma_n = \{a_0, a_1, \ldots, a_{n-1}\}$ as follows:

$$Z_1 = a_0 \quad \text{and} \quad Z_i = Z_{i-1} a_{i-1} Z_{i-1} \text{ for } 1 < i \leq n\,.$$

For example, $Z_3 = Z_2 a_2 Z_2 = Z_1 a_1 Z_1 a_2 Z_1 a_1 Z_1 = a_0 a_1 a_0 a_2 a_0 a_1 a_0$. Such strings are known in the literature as *Zimin words*, and it is well-known that the string Z_n is of length $2^n - 1$.We denote the suffix of the string Z_n of length k by W_k. Since the string Z_n is a

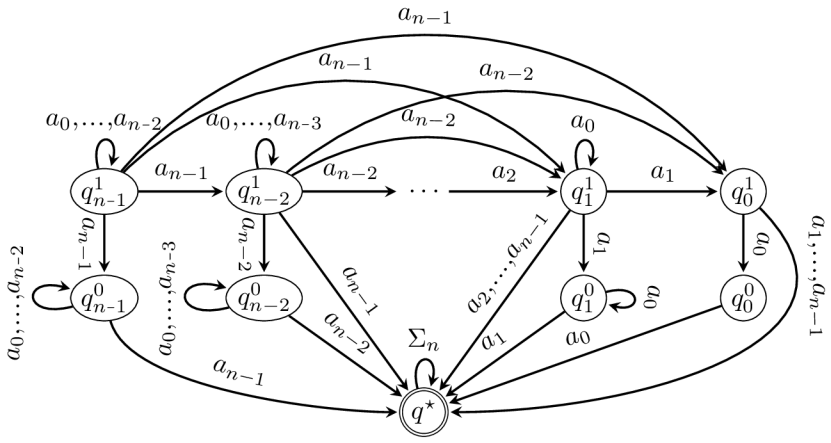


Figure 5.9: The NFA $\mathcal{A}_k$ of Theorem 5.31.

palindrome, the same event appears on positions ℓ and $2^n - 1 - \ell$. For instance, since 2 is encoded as 10 in binary, the event at the second positions from both sides of the string Z_3 is a_1.

Let $b_{n-1}b_{n-2}\cdots b_0$ be the binary representation of k, that is, $k = b_{n-1} \cdot 2^{n-1} + b_{n-2} \cdot 2^{n-2} + \cdots + b_0 \cdot 2^0$, where the leftmost bit is the most significant bit; in particular, we have that $b_{n-1} = 1$. We construct the NFA

$$\mathcal{A}_k = (Q, \Sigma_n, \delta, I, F)$$

where the set of states $Q = \{q^\star\} \cup \{q_i^1, q_i^0 \mid i = 0, \ldots, n-1\}$ consists of the state $q^\star$ and of two states q_i^1 and q_i^0 for every bit b_i of the binary representation of k; the state $q^\star$ is the only marked state, that is, $F = \{q^\star\}$; and the transition function δ is defined as follows, see Figure 5.9 for an illustration:

1. For every event $a \in \Sigma_n$, the self-loop $(q^\star, a, q^\star) \in \delta$;
2. For every state q_i^1,
   (a) the transition $(q_i^1, a_i, q_i^0) \in \delta$;
   (b) the self-loop $(q_i^1, a_j, q_i^1) \in \delta$, for $0 \le j \le i-1$;
   (c) the transition $(q_i^1, a_i, q_j^1) \in \delta$, for $0 \le j \le i-1$;
   (d) the transition $(q_i^1, a_j, q^\star) \in \delta$, for $i+1 \le j \le n-1$;
3. For every state q_i^0,
   (a) the transition $(q_i^0, a_i, q^\star) \in \delta$;
   (b) the self-loop $(q_i^0, a_j, q_i^0) \in \delta$, for $0 \le j \le i-1$;
   (c) the other transitions are undefined.

Finally, the set of initial states is defined as the set

$$I = \{q_{n-1}^{b_{n-1}}, q_{n-2}^{b_{n-2}}, \ldots, q_0^{b_0}\}$$

corresponding to the states encoding k in binary. The automaton $\mathcal{A}_k$ marks all strings over the alphabet Σ_n other than the prefixes of the string W_k.

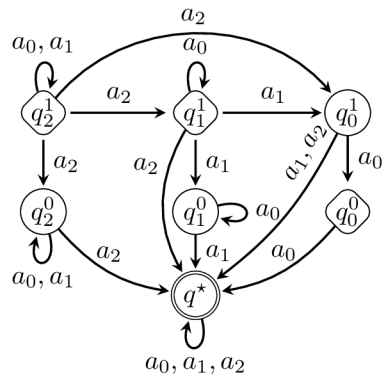


Figure 5.10: The NFA $\mathcal{A}_6$, where the initial states are diamond-shaped.

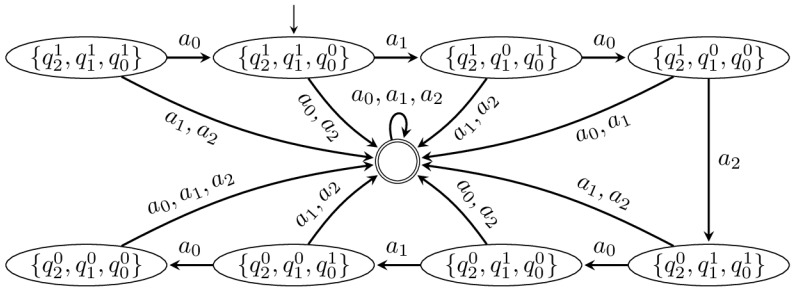


Figure 5.11: The minimized observer of the NFA $\mathcal{A}_6$ showing the behavior of the NFA $\mathcal{A}_6$ on the strings Z_3 and W_6. The initial state of the automaton $\mathcal{A}_6$ is denoted by the little arrow from above.

Example 5.32. We consider $k = 6$, for which $n = 3$ and the binary encoding of 6 is 110. Since the string $Z_3 = a_0a_1a_0a_2a_0a_1a_0$, its suffix of length 6 is the string $W_6 = a_1a_0a_2a_0a_1a_0$. The automaton $\mathcal{A}_6$ is depicted in Figure 5.10, where the initial states are q_2^1, q_1^1, and q_0^0 corresponding to the bits of 110. For the computation of the automaton $\mathcal{A}_6$ on the string $W_6 = a_1a_0a_2a_0a_1a_0$, see the observer of the automaton $\mathcal{A}_6$ depicted in Figure 5.11. It is clear from the observer that the automaton $\mathcal{A}_6$ does not mark any prefix of the string $W_6 = a_1a_0a_2a_0a_1a_0$, and that it marks all strings different from the string W_6. $\diamond$

5.6.3 The general case with neutral states

Even though the DES G_{CSO} that results from Transformation 5.28 applied to a system $G_{\infty\text{-}SO}$ can verify weak ∞-step opacity of the system $G_{\infty\text{-}SO}$ by checking current-state opacity of the system G_{CSO}, it is not suitable to verify weak k-step opacity of the system $G_{\infty\text{-}SO}$; indeed, the system G_{CSO} verifies any number of steps from the visited secret state rather than at most k steps. To overcome this issue, we extend Transformation 5.28 by adding a counter that allows us to count up to k observable events from a visited secret state.

However, we cannot simply add k states to model the counter, because adding k states requires k steps in the transformation, which is exponential in the size (the number of bits) of the binary representation of k. Instead, we model the counter with the help of the automaton $\mathcal{A}_k$ from Theorem 5.31 that can be constructed in time $O(\log^2(k))$.

Let the weak k-step opacity problem be represented by a DES $G_{k\text{-}SO}$. We transform it to a DES G_{CSO} in such a way that $G_{k\text{-}SO}$ is weakly k-step opaque if and only if G_{CSO} is current-state opaque.

Transformation 5.33. Let $G_{k\text{-}SO} = (Q, \Sigma, \delta, I)$ be a DES with the secret states Q_S, the non-secret states Q_{NS}, the corresponding projection $P\colon \Sigma^* \to \Sigma_o^*$, and the parameter $k \in \mathbb{N}$. We construct a DES

$$G_{CSO} = (Q', \Sigma', \delta', I)$$

consisting of the original system $G_{k\text{-}SO}$ along with its two modified copies and a k-step counter automaton. In more detail, we consider:

- two disjoint copies G^+ and G^- of the system $G_{k\text{-}SO}$, as in Transformation 5.28, with disjoint state sets $Q^+ = \{q^+ \mid q \in Q\}$ and $Q^- = \{q^- \mid q \in Q\}$, respectively, and
- the k-step counter automaton $\mathcal{A}_k$ constructed in Theorem 5.31.

By construction, $\mathcal{A}_k$ is of size polynomial in the logarithm of k, and its observer has a unique path of length k consisting solely of non-marked states, while all the other states are marked. However, before we connect the automata $G_{k\text{-}SO}$, G^+, G^-, and $\mathcal{A}_k$ together, we note that the automata $G_{k\text{-}SO}$, G^+, and G^- are over the alphabet Σ, while the automaton $\mathcal{A}_k$ is over the alphabet Σ_n, which is disjoint from Σ. Therefore, we change the alphabets of the automata to

$$\tilde{\Sigma} = \Sigma \cup (\Sigma_o \times \Sigma_n)\,.$$

Namely, in G^+ and G^-, we replace every *observable* transition (p, α, q) by $|\Sigma_n|$ transitions $(p, (\alpha, \beta), q)$, for every event $\beta \in \Sigma_n$, and we denote the results by $\tilde{G}^+$ and $\tilde{G}^-$. Similarly, in the automaton $\mathcal{A}_k$, we replace every transition (p, β, q) by $|\Sigma_o|$ transitions $(p, (\alpha, \beta), q)$, for every observable event $\alpha \in \Sigma_o$, and we denote the result by $\tilde{\mathcal{A}}_k$.

Now, we construct a DES

$$G_{CSO} \text{ as a disjoint union of the automata } G_{k\text{-}SO},\ \tilde{G}^+,\ \tilde{G}^-, \text{ and } \tilde{\mathcal{A}}_k,$$

over alphabet $\Sigma' = \tilde{\Sigma} \cup \{@\}$. We connect the parts of G_{CSO} with the transitions $(p, @, p^+)$ and $(p, @, q_0)$, for every secret state $p \in Q_S$ and every initial state $q_0 \in I$ of $\tilde{\mathcal{A}}_k$, and the transitions $(q, @, q^-)$, for every non-secret state $q \in Q_{NS}$, cf. Figure 5.12.

We define the projection $P'\colon (\tilde{\Sigma} \cup \{@\})^* \to (\Sigma_o \cup \{@\} \cup \Sigma_o \times \Sigma_n)^*$, and the sets of secret states $Q'_S = Q^+$ and of non-secret states $Q'_{NS} = Q^- \cup \{q^\star\}$, where $q^\star$ is the unique marked state of $\tilde{\mathcal{A}}_k$. The other states are neutral. ◇

Notice that Transformation 5.33 can be done in polynomial time in the size of the system and in the number of bits of the binary representation of k.

In G_{CSO}, every event after generating the event @ is either unobservable or pair of events of $\Sigma_o \times \Sigma_n$. Therefore, in the sequel we denote strings over $\Sigma_{uo} \cup \Sigma_o \times \Sigma_n$, such as $s = u(a, x)u(b, y)$, simply as a pair of the form $\Sigma^* \times \Sigma_n^*$ of concatenated strings of the corresponding alphabets, such as $s = (uaub, xy)$, where $u \in \Sigma_{uo}$, $a, b \in \Sigma_o$, and $x, y \in \Sigma_n$.

The following theorem justifies the correctness of Transformation 5.33.

Theorem 5.34. *A DES $G_{k\text{-}SO}$ is weakly k-step opaque with respect to Q_S, Q_{NS}, and P if and only if the DES G_{CSO} obtained by Transformation 5.33 is current-state opaque with respect to Q'_S, Q'_{NS}, and P'.* □

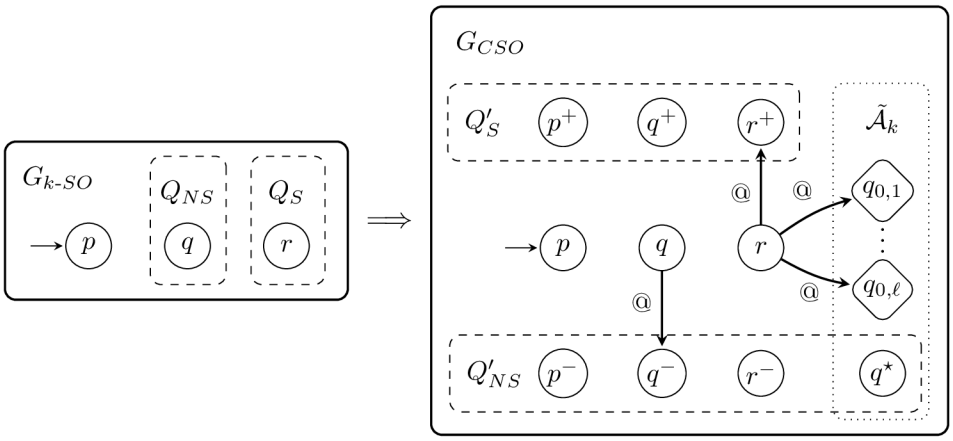


Figure 5.12: Transforming weak k-step opacity to current-state opacity with neutral states; the initial states of $\tilde{\mathcal{A}}_k$ are diamond-shaped.

5.6.4 The general case without neutral states

Finally, we show how to transform weak k-step opacity to current-state opacity without employing neutral states by modifying Transformation 5.33.

Transformation 5.35. Let $G_{k\text{-}SO} = (Q, \Sigma, \delta, I)$ be a DES with the secret states Q_S, the non-secret states Q_{NS}, the corresponding projection $P\colon \Sigma^* \to \Sigma_o^*$, and the parameter $k \in \mathbb{N}$. We first perform Transformation 5.33 on $G_{k\text{-}SO}$ to obtain automata $\tilde{G}^+$, $\tilde{G}^-$, and $\tilde{\mathcal{A}}_k$ over alphabet $\tilde{\Sigma}$. Now, we make all the states of the automaton $\tilde{G}^+$ initial and marked, and synchronize the computations of the automata $\tilde{G}^+$ and $\tilde{\mathcal{A}}_k$ by their synchronous product $\tilde{G}^+ \| \tilde{\mathcal{A}}_k$. We construct a DES

$$G_{CSO} = (Q', \Sigma', \delta', I) \text{ as a disjoint union of } G_{k\text{-}SO},\ \tilde{G}^-, \text{ and } \tilde{G}^+ \| \tilde{\mathcal{A}}_k,$$

connected together by transitions under a new observable event @ as follows:

1. we add transition $(q, @, (q^+, q_0))$ to δ', for every secret state $q \in Q_S$ and every initial state (q^+, q_0) of $\tilde{G}^+ \| \tilde{\mathcal{A}}_k$, and
2. we add transition $(q, @, q^-)$ to δ', for every non-secret state $q \in Q_{NS}$.

We define the projection $P'\colon (\tilde{\Sigma} \cup \{@\})^* \to (\Sigma_o \cup \{@\} \cup \Sigma_o \times \Sigma_n)^*$. The secret states Q'_S of the system G_{CSO} are defined as the non-marked states of the system $\tilde{G}^+ \| \tilde{\mathcal{A}}_k$, that is, the states of $\tilde{G}^+ \| \tilde{\mathcal{A}}_k$ where second part is not equal to $q^\star$. All the other states are non-secret, that is, $Q'_{NS} = Q' - Q'_S$. ◇

Notice that Transformation 5.35 can be done in polynomial time in the size of the system and in the number of bits of the binary representation of the parameter k. Since this transformation does not preserve determinisn and the number of the observable events, we apply Transformations 4.3 and 4.5 on the resulting system G_{CSO} to reduce its number of observable events and to determinize it.

The following theorem justifies the correctness of Transformation 5.35.

Theorem 5.36. *A DES $G_{k\text{-}SO}$ is weakly k-step opaque with respect to Q_S, Q_{NS}, and P if and only if the DES G_{CSO} obtained by Transformation 5.35 is current-state opaque with respect to Q'_S, Q'_{NS}, and P'.* □

We now apply our transformations to solve the open problem concerning the complexity of deciding weak k-step opacity. Transformation 5.35 allows us to transform an instance of weak k-step opacity decision problem to a current-state opacity decision problem that can be solved in polynomial space. Combined with the PSPACE-hardness of weak k-step opacity from Corollary 5.19, we can generalize Theorem 4.7 for weak k-step opacity.

Corollary 5.37. *Given a natural number k represented by $O(\log(k))$ bits and a DES G. The problem of deciding whether the system G satisfies weak k-step opacity is* PSPACE*-complete. The problem remains* PSPACE*-complete even if the system G is a DFA with three events, one of which is unobservable.*

5.6.5 The case of $|\Sigma_o| = 1$ with neutral states

To preserve the number of observable events, our transformation of weak k-step opacity to current-state opacity relies on binary encoding by Transformation 4.3. This transformation requires at least two observable events in $G_{k\text{-}SO}$, and hence it is not applicable to systems with a single observable event. For these systems, we provide two different transformations. First one, which allows neutral states, requires to add at most a quadratic number of new states.

Let the weak k-step opacity problem with a single observable event be represented by a DES $G^a_{k\text{-}SO}$. We transform it to a DES G^a_{CSO} in such a way that $G^a_{k\text{-}SO}$ is weakly k-step opaque if and only if G^a_{CSO} is current-state opaque.

Transformation 5.38. Let $G^a_{k\text{-}SO} = (Q, \Sigma, \delta, I)$ be a DES with a single observable event $\Sigma_o = \{a\}$, the secret states Q_S, the non-secret states Q_{NS}, the corresponding projection $P\colon \Sigma^* \to \{a\}^*$, and the parameter $k \in \mathbb{N}$. We construct a DES

$$G_{CSO} = (Q', \Sigma, \delta', I)$$

where δ' is initialized as δ and modified as follows using the function φ_k from Definition 2.5. For every state $p \in Q$ with $\varphi_k(p) > 0$, we add k new states $p_1, \ldots, p_k$ to Q' and k new transitions (p, a, p_1) and (p_i, a, p_{i+1}), for $i = 1, \ldots, k-1$, to δ'. Finally, we replace every observable transition (p, a, r) in δ' by the transition (p_k, a, r). We initialize the sets $Q'_S := Q_S$ and $Q'_{NS} := Q_{NS}$. For every state $p \in Q_S$ with $\varphi_k(p) = \ell > 0$, we add the corresponding states $p_1, \ldots, p_\ell$ to Q'_S and, for every $q \in Q_{NS}$ with $\varphi_k(q) = \ell > 0$, we add $q_1, \ldots, q_\ell$ to Q'_{NS}. ◇

Notice that Transformation 5.38 preserves determinism and, by the following remark, requires to add at most n^2 states, and hence it can be done in polynomial time.

Remark 5.39. *It follows directly from Definition 2.5 of the function φ_k that if $k \geq |Q|$, then a system with a single observable event is weakly k-step opaque if and only if it is weakly ∞-step opaque. Therefore, we may consider $k \leq |Q|$, which also covers the case of weak ∞-step opacity.*

The following theorem justifies the correctness of Transformation 5.38.

Theorem 5.40. *A DES $G^a_{k\text{-}SO}$ with a single observable event is weakly k-step opaque with respect to Q_S, Q_{NS}, and P if and only if the DES G^a_{CSO} obtained by Transformation 5.38 is current-state opaque with respect to Q'_S, Q'_{NS}, and P.* □

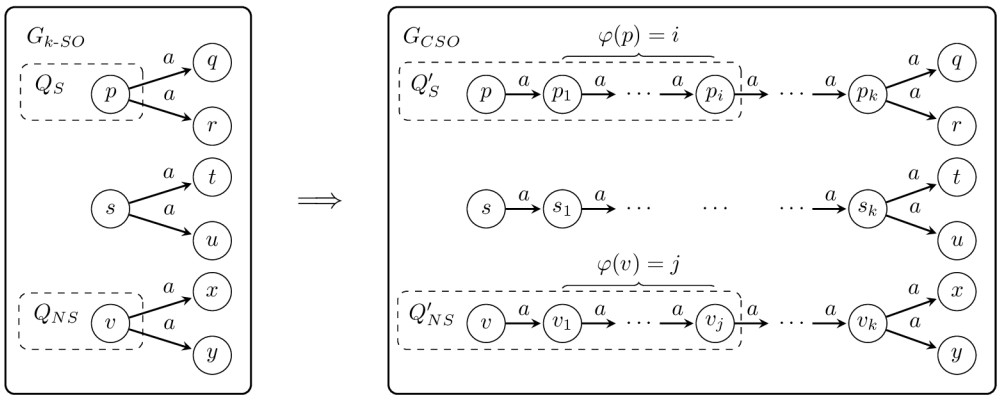


Figure 5.13: Transforming weak k-step opacity with a single observable event to current-state opacity.

Transformation 5.38 allows us to transform an instance of weak k-step opacity decision problem to a current-state opacity decision problem, while preserving a single observable event. Combined with the coNP-hardness of weak k-step opacity with a single observable event from Corollary 5.27, we can generalize Theorem 4.8 for weak k-step opacity. Additionally, Remark 5.39 allows us to state the same result for weak ∞-step opacity.

Corollary 5.41. *Given a natural number k represented by $O(\log(k))$ bits and a DES G with a single observable event. The problem of deciding whether the system G satisfies weak k-step opacity is* coNP*-complete. Analogously, the problem of deciding whether the system G satisfies weak ∞-step opacity is* coNP*-complete.*

5.6.6 The case of $|\Sigma_o| = 1$ without neutral states

To avoid introducing new neutral states into the system as in the previous transformation, we provide a separate transformation for cases where such states are not allowed.

Let the weak k-step opacity problem with a single observable event be represented by a DES G^a without neutral states. Since the following transformation does not change the structure of the system, we denote both the original and the resulting system simply by G^a. We transform the sets of secret and non-secret states of G^a in such a way that G^a is weakly k-step opaque with respect to Q_S and Q_{NS} if and only if G^a is current-state opaque with respect to Q'_S and Q'_{NS}.

Transformation 5.42. Let $G^a = (Q, \Sigma, \delta, I)$ be a DES with a single observable event $\Sigma_o = \{a\}$, the secret states Q_S, the non-secret states $Q_{NS} = Q - Q_S$, the corresponding projection $P\colon \Sigma^* \to \{a\}^*$, and the parameter $k \in \mathbb{N}$. We construct sets Q'_S and Q'_{NS} as follows. We determine (in linear time) whether the language $P(L(G^a))$ is finite.

(A) If so, we verify weak k-step opacity of the system G^a in linear time by checking the subsets of states $\delta(I, P^{-1}(a^i))$, for every $i \leq |Q| - 1$.

  (A1) If the system G^a is weakly k-step opaque, and hence also current-state opaque, we define the sets of secret states $Q'_S = Q_S$ and of non-secret states $Q'_{NS} = Q_{NS}$.

  (A2) If the system G^a is not weakly k-step opaque, we define the sets of secret states $Q'_S = Q$ and of non-secret states $Q'_{NS} = \emptyset$.

(B) If the language $P(L(G^a))$ is infinite, we define the set of non-secret states $Q'_{NS} = \{q \in Q_{NS} \mid \varphi_k(q) = k\}$ using the function φ_k from Definition 2.5, which assings to the state q the maximal number of observable steps that are possible from state q. Finally, we define the set of secret states to be $Q'_S = Q - Q'_{NS}$. ◇

Notice that Transformation 5.42 can be done in polynomial time and that it does not change the structure of the system in any way. Analogously to Transformation 5.38, we can consider $k \leq |Q|$ by Remark 5.39.

The following theorem justifies the correctness of Transformation 5.42.

Theorem 5.43. *A DES G^a with a single observable event $\Sigma_o = \{a\}$ is weakly k-step opaque with respect to Q_S, Q_{NS}, and P if and only if G^a is current-state opaque with respect to Q'_S, Q'_{NS}, and P defined by Transformation 5.42.* □

5.7 k-SSO to k-SO

In this section, we show how to transform strong k-step opacity to weak k-step opacity. Our transformation proceeds in two steps. The first step of the transformation is called *normalization* and we also use it to describe the relationship between strong 0-step opacity and current-state opacity. The second step then transforms the normalized system to the weak k-step opacity instance.

In what follows, we call the systems where there are no unobservable transitions from secret states to non-secret states *normal*. For systems that are not normal, we provide a construction to normalize them, that is, we eliminate unobservable transitions from secret states to non-secret states without affecting the property of being strongly k-step opaque.

Transformation 5.44. Let $G = (Q, \Sigma, \delta, q_0)$ be a deterministic DES with the secret states Q_S, the non-secret states $Q_{NS} = Q - Q_S$, the corresponding projection $P\colon \Sigma^* \to \Sigma_o^*$, and the parameter $k \in \mathbb{N}_\infty$. We construct a DES

$$G_{norm} = (Q_n, \Sigma, \delta_n, q_0)$$

where $Q_n = Q \cup Q'$ for $Q' = \{q' \mid q \in Q\}$ being a disjoint copy of Q, and the transition function δ_n is defined as follows. We initialize $\delta_n := \delta$ and further modify it in the following four steps:

1. For every $p \in Q_S$, $q \in Q_{NS}$, and $u \in \Sigma_{uo}$, we replace the transition (p, u, q) by (p, u, q') in δ_n.
2. For every unobservable transition (p, u, q) in δ, that is, $u \in \Sigma_{uo}$, we add the transition (p', u, q') to δ_n.
3. For every observable transition (q, a, r) in δ, that is, $a \in \Sigma_o$, we add the transition (q', a, r) to δ_n.
4. We remove unreachable states and corresponding transitions.

The set of secret states of G_{norm} is the set $Q_n^S = Q_S \cup Q'$. The set of non-secret states Q_{NS} remains unchanged. ◇

In the sequel, we call G_{norm} the *normalization* of G. If G and G_{norm} coincide, we say that G is *normal*.

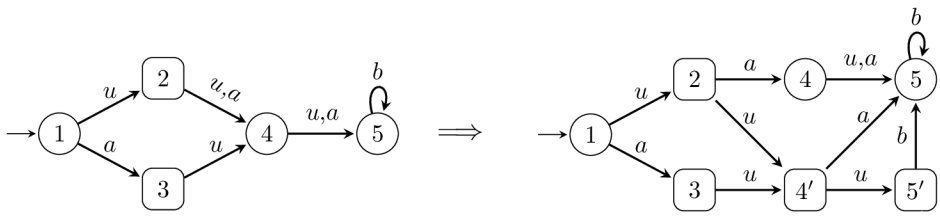


Figure 5.14: A deterministic DES G (left) and its normalization G_{norm} (right); the secret states are squared.

Example 5.45. To illustrate Transformation 5.44, consider the system depicted in Figure 5.14 (left). Its normalization G_{norm} is depicted in the same figure (right). States 2 and 3 of G are secret, events a and b are observable, and u is unobservable. The normalization G_{norm} of G initially contains five new secret states $1'$, $2'$, $3'$, $4'$, $5'$. Step (1) of Transformation 5.44 replaces transitions $(2, u, 4)$ and $(3, u, 4)$ by $(2, u, 4')$ and $(3, u, 4')$, respectively, step (2) adds four unobservable transitions $(1', u, 2')$, $(2', u, 4')$, $(3', u, 4')$, and $(4', u, 5')$, and step (3) adds the observable transitions $(1', a, 3)$, $(2', a, 4)$, $(4', a, 5)$ and $(5', b, 5)$. Finally, step (4) eliminates unreachable states $1'$, $2'$, $3'$, and the corresponding transitions. ◇

The following lemma describes the meaning of normalization and states the main properties of a normalized DES.

Lemma 5.46. *For a deterministic DES $G = (Q, \Sigma, \delta, q_0)$ with the secret states Q_S, the non-secret states $Q_{NS} = Q - Q_S$, the observation $P\colon \Sigma^* \to \Sigma_o^*$, and the parameter $k \in \mathbb{N}_\infty$, let G_{norm} be the normalization of G obtained by Transformation 5.44. Then, the following holds true:*

1. *$L(G) = L(G_{norm})$;*
2. *G_{norm} is deterministic;*
3. *In G_{norm}, there is no non-secret state reachable from a secret state by a sequence of unobservable events, i.e., $\delta_n(Q_n^S, P^{-1}(\varepsilon)) \cap (Q_n - Q_n^S) = \emptyset$;*
4. *G is strongly k-step opaque with respect to Q_S and P if and only if G_{norm} is strongly k-step opaque with respect to Q_n^S and P.* □

In the following theorem, we discuss the relationship between strong 0-step opacity and weak 0-step (current-state) opacity for normal deterministic DESs.

Theorem 5.47. *A normal deterministic DES $G = (Q, \Sigma, \delta, q_0)$ is strongly 0-step opaque with respect to Q_S and P if and only if G is weakly 0-step opaque with respect to Q_S, $Q_{NS} = Q - Q_S$, and P.* □

Let the strong k-step opacity problem be represented by a DES $G_{k\text{-}SSO}$. We transform it to a DES $G_{k\text{-}SO}$ in such a way that $G_{k\text{-}SSO}$ is strongly k-step opaque if and only if $G_{k\text{-}SO}$ is weakly k-step opaque. In the construction, we assume that $G_{k\text{-}SSO}$ is a normal deterministic DES. By Lemma 5.46, this assumption is without loss of generality, because if $G_{k\text{-}SSO}$ is not normal, then we can consider its normalization instead.

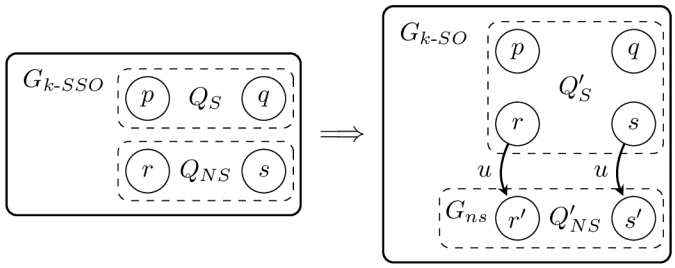


Figure 5.15: Transforming strong k-step opacity to weak k-step opacity.

Transformation 5.48. Let $G_{k\text{-}SSO} = (Q, \Sigma, \delta, q_0)$ be a normal deterministic DES with the secret states Q_S, the non-secret states $Q_{NS} = Q - Q_S$, the corresponding projection $P\colon \Sigma^* \to \Sigma_o^*$, and the parameter $k \in \mathbb{N}_\infty$. We construct a DES

$$G_{k\text{-}SO} = (Q \cup Q'_{NS}, \Sigma \cup \{u\}, \delta', q_0)$$

as a disjoint union of $G_{k\text{-}SSO}$ and $G_{ns} = (Q'_{NS}, \Sigma, \delta_{ns}, q'_0)$, where G_{ns} is obtained from $G_{k\text{-}SSO}$ by removing all secret states and corresponding transitions, and $Q'_{NS} = \{q' \mid q \in Q_{NS}\}$ is a copy of Q_{NS} disjoint from Q. We use a new unobservable event u to connect G_{ns} to $G_{k\text{-}SSO}$ so that we initialize $\delta' := \delta \cup \delta_{ns}$ and extend δ' by additional transitions (q, u, q') for every $q \in Q_{NS}$, cf. Figure 5.15 for an illustration. The states of Q'_{NS} are the only non-secret states of $G_{k\text{-}SO}$, that is, the set of secret states of $G_{k\text{-}SO}$ is the set $Q'_S = Q$. Finally, we define the projection $P'\colon (\Sigma \cup \{u\})^* \to \Sigma_o^*$. ◇

Notice that both Transformations 5.44 and 5.48 can be done in polynomial time and that they preserve determinism and the number of observable events. In addition, they are independent of the parameter k, and hence they work for any $k \in \mathbb{N}_\infty$ without affecting the size of the resulting system $G_{k\text{-}SO}$.

The following theorem justifies the correctness of Transformation 5.48.

Theorem 5.49. *A normal deterministic DES $G_{k\text{-}SSO}$ is strongly k-step opaque with respect to Q_S and P if and only if the DES $G_{k\text{-}SO}$ obtained by Transformation 5.48 is weakly k-step opaque with respect to Q'_S, Q'_{NS}, and P'.* □

We now apply our transformations to solve the open problem concerning the complexity of deciding strong k-step opacity. Transformation 5.48 allows us to transform an instance of strong k-step opacity decision problem to a weak k-step opacity decision problem. Combined with the PSPACE-hardness of strong k-step opacity from Corollary 5.19 and PSPACE-completeness of weak k-step opacity from Corollary 5.37, we can generalize Theorem 4.7 for strong k-step opacity.

Corollary 5.50. *Given a natural number k represented by $O(\log(k))$ bits and a DES G. The problem of deciding whether the system G satisfies strong k-step opacity is* PSPACE*-complete. The problem remains* PSPACE*-complete even if the system G is a DFA with three events, one of which is unobservable.*

Analogously, we generalize Theorem 4.8 for systems with a single observable event using Transformation 5.48 together with coNP-hardness of strong k-step opacity from Corollary 5.24 and coNP-completeness of weak k-step opacity from Corollary 5.27.

Corollary 5.51. *Given a natural number k represented in unary and a DES G with a single observable event. The problem of deciding whether the system G satisfies strong k-step opacity is* coNP*-complete.*

Chapter 6

Verification of opacity

In this chapter, we introduce three new algorithms for verifying language-based opacity and trace opacity (Algorithm 1), weak k-step opacity (Algorithm 2), and strong k-step opacity (Algorithm 4). Note that our algorithms for k-step notions are applicable with the parameter $k = \infty$, and thus can also verify weak and strong ∞-step opacity.

Each section contains an analysis of the complexity of the proposed algorithm, as well as a comparison with previously existing results.

6.1 Verification of LBO and TO

The algorithmic complexity of deciding whether a given DES is language-based opaque with respect to given secret and non-secret languages has been investigated in the literature. Lin [29] suggested an algorithm with the complexity $O(2^{2n})$, where n is the order of the state spaces of the automata representing the secret and non-secret languages. The same complexity has been achieved by Wu and Lafortune [42] using the transformation to current-state opacity. We improve this complexity with Algorithm 1.

The language-based opacity verification problem consists of a DES G, a projection $P\colon \Sigma^* \to \Sigma_o^*$, a secret language $L_S = L(\mathcal{A}_S)$ given by the non-blocking automaton $\mathcal{A}_S$, and a non-secret language $L_{NS} = L(\mathcal{A}_{NS})$ given by the non-blocking automaton $\mathcal{A}_{NS}$. The complexity improvement of Algorithm 1 comes from solving the language inclusion problem $P(L_S) \subseteq P(L_{NS})$ by the intersection of the projected automaton $P(\mathcal{A}_S)$ with the observer co-$\mathcal{A}_{NS}^{obs}$, instead of the intersection of two observer structures as in [29].

We now discuss the complexity of our algorithm.

Theorem 6.1. *The space and time complexity of Algorithm 1 is $O(n_1 2^{n_2})$ and $O((n_1 + m)2^{n_2})$, respectively, where n_1 is the number of states of the automaton $\mathcal{A}_S$, n_2 is the number of states of the automaton $\mathcal{A}_{NS}$, and m is the number of transitions of $P(\mathcal{A}_S)$. In particular, $m \leq \ell n_1^2$, where ℓ is the number of observable events.*

Proof. The projected automaton $P(\mathcal{A}_S)$ has n_1 states and m transitions, and co-$\mathcal{A}_{NS}^{obs}$, has at most 2^{n_2} states and $\ell 2^{n_2}$ transitions. Therefore, we search the automaton $P(\mathcal{A}_S) \cap$ co-$\mathcal{A}_{NS}^{obs}$ that has at most $O(n_1 2^{n_2})$ states and $O(m2^{n_2})$ transitions. Since $m > \ell$, the proof is complete. □

Notice that Algorithm 1 can be used to verify trace opacity, since Remark 5.14 provides a procedure for constructing automata $\mathcal{A}_S$ and $\mathcal{A}_{NS}$ from the trace opacity problem instance. Since the size of $\mathcal{A}_S$ is at most twice the size of the original n-state system and

Algorithm 1 Verification of language-based opacity

Require: A DES $G = (Q, \Sigma, \delta, I)$, automata $\mathcal{A}_S$ and $\mathcal{A}_{NS}$, and $\Sigma_o \subseteq \Sigma$
Ensure: `true` if and only if G is language-based opaque with respect to $L_S = L(\mathcal{A}_S)$, $L_{NS} = L(\mathcal{A}_{NS})$, and $P\colon \Sigma^* \to \Sigma_o^*$

1: Compute the projected automaton $P(\mathcal{A}_S)$ of $\mathcal{A}_S$
2: Compute the observer $\mathcal{A}_{NS}^{obs}$ of $\mathcal{A}_{NS}$
3: Compute the complement co-$\mathcal{A}_{NS}^{obs}$ of $\mathcal{A}_{NS}^{obs}$
4: Compute the instersection automaton $\mathcal{C} = P(\mathcal{A}_S) \cap$ co-$\mathcal{A}_{NS}^{obs}$
5: **if** $L_m(\mathcal{C}) = \emptyset$ **then**
6: **return** `true`
7: **else**
8: **return** `false`
9: **end if**

$\mathcal{A}_{NS}$ has exactly n states, we obtain the same complexity $O((n+m)2^n)$ also for verifying trace opacity.

Another use of the algorithm is verification of the special case of initial-and-final-state opacity considered in [42]. If the secret and non-secret pairs are of the form $Q_S = I_S \times F_S$ and $Q_{NS} = I_{NS} \times F_{NS}$, where $I_S, I_{NS} \subseteq I$ and $F_S, F_{NS} \subseteq Q$, then we use languages of $\mathcal{A}_S = (Q, \Sigma, \delta, I_S, F_S)$ and $\mathcal{A}_{NS} = (Q, \Sigma, \delta, I_{NS}, F_{NS})$ for the secret and non-secret languages, respectively.

6.2 Verification of k-SO

This particular version of the algorithm for verifying weak k-step opacity was presented in [9], which was itself a revision of our previous algorithm from [6]. Initially, we provide an overview of the algorithm and its use of Breadth-First Search. Following that, we analyse the time and space complexity of the algorithm and compare it with previously existing algorithms.

We remind that the weak k-step opacity verification problem consists of a DES G, a projection $P\colon \Sigma^* \to \Sigma_o^*$, a set of secret states $Q_S \subseteq Q$, a set of non-secret states $Q_{NS} \subseteq Q$, and parameter $k \in \mathbb{N}_\infty$. Note that the system may contain neutral states.

In Algorithm 2 we describe our new algorithm verifying weak k-step opacity. The idea of the algorithm is as follows. We first compute the observer of G, denoted by G^{obs}, and the projected automaton of G, denoted by $P(G)$. Then, for every reachable state X of G^{obs}, we add the pairs $(x, X \cap Q_{NS})$ to the set Y, where x is a secret state of X and $X \cap Q_{NS}$ is the set of all non-secret states of X. Intuitively, in these states, the intruder estimates that G may be in the secret state x or in the non-secret states of $X \cap Q_{NS}$. To verify that the intruder does not reveal the secret state, we need to check that every possible path of length up to k starting in x is accompanied by a path with the same observation starting in a non-secret state of $X \cap Q_{NS}$. To this end, we construct the automaton H as the part of the full observer of G consisting only of states reachable from the states forming the second components of the pairs in Y, and the automaton $\mathcal{C} = P(G) \times H$ as the product automaton of the projected automaton of G and H. In $\mathcal{C}$, all transitions are observable, and every path from a secret state x is synchronized with all the possible paths with the same observation starting in the states of $X \cap Q_{NS}$. Thus,

Algorithm 2 Verification of weak k-step opacity

Require: A DES $G = (Q, \Sigma, \delta, I)$, $Q_S, Q_{NS} \subseteq Q$, $\Sigma_o \subseteq \Sigma$, and $k \in \mathbb{N}_\infty$.
Ensure: `true` if and only if G is weakly k-step opaque with respect to Q_S, Q_{NS}, and $P\colon \Sigma^* \to \Sigma_o^*$

1: Set $Y := \emptyset$
2: Compute the observer G^{obs} of G
3: Compute the projected automaton $P(G)$ of G
4: **for** every state X of G^{obs} **do**
5: **for** every state $x \in X \cap Q_S$ **do**
6: add state $(x, X \cap Q_{NS})$ to set Y
7: **end for**
8: **end for**
9: Construct H as the part of the full observer of G accessible from the states of the second components of Y
10: Compute the product automaton $\mathcal{C} = P(G) \times H$
11: Use the Breadth-First Search (BFS) of Algorithm 3 to mark all states of $\mathcal{C}$ reachable from the states of Y in at most k steps
12: **if** $\mathcal{C}$ contains a marked state of the form $(q, \emptyset)$ **then**
13: **return** `false`
14: **else**
15: **return** `true`
16: **end if**

if there is a path from the secret state x of length up to k that is not accompanied by a path with the same observation from a state of $X \cap Q_{NS}$, then this path from the state x in $P(G)$ ends up in a state, say, q, whereas all paths in H with the same observation from the state $X \cap Q_{NS}$ end up in the state $\emptyset$. Here, $X \cap Q_{NS}$ and $\emptyset$ are understood as the states of the full observer of G. Thus, if the DES G is not weakly k-step opaque, there is a state of Y from which a state of the from $(q, \emptyset)$ is reachable in at most k steps. Therefore, we search the automaton $\mathcal{C}$ and mark all its states that are reachable from a state of Y in at most k steps. If a state of the from $(q, \emptyset)$ is marked, then G is not weakly k-step opaque; otherwise, it is.

Intuitively, the correctness follows from the fact that the BFS visits all nodes at distance d before visiting any nodes at distance $d + 1$. In other words, all states of $\mathcal{C}$ reachable from the states of Y in at most k steps are visited (and marked) before any state at distance $k+1$. The implementation of the BFS is, however, the key step to obtain the claimed complexity. Namely, the classical BFS of [17] maintains an array to store the shortest distances (aka the number of hops) of every node to an initial node. Since storing a number less than or equal to k requires $\log(k)$ bits, using the classical BFS requires the space of size $O(\log(k)n2^n)$ to store the shortest distance of every state of $\mathcal{C}$ to a state of Y, because $\mathcal{C}$ has $O(n2^n)$ states.

For our purposes, we do not need to know the shortest distance of every state to a state of Y, but we rather need to keep track of the number of hops from the states of Y made so far.

We can achieve this by modifying the classical BFS so that we do not store the shortest distances for every state of $\mathcal{C}$, but only the current distance. We store the current distance in the queue used by the BFS, see Algorithm 3. In particular, we first push number 0 to

Algorithm 3 The Breadth-First Search used in Algorithm 2

Require: A DES $G = (V, \Sigma, \delta, I)$, a set $S \subseteq V$, $k \in \mathbb{N}_\infty$
Ensure: G with all states at distance at most k from the states of S marked

```
1: Initialize the queue Q := ∅
2: Enqueue number 0 to Q
3: Mark every node s ∈ S and enqueue it to Q
4: Color every node u ∈ V − S white
5: while Q ≠ ∅ do
6:     u := DEQUEUE(Q)
7:     if u ∉ V and u = k then
8:         Terminate, states at distance ≤ k were visited
9:     else if u ∉ V and u < k then
10:        Enqueue u + 1 to Q
11:    else if u ∈ V is a state of G then
12:        for every state v reachable in one step from u do
13:            if the color of v is white then
14:                Mark state v and enqueue it to Q
15:            end if
16:        end for
17:        Color u black
18:    end if
19: end while
```

the queue, followed by all the states of Y. Assuming that $k > 0$, number 0 is processed in such a way that it is dequeued from the queue, and number 1 is enqueued. After processing all the states of Y from the queue, that is, having number 1 at the head of the queue, we know that all the elements of the queue after number 1 are the states at distance one from the states of Y and not less. The algorithm proceeds this way until it has either visited all the states of $\mathcal{C}$ or the number stored in the queue is k. The algorithm marks all states of $\mathcal{C}$ that it visits.

This approach requires to store only one $\log(k)$-bit number at a time rather than $n2^n$ such numbers, and hence the complexity of the algorithm then basically follows from the fact that the distance is bounded by the number of states of $\mathcal{C}$, and not by the parameter k.

Since Algorithm 3 is a minor modification of the BFS of Cormen et al. [17], very similar arguments show its correctness and complexity. For this reason, we do not further discuss the correctness and complexity of Algorithm 3.

We now discuss the complexity of our algorithm.

Theorem 6.2. *The space and time complexity of Algorithm 2 is $O(n2^n)$ and $O((n+m)2^n)$, respectively, where n is the number of states of the input DES G and m is the number of transitions of $P(G)$. In particular, $m \leq \ell n^2$, where ℓ is the number of observable events.*

Proof. Computing the observer and the projected NFA of G, lines 2 and 3, takes time $O(\ell 2^n)$ and $O(m+n)$, respectively. The cycle on lines 4–8 takes time $O(n2^n)$. Constructing the part H of the full observer of G, line 9, takes time $O(\ell 2^n)$. Constructing $\mathcal{C}$, line 10, takes time $O(n2^n + m2^n)$, where $O(n2^n)$ is the number of states and $O(m2^n)$ is the number of transitions of $\mathcal{C}$. The bounds come from the fact that we create at most 2^n copies of

the automaton $P(G)$. The BFS takes time linear in the size of $\mathcal{C}$, and the condition of line 11 can be processed during the BFS. Since $m \geq \ell$, the proof is complete. □

We now briefly review the complexity of existing algorithms verifying weak k-step opacity. First, notice that the complexity of existing algorithms is exponential, which seems unavoidable because the problem is PSPACE-complete by Corollary 5.37. In particular, Saboori and Hadjicostis [36] designed an algorithm with complexity $O(\ell(\ell+1)^k 2^n)$, where n is the number of states and ℓ is the number of observable events. Considering the verification of weak ∞-step opacity, Saboori and Hadjicostis [37] designed an algorithm with complexity $O(\ell 2^{n^2+n})$. Yin and Lafortune [44] introduced the notion of a two-way observer and applied it to the verification of weak k-step opacity with complexity $O(\min\{n2^{2n}, n\ell^k 2^n\})$, and to the verification of weak ∞-step opacity with complexity $O(n2^{2n})$; the formulae already include a correction by Lan et al. [28]. In [6] we designed algorithms verifying weak k-step opacity and weak ∞-step opacity with complexities $O((k+1)2^n(n+m\ell^2))$ and $O((n+m\ell)2^n)$, respectively, where $m \leq \ell n^2$ is the number of transitions in the projected automaton. These algorithms outperform the two-way observer if k is polynomial in n or larger than $2^n - 2$, since weak (2^n-2)-step opacity and weak ∞-step opacity coincide [44]. Wintenberg et al. [41] discussed and experimentally compared four approaches to the verification of weak k-step opacity based on (i) the secret observer, (ii) the reverse comparison, (iii) the state estimator, and (iv) the two-way observer. Their respective state complexities are $O(2^{n(k+3)})$, $O(n(k+1)3^n)$, $O((\ell+1)^k 2^n)$, and $O(\min\{2^n, \ell^k\}2^n)$.[1]

Notice that these bounds are formulated only in the number of states of the constructed automata, disregarding the number of transitions and the time of the construction. Therefore, the time-complexity bounds differ from the state-complexity bounds at least by the factor of ℓ, if the constructed automata are deterministic, or by a factor of $m \leq \ell n^2$ if the construction of the automaton involves an NFA, such as in the case of the reverse comparison. Namely, the time-complexity bounds are $O(\ell 2^{n(k+3)})$ for the secret observer, where n is the number of states and ℓ is the number of observable events, $O((n+m)(k+1)3^n)$ for the reverse comparison, where $m \leq \ell n^2$ is the number of transitions in an involved NFA, $O(\ell(\ell+1)^k 2^n)$ for the state estimator, and $O(\min\{n2^{2n}, n\ell^k 2^n\})$ for the two-way observer.

As the reader may notice, the above complexities depend on the parameter k. A partial exception is the two-way observer that does not depend on k if $\ell^k \geq 2^n$, that is, if k is larger than the number of states divided by the logarithm of the number of observable events.

Since the complexity of Algorithm 2 is $O((n+m)2^n)$, where n is the number of states of the input DES G and $m \leq \ell n^2$ is the number of transitions of the projected automaton of G, it does not depend on the parameter k and, in general, outperforms the existing algorithms. An exception is the case of a very small parameter k. In particular, if $k < 2\log(n)/\log(\ell)$, the algorithms based on the state estimator and on the two-way observer are, in the worst-case, faster than our algorithm. Notice that this theoretical result agrees with the experimental results of Wintenberg et al. [41].

[1] The state complexity of the two-way observer is correct. The correction of Lan et al. [28] consists in adding a time bound to compute the intersection of two sets, and hence it does not influence the number of states.

Algorithm 4 Verification of strong k-step opacity

Require: A deterministic DES $G = (Q, \Sigma, \delta, q_0)$, $Q_S \subseteq Q$, $\Sigma_o \subseteq \Sigma$, and $k \in \mathbb{N}_\infty$.
Ensure: `true` if and only if G is strongly k-step opaque with respect to Q_S and $P\colon \Sigma^* \to \Sigma_o^*$

1: Let G_{norm} be the normalization of G by Transformation 5.44
2: Transform G_{norm} to G' by Transformation 5.48
3: Use Algorithm 2 on G' with the set of secret states Q'_S, the set of non-secret states Q'_{NS}, observable events Σ_o, and k
4: **return** the answer of Algorithm 2

6.3 Verification of k-SSO

Theorem 5.49 gives us a clue how to verify strong k-step opacity of a given deterministic DES with the help of the verification algorithm for weak k-step opacity from the previous section. Given an instance of strong k-step opacity problem, we first transform it into an instance of weak k-step opacity problem using Transformation 5.48, and then verify the property with Algorithm 2. This idea is formulated as Algorithm 4.

The input of Algorithm 4 is the strong k-step opacity verification problem, which consists of a deterministic DES G, a projection $P\colon \Sigma^* \to \Sigma_o^*$, a set of secret states $Q_S \subseteq Q$, and a parameter $k \in \mathbb{N}_\infty$.

Algorithms verifying strong k-step opacity have been investigated in the literature. In particular, Falcone and Marchand [20] designed an algorithm based on a k-delay trajectory estimation, however, they did not analyze its complexity, and the complexity analyses in the literature are inconsistent. While Ma et al. [30] state that the complexity is $O(\ell 2^{n^2+n})$, where n is the number of states and ℓ is the number of observable events of the verified deterministic DES, Wintenberg et al. [41] state that the state complexity is $O((\ell+1)^k 2^n)$. According to [20, Definition 7], however, the k-delay trajectory estimator has $O(2^{n^{k+1}\cdot 2^k})$ states.

Recently, Ma et al. [30] designed another algorithm with complexity $O(\ell 2^{(k+2)n})$, and even more recently, Wintenberg et al. [41] discussed and experimentally compared algorithms based on (i) the secret observer with complexity $O(\ell(k+3)^n)$, on (ii) the reverse comparison with complexity $O((n+m)(k+1)2^n)$, where $m \leq \ell n^2$ is the number of transitions in the involved projected NFA, and on (iii) the construction of the k-delay trajectory estimator of Falcone and Marchand [20], which they claim to be of complexity $O(\ell(\ell+1)^k 2^n)$.

We now analyze the complexity of Algorithm 4 and show that its worst-case complexity is better than the complexity of existing algorithms. Namely, we show that the space and time complexity of Algorithm 4 is $O(n2^n)$ and $O((n+m)2^n)$, respectively, where n is the number of states of G and m is the number of transitions of $P(G)$. Notice that the complexity does not depend on the parameter k.

Before we formally state this result, notice that $m \leq \ell n^2$, where ℓ is the number of observable events. Since ℓn^2 is the maximum number of transitions in an n-state NFA with ℓ events, m is often significantly smaller than ℓn^2.

For a deterministic DES with n states, Transformation 5.44 results in a normalized DES with up to $2n$ states, and hence it may seem that the observer of the normalized DES could have up to 2^{2n} states. The following lemma states that the observer of the normalized DES has in fact at most 2^n states.

Lemma 6.3. *Let G be an n-state deterministic DES, and let G_{norm} be its normalization obtained by Transformation 5.44. Then, the observer of G_{norm} has at most 2^n states.* □

Notice that Lemma 6.3 does not claim that the number of states of the observer of G and of the observer of its normalization G_{norm} coincide. It only provides an upper bound on the worst-case complexity.

Similarly, for a normal deterministic DES G with n states, Transformation 5.48 results in a deterministic DES, denoted by G', with up to $2n$ states. The second lemma states that the observer of G' has as many states as the observer of G.

Lemma 6.4. *Let G be a normal deterministic DES with n states, and let G' be obtained from G by Transformation 5.48. Then, the numbers of states of the observer of G' and of the observer of G coincide.* □

We can now state the following result analyzing the complexity of Algorithm 4.

Theorem 6.5. *The space and time complexity of Algorithm 4 is $O(n2^n)$ and $O((n+m)2^n)$, respectively, where n is the number of states of G and m is the number of transitions of $P(G)$, that is, $m \leq \ell n^2$, where ℓ is the number of observable events.*

Proof. Let G be an n-state deterministic DES. In the first step, we construct the normalization G_{norm} of G with at most $2n$ states, the observer of which has at most 2^n states by Lemma 6.3. Then, we apply Algorithm 2 to G' obtained from G_{norm} by Transformation 5.48. In particular, by Lemma 6.4, we compute the observer of G' with at most 2^n states, and the projected automaton $P(G')$ with at most $4n$ states. Then, for every reachable state X of G'^{obs}, and for every $x \in X \cap Q'_S$, we add the pair $(x, X \cap Q'_{NS})$ to the set Y. This cycle takes time $O(n2^n)$. Afterwards, we construct the automaton H as the part of the full observer of G' accessible from the states of the second components of Y. Since H consists only of the subsets of Q'_{NS}, of which there is at most 2^n, the automaton H has $O(2^n)$ states. The automaton $\mathcal{C} = P(G') \times H$ thus has $O(n2^n)$ states and $O(m2^n)$ transitions, the sum of which is the time complexity of the BFS applied to mark states of $\mathcal{C}$ reachable from the states of Y in at most k steps. Therefore, the state complexity of Algorithm 4 is $O(n2^n)$ and the time complexity is $O(n2^n + (n+m)2^n) = O((n+m)2^n)$. □

Comparing the complexity $O((n+m)2^n)$ of Algorithm 4 with the complexity of the existing algorithms, the reader may see that (1) the complexity of Algorithm 4 does not depend on the parameter k, and (2) it is better than the complexity of the existing algorithms, because the minimum of the worst-case complexities $O(\ell 2^{n^{k+1} \cdot 2^k})$, $O(\ell 2^{(k+2)n})$, $O(\ell(k+3)^n)$, and $O((n+m)(k+1)2^n)$ of the existing algorithms discussed at the beginning of this subsection is $O((n+m)2^n)$ for $k = 1$, and $O((n+m)(k+1)2^n) = O((n+m)2^{2n})$ for $k \in O(2^n)$. Notice that the minimum worst-case complexity for large k is significantly higher than the complexity $O((n+m)2^n)$ of Algorithm 4. In fact, the complexity of Algorithm 4, and the minimum worst-case complexity of the existing algorithms for very small k, coincide. However, while the existing algorithms can handle only inputs with a very small k with this complexity, our algorithm can handle inputs with k of arbitrary value with this complexity. Consequently, our algorithm improves the complexity of the verification of strong k-step opacity.

Chapter 7

Conclusions

In this paper, we presented new results in three areas concerning opacity of discrete-event systems modeled by automata: the complexity of deciding opacity, the design of verification algorithms, and the relationships among various notions of opacity. We thus provided a complete and improved complexity picture of verifying the discussed notions of opacity.

In Chapter 4, we study the properties of current-state opacity in systems with a restricted alphabet and a restricted structure. We showed that the problem of deciding current-state opacity remains hard for almost all practial cases, cf. Table 1.1. Most notably, we showed that current-state opacity is:

1. PSpace-complete for systems modeled by DFAs/poDFAs with three events, one of which is unobservable (Theorem 4.7 and Corollary 4.15),
2. coNP-complete for systems modeled by NFAs/DFAs with a single observable event (Theorem 4.8 and Corollary 4.9), and
3. coNP-complete for systems modeled by acyclic NFAs/acyclic DFAs with two observable events (Theorem 4.10 and Corollary 4.12).

Chapter 5 is dedicated to transformations among the considered opacity notions. Our transformations are computable in polynomial time and preserve the number of observable events and determinism (whenever it is meaningful), allowing us to derive new results for corresponding opacity notions, see Table 1.2 for an overview. Below we summarize the results obtained from the transformations.

- **Language-based opacity** and **initial-and-final-state opacity** – By combining Theorems 4.7 and 4.8 with transformations of Wu and Lafortune [42], we can conclude that deciding LBO and IFO is PSpace-complete for systems with two or more observable events, and coNP-complete for systems with a single observable event.
- **Initial-state opacity** – We show that deciding ISO is PSpace-comp-lete for systems with two or more observable events. This result is established through Transformation 5.3 (hardness) and the membership result of Saboori [32]. Additionally, Theorem 5.7 shows that deciding ISO is NL-complete in the single observable event case.
- **Trace opacity** – We show that deciding TO is PSpace-complete for systems with two or more observable events. This result is established through Transformation 5.8 (hardness) and the membership result of Dubreil [18]. Additionally, Theorem 5.12 shows that deciding TO is NL-complete in the single observable event case.

- **Weak k-step opacity** – We show that deciding k-SO is PSpace-complete for systems with two or more observable events and the parameter $k \in \mathbb{N}_\infty$ encoded in binary. This result is established through Transformations 5.16 (hardness) and 5.35 (membership). In the single observable event case, deciding k-SO is coNP-complete by Transformations 5.25 (hardness) and 5.38 (membership).
- **Strong k-step opacity** – We show that deciding k-SSO is PSpace-complete for systems with two or more observable events and the parameter $k \in \mathbb{N}_\infty$ encoded in binary. This result is established through Transformations 5.16 (hardness) and 5.48 (membership). In the single observable event case, deciding k-SSO is coNP-complete by Transformations 5.21 (hardness) and 5.48 (membership). Additionally, Theorem 5.47 describes the relationship of 0-SO and 0-SSO.

In Chapter 6, we propose three algorithms for verifying language-based opacity and trace opacity (Algorithm 1), weak k-step opacity (Algorithm 2), and strong k-step opacity (Algorithm 4). We provide an analysis of all mentioned algorithms and we show that their time complexity is $O((n+m)2^n)$, where n stands for the number of states of the input automaton and m for the number of transitions in the projected automaton of the input automaton. In particular, the complexity of algorithms for verifying weak and strong k-step opacity does not depend on the parameter $k \in \mathbb{N}_\infty$. However, it remains an open question how our algorithms would perform if tested experimentally.

Bibliography

[1] R. Alur, P. Černý, and S. Zdancewic. Preserving secrecy under refinement. In *Automata, Languages and Programming*, pages 107–118. Springer, 2006.

[2] P. R. Asveld and A. Nijholt. The inclusion problem for some subclasses of context-free languages. *Theoretical Computer Science*, 230(1-2):247–256, 2000.

[3] E. Badouel, M. Bednarczyk, A. Borzyszkowski, B. Caillaud, and P. Darondeau. Concurrent secrets. *Discrete Event Dynamic Systems*, 17:425–446, 2007.

[4] J. Balun and T. Masopust. On opacity verification for discrete-event systems. *IFAC-PapersOnLine*, 53(2):2075–2080, 2020.

[5] J. Balun and T. Masopust. On verification of strong periodic D-detect-ability for discrete event systems. *IFAC-PapersOnLine*, 53(4):263–268, 2020. 15th IFAC Workshop on Discrete Event Systems WODES 2020.

[6] J. Balun and T. Masopust. Comparing the notions of opacity for discrete-event systems. *Discrete Event Dynamic Systems*, 31:553–582, 2021.

[7] J. Balun and T. Masopust. On verification of D-detectability for discrete event systems. *Automatica*, 133:109884, 2021.

[8] J. Balun and T. Masopust. On transformations among opacity notions. *2022 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 3012–3017, 2022.

[9] J. Balun and T. Masopust. On verification of weak and strong k-step opacity for discrete-event systems. *IFAC-PapersOnLine*, 55(28):108–113, 2022. 16th IFAC Workshop on Discrete Event Systems WODES 2022.

[10] N. BenHadj-Alouane, S. Lafrance, F. Lin, J. Mullins, and M. Yeddes. On the verification of intransitive noninterference in mulitlevel security. *IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics)*, 35(5):948–958, 2005.

[11] J. W. Bryans, M. Koutny, L. Mazaré, and P. Y. A. Ryan. Opacity generalised to transition systems. *International Journal of Information Security*, 7(6):421–435, 2008.

[12] J. W. Bryans, M. Koutny, and P. Y. Ryan. Modelling opacity using Petri nets. *Electronic Notes in Theoretical Computer Science*, 121:101–115, 2005.

[13] J. A. Brzozowski and F. E. Fich. Languages of R-trivial monoids. *Journal of Computer and System Sciences*, 20(1):32–49, 1980.

[14] C. G. Cassandras and S. Lafortune, editors. *Introduction to Discrete Event Systems*. Springer, third edition, 2021.

[15] F. Cassez. The dark side of timed opacity. In *Advances in Information Security and Assurance*, volume 5576, pages 21–30. Springer, 2009.

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

[17] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms*. MIT Press, 2009.

[18] J. Dubreil. Monitoring and supervisory control for opacity properties. 11 2009.

[19] J. Dubreil, P. Darondeau, and H. Marchand. Opacity enforcing control synthesis. In *WODES*, pages 28–35, 2008.

[20] Y. Falcone and H. Marchand. Enforcement and validation (at runtime) of various notions of opacity. *Discrete Event Dynamic Systems*, 25:531–570, 2014.

[21] R. Focardi and R. Gorrieri. A taxonomy of trace-based security properties for CCS. In *The Computer Security Foundations Workshop VII*, pages 126–136. IEEE Comput. Soc. Press, 1994.

[22] M. Holzer and M. Kutrib. Descriptional and computational complexity of finite automata—A survey. *Information and Computation*, 209(3):456–470, 2011.

[23] R. Jacob, J.-J. Lesage, and J.-M. Faure. Overview of discrete event systems opacity: Models, validation, and quantification. *Annual Reviews in Control*, 41:135–146, 2016.

[24] G. Jirásková and T. Masopust. On a structural property in the state complexity of projected regular languages. *Theoretical Computer Science*, 449:93–105, 2012.

[25] C. Keroglou and C. N. Hadjicostis. Initial state opacity in stochastic des. In *2013 IEEE 18th Conference on Emerging Technologies & Factory Automation (ETFA)*, pages 1–8, 2013.

[26] J. Komenda, D. Zorzenon, and J. Balun. Modeling of safe timed petri nets by two-level (max,+) automata. *IFAC-PapersOnLine*, 55(28):212–219, 2022. 16th IFAC Workshop on Discrete Event Systems WODES 2022.

[27] M. Krötzsch, T. Masopust, and M. Thomazo. Complexity of universality and related problems for partially ordered NFAs. *Information and Computation*, 255(1):177–192, 2017.

[28] H. Lan, Y. Tong, J. Guo, and A. Giua. Comments on "A new approach for the verification of infinite-step and K-step opacity using two-way observers" [Automatica 80 (2017) 162–171]. *Automatica*, 122:109290, 2020.

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

[30] Z. Ma, X. Yin, and Z. Li. Verification and enforcement of strong infinite- and k-step opacity using state recognizers. *Automatica*, 133:109838, 2021.

[31] L. Mazaré. Decidability of opacity with non-atomic keys. In *Formal Aspects in Security and Trust*, pages 71–84. Springer, 2004.

[32] A. Saboori. *Verification and enforcement of state-based notions of opacity in discrete event systems*. PhD thesis, University of Illinois at Urbana-Champaign, 2011.

[33] A. Saboori and C. N. Hadjicostis. Notions of security and opacity in discrete event systems. In *IEEE CDC*, pages 5056–5061, 2007.

[34] A. Saboori and C. N. Hadjicostis. Opacity-enforcing supervisory strategies for secure discrete event systems. In *Conference on Decision and Control*. IEEE, 2008.

[35] A. Saboori and C. N. Hadjicostis. Coverage analysis of mobile agent trajectory via state-based opacity formulations. *Control Engineering Practice*, 19(9):967–977, 2011. Special Section: DCDS'09 – The 2nd IFAC Workshop on Dependable Control of Discrete Systems.

[36] A. Saboori and C. N. Hadjicostis. Verification of K-step opacity and analysis of its complexity. *IEEE Transactions on Automation Science and Engineering*, 8(3):549–559, 2011.

[37] A. Saboori and C. N. Hadjicostis. Verification of infinite-step opacity and complexity considerations. *IEEE Transactions on Automatic Control*, 57(5):1265–1269, 2012.

[38] S. Schneider and A. Sidiropoulos. CSP and anonymity. In *Computer Security — ESORICS 96*, pages 198–218. Springer, 1996.

[39] T. Schwentick, D. Thérien, and H. Vollmer. Partially-ordered two-way automata: A new characterization of DA. In *Developments in Language Theory (DLT)*, volume 2295 of *LNCS*, pages 239–250, 2001.

[40] L. J. Stockmeyer and A. R. Meyer. Word problems requiring exponential time: Preliminary report. In *ACM Symposium on Theory of Computing (STOC)*, pages 1–9. ACM Press, 1973.

[41] A. Wintenberg, M. Blischke, S. Lafortune, and N. Ozay. A general language-based framework for specifying and verifying notions of opacity. *Discrete Event Dynamic Systems*, 32:253–289, 2022.

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

[43] Y.-C. Wu, K. A. Sankararaman, and S. Lafortune. Ensuring privacy in location-based services: An approach based on opacity enforcement. *IFAC Proceedings Volumes*, 47(2):33–38, 2014. 12th IFAC International Workshop on Discrete Event Systems (2014).

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

[45] D. Zorzenon, J. Balun, and J. Raisch. Weak consistency of P-time event graphs. *IFAC-PapersOnLine*, 55(40):19–24, 2022. 1st IFAC Workshop on Control of Complex Systems COSY 2022.