Discrete Event Dyn Syst (2009) 19:451–470
DOI 10.1007/s10626-009-0084-5

# Prognosis of $\omega$-Languages for the Diagnosis of $*$-Languages: A Topological Perspective

**Andreas Bauer · Sophie Pinchinat**

Received: 20 April 2009 / Accepted: 19 August 2009 / Published online: 10 September 2009


**Abstract** This article offers a novel perspective on the diagnosis of $*$-languages via a topological characterization of $\omega$-languages. This allows for the different concepts that currently exist in diagnosis of discrete-event systems to be related to one another in a uniform setting and to study their complexity. For this purpose, we introduce the notion of prognosability of an $\omega$-language, which in the classical setting corresponds to testing if a language is diagnosable and prediagnosable. We show that we can build a prognoser for some $\omega$-language if this language is open and saturated, where openness is usually implied in the finitary setting. For both of these problems we present PSPACE algorithms, and establish that prognosability (i.e., whether or not a prognoser exists) for an $\omega$-language is a PSPACE-complete problem. Our new characterization offers a novel point of view in the classical setting of diagnosis.

**Keywords** Diagnosis

## 1 Introduction

The diagnosis of (discrete-event) systems, originally formalized in Sampath et al. (1995, 1996), consists of establishing a diagnosis on the status of the actual computation of the system regarding a given property, based on externally observable events of this computation. Another closely related topic is the prediction where the veracity of the property is to be foreseen, alike diagnosis of inevitability (Jéron et al. 2008). Upon an occurring fault, a diagnosis points to the fault after finitely many

This research was supported by the Marie Curie Scientific Project MASLOG 021669 (FP6-2004-Mobility-6) and Univ. de Rennes 1.

A. Bauer
Australian National University, Canberra, Australia

S. Pinchinat (✉)
IRISA, Rennes, France
e-mail: Sophie.Pinchinat@irisa.fr

events have been observed. Hence, the central difficulty in diagnosis and prediction problems is the imperfect information about the computations. Given a stream of observations, there are in general several behaviors of the systems that are consistent with this information. To know with certainty that the actual computation has the desired property, one needs to make sure that the entire set of consistent behaviors have this property. This is precisely what *diagnosability* is about.

Several procedures to decide diagnosability have been studied. The most efficient one is based on the *twin plant construction* (Jiang et al. 2001), which runs in quadratic time when the property is described by a deterministic finite-state automaton. Diagnosability is proved to be a necessary and sufficient condition for the existence of a bound in the number of observations that are needed to establish that the property holds. On the other hand, when the property is specified in a logical formalism such as LTL formulae or Büchi automata, diagnosability is not sufficient anymore to guarantee that a diagnosis on the computation can be delivered in a finite amount of time. This can be achieved only when the system satisfies an additional property called *prediagnosability* (Jiang and Kumar 2004).

In this article, we adopt the terminology and give a decision procedure for calling an $\omega$-language *prognosable* if it is diagnosable and prediagnosable in the classical setting outlined above. We will characterize diagnosability and prediagnosability in terms of classical mathematical concepts, *saturation* and *openness* in the topological space of infinite words, and study the complexity of the underlying decision problems.

Saturation involves a language and an equivalence relation: The language is saturated if it does not distinguish between equivalent elements. We give a PSPACE decision procedure for the saturation problem when the language is $\omega$-regular and the equivalence relation is $\omega$-rational. Our method is inspired from Peled et al. (1998). We specialize this general problem to the case of the observational equivalence between words induced by the imperfect information setting and show that this instance of the saturation problem, which corresponds precisely to diagnosability, is PSPACE-complete.

Openness of an $\omega$-language refers to a categorization in the so called *Cantor topology* over infinite words. An $\omega$-language is open in this topology if its words can be described by the concatenation of words taken from some $*$-language, and some infinite suffix. Intuitively, openness of an $\omega$-language means that in order to tell whether or not some word is in the language, it is sufficient to examine a sufficiently long, but finite prefix. This is trivially the case when we deal with $*$-properties in the classical setting. In the infinitary setting, however, no direct decision procedure was known, when the property in question is given in terms of a nondeterministic Büchi automaton, which we consider as a means of property specification and representation. We will give a PSPACE algorithm to decide openness of an $\omega$-language and leave open the question of a lower bound. To the best of our knowledge no such bound has been established in the literature.

Putting together both results for saturation and openness, we are then able to show that deciding prognosability of an $\omega$-language, represented in terms of a nondeterministic Büchi automaton, is a PSPACE-complete problem in the size of the automaton.

The paper is organized as follows: in Section 2 we formally introduce topological concepts in the space of infinite words such as openness and the central notion of saturation. In Section 3, we formally introduce our notion of prognosability, and

show the relation to openness and saturation. As a preliminary to our decision procedure for saturation, we dedicate the Section 4 to rational relations between words and to 2-automata. In Section 5, we present the decision procedures for saturation and openness, and analyze their complexity, before we relate our setting back to the classical setting of finitary diagnosis in Section 6. We conclude the contribution with a comparison of our approach with existing work on centralized diagnosis.

## 2 Topologies on the space of infinite words

*Elementary notations and vocabulary* Given an alphabet $\Sigma = \{a, b, l, \ldots\}$, we denote by $\Sigma^*$ (resp. $\Sigma^\omega$) the set of finite (resp. infinite) words over $\Sigma$; we equip the set $\Sigma^*$ with the standard prefix partial order, written $\leq$. We use $u, u', v, \ldots$ (resp. $w, w', w_1, \ldots$) as typical elements of $\Sigma^*$ (resp. $\Sigma^\omega$). For $w \in \Sigma^\omega$, we write $w_k$ for the $k$-th prefix of $w$. A $*$-language (resp. $\omega$-language) is any subset of $\Sigma^*$ (resp. $\Sigma^\omega$)—we will indifferently use "language" and "set"—. We use $B, B', \ldots$ (resp. $L, L', S, \ldots$) for typical $*$-languages (resp. $\omega$-languages). For any $L \subseteq \Sigma^\omega$, let us denote by $L^c$ the complement of $L$, that is $\Sigma^\omega \setminus L$. Given a set $B \subseteq \Sigma^*$, we denote by $B\Sigma^\omega$ the set of words of the form $uw$ with $u \in B$ and $w \in \Sigma^\omega$.

Given any equivalence relation $\sim$ between finite (resp. infinite) words, we denote by $[u]_\sim$ (resp. $[w]_\sim$) the equivalence class of $u$ (resp. $w$).

*Refining Cantor topology* The *Cantor topology* over the set $\Sigma^\omega$ of infinite words is defined as follows: the basic open sets are the sets of the form $B\Sigma^\omega$ where $B \subseteq \Sigma^*$ (see Perrin and Pin 2004, Chapter 3). A set is *closed* if its complement is open. It is *clopen* if it is both open and closed. Clopen sets are of the form $B\Sigma^\omega$ where $B$ is a finite subset of $\Sigma^*$.

We refine the Cantor topology with respect to a fixed equivalence relation $\sim$ over $\Sigma^\omega$, by requiring that the open sets are "saturated" by $\sim$. Formally,

**Definition 1** The $\sim$-*saturation* of an $\omega$-language $L$ is the $\omega$-language $(L)_\sim$ defined by $(L)_\sim := \bigcup_{w \in L} [w]_\sim$. $L$ is $\sim$-*saturated*, whenever $L = (L)_\sim$.

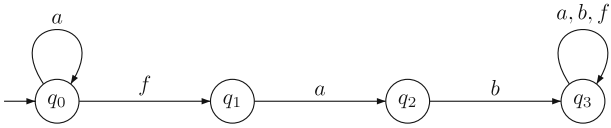
**Lemma 2** *Complement, union and intersection preserve $\sim$-saturation.*

We refer to Section 5.1 for the presentation of an algorithm to decide $\sim$-saturation, under particular assumptions on $\sim$. In the rest of the section, we fix an equivalence relation $\sim$ over $\Sigma^\omega$. The following lemma is a direct consequence of Definition 1.

**Lemma 3** *Let $L$ be a $\sim$-saturated language, and let $w \in L$. Then, $[w]_\sim \subseteq L$.*

We aim now at relativizing the notion of $\sim$-saturation to the set of infinite executions of some (discrete-event) system. For example, Fig. 1 shows a discrete-event system over alphabet $\Sigma = \{a, b, f\}$, whose set of infinite executions is $S := a^\omega \cup a^* fab\, \Sigma^\omega$, that is all possible infinite paths in the depicted structure; notice that, the obtained set is a closed subset of $\Sigma^\omega$.

**Fig. 1** A system




We say that $L$ *is* $\sim$*-saturated in* $S$ whenever $L \cap S = (L \cap S)_\sim \cap S$. Note that $\sim$-saturation and $\sim$-saturation in $\Sigma^\omega$ coincide, and that being $\sim$-saturated in $S$ implies being $\sim$-saturated in $S'$, for any $S' \subseteq S$.

We now combine openess and saturation.

**Definition 4** A language $L \subseteq \Sigma^\omega$ is $\sim$*-open* if it is open and $\sim$-saturated.

By the topology on infinite words (arbitrary union of open sets is an open set, and finite intersection of open sets is an open set) and by Lemma 2, we have:

**Lemma 5** *An arbitrary union of* $\sim$*-open sets is* $\sim$*-open, and a finite intersection of* $\sim$*-open sets is* $\sim$*-open.*

Therefore, we can consider a new topology with $\sim$-open sets. Relativizing $\sim$-openess to a set $S \subseteq \Sigma^\omega$ consists in considering the intersection of an open set with $S$.

**Definition 6** A language $L \subseteq \Sigma^\omega$ is $\sim$*-open in* $S$ if it coincides on $S$ with a $\sim$-open set, i.e. $L \cap S = L' \cap S$ for some $\sim$-open set $L'$.

Notice that a $\sim$-open set in $S$ is not necessarily $\sim$-open. By Definition 6, we trivially have:

**Lemma 7** *An open set is* $\sim$*-saturated in S if, and only if, it is* $\sim$*-open in S.*

We now focus on a family of particular equivalence relations that are obtained from a projection of the alphabet $\Sigma$ onto a particular sub-alphabet $\Sigma_o$ of *observables*. We call them *observational equivalences*.

*The particular case of the observational equivalences* We distinguish a subset $\Sigma_o \subseteq \Sigma$ of elements called *observables*, as opposed to *unobservables*, the elements of the complement of $\Sigma_o$ in $\Sigma$. Typical elements of $\Sigma_o^*$ (resp. $\Sigma_o^\omega$) are $\tau, \tau', \tau_1, \ldots$ (resp. $\pi, \pi', \pi_1$). Words (finite or infinite) over $\Sigma_o$ are *observations*.

We denote by $P : \Sigma \to \Sigma_o$ the canonical alphabet projection, which naturally extends to finite and infinite words. The mapping $P$ transforms any sequence of $\Sigma$ symbols into the sequence obtained by erasing in the former all the symbols not in $\Sigma_o$. The projection $P$ induces an observational equivalence relation $\approx \subseteq (\Sigma^* \times \Sigma^*) \cup (\Sigma^\omega \times \Sigma^\omega)$ which identifies sequences with identical $P$-images. Remark that for any observation $\tau \in \Sigma_o^*$, sets $P^{-1}(\tau)$ are $\approx$-saturated.

Regarding topology, observational equivalences have remarkable properties.

**Lemma 8** *For any* $B \subseteq \Sigma^*$, $(B\Sigma^\omega)_\approx = (B)_\approx \Sigma^\omega$, *where* $(B)_\approx = \bigcup_{u \in B} [u]_\approx$.

*Proof* If $B = \emptyset$ then the equality trivially holds. Assume now, $B \neq \emptyset$ and let $w \in (B\Sigma^\omega)_\approx$, then $w \approx u'w'$ for some $u' \in B$ and $w' \in \Sigma^\omega$. Necessarily there exists $u$ a prefix of $w$ such that $u \approx u'$ which establishes that $w \in (B)_\approx \Sigma^\omega$. Reciprocally, assume $w \in (B)_\approx \Sigma^\omega$. Then $w$ has some prefix $u$ such that $u \approx u'$ for some $u' \in B$. Therefore, $w \approx u'w_1 \in B\Sigma^\omega$ which concludes. □

**Corollary 9** *The $\approx$-open sets are of the form $(B)_\approx \Sigma^\omega$, where $(B)_\approx = \bigcup_{u\in B}[u]_\approx$.*

To illustrate the notions, we consider the alphabets $\Sigma = \{a, b, f\}$ and $\Sigma_o = \{a, b\}$, hence the only unobservable is $f$. The language $\Sigma^* f\Sigma^\omega$ is not $\approx$-open: indeed, $a^\omega \in (\Sigma^* f\Sigma^\omega)_\approx$ and yet $a^\omega \notin \Sigma^* f\Sigma^\omega$, so that $\Sigma^* f\Sigma^\omega$ is not $\approx$-saturated. However, it is $\approx$-open in $S$: indeed, $(\Sigma^* f\Sigma^\omega \cap S)_\approx = a^* fb\,\Sigma^\omega \not\ni a^\omega$, so that $(\Sigma^* f\Sigma^\omega \cap S)_\approx \cap S = \Sigma^* f\Sigma^\omega \cap S$.

We now aim at studying ways to cover a set $L$ with $\approx$-open sets so that prognosis is feasible. To every observation $\tau \in \Sigma_o^*$, we associate two sets $[\,\tau\,] \subseteq \Sigma^*$ and $\lfloor\tau\rfloor \subseteq \Sigma^\omega$, defined by:

$$[\,\tau\,] := P^{-1}(\tau) \cap \Sigma^*\Sigma_o \quad \text{and} \quad \lfloor\tau\rfloor := [\,\tau\,]\Sigma^\omega$$

By definition, $\lfloor\tau\rfloor$ is open. Moreover,

**Lemma 10** *The set $\lfloor\tau\rfloor$ is $\approx$-saturated.*

*Proof* Notice that $([P^{-1}(\tau) \cap \Sigma^*\Sigma_o]\Sigma^\omega)_\approx = (P^{-1}(\tau) \cap \Sigma^*\Sigma_o)_\approx \Sigma^\omega$, by Lemma 8. Now, $(P^{-1}(\tau) \cap \Sigma^*\Sigma_o)_\approx = (P^{-1}(\tau))_\approx \cap (\Sigma^*\Sigma_o)_\approx = P^{-1}(\tau) \cap \Sigma^*\Sigma_o(\Sigma \setminus \Sigma_o)^*$. Therefore, $(P^{-1}(\tau) \cap \Sigma^*\Sigma_o)_\approx \Sigma^\omega = P^{-1}(\tau)\Sigma^\omega \cap \Sigma^*\Sigma_o(\Sigma \setminus \Sigma_o)^*\Sigma^\omega$. Because $\Sigma^*\Sigma_o(\Sigma \setminus \Sigma_o)^*\Sigma^\omega = \Sigma^*\Sigma_o\Sigma^\omega$, we conclude that $(P^{-1}(\tau) \cap \Sigma^*\Sigma_o\Sigma^\omega)_\approx = [P^{-1}(\tau) \cap \Sigma^*\Sigma_o]\Sigma^\omega = \lfloor\tau\rfloor$. □

We extend $\lfloor . \rfloor$ to infinite observations according to $\lfloor\pi\rfloor := \{w \in \Sigma^\omega \mid P(w) = \pi\}$.

**Proposition 11** *Let $L \subseteq \Sigma^\omega$. $L$ is $\approx$-open if, and only if, for every observation $\pi \in \Sigma_o^\omega$ with $L \cap \lfloor\pi\rfloor \neq \emptyset$, there exists $k(\pi) \in \mathbb{N}$ such that $\lfloor\pi_{k(\pi)}\rfloor \subseteq L$.*

*Proof* $\Leftarrow$) is immediate since $L = \bigcup_{w\in L} \lfloor P(w)_{k(P(w))}\rfloor$ and the $\lfloor P(w)_{k(P(w))}\rfloor$'s are $\approx$-open sets, hence their union is a $\approx$-open set by Lemma 5.

$\Rightarrow$) Since $L$ and $\lfloor\pi\rfloor$ are $\approx$-saturated, $L \cap \lfloor\pi\rfloor \neq \emptyset$ implies $\lfloor\pi\rfloor \subseteq L$, by Lemma 3. For the readability of the proof we simply write $k$ instead of $k(\pi)$. Assume that for each $k \geq 0$ there exists $w'_k \in \lfloor\pi_k\rfloor \setminus L$. Since $L$ is open, it is of the form $B\Sigma^\omega$ with $B \subseteq \Sigma^*$. Since $w'_k \notin L$, the $j$-th prefix $(w'_k)_j$ of $w'_k$ is not in $B$, for any $j \in \mathbb{N}$. Because the alphabet $\Sigma$ is finite, we apply Koenig's lemma to the set $\{(w'_k)_j \mid j, k \in \mathbb{N}\}$ and obtain an infinite sequence of elements $u_0 < u_1 < u_2 \ldots$ such that for every $i$, $P(u_i)$ is the $i$-th prefix of $\pi$. Since the sets $\{u_i\}\Sigma^\omega$ are clopen sets, their intersection is closed and therefore contains the limit of the $u_i$'s, say $w'$, which lies outside the set $L$. But $P(w') = \pi$, since each $P(u_i)$ is a prefix of $\pi$, hence $w' \in \lfloor\pi\rfloor$, which contradicts $\lfloor\pi\rfloor \subseteq L$. □

Reconsider the system $S := a^\omega \cup a^* fab\,\Sigma^\omega$, and the language $\Sigma^* f\Sigma^\omega$, seen earlier. Recall that the open set $a^* f\Sigma^\omega$ is not $\approx$-saturated (hence it is not $\approx$-open) because for example, the executions $faf^\omega$, $f^2af^\omega$, $f^3af^\omega$, ... that are $\approx$-equivalent to $af^\omega$ are

missing. However, as established before, $a^* f \Sigma^\omega$ is $\approx$-open in $S$: it is clearly open, and it is also $\approx$-saturated in $S$, because no execution starting with $f^k a$ ($k \in \mathbb{N}$) exists in $S$. We formalize this phenomenon.

For the rest of this section, we fix a system $S \subseteq \Sigma^\omega$. Similarly to what has been done previously, but relativized to system $S$, we associate to any $\tau \in \Sigma_o^*$ a $\approx$-open set in $S$ defined by:

$$\lfloor \tau \rfloor_S := \lfloor \tau \rfloor \cap S \tag{1}$$

**Corollary 12** *Let $L \subseteq \Sigma^\omega$. $L$ is $\approx$-open in $S$ if, and only if, for every observation $\pi \in \Sigma_o^\omega$ with $L \cap \lfloor \pi \rfloor_S \neq \emptyset$, there exists $k(\pi) \in \mathbb{N}$ such that $\lfloor \pi_{k(\pi)} \rfloor_S \subseteq L$.*

*Proof* We simply write $k$ for $k(\pi)$. Assume $L$ is $\approx$-open in $S$. Then $L \cap S = (B\Sigma^\omega)_\approx \cap S$ for some $B \subseteq \Sigma^*$. Let $\pi \in \Sigma_o^\omega$ be such that $L \cap \lfloor \pi \rfloor_S \neq \emptyset$. Since $L \cap S \subseteq (B\Sigma^\omega)_\approx \cap S$, we also have $(B\Sigma^\omega)_\approx \cap \lfloor \pi \rfloor_S \neq \emptyset$. By applying Proposition 11 to the $\approx$-open $(B\Sigma^\omega)_\approx$, $\pi$ has some prefix $\pi_k$ with $\lfloor \pi_k \rfloor \subseteq (B\Sigma^\omega)_\approx$. Hence $\lfloor \pi_k \rfloor_S \subseteq L$.

Reciprocally, assume that for every $\pi \in \Sigma_o^\omega$, $L \cap \lfloor \pi \rfloor_S \neq \emptyset$ implies there exists $k \in \mathbb{N}$ such that $\lfloor \pi_k \rfloor_S \subseteq L$. We want to show that $L$ is of the form $(B\Sigma^\omega)_\approx \cap S$ for some $B \subseteq \Sigma^*$. The candidate for $B$ is $\bigcup_{\pi \in \Sigma_o^\omega} P^{-1}(\pi_k)$.

By Lemma 8, $B\Sigma^\omega$ is $\approx$-saturated, so that $B\Sigma^\omega = (B\Sigma^\omega)_\approx$. It remains to show that $L \cap S = B\Sigma^\omega \cap S$:

- $L \cap S \subseteq B\Sigma^\omega \cap S$: Let $w \in L \cap S$, and let $\pi = P(w)$. Since $w \in P^{-1}(\pi_k)\Sigma^\omega \cap S$ and $P^{-1}(\pi_k) \subseteq B$, $w \in B\Sigma^\omega \cap S$.
- $B\Sigma^\omega \cap S \subseteq L \cap S$: Let $w \in B\Sigma^\omega \cap S$, then $w \in P^{-1}(\pi_k)\Sigma^\omega = \lfloor \pi_k \rfloor$ which by hypothesis is contained in $L$ and we are done. □

## 3 Prognosis of $\omega$-languages

In this section we formalize the problem of prognosing membership in a given $\omega$-languages, in the sense to determine the membership of the infinite execution of the system in a given $\omega$-language, on the basis of its finite observations.

In the rest of the section, we assume two $\omega$-languages $S$, the system language, and $L$, the object of prognosis. A finite (resp. infinite) word $u \in \Sigma^*\Sigma_o$ ($w \in \Sigma^\omega$) is *consistent* with an observation $\tau \in \Sigma^*$ (resp. $\pi \in \Sigma^\omega$) if $u \in [\tau]$ (resp. $w \in \lfloor \pi \rfloor$).

Fix a finite observation $\tau \in \Sigma_o^*$ of the system. First, assume a situation where for every possible infinite continuation $\pi' > \tau$, any concrete execution $w \in S$ consistent with $\tau\pi'$ belongs to $L$: membership in $L$ of the forthcoming infinite system execution can be safely prognosed, by outputting, say, the prognosis "tt". Dually, if every execution consistent with $\tau\pi'$ for any $\pi' > \tau$ does not belong to $L$, we set the prognosis "ff". Otherwise, the prognosis is "$\perp$", thus denoting a confused situation where consistent executions spread inside and outside $L$.

According to the above, we let the *prognoser* be the following function, where we recall that $\lfloor \tau \rfloor_S = P^{-1}(\tau)\Sigma^\omega \cap S$:

$$\begin{aligned} &\textsc{Prog}_L : \Sigma_o^* \to \{\perp, \mathrm{tt}, \mathrm{ff}\} \\ &\textsc{Prog}_L(\tau) := \begin{cases} \mathrm{tt} & \text{if } \lfloor \tau \rfloor_S \subseteq L \\ \mathrm{ff} & \text{if } \lfloor \tau \rfloor_S \cap L = \emptyset \\ \perp & \text{otherwise.} \end{cases} \end{aligned} \tag{2}$$

In the following, we shall omit the subscript $L$ of $\textsc{Prog}_L$, when clear from the context.

By equipping the set $\Sigma_o^*$ with the word-prefix partial order and the domain $\{\bot, \mathrm{tt}, \mathrm{ff}\}$ with the partial order $\bot < \mathrm{tt}, \mathrm{ff}$, it is not difficult to remark that the prognoser is a monotonic function. Informally, it means that a prognosis of the system that is not "$\bot$", hence either "tt" or "ff", will not change in the future whatever the execution of the system becomes.

We finally establish two fundamental theorems that characterize situations where the prognosis value necessarily becomes "tt"; the situation is very similar to the diagnosis of $*$-languages, and the similarities will be considered in details in Section 6. First of all, consider $w \in S$ an infinite execution of the system and some observation length $k$. Since by definition of the prognoser, $\textsc{Prog}(P(w)_k) = \mathrm{tt}$ is equivalent to $\lfloor P(w)_k \rfloor \subseteq L$, we can apply Corollary 12 to $\pi = P(w)$, and state the following.

**Theorem 13** *$L$ is $\approx$-open in $S$ if, and only if, for every $w \in L$, there exists $k(w) \in \mathbb{N}$ such that $\textsc{Prog}(P(w)_{k(w)}) = tt$.*

Note that $\approx$-openness in $S$ only guarantees an eventual outcome "tt" if the actual execution of the system extends to an infinite execution in $L$, whereas nothing can be guaranteed for infinite executions not in $L$: Consider again the system $S := a^\omega \cup a^* fab\,\Sigma^\omega$ of Fig. 1, with $\Sigma = \{a, b, f\}$. Let $\Sigma_o := \{a, b\}$. As we already have seen on page 5, the language $L := a^* f \Sigma^\omega$ is $\approx$-open in $S$. Consider an infinite execution $w$ of the form $a^\ell fab\,w' \in L$; write $\pi' := P(w')$. By letting $k(a^\ell fab\,w') := \ell + 2$, we have $P(a^\ell fab\,w')_{k(a^\ell fab\,w')} = a^\ell ab$, and because $\lfloor a^\ell ab \rfloor_S \subseteq L$, we infer $\textsc{Prog}(a^\ell ab) = \mathrm{tt}$. However, because for $k > 1$ $\lfloor a^k \rfloor_S \cap L \ni a^{k-1} fa$ and $\lfloor a^k \rfloor_S \cap L^c \ni a^k$, we have $\textsc{Prog}(a^k) = \bot$. For the system execution $a^\omega$ the prognoser will never be able to prognose any information but the prognosis "$\bot$".

Actually, in order to prognose execution not in $L$ one needs the following property, which clearly does not apply to the example above ($a^* f \Sigma^\omega$ is not clopen).

**Theorem 14** *$L$ is $\approx$-clopen in $S$ if, and only if, for all $w \in S$, there exists $k(w) \in \mathbb{N}$ such that $\textsc{Prog}(P(w)_{k(w)}) \neq \bot$.*

*Proof* $\Rightarrow$) Let $w \in \Sigma^\omega$. If $w \in L$, because $L$ is $\approx$-open in $S$, we can apply Theorem 13 to infer the existence of $k(w)$. Otherwise $w \in S \setminus L$. Since $L$ is $\approx$-closed in $S$, $L^c$ is $\approx$-open in $S$. We therefore can apply Theorem 13 to $L^c$ and obtain the existence of $k(w)$ such that $(P(w)_{k(w)}) \subseteq L^c$, that is $\textsc{Prog}(P(w)_{k(w)}) = \mathrm{ff}$.

$\Leftarrow$) Consider the partition of $S$ into $L$ and $S \setminus L$. For every $w \in L$, $\textsc{Prog}(P(w)_k) \neq \bot$ is equivalent to $\textsc{Prog}(P(w)_k) = \mathrm{tt}$. By Theorem 13, $L$ is $\approx$-open in $S$. By a similar reasoning, $S \setminus L$ is $\approx$-open in $S$, hence $L$ is $\approx$-closed in $S$. □

As a consequence of Theorem 13, and by analogy with the theory of diagnosis for discrete-event systems, we consider the following definition:

**Definition 15** An $\omega$-language $L \subseteq \Sigma^\omega$ is *prognosable (with respect to $S \subseteq \Sigma^\omega$)* whenever it is $\approx$-open in $S$.

Before studying in Section 5 the decision problems of saturation and of openness for $\omega$-regular languages, which entail the decidability of "prognosability" (by Definition 15), we need to recall elementary notions on rational relations.

## 4 Rational relations

The class of *rational* relations and its subclasses ranging from *recognizable* relations to e.g. *synchronized* relations are of particular interest since they possess acceptors whose e.g. emptiness can be decided. A detailed literature on the topic can be found in Berstel (1979), Staiger (1997), Gire and Nivat (1984), Prieur (2000), and Frougny and Sakarovitch (1993). Also, we assume that the reader is familiar with automata on infinite words, and in particular Büchi automata; a detailed exposition of the subject can be found in Thomas (1990).

For the purpose of this work, we will focus on binary relations only, that is pairs of words over the alphabet $\Sigma$, and we simply call them *relations*. For finite words, a relation is a subset of the Cartesian product $\Sigma^* \times \Sigma^*$. *Rational* subsets of $\Sigma^* \times \Sigma^*$ are the elements of $\mathtt{Rat}$ the least subset of $2^{\Sigma^* \times \Sigma^*}$ such that: (1) every finite subset of $\Sigma^* \times \Sigma^*$ is in $\mathtt{Rat}$, and (2) if $R, R' \in \mathtt{Rat}$, then $R \cup R'$, $RR' \in \mathtt{Rat}$, and (3) if $R \in \mathtt{Rat}$, then $R^* := \bigcup_{i \geq 0} R^i \in \mathtt{Rat}$.

We recall that the following properties are decidable:

- Emptiness: for any rational $\omega$-relation $\rho \subseteq \Sigma^\omega \times \Sigma^\omega$, one can construct nondeterministic Büchi automata which accept the first and the second projection of $\rho$. Now, $\rho$ is empty if and only if either one of the two projections is empty. Büchi automata emptiness is decidable.
- Finiteness: whether a rational $\omega$-relation is finite amounts to checking that the two projections are finite $\omega$-regular languages.

According to the general theory, *recognizable* relations over $\Sigma$ are particular cases of recognizable subsets of a monoid (see Berstel 1979, Chapter III). In the case of the monoid $\Sigma^* \times \Sigma^*$ (for binary relations), we use the following intuitive characterization due to Elgot and Mezei (1965), known as the Mezei's Theorem: a *recognizable* relation $R \subseteq \Sigma^* \times \Sigma^*$ is a finite union of sets of the form $B_i \times B'_i$, where $B_i, B'_i \subseteq \Sigma^*$ are regular $*$-languages. As the monoid $\Sigma^* \times \Sigma^*$ is implicit in this work, we simply write $\mathtt{Rec}$ for the set of recognizable subsets of $\Sigma^* \times \Sigma^*$.

It is well established that $\mathtt{Rec} \subseteq \mathtt{Rat}$. However, because the monoid $\Sigma^* \times \Sigma^*$ is not free, the reciprocal does not hold. Closure properties of the classes $\mathtt{Rat}$ and $\mathtt{Rec}$ are as follows: $\mathtt{Rec}$ is closed under union, intersection and complement, whereas $\mathtt{Rat}$ is closed under union (by definition) but not under intersection in general, therefore neither under complementation. One way to achieve good closure properties while maintaining expressiveness is by mixing the two classes. The following lemma from Berstel (1979, Proposition 2.6, Chapter III) is at the basis of our analysis.

**Lemma 16** *If* $R \in \mathtt{Rat}$ *and* $R' \in \mathtt{Rec}$, *then* $R \cap R' \in \mathtt{Rat}$.

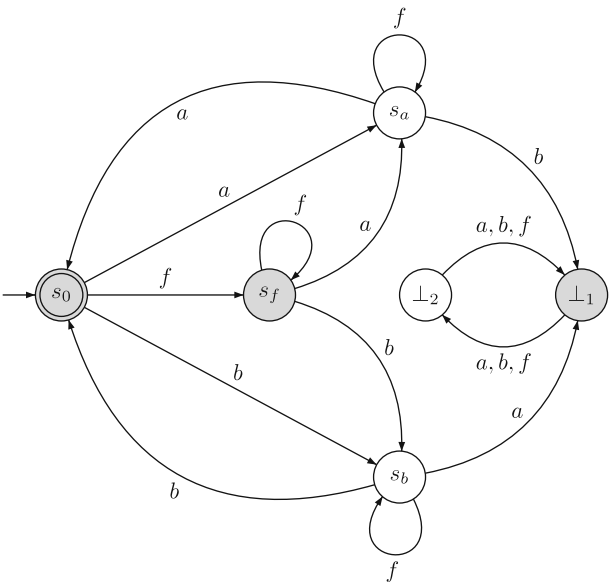
Operationally, rational $\omega$-relations are characterized by *2-automata* (Rabin and Scott 1959).

**Definition 17** A *2-automaton* over alphabet $\Sigma$ is a Büchi automaton $(Q, \Sigma, q_0, \delta, F)$ equipped with a partition $\{Q_1, Q_2\}$ of the state space $Q$ denoting the control states for the first and the second input tape respectively. We write $\Theta = (Q, \Sigma, q_0, \delta, A, Q_1, Q_2)$ this structure.

A 2-automaton reads a pair of words $(w_1, w_2)$ respectively placed on the first and the second input tape; the partition $\{Q_1, Q_2\}$ tells which tape is to be read in the current state. We illustrate the behavior of the 2-automaton in Fig. 2 (states filled in grey are in $Q_1$) characterizing the observational equivalence induced by $\Sigma_o = \{a, b\} \subseteq \Sigma = \{a, b, f\}$ for a pair of input words $(w_1, w_2)$ of the form $w_1 = afbfaaw_1'$ and $w_2 = ffabaffaw_2'$. As the initial state $s_0 \in Q_1$, the automaton first reads $a$ from $w_1$, then moves to state $s_a \in Q_2$ thus remembering that the last observable read on tape 1 was an $a$. It then reads $f$ from $w_2$, which causes a transition back to $s_a$. Eventually, the first $a$ of $w_2$ is read, and since this matches the expected observable, the automaton moves back to $s_0$. After a few more transitions, the automaton is in state $s_0$ and the tapes contain $w_1'$ and $w_2'$ respectively. Assume that $w_1' \not\approx w_2'$, henceforth $w_1 \not\approx w_2$, because say $w_1'$ starts with an $a$ whereas $w_2'$ starts with a $b$: the run then gets trapped into the non-accepting strongly connected component $\{\bot_1, \bot_2\}$.

We remark that any observational equivalence can be characterized by a 2-automaton of size in $O(|\Sigma_o| + 4)$. Also, notice that during the runs of a 2-automaton accepting an observational equivalence relation, the distance between the two heads on the tapes cannot be bounded in general; observational equivalences are not *synchronized* rational relations in the sense (Frougny and Sakarovitch 1993). Nevertheless, their complement are also a rational relation.

**Fig. 2** A 2-automaton



## 5 Algorithms for ≈-saturation and openness

We consider only $\omega$-regular languages, represented by (possibly nondeterministic) Büchi automata. We first analyze a procedure to check the $\approx$-saturation of an $\omega$-regular language $L$ in some $\omega$-regular language $S$.

We next turn to a procedure to decide openness of an $\omega$-regular language $L$ in the Cantor topology, as it would suffice to derive a procedure with the same complexity to check the openness of $L$ in $S$, as long as $S$ is a closed set. Clearly, when $S$ is a closed set, $L$ is open in $S$ if, and only if, $L \cup S^c$ is an open set. Moreover, building the automaton for $L \cup S^c$ can be done efficiently: as the system language $S$ is in general described by a deterministic Büchi automaton, all states of which are accepting (this why $S$ is closed), the Büchi automaton for $S^c$ is constructed with a very straightforward complementation procedure for deterministic Büchi automata. Finally, this new automaton and the nondeterministic automaton for $L$ are merged to get the automaton for $L \cup S^c$ whose size is linear in the size of the automata for $S$ and $L$ respectively.

In the following, we take the convention that a nondeterministic Büchi automaton is represented as a structure $\mathcal{A} = (\Sigma, Q, q_0, \delta, F)$, where as expected $\Sigma$ is the alphabet, $Q$ is the set of states, with a distinguished initial state $q_0$, the transition relation $\delta \subseteq Q \times \Sigma \times Q$, and $F \subseteq Q$ is the set of accepting states.

### 5.1 Deciding ≈-saturation

Let $L(\Theta) \subseteq \Sigma^\omega \times \Sigma^\omega$ denote the language accepted by a 2-automaton $\Theta$ (see Definition 17).

**Theorem 18** *Fix a nondeterministic Büchi automaton $\mathcal{A} = (\Sigma, Q, q_0, \delta, F)$ and a 2-automaton $\Theta$ whose language $L(\Theta)$ is an equivalence relation. The problem of whether $L(\mathcal{A})$ is $L(\Theta)$-saturated is* PSPACE*-complete.*

*Proof* Simply write $L$ for $L(\mathcal{A})$ and $\sim$ for $L(\Theta)$.

i) PSPACE membership: let $R := \sim \cap (L \times L^c)$. Clearly, $R = \emptyset$ if, and only if, $L$ is $\sim$-saturated, as both conditions mean that there are no two words $w_1 \sim w_2$ with $w_1 \in L$ and $w_2 \notin L$. By Mezei's Theorem, the relation $L \times L^c$ is recognizable since $L$ and $L^c$ are $\omega$-regular. Since $\sim$ is rational, so is $R$ (see a variant of Lemma 16 for $\omega$-relations in Gire and Nivat 1984) and a 2-automaton can be effectively constructed for $R$.
We present an algorithm to check emptiness of $R$. Let $\mathcal{B}$ be a nondeterministic Büchi automaton which accepts $L^c$ using $O(2^{|\mathcal{A}| \log |\mathcal{A}|})$ states. This can be constructed following Klarlund (1991) (see also Grädel et al. 2002, Chapter 4). Let $\Theta'$ be the 2-automaton which behaves like $\Theta$ but whose input is componentwise constrained by $\mathcal{A}$ and $\mathcal{B}$ respectively. The 2-automaton $\Theta'$ has $O((|\Sigma_o| + 4) \cdot |\mathcal{A}| \cdot 2^{|\mathcal{A}| \log |\mathcal{A}|})$ states encoded in space $O(\log(|\Sigma_o| + 4)) + \log |\mathcal{A}| + |A| \log |\mathcal{A}|)$. The nondeterministic Algorithm **A1** guesses an accepting run of $\Theta'$, namely a sequence of states $r_0 r_1 \ldots r_i \ldots r_n$ where $r_0$ is an initial state, $r_i$ is an accepting state, and $r_i = r_n$.

**Algorithm A1**

1. Let $r$ be the initial state of $\Theta'$
2. Choose a state $r'$
3. If $r'$ is a successor of $r$, let $r = r'$
   else halt (without accepting)
4. If $r$ is accepting, goto 5 or 2, else goto 2
5. Let $r_A = r$ // guess it is $r_i$
6. Choose a state $r'$
7. If $r'$ is a successor of $r$, let $r = r'$
   else halt (without accepting)
8. If $r = r_A$, accept, else goto 6

Algorithm **A1** can be implemented by a nondeterministic polyspace Turing machine, which concludes since NPSPACE = PSPACE by Savitch's Theorem (Savitch 1970).

ii) PSPACE hardness: let us denote by $\equiv$ the trivial relation $\Sigma^\omega \times \Sigma^\omega$. Given a Büchi automaton $\mathcal{C}$, we reduce the universality problem for $\mathcal{C}$ (whether $L(\mathcal{C}) = \Sigma^\omega$), known to be PSPACE-complete (Sistla et al. 1987), to the $\equiv$-saturation of $L(\mathcal{C})$.
If $L(\mathcal{C}) = \emptyset$, which can be checked linearly in the size of $\mathcal{C}$, then return "no". Otherwise, let $w_1 \in L(\mathcal{C})$. $\mathcal{C}$ is not universal if, and only if, there exists $w_2 \notin L(\mathcal{C})$. Since $w_1 \equiv w_2$ (because $\equiv$ is trivial) this is equivalent to saying that $L(\mathcal{C})$ is not $\equiv$-saturated. □

When an $\omega$-regular system $S$ is considered, Algorithm **A1** can be easily adapted to decide $\sim$-saturation of $L$ in $S$ by checking emptiness of $R$ intersected with the recognizable relation $S \times S$.

**Corollary 19** *The problem of checking $\approx$-saturation in S is PSPACE-complete.*

*Proof* For membership, use the fact that the set of states of the 2-automaton $\Theta'$ can be encoded in space $O(log(|\Sigma_o| + 4)) + log|\mathcal{A}| + |A|log|\mathcal{A}|)$. For hardness, take $S = \Sigma^\omega$ and $\Sigma_o = \emptyset$ and consider the proof for hardness in Theorem 18, as in this case $\approx$ and $\equiv$ match. □

## 5.2 Deciding openness

Although to the best of our knowledge no direct algorithm for checking whether or not the language of a Büchi automaton is open existed so far, openness could still be decided due to the following results. In Landweber (1969) Landweber proposes a polynomial procedure to decide whether the language of a deterministic Muller

automaton (MA), is an open set. MA are sufficiently expressive to cover the class of $\omega$-regular languages, and in contrast to Büchi automata, they can always be assumed deterministic. Basically, a (deterministic) MA $\mathcal{M}$ is given by a finite-state automaton $(Q, \Sigma, q_0, \delta)$ and a distinguished set $\mathcal{F} \subseteq 2^Q$ of *accepting sets*. An infinite word is accepted by $\mathcal{M}$ if along the (unique) run of $\mathcal{M}$ for this input word, the set of states that are visited infinitely often matches some element of $\mathcal{F}$. In an MA $\mathcal{M}$, the set of accepting sets $\mathcal{F}$ consists of nontrivial strongly connected components.

Therefore we can decide the question of whether the language of some nondeterministic Büchi automaton (NBA) is an open set or not by translating it first to a deterministic MA (McNaughton 1966) but with a worst-case exponential "blow up" in the size of the resulting MA. Alternatively, by complementing a given NBA, one can use the procedure described in Alpern and Schneider (1987) which determines whether a language of an NBA, $\mathcal{A}$, is closed (Alpern and Schneider 1987): the method consists of verifying that $\mathcal{A}$ and its *closure* (obtained by making all co-reachable states accepting) denote the same language. According to Sistla (1994) this verification problem is PSPACE-complete in general, and it is solvable in linear time if $\mathcal{A}$ is deterministic (Kupferman and Vardi 2001). However, the preliminary complementation of the NBA may have an exponential "blow up" as well.

In what follows, we state a direct algorithmic solution for determining whether or not the accepted language of an NBA is an open set. Moreover, this algorithm will use polynomial space with respect to the input NBA, and thus lowers the above implied EXPSPACE upper bound on the openness-problem to PSPACE.

Let us first exhibit the main idea of the algorithm, before we describe it in more detail. The intuition is now dual to the method described in Alpern and Schneider (1987) for checking closeness of an NBA-language; that is, instead of taking the closure of a given NBA, we take what we call the *suffix closure* of an NBA, and then compare whether or not the accepted language of that NBA remains the same. For a given NBA $\mathcal{A} = (\Sigma, Q, q_0, \delta, F)$ the suffix closure simply adds to the set $F$ of accepting states those states which are reachable from a nontrivial accepting strongly connected component in $\mathcal{A}$, by which we mean a cycle that contains a state from $F$ and which can be reached infinitely often upon some infinite input word to the automaton.

**Definition 20** Let $\mathcal{A} = (\Sigma, Q, q_0, \delta, F)$ be an NBA. The *suffix closure* of $\mathcal{A}$ is the NBA $\mathrm{scl}(\mathcal{A}) := (\Sigma, Q, q_0, \delta, F')$, where $F'$ is the set of states that can be reached from a nontrivial accepting strongly connected component in the graph $\mathcal{A}$, that is a strongly connected component $C$ such that with $C \cap F \neq \emptyset$.

Let us also introduce the notion of a *good prefix* which is closely related to openness of a language.

**Definition 21** Let $L \subseteq \Sigma^\omega$ and $w \in L$. A prefix $u < w \in \Sigma^*$ is a *good prefix*, if $\{u\}\Sigma^\omega \subseteq L$.

An alternative way to define openness of a language $L \subseteq \Sigma^\omega$ is thus to say that all elements in $L$ have a finite prefix which is a good (possibly empty) prefix.

**Theorem 22** *There exists a PSPACE procedure for deciding whether the language accepted by an NBA is an open set.*

The proof of this theorem will be given by showing Lemmas 23 and 24, which can be used to check whether or not the language by a given automaton is open using only polynomial space.

We assume in Lemma 23 that the NBA is *pruned*, in the sense of Alpern and Schneider (1987), where all transitions from which there exists no path in the state graph to an accepting strongly connected component have been removed; notice that pruning an automaton can be done in polynomial time.

**Lemma 23** *Let $\mathcal{A}$ be a pruned NBA. Then, $\mathcal{L}(scl(\mathcal{A})) = \mathcal{L}(\mathcal{A})$ if $\mathcal{L}(\mathcal{A})$ is an open set.*

*Proof* It follows almost immediately from Definition 20 that $\mathcal{L}(\mathcal{A}) \subseteq \mathcal{L}(\mathrm{scl}(\mathcal{A}))$ holds, because when the automaton is pruned we have $F \subseteq F'$.

In proving that $\mathcal{L}(\mathrm{scl}(\mathcal{A})) \subseteq \mathcal{L}(\mathcal{A})$ also holds, we only need to consider the case where $F$ is a proper subset of $F'$.

Assume that $\mathcal{L}(\mathcal{A})$ is open. As pointed out above, $\mathcal{L}(\mathcal{A})$ is an open set if, and only if, all words $w \in \mathcal{L}(\mathcal{A})$ have a good prefix. This implies that every word $w \in \mathcal{L}(\mathcal{A})$ has a good prefix $u$ and a run $\rho_u$ in the automaton over this prefix which leads to a state $q_{u,C}$, also called the *entry state*, of a nontrivial accepting strongly connected component $C$. Two cases arise: either (i) any such $C$, if interpreted by itself as an NBA with initial state $q_{u,C}$, is universal, i.e., it accepts $\Sigma^\omega$, in which case we are done as marking additional states does not add new words to the accepted language, or (ii) there exists a component $C$ such that the language accepted by $C$, if interpreted as an NBA with initial state $q_{u,C}$, written $\mathcal{L}(C, q_{u,C})$, and transitions between states in $C$, is not equal to $\Sigma^\omega$. Note that in the following, if for some $C$ there exist more than one reachable entry states $\{q^1_{u,C}, \ldots, q^m_{u,C}\} \subseteq \delta(q_0, u)$ thus giving rise to $m$ initial states, we have the choice of either interpreting the corresponding NBA resulting from $C$ as having $m$ nondeterministic initial states, $(C, q^1_{u,C}, \ldots, q^m_{u,C})$, or alternatively, as $m$ separate components that are structurally equivalent, i.e., each having the same transitions, but only one initial state, $(C, q^i_{u,C})$ where $i \leq m$, respectively.

By this argument, we now assume that each strongly connected component $C$, if interpreted as an NBA, has exactly one initial state. In case (ii) above, since $u$ is a good prefix, there must exist $n \geq 2$ components $C_{i \leq n}$ with reachable states $q_{u,i} \in \delta(q_0, u)$, respectively, such that

$$\bigcup_{C_i}^{n} \mathcal{L}(C_i, q_{u,i}) = \Sigma^\omega.$$

Again, if we mark now additional states which are reachable from these components as accepting, then this does not add new words to the accepted "overall language", $\bigcup_{C_i}^{n} \mathcal{L}(C_i, q_{u,i})$.

Hence, $\mathcal{L}(\mathrm{scl}(\mathcal{A})) \subseteq \mathcal{L}(\mathcal{A})$ for cases (i) and (ii), which concludes. □

Before we prove that "the other direction" of the above lemma holds as well, let us consider without loss of generality *reachable* and *complete* NBAs from this point forward.

Let $\mathcal{A} = (\Sigma, Q, q_0, \delta, F)$ be an NBA. $\mathcal{A}$ is *reachable* if for all $q \in Q$, there is a path from $q_0$ to $q$. $\mathcal{A}$ is *complete* if $\delta(q, a) \neq \emptyset$, for all $q \in Q$ and $a \in \Sigma$.

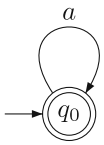



**Fig. 3** An "incomplete" NBA $\mathcal{A}$ with $\mathcal{L}(\mathcal{A}) = a^\omega$

An NBA $\mathcal{A} = (\Sigma, Q, q_0, \delta, F)$ can be *completed* by enriching $Q$ and $\delta$ as follows. We set $Q := Q \cup \{\dagger\}$, where the $\dagger$-state is a "trap", and for all $q \in Q \backslash \{\dagger\}$, for which $\delta(q, a)$ is not defined, we add a transition $(q, a, \dagger)$ to $\delta$. Finally we add the looping transition $(\dagger, a, \dagger)$ for every $a \in \Sigma$. Clearly, automata completion is a polytime operation in the size of $\mathcal{A}$, consisting of adding at most $(|Q| + 1) \times |\Sigma|$ new transitions. It is well known that automata completion preserves the accepted language.

To see why completion is important, consider the NBA $\mathcal{A}$ depicted in Fig. 3, where $\mathcal{L}(\mathcal{A}) = a^\omega$. It is easy to see that $\mathcal{L}(\mathcal{A})$ is not an open set in $\{a, b\}^\omega$. Further, it is easy to see that without completion we would have $\text{scl}(\mathcal{A}) = \mathcal{A}$ and, consequently, $\mathcal{L}(\text{scl}(\mathcal{A})) = \mathcal{L}(\mathcal{A})$ which then violates our intuition that the other direction of Lemma 23 also holds. Now consider $\mathcal{A}'$, the completion of $\mathcal{A}$, as depicted in Fig. 4, for which we obviously have $\mathcal{L}(\text{scl}(\mathcal{A}')) \neq \mathcal{L}(\mathcal{A}')$.

**Lemma 24** *Let $\mathcal{A}$ be a reachable and complete NBA. Then, $\mathcal{L}(\mathcal{A})$ is an open set if $\mathcal{L}(scl(\mathcal{A})) = \mathcal{L}(\mathcal{A})$.*

*Proof* Without loss of generality, we can assume that (*) the only rejecting non trivial strongly connected component which cannot reach an accepting state is the one with the single state $\dagger$ and a universal loop.

Assume $\mathcal{L}(\mathcal{A})$ is not open. There must exist a word $w \in \mathcal{L}(\mathcal{A})$ which has no good prefix. By definition of a good prefix, there exists a run in $\mathcal{L}(A)$ on $w$ which visits $n \geq 1$ nontrivial accepting but not universally accepting components, if interpreted as NBAs in the above sense, infinitely often (but not a universal one, or it would have a good prefix, which we ruled out).

Fact: As the $n$ components are not universal each, they have either a transition to the $\dagger$ state, or outside the component and to another state of $\mathcal{A}$ (from which, there must then also be transitions back into an accepting component because of (*).

Let us call $u$ the finite prefix of $w$ which lead to the $n$ components initially. Because of the Fact outlined before, there exists some finite suffix of $u$, call it $v \in \Sigma^*$, such that $\delta(q_0, uv)$ leads outside of all $n$ components, or $u$ would be a good prefix for $w$ which we have ruled out. To leave all $n$ components, can mean two things: (i) either we reached only a $\dagger$ state in which case we are done as for all $w' \in \Sigma^\omega$ we then have $uvw' \notin \mathcal{L}(\mathcal{A})$ but in $\mathcal{L}(scl(\mathcal{A}))$, or (ii) we reached at least one non-$\dagger$ state in $\mathcal{L}(\mathcal{A})$

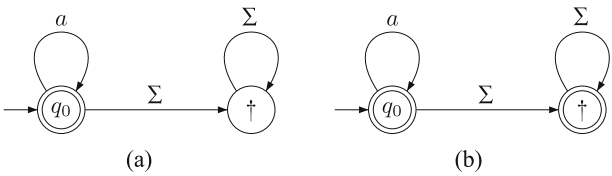



**Fig. 4** A complete NBA $\mathcal{A}'$ with $\mathcal{L}(\mathcal{A}') = a^\omega$ and its suffix closure. **a** $\mathcal{A}'$. **b** scl($\mathcal{A}'$)

on this $v$. In this case, there must exist some word $w' \in \Sigma^\omega$, such that $uvw' \notin \mathcal{L}(\mathcal{A})$, or else $u$ would already have been a good prefix, i.e., it was sufficient to reach the $n$ components as the only way out of them leads to inevitable acceptance.

However, by definition of $scl(\mathcal{A})$, all states after an accepting component are also accepting. Hence we have $uvw' \in \mathcal{L}(scl(\mathcal{A}))$, which means $\mathcal{L}(\mathcal{A}) \neq \mathcal{L}(scl(\mathcal{A}))$. □

To our knowledge, no lower bound for the openness problem of the accepted language of an NBA has been established.

## 6 On the diagnosis of $*$-languages

Traditionally, a discrete-event system is modeled by a finite-state deterministic $\Sigma$-labeled transition system $\mathcal{S}$. From the system $\mathcal{S}$, we naturally derive the $\omega$-language $S \subseteq \Sigma^\omega$ obtained by interpreting $\mathcal{S}$ as a Büchi automaton (all states of which are accepting); this language is an instance of the system languages we have considered so far in the previous sections—in particular, it is a closed subset of $\Sigma^\omega$—, and is $\omega$-regular because $\mathcal{S}$ is finite-state.

Hypotheses on the set $S$ are actually many as it aims at denoting the observable behavior of real systems. Among the classic hypothesis, the *observability* of the system is often put forward: it means that any execution in $S$ contains infinitely many observables. Actually, it is worthwhile noting that the observability assumption changes the topological properties of $S$, which is no longer closed in general.

The classic framework for the theory of diagnosis, originally considered by Sampath et al. (1996), deals with finite words. We therefore introduce the $*$-language $\bar{S}$ formed by the finite prefixes of words in $\mathcal{S}$; it is also characterized by the $*$-language recognized by the structure $\mathcal{S}$ interpreted as a finite-state automaton, all states of which are final; $\bar{S}$ is $*$-regular by construction.

Following the general diagnosis setting of Jeron et al. (2006), a *supervision pattern* $\Omega$ is a regular $*$-language of the form $B\Sigma^*$ (with $B \subseteq \Sigma^*$), that is an open set in the standard topology on $\Sigma^*$, (Staiger 1997). For $\tau \in \Sigma_o^* \setminus \{\epsilon\}$, let $[\,\tau\,]_{\bar{S}} := [\,\tau\,] \cap \bar{S} \cap \Sigma^*\Sigma_o$ be the set of words which are consistent with the non-empty observation $\tau$ and which end with an observable.

We define the *diagnoser* associated to $\bar{S}$ and $B\Sigma^*$ as the function:

$$\begin{aligned} \textsc{Diag}_{B\Sigma^*} : \quad & \Sigma_o^* \to \{\bot, \mathrm{tt}, \mathrm{ff}\} \\ \tau \mapsto & \begin{cases} \mathrm{tt} & \text{if } [\,\tau\,]_{\bar{S}} \subseteq B\Sigma^* \\ \mathrm{ff} & \text{if } [\,\tau\,]_{\bar{S}} \cap B\Sigma^* = \emptyset \\ \bot & \text{otherwise.} \end{cases} \end{aligned}$$

We recall from Jeron et al. (2006) the *diagnosability* property of the language $B\Sigma^*$ with respect to $\bar{S}$. This property guarantees the existence of some bound $N \in \mathbb{N}$ such that for every execution $u \in \bar{S} \cap B\Sigma^*$, and every maximal sequence of increasing observations $P(u) = \tau_1 < \tau_2 < \ldots$, such that $[\,\tau_i\,]_{\bar{S}} \neq \emptyset$, for all $i$, there exists $k \geq 1$ such that $\textsc{Diag}_{B\Sigma^*}(\tau_k) = \mathrm{tt}$. Informally, diagnosability witnesses the "usefulness" of the diagnoser when facing a faulty execution of the system.

We compare the approach of Jeron et al. (2006) and the one of Section 3 by relating the diagnosability of $B\Sigma^*$ and the prognosability of $B\Sigma^\omega$.

**Theorem 25** *Let $\mathcal{S}$ be a discrete-event system over $\Sigma$ (whose $\omega$-language is $S$ and whose $*$-language is $\bar{S}$), and let $B\Sigma^*$ be a supervision pattern. Then, $B\Sigma^*$ is diagnosable with respect to $\bar{S}$ if, and only if, $B\Sigma^\omega$ is prognosable with respect to $S$.*

*Proof* We first establish some facts. By Definition 15, $B\Sigma^\omega$ is prognosable with respect to $S$ whenever it is $\approx$-open in $S$. Since, $B\Sigma^\omega$ is open, by Lemma 7, this is equivalent to say that $B\Sigma^\omega$ is $\approx$-saturated in $S$.

Additionally, diagnosability fails whenever one can find arbitrarily long pairs of words which are observationally equivalent but which do not agree on membership in $B\Sigma^*$ (see Jiang et al. 2001; Yoo and Lafortune 2002; Jeron et al. 2006). In fact, if infinitely many such pairs exist, by applying Koenig's lemma on the finite-degree graph $\mathcal{S}$ (whose set of infinite paths is $S$), one can exhibit two infinite paths $w_1, w_2 \in S$ such that (a) $w_1 \approx w_2$, and (b) no prefix of $w_1$ belongs $B\Sigma^*$ whereas almost all prefixes of $w_2$ do. Relating diagnosis and prognosis is then natural:

$\Leftarrow$) Assume $B\Sigma^*$ is not diagnosable with respect to $\bar{S}$; hence there exists $w_1, w_2 \in S$ such that (a) and (b) hold, which entails that $B\Sigma^\omega$ is not $\approx$-saturated in $S$, with the counter example $w_1$ and $w_2$.

$\Rightarrow$) Assume $B\Sigma^\omega$ is not $\approx$-saturated in $S$. Therefore, there exists a pair of infinite executions $w_1, w_2 \in S$ such that (a) and (b) hold. Pick a finite prefix of $w_1$, $u \in B\Sigma^* \cap \Sigma^*\Sigma_o$. Now, choose an infinite sequence of increasing observations $\{\tau_j\}_j$, with $P(u) = \tau_1$—this is always possible for an observable system $S$. By construction $P(u) = \tau_1 < \tau_2 < \ldots$, and yet $\textsc{Diag}_{B\Sigma^*}(\tau_k) = \bot$ for all $k$, since each $[\,\tau_k\,]_{\bar{S}}$ contains a some prefix of $w_2$ which by assumption on $w_2$ does not. □

By Theorem 25, we benefit from the infinitary setting to easily infer the following: by closure properties of open sets and Lemma 2, we conclude that diagnosability is closed under arbitrary unions and finite intersections, but not under complementation in general, as openness is not preserved. Note however that complementation always preserves diagnosability of finite supervision pattern $\Omega$, by applying Theorem 14 to the clopen set $\Omega\Sigma^\omega$.

Importantly, whereas the diagnoser focuses only on what has happened so far, the prognoser can also foresee: in particular, it has a chance to switch to value "tt" earlier than the diagnoser by detecting the inevitability of a forthcoming execution belonging in $B\Sigma^\omega$ (in the sense of Jiang and Kumar 2004), in a spirit close to prediction. This quantitative kind of comparison between the diagnoser, the prognoser, and the predictor (see Jéron et al. 2008) is out of the scope of the paper, but draws very interesting future lines for research.

We now turn to the comparison of Algorithm **A1** to decide $\approx$-saturation (in $S$) and the standard algorithm to decide diagnosability, as originally proposed by Jiang et al. (2001). We informally recall this algorithm, according to its generalization in Jeron et al. (2006). The central object is a graph called the *twin plant* whose paths denote pairs of $\approx$-equivalent words, and where some vertices are marked. The twin plant construction highly relies on a synchronous product of automata: in this synchronous product, a vertex is *confusing* if in the pair of states it corresponds to, only the first state is final. The twin plant is build in three steps as done by the function $\mathbf{TP}(\mathcal{S}, \theta)$ in Algorithm **A2**.

It is not difficult to see that the existence of a reachable cycle in $\mathbf{TP}(\mathcal{S}, \theta)$ which contains confusing vertices witnesses a counter-example of the $\approx$-diagnosability of

$\Omega$ with respect to $\bar{S}$. This leads to the Algorithm **A2**. In fact Algorithm **A2** is a specialisation of Algorithm **A1** for input languages that are open (i.e. of the form $\Omega\Sigma^\omega$).

**Algorithm A2** // Diagnosability

**Function TP**$(\mathcal{S}, \theta)$ // Twin plant construction
**Inputs:** two finite automata $\mathcal{S}$ and $\theta$ (the latter one represents the language $\Omega$ and is assumed deterministic)
**Outputs:** a graph (the twin plant)
1) Build the product automaton $\mathcal{S}_\theta := \mathcal{S} \times \theta$ (a state is final if its second component is final);
2) Abstract away from unobservables in $\mathcal{S}_\theta$ by replacing every sequence of transitions carrying a word of $\Sigma^*\Sigma_o$ by a single transition carrying the unique observable of this word. This yields $\mathrm{OBS}(\mathcal{S}_\theta)$.
3) Return the graph $\mathrm{OBS}(\mathcal{S}_\theta) \times \mathrm{OBS}(\mathcal{S}_\theta)$,where confusing vertices are those whose first component is final whereas the second one is not.

**Inputs:** two finite automata $\mathcal{S}$ and $\theta$ (the latter one represents the language $\Omega$ and is assumed deterministic)
**Outputs:** "$\Omega$ is not $\approx$-diagnosable with respect to $\bar{S}$" if the graph **TP**$(\mathcal{S}, \theta)$ contains a cycle of confusing vertices, "$\Omega$ is $\approx$-diagnosable with respect to $\bar{S}$" otherwise.

The differences between **A1** and **A2** are the following: On the one hand, "$\approx$-saturation in S" is a PSPACE-complete problem (Corollary 19) and Algorithm **A1** is optimal. On the other hand, Algorithm **A2** is quadratic, by searching a cycle in the graph **TP**$(\mathcal{S}, \theta)$ whose size is in $O((|\mathcal{S}||\theta|)^2)$.

Although the complexity of **A2** seems considerably lower in that case, the assumption that $\theta$ is deterministic is very strong (it hides an exponential time preprocessing procedure to determinize a finite automaton). The twin plant approach hence solves fortunate instances of the saturation problem where rejection of a word by the automaton of $\Omega$ is witnessed by a single run. Algorithm **A2** would become "incomplete" for arbitrary nondeterministic automata $\theta$: a cycle of confusing vertices would not characterize a pair of words $(w_1, w_2)$ where $w_1 \approx w_2$, $w_1 \in \Omega\Sigma^\omega$, and $w_2 \notin \Omega\Sigma^\omega$. Because of nondeterminism, paths in **TP**$(\mathcal{S}, \theta)$ denote only pairs of runs $(\rho_1, \rho_2)$ over words $(w_1, w_2)$; and in general the fact that $\rho_2$ is not accepting does not imply $w_2 \notin \Omega\Sigma^\omega$, unless $\rho_2$ is the unique run. This would be the case if $L$ is a language whose complement is accepted by a deterministic Büchi automaton.

This last remark leads us to propose Algorithm **A3** as an extension of Algorithm **A2** from the class of open languages to the class of languages whose complement is deterministic Büchi definable. This class is characterized in the Borel hierarchy as $\Sigma_2$ which contains sets obtained by a countable union of closed sets. Membership in $\Sigma_2$ is decidable (Perrin and Pin 2004, Chapter I, Proposition 7.10). For this strictly larger class of languages, confusing cycles become cycles of the form $\{(q_1, q'_1), (q_2, q'_2), \ldots, (q_k, q'_k)\}$ where $q_i$ is accepting and all the $q'_j$'s are rejecting. Because open sets are strictly contained in $\Sigma_2$ (Perrin and Pin 2004, Chapter III,

Proposition 2.9), Algorithm **A3** is a true extension of Algorithm **A2**, but is still quadratic, although it solves instances of the "$\sim$-saturation" problem (where $\sim$ is an observational equivalence and $L \in \Sigma_2$).

## 7 Conclusions

We gave a topological characterization of the prognosis of $\omega$-languages and related it to the diagnosis of $*$-languages. We established that, in the classical setting of diagnosis, the concept of prognosability corresponds to diagnosability (Jiang et al. 2001) and prediagnosability (Jiang and Kumar 2004). In the infinitary setting, we could show that prognosability of an $\omega$-language corresponds to this language being an open set in the Cantor topology which is moreover saturated for the observational equivalence under consideration. Both are classical mathematical concepts which enabled us to study the complexity of the underlying decision problems. We have shown that, if the system language and the $\omega$-language in question are represented by a Büchi automaton, openness and saturation are decidable in PSPACE in the size of the automata, and consequently that deciding prognosability of an $\omega$-language (represented by a Büchi automaton) is a PSPACE-complete problem.

These results are not only interesting in an infinitary setting as we have shown in Section 6 that, given some $*$-language of the form $B\Sigma^*$ and some system represented by the language $S$, $B\Sigma^*$ is diagnosable with respect to (the prefixes of elements of) $S$ if, and only if, $B\Sigma^\omega$ is prognosable with respect to $S$. As such, we have shown that the standard algorithm for checking diagnosability (Jiang et al. 2001) is a particular case of the saturation problem corresponding to languages that are open sets.

Hence, the presented setting not only provides algorithms and decision procedures for the analysis of infinite executions of partially observed systems, but gives us new insights into the classical, finitary setting as well. Moreover, it allows us to relate the different notions that currently exist in the diagnosis of discrete event systems in a uniform setting with well-defined mathematical concepts.

## References

Alpern B, Schneider FB (1987) Recognizing safety and liveness. Distrib Comput 2:117–126
Berstel J (1979) Transductions and context-free languages. Teubner Studienbücher, Stuttgart
Elgot CC, Mezei JE (1965) On relations defined by generalized finite automata. IBM J Res Develop 9:47–68
Frougny C, Sakarovitch J (1993) Synchronized rational relations of finite and infinite words. Theor Comput Sci 108(1):45–82
Gire F, Nivat M (1984) Relations rationnelles infinitaires. Calcolo 21(2):91–125
Grädel E, Thomas W, Wilke T (eds) (2002) Automata, logics, and infinite games: a guide to current research (outcome of a Dagstuhl seminar, February 2001). Lecture Notes in Computer Science, vol 2500. Springer, New York
Jeron T, Marchand H, Pinchinat S, Cordier M-O (2006) Supervision patterns in discrete event systems diagnosis. In: 8th workshop on discrete event systems, WODES'06, Ann Arbor
Jéron T, Marchand H, Genc S, Lafortune S (2008) Predictability of sequence patterns in discrete event systems. In: IFAC World Congress, Seoul
Jiang S, Kumar R (2004) Failure diagnosis of discrete event systems with linear-time temporal logic fault specifications. IEEE Trans Automat Contr 49(6):934–945
Jiang S, Huang Z, Chandra V, Kumar R (2001) A polynomial time algorithm for diagnosability of discrete event systems. IEEE Trans Automat Contr 46(8):1318–1321

Klarlund N (1991) Progress measures for complementation of omega-automata with applications to temporal logic. In: FOCS. IEEE, pp 358–367
Kupferman O, Vardi MY (2001) Model checking of safety properties. Form Methods Syst Des 19(3):291–314
Landweber LH (1969) Decision problems for omega-automata. Math Syst Theory 3(4):376–384
McNaughton R (1966) Testing and generating infinite sequences by a finite automaton. Inf Control 9:521–530
Peled D, Wilke T, Wolper P (1998) An algorithmic approach for checking closure properties of $\omega$-regular languages. Theor Comp Sci 195(2):183–203
Perrin D, Pin J-E (2004) Infinite words, automata, semigroups, logic and games. Elsevier, Amsterdam
Prieur C (2000) Fonctions rationnelles de mots infinis et continuté. Thèse de Doctorat, Univ. Paris 7
Rabin MO, Scott D (1959) Finite automata and their decision problems. IBM J Res Develop 3:114–125
Sampath M, Sengupta R, Lafortune S, Sinaamohideen K, Teneketzis D (1995) Diagnosability of discrete event systems. IEEE Trans Automat Contr 40(9):1555–1575
Sampath M, Sengupta R, Lafortune S, Sinaamohideen K, Teneketzis D (1996) Failure diagnosis using discrete event models. IEEE Trans Control Syst Technol 4(2):105–124
Savitch WJ (1970) Relationships between nondeterministic and deterministic tape complexities. J Comput Syst Sci 4:177–192
Sistla AP (1994) Safety, liveness and fairness in temporal logic. Form Asp Comput 6(5): 495–512
Sistla AP, Vardi M, Wolper P (1987) The complementation problem for Buchï automata with applications to temporal logic. Theor Comp Sci 49:217–237
Staiger L (1997) $\omega$-languages. In: Rozenberg G, Salomaa A (eds) Handbook of formal languages, vol 3: beyond words, chapter 10. Springer, New York, pp 339–388
Thomas W (1990) Infinite trees and automaton definable relations over $\omega$-words. In: Proc. STACS 90, Rouen, LNCS 415. Springer, New York
Yoo T, Lafortune S (2002) Polynomial-time verification of diagnosability of partially-observed discrete-event systems. IEEE Trans Automat Contr 47(9):1491–1495

**Andreas Bauer** is currently a Senior Research Engineer at National ICT Australia (NICTA), and an adjunct Research Fellow at the Australian National University. Prior to working at NICTA, he was a full-time Research Fellow in the Logic and Computation group of the Australian National University (2007–2009). Andreas obtained his PhD from the Technische Universität München, Germany, where he worked as a Research Assistant in the Software and Systems Engineering group (2003–2007). His thesis was on a temporal logic based framework for monitoring and diagnosing distributed reactive systems. For more information, visit his homepage at http://users.rsise.anu.edu.au/~baueran/.

Since September 1994, **Sophie Pinchinat** has been an Associate Professor in Computer Science at the University of Rennes 1 in France, and an active research partner of the IRISA research center (Institute for Computer Science and Stochastic Systems), which comprises INRIA, CNRS, University and INSA staff. From 1999 to 2001, she was seconded to INRIA as a research scientist without teaching load, while remaining at the IRISA. Sophie focused first on Programming Environments for Real-Time Applications and since 2000, on System Synthesis and Supervision Scenarios. She received a Marie Curie International Fellowship to visit the Australian National University from August 2006 until July 2007. Currently, she is the Assistant Director of the Computer Science Department of University of Rennes. For more information, visit her homepage at http://www.irisa.fr/prive/Sophie.Pinchinat/.