The Complexity of Diagnosability and Opacity Verification for Petri Nets

Béatrice Bérard
Sorbonne Universités, UPMC Univ. Paris 06, LIP6, CNRS, Paris, France
beatrice.berard@lip6.fr

Stefan Haar
INRIA and LSV, ENS Paris-Saclay and CNRS, Université Paris-Saclay, France
stefan.haar@inria.fr

Sylvain Schmitz
LSV, ENS Paris-Saclay and CNRS, Université Paris-Saclay, France
schmitz@lsv.fr

Stefan Schwoon
LSV, ENS Paris-Saclay and CNRS, Université Paris-Saclay, and INRIA, France
schwoon@lsv.fr

Abstract. *Diagnosability* and *opacity* are two well-studied problems in discrete-event systems. We revisit these two problems with respect to expressiveness and complexity issues.

We first relate different notions of diagnosability and opacity. We consider in particular fairness issues and extend the definition of Germanos et al. [ACM TECS, 2015] of weakly fair diagnosability for safe Petri nets to general Petri nets and to opacity questions.

Second, we provide a global picture of complexity results for the verification of diagnosability and opacity. We show that diagnosability is NL-complete for finite state systems, PSPACE-complete for safe convergent Petri nets (even with fairness), and EXPSPACE-complete for general Petri nets without fairness, while non diagnosability is inter-reducible with reachability when fault events are not weakly fair. Opacity is ESPACE-complete for safe Petri nets (even with fairness) and undecidable for general Petri nets already without fairness.

Keywords: Diagnosability, Opacity, Verification, Complexity, Petri nets

1. Introduction

Diagnosability and opacity are two aspects of partially observable discrete-event systems that have each received considerable attention. Although they are usually considered separately, they form a

dual pair of tasks: an observer watches the current execution of a known system, where only some events are visible. As this execution evolves, the observer continually attempts to deduce whether the execution satisfies some property: in diagnosis, the observer strives to detect the occurrence of some *fault* event, while in opacity the observer may be hostile, and one requires to prevent her from being certain that a *secret* has occurred. These deductions are made on the basis of a finite prefix of the current execution; we will refer to this as the *Finite-Observation Principle*.

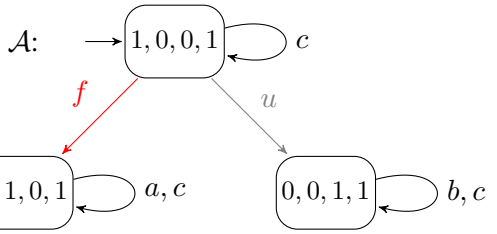
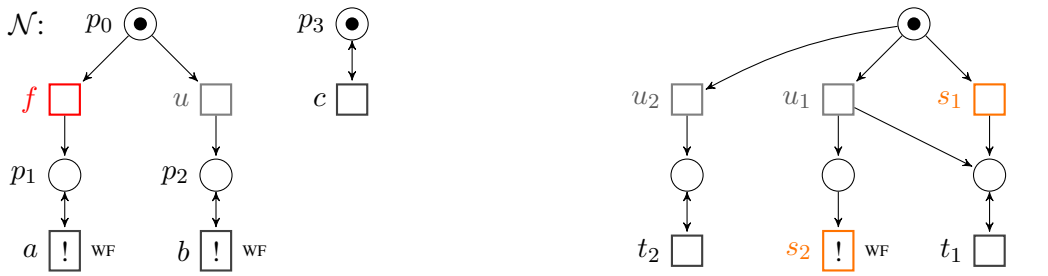


Figure 1. Example of a partially observable system.

Consider for instance the labelled transition system (LTS) of Figure 1, and suppose that an observer who knows this system wants to discover whether action f occurred. Moreover, let us assume that the observer can see the occurrences of actions a, b, and c, but not those of f nor u. Now, if the observer sees any occurrence of a, he can indeed deduce that f has happened, while observing some b allows to deduce that it has not. On the other hand, an observation of the form c^+ is ambiguous; it could be the result of staying in the initial state and possibly moving to the right-hand state, or that of executing action f and moving to the left-hand state.

Diagnosability. A system is *diagnosable* if, after the occurrence of a fault (which itself is invisible, like f in Figure 1), it is always possible to deduce that a fault has happened after a sufficiently long observation. For instance, the system in Figure 1 would be considered undiagnosable due to the presence of an ambiguous observation. If action c was removed, the system would become diagnosable, as a would occur immediately after the fault. A formal-language framework for both diagnosis and the analysis of diagnosability was introduced by Sampath et al. [1] in the context of finite automata, for which diagnosability can be checked in polynomial time w.r.t. the number of reachable states [2, 3].

Weak Fairness. The diagnosability framework from [1] is suitable for *sequential* but not for concurrent systems. Consider for instance the Petri net $\mathcal{N}$ shown in Figure 2(a), inspired by an example from [4]. It consists of two entirely independent components. Again we assume that an outside observer can see the actions a, b, c but not f, u, where f is a fault. If the system is eager to progress, then it is intuitively diagnosable because f will lead to the observation a, and the latter cannot occur otherwise. However, note that the naïve translation of this Petri net into a labelled transition system results in the system $\mathcal{A}$ from Figure 1, where a state is a tuple of token values for the places ordered from p_0 to p_3. If we now apply the methods from [1], the net will be declared non-diagnosable because the two executions fc^ω and uc^ω are observationally equivalent. If the component on the right-hand side is removed (which amounts to removing the fourth component in states and the c loops in Figure 1),



(a) Petri net that is not diagnosable but WF-diagnosable.

(b) Petri net that is opaque but not WF-opaque.

Figure 2. Examples for diagnosis and opacity in weakly fair Petri nets.

the system becomes diagnosable. In other words, the presence of the right-hand side component fully determines whether the system is diagnosable or not, although it is not related in any way to the faulty behaviour.

Such effects have motivated the study of diagnosability notions suitable for concurrent systems [5, 6, 7, 8, 4]. We focus in this paper on the *weakly fair* (WF) behaviours of the system in the sense of Vogler [9]: In a weakly fair run, no WF transition t that becomes enabled will remain idle indefinitely; either t itself eventually fires, or some conflicting transition does, thus (momentarily) disabling t. This notion of weak fairness is slightly weaker than the one studied by Jančar [10], but has the advantage that for safe Petri nets, the maximal partially ordered runs are exactly those generated by WF runs; and conversely, interleavings of such partially ordered runs yield WF firing sequences.

Under WF semantics, a fault can be diagnosed when the observations made so far can no longer be those of any fault-free WF execution. Then a net is WF-diagnosable if every infinite WF faulty execution has a prefix allowing the fault to be diagnosed. Under this characterisation, the net in Figure 2(a) is considered WF-diagnosable because the occurrence of f implies that a will eventually happen, and the observation of a is incompatible with any correct execution. In [8, 4], WF-diagnosability was shown decidable for safe PNs.

Opacity. The related notion of opacity was introduced for general transition systems in [11]. The system has a secret subset of executions which is *opaque* if for any secret execution, there is a non-secret one with the same observation: an observer can never be sure whether the current execution is secret or not. State-based variants (where the observation of an execution is related to the associated sequence of states) were later studied for instance in [12, 13] and shown in [12] to be PSPACE-complete for finite transition systems. Other language-based variants were also studied in [14, 15]. Our focus here is on secret executions defined by the occurrence of some secret *transition*, which makes opacity a more general notion than *Strong Nondeterministic Non-Interference (SNNI)* or, equivalently, *Non-deducibility on Composition (NDC)*: for these two properties, the observation mask is fixed and requires that (i) any non secret action is observable; (ii) the observation mask is the identity function for all non-secret actions. SNNI and NDC were studied in [16] on safe Petri nets and proven decidable in [17] on general Petri nets.

As for diagnosability, the notions of opacity developed for LTSs are not necessarily suitable for

Table 1. Complexity results for diagnosability and opacity.

Model	Diagnosability	Opacity
finite LTS	NL-c.	PSPACE-c. [12]
safe (WF-)PN	PSPACE-c.	ESPACE-c.
PN	EXPSPACE-c.	undecidable
strict WF-PN	$\mathsf{PNReach} \leq_m^{\mathsf{P}} \neg\mathsf{Diag} \leq_m^{\mathsf{EXP}} \mathsf{PNReach}$	

concurrent systems. Consider the net in Figure 2(b) and suppose that transitions s_1 and s_2 are secret, t_1 and t_2 visible for an attacker, and u_1 and u_2 invisible. Then an observation of t_2 shows that no secret has occurred, but observing occurrences of t_1 does not suffice to prove that s_1 or s_2 have occurred; therefore, according to traditional definitions, the Petri net from Figure 2(b) would be considered opaque. However, assuming that s_2 behaves in a *weakly fair* way, then, on observing t_1, the attacker can deduce with certainty that either s_1 has already fired, or s_2 will inevitably do so (or already has). We introduce a notion of *WF-opacity* that takes this into account and declares this net non-opaque.

Contributions. We first establish the relationships between several notions of diagnosability from the literature in the general setting of transition systems (Section 2). This results in a strict hierarchy of diagnosability definitions (Lemma 2.6), with our main definition of *trace-diagnosability* as the least stringent one. The hierarchy collapses to two levels in *convergent* systems, i.e., those having no infinite sequence of unobservable events. Thus our results are widely applicable under this frequently-made convergence assumption, which ensures that an observer will always eventually see some event. We furthermore demonstrate exactly why convergence is required in the classical approach to diagnosability through *twin-plants* (Remark 2.3) and how to soundly forgo this assumption (Lemma 2.5).

For concurrent systems, we extend in Section 3.3 the notion of WF-diagnosability from [4] to general Petri nets, building on the work on fairness in Petri nets of Howell et al. [18]. We define and study a refinement of opacity with weak fairness that similarly eschews the problems pointed out in Figure 2(b). We believe these are important conceptual contributions, as the usual notions of diagnosis and opacity are rather inadequate in a concurrent setting.

Moreover, we provide an almost complete picture of the complexity for diagnosability and opacity analysis for Petri nets with general and weakly fair semantics; see Table 1. For a start, we complete the picture for finite LTSs and show that diagnosability is complete for non-deterministic logarithmic space, while opacity had already been shown PSPACE-complete in [12]. For Petri nets, the outcome is roughly consistent with the 'rules of thumb' in Esparza's survey [19] when viewing diagnosability as a linear-time property and opacity as an inclusion problem. The salient points are as follows:

- As an auxiliary result for our lower bounds in Section 4, we provide in Appendix A a proof that trace inclusion in safe Petri nets is ESPACE-complete (Proposition 4.3), where ESPACE is the class of problems that can be solved in deterministic space $2^{O(n)}$. An ESPACE-complete problem is also EXPSPACE-complete—i.e., for deterministic space $2^{poly(n)}$—but the converse

is not necessarily true. The result seems to have been known since [20] but to the best of our knowledge no actual proof had been published so far.

- The upper bounds for safe Petri nets in Section 5.1 also hold for the more adequate, weakly fair variants of diagnosability and opacity. For WF-diagnosability, we analyse the complexity of the algorithm of Germanos et al. [4] for convergent nets and provide a PSPACE upper bound. This might come as a surprise, as this is a branching-time property (see Definition 3.3), which cannot be expressed in CTL due to its fairness aspect, while CTL* model-checking would yield an EXP upper bound. We also give an algorithm in ESPACE for WF-opacity.

- For general Petri nets, we leave the decidability of WF-diagnosability open, but nevertheless show two positive results in Section 5.2 in restricted settings.

  The first one is a tight EXPSPACE upper bound for diagnosability. Cabasino et al. [21] have proposed a procedure for a slightly different notion of diagnosability that turns out to be equivalent to ours for convergent nets. Since the method of [21] constructs coverability graphs with worst-case Ackermann size, our method represents a considerable improvement in the convergent case.

  The second one is an algorithm checking for non WF-diagnosability in convergent nets when fault transitions are not weakly fair, i.e., when a fault is a *possible* outcome in the system but not one that is *required* to happen. We call such systems *strict*, and as illustrated in [4, Sec. 5], this is a reasonable assumption in practice. Our complexity analysis uses a fragment of LTL studied by Jančar [10] and shares its complexity: at least as hard as reachability (noted 'PNReach' in Table 1), and at most exponentially harder; recall that the complexity of reachability in general Petri nets is a major open problem [22], with a gigantic gap between a forty years old EXPSPACE lower bound [23] and a cubic Ackermann upper bound obtained recently in [24].

The paper is organised as follows: Section 2 presents diagnosability and opacity in the general setting of LTSs; Section 3 is dedicated to fairness in Petri nets and the associated notions of WF-diagnosability and WF-opacity; we establish all our complexity lower bounds in Section 4 and all our complexity upper bounds in Section 5; finally, as a side contribution, Appendix A presents the ESPACE-completeness of the trace inclusion problem in safe Petri nets. This work extends [25] and fixes an oversight in the proof of Proposition 5.4, where our proof assumed the systems to be convergent; the revised proof now holds without this assumption.

2. Opacity and Diagnosability for Transition Systems

In this section, we recall and compare several notions of opacity and diagnosability for labelled transition systems (LTS), and we revisit the complexity of diagnosability for finite LTSs.

2.1. Transition Systems

Given a finite alphabet Σ, we denote by Σ^* the set of finite words over Σ, with ε the empty word, and by Σ^ω the set of infinite words over Σ. For a word $\sigma \in (\Sigma^* \cup \Sigma^\omega)$, $|\sigma|$ is its length in $\omega + 1$,

and for $0 \leq i < |\sigma|$, $\sigma[i]$ denotes its symbol in position i. The (strict) *prefix ordering* is defined for two words $\sigma_1 \in \Sigma^*$ and $\sigma_2 \in (\Sigma^* \cup \Sigma^\omega)$ by $\sigma_1 < \sigma_2$ if there exists a non empty word σ such that $\sigma_2 = \sigma_1\sigma$; we note $Pref(L) \stackrel{\text{def}}{=} \{\hat{\sigma} \in \Sigma^* \mid \exists \sigma \in L : \hat{\sigma} \leq \sigma\}$ for the set of finite prefixes of a language $L \subseteq (\Sigma^* \cup \Sigma^\omega)$; this defines a tree sharing common prefixes.

Labelled Transition System. A *labelled transition system* (LTS) is a tuple $\mathcal{A} = \langle Q, q_0, \Sigma, \Delta \rangle$ where Q is a set of states with $q_0 \in Q$ the initial state, Σ is a finite alphabet, and $\Delta \subseteq Q \times \Sigma \times Q$ is the set of transitions. We note $q \xrightarrow{a} q'$ for $\langle q, a, q' \rangle \in \Delta$; this transition is then said to be *enabled* in q. An LTS is *finitely branching* if, for all $q \in Q$, the set $\{ (a, q') \in \Sigma \times Q \mid (q, a, q') \in \Delta \}$ is finite.

An infinite *execution* is a sequence $\pi = q_0 a_1 q_1 a_2 \cdots \in (Q\Sigma)^\omega$, starting at the initial state and such that $q_i \xrightarrow{a_{i+1}} q_{i+1}$ for all $i \geq 0$. An infinite *trace* is the projection of π on Σ^ω and an infinite *run* is the projection of π on Q^ω; we say that such a run is over the trace $\sigma = a_1 a_2 \cdots \in \Sigma^\omega$, and we write $q_0 \stackrel{\sigma}{\Rightarrow}$ if such a run exists. Finite executions, traces and runs over $\sigma \in \Sigma^*$ are defined analogously, and we write $q \stackrel{\sigma}{\Rightarrow} q'$ if such an execution ends at state q'. A state q is *reachable* if there exists a run $q_0 \stackrel{\sigma}{\Rightarrow} q$ over some finite σ. An LTS $\mathcal{A}$ is *live* (aka deadlock-free) if for any reachable state there exists a transition enabled in that state.

Traces. The *finite trace language* $\mathit{Trace}^*(\mathcal{A}) \subseteq \Sigma^*$ and the *infinite trace language* $\mathit{Trace}^\omega(\mathcal{A}) \subseteq \Sigma^\omega$ of $\mathcal{A}$ are defined by:

$$\mathit{Trace}^*(\mathcal{A}) \stackrel{\text{def}}{=} \{ \sigma \in \Sigma^* \mid \exists q : q_0 \stackrel{\sigma}{\Rightarrow} q \} , \qquad \mathit{Trace}^\omega(\mathcal{A}) \stackrel{\text{def}}{=} \{ \sigma \in \Sigma^\omega \mid q_0 \stackrel{\sigma}{\Rightarrow} \} .$$

Note that for a live LTS $\mathcal{A}$, $Pref(\mathit{Trace}^\omega(\mathcal{A})) = \mathit{Trace}^*(\mathcal{A}) = Pref(\mathit{Trace}^*(\mathcal{A}))$. Also recall that a prefix-closed language $L = Pref(L)$ is *regular* if there exists a finite transition system $\mathcal{A}$ such that $L = \mathit{Trace}^*(\mathcal{A})$.

Observations. In order to formalise diagnosability and opacity, we introduce an observation mask $\mathcal{O}$. Given an LTS $\mathcal{A} = \langle Q, q_0, \Sigma, \Delta \rangle$, $\mathcal{O}$ is a mapping from Σ to $E \cup \{\varepsilon\}$, where E is a finite set of observable *events*: letters of Σ mapped to E correspond to events visible to an external observer, whereas letters mapped to ε remain invisible. We lift $\mathcal{O}$ to a homomorphism and to languages in the usual way.

When σ is an infinite trace, its observation $\mathcal{O}(\sigma)$ can be either finite or infinite; an LTS $\mathcal{A}$ is *convergent* (with respect to $\mathcal{O}$) if we forbid the former, i.e., if there is no infinite sequence of unobservable events from any reachable state (the system is said *divergent* if it is not convergent). Note that convergence is ensured in particular if $\mathcal{O}$ is *non erasing*, i.e., if $\mathcal{O}(\Sigma) \subseteq E$. Liveness and convergence are often assumed in diagnosability and opacity scenarios. The first property simply corresponds to the absence of deadlock, which can be useful in any system model, while the second one is only relevant when partial observation is involved.

For the weakest definitions (see Definition 2.1 and Definition 2.10), both diagnosability and opacity fix a set L of traces (for instance $L = \mathit{Trace}^*(\mathcal{A})$ or $L = \mathit{Trace}^\omega(\mathcal{A})$ for an LTS $\mathcal{A}$) and a particular subset M of L. Writing $\overline{M} \stackrel{\text{def}}{=} L \setminus M$, diagnosability requires $\mathcal{O}(M) \cap \mathcal{O}(\overline{M}) = \emptyset$, while opacity requires $\mathcal{O}(M) \subseteq \mathcal{O}(\overline{M})$. Observation sequences in $\mathcal{O}(M) \cap \mathcal{O}(\overline{M})$ are called 'ambiguous'; for

opacity, all sequences in $\mathcal{O}(M)$ must be ambiguous. The negation of diagnosability can then be seen as a weak form of opacity, as defined in [15], requiring only the existence of ambiguous sequences in $\mathcal{O}(M) \cap \mathcal{O}(\overline{M})$.

2.2. Diagnosability

For diagnosability, we distinguish a special set F of *fault* letters such that $\mathcal{O}(f) = \varepsilon$ for $f \in F$. A finite (resp. infinite) sequence σ is *faulty* if it belongs to $\Sigma^* F \Sigma^*$ (resp. $\Sigma^* F \Sigma^\omega$). Otherwise σ is called *correct*. For an LTS $\mathcal{A}$, we define $\mathit{Faulty}^*(\mathcal{A}) \stackrel{\text{def}}{=} \mathit{Trace}^*(\mathcal{A}) \cap \Sigma^* F \Sigma^*$ for the subset of finite faulty traces and $\mathit{Faulty}^\omega(\mathcal{A}) \stackrel{\text{def}}{=} \mathit{Trace}^\omega(\mathcal{A}) \cap \Sigma^* F \Sigma^\omega$ for the set of infinite faulty traces. Dually, let $\mathit{Correct}^*(\mathcal{A}) \stackrel{\text{def}}{=} \mathit{Trace}^*(\mathcal{A}) \cap (\Sigma \setminus F)^*$ and $\mathit{Correct}^\omega(\mathcal{A}) \stackrel{\text{def}}{=} \mathit{Trace}^\omega(\mathcal{A}) \cap (\Sigma \setminus F)^\omega$ denote the correct traces.

We first recall a language-based notion of diagnosability due to Madalinski and Khomenko [26], that we call *trace-diagnosability* here. Although it is based on languages of infinite words, we shall see that it respects the Finite-Observation Principle for any convergent LTS.

Definition 2.1. (Trace-diagnosability [26])
Given a set of faults F, an LTS $\mathcal{A}$ is *trace-diagnosable* if

$$\mathcal{O}(\mathit{Faulty}^\omega(\mathcal{A})) \cap \mathcal{O}(\mathit{Correct}^\omega(\mathcal{A})) = \emptyset \ .$$

Thus $\mathcal{A}$ is *not trace-diagnosable* if and only if there are two infinite traces σ and ρ in $\mathit{Trace}^\omega(\mathcal{A})$ such that σ is faulty, ρ is correct and $\mathcal{O}(\sigma) = \mathcal{O}(\rho)$.

2.2.1. Characterisation of Trace Diagnosability

In the literature, a standard tool for tackling diagnosability is the *twin-plant* construction [2, 3], illustrated in Figure 3(b) for the LTS in Figure 3(a).

The Twin-Plant Construction. The twin-plant instantiates the classical squaring construction for ambiguity detection [27] for diagnosis purposes. It consists of two 'copies' of an LTS running in parallel and synchronising their observations. The first copy is supposed to eventually commit a fault, while the second copy is prevented from doing so; see Figure 3 for an example.

Formally, let $\mathcal{A} = \langle Q, q_0, \Sigma, \Delta \rangle$ be an LTS, and let $\mathcal{O}$ be an observation mask. We set $\Sigma_u = \mathcal{O}^{-1}(\varepsilon)$ for the subset of Σ of unobservable actions and $\Sigma_o = \Sigma \setminus \Sigma_u$. Then $\mathit{Twin}(\mathcal{A})$ is the LTS $\langle Q', \langle q_0, q_0, 0 \rangle, \Sigma', \Delta' \rangle$ where $Q' \stackrel{\text{def}}{=} Q \times Q \times \{0,1\}$, $\Sigma' \stackrel{\text{def}}{=} (\Sigma_u \times \{\varepsilon\}) \cup (\{\varepsilon\} \times \Sigma_u) \cup \Sigma'_o$ with $\Sigma'_o = \{\langle a, a' \rangle \in \Sigma_o^2 \mid \mathcal{O}(a) = \mathcal{O}(a')\}$, and Δ' contains the following transitions for every $a, a' \in \Sigma$, $b \in \{0,1\}$, transitions $q \xrightarrow{a} q'$, $r \xrightarrow{a'} r'$ in Δ, and state $s \in Q$:

- if $\mathcal{O}(a) = \mathcal{O}(a') \neq \varepsilon$, then $\langle q, r, b \rangle \xrightarrow{\langle a, a' \rangle} \langle q', r', b \rangle$ in Δ';
- if $\mathcal{O}(a) = \varepsilon$, then $\langle q, s, b \rangle \xrightarrow{\langle a, \varepsilon \rangle} \langle q', s, b' \rangle$ in Δ', where $b' = 1$ iff $b = 1$ or $a \in F$;
- if $\mathcal{O}(a') = \varepsilon$ and $a' \notin F$, then $\langle s, r, b \rangle \xrightarrow{\langle \varepsilon, a' \rangle} \langle s, r', b \rangle$.

The mapping $\mathcal{O}$ is extended to Σ' by setting $\mathcal{O}(\langle a, \varepsilon\rangle) = \mathcal{O}(\langle \varepsilon, a'\rangle) \stackrel{\text{def}}{=} \varepsilon$ for all $a, a' \in \Sigma_u$ and $\mathcal{O}(\langle a, a'\rangle) \stackrel{\text{def}}{=} \mathcal{O}(a)$ for all $\langle a, a'\rangle \in \Sigma'_o$.

In the convergent case, trace-diagnosability then reduces to checking whether there exists an infinite run of $Twin(\mathcal{A})$ where the first copy made a faulty transition.

Fact 2.2. If $\mathcal{A}$ is convergent, then $\mathcal{A}$ is not trace-diagnosable if and only if there exists an infinite run in $Twin(\mathcal{A})$ that visits some state $\langle s, t, 1\rangle$ for $s, t \in Q$.

Remark 2.3. (Need for Convergence)
Observe that convergence is necessary for Fact 2.2 to hold, as illustrated by the non convergent LTS $\mathcal{A}$ of Figure 3(a), where $\mathcal{O}(u) = \mathcal{O}(f) = \varepsilon$ and $\mathcal{O}(a) = a$. This LTS is trace-diagnosable because the only non faulty infinite trace u^ω has observation ε while any faulty trace is observed as a^ω. However, $Twin(\mathcal{A})$, represented in Figure 3(b), has an infinite run visiting $\langle q_1, q_0, 1\rangle$.

(a) A trace-diagnosable LTS $\mathcal{A}$. (b) Twin-plant $Twin(\mathcal{A})$.

Figure 3. Infinite run in the twin-plant.

Thus Fact 2.2 is not sufficient to characterise trace diagnosability; the issue is that an infinite run in $Twin(\mathcal{A})$ does not necessarily yield two *infinite* traces σ and ρ of the original system. This is solved by focusing on *twin-fair* runs of $Twin(\mathcal{A})$.

Definition 2.4. (Twin-fairness)
An infinite trace $\langle a_0, a'_0\rangle\langle a_1, a'_1\rangle\langle a_3, a'_3\rangle \cdots$ in $(\Sigma')^\omega$ is *twin-fair* if there exists infinitely many i such that $a_i \neq \varepsilon$ and there exists infinitely many j such that $a'_j \neq \varepsilon$. A *twin-fair run* is a run over a twin-fair trace.

Put differently, $\tau \in (\Sigma')^\omega$ is twin-fair if and only if both $\pi_1(\tau)$ and $\pi_2(\tau)$ are infinite for the homomorphisms $\pi_1, \pi_2 \colon (\Sigma')^* \to \Sigma^*$ defined by $\pi_1(\langle a, a'\rangle) \stackrel{\text{def}}{=} a$ and $\pi_2(\langle a, a'\rangle) \stackrel{\text{def}}{=} a'$ for all $\langle a, a'\rangle \in \Sigma'$. In the LTS of Figure 3(a), the twin-fair runs have to remain in $\langle q_0, q_0, 0\rangle$ indefinitely and the fault cannot be witnessed.

Observe that, if $\mathcal{A}$ is convergent, then all the traces of $Twin(\mathcal{A})$ are twin-fair, hence the following lemma entails Fact 2.2.

Lemma 2.5. (Characterisation of Trace Diagnosability)
An LTS $\mathcal{A}$ is not trace-diagnosable if and only if there exists a twin-fair trace in $Twin(\mathcal{A})$ with an occurrence of $\langle f, \varepsilon\rangle$ for some $f \in F$, if and only if there exists a twin-fair run of $Twin(\mathcal{A})$ visiting some state $\langle s, t, 1\rangle$ for $s, t \in Q$.

Proof:
Let us first introduce an auxiliary notion. For $a \in \Sigma$, denote $a^{\triangleleft} \stackrel{\text{def}}{=} \langle a, \varepsilon \rangle$ and $a^{\triangleright} \stackrel{\text{def}}{=} \langle \varepsilon, a \rangle$; extend this to $\sigma \in \Sigma^* \cup \Sigma^\omega$ by applying the operation to each letter in Σ.

Let σ and ρ be two infinite traces in Σ^ω with $\mathcal{O}(\sigma) = \mathcal{O}(\rho)$. We factor $\sigma = \sigma_0 a_0 \sigma_1 a_1 \cdots$ and $\rho = \rho_0 a'_0 \rho_1 a'_1 \cdots$ so that, for all i, a_i and a'_i are observable symbols in Σ, and σ_i and ρ_i are finite or infinite unobservable sequences in $(\Sigma^* \cup \Sigma^\omega)$: thus $\mathcal{O}(\sigma_i) = \varepsilon$, $\mathcal{O}(\rho_i) = \varepsilon$, $\mathcal{O}(a_i) = \mathcal{O}(a'_i) \neq \varepsilon$, and $|\sigma_i| = \omega$ if and only if $|\rho_i| = \omega$ if and only if $|\mathcal{O}(\sigma)| = i$.

The *fair interleaving along observations* of σ and ρ is the trace $\tau \in (\Sigma')^\omega$ defined by $\tau \stackrel{\text{def}}{=} \tau_0 \langle a_0, a'_0 \rangle \tau_1 \langle a_1, a'_1 \rangle \cdots$, where for all i, $\tau_i \stackrel{\text{def}}{=} {\sigma_i}^{\triangleleft} {\tau_i}^{\triangleright}$ if $|\sigma_i| < \omega$ and $\tau_i \stackrel{\text{def}}{=} \sigma_i[0]^{\triangleleft} \rho_i[0]^{\triangleright} \sigma_i[1]^{\triangleleft} \rho_i[1]^{\triangleright} \cdots$ otherwise. Note that $\pi_1(\tau_i) = \sigma_i$ and $\pi_2(\tau_i) = \rho_i$. By construction, $\mathcal{O}(\tau) = \mathcal{O}(\sigma) = \mathcal{O}(\rho)$, $\pi_1(\tau) = \sigma$ and $\pi_2(\tau) = \rho$, and thus τ is twin-fair.

Returning to the proof of Lemma 2.5, if $\mathcal{A}$ is not trace-diagnosable, then by definition there exist $\sigma \in \mathit{Faulty}^\omega(\mathcal{A})$ and $\rho \in \mathit{Correct}^\omega(\mathcal{A})$ such that $\mathcal{O}(\sigma) = \mathcal{O}(\rho)$. Then their fair interleaving along observations τ belongs to $\mathit{Trace}^\omega(\mathit{Twin}(\mathcal{A}))$. More precisely, one can show that, for all finite prefixes $\hat{\tau}$ of τ, there is a run reaching $\langle s, t, b \rangle$ where $\pi_1(\hat{\tau})$ reaches s, $\pi_2(\hat{\tau})$ reaches t in $\mathcal{A}$, and $b = 1$ if and only if $\langle f, \varepsilon \rangle$ occurs in $\hat{\tau}$ for some $f \in F$. Thus τ is twin-fair and has an occurrence of $\langle f, \varepsilon \rangle$ for some $f \in F$ occurring in σ.

Conversely, if there exists a twin-fair trace τ visiting some state $\langle s, t \rangle$ in $\mathit{Twin}(\mathcal{A})$ and an occurrence of $\langle f, \varepsilon \rangle$ for some $f \in F$, then due to twin-fairness both $\sigma \stackrel{\text{def}}{=} \pi_1(\tau)$ and $\rho \stackrel{\text{def}}{=} \pi_2(\tau)$ are infinite sequences in Σ^ω and f occurs in σ. By definition of $\mathit{Twin}(\mathcal{A})$, σ and ρ are furthermore traces in $\mathit{Trace}^\omega(\mathcal{A})$ with $\mathcal{O}(\sigma) = \mathcal{O}(\rho)$, thus $\mathcal{A}$ is not trace-diagnosable.

Finally, by definition of $\mathit{Twin}(\mathcal{A})$, a trace $\tau \in \mathit{Trace}^\omega(\mathit{Twin}(\mathcal{A}))$ has an occurrence of some $f \in F$ if and only if it has a run that visits some state $\langle s, t, 1 \rangle$ for $s, t \in Q$. □

The characterisation of Lemma 2.5 can be expressed in linear temporal logic (LTL). Define

$$\Sigma_1 \stackrel{\text{def}}{=} \Sigma'_o \cup (\Sigma_u \times \{\varepsilon\}) = \{\langle a, a' \rangle \in \Sigma' \mid a \neq \varepsilon\}$$
$$\Sigma_2 \stackrel{\text{def}}{=} \Sigma'_o \cup (\{\varepsilon\} \times \Sigma_u) = \{\langle a, a' \rangle \in \Sigma' \mid a' \neq \varepsilon\}$$

and assume Σ' serves as set of atomic propositions (thus this is an 'action-based' LTL). Then $\mathcal{A}$ is trace-diagnosable if and only if $\mathit{Twin}(\mathcal{A})$ has a trace in $\mathit{Trace}^\omega(\mathit{Twin}(\mathcal{A}))$ satisfying

$$\left(\Diamond \bigvee_{f \in F} \langle f, \varepsilon \rangle \right) \wedge \left(\Box\Diamond \bigvee_{\langle a, a' \rangle \in \Sigma_1} \langle a, a' \rangle \right) \wedge \left(\Box\Diamond \bigvee_{\langle a, a' \rangle \in \Sigma_2} \langle a, a' \rangle \right) . \qquad (\varphi_{\text{diag}})$$

This in turn reduces to an emptiness check for the intersection of $\mathit{Trace}^\omega(\mathit{Twin}(\mathcal{A}))$ with the language of the Büchi automaton $\mathcal{B}_{\text{diag}}$ presented in Figure 4, where the doubly circled state must be reached infinitely often. Note that the number of states of $\mathcal{B}_{\text{diag}}$ does *not* depend on Σ or F. We will use φ_{diag} and $\mathcal{B}_{\text{diag}}$ in our upper bound proofs in sections 2.2.3 and 5.2.

2.2.2. Comparison of the Different Definitions of Diagnosability

Besides Definition 2.1, various other notions of diagnosability have been studied and discussed in [1, 21]. The strongest is *uniform diagnosability*: there exists a natural number K (that may depend on

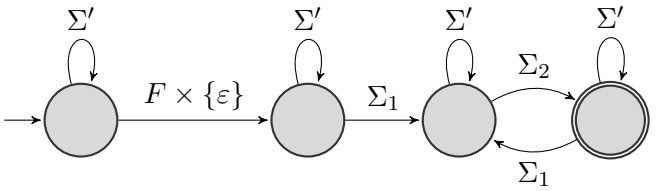
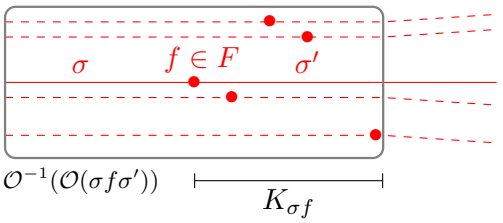


Figure 4. A Büchi automaton $\mathcal{B}_{\text{diag}}$ for trace-diagnosability.

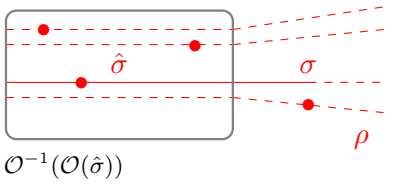
the LTS $\mathcal{A}$) such that the occurrence of a faulty transition is detected after at most K steps, i.e., for any faulty trace $\sigma f \in \mathit{Faulty}^*(\mathcal{A})$ and any suffix σ' with $|\sigma'| \geq K$ and $\sigma f \sigma' \in \mathit{Trace}^*(\mathcal{A})$, any trace $\rho \in \mathit{Trace}^*(\mathcal{A})$ such that $\mathcal{O}(\rho) = \mathcal{O}(\sigma f \sigma')$ is also faulty.

We use the term *dynamic diagnosability* for a less stringent notion studied in [21], which requires detection after a non-uniform finite number of steps. A fault f occurring after a prefix σ must be detectable after $K_{\sigma f}$ steps, where $K_{\sigma f}$ may depend on σf, see Figure 5(a). Dynamic diagnosability and uniform diagnosability coincide if $\mathit{Trace}^*(\mathcal{A})$ is regular, but differ in general [21, Rem. 5.5].

As we want to consider diagnosability in conjunction with fairness constraints, we shall need yet another notion of diagnosability able to take infinite runs into account while demanding that the observer diagnose the occurrence of a fault in finite time. We say that an LTS $\mathcal{A}$ is *finitely diagnosable* if, for all $\sigma \in \mathit{Faulty}^\omega(\mathcal{A})$, there exists a finite prefix $\hat{\sigma} < \sigma$ such that every $\rho \in \mathit{Trace}^\omega(\mathcal{A})$ with $\mathcal{O}(\hat{\sigma}) \leq \mathcal{O}(\rho)$ is also faulty. We argue that this notion, illustrated in Figure 5(b), captures the Finite-Observation Principle. The restriction of finite diagnosability to weakly fair runs (recalled later in Definition 3.3) is exactly the definition used in [4].



(a) Illustration of dynamic diagnosability. Each line represents a trace in $\mathcal{A}$ whose observation starts with $\mathcal{O}(\sigma f \sigma')$, and bullets indicate faults. Each trace must contain a fault at most $K_{\sigma f}$ steps after the earliest possible fault occurrence.



(b) Illustration of finite diagnosability. A faulty trace σ must possess a finite prefix $\hat{\sigma}$ such that all infinite traces whose observation starts with $\mathcal{O}(\hat{\sigma})$ are guaranteed to contain a fault, be it before completing the observation of $\mathcal{O}(\hat{\sigma})$ or at some point in the future.

Figure 5. Schematic illustration of dynamic diagnosability and finite diagnosability.

We are now in a position to establish the links between these various notions in the absence of fairness constraints. For completeness, we include the above-mentioned result about the relation between uniform and dynamic diagnosability.

Lemma 2.6. (Comparison of Diagnosability Notions)
Let $\mathcal{A}$ be an LTS. Then we have the implications $1 \Rightarrow 2 \Rightarrow 3 \Rightarrow 4$ where:

1. $\mathcal{A}$ is uniformly diagnosable;
2. $\mathcal{A}$ is dynamically diagnosable;
3. $\mathcal{A}$ is finitely diagnosable;
4. $\mathcal{A}$ is trace-diagnosable.

Moreover, 1 and 2 are equivalent if $\mathit{Trace}^*(\mathcal{A})$ is regular [21, Prop. 5.3], and 2, 3, and 4 are equivalent if $\mathcal{A}$ is finitely branching and convergent.

Proof:
$2 \Rightarrow 3$. By contraposition, suppose that $\mathcal{A}$ is not finitely diagnosable: there exists $\sigma \in \mathit{Faulty}^\omega(\mathcal{A})$ such that for any prefix $\hat{\sigma}$ of σ, there exists $\rho \in \mathit{Correct}^\omega(\mathcal{A})$ such that $\mathcal{O}(\hat{\sigma}) \leq \mathcal{O}(\rho)$. Then for $K > 0$, we consider the sequence of prefixes $\hat{\sigma}_K = \sigma_0 f \sigma_K$ of σ where f is an occurrence of a fault in σ and $|\hat{\sigma}_K| = K$. From the hypothesis, for any K, there exists $\rho_K \in \mathit{Correct}^\omega(\mathcal{A})$ such that $\mathcal{O}(\hat{\sigma}_K) \leq \mathcal{O}(\rho_K)$. Considering now for any K, the finite prefix $\hat{\rho}_K \in \mathit{Correct}^*(\mathcal{A})$ of ρ_K such $\mathcal{O}(\hat{\sigma}_K) = \mathcal{O}(\hat{\rho}_K)$, we obtain that $\mathcal{A}$ is not dynamically diagnosable.

$3 \Rightarrow 4$. By contraposition, suppose that $\mathcal{A}$ is not trace-diagnosable: then there exist $\sigma, \rho \in \mathit{Trace}^\omega(\mathcal{A})$ such that $\mathcal{O}(\sigma) = \mathcal{O}(\rho)$, σ is faulty, and ρ is correct. Then, for any prefix $\hat{\sigma}$ of σ, we have $\mathcal{O}(\hat{\sigma}) \leq \mathcal{O}(\rho)$, and since ρ is correct, $\mathcal{A}$ is not finitely diagnosable.

$3 \Rightarrow 2$ if $\mathcal{A}$ is finitely branching and convergent. By contraposition, assume that $\mathcal{A}$ is not dynamically diagnosable. Then there exists a trace $\sigma_0 f$ such that for all $n > 0$ there exists a pair σ_n, ρ_n such that $\sigma_0 f \sigma_n \in \mathit{Faulty}^*(\mathcal{A})$, $\rho_n \in \mathit{Correct}^*(\mathcal{A})$, $|\sigma_n| \geq n$, and $\mathcal{O}(\sigma_0 f \sigma_n) = \mathcal{O}(\rho_n)$. For each n, fix some (finite) execution π_n in $\mathit{Twin}(\mathcal{A})$ over the pair $\langle \sigma_0 f \sigma_n, \rho_n \rangle$. Consider the tree

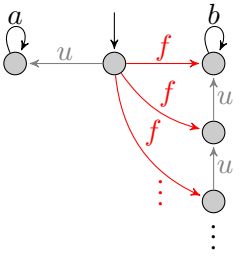
$$\mathcal{T} \stackrel{\text{def}}{=} \mathit{Pref}(\{\, \pi_n \mid n \in \mathbb{N} \,\}).$$

This tree is infinite, and of finite branching degree due to the assumptions on $\mathcal{A}$ and the construction of $\mathit{Twin}(\mathcal{A})$. By Kőnig's Lemma, $\mathcal{T}$ contains an infinite branch π. By construction, π is an infinite execution of $\mathit{Twin}(\mathcal{A})$. Recall that traces in $\mathit{Twin}(\mathcal{A})$ are pairs of traces in $\mathcal{A}$. Since $\mathcal{A}$ is convergent, we can extract from π a twin-fair run over a pair $\langle \sigma, \rho \rangle$. Since σ is infinite and every branch of the tree starts with the fixed prefix $\sigma_0 f$ in its left component, $\sigma_0 f$ must also be a prefix of σ. Thus, according to Lemma 2.5, $\mathcal{A}$ is not trace-diagnosable. By $3 \Rightarrow 4$, $\mathcal{A}$ is also not finitely diagnosable.

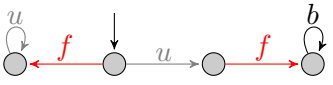
$4 \Rightarrow 3$ if $\mathcal{A}$ is finitely branching and convergent.

Let us call $\sigma \in \mathit{Trace}^\omega(\mathcal{A})$ *finitely indistinguishable* from $\mathit{Correct}^\omega(\mathcal{A})$ if for every finite prefix $\hat{\sigma}_n < \sigma$ of length n, we can find a correct $\rho_n \in \mathit{Correct}^\omega(\mathcal{A})$ with $\mathcal{O}(\hat{\sigma}_n) \leq \mathcal{O}(\rho_n)$. We first prove the following claim (which will also be used in the proof of Lemma 3.9):

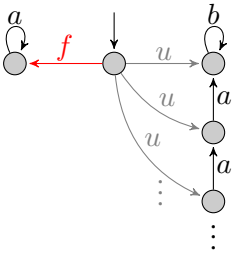
Claim 2.7. Let $\mathcal{A}$ be a finitely branching and convergent LTS. If $\sigma \in \mathit{Trace}^\omega(\mathcal{A})$ is finitely indistinguishable from $\mathit{Correct}^\omega(\mathcal{A})$, then there exists $\rho \in \mathit{Correct}^\omega(\mathcal{A})$ with $\mathcal{O}(\sigma) = \mathcal{O}(\rho)$.



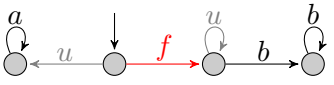
(a) Infinitely branching example for $3 \not\Rightarrow 2$.



(b) Divergent example for $3 \not\Rightarrow 2$.



(c) Infinitely branching example for $4 \not\Rightarrow 3$.



(d) Divergent example for $4 \not\Rightarrow 3$.

Figure 6. Counter-examples for Lemma 2.6, with $\mathcal{O}(u) = \mathcal{O}(f) = \varepsilon$, $\mathcal{O}(a) = a$, and $\mathcal{O}(b) = b$. The LTSs on the left are finitely diagnosable but not dynamically diagnosable while those on the right are trace-diagnosable but not finitely diagnosable.

To prove Claim 2.7, for all n let $\hat{\rho}_n$ be a finite prefix of ρ_n such that $\mathcal{O}(\hat{\rho}_n) = \mathcal{O}(\hat{\sigma}_n)$. Let $\hat{\pi}_n$ be the corresponding finite execution and define the tree $\mathcal{T} \stackrel{\text{def}}{=} \mathit{Pref}(\{\, \hat{\pi}_n \mid n \in \mathbb{N} \,\})$. Since $\mathcal{A}$ is finitely branching, $\mathcal{T}$ has finite degree. Since $\mathcal{A}$ is convergent, $\{\mathcal{O}(\hat{\sigma}_n) \mid n \in \mathbb{N}\}$ is infinite; hence, since $\mathcal{O}(\hat{\sigma}_n) = \mathcal{O}(\hat{\rho}_n)$ for every n, $\mathcal{T}$ is also infinite. By Kőnig's Lemma, $\mathcal{T}$ contains an infinite execution $\pi = q_0 a_1 q_1 a_2 \cdots$ such that every prefix of π is a prefix of $\hat{\pi}_n$ for some n. Let $\rho \stackrel{\text{def}}{=} a_1 a_2 \cdots$ be the projection of π on Σ^ω. We have $\rho \in \mathit{Correct}^\omega(\mathcal{A})$ as, by construction, $q_i \xrightarrow{a_{i+1}} q_{i+1}$ is a transition of $\mathcal{A}$ and $a_{i+1} \notin F$ for all $i \geq 0$. Finally, $\mathcal{O}(\rho) = \mathcal{O}(\sigma)$, as otherwise there would exist a prefix of some $\hat{\rho}_n$ whose observation would not be a prefix of $\mathcal{O}(\sigma)$; but this would be in contradiction with $\mathcal{O}(\hat{\rho}_n) = \mathcal{O}(\hat{\sigma}_n) < \mathcal{O}(\sigma)$. This concludes the proof of Claim 2.7.

For $4 \Rightarrow 3$, assume now by contraposition that $\mathcal{A}$ is not finitely diagnosable. Then, there exists $\sigma \in \mathit{Faulty}^\omega(\mathcal{A})$ such that σ is finitely indistinguishable from $\mathit{Correct}^\omega(\mathcal{A})$. By Claim 2.7, there is a $\rho \in \mathit{Correct}^\omega(\mathcal{A})$ with $\mathcal{O}(\sigma) = \mathcal{O}(\rho)$, hence $\mathcal{A}$ is not trace-diagnosable. □

Remark 2.8. (Counter-examples)
Figures 6(a) and 6(b) show that $\mathcal{A}$ must be both finitely branching and convergent for the equivalence $3 \Leftrightarrow 2$ to hold in Lemma 2.6. The LTS in Figure 6(a) is trace-diagnosable since $\mathcal{O}(ua^\omega) \neq \mathcal{O}(fu^n b^\omega)$ for all n. It is also finitely diagnosable because for any infinite faulty $\sigma = fu^n b^\omega$, the finite prefix $\hat{\sigma} = fu^n b$ with observation b can only be extended by observations of faulty infinite traces. It is however not dynamically diagnosable because the faulty prefix f may require an arbitrarily long finite delay K before being diagnosed by $u^K b$. In contrast, the LTS in Figure 6(b) is finitely branching but divergent; it is finitely diagnosable for the trivial reason that there is no infinite correct run. (We remark that there exist other examples without this particular property.) This LTS is not dynamically diagnosable because for the prefix f we have $\mathcal{O}(fu^K) = \mathcal{O}(u)$ for all K.

Figures 6(c) and 6(d) show that $\mathcal{A}$ must also be convergent and finitely branching for the equivalence $4 \Leftrightarrow 3$ to hold in Lemma 2.6. The LTS in Figure 6(c) is trace-diagnosable because $\mathcal{O}(fa^\omega) \neq$

$\mathcal{O}(ua^nb^\omega)$ for all n, but not finitely diagnosable: For the single infinite faulty trace $\sigma = fa^\omega$, any observation a^n of a finite prefix $\hat{\sigma}_n = fa^n$ can be extended by a^nb^ω which corresponds to the observation of a correct infinite trace. The LTS in Figure 6(d) is trace-diagnosable, as $\mathcal{O}(fu^\omega) \neq \mathcal{O}(ua^\omega) \neq \mathcal{O}(fu^nb^\omega)$ for all n, but is not finitely diagnosable because all the finite prefixes of the faulty fu^ω have the same observation $\varepsilon < \mathcal{O}(ua^\omega)$.

2.2.3. Complexity in Finite LTSs

In the case of finite-state LTSs, and assuming an explicit representation with $|\mathcal{A}| \stackrel{\text{def}}{=} |\Delta|+|Q|+|\Sigma|$, it is easy to show, using the twin-plant construction, that checking trace-diagnosability in convergent systems takes quadratic time w.r.t. $|\mathcal{A}|$ [2, 3]. We can strengthen this to show NL-completeness; note that under the conditions named in the following lower bound, all four notions of diagnosability previously discussed in Section 2.2.2 coincide.

Proposition 2.9. Verifying trace-diagnosability for finite LTSs is in NL; it is NL-hard already for finite, live and convergent LTSs.

Proof:
The following problem (STCON) is known to be NL-complete: given a directed graph $G = \langle V, E\rangle$ and two nodes $s, t \in V$, decide whether there exists a path from s to t. We first show that STCON is logspace-reducible to trace-diagnosability. Given $G = \langle V, E\rangle$ and $s, t \in V$, let us define the LTS $\mathcal{A}_G \stackrel{\text{def}}{=} \langle V \uplus \{v_f\}, s, \{a, f, u\}, \Delta_G\rangle$ with

$$\Delta_G \stackrel{\text{def}}{=} \{\, \langle q, a, q'\rangle, \langle q, a, v_f\rangle \mid \langle q, q'\rangle \in E \,\} \cup \{\langle t, f, v_f\rangle, \langle t, u, v_f\rangle, \langle v_f, a, v_f\rangle\}\ .$$

We set $F \stackrel{\text{def}}{=} \{f\}$, $\mathcal{O}(a) \stackrel{\text{def}}{=} a$, and $\mathcal{O}(f) \stackrel{\text{def}}{=} \mathcal{O}(u) \stackrel{\text{def}}{=} \varepsilon$. Thus, $\mathcal{A}_G$ is live and convergent, and is not trace-diagnosable if and only if t can be reached from s, which (with coNL = NL) proves NL-hardness.

For NL-membership, let $\mathcal{A}$ be an LTS with a finite number of states. By Lemma 2.5, trace-diagnosability reduces to checking the emptiness of the Büchi automaton obtained by synchronising on Σ' the twin-plant $Twin(\mathcal{A})$ with $\mathcal{B}_{\text{diag}}$ displayed in Figure 4. Note that $\mathcal{B}_{\text{diag}}$ has a constant number of states, independent of $\mathcal{A}$, and that an emptiness check can be performed by a 'lasso' search in the synchronous product, either in linear time w.r.t. the product (hence with quadratic time complexity w.r.t. $|\mathcal{A}|$) or in NL [28, Theorem 2.2]. To achieve the latter, $Twin(\mathcal{A})$ and its synchronisation with $\mathcal{B}_{\text{diag}}$ are not constructed explicitly but explored on the fly, i.e. the machine memorises the current triple $\langle s, t, b\rangle$ and state of $\mathcal{B}_{\text{diag}}$ and non-deterministically chooses a successor state according to the construction rules for $Twin(\mathcal{A})$ and $\mathcal{B}_{\text{diag}}$. □

2.3. Opacity

The classical notion of opacity, as defined in [11], deals with finite traces only. For our purpose, we fix a subset S of Σ containing special *secret* letters such that $\mathcal{O}(s) = \varepsilon$ for all $s \in S$. We consider as secret any sequence containing some $s \in S$, hence the set of finite secrets in an LTS $\mathcal{A}$ is $\mathit{Sec}^*(\mathcal{A}) \stackrel{\text{def}}{=} \mathit{Trace}^*(\mathcal{A}) \cap \Sigma^* S \Sigma^*$, while the set of infinite secrets is $\mathit{Sec}^\omega(\mathcal{A}) \stackrel{\text{def}}{=} \mathit{Trace}^\omega(\mathcal{A}) \cap \Sigma^* S \Sigma^\omega$; dually, the set of finite non-secret traces is $\mathit{Pub}^*(\mathcal{A}) \stackrel{\text{def}}{=} \mathit{Trace}^*(\mathcal{A}) \cap (\Sigma \setminus S)^*$ and the set of infinite non-secret ones is $\mathit{Pub}^\omega(\mathcal{A}) \stackrel{\text{def}}{=} \mathit{Trace}^\omega(\mathcal{A}) \cap (\Sigma \setminus S)^\omega$.

Definition 2.10. (Opacity [11])
The secret S in an LTS $\mathcal{A}$ is *opaque* for observation mask $\mathcal{O}$ if

$$\mathcal{O}(\mathit{Sec}^*(\mathcal{A})) \subseteq \mathcal{O}(\mathit{Pub}^*(\mathcal{A})) \ .$$

Note that *Strong Nondeterministic Non-Interference (SNNI)* as studied for example in [16, 17] is a particular case of opacity where $\mathcal{O}$ is restricted to be the projection from Σ^* onto $(\Sigma \setminus S)^*$, i.e. the homomorphism defined by $\mathcal{O}(a) \stackrel{\text{def}}{=} a$ if $a \in \Sigma \setminus S$ and $\mathcal{O}(a) \stackrel{\text{def}}{=} \varepsilon$ if $a \in S$.

The problem of checking opacity was proven PSPACE-complete for finite LTSs [12] for a state-based variant, and this is easily seen to hold for Definition 2.10 as well.

Finite Opacity. As with diagnosability, we shall need a notion of opacity able to consider infinite runs, which we will then refine in Definition 3.5 for weakly fair opacity: we say that the secret in an LTS $\mathcal{A}$ is *finitely opaque* if, for all $\hat{\sigma} \in \mathit{Sec}^*(\mathcal{A})$, there exists an infinite non-secret trace $\rho \in \mathit{Pub}^\omega(\mathcal{A})$ such that $\mathcal{O}(\hat{\sigma}) \leq \mathcal{O}(\rho)$.

Lemma 2.11. (Comparison of Opacity Notions)
Let $\mathcal{A}$ be a convergent live finitely-branching LTS. Then the secret in $\mathcal{A}$ is opaque if and only if it is finitely opaque.

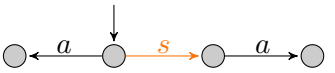
Proof:
Let us first show that finite opacity implies opacity. Assume $\mathcal{A}$ is finitely opaque, and select any trace $\hat{\sigma} \in \mathit{Sec}^*(\mathcal{A})$. Then there exists an infinite non-secret trace $\rho \in \mathit{Pub}^\omega(\mathcal{A})$ such that $\mathcal{O}(\hat{\sigma}) \leq \mathcal{O}(\rho)$, hence ρ possesses a suitable non-secret finite prefix $\hat{\rho}$ with $\mathcal{O}(\hat{\sigma}) = \mathcal{O}(\hat{\rho})$. Thus $\mathcal{A}$ is opaque.

Assume for the converse that $\mathcal{A}$ is opaque and pick any trace $\hat{\sigma} \in \mathit{Sec}^*(\mathcal{A})$. Since $\mathcal{A}$ is live, there exists an infinite $\sigma \in \mathit{Sec}^\omega(\mathcal{A})$ with $\hat{\sigma} < \sigma$. For every finite extension $\hat{\sigma}_n$ of length $|\hat{\sigma}| + n$ with $\hat{\sigma} \leq \hat{\sigma}_n < \sigma$, there exists a non-secret trace $\hat{\rho}_n \in \mathit{Pub}^*(\mathcal{A})$ with $\mathcal{O}(\hat{\sigma}_n) = \mathcal{O}(\hat{\rho}_n)$ and a corresponding execution $\hat{\pi}_n$. Consider the tree

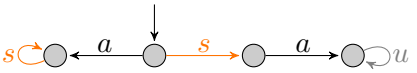
$$\mathcal{T} \stackrel{\text{def}}{=} \mathit{Pref}(\{\, \hat{\pi}_n \mid n \in \mathbb{N} \,\})$$

formed by all these non-secret executions. Since $\mathcal{A}$ is finitely branching, $\mathcal{T}$ has finite degree. Since $\mathcal{A}$ is convergent, $\{\mathcal{O}(\hat{\sigma}_n) \mid n \in \mathbb{N}\}$ is infinite, hence since $|\hat{\rho}_n| \geq |\mathcal{O}(\hat{\sigma}_n)|$, $\mathcal{T}$ is also infinite. By Kőnig's Lemma, $\mathcal{T}$ contains an infinite execution $\pi \stackrel{\text{def}}{=} q_0 a_1 q_1 a_2 \cdots$ with trace $\rho \stackrel{\text{def}}{=} a_1 a_2 \cdots \in \Sigma^\omega$. By construction, $\rho \in \mathit{Pub}^\omega(\mathcal{A})$ since $q_i \xrightarrow{a_{i+1}} q_{i+1}$ and $a_{i+1} \notin S$ for all $i \geq 0$. Also, since $\hat{\sigma} \leq \hat{\sigma}_n$ and $\mathcal{O}(\hat{\sigma}_n) = \mathcal{O}(\hat{\rho}_n)$ imply $\mathcal{O}(\hat{\sigma}) \leq \mathcal{O}(\hat{\rho}_n)$ for all $n \geq 0$, the observation of any infinite branch of $\mathcal{T}$ must have $\mathcal{O}(\hat{\sigma})$ as prefix, so $\mathcal{O}(\hat{\sigma}) \leq \mathcal{O}(\rho)$ holds. Therefore $\mathcal{A}$ is finitely opaque. □

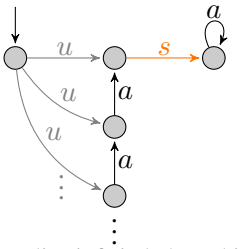
Remark 2.12. (Counter-examples)
Figure 7 shows that $\mathcal{A}$ must be convergent, live, and finitely branching for the equivalence between opacity and finite opacity to hold in Lemma 2.11. In both Figure 7(a) and Figure 7(b) the system is opaque since $\mathcal{O}(\mathit{Sec}^*(\mathcal{A})) = \{\varepsilon, a\} = \mathcal{O}(\mathit{Pub}^*(\mathcal{A}))$; in Figure 7(c), it is opaque since $\mathcal{O}(\mathit{Sec}^*(\mathcal{A})) = \{a^n \mid n \in \mathbb{N}\} = \mathcal{O}(\mathit{Pub}^*(\mathcal{A}))$. However, in all three cases, the system is not finitely opaque because there exist finite secret traces in $\mathit{Sec}^*(\mathcal{A})$ (e.g. s in Figures 7(a) and 7(b) and us in Figure 7(c)), but no infinite non-secret trace: $\mathit{Pub}^\omega(\mathcal{A}) = \emptyset$.



(a) Convergent non live finitely branching example.



(b) Divergent live finitely branching example.



(c) Convergent live infinitely branching example.

Figure 7. Counter-examples for Lemma 2.11, with $\mathcal{O}(u) = \mathcal{O}(s) = \varepsilon$ and $\mathcal{O}(a) = a$.

3. Opacity and Diagnosability for Petri Nets

After some reminders on Petri nets, we devote this section to the definitions of weakly fair Petri nets in Section 3.2 and of suitable variants of diagnosability and opacity in Section 3.3. We finally consider the case of weakly fair diagnosability when no faults are fair in Section 3.4.

3.1. Petri Nets

Syntax. A *Petri Net* (PN) is a tuple $\mathcal{N} = \langle P, T, w, \boldsymbol{m}_0 \rangle$ where P and T are finite sets of *places* and *transitions* respectively, $w\colon (P \times T) \cup (T \times P) \to \mathbb{N}$ is the *flow mapping*, and $\boldsymbol{m}_0 \in \mathbb{N}^P$ is the *initial marking*.

A *marking* is a mapping $\boldsymbol{m} \in \mathbb{N}^P$. As usual, in figures, transitions are represented as rectangles and places as circles. If $\boldsymbol{m}(p) \geq 1$, the corresponding number of black tokens are drawn in p. For a transition t, we denote its *preset* by ${}^\bullet t \stackrel{\text{def}}{=} \{p \in P \mid w(p,t) > 0\}$ and its *postset* by $t^\bullet \stackrel{\text{def}}{=} \{p \in P \mid w(t,p) > 0\}$.

Semantics. The operational semantics of a PN $\mathcal{N} = \langle P, T, w, \boldsymbol{m}_0 \rangle$ is an LTS $\mathcal{A}_\mathcal{N} = \langle \mathbb{N}^P, \boldsymbol{m}_0, T, \Delta \rangle$, whose states are the markings of $\mathcal{N}$, and whose transitions are labelled by T, where $\langle \boldsymbol{m}, t, \boldsymbol{m}' \rangle \in \Delta$ if and only if for each $p \in P$ we have $\boldsymbol{m}(p) \geq w(p,t)$, and $\boldsymbol{m}'(p) = \boldsymbol{m}(p) - w(p,t) + w(t,p)$ for all $p \in P$. Note that $\mathcal{A}_\mathcal{N}$ is finitely branching. It is also 'deterministic' as no two different runs can produce the same trace.

We shall abuse notations and write '*Trace*$^*(\mathcal{N})$' (resp. '*Trace*$^\omega(\mathcal{N})$', etc.) instead of '*Trace*$^*(\mathcal{A}_\mathcal{N})$' (resp. '*Trace*$^\omega(\mathcal{A}_\mathcal{N})$', etc.).

Note that adding an observation mask to a Petri net $\mathcal{N}$ results in what is usually called a 'labelled Petri net'. Then the various notions of diagnosability with respect to a subset $F \subseteq T$ of faulty transitions correspond to the same notion of diagnosability of the transition system $\mathcal{A}_\mathcal{N}$. As mentioned in the introduction, this notion declares the net from Figure 2(a) not to be trace-diagnosable. Similarly, a Petri net $\mathcal{N}$ is opaque in the sense of Definition 2.10 with respect to a subset $S \subseteq T$ of secret transitions if $\mathcal{A}_\mathcal{N}$ is opaque.

Safe Petri Nets. A Petri net $\mathcal{N}$ is *safe* if the reachable states form a subset of $\{0,1\}^P$; as a result $\mathcal{A}_\mathcal{N}$ is finite, and *Trace*$^*(\mathcal{N})$ is regular. Note however that safe Petri nets are *implicit* descriptions of $\mathcal{A}_\mathcal{N}$: the latter can be of (at most) exponential size in terms of $|\mathcal{N}|$. This immediately entails that, in safe Petri nets, trace-diagnosability is in PSPACE by Proposition 2.9 and opacity in EXPSPACE by [12]; we shall generalise and refine these upper bounds in Section 5 to take weak fairness into account.

3.2. Weak Fairness

We shall employ the following generalisation of weak fairness as defined in [8, 4]:

Definition 3.1. (Weak Fairness)
A *Petri net with weak fairness* (WF-PN) is a tuple $\mathcal{W} = \langle \mathcal{N}, W \rangle$, where $\mathcal{N}$ is a Petri net and $W \subseteq T$ a set of transitions called *weakly fair*.

Let $\sigma = (t_i)_{i\geq 1} \in T^\omega$ be an infinite trace, and $(\boldsymbol{m}_i)_{i\geq 0}$ the (uniquely determined) infinite run of $\mathcal{A}_\mathcal{N}$ over σ. Then σ is *weakly fair* if for every $t \in W$,

WF.1 there are infinitely many i with $t_i = t$, **or**

WF.2 there are infinitely many i where t_i *conflicts with* t *with respect to* $\boldsymbol{m}_{i-1}$, i.e. there exists $p \in P$ s.t. $\boldsymbol{m}_{i-1}(p) < w(p,t_i) + w(p,t)$.

Note that (WF.2) also covers the case where t is simply disabled. Informally, in a weakly fair sequence σ, each weakly fair transition t that is enabled either fires eventually, or some other transition that competes for a preset place with t fires. As shown by Jančar [10], it is decidable whether a WF-PN has at least one weakly fair trace.[1]

In drawings, we shall denote weakly fair transitions by the annotation 'WF' and a bang. For instance, in the net from Figure 2(b), s_2 is a weakly fair transition, and $u_1 t_1^\omega$ is not weakly fair since s_2 is continuously enabled and never fires. (In this case, no other transition conflicts with s_2.) We shall use '*WF*' subscripts to denote the restriction of a set of infinite traces to weakly fair ones, as in '*Trace*$^\omega_{WF}(\mathcal{N})$' or '*Faulty*$^\omega_{WF}(\mathcal{N})$'.

When the underlying Petri net $\mathcal{N}$ is safe, we show that Definition 3.1 coincides with the definition employed in [9, 8, 4].

Proposition 3.2. (Weak Fairness in Safe PNs)
Let $\mathcal{W} = \langle \mathcal{N}, W \rangle$ be a safe WF-PN. An infinite trace $\sigma = (t_i)_{i\geq 1}$ with run $(\boldsymbol{m}_i)_{i\geq 0}$ is *weakly fair* if and only if, for every $i > 0$ and every $t \in W$ enabled in $\boldsymbol{m}_{i-1}$, there exists some $j \geq i$ such that ${}^\bullet t \cap {}^\bullet t_j \neq \emptyset$.

Proof:
Let us first prove the 'if' part and assume for this that for every $i > 0$ and every $t \in W$ enabled in $\boldsymbol{m}_{i-1}$, there exists some $j \geq i$ such that ${}^\bullet t \cap {}^\bullet t_j \neq \emptyset$. If t fires infinitely often in σ, then (WF.1) holds and we are done. Otherwise $t \in W$ fires only finitely often in σ. If it is enabled only finitely often,

[1] In Jančar's definition, (WF.2) uses the simpler condition $\boldsymbol{m}_{i-1}(p) < w(p,t)$. We could easily adapt our treatment of weak fairness to work with that definition, but we preferred to remain compatible with [8, 4].

then for infinitely many i, there exists $p \in P$ s.t. $\boldsymbol{m}_{i-1}(p) < w(p,t)$ and therefore t_i conflicts with t in $\boldsymbol{m}_{i-1}$. If it is enabled infinitely often, then for infinitely many i, there exists $p \in {}^\bullet t \cap {}^\bullet t_i$, and since $\mathcal{N}$ is safe $\boldsymbol{m}_{i-1}(p) - w(p,t_i) = 0 < w(p,t)$.

Conversely, by contraposition for the 'only if' part of the statement, assume that there exists $i > 0$ and $t \in W$ enabled in $\boldsymbol{m}_{i-1}$, such that $\forall j \geq i$, ${}^\bullet t_j \cap {}^\bullet t = \emptyset$, and let us show that the run is not weakly fair:

- For all $j \geq i$, $t_j \neq t$ hence (WF.1) does not hold.
- By induction on $j \geq i$, since ${}^\bullet t \cap {}^\bullet t_j = \emptyset$ and t was enabled in $\boldsymbol{m}_{i-1}$, t remains enabled in $\boldsymbol{m}_{j-1}$. Hence, for all $j \geq i$ and all $p \in P$,
  - either $p \in P \setminus ({}^\bullet t \cup {}^\bullet t_j)$ and $\boldsymbol{m}_{j-1}(p) \geq 0 = w(p,t) + w(p,t_j)$,
  - or $p \in {}^\bullet t$ but $p \notin {}^\bullet t_j$, and $\boldsymbol{m}_{j-1}(p) \geq w(p,t) + 0 = w(p,t) + w(p,t_j)$ since t is enabled,
  - or $p \in {}^\bullet t_j$ but $p \notin {}^\bullet t$, and $\boldsymbol{m}_{j-1}(p) \geq 0 + w(p,t_j) = w(p,t) + w(p,t_j)$ since t_j is fired.

  Thus (WF.2) does not hold. □

3.3. Diagnosability and Opacity with Weak Fairness

In the context of Petri nets with weak fairness, the definitions of both notions must take into account the set of weakly fair transitions while maintaining the Finite-Observation Principle.

Weakly Fair Diagnosability. We restrict finite diagnosability to the set of weakly fair runs, as is done in [4], but with a generalised notion of weak fairness.

Definition 3.3. (Weakly Fair Diagnosability)
A WF-PN $\mathcal{W} = \langle \mathcal{N}, W \rangle$ is said to be *WF-diagnosable* if every infinite, weakly fair, faulty trace $\sigma \in \mathit{Faulty}^\omega_{WF}(\mathcal{N})$ has a finite prefix $\hat{\sigma}$ such that every infinite weakly fair trace $\rho \in \mathit{Trace}^\omega_{WF}(\mathcal{N})$ satisfying $\mathcal{O}(\hat{\sigma}) \leq \mathcal{O}(\rho)$ is faulty.

Consider again the net from Figure 2(a) and assume that transition a is WF. Then this net is WF-diagnosable since a weakly fair trace that contains f also eventually contains a, and a is only possible after f. Note that, as shown in [4], this definition is not equivalent to simply restricting trace-diagnosability according to Definition 2.1 to weakly fair traces. The precise relation of WF-diagnosability with other notions was not examined in [4]; however, by Lemma 2.6, we obtain:

Lemma 3.4. Let $\mathcal{W} = \langle \mathcal{N}, W \rangle$ be a convergent WF-PN such that $W = \emptyset$. Then $\mathcal{W}$ is WF-diagnosable if and only if $\mathcal{N}$ is trace-diagnosable.

Proof:
Note that, since $W = \emptyset$, every infinite run of $\mathcal{W}$ is weakly fair. Thus WF-diagnosability 'degrades' to finite diagnosability, which is equivalent to trace-diagnosability by Lemma 2.6 since Petri nets are finitely branching and we assumed $\mathcal{W}$ to be convergent. □

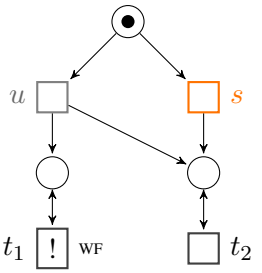


Figure 8. WF-PN $\mathcal{W}_2$ with t_1 weakly fair, s secret and u unobservable.

Weakly Fair Opacity. We now turn to opacity and provide a definition of weakly fair opacity that also respects the Finite-Observation Principle, again by restricting finite opacity to weakly fair runs. Informally, Definition 3.5 means that any finite observation can be extended in a way compatible with a weakly fair non-secret run, hence making the occurrence of a secret uncertain for the observer.

Definition 3.5. (Weakly Fair Opacity)
The secret in a WF-PN $\mathcal{W} = \langle \mathcal{N}, W \rangle$ is said to be *WF-opaque* if, for any trace $\hat{\sigma}$ in $Sec^*(\mathcal{N})$, there exists an infinite, weakly fair, non-secret trace $\rho \in Pub^{\omega}_{WF}(\mathcal{N})$ such that $\mathcal{O}(\hat{\sigma}) \leq \mathcal{O}(\rho)$.

Example 3.6. The WF-PN $\mathcal{W}_1 = \langle \mathcal{N}_1, \{s_2\} \rangle$ depicted in Figure 2(b) shows that weakly fair opacity is more discriminating that standard opacity. We consider the observation mask $\mathcal{O}$ defined by $\mathcal{O}(u_1) = \mathcal{O}(u_2) = \mathcal{O}(s_1) = \mathcal{O}(s_2) = \varepsilon$, for two secret transitions s_1 and s_2, $\mathcal{O}(t_1) = a$ and $\mathcal{O}(t_2) = b$.

The net $\mathcal{N}_1$ is opaque according to Definition 2.10 because the finite secret traces are observed as $\mathcal{O}(Sec^*(\mathcal{N}_1)) = a^*$, while the non-secret traces are observed as $a^* \cup b^*$, and the former is a subset of the latter.

On the other hand, the secret is not WF-opaque in $\mathcal{W}_1$ according to Definition 3.5. Let $\hat{\sigma} = s_1 t_1 \in Sec^*(\mathcal{N}_1)$. Then $\mathcal{O}(\hat{\sigma}) = a$, and the set of infinite, weakly fair traces ρ such that $\mathcal{O}(\hat{\sigma}) < \mathcal{O}(\rho)$ is $s_1 t_1^{\omega} \cup u_1 t_1^* s_2 t_1^{\omega}$, and all of these traces contain a secret transition.

As with Lemma 3.4, we obtain from Lemma 2.11 that WF-opacity and opacity coincide when no transition is weakly fair, thus Definition 3.5 is a proper generalisation of Definition 2.10.

Lemma 3.7. Let $\mathcal{W} = \langle \mathcal{N}, W \rangle$ be a live convergent WF-PN such that $W = \emptyset$. Then the secret is WF-opaque in $\mathcal{W}$ if and only if it is opaque in $\mathcal{W}$.

Comparing Definition 2.10 and Definition 3.5, we see that the formulation of WF-opacity is considerably more complex than the simple inclusion required by standard opacity. It is tempting to 'simplify' Definition 3.5 by mimicking Definition 2.10, but restricting to weakly fair traces, i.e., to demand that $\mathcal{O}(Sec^{\omega}_{WF}(\mathcal{N})) \subseteq \mathcal{O}(Pub^{\omega}_{WF}(\mathcal{N}))$. However, such a definition would not respect the Finite-Observation Principle.

Example 3.8. For the WF-PN $\mathcal{W}_2 = \langle \mathcal{N}_2, \{t_1\} \rangle$ depicted in Figure 8, we consider the observation mask $\mathcal{O}$ defined by $\mathcal{O}(u) = \mathcal{O}(s) = \varepsilon$, $\mathcal{O}(t_1) = a$ and $\mathcal{O}(t_2) = b$.

In $\mathcal{W}_2$, the system can either fire the secret transition s and then infinitely often t_2, or it can fire u and then arbitrarily often a and b, where the weak fairness condition requires to fire a infinitely often. Thus, $\mathcal{O}(\mathit{Sec}^{\omega}_{WF}(\mathcal{N}_2)) = b^{\omega}$, and $\mathcal{O}(\mathit{Pub}^{\omega}_{WF}(\mathcal{N}_2)) = (b^*a)^{\omega}$; since the first set is not included in the second, a definition based on the above inclusion would declare $\mathcal{W}_2$ non-opaque. However, even when s is fired, no *finite* observation is sufficient to determine that this was the case; indeed an observation b^n, for any $n \geq 0$, could also be the consequence of firing u first. Definition 3.5 captures this fact: for any $\hat{\sigma} = st_2^n \in \mathit{Sec}^*(\mathcal{N}_2)$ there exists an infinite WF trace without secret, e.g. $\rho = ut_2^n t_1^{\omega}$ satisfying $\mathcal{O}(\hat{\sigma}) < \mathcal{O}(\rho)$, thus $\mathcal{W}_2$ is WF-opaque.

3.4. No Weakly Fair Faults: The Strict Case

We finally investigate the special case where fault transitions are not weakly fair, i.e., a fault is a *possible* outcome in the system but not one that is *required* to happen: we call *strict WF-PN* a WF-PN $\mathcal{W} = \langle \mathcal{N}, W \rangle$ where $W \cap F = \emptyset$. Under this assumption, weakly fair diagnosability has a simple characterisation, reminiscent of Definition 2.1, which generalises [4, Lem. 3.4 and 3.5] to general Petri nets. Note that this also provides an alternative proof of Lemma 3.4.

Lemma 3.9. Let $\mathcal{W} = \langle \mathcal{N}, W \rangle$ be a strict convergent WF-PN. Then $\mathcal{W}$ is WF-diagnosable if and only if $\mathcal{O}(\mathit{Faulty}^{\omega}_{WF}(\mathcal{N})) \cap \mathcal{O}(\mathit{Correct}^{\omega}(\mathcal{N})) = \emptyset$.

Proof:
For the 'only if' part, assume there exists $\sigma \in \mathit{Faulty}^{\omega}_{WF}(\mathcal{N})$ and $\rho \in \mathit{Correct}^{\omega}(\mathcal{N})$ such that $\mathcal{O}(\sigma) = \mathcal{O}(\rho)$. If ρ is weakly fair, then $\mathcal{W}$ is not WF-diagnosable. Otherwise, consider some prefix $\hat{\sigma} < \sigma$ and let us build a suitable $\rho_{\hat{\sigma}} \in \mathit{Correct}^{\omega}_{WF}(\mathcal{N})$. Let $j \in \mathbb{N}$ be an index such that $\mathcal{O}(\hat{\sigma}) \leq \mathcal{O}(\hat{\rho})$ for the prefix $\hat{\rho}$ of length j of ρ.

Since ρ is not weakly fair, it must violate (WF.2) for some $t \in W$: writing $\boldsymbol{m}_0 \overset{t_1}{\Rightarrow} \boldsymbol{m}_1 \overset{t_2}{\Rightarrow} \cdots$ for its underlying run, this means that there are infinitely many indices i such that $\boldsymbol{m}_i(p) \geq w(p, t_{i+1}) + w(p, t)$ for all $p \in P$. Thus for infinitely many i, $\boldsymbol{m}_i(p) - w(p,t) + w(t,p) \geq w(p, t_{i+1})$ for all $p \in P$. Since $t \notin F$, this means that we can insert a transition by t in all those indices $i > j$ and still obtain a trace in $\mathit{Correct}^{\omega}(\mathcal{N})$; however this trace now satisfies (WF.1) for t. Applying this to all the $t \in W$ for which ρ was not weakly fair yields a weakly fair trace $\rho_{\hat{\sigma}} \in \mathit{Correct}^{\omega}_{WF}(\mathcal{N})$. Furthermore, since we inserted those occurrences of t (which might be observable) after the index j, $\mathcal{O}(\hat{\sigma}) \leq \mathcal{O}(\hat{\rho}) < \mathcal{O}(\rho_{\hat{\sigma}})$. Hence $\mathcal{W}$ is not WF-diagnosable.

Conversely, for the 'if' part, assume that $\mathcal{W}$ is not WF-diagnosable: there exists $\sigma \in \mathit{Faulty}^{\omega}_{WF}(\mathcal{N})$ such that for every prefix $\hat{\sigma} < \sigma$, there exists $\rho_{\hat{\sigma}} \in \mathit{Correct}^{\omega}_{WF}(\mathcal{N})$ with $\mathcal{O}(\hat{\sigma}) \leq \mathcal{O}(\rho_{\hat{\sigma}})$. Since $\mathit{Correct}^{\omega}_{WF}(\mathcal{N}) \subseteq \mathit{Correct}^{\omega}(\mathcal{N})$, σ is finitely indistinguishable from $\mathit{Correct}^{\omega}(\mathcal{N})$. Now Claim 2.7 implies that there is $\rho \in \mathit{Correct}^{\omega}(\mathcal{N})$ with $\mathcal{O}(\sigma) = \mathcal{O}(\rho)$, which concludes the proof. □

Remark 3.10. Lemma 3.9 no longer holds when one drops the condition that $\mathcal{A}$ be convergent. As a counterexample, it suffices to take the finite LTS from Figure 6(d), turned into a Petri net with no WF transitions, cf Remark 2.8.

4. Lower Bounds

In this section, we give reductions that yield lower bounds for the problems of diagnosability and opacity. Notice that we first study the problem variants *without* weak fairness. Thanks to Lemma 3.4 and Lemma 3.7, these lower bounds also apply to the WF variants of both problems: checking diagnosability/opacity for the special case of a WF-PN $\langle \mathcal{N}, \emptyset \rangle$ is equivalent to checking WF-diagnosability/WF-opacity for a PN $\mathcal{N}$. For the hardness of WF-diagnosability, we show a reduction from the reachability problem for PNs.

Live and Convergent Nets. As we saw in Lemma 2.6 and Lemma 2.11, in the absence of weak fairness constraints, (most of) the various definitions of diagnosability and opacity turn out to be equivalent when the transition systems under consideration are finitely branching, live, and convergent. As we wish our results to have the widest possible applicability, we shall require these properties of all the systems we study in lower bound proofs—but not necessarily in upper bound proofs. Because Petri nets yield finitely branching LTSs, we only need our nets to be live and convergent.

Remark 4.1. A Petri net can always be made live by adding an observable 'clock tick' transition connected back-and-forth to a single, initially marked place (like the transition c in Figure 2(a)). Intuitively, such a transition can be understood as modelling the passage of time marked by an observer when nothing else happens in the system.

Importantly, the addition of a 'clock tick' transition does not change the properties of diagnosability and opacity in our constructions. Thus the liveness assumption essentially comes 'for free' in Petri nets.

4.1. Diagnosability

For diagnosability, we reduce from the *coverability problem*: Given a PN $\mathcal{N}$ and a place p, is there a reachable marking $\boldsymbol{m}$ such that $\boldsymbol{m}(p) \geq 1$?

Proposition 4.2. (Hardness of Diagnosability)
Diagnosability is PSPACE-hard for safe Petri nets and EXPSPACE-hard in general, already for live convergent nets.

Proof:
We exhibit a polynomial time reduction from the coverability problem to non diagnosability. The coverability problem is known to be PSPACE-complete for safe Petri nets [29] and EXPSPACE-hard in general [23]. The statement follows because these two complexity classes are closed under complement.

Let $\mathcal{N} = \langle P, T, w, \boldsymbol{m}_0 \rangle$ be a PN and let $p \in P$. We construct a live and convergent PN $\mathcal{N}'$ and an observation mask $\mathcal{O}$ such that a marking $\boldsymbol{m}$ with $\boldsymbol{m}(p) \geq 1$ can be reached in $\mathcal{N}$ if and only if $\mathcal{N}'$ is not diagnosable; furthermore $\mathcal{N}'$ is safe whenever $\mathcal{N}$ is safe.

The construction consists in adding to $\mathcal{N}$ a single new place $q \notin P$, initially marked, and a single unobservable faulty transition $f \notin T$ taking one token from p and q to fire and putting the token

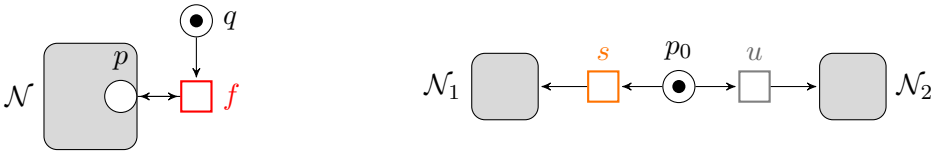


Figure 9. Constructions for the nets $\mathcal{N}'$ in Proposition 4.2 (left) and Proposition 4.4 (right).

back into p afterwards (see left part of Figure 9). Thus $\mathcal{N}' \stackrel{\text{def}}{=} \langle P', T', w', \boldsymbol{m}_0 \rangle$ with $P' \stackrel{\text{def}}{=} P \cup \{q\}$, $T' \stackrel{\text{def}}{=} T \cup \{f\}$ and w' coincides with w on $P \times T \cup T \times P$, with $w'(q, f) = w'(p, f) = w'(f, p) = 1$ in addition. For the observation mask $\mathcal{O}$, we let $E \stackrel{\text{def}}{=} T$ and all transitions from T are observable with $\mathcal{O}(t) \stackrel{\text{def}}{=} t$ and $\mathcal{O}(f) \stackrel{\text{def}}{=} \varepsilon$. The faulty transition f can fire once in $\mathcal{N}'$ if and only if there is a reachable marking $\boldsymbol{m}$ in $\mathcal{N}$ with $\boldsymbol{m}(p) \geq 1$. In this case, all the infinite runs in $\mathcal{N}'$ reaching $\boldsymbol{m}$ have ambiguous observations.

The construction ensures that if $\mathcal{N}$ is safe then it is also the case for $\mathcal{N}'$. Since f can fire only once and no transition from $\mathcal{N}$ is erased, $\mathcal{N}'$ is convergent. It is not necessarily live since $\mathcal{N}$ may contain a deadlock; however, $\mathcal{N}'$ can be made live by adding a 'clock tick' transition (cf. Remark 4.1) without affecting the validity of the reduction. □

4.2. Opacity

For the opacity problem, we prove our hardness results by reducing from the *trace-inclusion problem* for Petri nets: Given two PNs $\mathcal{N}_1$ and $\mathcal{N}_2$ with associated observation masks $\mathcal{O}_1$ and $\mathcal{O}_2$ into the same E, is $\mathcal{O}_1(\mathit{Trace}^*(\mathcal{N}_1)) \subseteq \mathcal{O}_2(\mathit{Trace}^*(\mathcal{N}_2))$? This problem is well-known to be undecidable for general Petri nets and EXPSPACE-complete for safe Petri nets [19]. The same reduction was used in [11, Thm. 6] to establish undecidability for the so-called *initial opacity* state-based variant of opacity in Petri nets. Undecidability results for other variants are also provided in [30].

However, because we insist on our systems being convergent, some additional care is required: in our main reduction (c.f. Proposition 4.4), we need the two PNs $\mathcal{N}_1$ and $\mathcal{N}_2$ to be convergent, hence we need to show that the trace-inclusion problem remains hard even for convergent instances. Along the way, we re-discovered that its complexity in the safe case can be refined and shown to be ESPACE-complete (see Proposition 4.3), based on a reduction from the universality problem for shuffle expressions studied by Mayer and Stockmeyer [20].

In Appendix A, we provide an inductive construction of a safe PN $\mathcal{N}(e)$ with coverability language $L(e)$ for e a shuffle expression. This is basically Thompson's inductive construction of a finite-state automaton from a regular expression with an extra case for shuffles, but some additional care is required in order to ensure that $\mathcal{N}(e)$ is convergent. One last pitfall is that we work with trace languages instead of coverability languages; this is handled using an additional endmarker symbol.

Proposition 4.3. (Appendix A)
The trace-inclusion problem is ESPACE-complete for safe convergent Petri nets.

Reduction from the Trace-Inclusion Problem. We wrap-up our lower bound proof using a reduction from the trace-inclusion problem in convergent nets to the opacity problem.

Proposition 4.4. (Hardness of Opacity)
Opacity is ESPACE-hard for safe Petri nets, and undecidable in general, already for live convergent nets.

Proof:
We exhibit a polynomial time reduction from the trace-inclusion problem for convergent PNs to the opacity problem, which preserves safety. As seen in Proposition 4.3, the trace-inclusion problem is ESPACE-hard for safe convergent Petri nets. In the general case, it is undecidable by the generic proof of Jančar [31] for equivalence and preorder problems in Petri nets: given a 2-counter machine, his proof builds two Petri nets $\mathcal{N}_1$ and $\mathcal{N}_2$ with non erasing observation masks $\mathcal{O}_1$ and $\mathcal{O}_2$—thus those nets are convergent—, such that the machine halts if and only if $\mathcal{O}_1(\mathit{Trace}^*(\mathcal{N}_1)) \neq \mathcal{O}_2(\mathit{Trace}^*(\mathcal{N}_2))$.

For the reduction, let $\mathcal{N}_1 = \langle P_1, T_1, w_1, \boldsymbol{m}_{0,1}\rangle$ and $\mathcal{N}_2 = \langle P_2, T_2, w_2, \boldsymbol{m}_{0,2}\rangle$ be two convergent PNs, with observation masks $\mathcal{O}_1$ and $\mathcal{O}_2$ into the same alphabet E; without loss of generality they have disjoint sets of places and transitions.

We first build a convergent PN $\mathcal{N}'$ by adding a new place $p_0 \notin P_1 \cup P_2$, initially marked, and two new transitions s and u not in $T_1 \cup T_2$. The observation mask $\mathcal{O}'$ of $\mathcal{N}'$ extends $\mathcal{O}_1$ and $\mathcal{O}_2$ by $\mathcal{O}'(s) = \mathcal{O}'(u) = \varepsilon$. The construction (see right part of Figure 9) consists in linking p_0 to $\mathcal{N}_1$ and $\mathcal{N}_2$ through the transitions s and u respectively, making them produce the initial markings of $\mathcal{N}_1$ and $\mathcal{N}_2$. The convergence of $\mathcal{N}'$ results from that of $\mathcal{N}_1$ and $\mathcal{N}_2$. The construction ensures that if $\mathcal{N}_1$ and $\mathcal{N}_2$ are safe, so is $\mathcal{N}'$. Now the set of secret words in $\mathcal{N}'$ is observed as $\mathcal{O}_1(\mathit{Trace}^*(\mathcal{N}_1))$ while the set of non-secret words is observed as $\mathcal{O}_2(\mathit{Trace}^*(\mathcal{N}_2))$. Thus, the secret is opaque in $\mathcal{N}'$ if and only if the inclusion $\mathcal{O}_1(\mathit{Trace}^*(\mathcal{N}_1)) \subseteq \mathcal{O}_2(\mathit{Trace}^*(\mathcal{N}_2))$ holds.

Finally, adding a 'clock tick' as in Remark 4.1 to $\mathcal{N}'$ with a fresh observation $\flat \notin E$ yields the desired $\mathcal{N}$ and $\mathcal{O}$. Indeed, using the notations of Appendix A for the *shuffle* operation $\sqcup\!\sqcup$, $\mathcal{O}(\mathit{Sec}^*(\mathcal{N})) = \mathcal{O}'(\mathit{Sec}^*(\mathcal{N}')) \sqcup\!\sqcup \{\flat^n \mid n \in \mathbb{N}\}$ and $\mathcal{O}(\mathit{Pub}^*(\mathcal{N})) = \mathcal{O}'(\mathit{Pub}^*(\mathcal{N}')) \sqcup\!\sqcup \{\flat^n \mid n \in \mathbb{N}\}$, and inclusion holds between these two languages if and only if $\mathcal{O}'(\mathit{Sec}^*(\mathcal{N}')) \subseteq \mathcal{O}'(\mathit{Pub}^*(\mathcal{N}'))$. □

It is interesting to observe that the decidability proof for SNNI given in [17] relies on the fact that inclusion becomes decidable between languages of Petri nets when the net on the righthand side is unlabelled [32], which is ensured by the restriction on the observation mask.

4.3. Weakly Fair Diagnosability

We prove that WF-diagnosability is at least as hard as reachability—and thus EXPSPACE-hard [23]. The reduction itself is inspired by a hardness proof by Howell et al. [18, Th. 4.9] for deciding the existence of a weakly fair run.

Proposition 4.5. (Hardness of WF-Diagnosability)
There is a polynomial time reduction from Petri nets reachability to non WF-diagnosability, which outputs live convergent nets with $W \cap F = \emptyset$.

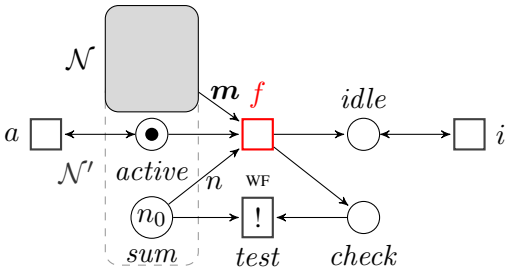


Figure 10. The Petri net $\mathcal{N}''$ in the proof of Proposition 4.5.

Proof:
Consider an instance $\langle \mathcal{N}, \boldsymbol{m} \rangle$ of the reachability problem where $\mathcal{N} = \langle P, T, w, \boldsymbol{m}_0 \rangle$ and $\boldsymbol{m} \in \mathbb{N}^P$. Define $n_0 \stackrel{\text{def}}{=} 1 + \sum_{p \in P} \boldsymbol{m}_0(p)$ and $n \stackrel{\text{def}}{=} 1 + \sum_{p \in P} \boldsymbol{m}(p)$.

We start by constructing a net $\mathcal{N}' \stackrel{\text{def}}{=} \langle P \uplus \{sum, active\}, T, w', \boldsymbol{m}'_0 \rangle$ that extends $\mathcal{N}$ with a 'checksum' place sum and a 'control' place $active$. The new initial marking $\boldsymbol{m}'_0$ extends $\boldsymbol{m}_0$ with $\boldsymbol{m}'_0(sum) \stackrel{\text{def}}{=} n_0$ and $\boldsymbol{m}'_0(active) \stackrel{\text{def}}{=} 1$. The flow w' is defined as w extended for all $t \in T$ with

- $w(sum, t) \stackrel{\text{def}}{=} \sum_{p \in P} w(p, t)$ and $w(t, sum) \stackrel{\text{def}}{=} \sum_{p \in P} w(t, p)$, which ensures that, in any reachable marking $\boldsymbol{m}'$ of $\mathcal{N}'$, $\boldsymbol{m}'(sum) = 1 + \sum_{p \in P} \boldsymbol{m}'(p)$;
- $w(active, t) \stackrel{\text{def}}{=} w(t, active) \stackrel{\text{def}}{=} 1$, which ensures that the original transitions in T can only be fired if $active$ is marked; we call such markings 'active'.

We now construct $\mathcal{N}''$ extending $\mathcal{N}'$ as shown in Figure 10. It features:

- a fault transition f that can be fired at most once, from an active marking $\boldsymbol{m}'$ that covers $\boldsymbol{m}$, and the projection of $\boldsymbol{m}'$ to P was equal to $\boldsymbol{m}$ if and only if sum is empty as a result of firing f;
- a weakly fair transition $test$ that can be fired at most once, necessarily at some point after f was fired, and whose purpose is to test whether sum is empty;
- two transitions a and i, idling respectively when $active$ or $idle$ is marked—those do not change the current marking.

We define $E \stackrel{\text{def}}{=} \{a, e\}$ and let our observation mask $\mathcal{O}$ map every transition to a, except for $\mathcal{O}(test) \stackrel{\text{def}}{=} e$ and $\mathcal{O}(f) \stackrel{\text{def}}{=} \varepsilon$; we set $W \stackrel{\text{def}}{=} \{test\}$ and $F \stackrel{\text{def}}{=} \{f\}$. Observe that $\mathcal{N}''$ is live and convergent with $W \cap F = \emptyset$. It remains to prove the following claim.

Claim 4.6. The marking $\boldsymbol{m}$ is reachable in $\mathcal{N}$ if and only if $\mathcal{N}''$ is not WF-diagnosable.

Since $W \cap F = \emptyset$, by Lemma 3.9, $\mathcal{N}''$ is not WF-diagnosable if and only if there exist $\sigma \in \mathit{Faulty}^{\omega}_{WF}(\mathcal{N}'')$ and $\rho \in \mathit{Correct}^{\omega}(\mathcal{N}'')$ such that $\mathcal{O}(\sigma) = \mathcal{O}(\rho)$.

For the 'only if' direction, assume $\boldsymbol{m}_0 \stackrel{\hat{\sigma}}{\Rightarrow} \boldsymbol{m}$ in $\mathcal{N}$. Then the same transition sequence $\hat{\sigma}$ leads in $\mathcal{N}''$ to an active marking equal to $\boldsymbol{m}$ over P, with n tokens in sum. Then $\sigma \stackrel{\text{def}}{=} \hat{\sigma}\, f\, i^{\omega}$ can be fired

in $\mathcal{N}''$, and is weakly fair because sum becomes empty once f has fired. Defining $\rho \stackrel{\text{def}}{=} a^\omega$, we get $\mathcal{O}(\sigma) = a^\omega = \mathcal{O}(\rho)$ and $\mathcal{N}''$ is therefore not WF-diagnosable.

For the 'if' direction, let us first consider any $\rho \in \mathit{Correct}^\omega(\mathcal{N})$: as $check$ cannot be marked in any correct run, $test$ cannot be fired, and $\mathcal{O}(\rho) = a^\omega$. Turning our attention to $\sigma \in \mathit{Faulty}^\omega_{WF}(\mathcal{N}'')$, since it is faulty, f has been fired, hence the run on σ is of the form $\boldsymbol{m}''_0 \stackrel{\hat{\sigma}}{\Rightarrow} \boldsymbol{m}' \stackrel{f}{\Rightarrow} \boldsymbol{m}'' \stackrel{\sigma'}{\Rightarrow}$ in $\mathcal{N}''$. We know that $\boldsymbol{m}'(p) \geq \boldsymbol{m}(p)$ for all $p \in P$ because f could be fired from $\boldsymbol{m}'$. Assume for the sake of contradiction that $\boldsymbol{m}'(p) > \boldsymbol{m}(p)$ for some $p \in P$, and let us show that it implies that σ is not weakly fair; this will prove that $\boldsymbol{m}$ was reachable in $\mathcal{N}$.

By the invariant on sum, $\boldsymbol{m}'(p) > \boldsymbol{m}(p)$ for some $p \in P$ entails $\boldsymbol{m}''(sum) > 0$ and therefore that $test$ is enabled in $\boldsymbol{m}''$. However, if $test$ were fired in σ', this would entail $\mathcal{O}(\sigma) \in a^*ea^\omega \neq \mathcal{O}(\rho)$ (thus σ does not satisfy (WF.1)). Furthermore, σ does not satisfy (WF.2) either, since, once f has fired, $test$ is the only fireable transition with either sum or $check$ in its preset. □

5. Upper Bounds

In this section, we give upper complexity bounds in Section 5.1 for safe WF-PNs, that match the lower bounds of the previous section. For general Petri nets in Section 5.2, since opacity is undecidable, we only consider diagnosability and show that the problem is EXPSPACE-complete in the absence of weak fairness. We also consider strict WF-PNs and show an exponential time reduction to the reachability problem in this case. The general case of WF-PN remains open.

5.1. Safe Petri Nets

In the case of safe Petri nets, our upper complexity bounds for checking diagnosability and opacity *with weak fairness* match the lower bounds of Section 4 for the variants without weak fairness. From this viewpoint, weak fairness can be included 'for free' in diagnosability and opacity checking for concurrent systems.

WF-Diagnosability. Germanos et al. [4] show that, given a convergent WF-PN $\mathcal{W} = \langle \mathcal{N}, W \rangle$, one can construct in polynomial time a PN $\mathcal{N}'$ and a state-based LTL formula φ, such that $\mathcal{W}$ is WF-diagnosable if and only if $\mathcal{N}'$ has an infinite run satisfying φ. Since LTL model-checking of safe PNs is in PSPACE [19], the same upper bound applies to WF-diagnosis, which shows that the lower bound in Proposition 4.2 is tight.

Proposition 5.1. WF-diagnosability is in PSPACE for safe convergent Petri nets.

Weakly Fair Opacity. In the case of WF-opacity, we argue directly that there is an ESPACE algorithm for safe Petri nets, matching the lower bound from Proposition 4.4.

Proposition 5.2. WF-opacity is in ESPACE for safe Petri nets.

Proof:
Let $\mathcal{W}$ be a safe WF-PN with n places. We sketch a non-deterministic algorithm $\mathcal{M}$ working in space $2^{O(n)}$ that checks for the negation of Definition 3.5 in $\mathcal{W}$. The result then follows from Savitch's Theorem showing NESPACE = ESPACE and the fact that ESPACE is deterministic.

We must look for a finite prefix $\hat{\sigma}$ of a run that uses a secret transition, and such that there exists no infinite WF trace $\rho \in \mathit{Pub}^{\omega}_{WF}(\mathcal{W})$ satisfying that $\mathcal{O}(\hat{\sigma}) \leq \mathcal{O}(\rho)$. The algorithm works in two phases:

1. the first phase nondeterministically picks a suitable prefix $\hat{\sigma}$ 'on the fly', along with the set $M \subseteq 2^n$ of possible markings reachable by some $\hat{\rho} \in \mathit{Pub}^*(\mathcal{W})$ with $\mathcal{O}(\hat{\sigma}) = \mathcal{O}(\hat{\rho})$—this can be carried in space $2^{O(n)}$—and

2. the second phase checks whether any marking $\boldsymbol{m} \in M$ can start a weakly fair infinite run; this can be verified by a model-checking algorithm for LTL—in PSPACE [19]. □

5.2. General Petri Nets

Because opacity is undecidable for general Petri nets by Proposition 4.4, this section shall focus on (WF-)diagnosability. We rely on decidable fragments of LTL on Petri net runs. The first step in the following reductions is to build (in polynomial time) a suitable *verifier net*, which is the counterpart for Petri nets of the twin-plant construction.

5.2.1. Verifier Net

From the WF-PN $\mathcal{W}$, the construction of the verifier net $\mathcal{V}(\mathcal{W})$ consists simply in synchronising two copies $\mathcal{W}_1$ and $\mathcal{W}_2$ of $\mathcal{W}$ on their observations while letting unobservable transitions run asynchronously, and discarding fault transitions from the second copy. Variants of this construction were used for instance in [26, 21, 4]. We give here the full construction for the sake of completeness, including a special place *fault* that receives a token whenever a faulty transition is fired.

Let $\mathcal{W} = \langle \mathcal{N}, W \rangle$ be the original WF-PN with $\mathcal{N} = \langle P, T, w, \boldsymbol{m}_0 \rangle$ and $F \subseteq T$ be the set of faults. It is convenient to make a distinction between the observable transitions $O \stackrel{\text{def}}{=} \{t \in T \mid \mathcal{O}(t) \in E\}$ and the non-observable ones $U \stackrel{\text{def}}{=} \{t \in T \mid \mathcal{O}(t) = \varepsilon\}$. Thus $F \subseteq U$ and $T = O \uplus U$.

We start by making two disjoint copies of $\mathcal{W}$, called respectively $\mathcal{W}_1 \stackrel{\text{def}}{=} \langle P_1, T_1, w_1, \boldsymbol{m}_{0,1} \rangle$ and $\mathcal{W}_2 \stackrel{\text{def}}{=} \langle P_2, T_2, w_2, \boldsymbol{m}_{0,2} \rangle$, where $\mathcal{W}_1$ is an exact copy of $\mathcal{W}$ but $\mathcal{W}_2$ omits the fault transitions: T_2 is a copy of $T \setminus F$. We write O_1, O_2 (resp. U_1, U_2, resp. W_1, W_2) for the sets of observable (resp. non observable, resp. weakly fair) transitions in $\mathcal{W}_1$ and $\mathcal{W}_2$; hence e.g. W_2 is a copy of $W \setminus F$ and U_2 of $U \setminus F$.

The verifier net is then $\mathcal{V}(\mathcal{W}) \stackrel{\text{def}}{=} \langle \mathcal{N}', W' \rangle$ where $W' \stackrel{\text{def}}{=} W_1 \uplus W_2$ and $\mathcal{N}' \stackrel{\text{def}}{=} \langle P', T', w', \boldsymbol{m}'_0 \rangle$ is defined as follows:

$$
\begin{aligned}
&P' \stackrel{\text{def}}{=} P_1 \uplus P_2 \uplus \{\mathit{fault}\}\,,\\
&O' \stackrel{\text{def}}{=} \{\langle t_1, t_2 \rangle \mid t_1 \in O_1, t_2 \in O_2, \mathcal{O}(t_1) = \mathcal{O}(t_2)\}\,,\ U'_1 \stackrel{\text{def}}{=} U_1 \times \{\varepsilon\}\,,\ U'_2 \stackrel{\text{def}}{=} \{\varepsilon\} \times U_2\\
&T' \stackrel{\text{def}}{=} O' \cup U'_1 \cup U'_2\,,\\
&F' \stackrel{\text{def}}{=} F_1\,,
\end{aligned}
$$

with flow mapping

$$\begin{aligned}
w'(p,\langle t_1,t_2\rangle) \stackrel{\text{def}}{=} w(p,t_i)\,,\ w'(\langle t_1,t_2\rangle,p) \stackrel{\text{def}}{=} w(t_i,p) \qquad & \forall \langle t_1,t_2\rangle \in O'\,\forall p\in P_i\,\forall i\in\{1,2\}\,,\\
w'(p,t) \stackrel{\text{def}}{=} w(p,t)\,,\ w'(t,p) \stackrel{\text{def}}{=} w(t,p) \qquad & \forall t\in U_i'\,\forall p\in P_i\,\forall i\in\{1,2\}\,,\\
w'(p,t) \stackrel{\text{def}}{=} w'(t,p) \stackrel{\text{def}}{=} 0 \qquad & \forall t\in U_i'\,\forall p\in P_{3-i}\,\forall i\in\{1,2\}\,,\\
w'(\mathit{fault},t) \stackrel{\text{def}}{=} 0 \qquad & \forall t\in T'\,,\\
w'(t,\mathit{fault}) \stackrel{\text{def}}{=} 0 \qquad & \forall t\in T'\setminus F'\,,\\
w'(t,\mathit{fault}) \stackrel{\text{def}}{=} 1 \qquad & \forall t\in F'\,,
\end{aligned}$$

initial marking

$$\begin{aligned}
\boldsymbol{m}_0'(p) \stackrel{\text{def}}{=} \boldsymbol{m}_{0,i}(p) \qquad & \forall p\in P_i\,\forall i\in\{1,2\}\,,\\
\boldsymbol{m}_0'(\mathit{fault}) \stackrel{\text{def}}{=} 0\,,
\end{aligned}$$

and observation mask

$$\begin{aligned}
\mathcal{O}'(\langle t_1,t_2\rangle) \stackrel{\text{def}}{=} \mathcal{O}(t_1) \qquad & \forall \langle t_1,t_2\rangle\in O'\,,\\
\mathcal{O}'(t) \stackrel{\text{def}}{=} \varepsilon \qquad & \forall t\in U_1'\cup U_2'\,.
\end{aligned}$$

The place *fault* receives a token whenever a faulty transition is fired and plays the role of the additional component of the twin plant. Hence, $\mathcal{A}_{\mathcal{V}(\mathcal{W})}$ behaves exactly like $\mathit{Twin}(\mathcal{A}_{\mathcal{N}})$.

Example 5.3. In the (safe) divergent PN $\mathcal{N}$ without WF transition depicted in Figure 11(a), $O=\{t_1\}$ and $U=\{u,f\}$. Since no observable transition can be synchronised, with $O'=\emptyset$, and we see that $U_1'=\{\langle u,\varepsilon\rangle,\langle f,\varepsilon\rangle\}$, $U_2'=\{\langle\varepsilon,u\rangle\}$, and $F'=\{\langle f,\varepsilon\rangle\}$. The verifier net $\mathcal{V}(\mathcal{N})$ in Figure 11(b) is a juxtaposition of

- the original net without the fault f nor t_1, and
- a copy with the additional *fault* place where p_1', p_2' are copies of p_1, p_2, and without t_1.

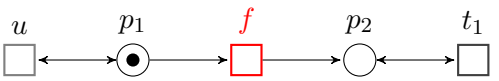
The reachability graph of $\mathcal{V}(\mathcal{N})$ depicted in Figure 11(c) presents markings as vectors according to the order $p_1', p_2', p_1, \mathit{fault}$. It can be noticed that it coincides with the twin plant of Figure 3.

5.2.2. Trace-Diagnosability

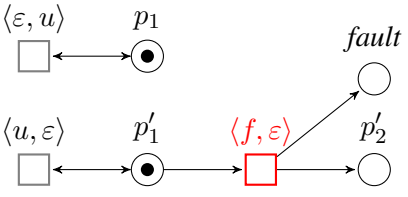
As shown in Lemma 2.5, $\mathcal{W}$ is not trace-diagnosable (thus ignoring weak fairness constraints for the moment) if and only if there exists an infinite run σ in $\mathit{Twin}(\mathcal{A}_{\mathcal{N}})$ that satisfies φ_{diag}. Translated in terms of $\mathcal{V}(\mathcal{W})$, the formula becomes

$$\Diamond \bigvee_{f\in F'} f \quad\wedge\quad \Box\Diamond \bigvee_{t\in O'\cup U_1'} t \quad\wedge\quad \Box\Diamond \bigvee_{t\in O'\cup U_2'} t\,. \tag{1}$$

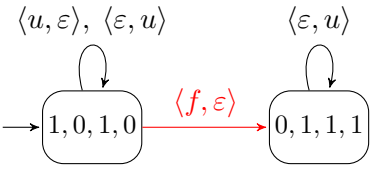
This is an *action-based* LTL formula, for which model-checking in general Petri nets can be performed in EXPSPACE [33], hence in the absence of weakly fair transitions the lower bound from Proposition 4.2 is tight.



(a) Divergent net $\mathcal{N}$ that is 1-diagnosable.



(b) Verifier net $\mathcal{V}(\mathcal{N})$.



(c) Reachability graph of $\mathcal{V}(\mathcal{N})$.

Figure 11. Example of a verifier net where $\mathcal{O}(t_1) = a$ and $\mathcal{O}(u) = \mathcal{O}(f) = \varepsilon$.

Proposition 5.4. Trace-diagnosability for Petri nets is in EXPSPACE.

In the convergent case, this result dramatically improves over the procedures proposed in [21] for uniform diagnosability and dynamic diagnosability, which relied on the explicit construction of the coverability graph with a worst-case Ackermannian complexity. Let us also note that convergence seems to be necessary for these procedures, as Figure 11 provides a counter-example to their Theorem 6.7 in the divergent case.

Still in the convergent case, Proposition 5.4 also eschews an issue in a very recent article of Yin and Lafortune [34], which claims the same EXPSPACE upper bound for dynamic diagnosability, but relies crucially on a flawed result of Yen debunked by Atig and Habermehl [35].

5.2.3. Strict WF-Diagnosability

By Lemma 3.9, if no fault is weakly fair in a convergent WF-PN, then non WF-diagnosability is equivalent to the existence of a run satisfying φ_{diag} and whose projection on transitions from the first copy is weakly fair. In order to check those conditions, we are going to use another fragment of LTL proven decidable over Petri nets by Jančar [10]. The fragment LTL($\Box\Diamond$) can use both actions and states in its atomic propositions, but only allows positive Boolean combinations of 'infinitely often' $\Box\Diamond$ formulæ at top-level.

As LTL($\Box\Diamond$) does not feature $\Diamond$ on its own, we cannot use φ_{diag} directly, and we use the fact that in our construction of $\mathcal{V}(\mathcal{W})$ in Section 5.2.1, all the fault transitions add a token to the initially empty *fault* place; once *fault* is marked, it remains so forever. Then non WF-diagnosability is equivalent to the existence of an infinite run of $\mathcal{V}(\mathcal{W})$ satisfying

$$\Box\Diamond(\mathit{fault} > 0) \wedge \bigwedge_{t \in W_1} \left((\Box\Diamond t) \vee \left(\Box\Diamond \bigvee_{t' \in T_1} \bigvee_{p \in P_1} t' \wedge \big(p < w(p,t) + w(p,t')\big) \right) \right). \tag{2}$$

Because Jančar [10] proved existential LTL($\Box\Diamond$) model checking of Petri nets to reduce in exponential time to the reachability problem, by Proposition 4.5 we get an equivalence between non WF-diagnosability when $W \cap F = \emptyset$ and reachability, modulo exponential-time many-one reductions.

Proposition 5.5. There is an exponential time reduction from non WF-diagnosability in strict convergent WF-PNs to the reachability problem.

6. Concluding Remarks

We have revisited the problems of diagnosability and opacity with a focus on expressivity for concurrent systems, and introduced a new notion of opacity for Petri nets under weakly fair semantics.

We have conducted a comparative study of complexity for both diagnosability and opacity analysis. Not surprisingly, opacity is always harder than diagnosability, and complexity also increases when moving from automata to safe Petri nets to general Petri nets, i.e., from the sequential to the concurrent to the infinite.

Safe Petri Nets. Note that the price to pay in safe Petri nets for the extra precision of analysis under *weak fairness*—which allows to capture indirect dependencies, as seen above and in [6, 8]—is not higher than for the corresponding analyses with ordinary semantics. We therefore argue that the refined notions of WF-diagnosability from [8, 4], and of WF-opacity that we have introduced in this paper, are valid and important contributions to the design and monitoring of concurrent systems. Future work should investigate efficient algorithms for the analysis of partially observed Petri nets.

General Petri Nets. For strict WF-PNs, Proposition 4.5 leaves an exponential complexity gap with our upper bound in Proposition 5.5. It might be worth investigating whether this gap could be filled by considering a reduction from reachability in *succinctly* presented Petri nets. In the general case, the main difficulty is that Definition 3.3 is essentially a branching-time property, which are generally undecidable in Petri nets. It is however quite a specific property, as can be seen in the case of safe Petri nets where it can be reduced to a linear-time property [4, Lem. 3.4]—unfortunately this reduction does not hold in general Petri nets—, and this might explain why we could not prove it undecidable either.

Acknowledgements

The authors express their gratitude to the anonymous reviewers, in particular to the one who provided us with the example in Figure 6(c), which led us to correct lemmata 2.6, 2.11, and 3.9.

A large part of the work was done while the first author was on an Inria-funded leave at LSV, ENS Paris-Saclay.

A. Trace Inclusion in Safe Petri Nets

We establish here the complexity of the trace inclusion problem for safe Petri nets, namely Proposition 4.3. The key technical argument is the ESPACE-hardness of the related *language inclusion problem*, where we define the (coverability) *language* of a Petri net $\mathcal{N} = \langle P, T, w, \boldsymbol{m}_0 \rangle$ with target marking $\boldsymbol{m}$ and observation mask $\mathcal{O}$ as

$$Lang^*(\mathcal{N}) \stackrel{\text{def}}{=} \{\mathcal{O}(\sigma) \in E^* \mid \exists \boldsymbol{m}' \geq \boldsymbol{m} : \boldsymbol{m}_0 \xrightarrow{\sigma} \boldsymbol{m}'\} \ ,$$

where $\geq$ denotes the componentwise ordering: $\boldsymbol{m}' \geq \boldsymbol{m}$ if and only if $\forall p \in P, \boldsymbol{m}'(p) \geq \boldsymbol{m}(p)$; we also say that $\boldsymbol{m}'$ *covers* $\boldsymbol{m}$ in such a case. As explained in the main text, our proof relies on a reduction from the universality problem for shuffle expressions, which was shown ESPACE-complete by Mayer and Stockmeyer [20].

Shuffle Expressions. Recall that for some alphabet E, the *shuffle* of two words σ and ρ in E^* is the language $\sigma \sqcup\!\sqcup \rho \stackrel{\text{def}}{=} \{\sigma_1\rho_1\sigma_2\rho_2 \cdots \sigma_n\rho_n \mid n \in \mathbb{N}, \sigma_1 \cdots \sigma_n = \sigma, \rho_1 \cdots \rho_n = \rho\}$ (σ_i and ρ_i are words in E^*); this is lifted to $L \sqcup\!\sqcup M \stackrel{\text{def}}{=} \bigcup_{\sigma\in L,\rho\in M} \sigma \sqcup\!\sqcup \rho$ for two languages L and M.

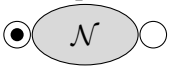
Shuffle expressions in SE are built according to the abstract syntax

$$e := \varepsilon \mid a \mid e + e \mid e \cdot e \mid e \sqcup\!\sqcup e \mid e^* \ ,$$

for $a \in E$. The language of an expression in SE is defined inductively by $L(\varepsilon) \stackrel{\text{def}}{=} \{\varepsilon\}$, $L(a) \stackrel{\text{def}}{=} \{a\}$ for all $a \in E$, $L(e_1 + e_2) \stackrel{\text{def}}{=} L(e_1) \cup L(e_2)$, $L(e_1 \cdot e_2) \stackrel{\text{def}}{=} L(e_1) \cdot L(e_2)$, $L(e_1 \sqcup\!\sqcup e_2) \stackrel{\text{def}}{=} L(e_1) \sqcup\!\sqcup L(e_2)$, and $L(e^*) \stackrel{\text{def}}{=} L(e)^*$, where e_1 and e_2 are two expressions. For instance, the finite-trace language of the LTS $\mathcal{A}$ from Figure 1, which corresponds to the net $\mathcal{N}$ from Figure 2(a), is the shuffle of the traces from both sub-nets: $\mathit{Trace}^*(\mathcal{A}) = L((fa^* + ub^*) \sqcup\!\sqcup c^*)$.

The *universality problem* for shuffle expressions asks, given e a shuffle expression over E, whether $E^* \subseteq L(e)$. This problem is ESPACE-complete since its complement is ESPACE-complete [20] and since ESPACE is closed under complement.

From Shuffle Expressions to Safe Petri Nets. Given a shuffle expression e, we shall construct a safe Petri net $\mathcal{N}(e)$ with $\mathit{Lang}^*(\mathcal{N}) = L(e)$. More precisely, $\mathcal{N}(e)$ is a *standard* safe Petri net, i.e. with a single initial place marked, which is not in the postset of any transition, and a single final place, not in the preset of any transition. Such a net is depicted as follows:

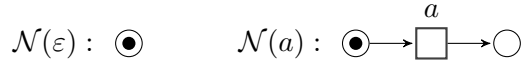


The initial marking has then a single token in the initial place, and the target marking a single token in the final place.

Lemma A.1. Given a shuffle expression e, we can construct in polynomial time a safe convergent Petri net $\mathcal{N}(e)$ with an observation mask and target marking such that $\mathit{Lang}^*(\mathcal{N}(e)) = L(e)$.

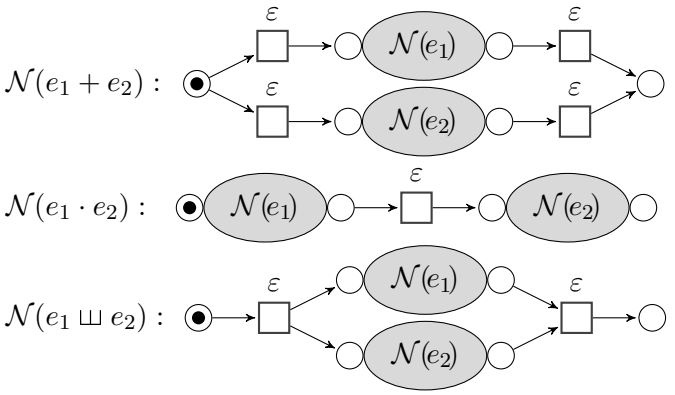
Proof:
By induction, we prove a slightly stronger result, also associating with an expression e a safe convergent PN $\mathcal{N}'(e)$ for $L(e) \setminus \{\varepsilon\}$; this will be needed in order to construct a convergent net for $L(e^*)$.

We start for the base case with the two safe convergent nets $\mathcal{N}(\varepsilon)$ and $\mathcal{N}(a) = \mathcal{N}'(a)$ associated respectively with ε and $a \in E$:



Removing ε from $L(\varepsilon)$ yields the empty set; the corresponding PN $\mathcal{N}'(\varepsilon)$ is therefore obtained with a single (unmarked) final place.

For the induction step, given two safe convergent PNs $\mathcal{N}(e_1)$ and $\mathcal{N}(e_2)$ for e_1, e_2 in SE, we build safe convergent PNs $\mathcal{N}(e_1 + e_2)$, $\mathcal{N}(e_1 \cdot e_2)$, and $\mathcal{N}(e_1 \sqcup\!\sqcup e_2)$:



The construction is the same for $\mathcal{N}'(e_1 + e_2)$ when combining $\mathcal{N}'(e_1)$ with $\mathcal{N}'(e_2)$ instead, since

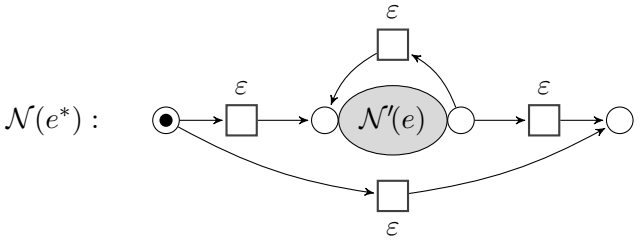
$$L(e_1 + e_2) \setminus \{\varepsilon\} = \big(L(e_1) \setminus \{\varepsilon\}\big) \cup \big(L(e_2) \setminus \{\varepsilon\}\big) \,. \tag{3}$$

For $\mathcal{N}'(e_1 \cdot e_2)$ and $\mathcal{N}'(e_1 \sqcup\!\sqcup e_2)$, we construct $\mathcal{N}'(e_1 \cdot e_2)$ and $\mathcal{N}'(e_1 \sqcup\!\sqcup e_2)$ using the construction for unions and the combinations of $\mathcal{N}(e_1)$ with $\mathcal{N}'(e_2)$ and of $\mathcal{N}'(e_1)$ with $\mathcal{N}(e_2)$:

$$L(e_1 \cdot e_2) \setminus \{\varepsilon\} = \big(L(e_1) \cdot (L(e_2) \setminus \{\varepsilon\})\big) \cup \big((L(e_1) \setminus \{\varepsilon\}) \cdot L(e_2)\big) \,, \tag{4}$$

$$L(e_1 \sqcup\!\sqcup e_2) \setminus \{\varepsilon\} = \big(L(e_1) \sqcup\!\sqcup (L(e_2) \setminus \{\varepsilon\})\big) \cup \big((L(e_1) \setminus \{\varepsilon\}) \sqcup\!\sqcup L(e_2)\big) \,. \tag{5}$$

Finally, the last remaining case of the induction step is that of Kleene stars. Given a safe convergent PN $\mathcal{N}'(e)$ for expression $L(e) \setminus \{\varepsilon\}$, we build a safe convergent PN $\mathcal{N}(e^*)$ using the fact that $L(e^*) = (L(e) \setminus \{\varepsilon\})^*$.



Here, assuming by induction that $Lang^*(\mathcal{N}'(e)) = L(e) \setminus \{\varepsilon\}$ and $\mathcal{N}'(e)$ is convergent, the resulting net $\mathcal{N}(e^*)$ is convergent as well. The corresponding net $\mathcal{N}'(e^*)$ is built in the same manner, except that it does not feature the bottom transition linking the initial and final places (whose purpose was precisely to add ε to the language).

This concludes the construction; its correctness is a straightforward induction on the shuffle expression. □

Proposition 4.3. The trace-inclusion problem is ESPACE-complete for safe convergent Petri nets.

Proof:
For membership in ESPACE, observe that the state space of $\mathcal{A}_\mathcal{N}$ is of size at most $2^{|P|}$ for a safe Petri

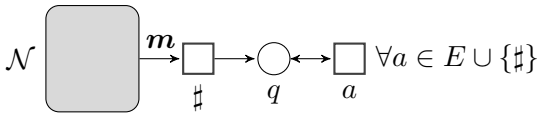


Figure 12. The Petri net $\mathcal{N}_2$ in the proof of Proposition 4.3.

net $\mathcal{N}$. Using standard arguments, trace non-inclusion $\mathit{Trace}^\omega(\mathcal{N}_1) \not\subseteq \mathit{Trace}^\omega(\mathcal{N}_2)$ is witnessed by the traces in the synchronous product of $\mathcal{A}_{\mathcal{N}_1}$ with the complement of $\mathcal{A}_{\mathcal{N}_2}$, the latter having at most $2^{2^{|P_2|}}$ states. If such a witness exists, there is thus one of length at most $2^{2^{|P_2|}+|P_1|}$, yielding an 'on the fly' nondeterministic algorithm working in space $O(2^{|P_2|} + |P_1|)$. As NESPACE = ESPACE by Savitch's Theorem, the upper bound follows.

For ESPACE-hardness, Lemma A.1 combined with [20, Th. 7.1] shows that the related *language universality problem* is ESPACE-hard: given a safe convergent Petri net $\mathcal{N} = \langle P, T, w, \boldsymbol{m}_0\rangle$ with an observation mask $\mathcal{O}$ and target marking $\boldsymbol{m}$, is $E^* \subseteq \mathit{Lang}^*(\mathcal{N})$? We reduce this problem to the trace inclusion problem. Let $\sharp \notin E$ be a fresh symbol. We first construct another safe convergent Petri net $\mathcal{N}_2 \stackrel{\text{def}}{=} \langle P_2, T_2, w_2, \boldsymbol{m}_{0,2}\rangle$ as described in Figure 12. It features a new place q that can only be marked from a marking that covers $\boldsymbol{m}$ by firing a new transition t with observation $\mathcal{O}_2(t) \stackrel{\text{def}}{=} \sharp$. From that point on, any sequence in $(E \cup \{\sharp\})^*$ can be observed. Note that $\mathcal{N}_2$ is also safe and convergent.

Claim A.2. $(E \cup \{\sharp\})^* \subseteq \mathcal{O}_2(\mathit{Trace}^*(\mathcal{N}_2))$ if and only if $E^* \subseteq \mathit{Lang}^*(\mathcal{N})$.

Indeed, assume σ is a word in $(E \cup \{\sharp\})^*$ and $E^* \subseteq \mathit{Lang}^*(\mathcal{N})$. If σ does not contain $\sharp$, then its run in $\mathcal{N}$ is also a run in $\mathcal{N}_2$ and thus $\sigma \in \mathcal{O}_2(\mathit{Trace}^*(\mathcal{N}_2))$. Otherwise, write $\sigma = \sigma'\sharp\sigma''$ with $\sigma' \in E^*$; then there is a run covering m in $\mathcal{N}_2$ with observation σ', and by subsequently firing t followed by the appropriate transition sequence once q is marked, $\sigma \in \mathcal{O}_2(\mathit{Trace}^*(\mathcal{N}_2))$. Conversely, assume σ is a word in E^* and $(E \cup \{\sharp\})^* \subseteq \mathcal{O}_2(\mathit{Trace}^*(\mathcal{N}_2))$. Then $\sigma\sharp \in \mathcal{O}_2(\mathit{Trace}^*(\mathcal{N}_2))$, and any run for it in $\mathcal{N}_2$ must finish by firing t after a run for σ that covers $\boldsymbol{m}$ in $\mathcal{N}$, which shows $\sigma \in \mathit{Lang}^*(\mathcal{N})$. This completes the proof of the claim.

Finally, we construct a safe convergent Petri net $\mathcal{N}_1$ and observation mask $\mathcal{O}_1$ with the property that $\mathcal{O}_1(\mathit{Trace}^*(\mathcal{N}_1)) = (E \cup \{\sharp\})^*$. As the construction of $\mathcal{N}_1$ and $\mathcal{N}_2$ can be carried in polynomial time, this proves the lower bound. □

References

[1] Sampath M, Sengupta R, Lafortune S, Sinnamohideen K, Teneketzis D. Diagnosability of discrete-event systems. *IEEE Transactions on Automatic Control*, 1995. **40**(9):1555–1575. doi:10.1109/9.412626.

[2] Jiang S, Huang Z, Chandra V, Kumar R. A polynomial algorithm for testing diagnosability of discrete event systems. *IEEE Transactions on Automatic Control*, 2001. **46**(8):1318–1321. doi:10.1109/9.940942.

[3] Yoo TS, Lafortune S. Polynomial-time verification of diagnosability of partially observed discrete event systems. *IEEE Transactions on Automatic Control*, 2002. **47**(9):1491–1495. doi:10.1109/TAC.2002.802763.

[4] Germanos V, Haar S, Khomenko V, Schwoon S. Diagnosability under weak fairness. *ACM Transactions on Embedded Computer Systems*, 2015. **14**(4:69). doi:10.1145/2832910.

[5] Haar S. Qualitative diagnosability of labeled Petri nets revisited. In: Proceedings of CDC'09 and CCC'09. IEEE, 2009 pp. 1248–1253. doi:10.1109/CDC.2009.5400917.

[6] Haar S. Types of asynchronous diagnosability and the *reveals*-relation in occurrence nets. *IEEE Transactions on Automatic Control*, 2010. **55**(10):2310–2320. doi:10.1109/TAC.2010.2063490.

[7] Haar S. What topology tells us about diagnosability in partial order semantics. *Discrete Event Dynamic Systems*, 2012. **22**(4):383–402. doi:10.1007/s10626-011-0121-z.

[8] Haar S, Rodríguez C, Schwoon S. Reveal your faults: it's only fair! In: Proceedings of ACSD'13. IEEE, 2013 pp. 120–129. doi:10.1109/ACSD.2013.15.

[9] Vogler W. Fairness and partial order semantics. *Information Processing Letters*, 1995. **55**(1):33–39. doi:10.1016/0020-0190(95)00049-I.

[10] Jančar P. Decidability of a temporal logic problem for Petri nets. *Theoretical Computer Science*, 1990. **74**(1):71–93. doi:10.1016/0304-3975(90)90006-4.

[11] Bryans J, Koutny M, Mazaré L, Ryan PYA. Opacity generalised to transition systems. *International Journal of Information Security*, 2008. **7**(6):421–435. doi:10.1007/s10207-008-0058-x.

[12] Cassez F, Dubreil J, Marchand H. Dynamic observers for the synthesis of opaque systems. In: Proceedings of ATVA'09, volume 5799 of *Lecture Notes in Computer Science*. Springer, 2009 pp. 352–367. doi:10.1007/978-3-642-04761-9_26.

[13] Tong Y, Li Z, Seatzu C, Giua A. Verification of initial-state opacity in Petri nets. In: Proceedings of CDC'15. IEEE, 2015 pp. 344–349. doi:10.1109/CDC.2015.7402224.

[14] Badouel E, Bednarczyk MA, Borzyszkowski AM, Caillaud B, Darondeau P. Concurrent secrets. *Discrete Event Dynamic Systems*, 2007. **17**(4):425–446. doi:10.1007/s10626-007-0020-5.

[15] Lin F. Opacity of discrete event systems and its applications. *Automatica*, 2011. **47**(3):496–503. doi:10.1016/j.automatica.2011.01.002.

[16] Busi N, Gorrieri R. Structural non-interference in elementary and trace nets. *Mathematical Structures in Computer Science*, 2009. **19**(6):1065–1090. doi:10.1017/S0960129509990120.

[17] Best E, Darondeau P, Gorrieri R. On the decidability of non interference over unbounded Petri nets. In: Proceedings of SecCo'10, volume 51 of *Electronic Proceedings in Theoretical Computer Science*. 2010 pp. 16–33. doi:10.4204/EPTCS.51.2.

[18] Howell RR, Rosier LE, Yen HC. A taxonomy of fairness and temporal logic problems for Petri nets. *Theoretical Computer Science*, 1991. **82**(2):341–372. doi:10.1016/0304-3975(91)90228-T.

[19] Esparza J. Decidability and complexity of Petri net problems—an introduction. In: Lectures on Petri Nets I: Basic Models, volume 1491 of *Lecture Notes in Computer Science*. Springer, 1996 pp. 374–428. doi:10.1007/3-540-65306-6_20.

[20] Mayer AJ, Stockmeyer LJ. The complexity of word problems—this time with interleaving. *Information and Computation*, 1994. **115**(2):293–311. doi:10.1006/inco.1994.1098.

[21] Cabasino MP, Giua A, Lafortune S, Seatzu C. A new approach for diagnosability analysis of Petri nets using verifier nets. *IEEE Transactions on Automatic Control*, 2012. **57**(12):3104–3117. doi:10.1109/TAC.2012.2200372.

[22] Schmitz S. Automata column: The complexity of reachability in vector addition systems. *ACM SIGLOG News*, 2016. **3**(1):3–21. doi:10.1145/2893582.2893585.

[23] Lipton R. The reachability problem requires exponential space. Technical Report 62, Yale University, 1976.

[24] Leroux J, Schmitz S. Demystifying reachability in vector addition systems. In: Proceedings of LICS'15. IEEE, 2015 pp. 56–67. doi:10.1109/LICS.2015.16.

[25] Bérard B, Haar S, Schmitz S, Schwoon S. The complexity of diagnosability and opacity verification for Petri nets. In: Proceedings of PN 2017, volume 10258 of *Lecture Notes in Computer Science*. Springer, 2017 pp. 200–220. doi:10.1007/978-3-319-57861-3_13.

[26] Madalinski A, Khomenko V. Diagnosability verification with parallel LTL-X model checking based on Petri net unfoldings. In: Proceedings of SysTol'10. IEEE, 2010 pp. 398–403.

[27] Even S. On information lossless automata of finite order. *IEEE Transactions on Electronic Computers*, 1965. **EC-14**(4):561–569. doi:10.1109/PGEC.1965.263996.

[28] Vardi MY, Wolper P. An automata-theoretic approach to automatic program verification. In: Proceedings of LICS 1986. IEEE, 1986 pp. 332–344.

[29] Jones ND, Landweber LH, Lien YE. Complexity of some problems in Petri nets. *Theoretical Computer Science*, 1977. **4**(3):277–299. doi:10.1016/0304-3975(77)90014-7.

[30] Tong Y, Li Z, Seatzu C, Giua A. Decidability of opacity verification problems in labeled Petri net systems. *Automatica*, 2017. **80**:48–53. doi:10.1016/j.automatica.2017.01.013.

[31] Jančar P. Nonprimitive recursive complexity and undecidability for Petri net equivalences. *Theoretical Computer Science*, 2001. **256**(1–2):23–30. doi:10.1016/S0304-3975(00)00100-6.

[32] Pelz E. Closure properties of deterministic Petri nets. In: Proceedings of STACS'87, volume 247 of *Lecture Notes in Computer Science*. Springer, 1987 pp. 371–382. doi:10.1007/BFb0039620.

[33] Habermehl P. On the complexity of the linear-time μ-calculus for Petri nets. In: Proceedings of PN'97, volume 1248 of *Lecture Notes in Computer Science*. Springer, 1997 pp. 102–116. doi:10.1007/3-540-63139-9_32.

[34] Yin X, Lafortune S. On the decidability and complexity of diagnosability for labeled Petri nets. *IEEE Transactions on Automatic Control*, 2017. **62**(11):5931–5938. doi:10.1109/TAC.2017.2699278.

[35] Atig MF, Habermehl P. On Yen's path logic for Petri nets. *International Journal of Foundations of Computer Science*, 2011. **22**(4):783–799. doi:10.1142/S0129054111008428.