Active Prediction for Discrete Event Systems

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

Serge Haddad
LSV, ENS Paris-Saclay, CNRS, INRIA, Université Paris-Saclay, France
haddad@lsv.fr

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

Lina Ye
LRI, Université Paris-Saclay, CentraleSupélec, France
lina.ye@lri.fr

Abstract

A central task in partially observed controllable system is to detect or prevent the occurrence of certain events called *faults*. Systems for which one can design a controller avoiding the faults are called *actively safe*. Otherwise, one may require that a fault is eventually detected, which is the task of *diagnosis*. Systems for which one can design a controller detecting the faults are called *actively diagnosable*. An intermediate requirement is *prediction*, which consists in determining that a fault will occur whatever the future behaviour of the system. When a system is not predictable, one may be interested in designing a controller to make it so. Here we study the latter problem, called *active prediction*, and its associated property, *active predictability*. In other words, we investigate how to determine whether or not a system enjoys the active predictability property, i.e., there exists an active predictor for the system.

Our contributions are threefold. From a semantical point of view, we refine the notion of predictability by adding two quantitative requirements: the minimal and maximal delay before the occurence of the fault, and we characterize the requirements fulfilled by a controller that performs predictions. Then we show that active predictability is EXPTIME-complete where the upper bound is obtained via a game-based approach. Finally we establish that active predictability is equivalent to active safety when the maximal delay is beyond a threshold depending on the size of the system, and we show that this threshold is accurate by exhibiting a family of systems fulfilling active predictability but not active safety.

2012 ACM Subject Classification Theory of computation → Formal languages and automata theory; Mathematics of computing → Discrete mathematics

Keywords and phrases Automata Theory, Partially observed systems, Diagnosability, Predictability

Digital Object Identifier 10.4230/LIPIcs.FSTTCS.2020.48

Related Version A full version of the paper is available at [14], `https://hal.archives-ouvertes.fr/hal-02951944`.

Funding *Lina Ye*: This research was done while Lina Ye was on leave at MEXICO team of INRIA.

1 Introduction

Monitoring faulty systems. In monitoring faulty systems, two central tasks consist in detecting a fault that has occurred, resp. will occur, i.e. the tasks of *diagnosis* and *prediction*, respectively, based on observations. However, such tasks may be defeasible due to ambiguity (i.e. observations associated with both correct and faulty runs). In this case, one may

© Stefan Haar, Serge Haddad, Stefan Schwoon, and Lina Ye;
licensed under Creative Commons License CC-BY
40th IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2020).
Editors: Nitin Saxena and Sunil Simon; Article No. 48; pp. 48:1–48:16
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

introduce a *controller* to restrict the behaviour in order to enforce diagnosis (resp. prediction) to be processed. Such a controller is called an *active diagnoser* (resp. *active predictor*). Here we focus on the existence of an active predictor, a problem called *active predictability*.

Diagnosis. In partially observed discrete-event systems, diagnosis was defined and studied in the seminal paper by Sampath et al [17] (see also [6, 7]). That work builds a deterministic version of the original model, a so-called *diagnoser*, that tries to detect the occurrence of faults. A system is called *diagnosable* if the diagnoser can detect every fault occurrence, possibly after some delay. As an illustration, consider the system in Figure 1, which we shall use as a running example, sometimes with different values for Σ_1 and Σ_2, where Σ_1 and Σ_2 are subsets of events in the system. Precisely, $\Sigma_1, \Sigma_2 \subseteq \{a, b, c, d\}$, all of which are observable, while f represents a fault that is not directly observable. If, e.g., a is contained in both Σ_1 and Σ_2, then the system is not diagnosable because any observation ada^n may belong to a faulty run or a correct one.

The diagnosability problem is in PTIME [22], via an approach called *twin-plant construction*. When the system is not diagnosable, it may have to be redesigned, e.g. by adding further sensors to enhance observability, or by forbidding some actions. Sampath et al [16] followed the last approach, called *active diagnosis*: one strives to synthesise a controller, based on partial observations, that forces the behaviour of the system to stay within a diagnosable subset of its behaviours. For instance, if the system in Figure 1 has $\Sigma_1 = \Sigma_2 = \{b\}$ and the controller has the right to block a, then the system is actively diagnosable.

The algorithm for the active-diagnosability problem in [16] operates in doubly exponential time. In [13], we revisited the problem using automata and game theory and established that in fact the active-diagnosability problem is EXPTIME-complete. Later on, we generalised the framework, e.g. allowing the controller to be aware of deadlocks [4]. We also studied active diagnosis for probabilistic systems [1].

In loosely related works. Chanthery and Pencolé [9] proposed a planning-based approach that allows the verdict of the diagnoser to be ambiguous; the works in [8, 10, 20] studied the problem of dynamic sensor activation to ensure some observation properties. In work more closely related to ours [19], Yin and Lafortune proposed a uniform approach for synthesizing property-enforcing supervisor by mapping the considered property to a suitably-defined information state, which is applicable to a class of properties that can be expressed in terms of such information states, including safety, diagnosability, opacity and so on. Note that predictability cannot be formulated as an information state in that framework since it depends also on future behaviours of the system; its enforcement thus requires new methods.

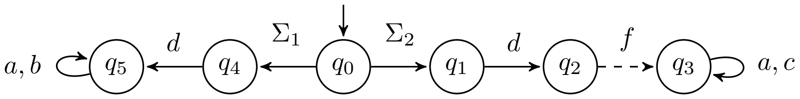

Figure 1 Running example, with unobservable events indicated by dashed lines.

Prediction. Several works have studied the (passive) predictability problem, i.e. where no control is involved. For instance, if $\Sigma_1 = \{b\}$ and $\Sigma_2 = \{c\}$ in Figure 1, then upon first seeing c, an observer can predict that a fault will necessarily occur. In [11], Genc and Lafortune introduced a diagnoser construction to derive a necessary and sufficient condition for predictability in systems modeled by regular languages. Ye, Dague, and Nouioua [18] proposed a polynomial time algorithm for predictability analysis in a centralized way and then extend it to a distributed framework. Brandan Briones and Madalinski [5] introduced and studied two variants of predictability by defining an additional requirement about either a lower bound or an upper bound on the number of events between the fault prediction and the fault occurrence. Then Yin and Li [21] investigated the bounded predictability in the decentralized framework, and proposed a polynomial-time algorithm for its verification. Madalinski and Khomenko [15] reduce the predictability problem for a Petri net to LTL-X model checking. All these previous works focus on passive predictability.

Our contributions. First we refine the paradigm of prediction by allowing an observer to quantify its observations. Unlike [5] but similar to [21], our predictors can at the same time provide both lower and upper bounds on the number of observations before a fault may (resp. must) occur. For instance, upon seeing c in the previous example, an observer can not only predict that a fault will eventually happen but that it will necessarily happen between the first and the second observable event after c. In practice, if a fault prediction is issued, the reaction procedure of the system should be triggered. As such interventions may require a certain amount of time to take effect, having both lower and upper bounds are relevant performance criteria for capture such timing issues.

We then turn to the case of active prediction, where a controller tries to restrict the system's behaviour so that faults can be reliably predicted. For instance, if $\Sigma_1 = \{a, b\}$ and $\Sigma_2 = \{a, c\}$ in Figure 1, then faults are unpredictable, but if a controller has the right to block a, it becomes actively predictable (with the aforementioned bounds). We formalize the idea of active predictability and then propose a class of controller, called active predictor. We then show that active predictability is equivalent to the existence of an active predictor.

Next, we focus on the decision and synthesis problems, i.e. to decide whether the system is actively predictable, and if so, how to build an active predictor. In active *diagnosability* [13], the solution exploited the fact that whether a sequence of observations is ambiguous (i.e. corresponds to both faulty and correct runs) is *independent of the control* that was applied in the past. In prediction, by contrast, the eventuality of a fault occurence in the future *depends on the control* that is going to be applied. Thus solving the active-predictability problems requires new techniques.

We establish that the decision problem is EXPTIME-complete by reducing it to a turn-based game with a Büchi objective of exponential size. A memoryless winning strategy of this game provides the main ingredient to build an active predictor. Furthermore we show that instead of solving this Büchi game (which takes quadratic time), one can equivalently in linear time (1) solve a reachability game, (2) build a safety game that depends on the winning states of the reachability game, and (3) solve it and combine the winning strategies to get a winning strategy for the Büchi game when it exists (see [14] for all details).

Finally we study the relation between the lower prediction bound k and the number of states n of the system. We establish that if $k \geq 2^n$ then a system is k-actively predictable if and only if it is actively safe. This bound is tight since we exhibit a family of systems of size $\mathcal{O}(n)$ such that the system is 2^n-actively predictable but not actively safe.

Organization. In Section 2, we introduce prediction in both the uncontrollable and controllable framework and establish a class of controller called *active predictor*. The existence of such a controller is equivalent to active predictability. The construction of an active predictor (if it exists) is carried out in Section 3, providing simultaneously the solutions to the decision and synthesis problems. Section 4 complements these results by a tight analysis of complexity bounds. We conclude and give some perspectives to this work in Section 5. The missing proofs are developed in [14].

2 The Active Prediction Problem

As usual, for an alphabet Σ, we use Σ^* and Σ^ω, to denote the finite and infinite words over Σ, and $\Sigma^\infty := \Sigma^* \cup \Sigma^\omega$. The length of a word $\sigma \in \Sigma^*$ is denoted $|\sigma|$, and $\preceq$ represents the prefix notation.

Labeled transition systems

When dealing with discrete event systems (DES), systems are often modeled using labeled transition systems (LTS).

▶ **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 a set of transitions.*

We note $q \xrightarrow{a}_{\mathcal{A}} q'$ for $(q, a, q') \in T$; this transition is said to be *enabled* in q. A *run* over the infinite word $\sigma = a_1 a_2 \ldots \in \Sigma^\omega$ is a sequence of states $(q_i)_{i \geq 0}$ with $q_i \xrightarrow{a_{i+1}}_{\mathcal{A}} q_{i+1}$ for all $i \geq 0$, and we write $q_0 \overset{\sigma}{\underset{\mathcal{A}}{\Longrightarrow}}$ if such a run exists. A finite run over $\sigma \in \Sigma^*$ is defined analogously, and we write $q \overset{\sigma}{\underset{\mathcal{A}}{\Longrightarrow}} q'$ if it ends at state q'. A state q is *reachable* if there exists a run $q_0 \overset{\sigma}{\underset{\mathcal{A}}{\Longrightarrow}} q$ for some σ. The index $\mathcal{A}$ in those relations will be omitted if unambiguous.

In order to formalize problems related to prediction, 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 $f \in \Sigma_{uo}$. We say σ is *correct* if $\sigma \in (\Sigma \setminus \{f\})^*$ (we will denote $\Sigma \setminus \{f\}$ with the short form $\Sigma^{\setminus f}$ in the following), and that σ is *faulty* otherwise. For $\Sigma' \subseteq \Sigma$, define its projection $\mathcal{P}_{\Sigma'}(\sigma)$ inductively by: $\mathcal{P}_{\Sigma'}(\varepsilon) = \varepsilon$; $\mathcal{P}_{\Sigma'}(\sigma a) = \mathcal{P}_{\Sigma'}(\sigma) a$ when $a \in \Sigma'$, and $\mathcal{P}_{\Sigma'}(\sigma a) = \mathcal{P}_{\Sigma'}(\sigma)$ otherwise. For the sake of simplicity, write $\mathcal{P}$ for $\mathcal{P}_{\Sigma_o}$, $|\sigma|_o$ for $|\mathcal{P}(\sigma)|$, $|\sigma|_{\Sigma'}$ 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.

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

- *$\mathcal{L}^*(\mathcal{A}) = \{\, \sigma \in \Sigma^* \mid \exists q\ q_0 \overset{\sigma}{\Longrightarrow} q \,\}$ and $\mathcal{L}^\omega(\mathcal{A}) = \{\, \sigma \in \Sigma^\omega \mid q_0 \overset{\sigma}{\Longrightarrow} \,\}$;*
- *$\mathcal{L}^*_c(\mathcal{A}) = \{\, \sigma \in (\Sigma^{\setminus f})^* \mid \exists q\ q_0 \overset{\sigma}{\Longrightarrow} q \,\}$ and $\mathcal{L}^\omega_c(\mathcal{A}) = \{\, \sigma \in (\Sigma^{\setminus f})^\omega \mid q_0 \overset{\sigma}{\Longrightarrow} \,\}$*

$\mathcal{A}$ is safe *if $\mathcal{L}^*(\mathcal{A}) = \mathcal{L}^*_c(\mathcal{A})$ (i.e. no fault ever occurs).*

The union of finite and infinite languages of $\mathcal{A}$ is denoted $\mathcal{L}^\infty(\mathcal{A}) = \mathcal{L}^*(\mathcal{A}) \cup \mathcal{L}^\omega(\mathcal{A})$. The inverse observable projection with respect to $\mathcal{A}$ and $w \in \Sigma_o^*$ is defined as $\mathcal{P}_{\mathcal{A}}^{-1}(w) = \{\sigma \in \mathcal{L}^*(\mathcal{A}) \mid \mathcal{P}(\sigma) = w\}$, which can be simply denoted by $\mathcal{P}^{-1}(w)$ if there is no ambiguity. 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 LTS we write $T(q, a) = q'$ if $q \xrightarrow{a} q'$. As is the case for classical diagnosis problems, we make two **assumptions** on $\mathcal{A}$:

- Liveness: $\forall q \in Q$, $\exists a, q'$, $q \xrightarrow{a} q'$.
- Convergence: $\mathcal{L}^\omega(\mathcal{A}) \cap \Sigma^* \Sigma_{uo}^\omega = \emptyset$.

Liveness implies that from any reachable state of an LTS, there exists at least one transition enabled in that state. Convergence guarantees that there is no infinite sequence of unobservable events. When $\mathcal{A}$ is convergent, then for all $\sigma \in \mathcal{L}^\omega(\mathcal{A})$, one has $\mathcal{P}(\sigma) \in \Sigma_o^\omega$.

▶ **Example 3.** Figure 1 shows a live and convergent LTS with $\Sigma_o = \{a, b, c, d\}$, $\Sigma_{uo} = \{f\}$, $\Sigma_1 \subseteq \Sigma_o$, $\Sigma_2 \subseteq \Sigma_o$ and $\Sigma_1 \cup \Sigma_2 \neq \emptyset$. Transitions labelled by unobservable events are dashed. We also factorize transitions with same source and target states. Depending on Σ_1 and Σ_2, this LTS may have different levels of predictability (see Example 7 for further explanation).

Predictability

Intuitively, a system is predictable with respect to a fault f if in every faulty run, an observer can be certain that f is going to occur before it actually happens. Before formally defining predictability, we first present some useful notations. Given $\sigma \in \mathcal{L}^\infty(\mathcal{A})$ and $n \leq |\sigma|_o$, $pre_n(\sigma)$ denotes the minimal (w.r.t. $\preceq$) prefix of σ such that $|pre_n(\sigma)|_o = n$. As an abbreviation, $pre(\sigma) := pre_{|\sigma|_o}(\sigma)$ removes unobservable events from the end of σ. For example, in the LTS of Figure 1, we have (as f is unobservable) $pre_0(bdf) = \varepsilon$, $pre_1(bdf) = b$ and $pre(bdf) = pre_2(bdf) = bd$. We naturally extend pre to sets of words.

An observed sequence w forbids prediction of a fault when a correct, infinite future behavior is still possible. We introduce different kinds of observed sequences.

▶ **Definition 4.** *(observation properties) Let $\mathcal{A}$ be an LTS, $w \in \Sigma_o^*$, and $m \in \mathbb{N}$. Then w is:*

- surely correct *in $\mathcal{A}$ if $pre(\mathcal{P}_\mathcal{A}^{-1}(w)) \cap \Sigma^* f \Sigma^* = \emptyset$;*
- surely faulty *in $\mathcal{A}$ if $\mathcal{P}_\mathcal{A}^{-1}(w) \cap \mathcal{L}_c^*(\mathcal{A}) = \emptyset$;*
- ambiguous *in $\mathcal{A}$ if it is neither surely correct nor surely faulty in $\mathcal{A}$;*
- m-correct *in $\mathcal{A}$ if ww' is surely correct in $\mathcal{A}$ for all $w' \in \Sigma_o^m$;*
- m-faulty *in $\mathcal{A}$ if ww' is surely faulty in $\mathcal{A}$ for all $w' \in \Sigma_o^m$;*
- ω-faulty *in $\mathcal{A}$ if there exists $m \in \mathbb{N}$ such that w is m-faulty.*

We now define the notion of k-l-predictability, which means that the occurrence of a fault can be predicted with certainty, based on what has been observed so far, at least k observations before it does occur, and such that the fault definitely occurs before the l-th additional observation. In the sequel, $\mathbb{N}^+$ denotes $\mathbb{N} \setminus \{0\}$ and $\mathbb{N}_\omega$ (resp. $\mathbb{N}_\omega^+$) denotes $\mathbb{N}$ (resp. $\mathbb{N}^+$) enlarged with ω which is an absorbing item for addition.

▶ **Definition 5.** *(Predictability) Let $\mathcal{A}$ be an LTS, $w \in \Sigma_o^*$, $k \in \mathbb{N}$, and $l \in \mathbb{N}_\omega^+$.*

- *w is* k-l-faulty *in $\mathcal{A}$ if w is k-correct and $(k+l)$-faulty in $\mathcal{A}$.*
- *$\mathcal{A}$ is* k-l-predictable *if for all $\sigma f \in \mathcal{L}^*(\mathcal{A})$, $\mathcal{P}(\sigma)$ has a k-l-faulty prefix.*

▶ **Remark 6.** If w is k-l-faulty in $\mathcal{A}$, then w is also k'-l'-faulty in $\mathcal{A}$ for all $k' \leq k$ and $k' + l' \geq k + l$.

As an abbreviation, we will call $\mathcal{A}$ *k-predictable* if it is k-ω-predictable, and simply *predictable* if it is 0-predictable. Thus, Remark 6 implies that predictability is weaker than any other notion of k-l-predictability.

▶ **Example 7.** Consider the LTS of Figure 1:

- it is not predictable if $\Sigma_1 \cap \Sigma_2 \neq \emptyset$;
- it is 1-1-predictable and not 2-predictable if $\Sigma_1 \cap \Sigma_2 = \emptyset$, and both of them are not empty;
- it is 2-1-predictable if $\Sigma_1 = \emptyset$ and $\Sigma_2 \neq \emptyset$.

Proposition 8 establishes bounds for predictability in finite LTS:

▶ **Proposition 8.** Let $\mathcal{A}$ be a k-predictable LTS with n states, where n is finite.

(i) $\mathcal{A}$ is k-n-predictable.
(ii) If $\mathcal{A}$ is not safe, then $k < n$.

Active predictability

We suppose that Σ_o is partitioned into subsets $\Sigma_c \subseteq \Sigma_o$ of *controllable* and $\Sigma_{uco} = \Sigma_o \backslash \Sigma_c$ 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 9** (Controlled LTS). *Let $\mathcal{A}$ be an LTS. A* controller *for $\mathcal{A}$ is a mapping $cont : \mathcal{P}(\mathcal{L}^*(\mathcal{A})) \to 2^{\Sigma}$ such that for all w, $\Sigma_{uco} \cup \Sigma_{uo} \subseteq cont(w)$. The controlled LTS $\mathcal{A}_{cont} = \langle Q_{cont}, q_{0cont}, \Sigma, T_{cont} \rangle$ is defined as the smallest LTS satisfying:*

- *$q_{0cont} = \langle \varepsilon, q_0 \rangle \in Q_{cont}$;*
- *if $\langle w, q \rangle \in Q_{cont}$, $a \in cont(w)$, and $q \xrightarrow{a}_{\mathcal{A}} q'$, then $\langle w\mathcal{P}(a), q' \rangle \in Q_{cont}$ and $\langle w, q \rangle \xrightarrow{a}_{\mathcal{A}_{cont}} \langle w\mathcal{P}(a), q' \rangle$.*

The goal of our controllers is to make the system predictable by preserving liveness and to perform prediction at the same time. Before formally defining prediction verdicts in Definition 11, we discuss their intuitive meanings: $\top$ means that the controller is currently unable to predict a fault, while $\langle k, l \rangle$ means that the run is correct so far but a fault can be predicted to happen between the next k and $k + l$ observations. When $l = \omega$, a fault is predicted but without an upper bound. $\langle ?, m \rangle$ means that a fault may or may not have happened yet but one will surely occur within m further observations, and $\perp$ means that a fault has definitely already occurred.

▶ **Example 10.** Consider again the LTS from Figure 1 and assume that $\Sigma_1 = \{a\}$ and $\Sigma_2 = \{b\}$. At the beginning, no fault can be predicted, so a controller would be expected to emit the prediction $\top$. After observing b, the controller could predict that a fault will happen between the first and second next observation to come, i.e. $\langle 1, 1 \rangle$. After seeing d, this would change to $\langle 0, 1 \rangle$, and finally to $\perp$.

▶ **Definition 11** (predictions). *Let $\mathbb{P} := \{\top\} \cup (\mathbb{N} \times \mathbb{N}_{\omega}^{+}) \cup (\{?\} \times \mathbb{N}_{\omega}^{+}) \cup \{\perp\}$ be the set of* possible predictions. *We define the following measures $\kappa, \mu : \mathbb{P} \to \mathbb{N}_{\omega} \cup \{-1, \omega + 1\}$:*

- *$\kappa(\top) = \omega + 1$, $\kappa(\langle k, l \rangle) = k$, and $\kappa(p) = -1$ otherwise;*
- *$\mu(\top) = \omega + 1$, $\mu(\langle k, l \rangle) = k + l$, $\mu(\langle ?, m \rangle) = m$, and $\mu(\perp) = 0$.*

We also define two particular types of subsets of $\mathbb{P}$: For $k \in \mathbb{N}$ and $l \in \mathbb{N}^+$, let $\mathbb{P}_{k,l} := \{\top, \perp\} \cup \{\, \langle k', l' \rangle \mid k' \leq k,\ l' \leq l \,\} \cup \{\, \langle ?, m \rangle \mid m < l \,\}$ and $\mathbb{P}_{k,\omega} := \{\top, \perp, \langle ?, \omega \rangle\} \cup \{\, \langle k', \omega \rangle \mid k' \leq k \,\}$.

The values $\kappa(p)$ and $\mu(p)$ define the "window" (lower and upper bound on future observations) within which a fault is to occur according to prediction p. Here, -1 indicates that the fault may or must have occurred in the past, and in the case of $\top$, $\omega + 1$ is chosen for technical convenience. A predictor using values from $\mathbb{P}_{k,l}$ makes firm commitments on both the lower and upper bounds within which a fault is going to occur, while a predictor with values from $\mathbb{P}_{k,\omega}$ only commits to a lower bound.

▶ **Definition 12** (compatible predictions). *Let $p, p' \in \mathbb{P}$ and $k \in \mathbb{N}, l \in \mathbb{N}_{\omega}^{+}$. We say that $\langle p, p' \rangle$ are k-l-*compatible *if the following conditions are all satisfied:*

- *if $p = \top$, then $\kappa(p') \geq k$ else $\kappa(p') \geq \kappa(p) - 1$;*
- *$\mu(p') \leq \mu(p)$, and if $0 < \mu(p) < \omega$, then $\mu(p') < \mu(p)$;*
- *if $p' \neq \top$, then $\mu(p') \leq k + l$.*

*Moreover, p is called k-l-*initial *if $\langle \top, p \rangle$ are k-l-compatible.*

The conditions in Definition 12 describe the relations that should reasonably hold between a prediction p made for some observation w and the prediction p' made when one has observed one additional event in a k-l-predictable controlled LTS. Intuitively these are:

1. When a fault is first predicted, it should be at least k observations in advance, and the gap between this lower bound and the upper bound should be at most l. This is why $p = \top$ should imply $\kappa(p') \geq k$. In particular, one cannot switch from $\top$ to $\langle k', l' \rangle$ for any $k' < k$, nor directly to $\langle ?, m \rangle$ or $\bot$. Moreover, the third condition ensures that when switching from $\top$ to $\langle k', l' \rangle$, we have $k' + l' \leq k + l$, which with $k' \geq k$ implies $l' \leq l$.
2. Having predicted a fault within a certain "window", the subsequent predictions can only become more precise. Thus, one can maintain or shrink that window, but not enlarge, shift, or forget about it. Figure 2 illustrates this idea. E.g., when a predictor announces a fault between the 3rd and 7th following observation, expressed by $p = \langle 3, 4 \rangle$, then one step later it must give $p' = \langle 2, 4 \rangle$ or a more precise verdict such as $\langle 3, 2 \rangle$. As another example, if the controller has arrived at a verdict of $\langle ?, 6 \rangle$, meaning "a fault has occurred, or will occur within six further observations", then the information gained from an additional observation may lead it to conclude that the fault has now definitely occurred ($\bot$), will occur later (e.g., $\langle 1, 3 \rangle$), or to maintain the prediction (e.g., $\langle ?, 5 \rangle$). Note that $\langle ?, 6 \rangle$ could only be reached by passing through $\langle 0, m \rangle$, for some $m > 6$, earlier in the observation. These relations are ensured by allowing κ to decrease by at most one and requiring μ to strictly decrease (if an upper bound was given).

A k-l-initial prediction is one that is admissible for the empty observation.

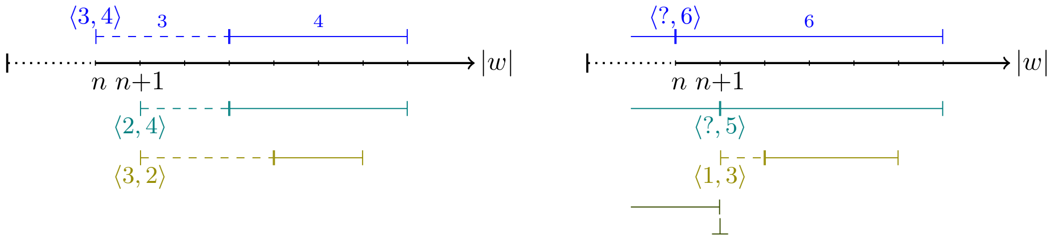

Figure 2 Examples of compatible predictions $\langle p, p' \rangle$ after n resp. $n + 1$ observations, where p is illustrated above the timeline, and p' is one of the predictions below. Solid intervals indicate periods in which a fault is predicted.

▶ **Definition 13** (active predictor). *Let $\mathcal{A}$ be an LTS, $\mathbb{P}' \subseteq \mathbb{P}$, and $h = \langle cont, pred \rangle$, where cont is a controller and pred is a mapping from $\mathcal{P}(\mathcal{L}^*(\mathcal{A}_{cont}))$ to $\mathbb{P}'$. We call h a* k-l-active predictor *over $\mathbb{P}'$, for $k \in \mathbb{N}$ and $l \in \mathbb{N}^+_\omega$, if and only if:*

(i) *$\mathcal{A}_{cont}$ is live;*

(ii) *$pred(\varepsilon)$ is k-l-initial;*

(iii) *for $w \in \mathcal{P}(\mathcal{L}^*(\mathcal{A}_{cont}))$, the prediction satisfies the following:*

- *if $pred(w) = \top$, then w is $(k+1)$-correct in $\mathcal{A}_{cont}$;*
- *if $pred(w) = \langle k', l' \rangle$, then w is k'-l'-faulty in $\mathcal{A}_{cont}$;*
- *if $pred(w) = \langle ?, m \rangle$, then w is ambiguous and m-faulty in $\mathcal{A}_{cont}$;*
- *if $pred(w) = \bot$, then w is surely faulty in $\mathcal{A}_{cont}$;*

(iv) *for $a \in \Sigma_o$, $w, wa \in \mathcal{P}(\mathcal{L}^*(\mathcal{A}_{cont}))$, $\langle pred(w), pred(wa) \rangle$ are k-l-compatible.*

Intuitively, condition (i) requires that the control cannot introduce deadlocks, and conditions (ii),(iii) ensure that the predictions have the intended semantics. Condition (iv) ensures compatibility between two subsequent predictions along an observation. If there exists a k-l-active predictor for $\mathcal{A}$, we call $\mathcal{A}$ *k-l-active-predictable*, or just *actively predictable*. Moreover, $\mathcal{A}$ is called *actively safe* if there exists an active predictor for $\mathcal{A}$ over $\{\top\}$, which entails that $\mathcal{A}_{cont}$ is safe.

▶ **Example 14.** In the LTS $\mathcal{A}$ of Figure 1, assume that $\Sigma_1 = \{a, c\}$, $\Sigma_2 = \{a, b\}$, $\Sigma_c = \{a, b, c\}$. Let $h = \langle cont, pred\rangle$ be defined by:

- $cont(\varepsilon) = \{b, c, d, f\}$, and $cont(w) = \Sigma$ otherwise;
- $pred(\varepsilon) = pred(w) = \top$, where $w \in c\Sigma_o^* \cap \mathcal{P}(\mathcal{L}^*(\mathcal{A}))$, $pred(b) = \langle 1, 1\rangle$, $pred(bd) = \langle 0, 1\rangle$, and $pred(bda^+) = \bot$.

In this example, h is a 1-1-active predictor.

Proposition 15 and Proposition 16 will exhibit a tight correspondence between the existence of a k-l-predictor for $\mathcal{A}$ and the existence of a controller that makes $\mathcal{A}$ k-l-predictable. Additionally, Proposition 16 shows that the set of predictions used in a predictor can be limited to a finite set, either committing the prediction to a lower and upper bound for the occurrence of a fault, or just a lower bound.

▶ **Proposition 15.** If $h = \langle cont, pred\rangle$ is a k-l-active predictor for an LTS $\mathcal{A}$, then $\mathcal{A}_{cont}$ is k-l-predictable.

▶ **Proposition 16.** Let $\mathcal{A}$ be an LTS. If there exists a controller $cont$ such that $\mathcal{A}_{cont}$ is live and k-l-predictable, then there exist k-l-active predictors $h = \langle cont, pred\rangle$ for $\mathcal{A}$ over both $\mathbb{P}_{k,l}$ and $\mathbb{P}_{k,\omega}$.

Finally, we introduce the notion of *pilot* as an automata-based representation of k-l-active predictors. In Section 3 we will show how to find a finite-state pilot when $\mathcal{A}$ is actively predictable and finite-state.

▶ **Definition 17** (pilot). *Let $\mathcal{A}$ be an LTS, then $\mathcal{C} = \langle \mathcal{B}_\mathcal{C}, cont_\mathcal{C}, pred_\mathcal{C}\rangle$ is called* pilot *for $\mathcal{A}$ over $\mathbb{P}' \subseteq \mathbb{P}$ if $\mathcal{B}_\mathcal{C} = \langle Q^c, q_0^c, \Sigma_o, T^c\rangle$ is a deterministic LTS with labellings $\langle cont_\mathcal{C}, pred_\mathcal{C}\rangle : Q^c \to 2^\Sigma \times \mathbb{P}'$. Let $h_\mathcal{C} = \langle cont, pred\rangle$ associated with $\mathcal{C}$ be defined by $cont(w) = cont_\mathcal{C}(q)$ and $pred(w) = pred_\mathcal{C}(q)$ for all $w \in \mathcal{P}(\mathcal{L}^*(\mathcal{A}))$, where q is the unique state such that $q_0^c \overset{w}{\Longrightarrow} q$. Then $\mathcal{C}$ is a k-l-active predictor for $\mathcal{A}$ if $h_\mathcal{C}$ is one.*

3 Controller construction

We solve the decision and synthesis problems simultaneously. We try to construct a pilot-based k-l-active predictor over some $\mathbb{P}' \subseteq \mathbb{P}$ for an LTS $\mathcal{A}$. The construction succeeds if and only if $\mathcal{A}$ is k-l-actively predictable. According to Definition 13, the main challenges in building an active predictor are to ensure that (i) the controlled system remains live, (ii) the fault can be predicted at least k observations before its occurrences, and (iii) the prediction information is provided.

Our solution consists in building a turn-based game (see [12] for turn-based games) by taking into account the control that has already been performed.

▶ **Definition 18** (turn-based game). *A game $\mathcal{G}$ with two players called* Control *and* Environment *is a tuple $\langle V_C, V_E, E, v_0, WIN\rangle$, where:*

- *V_C, V_E are the vertices owned by Control and Environment, respectively, and $V_\mathcal{G} = V_C \uplus V_E$ denoting all vertices, with $v_0 \in V_C$ being an* initial vertex*;*
- *$E \subseteq V_\mathcal{G} \times V_\mathcal{G}$ is a set of directed edges such that for all $v \in V_\mathcal{G}$, there exists $(v, v') \in E$;*
- *$WIN \subseteq V_\mathcal{G}^\omega$ is a set of winning sequences.*

Given a sequence $\rho = v_0v_1...v_n$, we denote $\rho[i] = v_i$. A *play* is a sequence of $V_\mathcal{G}^\omega$ 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 define $last(\rho^k) := \rho[k]$. We write $Play^*(\mathcal{G})$ for the set of partial plays of $\mathcal{G}$. A play ρ is called *winning* (for Control) if $\rho \in WIN$.

A *Büchi game* $\langle V_C, V_E, E, v_0, V_F\rangle$ defines a game $\langle V_C, V_E, E, v_0, \mathit{WIN}\rangle$ such that $\mathit{WIN} = \{\, \rho \in V_\mathcal{G}^\omega \mid \rho[i] \in V_F$ for infinitely many $i\,\}$. A *reachability game* $\langle V_C, V_E, E, v_0, V_F\rangle$ defines a game $\langle V_C, V_E, E, v_0, \mathit{WIN}\rangle$ such that $\mathit{WIN} = V_\mathcal{G}^* V_F V_\mathcal{G}^\omega$. A *safety game* $\langle V_C, V_E, E, v_0, V_F\rangle$ defines a game $\langle V_C, V_E, E, v_0, \mathit{WIN}\rangle$ such that $\mathit{WIN} = V_F^\omega$.

▶ **Definition 19** (strategy). *Let $\mathcal{G} = \langle V_C, V_E, E, v_0, \mathit{WIN}\rangle$ be a game. A* strategy *(for Control) is a function $\theta\colon \mathit{Play}^*(\mathcal{G}) \to V_\mathcal{G}$ such that $(\mathit{last}(\xi), \theta(\xi)) \in E$ for all $\xi \in \mathit{Play}^*(\mathcal{G})$. 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 (also called memoryless) strategy *is a function $\theta'\colon V_C \to V_\mathcal{G}$ such that $(v, \theta'(v)) \in E$ for all $v \in V_C$; we call θ'* winning *if the strategy θ with $\theta(\xi) = \theta'(\mathit{last}(\xi))$ is winning.*

To verify k-l-active predictability of a given system, the controller that we propose needs to memorize two subsets of states with the corresponding prediction information $\langle Q_c, Q_f, p\rangle$. The subset Q_c (resp. Q_f) represents the possible states reached by a correct (resp. faulty) run after the last observable action, and $Q_c \cup Q_f \neq \emptyset$. The prediction information $p \in \mathbb{P}'$ is (non-deterministically) decided based on the current observations. We denote $\mathit{Reach}(\langle Q_c, Q_f, p\rangle) := Q_c \cup Q_f$ and $\widetilde{Q} := 2^Q \setminus \{\emptyset\}$. The set of possible tuples memorized by the controller is defined as $S_{\mathbb{P}'} = S_{\mathbb{P}'}^c \cup S_{\mathbb{P}'}^a \cup S_{\mathbb{P}'}^f$, where:

- $S_{\mathbb{P}'}^c = \widetilde{Q} \times \{\emptyset\} \times \{\, p \in \mathbb{P}' \mid \kappa(p) \geq 0\,\}$
- $S_{\mathbb{P}'}^a = \widetilde{Q} \times \widetilde{Q} \times \big(\mathbb{P}' \cap (\{?\} \times \mathbb{N}_\omega^+)\big)$
- $S_{\mathbb{P}'}^f = \{\emptyset\} \times \widetilde{Q} \times \{\bot\}$

In the following, we will simply write S for $S_{\mathbb{P}'}$ when $\mathbb{P}'$ is clear from context.

The controller needs to update the state subsets after an observable action, for which we first define some sets of possible next states from a given state q after $a \in \Sigma_o$.

- $\mathit{NO}_\mathcal{A}(q, a) = \{\, q' \mid q \overset{\sigma}{\underset{\mathcal{A}}{\Longrightarrow}} q',\ \sigma \in \Sigma_{uo}^* a\,\}$
- $\mathit{NOC}_\mathcal{A}(q, a) = \{\, q' \mid q \overset{\sigma}{\underset{\mathcal{A}}{\Longrightarrow}} q',\ \sigma \in (\Sigma_{uo} \setminus \{f\})^* a\,\}$
- $\mathit{NOF}_\mathcal{A}(q, a) = \{\, q' \mid q \overset{\sigma}{\underset{\mathcal{A}}{\Longrightarrow}} q',\ \sigma \in \Sigma_{uo}^* f \Sigma_{uo}^* a\,\}$

One can omit the subscript $\mathcal{A}$ when there is no ambiguity. The extension to a set of states is defined in a natural way, e.g. $\mathit{NO}(Q', a) = \bigcup_{q \in Q'} \mathit{NO}(q, a)$. We now define how the controller updates its tuple once an observable action occurs. In the following, $\oslash$ represents a state in which the controller has lost, and we denote $S^\oslash := S \cup \{\oslash\}$.

▶ **Definition 20** (knowledge update). *Let $\mathcal{A}$ be an LTS, $\mathbb{P}' \subseteq \mathbb{P}$, and $k \geq 0$. Then the knowledge transition relation $\Delta_\mathcal{A}^k \subseteq S \times \Sigma_o \times S^\oslash$ is defined as follows. Let $s = \langle Q_c, Q_f, p\rangle \in S$ and $a \in \Sigma_o$. Then $\langle s, a, s'\rangle \in \Delta_\mathcal{A}^k$ if and only if:*

1. *either $s' = \langle \mathit{NOC}(Q_c, a), \mathit{NOF}(Q_c, a) \cup \mathit{NO}(Q_f, a), p'\rangle \in S$ and $\langle p, p'\rangle$ are k-l-compatible;*
2. *or $s' = \oslash$ when there is no $s'' \in S$ such that $\langle s, a, s''\rangle \in \Delta_\mathcal{A}^k$.*

Notice that, given s and a, the choice of s' is largely deterministic except for p', which must be k-l-compatible with p. When s' has no prediction consistent with the updated correct resp. faulty state subsets, cf Definition 13(iii), then the only possible update is to $\oslash$.

▶ **Example 21.** Consider the LTS in Figure 1 and assume that $\Sigma_1 = \{a, c\}$, $\Sigma_2 = \{a, b\}$ and $\Sigma_c = \{a, b, c\}$.

1. Let $s = \langle \{q_0\}, \emptyset, \top\rangle$. If the observable action a is chosen, then we have $\langle s, a, s'\rangle \in \Delta_\mathcal{A}^k$, where $s' = \langle \{q_1, q_4\}, \emptyset, \top\rangle$. Notice that $\langle \top, \top\rangle$ are k-l-compatible.
2. Let $s = \langle \{q_2, q_5\}, \emptyset, \top\rangle$ after observing a and d. If a is chosen from here, we can only have $\langle s, a, \oslash\rangle \in \Delta_\mathcal{A}^k$. The reason is that after a, the system can end up in either q_3 (with a fault) or in q_5 (without fault), the next prediction should thus be an ambiguous one,

i.e., $\langle ?, m\rangle$. However, $\langle \top, \langle ?, m\rangle\rangle$ are not k-l-compatible. It follows that there does not exist $s'' \in S$ such that $\langle s, a, s''\rangle \in \Delta^k_{\mathcal{A}}$. Hence we have $\langle s, a, \oslash\rangle \in \Delta^k_{\mathcal{A}}$ by Definition 20.

The objective of Control is to obtain a winning play by suitably restricting the possible actions, and any winning strategy corresponds to a controller with which the controlled system is predictable. The game begins with Control to choose a prediction for ε. Then the game proceeds in rounds: 1) Control restricts the set of possible actions to some Σ'; 2) Environment chooses $a \in \Sigma'$ to determine the next state. 3) Control updates its knowledge.

The choices of Control are subject to some restrictions. Indeed, each state $s = \langle Q_c, Q_f, p\rangle$ represents Control's knowledge about the current potential states of $\mathcal{A}$ as well as the corresponding prediction information. To ensure that the controlled system remains live, the set of possible actions Σ' must not cause deadlocks in any state reachable by unobservable actions from $Q_c \cup Q_f$. Also, Control cannot prevent the uncontrollable actions. So we define the admissible sets and the game as follows, where we use $\Sigma_{PO}(q) = \{a \in \Sigma_o \mid q \overset{\sigma}{\Longrightarrow} q'', \sigma \in \Sigma^*_{uo}a\ \}$ to denote the possible next observable actions from the state q, which can be extended to a set of states in a natural way.

▶ **Definition 22** (admissible action set). *Let $\mathcal{A} = \langle Q, q_0, \Sigma, T\rangle$ be an LTS and $Q' \subseteq Q$ be a subset of states. We call $\Sigma' \subseteq \Sigma_o$ an admissible set for Q' if it fulfills the following conditions:*

- *$\Sigma_{uco} \subseteq \Sigma'$ as any action in Σ_{uco} is observable but not controllable.*
- *for all $q' \in Q'$, $q \in Q$, and $\sigma \in \Sigma^*_{uo}$, $q' \overset{\sigma}{\Longrightarrow} q$ implies $\Sigma_{PO}(q) \cap \Sigma' \neq \emptyset$.*

The set of admissible sets for Q' are denoted as $adm(Q')$, which is not empty when $Q' \neq \emptyset$ as $\mathcal{A}$ is a live and convergent LTS.

▶ **Example 23.** Consider the same LTS as in Example 21. Let $Q' = \{q_0\}$. Then $adm(Q') = \{\Sigma' \mid \Sigma' \subseteq \Sigma_o, \{d\} \subsetneq \Sigma'\}$. In other words, $adm(Q')$ contains all subsets of $\Sigma_o = \{a, b, c, d\}$ that include d, except the singleton $\{d\}$, which is not an admissible set as it blocks the system. More precisely, the set of possible next observable actions from q_0 is $\Sigma_{PO}(q_0) = \{a, b, c\}$, whose intersection with $\{d\}$ is empty. Thus $\{d\}$ cannot be an admissible set for Q'.

The vertices of our controller-synthesis game consist of an initial vertex ι, the states of $S^{\oslash}$, a set $V_1 := S \times 2^{\Sigma_o}$ where Control has chosen a set of permitted actions, and a set $V_2 := S \times \Sigma_o$ where Environment has chosen an observable action. The winning condition assures that once a fault has been predicted, it will eventually happen.

▶ **Definition 24** (controller-synthesis game). *Let $\mathcal{A}$ be an LTS and $\mathbb{P}' \subseteq \mathbb{P}$. We denote $\mathcal{G}^{k,l}_{\mathcal{A},\mathbb{P}'}$ the Büchi game $\langle V_C, V_E, E, \iota, V_F\rangle$, where $V_C = \{\iota\} \cup S^{\oslash} \cup V_2$, $V_E = V_1$, $V_F = \big(\widetilde{Q} \times \{\emptyset\} \times \{\top\}\big) \cup \big(\{\emptyset\} \times \widetilde{Q} \times \{\bot\}\big) \subseteq S$, and $E = E_\iota \cup E_1 \cup E_2 \cup E_3 \cup \{\langle \oslash, \oslash\rangle\}$, where*

- *$E_\iota = \big\{ \langle \iota, \langle \{q_0\}, \emptyset, p\rangle\rangle \mid p \text{ is } k\text{-}l\text{-initial} \big\} \subseteq \{\iota\} \times S$;*
- *$E_1 = \big\{ \langle s, \langle s, \Sigma'\rangle\rangle \mid s \in S,\ \Sigma' \in adm(Reach(s)) \big\} \subseteq S \times V_1$;*
- *$E_2 = \big\{ \langle \langle s, \Sigma'\rangle, \langle s, a\rangle\rangle \mid s \in S,\ a \in \Sigma_{PO}(Reach(s)) \cap \Sigma' \big\} \subseteq V_1 \times V_2$;*
- *$E_3 = \big\{ \langle \langle s, a\rangle, s'\rangle \mid \langle s, a, s'\rangle \in \Delta^k_{\mathcal{A}} \big\} \subseteq V_2 \times S^{\oslash}$.*

Note that the set V_2 records the sequence of observable actions that occur during a play.

▶ **Example 25.** Figure 3 depicts a part of a game for some k, l and the LTS of Figure 1, for which we assume again $\Sigma_1 = \{a, c\}$, $\Sigma_2 = \{a, b\}$ and $\Sigma_c = \{a, b, c\}$. From ι, Controller can choose any k-l-initial prediction; we consider the case where $\top$ is chosen, so $s_0 = \langle \{q_0\}, \emptyset, \top\rangle$. Then from Example 23, we have $adm(Reach(s_0)) = adm(\{q_0\}) = \{\Sigma' \mid \Sigma' \subseteq \Sigma_o, \{d\} \subsetneq \Sigma'\}$. Environment cannot choose the action d even when d is in the admissible set since $d \notin \Sigma_{PO}(Reach(s_0))$. After Environment chooses an available action (say a, leading to $\langle s_0, a\rangle$), Control updates its knowledge and chooses a new prediction, say $\top$, leading to s_1, with q_1, q_4

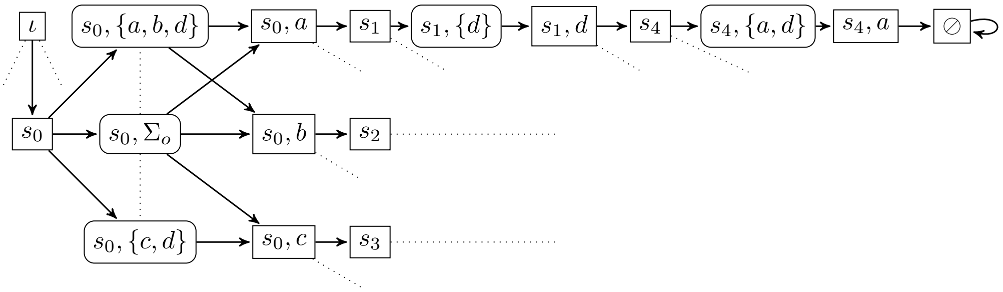

Figure 3 Part of the game for the LTS in Figure 1 (Example 25):
$s_0 = \langle\{q_0\}, \emptyset, \top\rangle$, $s_1 = \langle\{q_1, q_4\}, \emptyset, \top\rangle$, $s_2 = \langle\{q_1\}, \emptyset, p_2\rangle$, $s_3 = \langle\{q_4\}, \emptyset, p_3\rangle$, and $s_4 = \langle\{q_2, q_5\}, \emptyset, \top\rangle$.

as the possible new states. From here, d is the only choice for Environment. Suppose that Control then again chooses $\top$ as its new prediction in s_4, thus $s_4 = \langle\{q_2, q_5\}, \emptyset, \top\rangle$. If a is now chosen, from the second case of Example 21, we know that the game enters $\oslash$. To avoid losing, Control needs to switch to a different prediction early enough.

Now we establish the strong connection between winning strategies and active predictors.

▶ **Proposition 26.** Given $h = \langle cont, pred\rangle$ a k-l-active predictor over $\mathbb{P}'$ for an LTS $\mathcal{A}$, there exists a corresponding winning strategy θ_h in the game $\mathcal{G}^{k,l}_{\mathcal{A},\mathbb{P}'}$.

The existence of a winning strategy implies the existence of a positional one due to well-known results of game theory (see e.g. [12] for all results here related to turn-based games). For the reverse direction, we next define a pilot from a positional winning strategy in $\mathcal{G}^{k,l}_{\mathcal{A},\mathbb{P}'}$ before proving that this pilot is a k-l-active predictor.

▶ **Definition 27.** *Let θ be a positional winning strategy in $\mathcal{G}^{k,l}_{\mathcal{A},\mathbb{P}'}$. We define a pilot $\mathcal{C}_\theta := \langle\mathcal{B}_\theta, cont_\theta, pred_\theta\rangle$ over $\mathbb{P}'$ as follows:*

- *$\mathcal{B}_\theta = \langle Q^\theta, q_0^\theta, \Sigma_o, T^\theta\rangle$, where*
  1. *$Q^\theta = \{q \in S \mid q = last(\xi_\theta)$ and $\xi_\theta \in Play^*(\mathcal{G}^{k,l}_{\mathcal{A},\mathbb{P}'})$ adhering to $\theta\}$*
  2. *$q_0^\theta = \theta(\iota)$*
  3. *$T^\theta(s,a) = \theta(\langle s, a\rangle)$*
- *$cont_\theta(s) = \Sigma' \cup \Sigma_{uo}$ for any $s \in Q^\theta$, where $\theta(s) = \langle s, \Sigma'\rangle$;*
- *$pred_\theta(s) = p$, for any $s = \langle Q_c, Q_f, p\rangle \in Q^\theta$*

▶ **Proposition 28.** Let θ be a positional winning strategy in $\mathcal{G}^{k,l}_{\mathcal{A},\mathbb{P}'}$. Then $\mathcal{C}_\theta$ is a k-l-active predictor over $\mathbb{P}'$ for $\mathcal{A}$.

Combining the results of Propositions 26 and 28, we obtain that the active-predictability problem for an LTS $\mathcal{A}$ with n states reduces to solving a Büchi game with $2^{\mathcal{O}(n)}$ vertices. Since Büchi games can be solved in polynomial time, we obtain the following result:

▶ **Theorem 29.** *The active-predictability problem for finite-state LTS belongs to EXPTIME.*

We conclude the section with a supplementary result showing that due to the special structure of $\mathcal{G}^{k,l}_{\mathcal{A},\mathbb{P}'}$ it can actually be solved in linear time (w.r.t. the size of the game), and not in quadratic time as performed for general Büchi games.

▶ **Proposition 30.** If $\mathcal{A}$ is a finite-state LTS and $\mathbb{P}' \subseteq \mathbb{P}$, then $\mathcal{G}^{k,l}_{\mathcal{A},\mathbb{P}'}$ can be solved in $\mathcal{O}(|E|)$.

4 Bound analysis

We first prove that it is EXPTIME-hard to decide whether a given LTS $\mathcal{A}$ is actively k-l-predictable, independently of k and ℓ. The proof (developed in [14]) is similar to the proof in [13] that active *diagnosability* is EXPTIME-hard and relies on a reduction from safety games with imperfect information [3].

▶ **Theorem 31.** *The active-predictability decision problem is EXPTIME-hard.*

Together with Theorem 29, we obtain the following corollary.

▶ **Corollary 32.** *The active-predictability decision problem is EXPTIME-complete.*

We study the relation between active predictability and active safety. Theorem 33 relates the maximal advance warning for fault predictions to the number of states in $\mathcal{A}$.

▶ **Theorem 33.** *Let $\mathcal{A}$ be an LTS with n states. If $\mathcal{A}$ is 2^n-active-predictable, then it is actively safe.*

Proof. If $\mathcal{A}$ is 2^n-ω-active-predictable then by definition there exists a 2^n-ω-active predictor $h = \langle cont, pred \rangle$ over $\mathbb{P}' := \mathbb{P}_{k,\omega}$ for $\mathcal{A}$, and by Proposition 26 there exists a winning strategy θ in $\mathcal{G}^{k,\omega}_{\mathcal{A},\mathbb{P}'}$. In turn, this winning strategy provides a pilot $\mathcal{C}_\theta = \langle \mathcal{B}, cont', pred' \rangle$ according to Proposition 28; let $\mathcal{B} = \langle Q, q_0, \Sigma_o, T \rangle$. We shall construct a new pilot $\mathcal{C}$ for $\mathcal{A}$ over $\{\top\}$, proving that $\mathcal{A}$ is actively safe.

Remember that Q is the set of Controller-owned vertices in $\mathcal{G}^{k,\omega}_{\mathcal{A},\mathbb{P}'}$ that can be reached by plays adhering to θ and that these vertices are a subset of $S_{\mathbb{P}'}$. For $q, q' \in Q$, let us write $q \prec q'$ if q' is reachable from q in $\mathcal{B}$. Since θ is positional and winning, $\prec$ must be an acylic relation between those states of Q that are not members of V_F, i.e. their associated prediction is neither $\top$ nor $\perp$ (cf Definition 24). We now call $q \in Q$ a *cutoff* if q is of the form $\langle Q_c, Q_f, p \rangle$ and there exists a state $q' = \langle Q_c, Q_f, p' \rangle$ with $p' \neq p$ and $q' \prec q$. Let $co(q)$, the *corresponding state* of q, denote the state that is $\prec$-minimal among all the choices for q'; due to the structure of the states outside V_F, $co(q)$ is unique and not a cutoff itself. Moreover, a state of Q is called *useless* if it is either a cutoff or all its (immediate) predecessors in $\mathcal{B}$ are useless, and *useful* otherwise.

Remember that $S_{\mathbb{P}'}$ is a union of $S^c_{\mathbb{P}'}$, $S^a_{\mathbb{P}'}$, and $S^f_{\mathbb{P}'}$, where $S^c_{\mathbb{P}'}$ contains the states of the form $\langle Q_c, \emptyset, p \rangle$, with $\kappa(p) \geq 0$. Thus, states in $S^c_{\mathbb{P}'}$ are only reached through correct runs in $\mathcal{A}_{cont'}$. Let $S' := \{\, \langle Q_c, \emptyset, p \rangle \mid \kappa(p) = 0 \,\}$. It follows from the construction of $\mathcal{G}^{k,\omega}_{\mathcal{A},\mathbb{P}'}$ (cf Definition 20 and Definition 24) that any path from q_0 to a state from S' is of length at least 2^n, so by pigeonhole principle, any path leading to S' contains a cutoff. Since $S^a_{\mathbb{P}'} \cup S^f_{\mathbb{P}'}$ can only be reached by going through S', those states are useless.

We can now construct the desired pilot $\mathcal{C}$ by "folding" cutoffs back onto their corresponding states. We remark in this context that $Reach(q) = Reach(co(q))$, and therefore the admissible control choices for both states are the same; proving that the resulting controlled system is live depends only on this property. Since the controlled system never admits a fault, the prediction can be $\top$ in all cases. More formally, $\mathcal{C} := \langle \langle Q', q_0, \Sigma_o, T' \rangle, cont', pred'' \rangle$, where Q' is the useful subset of Q, and for all $q \in Q'$, $a \in \Sigma_o$:

- $T'(q, a) = T(q, a)$ if $T(q, a) \in Q'$ and $T'(q, a) = co(T(q, a))$ otherwise;
- $pred''(q) = \top$. ◀

Theorem 33 implies that if a system is not actively safe, then there is an exponential upper bound on the advance warning that an active predictor can issue. This bound is asymptotically precise, as the following family of examples shows.

▶ **Theorem 34.** *There exists a family of systems $(\mathcal{A}_n)_{n\geq 1}$ with $\mathcal{O}(n)$ states such that $\mathcal{A}_n$ is not actively safe but 2^n-active-predictable.*

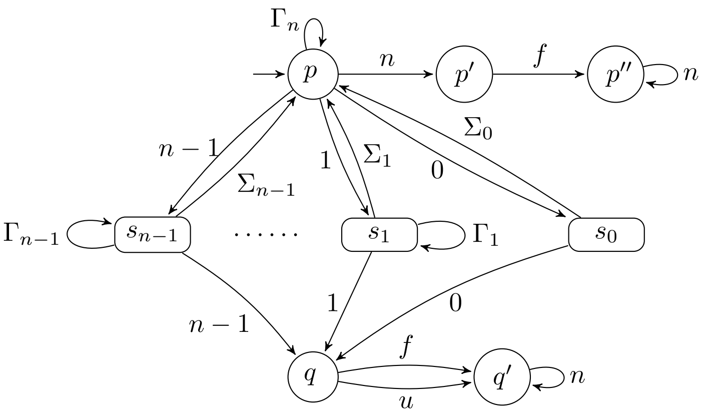

Figure 4 A 2^n-active predictable LTS with $\mathcal{O}(n)$ states, where $\Sigma_o = \Sigma_c = \{0, ..., n\}$, $\Sigma_i = \{i+1, ..., n\}$, and $\Gamma_i = \{0, ..., i-1\}$.

Proof. Figure 4 shows a family of LTS with $\mathcal{O}(n)$ states but an alphabet of size $\mathcal{O}(n)$ and $\mathcal{O}(n^2)$ transitions. We first provide a proof for this family as it is easier to understand. After this, we provide a more complex example with a constant-size alphabet and $\mathcal{O}(n)$ states and transitions.

Variable-size alphabet

Consider the LTS shown in Figure 4. The observable actions are $\{0, \ldots, n\}$, all of which are controllable. There are only two unobservable actions, u and the fault f. We abbreviate by $\Sigma_i := \{i+1, \ldots, n\}$ the actions larger than i for $0 \leq i < n$, and by $\Gamma_i := \{0, \ldots, i-1\}$ the actions smaller than i for $0 < i \leq n$.

The initial state is p. Evidently $\mathcal{A}_n$ is actively safe if a controller can avoid both p' and q; as we shall see, this is impossible. However, the system is actively predictable if the controller can at least avoid q. We shall see that this is indeed possible while entering p' only after 2^n steps, by simulating a binary counter.

We can assume (w.l.o.g.) that the controller permits a single action from Σ_o in each step and hence the controlled system will admit a single infinite observation sequence ρ. Having allowed a prefix σ of ρ, let $R(\sigma)$ be the set of states that this sequence can lead to. If the controller wants to keep the system from making a fault, it must ensure that $R(\sigma)$ remains within the set $R := \{p, s_0, \ldots, s_{n-1}\}$. When $R(\sigma) \subseteq R$, let us associate a measure defined as $I(\sigma) := \sum_{s_i \in R(\sigma)} 2^i$. We observe the following:

- $R(\varepsilon) = \{p\}$, hence $I(\varepsilon) = 0$.
- If $s_i \in R(\sigma)$, then the controller must not allow action i in the next step, otherwise the system may go to q, rendering it unpredictable.
- As long as $I(\sigma) < 2^n - 1$, the controller must permit an action i such that $I(\sigma i) > I(\sigma)$. To see this, let $s_i \notin R(\sigma)$, then $R(\sigma i) = (R(\sigma) \cup \{s_i\}) \setminus \{s_0, \ldots, s_{i-1}\}$. We shall assume that i is chosen minimally, so $I(\sigma i) = I(\sigma) + 1$.
- Therefore, after $2^n - 1$ steps, the controlled system will have performed a sequence $\hat{\sigma}$ with $I(\hat{\sigma}) = 2^n - 1$. The only possible course of action for the controller is to permit n from now on, i.e. $\rho = \hat{\sigma} n^\omega$. We then have $R(\hat{\sigma} n) = \{p, p'\}$, $R(\hat{\sigma} nn) = \{p', p''\}$, and $R(\hat{\sigma} nnn) = \{p''\}$.

Going backwards, we can now associate predictions with each prefix of ρ: $\mathit{pred}(\hat{\sigma}n^k) = \bot$ for $k \geq 3$, $\mathit{pred}(\hat{\sigma}nn) = \langle ?, 1\rangle$, $\mathit{pred}(\hat{\sigma}n) = \langle 0, 2\rangle$, and $\mathit{pred}(\sigma) = \langle 2^n - |\sigma|, 2\rangle$ for every prefix σ of $\hat{\sigma}$. Thus, $\mathcal{A}_n$ is 2^n-2-active predictable. Notice that the system could be made 2^n-1-active predictable if states $s_0, \dots, s_{n-1}$ transitioned with n to p' instead, which we avoided simply to keep the drawing of the automaton planar.

Constant-size alphabet

To see that the proof with a variable-size alphabet can be adapted to an alphabet of constant size, consider the LTS $\mathcal{A}'_n$ in Figure 5. $\mathcal{A}'_n$ has $\mathcal{O}(n)$ states and three observable and controllable actions $0, 1, a$ and two unobservable actions u and f. Initially, the LTS performs an a going to either p or r. The LTS then simulates $\mathcal{A}_n$ of Figure 4, using a unary encoding, in the following sense: Let $\mathit{code}(i) = 1^i 0^{n-i} a$, for $i = 0, \dots, n$. The reader can verify, case-by-case, that for any two states $u, v \in \{p, p', s_0, \dots, s_{n-1}, q\}$ and $i \in \{0, \dots, n\}$, we have $u \xrightarrow{i} v$ in $\mathcal{A}_n$ iff $u \xrightarrow{\mathit{code}(i)} v$ in $\mathcal{A}'_n$. Moreover, the controller must account for the possibility that the system has gone to state r. Then, to keep the controlled system live, the only possible sequences that the controller can enforce are $\mathit{code}(i)$ for $i = 0, \dots, n$, and we have $r \xrightarrow{\mathit{code}(i)} r$ for $i < n$. After the initial a, the controller must therefore admit $\mathit{code}(\hat{\sigma}n)$, for $\hat{\sigma}$ as in $\mathcal{A}_n$. On this basis, a closer look shows that $\mathcal{A}'_n$ is k-l-active predictable for $k = 1 + (n + 1) \cdot 2^n$ and $l = n + 2$. ◀

Note that Theorem 34 does not contradict Proposition 8, which establishes linear prediction bounds w.r.t. the number of states of $\mathcal{A}$. However, Proposition 8 talks about passive predictability, whereas Theorem 34 is about active predictability.

5 Conclusion and perspectives

We have extended the prediction paradigm by introducing parameters related to the number of observations before fault may or must occur. Within this framework, we have established that active predictability is EXPTIME-complete through a procedure for synthetising active predictors that builds a Büchi game. Solving this game is proved linear in the number of edges in the game. We have shown that if the observation threshold for *eventual* prediction is chosen large enough (namely $\geq 2^n$ with n the number of states in the system), then active predictability is equivalent to active safety. Furthermore we have exhibited a family of systems proving that this bound is tight.

Out of several possible extensions for the present results, three stand out as natural continuations. First, we want to introduce a measure that quantifies the faultiness of the system, and then aim to find an active predictor that minimizes this criterium, or at least ensures a value below some threshold. Second, we plan to study the notion of prediagnosis introduced in [2] that combines predictability and diagnosability for controllable systems. Finally, we also want to study active predictability for probabilistic systems, as we had previously done for diagnosis in [1].

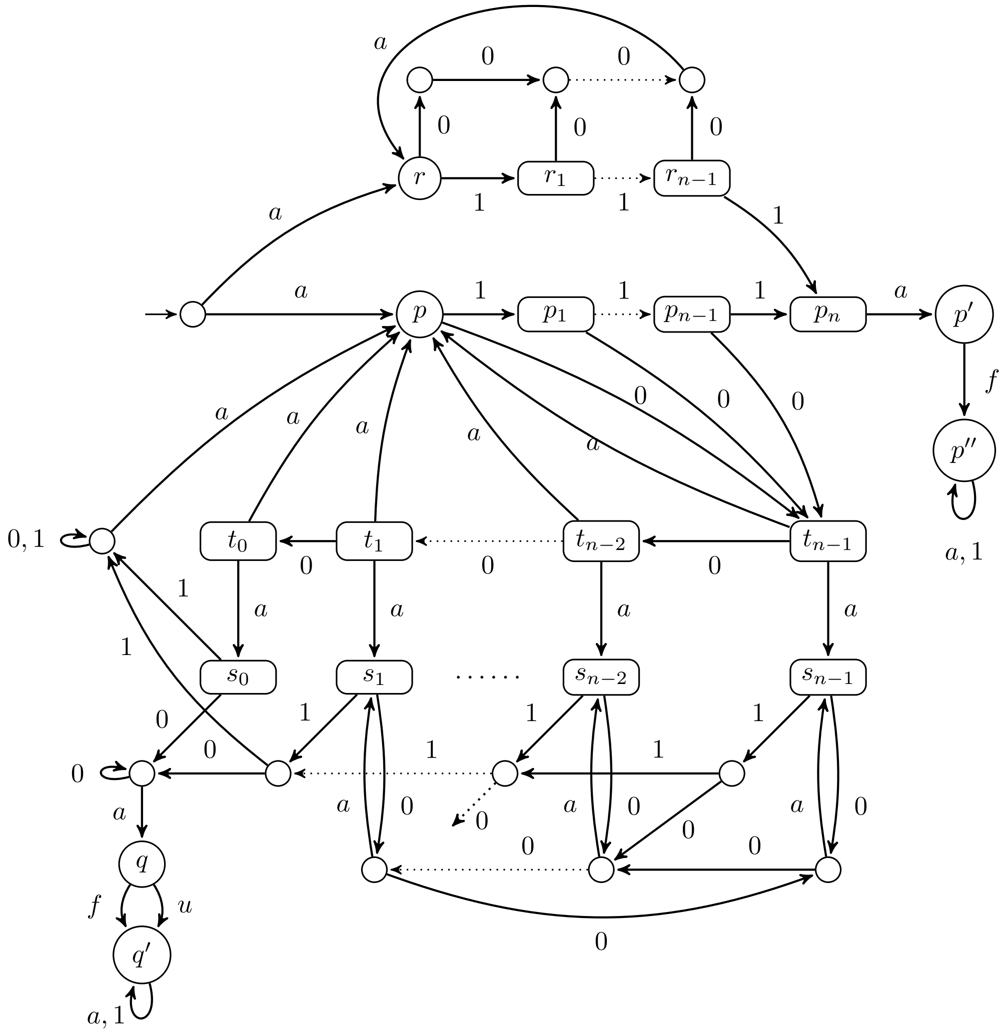

Figure 5 Variant of Figure 4 with constant-size alphabet, with $\Sigma_o = \Sigma_c = \{0, 1, a\}$.

References

1 N. Bertrand, E. Fabre, S. Haar, S. Haddad, and L. Hélouët. Active diagnosis for probabilistic systems. In *FOSSACS 2014, Grenoble, France*, volume 8412 of *LNCS*, pages 29–42, 2014.

2 N. Bertrand, S. Haddad, and E. Lefaucheux. Foundation of Diagnosis and Predictability in Probabilistic Systems. In *IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS'14)*, volume 29 of *LIPIcs*, pages 417–429, New Delhi, India, December 2014.

3 D. Berwanger and L. Doyen. On the power of imperfect information. In *Proc. FSTTCS*, volume 2 of *LIPICS*, pages 73–82, Bangalore, India, 2008.

4 S. Böhm, S. Haar, S. Haddad, P. Hofman, and S. Schwoon. Active diagnosis with observable quiescence. In *Proc. CDC: 54th IEEE Conf. on Decision and Control*, pages 1663–1668, Osaka, Japan, December 2015.

5 L. Brandán Briones and A. Madalinski. Bounded predictability for faulty discrete event systems. In *30nd International Conference of the Chilean Computer Science Society, SCCC*, pages 142–146, Curico, Chile, November 2011.

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

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

8 F. Cassez and S. Tripakis. Fault diagnosis with static and dynamic observers. *Fundam. Informaticae*, 88(4):497–540, 2008.

9 E. Chanthery and Y. Pencolé. Monitoring and active diagnosis for discrete-event systems. In *Proc. SafeProcess'09*, pages 1545–1550, 2009.

10 E. Dallal and S. Lafortune. On most permissive observers in dynamic sensor activation problems. *IEEE Trans. Autom. Control.*, 59(4):966–981, 2014.

11 S. Genc and S. Lafortune. Predictability of event occurrences in partially-observed discrete-event systems. *Autom.*, 45(2):301–311, 2009. `doi:10.1016/j.automatica.2008.06.022`.

12 E. Grädel, W. Thomas, and T. Wilke, editors. *Automata, Logics, and Infinite Games: A Guide to Current Research*, volume 2500 of *Lecture Notes in Computer Science*. Springer, 2002.

13 S. Haar, S. Haddad, T. Melliti, and S. Schwoon. Optimal constructions for active diagnosis. *Journal of Computer and System Sciences*, 83(1):101–120, 2017.

14 Stefan Haar, Serge Haddad, Stefan Schwoon, and Lina Ye. Active Prediction for Discrete Event Systems. working paper or preprint, September 2020. URL: `https://hal.archives-ouvertes.fr/hal-02951944`.

15 A. Madalinski and V. Khomenko. Predictability verification with parallel LTL-X model checking based on Petri net unfoldings. *IFAC Proceedings Volumes*, 45(20):1232–1237, 2012. 8th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes.

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

17 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.

18 L. Ye, P. Dague, and F. Nouioua. Predictability Analysis of Distributed Discrete Event Systems. In *52nd IEEE Conference on Decision and Control*, pages 5009–5015, Florence, Italy, December 2013.

19 X. Yin and S. Lafortune. A uniform approach for synthesizing property-enforcing supervisors for partially-observed discrete-event systems. *IEEE Trans. Autom. Control.*, 61(8):2140–2154, 2016.

20 X. Yin and S. Lafortune. A general approach for optimizing dynamic sensor activation for discrete event systems. *Autom.*, 105:376–383, 2019.

21 X. Yin and Z. Li. Decentralized fault prognosis of discrete event systems with guaranteed performance bound. *Autom.*, 69:375–379, 2016.

22 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.