On Verification of D-Detectability for Discrete Event Systems ★

Jiří Balun and Tomáš Masopust

Department of Computer Science, Faculty of Science, Palacky University in Olomouc, Czechia

Abstract

Detectability is a state-estimation property asking whether the current and subsequent states of a system can be determined based on observations. To exactly determine the current and subsequent states may be, however, too strict in some applications. Therefore, Shu and Lin relaxed detectability to D-detectability distinguishing only certain pairs of states rather than all states. Four variants of D-detectability were defined: strong (periodic) D-detectability and weak (periodic) D-detectability. Deciding weak (periodic) D-detectability is PSpace-complete, while deciding strong (periodic) detectability or strong D-detectability is polynomial, and we show that it is NL-complete. To the best of our knowledge, it is an open problem whether there exists a polynomial-time algorithm deciding strong periodic D-detectability. We show that deciding strong periodic D-detectability is a PSpace-complete problem, which means that there is no polynomial-time algorithm, unless every problem solvable in polynomial space can be solved in polynomial time. We further show that there is no polynomial-time algorithm even for systems with a single observable event, unless P = NP. Finally, we propose a class of systems for which the problem is tractable.

Key words: Discrete event systems, finite automata, state estimation, detectability, verification, complexity

1 Introduction

Detectability of discrete event systems (DESs) modeled by finite automata was introduced by Shu et al. (2007) as a generalization of other notions studied in the literature, including stability of Ozveren and Willsky (1990) and observability of Caines et al. (1988) or of Ramadge (1986). An evidence that many practical problems can be formulated as the detectability problem for DESs was provided by Shu and Lin (2011). Furthermore, Lin (2011) showed that detectability is closely related to other important properties, such as observability, diagnosability, and opacity.

Detectability is a state-estimation property asking whether the current and subsequent states of a DES can be determined after a finite number of observations. Shu et al. (2007) defined four variants of detectability: strong (periodic) detectability and weak (periodic) detectability. In their work, they first studied detectability for deterministic DESs, which they define as DESs modeled by deterministic finite automata with a set of initial states. The motivation for a set of initial states rather than a single initial state results from the observation that it is often unknown which state the system is initially in. They proposed an exponential algorithm for deciding detectability of a deterministic DES based on the computation of an observer. Shortly after, Shu and Lin (2011) extended the problem to nondeterministic DESs (DESs modeled by general nondeterministic finite automata) and designed an algorithm deciding strong (periodic) detectability of nondeterministic DESs in polynomial time. The problem of deciding strong (periodic) detectability was later shown to be NL-complete by Masopust (2018a), which means that this problem is efficiently solvable on a parallel computer.

Recently, Zhang (2017) showed that deciding weak (periodic) detectability is a PSpace-complete problem and that it remains PSpace-hard even for deterministic DESs with all events observable. Masopust (2018a) strengthened these results by proving the same complexity for structurally "simplest" deadlock-free DESs that are modeled by deterministic finite automata without non-trivial cycles.

Since the requirement to exactly determine the current and subsequent states after a finite number of observations may be too strict in some applications, Shu and Lin (2011) relaxed the notion of detectability to a so-called *D-detectability* property. The idea behind the relaxation is to distinguish only certain pairs of states rather than all states of the system. Four variants of D-detectability were defined: strong (periodic) D-detectability and weak (periodic) D-detectability.

★ Part of this work was presented at the International Workshop on Discrete-Event Systems WODES 2020 (Balun and Masopust, 2020). Corresponding author: T. Masopust.

Email addresses: jiri.balun01@upol.cz (Jiří Balun), tomas.masopust@upol.cz (Tomáš Masopust).

The notion of (D-)detectability has been extended in many directions. To mention a few, Shu and Lin (2013) extended strong (D-)detectability to delayed (D-)detectability, motivated by discrete event systems with delays, and designed a polynomial-time algorithm to check strong (D-)detectability for delayed DESs. Zhang and Giua (2019) improved the algorithm for checking strong delayed (D-)detectability, and introduced other notions of detectability, see also Zhang et al. (2020). Alves and Basilio (2019) studied (D-)detectability for discrete event systems with multi-channel communication networks. Yin and Lafortune (2017) examined the verification of weak and strong detectability properties for modular DESs, and showed that checking both is PSpace-hard. The exact complexities of these two problems were resolved by Masopust and Yin (2019); they are, respectively, PSpace-complete and ExpSpace-complete. We refer the reader to Hadjicostis (2020) for the latest development of state-estimation properties.

Since detectability is a special case of D-detectability, the problem of deciding D-detectability is at least as hard as deciding detectability. An immediate consequence of this observation, together with the observer-based algorithm of Shu and Lin (2011), is that the complexity of deciding weak (periodic) D-detectability is a PSpace-complete problem.

The case of strong D-detectability is similar to that of strong detectability. For strong (periodic) detectability, Shu and Lin (2011) designed a detector that can decide, in polynomial time, whether a DES satisfies strong (periodic) detectability. They further showed that their detector is also suitable for deciding strong D-detectability. Consequently, the complexity of verifying strong D-detectability is polynomial. We further improve this result by showing that deciding whether a DES satisfies strong D-detectability is NL-complete (Theorem 1). Since NL is the class of problems that can be efficiently parallelized, see Arora and Barak (2009) for details, we obtain that the verification of strong D-detectability can be efficiently verified on a parallel computer.

However, deciding strong *periodic* D-detectability is more involved. Although the detector-based technique provides a polynomial-time algorithm to decide strong periodic detectability, Shu and Lin (2011) give an example that this algorithm does not work for checking strong periodic D-detectability. They leave the question of the existence of a polynomial-time algorithm deciding strong periodic D-detectability of a DES open. To the best of our knowledge, this question has not yet been answered in the literature. We answer this question by showing that there does not exist any algorithm that would decide, in polynomial time, whether a DES satisfies strong periodic D-detectability (Theorem 3), unless P = PSpace. The question whether P = PSpace is a longstanding open problem of computer science asking whether every problem solvable in polynomial space can also be solved in polynomial time. It is generally believed that it is not the case. In particular, Theorem 3 shows that the strong periodic D-detectability problem is PSpace-complete. Formulated differently, the result says that the technique based on the computation of the observer is in principle optimal. Notice that since NL is a strict subclass of PSpace, strong periodic D-detectability is significantly more complex than its non-periodic counterpart—strong D-detectability.

We further show that strong *periodic* D-detectability is more complex than strong D-detectability even for systems having only a single observable event. Namely, we show that strong periodic D-detectability cannot be verified in polynomial time even for DESs that have only a single observable event (Theorem 4), unless P = NP. The problem whether P = NP is another longstanding open problem of computer science.

Finally, we specify a class of DESs for which deciding strong *periodic* D-detectability is in polynomial time. Namely, we consider the class of systems modeled by NFAs where all cycles are in the form of self-loops and where there is no nondeterministic choice between a step changing the state and a step not changing the state under the same observation. These restrictions are purely structural, and the models are called *rpoDES*, see Section 5 for details.

Our contributions, compared with known results, are summarized in Tables 1 and 2.

2 Preliminaries and Definitions

For a set A, $|A|$ denotes the cardinality of A and 2^A its power set. An alphabet Σ is a finite nonempty set of events. A string over Σ is a sequence of events of Σ. Let Σ^* denote the set of all finite strings over Σ, and let Σ^ω denote the set of all infinite strings over Σ; the empty string is denoted by ε. For a string $u \in \Sigma^*$, $|u|$ denotes its length. As usual, the notation Σ^+ stands for Σ^* without the empty string ε, that is, $\Sigma^* \setminus \{\varepsilon\}$.

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, $I \subseteq Q$ is a set of initial states, $F \subseteq Q$ is a set of marked states, and $\delta \colon Q \times \Sigma \to 2^Q$ is a transition function that can be extended to the domain $2^Q \times \Sigma^*$ by induction. The *language recognized by* $\mathcal{A}$ is the set $L(\mathcal{A}) = \{w \in \Sigma^* \mid \delta(I, w) \cap F \neq \emptyset\}$. Equivalently, the transition function δ is a relation $\delta \subseteq Q \times \Sigma \times Q$, where, for instance, $\delta(q, a) = \{s, t\}$ denotes the two transitions (q, a, s) and (q, a, t). The NFA $\mathcal{A}$ is *unary* if its input alphabet Σ is a singleton.

The NFA $\mathcal{A}$ is *deterministic* (DFA) if it has a unique initial state, i.e., $|I| = 1$, and no nondeterministic transitions, i.e., $|\delta(q, a)| \leq 1$ for every $q \in Q$ and $a \in \Sigma$. The DFA $\mathcal{A}$ is *total* if in every state, a transition under every event is defined, i.e., $|\delta(q, a)| = 1$ for every $q \in Q$ and $a \in \Sigma$. For DFAs, we identify singletons with their elements and simply write p instead of $\{p\}$. Specifically, we write $\delta(q, a) = p$ instead of $\delta(q, a) = \{p\}$.

A *discrete event system* (DES) is an NFA G. Since the marking of states does not play a role when studying detectability,

	DESs			rpoDESs		
	detectability	D-detectability		detectability	D-detectability	
	known	known	new	known	known	new
strong	NL-c [1]	in P [2]	NL-c (Thm 1)	NL-c [1]	in P [2]	NL-c (Thm 1 & Cor 9)
weak	PSpace-c [3]	PSpace-c	—	PSpace-c [1]	PSpace-c	—
strong periodic	NL-c [1]	?	PSpace-c (Thm 3)	NL-c [1]	?	NL-c (Thm 8)
weak periodic	PSpace-c [3]	PSpace-c	—	PSpace-c [1]	PSpace-c	—

Table 1
Summary of known and new results for DESs and rpoDESs; ? means that the problem was open; results easily derivable from existing results are also placed among known results; [1] = Masopust (2018a), [2] = Shu and Lin (2011), and [3] = Zhang (2017).

	unary DESs		unary rpoDESs	
	detectability	D-detectability	detectability	D-detectability
strong	NL-c [1]	NL-c (Thm 1)	NL-c (Cor 9)	NL-c (Cor 9)
weak	NL-c (Thm 6)	NL-c (Thm 6)	NL-c (Thm 6)	NL-c (Thm 6)
strong periodic	NL-c [1]	NP-c (Thm 4)	NL-c (Thm 8)	NL-c (Thm 8)
weak periodic	NL-c (Thm 6)	NP-c (Thm 6)	NL-c (Thm 6)	NL-c (Thm 6)

Table 2
Summary of known and new results for DESs and rpoDESs with a single observable event; [1] stands for Masopust (2018a).

we neglect it and simply write $G = (Q, \Sigma, \delta, I)$ without specifying the set of marked states. In addition, the alphabet Σ of a DES $G = (Q, \Sigma, \delta, I)$ is partitioned into the set Σ_o of *observable events* and the set $\Sigma_{uo} = \Sigma \setminus \Sigma_o$ of *unobservable events*. If the DES has only a single observable event, we call it a *unary* DES.

State-estimation properties are based on the observation of events. The observation is described by projections. The projection $P\colon \Sigma^* \to \Sigma_o^*$ is a morphism defined by $P(a) = \varepsilon$ for $a \in \Sigma \setminus \Sigma_o$, and $P(a) = a$ for $a \in \Sigma_o$. The action of P on a string $w = a_1 a_2 \cdots a_n$, where $a_i \in \Sigma$ for $1 \le i \le n$, is to erase all events from w that do not belong to Σ_o; in particular, $P(a_1 a_2 \cdots a_n) = P(a_1)P(a_2)\cdots P(a_n)$. The definition can readily be extended to infinite strings and languages.

Shu and Lin (2011) make the following two reasonable assumptions on the DES $G = (Q, \Sigma, \delta, I)$ under which they define the notions of detectability:

(1) G is *deadlock free* – it means that for every state of the system, at least one event can occur; formally, for every $q \in Q$, there exists $\sigma \in \Sigma$ such that $\delta(q, \sigma) \neq \emptyset$.
(2) No loop in G consists solely of unobservable events – it means that for every $q \in Q$ and every $w \in \Sigma_{uo}^+$, we have that $q \notin \delta(q, w)$.

We point out that to verify whether a system satisfies these two properties is easy.

The set of infinite sequences of events (or trajectories) generated by the DES G is denoted by $L^\omega(G)$. Given a set $Q' \subseteq Q$, the set of all possible states reachable from Q' after observing a string $t \in \Sigma_o^*$ is denoted by

$$R(Q', t) = \bigcup_{w \in \Sigma^*, P(w)=t} \delta(Q', w)\,.$$

For $w \in L^\omega(G)$, we denote the set of prefixes of w by $Pr(w)$.

2.1 A Brief Complexity Review

We now briefly review the basics of complexity theory needed to understand the results. A *decision problem* is a yes-no question. A decision problem is *decidable* if there exists an algorithm that can solve the problem. Complexity theory classifies decidable problems to classes according to the time or space an algorithm needs to solve the problem. The complexity classes we consider in this paper are NL, P, NP, and PSpace. They denote the classes of problems that are solvable by a nondeterministic logarithmic-space, deterministic polynomial-time, nondeterministic polynomial-time, and deterministic polynomial-space algorithm, respectively. The hierarchy of classes is NL $\subseteq$ P $\subseteq$ NP $\subseteq$ PSpace. Which of the inclusions are strict is a longstanding open problem of computer science. The widely accepted conjecture is that all inclusions are strict. However, so far only the inclusion NL $\subseteq$ PSpace is known to be strict. A decision problem is NL-complete (resp. NP-complete, PSpace-complete) if it belongs to NL (resp. NP, PSpace) and every problem from NL (resp. NP, PSpace) can be reduced to it by a deterministic logarithmic-space (resp. polynomial-time) algorithm.

3 Definitions of the D-Detectability Problems

Shu and Lin (2011) defined D-detectability as a generalization of detectability by explicitly identifying the states that need to be distinguished.

Let $G = (Q, \Sigma, \delta, I)$ be a DES, and let $T_{spec} \subseteq Q \times Q$ be a relation on the set of states of G. The relation T_{spec} specifies pairs of states that must be distinguished, and is therefore called a *specification*. The idea behind the definition of D-detectability is to ensure that the pairs of states from T_{spec} are distinguished after a finite number of observations. We now recall the definitions of the four variants of D-detectability.

For a DES $G = (Q, \Sigma, \delta, I)$, projection $P \colon \Sigma^* \to \Sigma_o^*$, and a specification T_{spec}, we say that G is

- *strongly D-detectable* if, for all trajectories of the system, the pairs of states of T_{spec} can be distinguished in every step of the system after a finite number of observations; formally:

$$(\exists n \in \mathbb{N})(\forall s \in L^\omega(G))(\forall t \in Pr(s))\ |P(t)| > n \\ \Rightarrow (R(I, P(t)) \times R(I, P(t))) \cap T_{spec} = \emptyset\,.$$

- *weakly D-detectable* if, for some trajectories of the system, the pairs of states of T_{spec} can be distinguished in every step of the system after a finite number of observations; formally:

$$(\exists n \in \mathbb{N})(\exists s \in L^\omega(G))(\forall t \in Pr(s))\ |P(t)| > n \\ \Rightarrow (R(I, P(t)) \times R(I, P(t))) \cap T_{spec} = \emptyset\,.$$

- *strongly periodically D-detectable* if the pairs of states of T_{spec} can be periodically distinguished for all trajectories of the system; formally:

$$(\exists n \in \mathbb{N})(\forall s \in L^\omega(G))(\forall t \in Pr(s))(\exists t' \in \Sigma^*) \\ tt' \in Pr(s)\ \wedge\ |P(t')| < n \\ \wedge\ (R(I, P(tt')) \times R(I, P(tt'))) \cap T_{spec} = \emptyset\,.$$

- *weakly periodically D-detectable* if the pairs of states of T_{spec} can be periodically distinguished for some trajectories of the system; formally:

$$(\exists n \in \mathbb{N})(\exists s \in L^\omega(G))(\forall t \in Pr(s))(\exists t' \in \Sigma^*) \\ tt' \in Pr(s)\ \wedge\ |P(t')| < n \\ \wedge\ (R(I, P(tt')) \times R(I, P(tt'))) \cap T_{spec} = \emptyset\,.$$

4 Main Results

We now discuss the complexity of checking whether a DES is D-detectable. As already pointed out in the introduction, the complexity of deciding weak (periodic) D-detectability follows from the complexity of checking weak (periodic) detectability. Indeed, a polynomial space is sufficient for an algorithm based on the inspection of states in the observer and works for all the D-detectability variants. Deciding weak (periodic) D-detectability is therefore in PSPACE. On the other hand, detectability is a special case of D-detectability for $T_{spec} = Q \times Q \setminus \{(q, q) \mid q \in Q\}$. Therefore, deciding weak (periodic) D-detectability is at least as hard as deciding weak (periodic) detectability. Since the latter is PSPACE-hard, so is the former.

4.1 Verification of Strong D-Detectability

Shu and Lin (2011) designed an algorithm that verifies strong (periodic) detectability in polynomial time. Their algorithm is based on the construction of a finite automaton called a *detector*. Intuitively, given a DES G, the detector G_{det} is constructed from G so that (i) the set of initial states of G_{det} is the set of all states of G reachable from the initial states of G under strings consisting only of unobservable events, (ii) all the other states of G_{det} are one- or two-element subsets of the set of states of G, and (iii) the transition relation of G_{det} is constructed in the similar way as that of the observer, but if the reached state X in the observer consists of more than two states, then the detector G_{det} has several transitions each leading to a two-element subset of X, see Shu and Lin (2011) for details. Since the states of the detector are one- or two-element subsets, their number is polynomial.

Shu and Lin (2011) showed that G is strongly (periodically) detectable if and only if any state reachable from any loop in G_{det} consists solely (periodically) of distinguishable states. They further proved that their algorithm, respectively the detector, works for checking strong D-detectability, which in particular implies that the complexity of verifying strong D-detectability is polynomial.

We now show that deciding strong D-detectability is an NL-complete problem. Consequently, since NL is the class of problems that can be efficiently parallelized, see Arora and Barak (2009) for details, our result shows that strong D-detectability can be efficiently verified on a parallel computer.

Theorem 1. *Deciding whether a DES is strongly D-detectable is an* NL*-complete problem.*

Proof. We prove membership of the problem in NL by giving a nondeterministic logarithmic-space algorithm checking whether strong D-detectability does not hold. Since NL is closed under complement (Immerman, 1988; Szelepcsényi, 1988), it shows that there is a nondeterministic logarithmic-space algorithm checking whether the property is satisfied.

To check that strong D-detectability is not satisfied, our NL algorithm guesses two states of G_{det}, say x and y, where y contains indistinguishable states, and verifies that (i) y is reachable from x, (ii) x is reachable from the initial state of G_{det}, and (iii) x is in a cycle, i.e., x is reachable from x by a

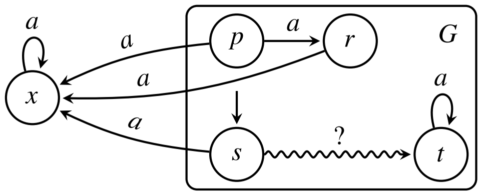

Figure 1. The DES $\mathcal{A}$ constructed from G in the NL-hardness proof of Theorem 1.

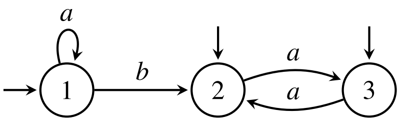

Figure 2. The DES G from Example 2.

path having at least one transition. Notice that our algorithm does not construct the detector G_{det}. It only stores a constant number of states of G_{det} and computes the required states of G_{det} on demand. Therefore, our algorithm does not need more than a logarithmic space. For more details how to check reachability in NL, we refer the reader to Masopust (2018b).

To show NL-hardness, we reduce the *DAG non-reachability problem* (Cho and Huynh, 1991) that asks, given a directed acyclic graph $G = (V, E)$ and two nodes $s, t \in V$, whether t is not reachable from s. From G, we construct a DES $\mathcal{A} = (V \cup \{x\}, \{a\}, \delta, s)$, where $x \notin V$ is a new state and a is an observable event. For every edge $(p, r) \in E$, we add the transition (p, a, r) to δ, and for every $p \in V \setminus \{t\}$, we add the transition (p, a, x) to δ. Moreover, we add the self-loop transitions (x, a, x) and (t, a, t) to δ. The construction is depicted in Figure 1. Notice that $\mathcal{A}$ is deadlock-free and has no unobservable events. Let the specification T_{spec} be defined as $T_{spec} = \{(t, x)\}$. We now show that t is not reachable from s in the graph G if and only if the DES $\mathcal{A}$ is strongly D-detectable. If t is not reachable from s in G, then, for every $k \geq |V|$, $\delta(s, a^k) = \{x\}$. Therefore, $\mathcal{A}$ is strongly D-detectable. If t is reachable from s in G, then, for every $k \geq |V|$, $\delta(s, a^k) = \{t, x\}$. Therefore, $\mathcal{A}$ is not strongly D-detectable, because $(t, x) \in T_{spec}$. □

4.2 Verification of Strong Periodic D-Detectability

Although the detector-based technique leads to a polynomial-time algorithm deciding strong periodic detectability, Shu and Lin (2011) showed that this algorithm does not work for checking strong periodic D-detectability. To give the reader an idea of the detector-based polynomial-time algorithm and why it does not work for strong periodic D-detectability, we slightly elaborate the example of Shu and Lin (2011).

Example 2. Let $G = (\{1, 2, 3\}, \{a, b\}, \delta, \{1, 2, 3\})$ be the DES depicted in Figure 2, where both events are observable, and define $T_{spec} = \{(1, 3)\}$. The detector G_{det} is depicted in Figure 3. In G_{det}, we can see that G is not strongly D-detectable, because there is an infinite path alternating between the states $\{1, 2\}$ and $\{1, 3\}$ going periodically through state $\{1, 3\}$, violating thus strong D-detectability. From this path, we could get an impression that G is strongly periodically D-detectable. However, this is not the case as can be seen from the observer depicted in Figure 4, where the trajectory a^{ω} generated by the self-loop in state $\{1, 2, 3\}$ violates strong periodic D-detectability. ⋄

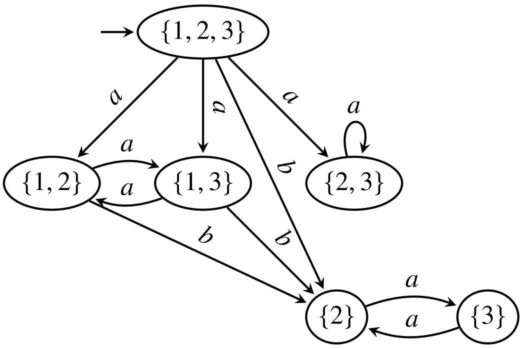

Figure 3. Detector G_{det} constructed from the DES G of Example 2.

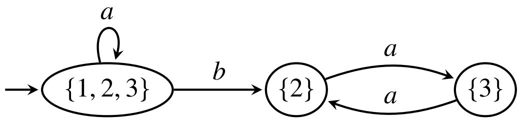

Figure 4. The observer of the DES G from Example 2.

Shu and Lin (2011) left the question whether there exists a polynomial-time algorithm deciding strong periodic D-detectability of a DES open. To the best of our knowledge, this question has not yet been resolved in the literature. We answer it in the sequel. In addition, we distinguish two cases based on the number of observable events in the system:

(i) The general case where the system has two or more observable events;

(ii) A special case where the system has only a single observable event.

The case of two or more observable events

As pointed out above, the problem whether a DES is strongly or weakly (periodically) D-detectable is in PSpace. In this section, we show that deciding strong periodic D-detectability is PSpace-hard, and hence PSpace-complete. Our result says that there is no algorithm solving this problem in polynomial time, unless P = PSpace.

Theorem 3. *Deciding whether a DES is strongly periodically D-detectable is a* PSpace*-complete problem. The problem is* PSpace*-hard even if the DES has only two observable and no unobservable events.*

Proof. Membership in PSpace follows from the inspection of states of the observer that are built on demand (Shu and Lin, 2011; Zhang, 2017; Masopust, 2018b).

To show PSpace-hardness, we reduce the intersection emptiness problem. The problem is PSpace-complete (Garey and Johnson, 1990) and asks, given a sequence $\mathcal{A}_1, \dots, \mathcal{A}_n$ of total DFAs over a common alphabet Σ with $|\Sigma| \geq 2$, whether the language $\cap_{i=1}^{n} L(\mathcal{A}_i)$ is empty. Without loss of generality, we may assume that $\Sigma = \{0, 1\}$. From the total DFAs $\mathcal{A}_1, \dots, \mathcal{A}_n$, we construct a DES G that is strongly periodically D-detectable if and only if the intersection of the languages of $\mathcal{A}_1, \dots, \mathcal{A}_n$ is empty.

The main idea of our proof is to construct G as a nondeterministic union of the automata $\mathcal{A}_1, \dots, \mathcal{A}_n$ together with $n+1$ new states, such that all and only these new states are reachable at the same time if and only if the intersection is nonempty (cf. Figure 5). This in particular means that if the intersection is empty, only a strict subset of the new states can be reached at the same time. After reaching the new states, the computation remains in the new states. The new states (except for one state, q^-) form a cycle, and therefore, during any further computation, the current states are periodically rotated. This allows us to make one of the new states, say q_1^+, periodically indistinguishable from the state q^-, and the other periodically distinguishable.

Formally, let $\mathcal{A}_1, \dots, \mathcal{A}_n$ be total DFAs over a common alphabet Σ, and let $\mathcal{A}_i = (Q_i, \Sigma, \delta_i, q_{0,i}, F_i)$. Without loss of generality, we may assume that the states of the DFAs are pairwise disjoint. Let a and b be new events, and let $f : \Sigma^* \to \{a, b\}^*$ be a morphism defined by $f(0) = ba$ and $f(1) = bb$. We modify every $\mathcal{A}_i$ by encoding the events of Σ as follows. We replace each transition $t = (p, 0, q)$ by two transitions (p, b, p_t) and (p_t, a, q), where p_t is a new state, and each transition $r = (p, 1, q)$ by two transitions (p, b, p_r) and (p_r, b, q), where p_r is a new state; see Figure 7 for an illustration how to replace the transitions of the DFAs of Figure 6. This replacement requires to add a new state for each transition of the DFAs, which can be done in polynomial time. The modified $\mathcal{A}_i$ is denoted by $\mathcal{A}'_i$.

We now construct a DES G as a nondeterministic union of all the automata $\mathcal{A}'_i$, i.e., G contains all states and transitions of every $\mathcal{A}'_i$, and we add $n+1$ new states $q^-, q_1^+, \dots, q_n^+$ and several new transitions under event a as depicted in Figure 5. Namely, for $i = 1, \dots, n$, we add the transition (q, a, q^-) for every non-marked state $q \in Q_i \setminus F_i$, and the transition (q, a, q_i^+) for every marked state $q \in F_i$. We further add the self-loop (q^-, σ, q^-) for every $\sigma \in \{a, b\}$. Finally, for every $\sigma \in \{a, b\}$, we create a cycle on the states $Q^+ = \{q_1^+, \dots, q_n^+\}$ by adding the transitions $(q_i^+, \sigma, q_{i+1}^+)$, for $1 \leq i < n$, and the transition (q_n^+, σ, q_1^+). The set of initial states of G is the set $I = \{q^-, q_{0,1}, \dots, q_{0,n}\}$ of initial states of the automata $\mathcal{A}_i$ plus the newly added state q^-. The alphabet of G is $\Sigma' = \{a, b\}$, all events of which are observable.

We define the specification $T_{spec} = \{(q^-, q_1^+)\}$, and show that G is strongly periodically D-detectable if and only if the intersection $\cap_{i=1}^{n} L(\mathcal{A}_i)$ is empty.

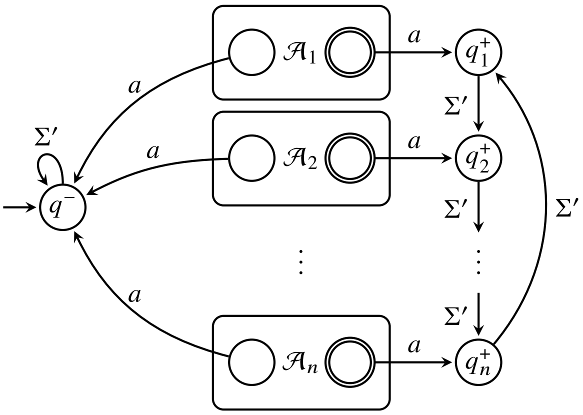

Figure 5. Construction of the DES G from the PSpace-hardness part of the proof of Theorem 3.

Assume that the intersection is empty. A trajectory that never reaches the states of Q^+ cannot violate strong periodic D-detectability, because it cannot enter state q_1^+ from the specification. Therefore, assume that G eventually enters a state of Q^+. When G enters a state of Q^+, it leaves all states out of $Q^+ \cup \{q^-\}$. Thus, let $s \in L^\omega(G)$ be an arbitrary trajectory that enters Q^+. Then, $s = s_1 s_2$ where G generates s_1 in states outside Q^+, and s_2 is the part after G first enters Q^+. In this case, $s_1 = f(w) \in \{a, b\}^*$, for some $w \in \{0, 1\}^*$, and $s_2 \in a\{a, b\}^\omega$. Then, after generating the first event of s_2, the observer of G is in a set of states consisting of q^- and a strict subset of Q^+; indeed, G cannot transit to all states of Q^+ at the same time, because the assumption that the intersection is empty implies that, for every $w \in \Sigma^*$, there exists $i \in \{1, \dots, n\}$ such that $w \notin L(\mathcal{A}_i)$.

Let $p_i \in Q^+ = \{q_1^+, \dots, q_n^+\}$ denote the state of Q^+ with the minimal index, in which G *cannot be* when the ith event of s_2 is generated. By construction, the cycle on Q^+ ensures that p_i periodically alternates among q_1^+ and some other states of Q^+ when generating s_2. Therefore, in the infinite sequence $p_1, p_2, \dots$, there are infinitely many j such that $p_j = q_1^+$, and hence q^- and q_1^+ are periodically distinguished, which shows that G is strongly periodically D-detectable.

On the other hand, assume that the intersection is nonempty, and let $w \in \cap_{i=1}^{n} L(\mathcal{A}_i)$. Then, after generating the string $f(w)a$, the observer of G reaches the state $\{q^-\} \cup Q^+$. Now, every transition keeps G in all states of $\{q^-\} \cup Q^+$, and hence it results in a self-loop in the observer of G. However, this self-loop violates strong periodic D-detectability, because it contains both state q^- and state q_1^+. Therefore, any trajectory $s \in L^\omega(G)$ with $f(w)a$ as a prefix leads to a set of states where the states of T_{spec} can never be distinguished, and hence G is not strongly periodically D-detectable. □

To illustrate the construction, we provide two examples. Let $\mathcal{A}_1$ and $\mathcal{A}_2$ be the total DFAs over $\{0, 1\}$ depicted in Figure 6, where $L(\mathcal{A}_1)$ consists of strings of odd length, and $L(\mathcal{A}_2)$ of strings of even length. Our construction results in

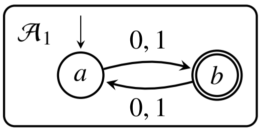

Figure 6. The DFAs $\mathcal{A}_1$ and $\mathcal{A}_2$ over $\Sigma = \{0, 1\}$.

a DES G depicted in Figure 7. Since $L(\mathcal{A}_1) \cap L(\mathcal{A}_2) = \emptyset$, the construction gives that G is strongly periodically D-detectable; indeed, it is evident from the observer of G depicted in Figure 8 where, on every trajectory, states q^- and q_1^+ can be periodically distinguished.

For our second example, let $\mathcal{A}_1$ be as above, and let $\mathcal{A}_2$ be the total minimal DFA such that $L(\mathcal{A}_2)$ consists of all strings starting with event 0. Our construction results in a DES H depicted in Figure 9. Here, $L(\mathcal{A}_1) \cap L(\mathcal{A}_2) \neq \emptyset$, and hence H is not strongly periodically D-detectable; indeed, Figure 10 shows a fragment of the observer of H that contains a trajectory through the self-loop in state $\{q^-, q_1^+, q_2^+\}$, where the states q^- and q_1^+ cannot be periodically distinguished.

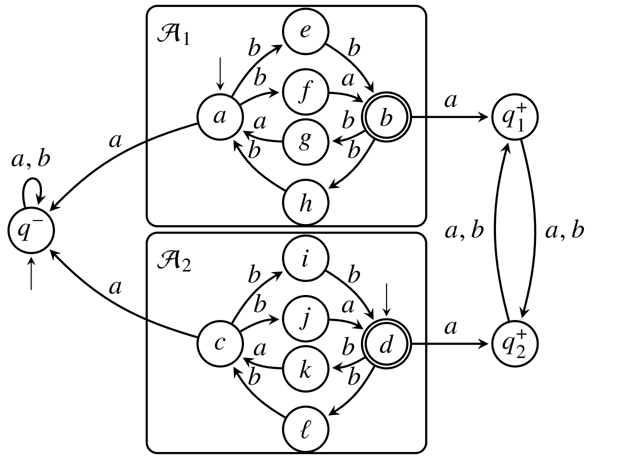

Figure 7. The DES G with $\Sigma' = \{a, b\}$ constructed from the DFAs of Figure 6, where 0 is encoded as ba and 1 as bb.

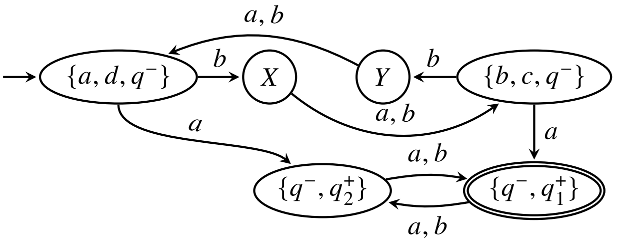

Figure 8. The observer of G; states marked by double circles contain indistinguishable states of G; here $X = \{e, f, k, \ell, q^-\}$ and $Y = \{g, h, i, j, q^-\}$.

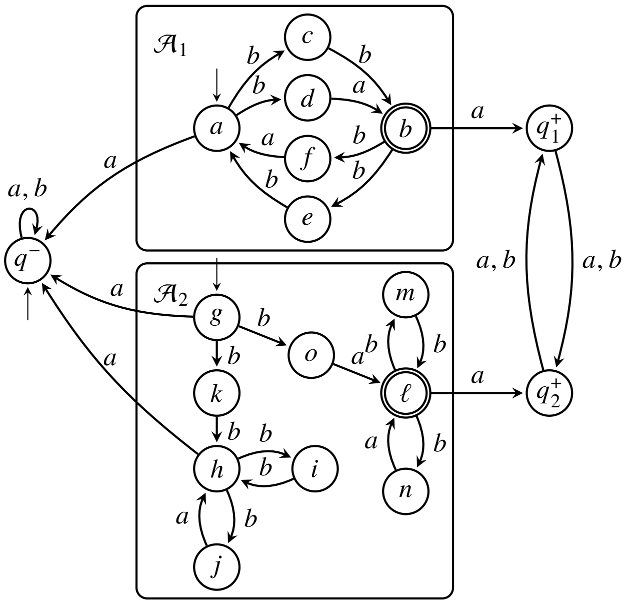

Figure 9. The DES H with $\Sigma' = \{a, b\}$.

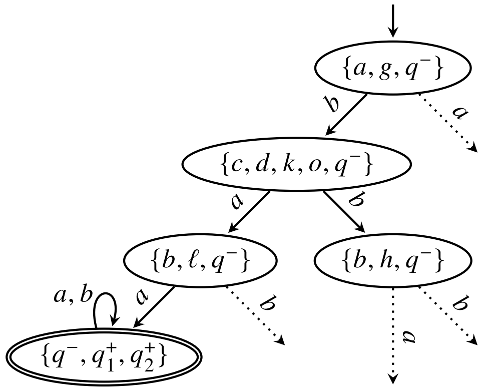

Figure 10. A fragment of the observer of H showing the trajectory that cannot periodically distinguish states q^- and q_1^+.

The case of a single observable event

In the previous subsection, we have shown that deciding strong periodic D-detectability is PSPACE-complete for DESs with at least two observable events. We now show that the problem is still more difficult than its non-periodic counterpart even for DESs having only a single observable event.

Theorem 4. *Deciding strong periodic D-detectability for DESs with a single observable event is* NP*-complete.*

Proof. Consider a DES with a single observable event $\{a\}$. If the DES has unobservable events, we can eliminate them as follows. First, we replace each unobservable transition, i.e., a transition of the form (p, u, q) with u being an unobservable event, by an ε-transition (p, ε, q). Then, we use the standard technique to eliminate ε-transitions (Hopcroft et al., 2007). This elimination results in a DES and can be done in polynomial time. Therefore, without loss of generality, we may assume that the DES is of the form $G = (Q, \{a\}, \delta, I)$.

A proof that we can decide strong periodic D-detectability of G in nondeterministic polynomial time uses a so-called *fast matrix multiplication technique*. The basic idea of this technique is to represent the transition function δ of G as a binary matrix M, where $M[i, j] = 1$ if and only if there is a transition from state i to state j in G. Then, for $r \geq 1$, M^r represents the reachability in G under the string a^r. Furthermore, using the fact that $M^2 = M \times M$, $M^4 = M^2 \times M^2$, etc., we can compute M^r by $O(\log r)$ matrix multiplications, each multiplication in polynomial time. For more details and examples on this technique, we refer to Masopust (2018b).

Assume that G has n states. Then the observer of G consists of a sequence of k states followed by a cycle consisting of ℓ states, that is, the language of G is $a^k(a^\ell)^*$. Since the number of states of the observer of G is at most 2^n, $k + \ell \leq 2^n$.

Now, G is strongly periodically D-detectable if and only if there is a state $X \subseteq Q$ in the cycle of the observer of G (we assume that the observer is constructed by the standard subset construction (Hopcroft et al., 2007)) that is disjoint from the specification, that is, $X \cap T_{spec} = \emptyset$. Indeed, to check whether $X \cap T_{spec} = \emptyset$ can be done in polynomial time. It remains to show how to find X in polynomial time. This means to find $m \leq 2^n$ such that $\delta(I, a^{2^n+m}) \cap T_{spec} = \emptyset$. However, an NP algorithm can guess m in binary and verify the guess in polynomial time by computing $\delta(I, a^{2^n+m})$ using the fast matrix multiplication, cf. Masopust (2018b) for more details and an example.

To prove NP-hardness, we adapt the construction of Stockmeyer and Meyer (1973) encoding a boolean formula in 3CNF in the form of a unary NFA.[1] For an illustration, the reader may follow Example 5 in parallel with the proof.

Let φ be a formula in 3CNF with n variables and m clauses, and let C_k be the set of literals in the kth clause, $k = 1, \ldots, m$. The assignment to the variables is represented by a binary vector of length n. Let $p_1, \ldots, p_n$ be the first n prime numbers. For a natural number z congruent with 0 or 1 modulo p_i, for every $i = 1, \ldots, n$, we say that z satisfies φ if the assignment $(z \bmod p_1, z \bmod p_2, \ldots, z \bmod p_n) \in \{0, 1\}^n$ satisfies φ. Let $\mathcal{A}_0$ be an NFA recognizing the language of the expression $\bigcup_{i=1}^{n} \bigcup_{j=2}^{p_i-1} 0^j \cdot (0^{p_i})^*$, that is, $L(\mathcal{A}_0) = \{0^z \mid \exists k \leq n, z \not\equiv 0 \bmod p_k \text{ and } z \not\equiv 1 \bmod p_k\}$ is the set of all natural numbers that *do not* encode an assignment to the variables.

For each C_k, we construct an NFA $\mathcal{A}_k$ such that if $0^z \in L(\mathcal{A}_k)$ and z is an assignment, then z does not assign 1 (`true`) to any literal in C_k; e.g., if $C_k = \{x_r, \neg x_s, x_t\}$, $1 \leq r, s, t \leq n$ and r, s, t are distinct, let z_k be the unique integer such that $0 \leq z_k < p_r p_s p_t$, $z_k \equiv 0 \bmod p_r$, $z_k \equiv 1 \bmod p_s$, and $z_k \equiv 0 \bmod p_t$. Then $L(\mathcal{A}_k) = 0^{z_k} \cdot (0^{p_r p_s p_t})^*$.

Now, φ is satisfiable if and only if there exists z such that z encodes an assignment to φ and $0^z \notin L(\mathcal{A}_k)$ for all $1 \leq k \leq m$, which is if and only if $L(\mathcal{A}_0) \cup \bigcup_{k=1}^{m} L(\mathcal{A}_k) \neq 0^*$. The construction of all the automata $\mathcal{A}_0, \mathcal{A}_1, \ldots, \mathcal{A}_k$ can be done in polynomial time (Stockmeyer and Meyer, 1973).

Let $\mathcal{A}$ denote the NFA obtained by taking all the automata $\mathcal{A}_0, \mathcal{A}_1, \ldots, \mathcal{A}_k$ as a single NFA, and let $p = \Pi_{i=1}^{n} p_i$. If z encodes an assignment to φ, then so does $z + cp$ for any natural c: indeed, if $z \equiv x_i \bmod p_i$, then $z + cp \equiv x_i \bmod p_i$, for every $1 \leq i \leq n$, as well. Thus, if $0^z \notin L(\mathcal{A}_k)$ for all k, then $0^z(0^p)^* \cap L(\mathcal{A}) = \emptyset$. Since both languages are infinite, the minimal DFA recognizing $L(\mathcal{A})$ must have a nontrivial cycle alternating between marked and non-marked states, and hence the same holds for the observer of $\mathcal{A}$.

We now show that φ is satisfiable if and only if $\mathcal{A}$ is strongly periodically D-detectable with respect to the specification T_{spec} consisting of all pairs of states, where the states come from two different automata $\mathcal{A}_i$ and $\mathcal{A}_j$, $i \neq j$, and at least one state is marked in its automaton.

Assume that φ is satisfiable. As shown above, this is if and only if $L(\mathcal{A}) \neq 0^*$. We have further shown that $L(\mathcal{A})$ is infinite and that the observer of $\mathcal{A}$ consists of a single trajectory with a non-marked state, X, in its cycle part, i.e., X contains only non-marked states of the automata $\mathcal{A}_0, \ldots, \mathcal{A}_k$. Since T_{spec} consists of pairs of states of two different automata $\mathcal{A}_i$ and $\mathcal{A}_j$, $0 \leq i \neq j \leq k$, with at least one state marked in its automaton, we have that $(X \times X) \cap T_{spec} = \emptyset$. Therefore, $\mathcal{A}$ is strongly periodically D-detectable with respect to the specification T_{spec}.

On the other hand, assume that φ is not satisfiable. As shown above, this is if and only if $L(\mathcal{A}) = 0^*$. But then every state of the observer of $\mathcal{A}$ must be marked, i.e., every state, X, of the observer contains a marked state of some $\mathcal{A}_i$, $0 \leq i \leq k$, and hence $(X \times X) \cap T_{spec} \neq \emptyset$. Therefore, $\mathcal{A}$ is not strongly periodically D-detectable with respect to T_{spec}. □

The following example illustrates the construction.

Example 5. Let $\varphi_1 = (x \vee y) \wedge (\neg x \vee y)$ and $\varphi_2 = x \wedge \neg x$. Obviously, φ_1 is satisfiable and φ_2 is not. For both formulae,

[1] A boolean formula is built from propositional variables, operators conjunction, disjunction, and negation, and parentheses. A formula is satisfiable if there is an assignment of 1 (`true`) and 0 (`false`) to its variables making it `true`. A literal is a variable or its negation. A clause is a disjunction of literals. A formula is in conjunctive normal form (CNF) if it is a conjunction of clauses; e.g., $\varphi = (x \vee y \vee z) \wedge (\neg x \vee y \vee z)$ is a formula in CNF with two clauses $x \vee y \vee z$ and $\neg x \vee y \vee z$. If every clause has at most three literals, the formula is in 3CNF. Given a formula in 3CNF, 3CNF satisfiability asks whether the formula is satisfiable; e.g., φ is satisfiable for $(x, y, z) = (0, 1, 0)$. 3CNF satisfiability is NP-complete.

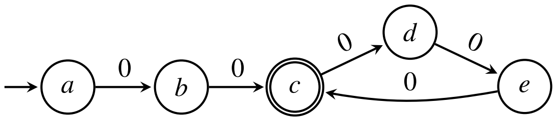

Figure 11. Automaton $\mathcal{A}_{1,0}$.

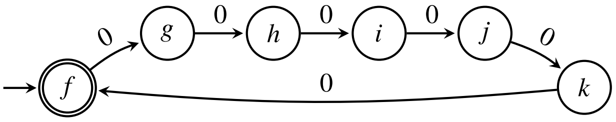

Figure 12. Automaton $\mathcal{A}_{1,1}$.

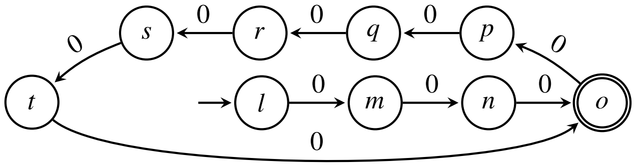

Figure 13. Automaton $\mathcal{A}_{1,2}$.

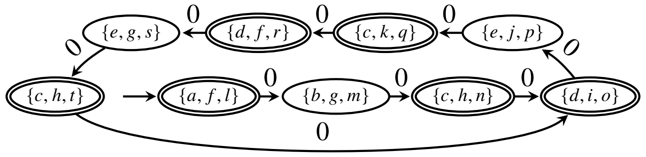

Figure 14. The observer of the NFA $\mathcal{A}_1$; states marked by a double circle contain indistinguishable states of $\mathcal{A}_1$.

we can construct the unary automata $\mathcal{A}_1$ and $\mathcal{A}_2$, respectively, and show that $\mathcal{A}_1$ is strongly periodically D-detectable while $\mathcal{A}_2$ is not.

The formula $\varphi_1 = (x \vee y) \wedge (\neg x \vee y)$ has two variables, and therefore we set $p_1 = 2$ and $p_2 = 3$, the first two prime numbers. The automaton $\mathcal{A}_{1,0}$, depicted in Figure 11, recognizes the language $0^2(0^3)^*$ of all strings that do not encode the assignment to φ_1. Since φ_1 consists of two clauses, we further construct two automata: $\mathcal{A}_{1,1}$ recognizing the language $(0^6)^*$, and $\mathcal{A}_{1,2}$ recognizing the language $0^3(0^6)^*$; the reader can verify that if $0^z \in L(\mathcal{A}_{1,1})$ and z is an assignment, then z assigns `true` neither to x nor to y, and if $0^z \in L(\mathcal{A}_{1,2})$ and z is an assignment, then z assigns `true` to x (that is, it assigns `false` to the literal $\neg x$) and `false` to y. The automata are depicted in Figures 12 and 13, respectively. The specification T_{spec} is then the symmetric closure of the relation

$$\begin{aligned} T = &\{c\} \times \{f, g, h, i, j, k, l, m, n, o, p, q, r, s, t\} \\ &\cup \{a, b, c, d, e, l, m, n, o, p, q, r, s, t\} \times \{f\} \\ &\cup \{a, b, c, d, e, f, g, h, i, j, k\} \times \{o\} \,. \end{aligned}$$

Let $\mathcal{A}_1$ be the NFA consisting of the automata $\mathcal{A}_{1,0}$, $\mathcal{A}_{1,1}$, and $\mathcal{A}_{1,2}$. The observer of $\mathcal{A}_1$ is depicted in Figure 14. The reader can see that the observer contains a cycle where the state $\{e, j, p\}$ appears periodically. Since this state does not contain any pair from the specification T_{spec}, the NFA $\mathcal{A}_1$ is strongly periodically D-detectable.

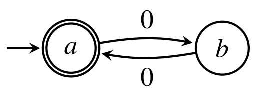

Figure 15. Automaton $\mathcal{A}_{2,1}$.

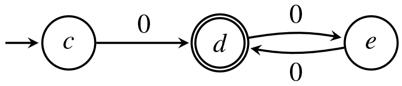

Figure 16. Automaton $\mathcal{A}_{2,2}$.

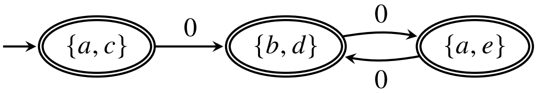

Figure 17. The observer of the DES $\mathcal{A}_2$; states marked by a double circle contain indistinguishable states of $\mathcal{A}_2$.

Similarly, we construct the automata for the formula $\varphi_2 = x \wedge \neg x$. The formula has only one variable, and therefore we set $p_1 = 2$. Then the language $L(\mathcal{A}_{2,0}) = \emptyset$, since any natural number modulo 2 is either 0 or 1. Therefore, the automaton $\mathcal{A}_{2,0}$ consists of a single non-marked state, and hence we can ignore it in the following construction. Since φ_2 consists of two clauses, we construct two automata $\mathcal{A}_{2,1}$ recognizing the language $(0^2)^*$, and $\mathcal{A}_{2,2}$ recognizing the language $0(0^2)^*$; the reader can verify that if $0^z \in L(\mathcal{A}_{2,1})$ and z is an assignment, then z does not assign `true` to x, and that if $0^z \in L(\mathcal{A}_{2,2})$ and z is an assignment, then z assigns `true` to x. The automata are depicted in Figures 15 and 16, respectively. The specification T_{spec} is then the symmetric closure of the relation $\{(a, c), (a, d), (a, e), (b, d)\}$. Let $\mathcal{A}_2$ be the NFA consisting of both automata $\mathcal{A}_{2,1}$ and $\mathcal{A}_{2,2}$. The observer of $\mathcal{A}_2$ is depicted in Figure 17. The reader can see that the state set of the observer consists of states that are indistinguishable by the specification T_{spec}, and hence $\mathcal{A}_2$ is not strongly periodically D-detectable. ⋄

As already pointed out above, the observer of a unary DES, that is, of a DES with a single observable event, consists of a single trajectory ending with a cycle. We now formulate the consequences of this observation showing the relationship between weak and strong detectabilities.

Theorem 6. *For a DES with a single observable event, deciding weak detectability coincides with deciding strong detectability, and deciding weak periodic detectability coincides with deciding strong periodic detectability. The same holds true for D-detectability.*

Proof. Let G be a DES with n states and a single observable event $\Sigma_o = \{a\}$. Then the observer of G consists of a sequence of k states followed by a cycle consisting of ℓ states, that is, the language of G is $a^k(a^\ell)^*$ with $k + \ell \leq 2^n$. The existence of the cycle follows from the two assumptions on Page 3. Since the observer of G consists of a single infinite trajectory, deciding strong (periodic) detectability coincides with deciding weak (periodic) detectability. The same result holds when we replace detectability with D-detectability. □

5 A Tractable Case

In the previous sections, we have shown that deciding strong periodic D-detectability is a difficult problem for DES modeled by NFAs. Therefore, we now discuss a special case of systems for which the verification of strong periodic D-detectability is polynomial.

Looking for DESs for which the problem is tractable, we first inspect the proof of Theorem 3. This proof is based on the intersection emptiness problem for DFAs that was shown to be PSpace-complete by Kozen (1977). His proof relies on DFAs with cycles. Allowing only self-loops instead of cycles in the DFAs makes the problem easier (Masopust and Krötzsch, 2019). Therefore, we consider DESs modeled by NFAs where all cycles in the transition graph are self-loops. Such NFAs recognize a strict subclass of regular languages that are strictly included in *star-free languages* (Brzozowski and Fich, 1980; Krötzsch et al., 2017). Star-free languages are languages definable by *linear temporal logic*, which is a logic widely used as a specification language in automated verification. Notice that a suitable self-loop may be added to any state, if needed, and hence the system may fulfill the deadlock-free requirement.

A practical motivation for such systems comes from the following observation. Every infinite trajectory in a system describes a task that is possibly repeated ad infinitum. Every task of such a trajectory is a finite sequence of events. For example, imagine the situation where a user accesses an article on Wikipedia. Displaying the article, Wikipedia imports some parts from its knowledge graph called Wikidata. Thus, every time a user accesses a Wikipedia article, the server connects to Wikidata, sends a query to Wikidata, obtains a response, and terminates the connection. The response is then used to build the page to be displayed.

Such a system is a parallel composition of many terminating processes, where each process executes the following sequence of steps: it opens the connection to the database, then it sends the query to the database, it receives the response, and finally it terminates the connection to the database. On the side of the database, such a communication may be modeled as a single state that idles in that state or connects the user to the database, and hence we may model it as a single state with several self-loops; indeed, we abstract here from the details of opening and closing the connection, but it is not important for our modeling.

Notice that the users come continuously, and hence the system is infinite in general. However, if we are interested in the properties of such a system in a finite period of time, then the system, considered as the parallel composition of all the processes that were active during that period of time, is a huge automaton where every loop is in the form of a self-loop, and hence in the form we discuss in this section.

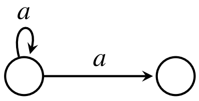

Figure 18. The forbidden pattern of rpoNFAs.

Let $\mathcal{A} = (Q, \Sigma, \delta, I, F)$ be an NFA. The reachability relation $\leq$ on the state set Q is defined by $p \leq q$ if there is $w \in \Sigma^*$ such that $q \in \delta(p, w)$. 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 in the sense that the pattern of Figure 18 does not appear. Formally, for every state q and every event a, if $q \in \delta(q, a)$ then $\delta(q, a) = \{q\}$. We now formally define so-called rpoDES. The name comes from restricted partially ordered DES.

Definition 7. Let $G = (Q, \Sigma, \delta, I)$ be a DES with Σ_o being the set of observable events. Let $P \colon \Sigma^* \to \Sigma_o^*$ be the corresponding projection. We say that G is an rpoDES if the NFA $P(G) = (Q, \Sigma_o, \delta', I)$ obtained from G by replacing every transition (p, a, q) by $(p, P(a), q)$, and by eliminating the ε-transitions (Hopcroft et al., 2007) is an rpoNFA.

Notice that $P(G)$ is an NFA that can be constructed from G in polynomial time (Hopcroft et al., 2007). Therefore, the question whether a DES is an rpoDES is decidable in polynomial time.

What do we know about rpoDES? Deciding weak (periodic) detectability for rpoDESs is PSpace-complete (Masopust, 2018a), and hence so is deciding weak (periodic) D-detectability.

We now show that the complexity of deciding strong periodic D-detectability for rpoDESs coincides with the complexity of deciding strong D-detectability.

Theorem 8. *Deciding strong periodic D-detectability for rpoDESs is* NL*-complete.*

Proof. To prove the theorem, we show that the observer of an rpoDES is a partially ordered DFA. Then, since there are no nontrivial cycles in the observer, strong periodic D-detectability coincides with strong D-detectability. Theorem 1 then finishes the proof.

Let $\mathcal{A} = (Q, \Sigma_o, \delta, I, F)$ be an rpoNFA. We show that the DFA $\mathcal{D}$ computed from $\mathcal{A}$ by the standard subset construction (the observer) is partially ordered. To this end, let $X = \{p_1, \dots, p_n\}$ with $p_i < p_j$ for $i < j$ be a state of $\mathcal{D}$, and let $w \in \Sigma^*$ be a nonempty string such that $\delta_{\mathcal{D}}(X, w) = X$. First, we show that $\delta(p_i, w) = \{p_i\}$ for all i. For the sake of contradiction, let $1 \leq k \leq n$ be the minimal integer such that $\delta(p_k, w) \neq \{p_k\}$. Since $X = \delta(X, w) = \cup_{i=1}^{n} \delta(p_i, w)$, $\delta(p_i, w) = \{p_i\}$ for all $i < k$, and $p_k < p_i \leq q$ for every $q \in$

$\delta(p_i, w)$, for all $i > k$, we have that $p_k \notin \cup_{i=1}^{n} \delta(p_i, w) = X$, which is a contradiction. Therefore, $p_k \in \delta(p_k, w)$, and the definition of rpoNFAs implies that every event of w is in a self-loop in state p_k. Because rpoNFAs have no choice between staying in the state and leaving it under the same event, $\delta(p_k, w) = \{p_k\}$. Thus, for $i = 1, \ldots, n$, $\delta(p_i, a) = \{p_i\}$ for every event a occurring in w. Consequently, for any state Y of $\mathcal{D}$ and any strings w_1 and w_2, if $\delta(X, w_1) = Y$ and $\delta(Y, w_2) = X$, the previous argument gives that $X = Y$, and hence $\mathcal{D}$ is partially ordered. □

Finally, we have the following corollary of Theorem 1.

Corollary 9. *Deciding whether an rpoDES with a single observable event is strongly (D-)detectable is* NL*-complete.*

Proof. This result is an immediate consequence of the proof of Theorem 1, since in the hardness part we actually construct a unary rpoNFA, cf. Figure 1. □

6 Conclusions

In this paper, we answered the open question concerning the complexity of deciding whether a DES is strongly periodically D-detectable. We further provided the full complexity picture of the problem of deciding D-detectability. Since our results for general DESs are mainly negative, we also discussed a class of DESs, so-called rpoDESs, for which the complexity of deciding strong (periodic) D-detectability is tractable.

Acknowledgements

This research was supported by the Ministry of Education, Youth and Sports under the INTER-EXCELLENCE project LTAUSA19098 and by the University projects IGA PrF 2020 019 and IGA PrF 2021 022.

The authors acknowledge valuable comments and suggestions of the anonymous referees.

References

Alves, M.V.S., Basilio, J.C., 2019. State estimation and detectability of networked discrete event systems with multichannel communication networks, in: American Control Conference, IEEE. pp. 5602–5607.

Arora, S., Barak, B., 2009. Computational Complexity – A Modern Approach. Cambridge University Press.

Balun, J., Masopust, T., 2020. On verification of strong periodic d-detectability for discrete event systems, in: Workshop on Discrete Event Systems, IFAC. pp. 263–268.

Brzozowski, J.A., Fich, F.E., 1980. Languages of R-trivial monoids. Journal of Computer and Systems Sciences 20, 32–49.

Caines, P.E., Greiner, R., Wang, S., 1988. Dynamical logic observers for finite automata, in: Conference on Decision and Control, pp. 226–233.

Cho, S., Huynh, D.T., 1991. Finite-automaton aperiodicity is PSPACE-complete. Theoretical Computer Science 88, 99–116.

Garey, M.R., Johnson, D.S., 1990. Computers and Intractability; A Guide to the Theory of NP-Completeness. W. H. Freeman & Co., New York, NY, USA.

Hadjicostis, C.N., 2020. Estimation and Inference in Discrete Event Systems. Springer.

Hopcroft, J.E., Motwani, R., Ullman, J.D., 2007. Introduction to automata theory, languages, and computation. 3rd ed., Addison-Wesley.

Immerman, N., 1988. Nondeterministic space is closed under complementation. SIAM Journal on Computing 17, 935–938.

Kozen, D., 1977. Lower bounds for natural proof systems, in: Symposium on Foundations of Computer Science, pp. 254–266.

Krötzsch, M., Masopust, T., Thomazo, M., 2017. Complexity of universality and related problems for partially ordered NFAs. Information and Computation 255, 177–192.

Lin, F., 2011. Opacity of discrete event systems and its applications. Automatica 47, 496–503.

Masopust, T., 2018a. Complexity of deciding detectability in discrete event systems. Automatica 93, 257–261.

Masopust, T., 2018b. Complexity of verifying nonblockingness in modular supervisory control. IEEE Transactions on Automatic Control 63, 602–607.

Masopust, T., Krötzsch, M., 2019. Partially ordered automata and piecewise testability. ArXiv:1907.13115.

Masopust, T., Yin, X., 2019. Complexity of detectability, opacity and A-diagnosability for modular discrete event systems. Automatica 101, 290–295.

Ozveren, C.M., Willsky, A.S., 1990. Observability of discrete event dynamic systems. IEEE Transactions on Automatic Control 35, 797–806.

Ramadge, P.J., 1986. Observability of discrete event systems, in: Conference on Decision and Control, pp. 1108–1112.

Shu, S., Lin, F., 2011. Generalized detectability for discrete event systems. Systems & Control Letters 60, 310–317.

Shu, S., Lin, F., 2013. Delayed detectability of discrete event systems. IEEE Transactions on Automatic Control 58, 862–875.

Shu, S., Lin, F., Ying, H., 2007. Detectability of discrete event systems. IEEE Transactions on Automatic Control 52, 2356–2359.

Stockmeyer, L.J., Meyer, A.R., 1973. Word problems requiring exponential time: Preliminary report, in: ACM Symposium on the Theory of Computing, ACM. pp. 1–9.

Szelepcsényi, R., 1988. The method of forced enumeration for nondeterministic automata. Acta Informatica 26, 279–284.

Yin, X., Lafortune, S., 2017. Verification complexity of a class of observational properties for modular discrete events systems. Automatica 83, 199–205.

Zhang, K., 2017. The problem of determining the weak (periodic) detectability of discrete event systems is PSPACE-complete. Automatica 81, 217–220.

Zhang, K., Giua, A., 2019. Revisiting delayed strong detectability of discrete-event systems. ArXiv:1910.13768.

Zhang, K., Zhang, L., Xie, L., 2020. Discrete-Time and Discrete-Space Dynamical Systems. Springer.