Optimal Constructions for Active Diagnosis

Stefan Haar, Serge Haddad, Tarek Melliti, and Stefan Schwoon

July 23, 2013

Research report LSV-13-12

Laboratoire Spécification & Vérification

École Normale Supérieure de Cachan
61, avenue du Président Wilson
94235 Cachan Cedex France

Optimal Constructions for Active Diagnosis*

Stefan Haar[1], Serge Haddad[1], Tarek Melliti[2], and Stefan Schwoon[1]

Address [1] LSV (CNRS & ENS Cachan) & INRIA, France
[2] IBISC (Univ Evry Val-Essonne), France

Abstract The task of *diagnosis* consists in detecting, without ambiguity, occurrence of faults in a partially observed system. Depending on the degree of observability, a discrete event system may be *diagnosable* or not. *Active* diagnosis aims at controlling the system in order to make it diagnosable. Solutions have already been proposed for the active diagnosis problem, but their complexity remains to be improved. We solve here the active diagnosability decision problem and the active diagnoser synthesis problem, proving that (1) our procedures are optimal w.r.t. to computational complexity, and (2) the memory required for the active diagnoser produced by the synthesis is minimal. We then focus on the delay between the occurrence of a fault and its detection by the diagnoser. We construct a memory-optimal diagnoser whose delay is at most twice the minimal delay, whereas the memory required for a diagnoser with optimal delay may be highly greater.

1 Introduction

In monitoring discrete event systems, one of the central tasks is that of *diagnosis*: Given a finite labeled transition system $\mathcal{A}$ (also called "plant") whose events are partially observable, our task is to decide – based on the stream of observation labels – whether or not particular unobservable events, called *faults*, have occurred. More precisely, the system is considered k-diagnosable iff at most k events after the occurrence of a fault, the observation is sufficient to detect that occurrence with certainty, i.e. all possible system runs compatible with the partial observation collected so far are faulty. The system $\mathcal{A}$ is *diagnosable* iff there exists $k \geq 1$ such that $\mathcal{A}$ is k-diagnosable. As the system may be insufficiently observable, or the observation not discriminating enough, *diagnosability* verification has received considerable attention since the seminal paper by Sampath et al [11]; see also [4, 3]. Those works construct a dedicated deterministic version of the original plant, a so-called *diagnoser*; the absence of indeterminate cycles in this auxiliary automaton is equivalent to diagnosability.

On the other hand, once a system has been shown to be undiagnosable - in a sense that we will formalize later - several actions can follow, such as complete redesign of the system, or adding further sensors to enhance observability. Sampath et al [10] have initiated a different approach, that of *active diagnosis*: if the given plant $\mathcal{A}$ is not diagnosable, synthesize a partial-observation controller $\mathcal{C}$ that, while letting the system live, forces the behavior of $\mathcal{A}$ to stay within a diagnosable subset of its behaviors (or, equivalently, such that the controlled plant $\mathcal{A}_C$ is diagnosable). The pair consisting of the controller and the diagnoser is called an *active diagnoser*. Later, Chanthery and Pencolé [5] have proposed a planning-based approach via a twin plant construction.

*This work has been supported by project ImpRo ANR-2010-BLAN-0317 and the European Union Seventh Framework Programme FP7/2007-2013 under grant agreement 257462 HYCON2 NOE.

Our contributions In this paper, we follow the approach of Sampath et al [10], but via a different method based on automata and game theory. This allows us to improve the construction of diagnosers and moreover establish complexity results, which were not treated in previous work:

1. We build a deterministic Büchi automaton that accepts the sublanguage of infinite *unambiguous* observable sequences, i.e. those that are either (i) triggered by a set of correct runs or (ii) triggered by a set of faulty runs. Its size is upper-bounded by $2^{\mathcal{O}(n)}$ where n is the number of states, which is better than all previous constructions. In addition we show the optimality of our construction proving that there is a family of systems for which any corresponding deterministic Büchi automaton must have a size in $2^{\Omega(n)}$.

2. Based on these Büchi automata, we design a Büchi game; a winning strategy for it yields an active diagnoser for the system, and vice versa. We thus solve the active diagnosis problem by deciding whether there exists a winning strategy, and the synthesis problem by giving an active diagnoser associated with a positional strategy. The size of the active diagnoser is singly exponential w.r.t. the size of the system, while that of [10] is doubly exponential. We also prove that the decision procedure is EXPTIME-complete and that the synthesis procedure is optimal w.r.t. the number of states of the active diagnoser (still in $2^{\mathcal{O}(n)}$).

3. We then study the delay between a fault and its detection by an active diagnoser. We first present a family of systems for which a minimal-delay diagnoser must have $2^{\Omega(n \log(n))}$ states. However, refining our earlier construction yields an active diagnoser with size $2^{\mathcal{O}(n)}$, whose delay is at most twice the minimal possible delay. In addition, we sketch the construction of a minimal-delay active diagnoser with at most $2^{\mathcal{O}(n^2)}$ states.

4. The aforementioned work allows to minimize the delay between the occurrence of a fault and its detection. However, in designing a controller there is a tradeoff to be made between minimizing the detection delay and the permissiveness of the controller – the smaller the delay, the more restrictive the controller needs to be. We have therefore developed a *parametrized* version of active diagnosis where the parameter d is a delay. Our algorithm builds a controller with delay at most $2d + 1$ that is more or equally permissive than any controller with delay d. Since controllers with same delay can be incomparable w.r.t. permissiveness, such a controller achieves an almost optimal tradeoff.

Organization In Section 2, we recall notions related to diagnosis and active diagnosis. In Section 3, we establish the lower bounds related to the computational complexity, the memory requirements and the index. Section 4.1 presents the construction of the deterministic Büchi automaton. Then in Section 4.2, we solve the decision and the synthesis problems for active diagnosis. After that, Section 4.3 refines the synthesis problem w.r.t. the delay. Section 5 designs a parametrized version of active diagnoser. Section 6 concludes and gives some perspectives of this work.

2 The active diagnosis problem

2.1 Labeled transition systems

When dealing with discrete event systems (DES) diagnosis, systems are often modeled using labeled transition systems (LTS). So we define LTS, their properties and languages.

Definition 1 *A labeled transition system is a tuple $\mathcal{A} = \langle Q, q_0, \Sigma, T\rangle$ where:*

- *Q is a set of states with $q_0 \in Q$ the initial state;*
- *Σ is a finite set of events;*
- *$T \subseteq Q \times \Sigma \times Q$ is the set of transitions.*

We note $q \xrightarrow{a} q'$ for $(q, a, q') \in T$; this transition is then said to be *enabled* in q. A *run* over the word $\sigma = a_1 a_2 \ldots \in \Sigma^\omega$ is a sequence of states $(q_i)_{i\geq 0}$ such that $q_i \xrightarrow{a_{i+1}} q_{i+1}$ for all $i \geq 0$, and we write $q_0 \overset{\sigma}{\Rightarrow}$ if such a run exists. A finite run over $w \in \Sigma^*$ is defined analogously, and we write $q \overset{w}{\Rightarrow} q'$ if such a run ends at state q'. A state q is *reachable* if there exists a run $q_0 \overset{w}{\Rightarrow} q$ for some w.

Definition 2 (Languages of an LTS) *Let $\mathcal{A} = \langle Q, q_0, \Sigma, T\rangle$ be an LTS. The finite language $\mathcal{L}^*(\mathcal{A}) \subseteq \Sigma^*$ of $\mathcal{A}$ and the infinite language $\mathcal{L}^\omega(\mathcal{A}) \subseteq \Sigma^\omega$ of $\mathcal{A}$ are defined by:*

$$\mathcal{L}^*(\mathcal{A}) = \{\, w \in \Sigma^* \mid \exists q : q_0 \overset{w}{\Rightarrow} q \,\} \qquad \mathcal{L}^\omega(\mathcal{A}) = \{\, \sigma \in \Sigma^\omega \mid q_0 \overset{\sigma}{\Rightarrow} \,\}$$

An LTS $\mathcal{A}$ is *live* if for any reachable state there exists a transition enabled in that state. An LTS $\mathcal{A}$ is *deterministic* if for every pair $q \in Q, a \in \Sigma$ there is at most one q' such that $q \xrightarrow{a} q'$. For a deterministic automaton we write $T(q, a) = q'$ if $q \xrightarrow{a} q'$.

2.2 Observations

In order to formalize problems related to diagnosis, we partition Σ into two disjoint sets Σ_o and Σ_{uo}, the sets of *observable* and of *unobservable events*, respectively. Moreover, we distinguish a special *fault* event $f \in \Sigma_{uo}$. Let σ be a finite word; its length is denoted $|\sigma|$. For $\Sigma' \subseteq \Sigma$, define $\mathcal{P}_{\Sigma'}(\sigma)$ inductively by: $\mathcal{P}_{\Sigma'}(\varepsilon) = \varepsilon$; for $a \in \Sigma'$, $\mathcal{P}_{\Sigma'}(\sigma a) = \mathcal{P}_{\Sigma'}(\sigma)a$; and $\mathcal{P}_{\Sigma'}(\sigma a) = \mathcal{P}_{\Sigma'}(\sigma)$ for $a \notin \Sigma'$. Write $|\sigma|_{\Sigma'}$ for $|\mathcal{P}_{\Sigma'}(\sigma)|$, and for $a \in \Sigma$, write $|\sigma|_a$ for $|\sigma|_{\{a\}}$. When σ is an infinite word, its projection is the limit of the projections of its finite prefixes. This projection can be either finite or infinite. As usual the projection is extended to languages. $\mathcal{P}_{\Sigma_o}$ will be more simply denoted by $\mathcal{P}$.

With respect to the partition of $\Sigma = \Sigma_o \uplus \Sigma_{uo}$, an LTS $\mathcal{A}$ is *convergent* if $\mathcal{L}^\omega(\mathcal{A}) \cap \Sigma^*\Sigma_{uo}^\omega = \emptyset$ (i.e. no infinite sequence of unobservable events from any reachable state). When $\mathcal{A}$ is convergent, then for all $\sigma \in \mathcal{L}^\omega(\mathcal{A})$, one has $\mathcal{P}(\sigma) \in \Sigma_o^\omega$. In this paper, we shall assume that the system under diagnosis is live and convergent.

Example 1 *Figure 1 shows a live and convergent LTS with $\Sigma_o = \{a, b, c\}$ and $\Sigma_{uo} = \{f\}$.*

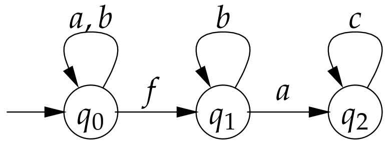

Figure 1: An LTS.

2.3 Diagnosability

A finite (resp. infinite) sequence σ is *correct* if it belongs to $(\Sigma \setminus \{f\})^*$ (resp. $(\Sigma \setminus \{f\})^\omega$). Otherwise σ is called *faulty*. An observation sequence may be the projection of both a correct and a faulty sequence, hence ambiguous.

Definition 3 (ambiguous and surely faulty sequence) *Let $\mathcal{A}$ be an LTS, $\sigma_1, \sigma_2 \in \mathcal{L}^\omega(\mathcal{A})$ be two sequences and $\sigma \in \Sigma_o^\omega$ be an observable sequence such that:*

(1) $\mathcal{P}(\sigma_1) = \mathcal{P}(\sigma_2) = \sigma$, (2) σ_1 is correct and (3) σ_2 is faulty.

Then σ is called ambiguous *and the pair (σ_1, σ_2) is a* witness *for the ambiguity of σ. Ambiguous finite observable sequences are defined analogously.*
A sequence $\sigma' \in \mathcal{P}(\mathcal{L}^*(\mathcal{A}))$ is surely faulty *iff $\mathcal{P}^{-1}(\sigma') \cap \mathcal{L}^*(\mathcal{A}) \subseteq \Sigma^* f \Sigma^*$.*

Definition 4 (Diagnosability) *Let $k \in \mathbb{N}$. An LTS $\mathcal{A}$ is k-diagnosable if:*

$$\forall \sigma = \sigma' f \sigma'' \in \mathcal{L}^*(\mathcal{A})\ |\sigma''|_{\Sigma_o} \geq k \Rightarrow \mathcal{P}(\sigma) \text{ is a surely faulty sequence,}$$

Furthermore, $\mathcal{A}$ is diagnosable *if there exists a k such that $\mathcal{A}$ is k-diagnosable.*

Our definition of diagnosability is a slight variation of the one given in [11]. Indeed the number k above is related to observable events while in former works, it is related to any kind of events. However for finite-state convergent systems (which are the ones addressed by both works) the definitions of diagnosability coincide.

Example 2 *The LTS of Figure 1 is not diagnosable since the correct infinite trace b^ω and the faulty infinite trace fb^ω have the same projection.*

2.4 Active diagnosability

We suppose that Σ_o is partitioned into subsets $\Sigma_c \subseteq \Sigma_o$ of *controllable* and $\Sigma_{uc} = \Sigma_o \setminus \Sigma_c$ of *uncontrollable* actions. Intuitively, a controller may forbid a subset of the controllable actions based on the observations made so far, thereby restricting the behaviour of $\mathcal{A}$.

Definition 5 (Controller) *Let $\mathcal{A}$ be an LTS. A* controller *for $\mathcal{A}$ is a mapping $cont : \mathcal{P}(\mathcal{L}^*(\mathcal{A})) \rightarrow 2^\Sigma$ such that for all σ, $\Sigma_{uc} \cup \Sigma_{uo} \subseteq cont(\sigma)$. The controlled LTS $\mathcal{A}_{cont} = \langle Q_{cont}, q_{0cont}, \Sigma, T_{cont} \rangle$ is defined by:*

- *Q_{cont} is the smallest subset of $\Sigma_o^* \times Q$ such that*
  1. *$(\varepsilon, q_0) \in Q_{cont}$;*
  2. *$(\sigma, q) \in Q_{cont} \wedge a \in cont(\sigma) \wedge q \xrightarrow{a} q'$ implies $(\mathcal{P}(\sigma a), q') \in Q_{cont}$.*

- $q_{0cont} = (\varepsilon, q_0)$
- $((\sigma, q), a, (\sigma', q')) \in T_{cont}$ *iff* $q \xrightarrow{a} q' \wedge a \in cont(\sigma) \wedge \sigma' = \mathcal{P}(\sigma a)$

In the diagnosis framework, the goal of our controller is to make the system diagnosable, and to perfom diagnosis. However, one requires that the control cannot introduce deadlocks.

Definition 6 (Pilot and Active Diagnoser) *Let $\mathcal{A}$ be an LTS. We call $h = \langle cont, diag \rangle$ a* pilot *for $\mathcal{A}$ if cont is a controller and diag is a mapping from $\mathcal{P}(\mathcal{L}^*(\mathcal{A}_{cont}))$ to $\{\bot, \top\}$. Moreover, h is called an* active diagnoser *if:*

1. *$\mathcal{A}_{cont}$ is live;*
2. *$\mathcal{P}(\mathcal{L}^\omega(\mathcal{A}_{cont}))$ does not contain any ambiguous sequence;*
3. *for all $\sigma \in \mathcal{P}(\mathcal{L}^*(\mathcal{A}_{cont}))$, $diag(\sigma) = \top$ if and only if σ is a surely faulty sequence.*

*Moreover, we say that h is a k-*active diagnoser*, for $k \geq 1$, if for all $\sigma = \sigma' f \sigma'' \in \mathcal{L}^*(\mathcal{A}_{cont})$ with $|\sigma''|_{\Sigma_o} \geq k$, $diag(\mathcal{P}(\sigma)) = \top$; in other words, every fault is diagnosed within at most k observations. The minimal value k such that h is a k-active diagnoser is called the* delay *of h. We call $\mathcal{A}$ (k-)*actively diagnosable *if a (k-)active diagnoser exists, and the minimal such k the* index *of $\mathcal{A}$.*

Example 3 *In the LTS of Figure 1, assume that $\Sigma_c = \{a, b\}$. Let $h_n = \langle cont_n, diag \rangle$, with $n \geq 1$, be the pilot defined by:*

- *$cont_n(\sigma b^n) = \{a, c, f\}$ for $\sigma \in \Sigma_c^*$ and $cont_n(\sigma) = \Sigma$ otherwise;*
- *$diag(\sigma) = \top$ iff $\sigma \in \Sigma_o^* c \Sigma_o^*$.*

Then h_n is an active diagnoser with delay $n + 2$.

Notice that an active diagnoser does not necessarily have a finite delay. For instance, in Figure 1, there is an active diagnoser that admits the sequence $bab^2ab^3a\cdots$ and is not an k-active diagnoser for any k. However, we will see that if $\mathcal{A}$ is actively diagnosable, there does exist a k-active diagnoser (for some k). We come back to this point in Section 4.3.

We are now in a position to formally state the relevant problems for active diagnosis. Let $\mathcal{A}$ be a live and convergent LTS with finitely many states. We are interested in:

- the *active diagnosis decision problem*, i.e. decide whether $\mathcal{A}$ is actively diagnosable;
- the *synthesis problem*, i.e. decide whether $\mathcal{A}$ is actively diagnosable and in the positive case build an active diagnoser.
- the *minimal-delay synthesis problem*, i.e. decide whether $\mathcal{A}$ is actively diagnosable and in the positive case build an active diagnoser with minimal delay.

To implement an active diagnoser, it must be finitely representable; for this, we introduce the notion of *state-based pilot*.

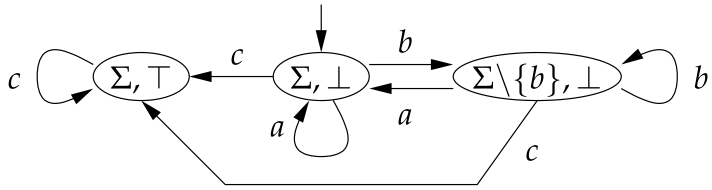

Figure 2: A state-based pilot.

Definition 7 (state-based pilot) *A* state-based pilot $\mathcal{C} = \langle \mathcal{B}, cont_{\mathcal{C}}, diag_{\mathcal{C}} \rangle$ *consists of a deterministic LTS* $\mathcal{B} = \langle Q^c, q_0^c, \Sigma_o, T^c \rangle$ *and labellings* $cont_{\mathcal{C}}, diag_{\mathcal{C}} : Q^c \to 2^{\Sigma} \times \{\bot, \top\}$*, such that (1)* $\mathcal{P}(\mathcal{L}^*(\mathcal{A})) \subseteq \mathcal{L}^*(\mathcal{B})$ *and, (2) for all* $q \in Q^c$, $\Sigma_{uc} \cup \Sigma_{uo} \subseteq cont_{\mathcal{C}}(q)$*. The pilot* $h_{\mathcal{C}} = \langle cont, diag \rangle$ *associated with* $\mathcal{C}$ *is given by* $cont(\sigma) = cont_{\mathcal{C}}(q)$ *and* $diag(\sigma) = diag_{\mathcal{C}}(q)$ *for all* $\sigma \in \mathcal{P}(\mathcal{L}^*(\mathcal{A}))$*, where* q *is the unique state such that* $q_0^c \overset{\sigma}{\Rightarrow} q$*.*

Example 4 *Figure 2 shows a state-based pilot for the LTS of Figure 1. Observe that there is an outgoing transition* b *from the rightmost state (in order to fulfill the language inclusion requirement) but* b *is disabled in this state (in order to implement the active diagnoser* h_1*).*

3 Lower bounds

We first establish that the active diagnosis decision problem is EXPTIME-hard. The proof relies on a reduction from safety games with imperfect information [1].

Theorem 1 (hardness) *The active diagnosis decision problem is EXPTIME-hard.*

Proof. A safety game $\mathcal{G} = (L, l_0, \Sigma, \Delta, O, F, obs)$ with imperfect information is defined by:

- L a finite set of locations with $l_0 \in L$ the initial location;
- Σ a finite alphabet;
- $\Delta \subseteq L \times \Sigma \times L$ the transition relation such that for all $l \in L$ and $a \in \Sigma$ there exists at least one l' with $(l, a, l') \in \Delta$;
- O a finite set of observations with $F \subseteq O$ the final observations;
- $obs : L \mapsto O$ the observation mapping.

$\mathcal{G}$ is a turn-based game played by two players A and B. It starts in location l_0 with A to play. In the first round, A chooses a letter a_0 in Σ, and then B chooses a location l_1 such that $(l_0, a_0, l_1) \in \Delta$. A only observes $o_1 = obs(l_1)$. The next rounds are played similarly. Player A wins if for all i, $o_i \notin F$.

The problem of existence of a winning strategy for player A is EXPTIME-complete [1]. We now describe the reduction of this problem to an active-diagnosis decision problem with LTS $\mathcal{A}$ defined as follows.

- Q, the set of states, is defined by $Q = L \uplus ((L \setminus obs^{-1}(F)) \times \Sigma) \uplus \{\bot\}$ and $q_0 = l_0$.

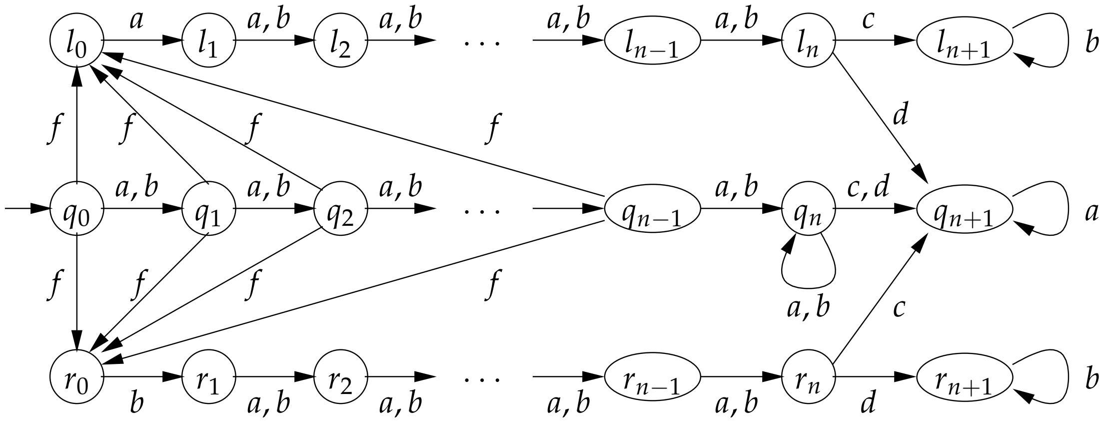

Figure 3: An LTS $\mathcal{A}_n$ with $\Sigma_o = \{a, b, c, d\}$, $\Sigma_c = \{c, d\}$ used in Theorem 2.

– The alphabet $\Sigma' = \Sigma \uplus O \uplus \{u, f, z\}$. The unobservable events are u and f and the (observable) uncontrollable events are $O \uplus \{z\}$.

– T the transition relation is defined as follows.

1. For all $l \in L \setminus obs^{-1}(F)$ and $a \in \Sigma$, $(l, a, (l, a)) \in T$.
2. For all $l \in L \setminus obs^{-1}(F)$, $a \in \Sigma$ and $l' \in L$, $((l, a), obs(l'), l') \in T$ if $(l, a, l') \in \Delta$.
3. For all $l \in obs^{-1}(F)$, $(l, u, \perp)$ and $(l, f, \perp)$ belong to T.
4. $(\perp, z, \perp) \in T$.

From the very definition of $\mathcal{A}$, a sequence is ambiguous if and only if it contains an occurrence of z. Thus a pilot is an active diagnoser if and only if it avoids states $obs^{-1}(F)$. In addition, such a pilot only "controls" the subset of states $L \setminus obs^{-1}(F)$ and due to the assumptions on $\mathcal{G}$, it can safely restrict the allowed events to a single one. Furthermore since the information available to the pilot is exactly that of player A, a winning strategy for player A in $\mathcal{G}$ provides an active diagnoser for $\mathcal{A}$ and vice versa. ■

The next theorems focus on the memory required for synthesis problems related to active diagnosis. We start with the language of unambiguous sequences of an LTS. It is stated with a Büchi condition.

Definition 8 (Büchi automaton) *A* Büchi automaton *over Σ is a tuple $\mathcal{B} = \langle \mathcal{B}', F \rangle$, where $\mathcal{B}' = \langle S, s_0, \Sigma, \delta \rangle$ is an LTS such that S is finite, and $F \subseteq S$ an* acceptance condition. *A* run $(q_i)_{i \geq 0}$ *is* accepting *if $q_i \in F$ for infinitely many values of i. The language $\mathcal{L}(\mathcal{B})$ consists of all words in $\mathcal{L}^\omega(\mathcal{B}')$ for which there exists an accepting run. A Büchi automaton is called deterministic (live) if its underlying LTS is.*

Theorem 2 (lower bound for determinization) *There exists a family $(\mathcal{A}_n)_{n \geq 1}$ of LTS with the size of $\mathcal{A}_n$ in $\mathcal{O}(n)$ such that any deterministic Büchi automaton recognizing the unambiguous sequences of A_n has at least 2^n states.*

Proof. The family of LTS $(\mathcal{A}_n)_{n\geq 1}$ is depicted in Figure 3, where $\Sigma_o = \{a, b, c, d\}$, $\Sigma_c = \{c, d\}$, and the initial state is q_0. Intuitively, during the n first steps a fault can occur leading to the upper (resp. lower) "branch" of the LTS when followed by a (resp. b).

Formally, let $\sigma = w_1 w_2 y a^\omega \in \Sigma_o^*$ be an observable sequence, where $w_1 w_2 \in \{a, b\}^*$, $1 \leq |w_1| \leq n$, $|w_2| = n - 1$, $y \in \{c, d\}$. Let $x_1 \cdots x_{|w_1|}$ be the letters of w_1. There are two possible sequences that have triggered $\sigma' = w_1 w_2 y$: the correct sequence σ' itself and the faulty sequence $x_1 \cdots x_{|w_1|-1} f x_{|w_1|} w_2 y$. If $x_{|w_1|} = a$, before the occurrence of y, the current state is q_n in the correct sequence and ℓ_n in the faulty sequence. So if $y = d$ the two sequences will lead to the same state q_{n+1} while if $y = c$ one sequence will lead to ℓ_{n+1} and the other one to q_{n+1} and they will be discriminated by the next observation. The case $x_{|w_1|} = b$ is symmetrical. So σ is ambiguous iff $x_{|w_1|} = a$ and $y = d$ or $x_{|w_1|} = b$ and $y = c$.
Thus any automaton that distinguishes ambiguous and unambiguous sequences must remember the first n observations, which requires at least 2^n states. ■

With an appropriate choice of controllable events, the family from Figure 3 also provides a lower bound for a state-based active diagnoser.

Theorem 3 (lower bound for pilots) *There exists a family $(\mathcal{A}_n)_{n\geq 1}$ of actively diagnosable LTS with the size of $\mathcal{A}_n$ in $\mathcal{O}(n)$ such that the LTS of any state-based pilot $\mathcal{C}$, where $h_\mathcal{C}$ is an active diagnoser for $\mathcal{A}$, has at least 2^n states.*

Proof. The family is the same as in Theorem 2. The LTS $\mathcal{A}_n$, shown in Figure 3, is actively diagnosable. However assume that one observes a word $\sigma = a_1 \ldots a_m \in \{a, b\}^*$ such that $n \leq m \leq 2n - 1$. Then when $a_{m-n+1} = a$, $\mathcal{A}$ may be in either q_n or l_n, and when $a_{m-n+1} = b$, $\mathcal{A}$ may be in either q_n or r_n. In the former case the controller must forbid d while in the latter it must forbid c. This implies that a corresponding state-based pilot $\mathcal{C}$ must be in two different states after seeing two different words from $\{a, b\}^n$, therefore it must have at least 2^n states. ■

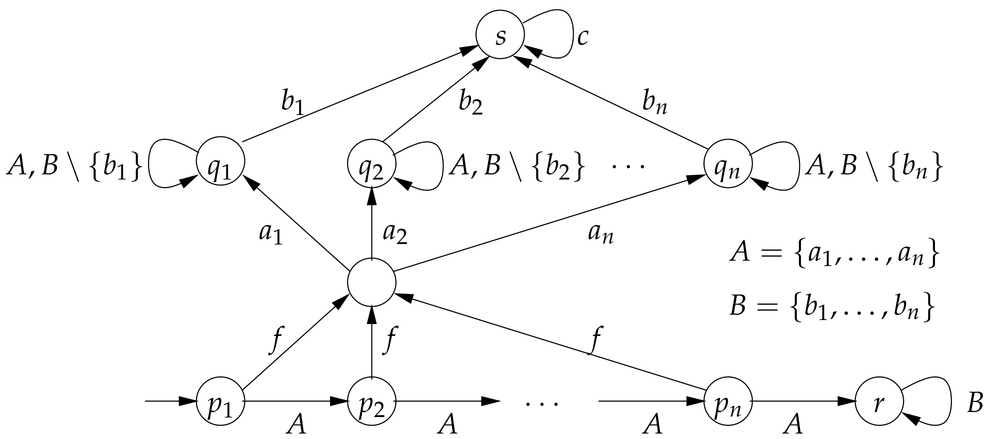

Figure 4: An LTS $\mathcal{A}_n$ with $\Sigma_o = A \cup B \cup \{c\}$, $\Sigma_c = B$ whose minimal-delay active diagnoser requires at least $n!$ states.

We next come to the question how expensive it is to build an active diagnoser that realizes the minimal delay possible. It turns out that this requires an even larger controller. Consider the LTS $\mathcal{A}_n$ of Figure 4, which contains the observable (and uncontrollable) observation sequence $a_{\pi(1)} \ldots a_{\pi(n)}$, where π is a permutation. Such a sequence is ambiguous since a fault may have occurred before any observable event. To remove ambiguity with minimal delay (i.e. $n+2$) an active diagnoser must disallow at time $n+i$ all events in B except $b_{\pi(i)}$ that forces the potential faulty sequence to reach state s where only c is possible. While this LTS has a variable alphabet, the following result also holds for a fixed-size alphabet (see Figure 5).

Theorem 4 (minimal-delay diagnoser) *There exists a family $(A_n)_{n\geq 1}$ of $f(n)$-actively diagnosable LTS (for some function f) with $\mathcal{O}(n)$ states such that the LTS of any state-based pilot $\mathcal{C}$, where $h_{\mathcal{C}}$ is an $f(n)$-active diagnoser for $\mathcal{A}$, has at least $n!$ states.*

Proof. Consider the LTS $\mathcal{A}_n$ in Figure 4. The alphabet Σ of $\mathcal{A}_n$ is $A \cup B \cup \{c, f\}$, where $A := \{a_1, \ldots, a_n\}$ and $B := \{b_1, \ldots, b_n\}$; moreover, we use the abbreviation $B_i := B \setminus \{b_i\}$. As usual, $\Sigma_{uo} = \{f\}$ and $\Sigma_o = \Sigma \setminus \Sigma_{uo}$, and moreover, the set of controllable actions is $\Sigma_c = B$.

Intuitively, in this example, the first n observations can be uncontrollable and can be used by the environment to encode a permutation π. In order to minimize the maximal delay between the occurrence of any fault and its detection, the controller must remember π and repeat it, in a sense made more precise below.

For $i = 1, \ldots, n$, when the system is in p_i, the system can do any action from A or commit a fault, do a_j, and go to q_j (for any $(j = 1, \ldots, n)$. In q_i the system can do b_i, which leads to s (after which the fault becomes obvious), or loop with any other action from $A \cup B$.

Suppose that in the first n steps, one observes $\sigma = a_{\pi(1)} \cdots a_{\pi(n)} \in \Sigma_a^n$, where π is a bijection on $\{1, \ldots, n\}$ (i.e., a permutation). Then the system could be either in state r, or for any $i = 1, \ldots, n$, a fault may have occurred immediately before $a_{\pi(i)}$, in which case the system has been in state q_i for $n + 1 - \pi^{-1}(i)$ steps.

After the observation σ, the system could be in state r. Therefore, any active diagnoser must from then on admit at least one action from B to leave the system alive. Moreover, suppose that the system is actually in state q_i. Then, if the controller blocks the set B_i, the only possible actions are $A \cup \{b_i\}$, which reveal the fault either immediately by observing some action in A or one step later, by observing b_i followed by c.

Thus, if the controller blocks the action sets $B_{\pi(1)}, \cdots, B_{\pi(n)}$, in that order, that is equivalent to forcing the sequence $b_{\pi(1)} \cdots b_{\pi(n)}$, then any error will be discovered after at most $f(n) = n + 2$ observations. It is easy to see that $n + 2$ is also a lower bound for the index of $\mathcal{A}_n$, since initially the controller cannot prevent a sequence of n consecutive actions from the uncontrollable set A, and after this it takes at least two more steps to force the system to produce a c if a fault has previously occurred.

When the initial observations described by π do not correspond to a permutation, then the job of the controller becomes easier. Indeed it only has to memorize the earliest occurrence of any a_i for discarding the possibility of a fault leading to q_i as soon as possible.

To conclude, in order to enforce the correct sequence of actions, any state-based pilot must remember π, requiring at least $n!$ states.

The example in Figure 4 has an alphabet of size $\mathcal{O}(n)$. However, the result of Theorem 4 also holds for a fixed alphabet size. Consider the system $\mathcal{A}'_n$ shown in Figure 5, a variant of Figure 4 where the observable alphabet is $\{a, b, c, 0, 1, \bar{0}, \bar{1}\}$, with f the only invisible action and $0, 1, \bar{0}, \bar{1}$ controllable. Notice that the size of $\mathcal{A}'_n$ is in $\mathcal{O}(n)$, like that of $\mathcal{A}_n$.

The system of Figure 5 works mostly like the one of Figure 4, but with indices from $1, \ldots, n$ encoded in unary. For $i = 1, \ldots, n$, let $code(a_i) = 1^i 0^{n-i} a$ and $code(b_i) = \bar{1}^i \bar{0}^{n-i} b$. Moreover, for the sake of completeness $code(c) = c$. The reader can convince himself that, modulo this encoding, $\mathcal{A}'$ "simulates" $\mathcal{A}$ in the following sense: Let $s, s' \in \{p_1, \ldots, p_n, q_1, \ldots, q_n, r, s\}$ be two states of $\mathcal{A}_n$ and x be an observable action in $\mathcal{A}_n$. Then $s \xrightarrow{x} s'$ (respectively $s \overset{fx}{\Longrightarrow} s'$) in $\mathcal{A}_n$ if and only if $s \xRightarrow{code(x)} s'$ (respectively $s \xRightarrow{f\,code(x)} s'$) in $\mathcal{A}'_n$.

We then observe that an active diagnoser of $\mathcal{A}'_n$ cannot enforce any sequence that does not correspond to an encoding of $A \cup B \cup \{c\}$:

- Suppose that the system is (potentially) in states p_i and q_j (having seen $code(a_j)$ before). Because of p_i, the controller must now admit 1, and if 1 occurs, the system is potentially in states t_1 or t'_1. Thus for the next $n-1$ steps, the controller cannot forbid 0 ant it can forbid a 1 only after the occurence of a 0. So the environment can therefore play $code(a_i)$, for any $i = 1, \ldots, n$. Only when the system has potentially reached t_n or t'_n, a controller can forbid both $0, 1$, thus forcing the system to make either c (in case it really is in t_n or t'_n), or a. Indeed, unless the controller enforces this, the system may loop indefinitely between p_i and p_{i+1} (without fault occurrence) as well as in q_j (after fault occurrence), which is undesirable for achieving diagnosis.
- Suppose that the system is potentially in state r. Then a similar mechanism (using the states from r_1 to r_n, r'_n) ensures that the system must admit n successive observations of $\bar{1}$ and $\bar{0}$, followed by a b, encoding an element of B.

This observation, together with the 'simulation' between $\mathcal{A}_n$ and $\mathcal{A}'_n$, means that the controller once again needs to remember an n-permutation π. Then, the controller achieves a delay corresponding to the length of $n+1$ encodings of symbols from $A \cup B$ plus one occurrence of c, that is $(n+1)^2 + 1 = n^2 + 2n + 2$, which is again the minimum possible. ■

While the previous examples exhibit an index linear or quadratic w.r.t. the size of the LTS, this index may be exponential in the worst case (and no more as shown in the next section).

Theorem 5 (lower bound for index) *There exists a family $(A_n)_{n \geq 1}$ of actively diagnosable LTS with $\mathcal{O}(n)$ states such that the index of $\mathcal{A}_n$ is at least 2^n.*

Proof. Consider the LTS $\mathcal{A}_n$ shown in Figure 6. The alphabet of $\mathcal{A}_n$ consists of the observable actions b and 0 to n and the unobservable fault f, which may only happen in the beginning. The actions 0 to n are controllable, and b shall serve only to reveal the initial fault. Notice that the size of the alphabet of $\mathcal{A}_n$ depends on n; we shall later show that the result even holds for a fixed alphabet.

In this automaton, as in the Figure 7, we sometimes label transition with multiple letters; this is merely for clarity, and these transitions could be replaced by a sequence of transitions with intermediate states without changing the fact that $\mathcal{A}_n$ has $\mathcal{O}(n)$ states.

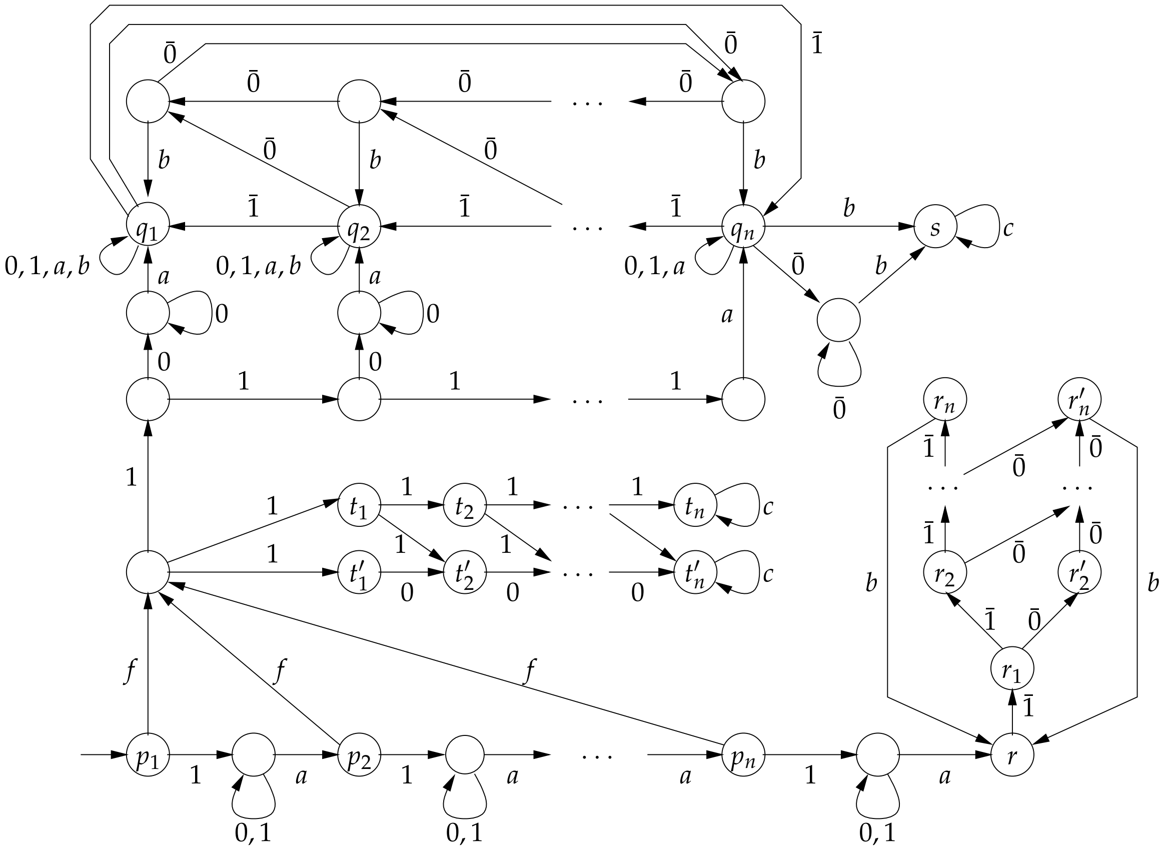

Figure 5: A variant of Figure 4 with fixed alphabet size.

We claim that $\mathcal{A}_n$ is actively diagnosable with an index in $\Omega(2^n)$. Indeed, suppose that the two initial observations, which a controller cannot prevent to leave the system alive, are nn. After these, the system may be in the state q (after committing a fault), in p, or in any state from the set $R := \{r_0, \ldots, r_{n-1}\}$ (without fault). Since the system can loop in q with all actions from 0 to $n-1$, the goal of the controller must be to force an occurrence of n (by prohibiting actions 0 to $n-1$). This would cause the system to leave the state q (if indeed it is there) and play a b, which would reveal the fault. However, the actions available to the states in R are limited, so in order to respect the liveness criterion, the controller must first try to exclude the possibility of the system being in any state from R before forcing n. Then again, this takes at least $2^n - 1$ steps, as we explain now.

For an observation sequence σ, let us define as $U(\sigma)$ (uncertainty set) the subset of R to which σ can lead. Moreover, for $R' \subseteq R$, define the value $v(R') := \sum_{r_i \in R'} 2^i$. Thus, $v(U(nn)) = 2^n - 1$, and the controller must enforce a suffix σ' such that $v(U(nn\sigma')) = 0$. One can now see that the controller can decrease the value v by at most one per observable action, where the uncertainty set behaves like a binary counter. For instance, if $U(\sigma)$ contains r_0, then the controller must allow 0, the only action available to r_0. If the system then indeed plays 0, then $U(\sigma 0) = U(\sigma) \setminus \{r_0\}$ since all other states in R (and p, q) can loop with 0, so $v(U(\sigma 0)) = v(U(\sigma)) - 1$. More generally, if i is the least value such that $r_i \in U(\sigma)$, then the controller must admit at least one of the actions from 0 to i. However, the actions from 0 to $i-1$ will cause all states in $U(\sigma)$ to loop, so the controller gains nothing from allowing these actions, and it will allow only i (and optionally higher values). If the system then plays the i action, this will remove r_i from the uncertainty set but add all the states from r_0 to r_{i-1} (from p). Thus, $U(\sigma i) = U(\sigma) \setminus \{i\} \cup \{r_0, \ldots, r_{i-1}\}$, and again the value has decreased by exactly one.

This concludes the proof; the index is precisely $2^n + 3$ since the shortest sequence enforceable by a controller that reveals the initial occurrence of the fault is $nn\sigma nb$, where σ is a sequence of length $2^n - 1$.

We now demonstrate that the result still holds even when the alphabet is of a fixed size (independently of n). Consider the automaton $\mathcal{A}'_n$ in Figure 7, a version of Figure 6 in which the values 0 to n are encoded in unary: for $i \in \{0, \ldots, n\}$, let $code(i) = 1^i 0^{n-i} a$ the action sequence in Figure 7 that corresponds to the action i in Figure 6. Again, we allow multiple letters on transitions for the sake of clarity, but the automaton has still $\mathcal{O}(n)$ states even without this trick.

$\mathcal{A}'_n$ allows loops starting and ending in q that read exactly the sequences $code(i)$, for $i = 0, \ldots, n$. If ever the environment causes a sequence of symbols from $0, 1, a$ that does not correspond to this encoding, it will reveal that the initial fault did not happen, but the controller cannot actually enforce such an invalid encoding sequence due to liveness constraints in the states $q_1, \ldots, q_n$ and $q'_1, \ldots, q'_n$. Therefore, w.l.o.g., we can suppose that the environment always chooses sequences that correspond to valid encodings, and the controller can only limit the choices among these encodings. For the sake of completeness, we also let $code(b) = b$.

Like in the proof of Theorem 4, we remark that $\mathcal{A}'_n$ "simulates" $\mathcal{A}_n$ in the following sense: Let v, v' be any two states of $\mathcal{A}_n$ and x an observable symbol. Then $v \xrightarrow{x} v'$ (respectively $v \overset{fx}{\Longrightarrow} v'$) in $\mathcal{A}_n$ if and only if $v \xRightarrow{code(x)} v'$ (respectively $v \xRightarrow{f\,code(x)} v'$) in $\mathcal{A}'_n$. Thus, the controller has the same options as in Figure 6, modulo the aforementioned encoding, and the index is proportional to $n \cdot 2^n$, hence in $\Omega(2^n)$. ■

4 Size-Optimal Controller

4.1 Characterization of unambiguous sequences

In this section, we characterize the infinite unambiguous sequences in an efficient way. Fix a finite-state live, convergent LTS $\mathcal{A} = \langle Q, q_0, \Sigma, T \rangle$ for the rest of the section. We build a Büchi automaton $\mathcal{B} = (\mathcal{B}', F)$ that accepts the unambiguous observation sequences. Since $\mathcal{B}$ is the base of the active diagnoser constructed in Section 4.2, we want $\mathcal{B}$ to be deterministic.

A potential procedure for obtaining a deterministic automaton accepting unambiguous sequences is as follows: First, build a non-deterministic Büchi automaton which accepts observable sequences that can be triggered by both a correct and a faulty sequence, leading to a quadratic blow up w.r.t. the size of $\mathcal{A}$. Then, determinize it by the Safra procedure [9], yielding a deterministic Rabin automaton, and complement it so it accepts the unambiguous

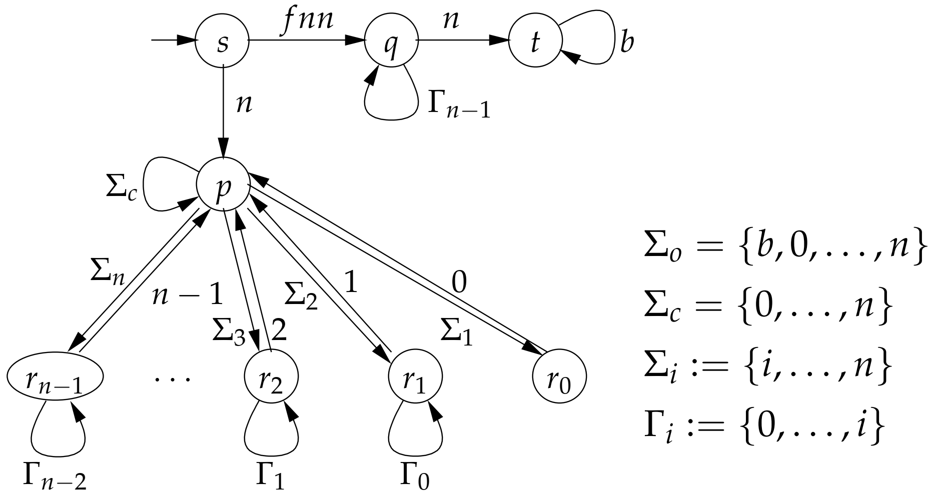

Figure 6: A system whose delay is $2^n + 3$.

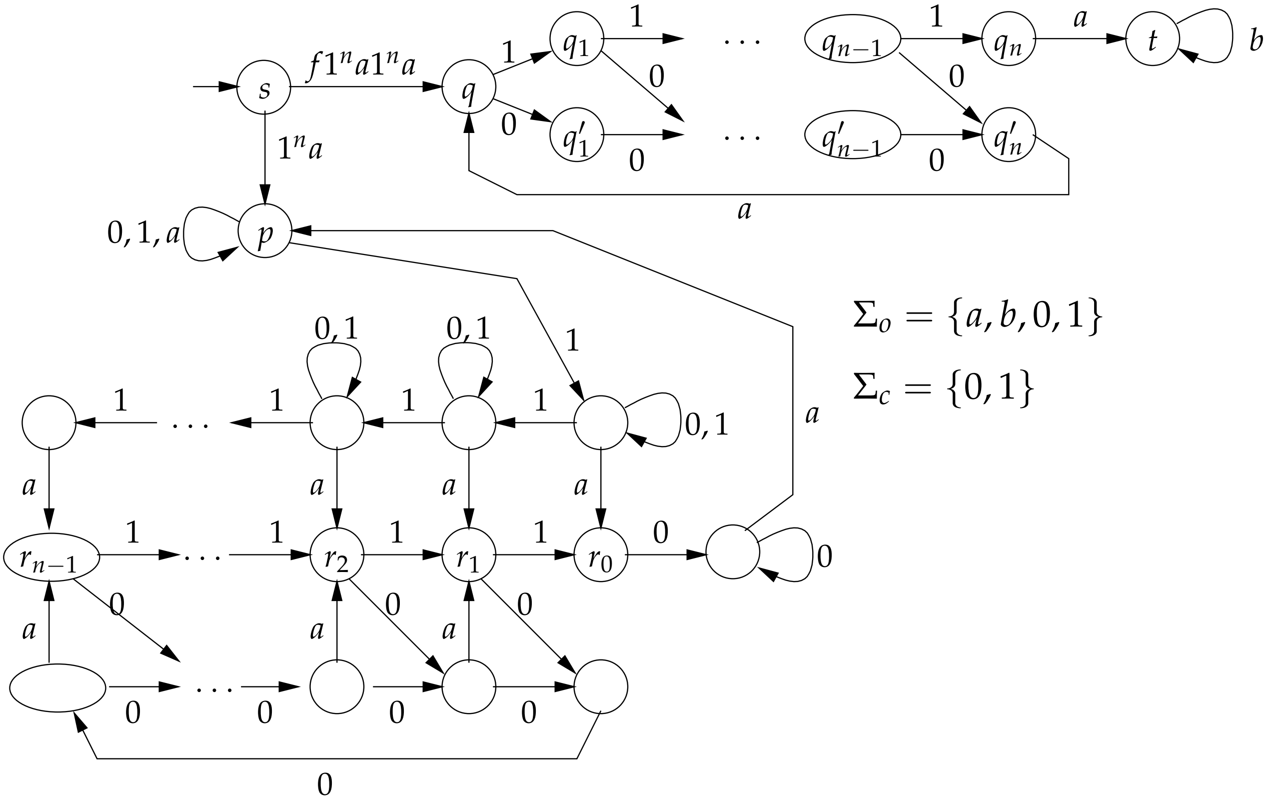

Figure 7: A variant of Figure 6 with fixed alphabet size.

sequences. However, we now provide the construction of a simpler and smaller deterministic Büchi automaton. More precisely, the automaton that we build has the following properties:

– $\mathcal{B}'$ is deterministic;
– $\mathcal{B}'$ "reads" the observable sequences of $\mathcal{A}$, i.e. $\mathcal{L}^\omega(\mathcal{B}') = \mathcal{P}(\mathcal{L}^\omega(\mathcal{A}))$;
– $\mathcal{B}$ accepts exactly the unambiguous observation sequences.

We first give some intuition about the way $\mathcal{B}$ works. Its states are triples $\langle U, V, W\rangle$, where $U, V, W \subseteq Q$. The states in U represent states reachable by non-faulty traces in $\mathcal{A}$, whereas $V \cup W$ are states reachable by committing a fault. Let $\sigma = a_1 a_2 \ldots \in \Sigma_o^\omega$ be an observation sequence. An ambiguous prefix of σ will lead to a state in which both U and $V \cup W$ are non-empty, and if σ is ambiguous, then its run will eventually remain in such states forever. Unfortunately, the reverse implication is not true, as the example from Figure 1 shows: every finite prefix of the sequence a^ω is ambiguous, but a^ω is not. In order to distinguish ambiguous sequences from those that merely have infinitely many ambiguous prefixes, V and W assume different functions: W represents a "watchlist", initially empty. Suppose that the observation $a_1 \ldots a_j$, for some j, corresponds to some faulty execution. Then we put the state reachable by that faulty execution into W and trace its successor states there while making further observations. If W never becomes empty, then indeed there exists a faulty element of $\mathcal{P}(\sigma)$ in $\mathcal{L}^\omega(\mathcal{A})$. On the other hand, if some observation $a_{j'}$, for $j' > j$, is impossible in all states of W, then we can conclude that no fault has occurred before a_j. In the meantime, V serves as a "waiting room": it stores states that can be reached by faulty sequences where the fault has occurred between observations a_j and $a_{j'}$. When W becomes empty, those states are shifted from V to W to form the new watchlist.

Let us introduce some notations. Let $S' \subseteq S$, $a \in \Sigma_o$, and $\mathcal{L} \subseteq \Sigma_{uo}^*$ be a language of unobservable actions. We denote $\delta_{\mathcal{L}}(S', a) := \{\, q \in Q \mid \exists q' \in S', w \in \mathcal{L} : q' \overset{wa}{\Longrightarrow} q \,\}$, and introduce the abbreviations

– δ_n for $\mathcal{L} = (\Sigma_{uo} \setminus \{f\})^*$ (non-faulty executions),
– δ_f for $\mathcal{L} = \Sigma_{uo}^* f \Sigma_{uo}^*$ (faulty executions),
– and δ_* for $\mathcal{L} = \Sigma_{uo}^*$ (arbitrary executions).

We can now state the formal construction of $\mathcal{B} = \langle\langle S, s_0, \Sigma_o, \delta\rangle, F\rangle$ as follows:

– $S = 2^Q \times 2^Q \times 2^Q \setminus \{\langle\emptyset, \emptyset, \emptyset\rangle\}$;
– $s_0 = \langle\{q_0\}, \emptyset, \emptyset\rangle$;
– for $s = \langle U, V, W\rangle \in S$ and $a \in \Sigma_o$ such that $\delta_*(U \cup V \cup W, a) \neq \emptyset$, let $\Delta := \delta_f(U, a) \cup \delta_*(V, a)$; then

$$\delta(s, a) = \begin{cases} \langle \delta_n(U, a), \emptyset, \Delta\rangle & \text{if } W = \emptyset \\ \langle \delta_n(U, a), \Delta \setminus \delta_*(W, a), \delta_*(W, a)\rangle & \text{otherwise;} \end{cases}$$

– $F = \{\, \langle\emptyset, S_1, S_2\rangle, \langle S_1, S_2, \emptyset\rangle \mid S_1, S_2 \subseteq Q \,\}$.

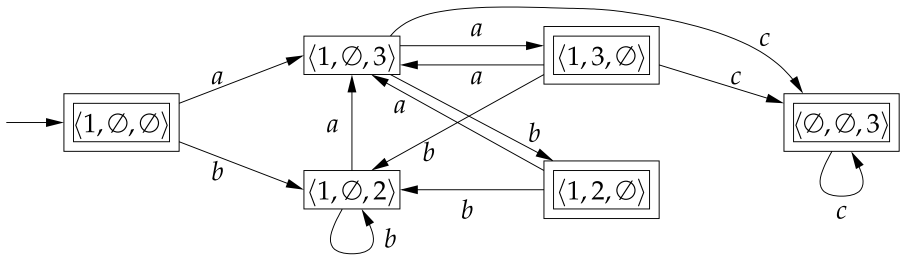

Figure 8: Büchi automaton resulting from Figure 1; accepting states have double frames.

Observe that disregarding the acceptance condition, the sequences read by $\mathcal{B}$ exactly correspond to observable sequences of $\mathcal{A}$, i.e. $\mathcal{P}(\mathcal{L}^\omega(\mathcal{A}))$.

Theorem 6 *A sequence of observations $\sigma \in \Sigma_o^\omega$ is accepted by $\mathcal{B}$ iff it is an unambiguous sequence of $\mathcal{A}$.*

Proof. Fix an observable sequence σ of $\mathcal{A}$, and a run $(q_i := \langle U_i, V_i, W_i\rangle)_{i\geq 0}$ of $\mathcal{B}$ for σ.

For one direction assume that σ is ambiguous; we show that the run is non-accepting. Let (σ', σ'') be a pair of witnesses for σ. Because of σ', we have $U_i \neq \emptyset$ for all $i \geq 0$. Moreover, we will show that σ'' implies the existence of some i_0 such that for all $i \geq i_0$ we have $W_i \neq \emptyset$. These two facts together mean that the run is non-accepting, since $q_i \notin F$ for all $i \geq i_0$. So, let w the minimal prefix of σ'' containing f, and let $|w|_{\Sigma_o} = j$. Then clearly, $V_i \cup W_i \neq \emptyset$ for all $i > j$.

- Either $W_j = \emptyset$, i.e. the watchlist is empty after j observations; then the faulty run of $\mathcal{A}$ for σ'' will be recorded in the watchlist after the next observation, and remain there; we take $i_0 := j + 1$.
- Or $W_j \neq \emptyset$; then the possibility of a fault will be recorded in the "waiting room". Then either the watchlist becomes empty at a later time, i.e. $W_{j'} = \emptyset$ for some $j' > j$, in which case the faulty run for σ'' is transferred to the watchlist in the next step, and we take $i_0 := j' + 1$; or the watchlist never becomes empty, in which case there exists another witness pair, and we take $i_0 := j$.

For the other direction, assume that this run is non-accepting. Let j be the highest index such that $q_j \in F$. Then $U_i, W_i \neq \emptyset$ for all $i > j$. The structure of δ implies (i) the existence of some non-faulty prefix that can reach a state of U_j and continue from there (without faults), and (ii) the existence of a faulty prefix that can reach a state from W_j that can continue forever (with or without faults). Thus σ is ambiguous. ■

Example 5 *Figure 8 shows the result of the construction on the system from Figure 1. Since all non-empty sets are singletons we have represented them by their item. Notice that any sequence ending in b^ω is ambiguous in Figure 1 and hence not accepted in Figure 8. On the other hand, e.g., sequence a^ω is accepted: while every prefix a^i, for $i \geq 1$, is ambiguous, we always know after $i+1$ observation that no fault has occurred before the i-th observation.*

We briefly discuss the relationship of our determinization construction with other standard constructions in diagnosis and automata theory. In [14], diagnosability of an LTS $\mathcal{A}$ is decided by building two automata: one is a modification of $\mathcal{A}$ that accepts the projections all non-faulty sequences, the other accepts the projections of all faulty sequences, remembering whether a fault has occurred in the current state. The cross product of these two is a non-deterministic Büchi automaton of size $2n^2$ (for $|Q| = n$) that accepts all ambiguous sequences. A direct determinization [9] of that cross product would yield a Rabin automaton of size $2^{\mathcal{O}(n^2 \log n)}$. However, given that the cross product is *weak* in the sense that all its strongly connected components are either fully accepting or fully non-accepting, one could apply the *breakpoint construction* of Miyano and Hayashi [7] to obtain a deterministic Büchi automaton of its complement language, of size 3^{2n^2}. Our construction, while similar in spirit to that of [7], is more efficient than that: for a reachable Büchi state $\langle U, V, W\rangle \in S$, any LTS state $q \in Q$ may or may not appear in U, and it may appear in at most one of V or W, but not in both. Thus, the number of reachable states in $\mathcal{B}$ is bounded by $2^n \cdot 3^n = 6^n = 2^{\mathcal{O}(n)}$. Theorem 2 shows that an exponential blowup in n is unavoidable in general, i.e. our construction is optimal up to a constant factor in the exponent.

4.2 Synthesizing the controller

We simultaneously solve the decision and synthesis problems. As before, we fix an LTS $\mathcal{A} = \langle Q, q_0, \Sigma, T\rangle$. We shall try to construct a state-based pilot $\mathcal{C}$ such that $h_{\mathcal{C}}$ is an active diagnoser for $\mathcal{A}$. The construction succeeds iff $\mathcal{A}$ is actively diagnosable. According to Definition 6, the main challenges in building an active diagnoser are to ensure that (i) the controlled system remains live, (ii) the controller excludes the ambiguous sequences, and (iii) diagnosis information is provided. For this, we introduce Büchi games.

Definition 9 (game) *A game $\mathcal{G}$ (between two players called Control and Environment) is a tuple $\langle V_C, V_E, E, v_0, V_F\rangle$, where V_C are the vertices owned by Control, V_E are the vertices owned by Environment; $V_{\mathcal{G}}$ denotes all vertices, and $v_0 \in V_C$ is an* initial vertex. *$E \subseteq V_{\mathcal{G}} \times V_{\mathcal{G}}$ is a set of directed edges such that for all $v \in V_C$ there exists $(v, w) \in E$, and $V_F \subseteq V_{\mathcal{G}}$ is a* winning condition.
A play *is a function $\rho\colon \mathbb{N} \to V_{\mathcal{G}}$ such that $\rho(0) = v_0$ and $\langle \rho_i, \rho_{i+1}\rangle \in E$ for all $i \geq 0$; we call $\rho^k := \rho(0)\cdots\rho(k)$, for some $k \geq 0$, a* partial play *if $\rho(k) \in V_C$, and set $state(\rho^k) := \rho(k)$. We write $Play^*(\mathcal{G})$ for the set of partial plays of $\mathcal{G}$. A play ρ is called* winning *(for Control) if $\rho(i) \in V_F$ for infinitely many i.*

Definition 10 (strategy) *Let $\mathcal{G} = \langle V_C, V_E, E, v_0, V_F\rangle$ be a game. A* strategy *(for Control) is a function $\theta\colon Play^*(\mathcal{G}) \to V_{\mathcal{G}}$ such that $\langle state(\xi), \theta(\xi)\rangle \in E$ for all $\xi \in Play^*(\mathcal{G})$. A play ρ* adheres *to θ if $\rho(i) \in V_C$ implies $\rho(i+1) = \theta(\rho^i)$ for all $i \geq 0$. A strategy is called* winning *if every play ρ that adheres to θ is winning. A* positional strategy *is a function $\theta'\colon V_C \to V_{\mathcal{G}}$ such that $\langle v, \theta'(v)\rangle \in E$ for all $v \in V_C$; we call θ'* winning *if the strategy θ with $\theta(\xi) = \theta'(state(\xi))$ is winning.*

In the games that we have defined, a play can only be stuck in a state of Environment. Thus we do not consider finite maximal plays for defining the winning strategies of Control. Let $\mathcal{B} = \langle \mathcal{B}', F\rangle$, with $\mathcal{B}' = \langle S, s_0, \Sigma_o, \delta\rangle$, be the deterministic Büchi automaton constructed from $\mathcal{A}$ in Section 4.1. We shall take $\mathcal{B}'$ as the LTS component of $\mathcal{C}$. To determine $cont_{\mathcal{C}}$, we construct a Büchi game based on $\mathcal{B}$. The objective of Control is to obtain an accepting run by suitably restricting the possible actions, and any winning strategy will be a suitable candidate for $cont_{\mathcal{C}}$. Intuitively, a round of the game is played as follows:

1. Control restricts the set of possible actions to Σ'.

2. Environment chooses an action $a \in \Sigma'$ to determine the next state of $\mathcal{B}$.

The choices of Control are subject to some restrictions. Indeed, each state $s = \langle U, V, W \rangle$ represents Control's knowledge about the current potential states of $\mathcal{A}$. To ensure that the controlled system remains live, Σ' must not cause deadlocks in any state reachable by unobservable events from $U \cup V \cup W$. Also, Control cannot prevent the uncontrollable events. So we define the admissible sets and the game as follows.

Definition 11 (admissible action set) *Let $s = \langle U, V, W \rangle$ be a state of $\mathcal{B}$. We call $\Sigma' \subseteq \Sigma_o$* admissible *for s if (i) $\Sigma_{uc} \subseteq \Sigma'$ and (ii) for all states q' of $\mathcal{A}$ with $q \overset{w}{\Rightarrow} q'$ for some $q \in U \cup V \cup W$ and $w \in \Sigma_{uo}^*$, there exists $a \in \Sigma'$ and $q'' \in Q$ with $q' \xrightarrow{a} q''$. The admissible sets for s are denoted $adm(s)$.*

Definition 12 (controller-synthesis game) *Let $\mathcal{B} = \langle \langle S, s_0, \Sigma_o, \delta \rangle, F \rangle$ be a Büchi automaton. We denote $\mathcal{G}(\mathcal{B})$ the game $\langle V_C, V_E, E, s_0, F \rangle$, where $V_C = S$, $V_E = (S \times 2^{\Sigma_o}) \cup (S \times \Sigma_o)$, and $E = E_1 \cup E_2 \cup E_3$, where*

- $E_1 = \{ \langle s, \langle s, \Sigma' \rangle \rangle \mid s \in S,\ \Sigma' \in adm(s) \}$;
- $E_2 = \{ \langle \langle s, \Sigma' \rangle, \langle s, a \rangle \rangle \mid s \in S,\ a \in \Sigma' \}$;
- $E_3 = \{ \langle \langle s, a \rangle, s' \rangle \mid \delta(s, a) = s' \}$.

The set E_3 is only introduced to record the sequence of observable actions that occur during a play. Furthermore Environment can be stuck in a vertex of E_3 meaning that the action chosen by Environment does not correspond to a possible behavior of the system.

Example 6 *Figure 9 depicts an excerpt of the game for Example 1. In the initial state, there are three possible admissible sets, all including c, the uncontrollable observable action. $\{c\}$ is not an admissible set as it blocks the system. If Environment chooses action c, it immediately loses since c is not possible initially even after a fault.*

We can now address the decision and synthesis problems. To this aim, we shall mainly exploit the following facts: (1) Büchi games can be solved in polynomial time, (2) a positional winning strategy can always be chosen for Control if it wins and (3) there is a tight correspondence between winning strategies and active diagnosers.

In the following proofs, we will, for any $\sigma \in \mathcal{L}^*(B')$, denote by $\delta_0(\sigma)$ the unique state s such that $s_0 \overset{\sigma}{\Rightarrow} s$. Recall also that $\mathcal{L}^\omega(B') = \mathcal{P}(\mathcal{L}^\omega(\mathcal{A}))$. Moreover, let $\xi \in Play^*(\mathcal{G}(\mathcal{B}))$ be a partial play. We define $word(\xi)$ as the observable actions played along ξ, i.e. $word(\varepsilon) = \varepsilon$, $word(\xi v) = word(\xi)$ if $v \neq S \times \Sigma_o$, and $word(\xi \langle s, a \rangle) = word(\xi) a$. In a similar way, $run(\xi)$ are the states of S touched along ξ, formally $run(\xi) = \mathcal{P}_S(\xi)$. We extend these notions to plays ρ in the natural way. The following remark is obvious by construction of $\mathcal{G}(\mathcal{B})$.

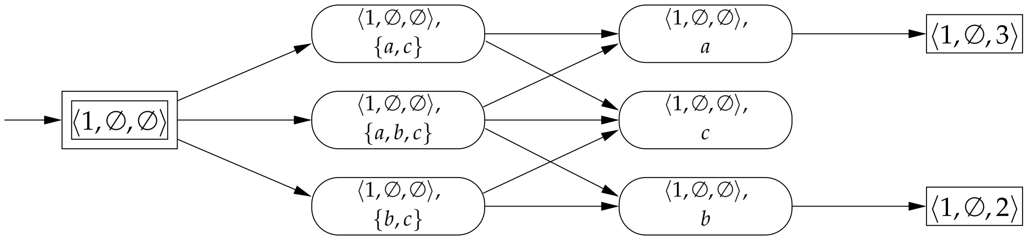

Figure 9: Excerpt of the Büchi game for Example 1.

Remark 1 *Let ρ be a play of $\mathcal{G}(\mathcal{B})$. Then $word(\rho) \in \mathcal{L}^\omega(\mathcal{B}')$ and $run(\rho)$ is the corresponding run in $\mathcal{B}'$. Moreover, for a partial play ξ, we have $state(\xi) = \delta_0(word(\xi))$.*

Lemma 1 *Let $h = \langle cont, diag \rangle$ be an active diagnoser for $\mathcal{A}$. Then there exists a winning strategy θ_h in the game $\mathcal{G}(\mathcal{B})$. Moreover, there also exists a winning positional strategy θ in $\mathcal{G}(\mathcal{B})$.*

Proof. Suppose that $h = \langle cont, diag \rangle$ is any active diagnoser for $\mathcal{A}$. Moreover, *cont* defines a (not necessarily positional) strategy θ_h in $\mathcal{G} = \mathcal{G}(\mathcal{B})$: for $\xi \in Play^*(\mathcal{G})$, let $\theta_h(\xi) = \langle state(\xi), cont(word(\xi)) \rangle$. Since *cont* depends only on the observable actions and $\mathcal{A}_{cont}$ is live, it is easy to see that for any $\sigma \in \mathcal{P}(\mathcal{L}^\omega(\mathcal{A}))$, $cont(\sigma)$ must be admissible for $\delta_0(\sigma) = state(\xi)$. Thus, there is indeed an edge from $state(\xi)$ to $\theta_h(\xi)$ in $\mathcal{G}$.

Now, for any play ρ that adheres to θ_h we have that $word(\rho) \in \mathcal{P}(\mathcal{L}^\omega(\mathcal{A}_{cont}))$, thus by Definition 5, $word(\rho)$ is not ambiguous and hence by Theorem 6 accepted by $\mathcal{B}$, so by Remark 1 $run(\rho)$ touches F infinitely often and ρ is winning, which means that θ_h is a winning strategy.

Finally, the existence of θ_h implies the existence of a positional winning strategy due to well-known results of game theory, see, e.g., [6]. ■

Let θ be a positional strategy in $\mathcal{G}(\mathcal{B})$. We define a state-based pilot $\mathcal{C}(\theta) := \langle \mathcal{B}', cont_\mathcal{C}, diag_\mathcal{C} \rangle$ by setting, for any $s = \langle U, V, W \rangle \in S$, $cont_\mathcal{C}(s) := \Delta' \cup \Sigma_{uo}$, where $\theta(s) = \langle s, \Delta' \rangle$, and $diag_\mathcal{C}(s) := \top$ iff $U = \emptyset$.

Lemma 2 *Let θ be a positional winning strategy in $\mathcal{G}(\mathcal{B})$. Then $h_\mathcal{C}$, for $\mathcal{C} = \mathcal{C}(\theta)$, is an active diagnoser for $\mathcal{A}$.*

Proof. We note that for any state $s \in S$, $cont_\mathcal{C}(s)$ contains Σ_{uo} (by construction) and Σ_{uc} (by Definition 11 (i)). Thus, in $h_\mathcal{C} = \langle cont, diag \rangle$, *cont* is a controller according to Definition 5. Moreover, we show that $h_\mathcal{C}$ fulfils the three conditions from Definition 6, which shows that $\mathcal{A}$ is actively diagnosable.

1. Let $\sigma' \in \Sigma^*$ and suppose that $\langle \varepsilon, q_0 \rangle \xrightarrow{\sigma'} \langle \sigma, q \rangle$ in $\mathcal{A}_{cont}$. Then $\sigma = \mathcal{P}(\sigma')$, and for $s = \langle U, V, W \rangle := \delta_0(\sigma)$ we have $cont(\sigma) = cont_\mathcal{C}(s)$. Moreover, by construction of $\mathcal{A}_{cont}$ and $\mathcal{B}$ we have $q' \xrightarrow{w} q$, for some $q' \in U \cup V \cup W$ and where w is the maximal suffix of σ' consisting of letters from Σ_{uo}. Since $cont(\sigma)$ is admissible for s and by Definition 11 (ii), there exists an $a \in cont(\sigma)$ such that $q \xrightarrow{a} q''$ for some q'', so by Definition 5, $\mathcal{A}_{cont}$ is live.

2. Let $\sigma \in \mathcal{P}(\mathcal{L}^\omega(\mathcal{A}_{cont}))$, let $\rho = (s_i)_{i\geq 0}$ its unique run in $\mathcal{B}'$, and denote by w_i, for $i \geq 0$, the prefix of σ of length i. By construction of $\mathcal{A}_{cont}$, we have that $w_{i+1} = w_i a$ iff $a \in cont(w_i) \cap \Sigma_o$, by construction of $\mathcal{B}'$, $s_{i+1} = \delta(s_i, a)$, and by construction, $cont(s_i) \cap \Sigma_o \in adm(s_i)$ and $\theta(s_i) = \langle s_i, cont(s_i) \cap \Sigma_o \rangle$. Therefore, the sequence $s_0, \langle s_0, \theta(s_0)\rangle, \langle s_0, a_1\rangle, s_1, \ldots$ is a play of $\mathcal{G}(\mathcal{B})$ that adheres to θ. Since by assumption that play is winning, $s_i \in F$ for infinitely many i, so $\sigma \in \mathcal{L}(\mathcal{B})$, and by Theorem 6, σ is non-ambiguous.

3. For $s = \langle U, V, W\rangle := \delta_0(\sigma)$ we have by construction of $\mathcal{B}$ that (a) $q \in U$ iff there exists $w \in \mathcal{P}^{-1}(\sigma) \cap \mathcal{L}^*(\mathcal{A}) \cap (\Sigma \setminus \{f\})^*$ and (b) $q \in V \cup W$ iff there exists $w \in \mathcal{P}^{-1}(\sigma) \cap \mathcal{L}^*(\mathcal{A}) \cap \Sigma^* f \Sigma^*$. Now, $diag(\sigma) = diag(s) = \top$ iff $U = \emptyset$, which by the above and Definition 3 is equivalent to saying that σ is surely faulty.

■

We are now in a position to state the main result of this section.

Theorem 7 *Let $\mathcal{A}$ be an LTS with n states and m controllable actions. The active diagnosis decision and synthesis problems for $\mathcal{A}$ can be solved in $2^{\mathcal{O}(n+m)}$ time. Moreover, if $\mathcal{A}$ is actively diagnosable, then one can synthesize a state-based pilot C with at most 6^n states such that h_C is an active diagnoser for $\mathcal{A}$.*

Proof. Lemma 1 and Lemma 2 imply that $\mathcal{A}$ is actively diagnosable iff there is a positional winning strategy for s_0 in $\mathcal{G}(\mathcal{B})$, and the second part of the theorem follows from Lemma 2 and Theorem 6. As for the complexity statement, we note that the game $\mathcal{G}(\mathcal{B})$ has $\mathcal{O}(6^n \cdot 2^m)$ vertices and edges, and a winning strategy can be computed in polynomial time in the size of the game [6], which gives the result. ■

We briefly discuss the relationship of our construction with that of [10]. There, an active diagnoser is built on the basis of a powerset construction that is similar to ours but without splitting the possibly faulty states into a 'watchlist' W and a 'waiting room' V. However, they then face the aforementioned problem of distinguishing sequences with infinitely many ambiguous prefixes (like a^ω in Example 1) from truly ambiguous sequences (like b^ω), which they resolve by examining each cycle of the automaton. Since the number of states in that automaton is 3^n,[1] and there can be exponentially many cycles, this procedure is doubly exponential in n. Our construction is only singly exponential in n.

Using Theorems 1 and 7, we get the following corollary.

Corollary 1 *The active diagnosis decision problem is EXPTIME-complete.*

4.3 Index and waiting time

We assume that $\mathcal{A}$ is actively diagnosable and develop the construction of an active diagnoser with a delay close to the index of $\mathcal{A}$, and a computational complexity still in $2^{\mathcal{O}(n)}$. For

1. This is the result when only one fault type is considered; [10] actually provides for several fault types, which we omit here for sake of simplicity.

simplicity, we denote the game $\mathcal{G}(\mathcal{B})$ by $\mathcal{G}$. Let $\mathcal{G}'$ be any game. Given a strategy θ for $\mathcal{G}'$, we denote by $Play^{\omega}_{\theta}(\mathcal{G}')$ the set of plays that adhere to strategy θ, and by $R(\theta)$ the subset of states of S that are visited by a play of $Play^{\omega}_{\theta}(\mathcal{G}')$. We are now in the position to introduce the main concept of this section, the *waiting time* of a strategy: the maximal number of states visited without encountering an accepting state.

Definition 13 (waiting time) *Let θ be a strategy for $\mathcal{G}$. Then the* waiting time $K(\theta)$ *is defined as* $\sup(|\{k \mid i \leq k \leq j \wedge \rho(k) \in S\}| \mid \exists i, j\ \exists \rho \in Play^{\omega}_{\theta}(\mathcal{G})\ F \cap \{\rho(k)\}_{i \leq k \leq j} = \emptyset)$ *with the convention* $\sup(\emptyset) = 0$.

Observe that $K(\theta)$ may be infinite for a non-positional winning strategy. However, it is finite and strictly smaller than $|S|$ (since there is at least one accepting state) for a winning positional strategy. In fact, for θ a winning positional strategy, $K(\theta)$ can be computed in linear time (with appropriate data structures) w.r.t. the size of $\mathcal{G}$. In order to present it and for subsequent use, we introduce the following notation. Let s be a state of the Büchi automaton, $Out(s) := \{a \in \Sigma_o \mid \delta(s, a) \text{ is defined }\}$.

First one computes, by increasing values, the minimal solution of the following equation system:

$$V_{\theta}(s) = \begin{cases} 0 & \text{if } s \in R(\theta) \cap F; \\ 1 + \max(V_{\theta}(\delta(s,a)) \mid a \in \Sigma' \cap Out(s) \text{ s.t. } (s, \Sigma') = \theta(s)) & \text{if } s \in R(\theta) \setminus F. \end{cases}$$

Then $K(\theta) = \max(V_{\theta}(s) \mid s \in R(\theta))$.

Denote by $D(\theta)$ the delay of the active diagnoser related to strategy θ. The next lemma shows that $K(\theta)$ provides useful information about $D(\theta)$.

Lemma 3 *Let θ be a strategy for game $\mathcal{G}$ with finite waiting time. Then:*

$$1 + K(\theta) \leq D(\theta) \leq 1 + 2K(\theta)$$

Proof. Intuitively, the upper bound is potentially due to a fault staying in the "waiting room" of $\mathcal{B}$ for at most $K(\theta)$ steps, then in the "watchlist" for at most $K(\theta) + 1$ steps. The lower bound is due to the fact that along a subrun with a non-empty watchlist, a possible fault could have occurred before this subrun. Formally, let us denote $h_{\mathcal{C}(\theta)} = \langle cont, diag \rangle$ the active diagnoser associated with strategy θ.

"Upper bound:" We first prove that $D(\theta) \leq 1 + 2K(\theta)$. Let $\sigma = \sigma' f \sigma'' \in \mathcal{L}^*(\mathcal{A}_{cont})$ be an arbitrary faulty sequence with $M := |\sigma''|_{\Sigma_o} \geq 2K(\theta) + 1$. Consider the state $s' = \delta_0(\mathcal{P}(\sigma'))$ denoted by $\langle U', V', W' \rangle$, and for $1 \leq i \leq M$, the states $s_i = \delta_0(\mathcal{P}(\sigma' f \sigma_i))$ denoted by $\langle U_i, V_i, W_i \rangle \in S$ and some q_i with $q_0 \overset{\sigma' f \sigma_i}{\Longrightarrow} q_i$, where σ_i is the minimal prefix of σ'' with $|\sigma_i|_{\Sigma_o} = i$. Notice that it suffices to show $U_j = \emptyset$ for some $j \leq 2K(\theta) + 1$ to prove the desired property. There are three possibilities:

- Either $W' = \emptyset$, then by construction of $\mathcal{B}$ we have $q_i \in W_i$ for $i = 1$ and indeed for all $i \leq M$ since $\mathcal{A}_{cont}$ is live. Then by assumption on $K(\theta)$, $s_k \in F$ for some $k \leq K(\theta) + 1$. Since $W_k \neq \emptyset$, this means that $U_k = \emptyset$, so we set $j := k$.
- Or $W' \neq \emptyset$ and $U' = \emptyset$, then we set $j := 1$.

– Or $W' \neq \emptyset$ and $U' \neq \emptyset$, so $s' \notin F$. then $q_1 \in V_1$, i.e. in the waiting room. Let $k \in \{1, \ldots, K(\theta)\}$ be the minimal value with $s_k \in F$, then $q_k \in V_k$. If $U_k = \emptyset$, we set $j := k$. Otherwise $W_k = \emptyset$. Then the successor of q_k will be "transferred" to the watchlist in the next step, i.e. $q_i \in W_i$ for $i > k$. Using again the definition of $K(\theta)$, there is $s_{k'} \in F$ for some $k < k' \leq k + K(\theta) + 1 \leq 2K(\theta) + 1$, and since $W_{k'} \neq \emptyset$, this means $U_{k'} = \emptyset$, so we set $j := k'$.

"Lower bound:" We now prove that $1 + K(\theta) \leq D(\theta)$. Let $\sigma = \sigma' a_1 \cdots a_{K(\theta)} \in \mathcal{P}(\mathcal{L}^*(\mathcal{A}_{cont}))$, and $t_i = \langle U_i, V_i, W_i \rangle := \delta_0(\sigma' a_1 \cdots a_i)$, for $i = 0, \ldots, K(\theta)$, where $t_0 \in F$ and $t_1, \ldots, t_{K(\theta)} \notin F$. Such a word exists by assumption on $K(\theta)$.
First, $t_{K(\theta)} \notin F$ implies $U_{K(\theta)}, W_{K(\theta)} \neq \emptyset$. Since $U_{K(\theta)} \neq \emptyset$, there is a non faulty sequence $w_1 \in \mathcal{A}_{cont}$ ending in some state of $U_{K(\theta)}$ with $\mathcal{P}(w_1) = \sigma$.
On the other hand, since $W_i \neq \emptyset$ for all $0 < i \leq K(\theta)$, there is a faulty sequence $w_2 = w' f w''$ with $\mathcal{P}(w_2) = \sigma$ and where $|w''|_{\Sigma_o} \geq K(\theta)$ (i.e. a fault that occurs "between" t_0 and t_1 at the latest). But this implies that $h_{\mathcal{C}(\theta)}$ admits the observable sequence σ but cannot diagnose it as surely faulty even $K(\theta)$ observations after the possible occurrence of f in w_2. ■

Define $K_{\mathcal{A}} = \min(K(\theta))$, where θ ranges over the winning strategies for $\mathcal{G}$. Since a positional such strategy exists, we know that $K_{\mathcal{A}}$ is finite and belongs to $2^{\mathcal{O}(n)}$. Let us note $D_{\mathcal{A}} = \min(D(\theta))$ the index of $\mathcal{A}$. The following corollary provides a tight frame for $D_{\mathcal{A}}$ and shows that the index is in $2^{\mathcal{O}(n)}$.

Corollary 2 *Let $\mathcal{A}$ be actively diagnosable. Then:* $1 + K_{\mathcal{A}} \leq D_{\mathcal{A}} \leq 1 + 2K_{\mathcal{A}}$

Let us compute an active diagnoser or, equivalently, a strategy θ that achieves $K(\theta) = K_{\mathcal{A}}$. To this aim, we introduce a family of games $(\mathcal{G}_i)_{i \in \mathbb{N}}$ defined as follows. The set of vertices of $\mathcal{G}_i$ are: $V_{\mathcal{G}_i} = \{v^j \mid v \in V_{\mathcal{G}} \wedge 0 \leq j \leq i\} \cup \{lost\}$ where the subset of vertices owned by Control are $\{v^j \mid v \in V_C \wedge 0 \leq j \leq i\} \cup \{lost\}$, the initial vertex is s_0^0, and the set of accepting states are $\{s^0 \mid s \in F\}$. Its set of edges $E' = E'_1 \cup E'_2 \cup E'_3$ is defined by:

– for all $j \leq i$, $\langle v^j, w^j \rangle$ belongs to E'_1 iff $\langle v, w \rangle$ belongs to E_1;
– for all $j \leq i$, $\langle v^j, w^j \rangle$ belongs to E'_2 iff $\langle v, w \rangle$ belongs to E_2;
– for all $j \leq i$, $\langle \langle s, a \rangle^j, s'^0 \rangle \in E'_3$ iff $\langle \langle s, a \rangle, s' \rangle \in E_3$ and $s' \in F$;
– for all $j < i$, $\langle \langle s, a \rangle^j, s'^{j+1} \rangle \in E'_3$ iff $\langle \langle s, a \rangle, s' \rangle \in E_3$ and $s' \notin F$;
– $\langle \langle s, a \rangle^i, lost \rangle$ belongs to E'_3 iff $\langle \langle s, a \rangle, s' \rangle$ belongs to E_3 and $s' \notin F$;
– $\langle lost, lost \rangle$ belongs to E'_3 and there is no other edge.

Game $\mathcal{G}_i$ has the following properties: an infinite play either ends up in *lost* or visits the accepting states infinitely often, with at most i visits of the set $\{v^j \mid v \in S \setminus F,\ 0 \leq j \leq i\}$ between two visits of accepting states. The following lemma relates strategies in $\mathcal{G}$ and $\mathcal{G}_i$. Based on it an efficient computation of an optimal strategy w.r.t. $K(\theta)$ can be performed.

For the remainder, we use the following notations to classify the states according to their behaviour in $\mathcal{G}_i$. Let WS_i be the set of states that are part of a winning strategy with waiting time i, i.e. $s \in WS_i$ if there exists a strategy θ with $K(\theta) = i$ and $s \in R(\theta)$. Obviously, $K_{\mathcal{A}}$ is the smallest i such that $WS_i \neq \emptyset$.

Lemma 4 *There is a winning strategy θ in $\mathcal{G}$ with $K(\theta) \leq i$ iff there is a winning strategy θ_i in $\mathcal{G}_i$. Moreover, in the positive case, θ can be chosen to be positional.*

Proof. Let θ be a winning strategy of $\mathcal{G}$ with $K(\theta) \leq i$. Let $\rho_i = v_0^{\alpha(0)} \cdots v_n^{\alpha(n)}$ be a play (not visiting *lost*) in $\mathcal{G}_i$ with $v_n \in S$. Define θ_i by: $\theta_i(\rho_i) = \theta(v_0 \cdots v_n)$. Now a finite play $\rho_i = v_0^{\alpha(0)} \cdots v_n^{\alpha(n)}$ that adheres to θ_i corresponds to the play $\rho = v_0 \cdots v_n$ that adheres to θ and $\alpha(n)$ is the number of consecutive states of $S \setminus F$ without visiting F up to v_n. Since $K(\theta) \leq i$ such a play will never visit *lost* at the next state. So all the infinite plays of G_i that adhere to θ_i are $\rho_i = v_0^{\alpha(0)} \cdots v_n^{\alpha(n)} \cdots$ with $\rho = v_0 \cdots v_n \cdots$ a play that adheres to θ and $\alpha(n)$ the number of consecutive states of $S \setminus F$ without visiting F up to v_n. This proves that such plays are winning in $\mathcal{G}_i$.

Let θ_i be a winning strategy of $\mathcal{G}_i$. Since $\mathcal{G}_i$ is a Büchi game, w.l.o.g. we assume that θ_i is positional. For $v \in S$ such that $\{v^j\}_{j \leq i} \cap R(\theta_i) \neq \emptyset$, define $m(v) = \max\{ j \mid v^j \in R(\theta_i) \}$. We denote the subset of such states by S'. Let us (partially) define the positional strategy θ'_i by $\theta'_i(v^j) = \theta_i(v^{m(v)})$ for $v \in S'$. In order to prove that θ'_i is well-defined we show by induction that $v^j \in R(\theta'_i) \cap S$ implies that $v \in S'$ and $j \leq m(v)$. The only interesting case is the one of a finite play ρ that adheres to θ'_i ends by: $v^j \langle v^j, \Sigma' \rangle \langle v^j, a \rangle$ with $\delta_o(v, a) = v'$. Assume that $v \in S'$. This means that $v^{m(v)} \in R(\theta_i)$ and $j \leq m(v)$. So $\Sigma' = \theta_i(v^{m(v)})$ and from $v^{m(v)}$ adhering to θ_i one can reach $\langle v^{m(v)}, a \rangle$ and since $\delta_o(v, a) = v'$, one reaches either v'^0 if $v' \in F$ or $v'^{m(v)+1}$ if $v' \notin F$. So from $\langle v^j, a \rangle$ one reaches either v'^0 if $v' \in F$ or v'^{j+1} if $v' \notin F$ since $j + 1 \leq m(v) + 1 \leq m(v') \leq i$. In both cases, the induction is proved. Thus an infinite play adhering to θ'_i will never reach *lost*. Due to the preliminary observation θ'_i is winning strategy.

Now let us (partially) define in $\mathcal{G}$ the positional strategy $\theta(v) = \theta'_i(v^j)$ for $v \in S'$ and an arbitrary $j \leq m(v)$ (since it is irrelevant). A play $\rho = v_0 \cdots v_n \cdots$ that adheres to θ corresponds to a play $\rho_i = v_0^{\alpha(0)} \cdots v_n^{\alpha(n)} \cdots$ that adheres to θ'_i with $\alpha(n)$ the number of consecutive states of $S \setminus F$ without visiting F up to v_n. So θ is well-defined and it is a winning strategy with $K(\theta) \leq i$. ■

Theorem 8 *If $\mathcal{A}$ is actively diagnosable, there exists a positional strategy θ that fulfills $K(\theta) = K_{\mathcal{A}}$. Moreover, such a strategy can be computed in $2^{\mathcal{O}(n)}$.*

Proof. Using Lemma 4, one knows that there is positional strategy θ that fulfills $K_{\mathcal{A}} = \min(K(\theta))$. The synthesis algorithm consists to look for a winning strategy in $\mathcal{G}_i$ by increasing values of i starting from $i = 0$ and stop as soon as such a strategy is found. Since i cannot reach $|S|$, the size of the game $\mathcal{G}_i$ is quadratic w.r.t. the size of $\mathcal{G}$. For the same reason, the number of iterations is bounded by the size of $\mathcal{G}$. So the positional winning strategy θ is found in polynomial time w.r.t. the size of $\mathcal{G}$ i.e. still in $2^{\mathcal{O}(n)}$. ■

Due to Theorem 4, this construction represents a reasonable tradeoff, since an active diagnoser that realizes a delay equal to the index of $\mathcal{A}$ may need to be much larger, i.e. $2^{\Omega(n \log(n))}$. We sketch the construction of a controller with minimal delay once one knows that the system is actively diagnosable. One iteratively builds a safety game $\mathcal{G}'_i$ parametrized by increasing values of i. A controller state of this game is defined by (U, d) where U is the set of states reached by a correct sequence while d (defined when $U \neq \emptyset$) associates with every state s reached by a faulty sequence a duration $d(s) \leq i+1$ since the occurrence of the earliest fault that would lead to s. As in the previous games the controller selects a subset of observable actions letting the environment select an action among them. The aim of the controller is to avoid states with some $d(s) = i+1$. The first i for which $\mathcal{G}'_i$ has a winning strategy is the index and the winning strategy yields an active diagnoser with minimal delay. Observe that since the index is bounded by $2^{\mathcal{O}(n)}$, in the worst case the final game has $2^{\mathcal{O}(n^2)}$ states.

5 Parametrized active diagnosis

In this section, we discuss a parametrized version of the synthesis problem for active diagnosers. Consider once again the example from Figure 1. As we have already seen in Section 2, it is possible to construct active diagnosers with a delay of k, for every $k \geq 2$, where such a diagnoser can admit at most $k-2$ consecutive occurrences of b. This example shows that there is a certain trade-off between the permissivity of the control component of the active diagnoser and the delay to diagnose a fault. In the following, we propose the construction of a parametrized active diagnoser in which the user can determine this trade-off by fixing the value of the parameter.

Fix an LTS $\mathcal{A} = \langle Q, q_0, \Sigma, T \rangle$, the corresponding Büchi automaton $\mathcal{B} = \langle \mathcal{B}', F \rangle$, with $\mathcal{B}' = \langle S, s_0, \Sigma_o, \delta \rangle$, and $\mathcal{G} := \mathcal{G}(\mathcal{B})$ as in the previous sections.

Definition 14 (permissiveness) *Let $h = \langle cont, diag \rangle$ and $h' = \langle cont', diag' \rangle$ be two pilots for $\mathcal{A}$. Then h is said to be more* permissive *than h' (written $h \succ h'$) if $\mathcal{L}^*(\mathcal{A}_{cont'}) \subseteq \mathcal{L}^*(\mathcal{A}_{cont})$.*

For instance, in Example 1, $h_k \succ h_{k-1}$, but the delay of h_k is $k+2$ while that of h_{k-1} is only $k+1$.

In Section 4.3, we have established a factor of 2 between the waiting time and the delay of our active diagnosers. We shall therefore construct, for some starting value d_0, a family $(\mathcal{C}_d)_{d \geq d_0}$ of state-based pilots such that for all $d \geq d_0$: (i) $h_{\mathcal{C}_d}$ is a $(2d+1)$-active diagnoser for $\mathcal{A}$, and (ii) $h_{\mathcal{C}_d} \succ h$ for any $(d+1)$-active diagnoser h of $\mathcal{A}$. Notice that in item (ii), h can be any active diagnoser, whether it is based on a finite-state pilot or not.

Let $WS = \bigcup_{i \geq 0} WS_i$, and $K^+_{\mathcal{A}} (< |S|)$ the smallest i such that $WS = WS_i$. As for the starting value, we shall first present a solution for $d_0 = K^+_{\mathcal{A}}$; there, we represent the entire family $(\mathcal{C}_d)_{d \geq d_0}$ in the form of a single, so-called parametrized counter LTS. A solution for $d_0 = K_{\mathcal{A}}$, conceptually similar but requiring more overhead, is presented in Section 5.1.

Definition 15 (parametrized counter LTS) *A* parametrized counter LTS *(p-LTS) is a tuple $\mathcal{A}_{\mathsf{p}} = \langle R, r_0, \Sigma', T' \rangle$ with a finite set R of states, initial state $r_0 \in R$, and transitions $T' \subseteq R \times Op \times \Sigma' \times R$, which are labelled with letters from Σ' and an additional* counter operation, *where $Op = \{\top\} \cup \mathbb{N}_{\geq 1}$.*

The concrete semantics of $\mathcal{A}_{\mathsf{p}}$ for a value d is an LTS equipped with a finite counter, i.e. $\mathcal{A}_{\mathsf{p}}(d) := \langle R \times \{1, \ldots, d+1\} \cup \{\bot\}, \langle r_0, d+1 \rangle, \Sigma', T'_d \rangle$, where $\langle \langle r, x \rangle, a, q \rangle \in T'_d$ iff there exists $\langle r, o, a, r' \rangle \in T'$ such that

- *either* $o = \top$ *and* $q = \langle r', d+1 \rangle$;
- *or* $o = c \in \mathbb{N}_{\geq 1}$, $x > c$, *and* $q = \langle r', x-1 \rangle$;
- *or* $o = c \in \mathbb{N}_{\geq 1}$, $x \leq c$, *and* $q = \bot$.

Moreover, $\langle \bot, a, \bot \rangle \in T'_d$ *for all* $a \in \Sigma'$.

$\mathcal{A}_\mathsf{p}$ is called *deterministic* if for every $r \in R$ and $a \in \Sigma'$ there is at most one transition $(r, o, a, r') \in T'$, and in this case every $\mathcal{A}_\mathsf{p}(d)$ is also deterministic. The "garbage state" $\bot$ collects all inputs where the guard on the counter value x is unsatisfied at some point.

We shall now synthesize such a p-LTS $\mathcal{B}_\mathsf{p}$ from the Büchi automaton $\mathcal{B}$ and the game $\mathcal{G}$. It uses the states of $\mathcal{B}$, and for any $d \geq d_0$, $\mathcal{B}_\mathsf{p}(d)$ can serve as the LTS component of a state-based pilot for $\mathcal{A}$, where no run stays outside the accepting states F for more than d steps. For this latter property, consider the following equation system, whose variables are the elements of S.

$$V(s) = \begin{cases} \infty & \text{if } s \notin WS; \\ 0 & \text{if } s \in F \cap WS; \\ 1 + \min_{\Sigma' \in adm(s)} \max_{a \in \Sigma' \cap Out(s)} V(\delta(s,a)) & \text{otherwise} \end{cases} \tag{1}$$

Let v^* denote the smallest solution for V. Recall that states outside WS can never be touched by any winning strategy. It is therefore easy to see that when $v^*(s) = k$ for $s \notin F$, then the "fastest" strategy for Control to force $\mathcal{G}$ back into F requires k further observations in the worst case. Moreover, the largest value in v^* equals $K^+_\mathcal{A}$. The guards of $\mathcal{B}_\mathsf{p}$ will take v^* into account.

Now, let $\mathcal{B}_\mathsf{p} := \langle S, s_0, \Sigma_o, \delta' \rangle$, where $\langle s, o, a, s' \rangle \in \delta'$ if $\langle s, a, s' \rangle \in \delta$ and either $s' \in F$ and $o = \top$, or $s' \notin F$ and $o = v^*(s')$.

A controller for a concrete value $d \geq K^+_\mathcal{A}$ is now easily derived from $\mathcal{B}_\mathsf{p}$ by taking $\mathcal{C}_d := \langle \mathcal{B}_\mathsf{p}(d), cont_d, diag_d \rangle$, where, for all states $q = \langle \langle U, V, W \rangle, x \rangle$ of $\mathcal{B}_\mathsf{p}(d)$ we set $cont_d(q) = \{\, a \mid \exists q' \in WS \times \mathbb{N} : q \xrightarrow{a} q' \,\} \cup \Sigma_{uc} \cup \Sigma_{uo}$ and $diag_d(q) = \top$ iff $U = \emptyset$. (The values for state $\bot$ are immaterial.)

Example 7 *Consider the LTS of Figure 1, for which the deterministic Büchi automaton $\mathcal{B}$ is shown in Figure 8. The values of v^* in $\mathcal{B}$ are* 0 *for the accepting states and* 1 *or* 2 *for the non-accepting states. Figure 10 shows the automaton with states renamed for simplicity and their v^* values as annotations. The automaton $\mathcal{B}_\mathsf{p}$ has exactly the structure of $\mathcal{B}$, where transitions into accepting states are guarded by $\top$ and transitions into the non-accepting states by their value* 1 *and* 2*, respectively. In this example, we have $K^+_\mathcal{A} = 2$. Thus, in $\mathcal{B}_\mathsf{p}(3)$, for instance, the initial state would be $\langle s_0, 4 \rangle$, allowing to move into $\langle s_2, 3 \rangle$ with b. In $\langle s_2, 3 \rangle$, the control allows $\{a, b\}$, and if the environment chooses b, that would lead us to $\langle s_2, 2 \rangle$. In that state, however, the value of the counter obliges the controller to block action b and force an a, going to $\langle s_1, 1 \rangle$. In this case, the actual delay of the controller is* 4 *(realised, for instance, by the observation bbac). Notice that, for uniformity, the transitions with uncontrollable actions (such as c) are also labelled with guards. However, the constructions of $\mathcal{B}_\mathsf{p}$ and $\mathcal{C}_d$ are such that uncontrollable actions will never be blocked.*

The following remark will be useful later:

Remark 2 *The construction of $\mathcal{G}$ and Equation (1) imply that for every state s of WS there exists a set $\Sigma' \subseteq \Sigma$ such that Σ' is admissible for s, for every $a \in \Sigma' \cap Out(s)$, $\delta(s,a) = s' \in WS$, and additionally, (i) if $s \notin F$ and $v^*(s) = k$, then $v^*(s') < k$; (ii) if $s \in F$, then $v^*(s') \leq K^+_{\mathcal{A}}$.*

Theorem 9 (parameterized active diagnosis) *Let $\mathcal{A}$ be an actively diagnosable LTS with n states. Then $\mathcal{B}_\mathsf{p}$ can be computed in $2^{\mathcal{O}(n)}$ time, and for all $d \geq K^+_{\mathcal{A}}$,*

1. *$\mathcal{C}_d$ is a state-based pilot for $\mathcal{A}$;*
2. *$h_{\mathcal{C}_d} = \langle cont, diag \rangle$ is a $(2d+1)$-active diagnoser for $\mathcal{A}$;*
3. *and $h_{\mathcal{C}_d} \succ h$ for any $(d+1)$-active diagnoser of $\mathcal{A}$.*

Proof. The complexity result follows mostly from previous results (Theorem 6 and Theorem 7), coupled with the fact that v^* can be obtained by a Kleene fixpoint computation in at most $K^+_{\mathcal{A}}$ iterations, where $K^+_{\mathcal{A}}$ is trivially bounded by the number of non-accepting states, i.e. less than 6^n.

As for the remaining items, let us first denote, for $\sigma \in \Sigma_o^*$, the unique state q such that $\langle s_0, d+1 \rangle \stackrel{\sigma}{\Rightarrow} q$ in $\mathcal{B}_\mathsf{p}(d)$ by $\delta_0'(\sigma)$. We then prove:

1. Observe that $\mathcal{C}_d$ fulfils requirements (1) and (2) of Definition 7 by construction. Moreover, we need to show that $h_{\mathcal{C}_d}$ fulfils the three conditions from Definition 6.

   (i) $\mathcal{A}_{cont}$ is live: Let $\sigma' \in \Sigma^*$ and $\langle \varepsilon, q_0 \rangle \stackrel{\sigma'}{\Rightarrow} \langle \sigma, q \rangle$ in $\mathcal{A}_{cont}$. Then $\sigma = \mathcal{P}(\sigma')$. We prove that the following statements are invariant by induction over the length of σ:

   (I) $\delta_0'(\sigma) = \langle s, x \rangle$ for $s = \delta_0(\sigma)$, $s \in WS$, and some $x > 0$;
   (II) $s \in F$ iff $x = d+1$;
   (III) if $s \notin F$ then $x \geq v^*(s)$;
   (IV) $cont(\langle s, x \rangle)$ is admissible for s.

   The liveness statement then follows from the last item like in the proof of Lemma 2, part 1.

   Certainly (I)–(III) hold for $\sigma = \varepsilon$, where $s = s_0 \in F$ and $x = d+1$. All outgoing edges from s_0 in $\mathcal{B}_\mathsf{p}$ are guarded either by $\top$ or by a value $v^*(s') \leq K^+_{\mathcal{A}} < d+1$, so $cont(\langle s_0, d+1 \rangle)$ authorizes all actions that remain in WS. Then (IV) follows from Remark 2.

   For the induction step, suppose that (I)–(IV) hold for σ, and we shall prove it for $\sigma' = \sigma a$, for some $\sigma a \in \mathcal{P}(\mathcal{L}^*(\mathcal{A}_{cont}))$, and let $\delta_0'(\sigma') = \langle s', x' \rangle$. Then (I) follows from the construction of $\mathcal{B}_\mathsf{p}$ and the fact that $a \in cont(\langle s, x \rangle)$ implies $\delta(s,a) \in WS$. (II) and (III) follow immediately from the construction of $\mathcal{B}_\mathsf{p}$. Finally, (IV) follows again from Remark 2.

   (ii) This follows from the fact that the counter x is decremented whenever $\mathcal{B}_\mathsf{p}(d)$ is in a state $\langle s, x \rangle$ with $s \notin F$; for $x = 1$, the only remaining possibility is to choose an admissible set that forces the run into F. Since the maximal value of F is $d+1$, a run can stay outside F for at most d consecutive steps.

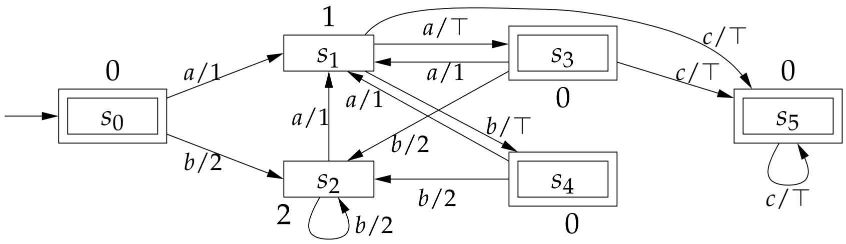

Figure 10: The p-LTS $\mathcal{B}_p$ arising from Figure 1.

(iii) $diag(\sigma) = \top$ iff σ is surely faulty: obvious by construction, cf the proof of Lemma 2, part 3.

2. Let $\theta_{h_{\mathcal{C}_d}}$ be the strategy obtained from $h_{\mathcal{C}_d}$ in Lemma 1. Then, by item 1 (ii) in this proof, $K(\theta_{h_{\mathcal{C}_d}}) \leq d$. The statement then follows from Lemma 3.

3. Let $h = \langle cont', diag' \rangle$ be a $(d+1)$-active diagnoser and suppose by contradiction that $h_{\mathcal{C}_d} \succ h$ does not hold. Then let us choose a minimal (by prefix order) word σ' from $\mathcal{L}^*(\mathcal{A}_{cont'}) \setminus \mathcal{L}^*(\mathcal{A}_{cont})$ (cf. Definition 14). Since $cont$ can only block controllable actions, we have $\mathcal{P}(\sigma') = \sigma a$ (for some $\sigma \in \Sigma_o^*$, $a \in \Sigma_o$ such that $a \notin cont(\sigma)$. According to the proof of item 1, we know that $cont(\sigma) = cont_d(\langle s, x \rangle)$, where $\langle s, x \rangle = \delta_0'(\sigma)$. So either $s' := \delta(s, a) \notin WS$, but then h cannot avoid all ambiguous sequences and is not an active diagnoser. Or $s, s' \notin F$ and $v^*(s') = x' \geq x$. Notice the run of σ in $\mathcal{B}$ has already seen $d + 1 - x$ consecutive non-accepting states, s' adds one more, and from s' the environment can force us to see $x - 1$ more such states before returning to F. So if θ_h is the strategy obtained from h, then $K(\theta_h) \geq (d + 1 - x) + 1 + (x - 1) = d + 1$. So by Lemma 3, $D(\theta_h) \geq d + 2$, so h is not a $(d+1)$-diagnoser. Thus, the initial contradiction is impossible.

■

5.1 Extending the parametrized construction

We briefly sketch how the synthesis of the family (C_d) can be extended to values $K_{\mathcal{A}} \leq d < K_{\mathcal{A}}^+$. The general idea is the same, but is less conveniently represented in parametrized fashion.

In the construction for $d \geq K_{\mathcal{A}}^+$, the control admits actions that let $\mathcal{C}_d$ remain in states from WS, while making sure that the runs return to states of F in "due time". For values between $K_{\mathcal{A}}$ and $K_{\mathcal{A}}^+$, two things change:

- The control must keep the runs of $\mathcal{C}_d$ in states of $WS_d \subset WS$; indeed states in $WS \setminus WS_d$ cannot assure a waiting time of d.
- As a result, certain states of WS_d must change their strategy. Consider Figure 11, displaying a hypothetical Büchi automaton, and suppose that in state s_1 we can choose to block either action a or b (but not both). Then state s_1 has a strategy to go to an

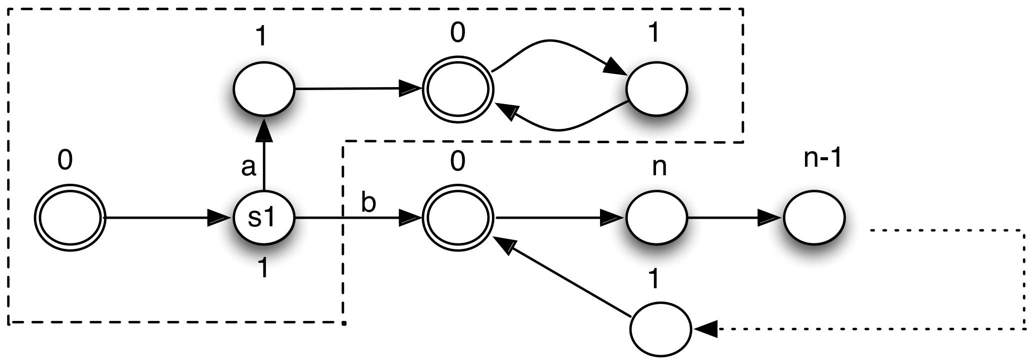

Figure 11: Illustrating the extended parametrized construction.

accepting state in one single step by blocking a, hence $v^*(s_1) = 1$; the v^* values are annotated next to each state.
However, when we want to construct $\mathcal{C}_2$ in this example, then we must remain in the set WS_2, indicated by the dashed box. Now, blocking a is no longer an option for s_1; instead, it must block b and can reach F in only two steps. We must therefore replace Equation (1) to take into account the restricted choices:

$$V_d(s) = \begin{cases} \infty & \text{if } s \notin WS_d; \\ 0 & \text{if } s \in F \cap WS_d; \\ 1 + \min_{\Sigma' \in adm(s)} \max_{a \in \Sigma' \cap Out(s)} V_d(\delta(s,a)) & \text{otherwise} \end{cases} \tag{2}$$

The procedure now works as follows for a given d:

1. Solve Equation (2) to obtain the minimal solution v_d^*.
2. Create $\mathcal{B}_\mathsf{p}$ as before, but replace all labels $v^*(s')$ by $v_d^*(s')$.
3. With the modified version of $\mathcal{B}_\mathsf{p}$, synthesize $\mathcal{C}_d$ as before, but setting $cont_d(q) = \{ a \mid \exists q' \in WS_d \times \mathbb{N} : q \xrightarrow{a} q' \}$.

In particular, this means that the construction of $\mathcal{B}_\mathsf{p}$ is no longer independent of d, unlike for the case $d_0 = K_\mathcal{A}^+$.

6 Conclusion and Perspectives

We have developed an active-diagnosis method for finite-state systems, shown it to be optimal w.r.t. several criteria, and developed a parametrized version of active diagnosis that allows to obtain a controller achieving an almost optimal tradeoff between permissivity and delay.

Future work has several important research leads to address. First, it remains to determine the precise memory requirement for the minimal-delay diagnoser since our results show that it lies between $2^{\Theta(n \log(n))}$ and $2^{\Theta(n^2)}$. In another lead, the control used for active diagnosis could be refined into a *safe* control, i.e. one that does not "encourage" the faulty behaviours. Finally we aim at addressing infinite-state systems or systems with quantitative features, as for passive diagnosability in pushdown systems [8], Petri nets [2], timed [15, 13] and probabilistic systems [12].

References

[1] D. Berwanger and L. Doyen. On the power of imperfect information. In *Proc. FSTTCS*, volume 2 of *LIPICS*, Bangalore, India, 2008.

[2] M.P. Cabasino, A. Giua, S. Lafortune, and C. Seatzu. Diagnosability analysis of unbounded Petri nets. In *CDC09: 48th IEEE Conf. on Decision and Control*, pages 1267–1272, 2009.

[3] C. G. Cassandras and S. Lafortune. *Introduction to Discrete Event Systems - Second Edition*. Springer, 2008.

[4] F. Cassez and S. Tripakis. Fault diagnosis with static and dynamic observers. *Fundamenta Informaticae*, 88:497–540, 2008.

[5] E. Chanthery and Y. Pencolé. Monitoring and active diagnosis for discrete-event systems. In *Proc. SafeProcess'09*, 2009.

[6] R. Küsters. *Memoryless Determinacy of Parity Games*, chapter 6, pages 95–106. LNCS 2500. Springer, 2002.

[7] S. Miyano and T. Hayashi. Alternating finite automata on ω-words. *Theoretical Computer Science*, 32:321–330, 1984.

[8] C. Morvan and S. Pinchinat. Diagnosability of pushdown systems. In *Proceedings of the Haifa Verification Conference*, LNCS 6405, 2009.

[9] S. Safra. On the complexity of omega-automata. In *FOCS*, pages 319–327. IEEE, 1988.

[10] M. Sampath, S. Lafortune, and D. Teneketzis. Active diagnosis of discrete-event systems. *IEEE Transactions on Automatic Control*, 43(7):908–929, July 1998.

[11] M. Sampath, R. Sengupta, S. Lafortune, K. Sinnamohideen, and D. Teneketzis. Diagnosability of discrete-event systems. *IEEE Trans. Aut. Cont.*, 40(9):1555–1575, 1995.

[12] D. Thorsley and D. Teneketzis. Diagnosability of stochastic discrete-event systems. *IEEE Transactions on Automatic Control*, 50(4):476–492, 2005.

[13] S. Xu, S. Jiang, and R. Kumar. Diagnosis of dense-time systems using digital clocks. *IEEE Transactions on Automation Science and Engineering*, 7(4):870–878, 2010.

[14] T-S. Yoo and S. Lafortune. Polynomial-time verification of diagnosability of partially observed discrete-event systems. *IEEE Trans. Automat. Contr.*, 47(9):1491–1495, 2002.

[15] S. Hashtrudi Zad, R.H. Kwong, and W.M. Wonham. Fault diagnosis in discrete-event systems: Incorporating timing information. *Trans. Aut. Cont.*, 50(7):1010–1015, 2005.