Monitorability Under Assumptions[⋆]

Thomas A. Henzinger and N. Ege Saraç

IST Austria, Klosterneuburg, Austria
{tah,ege.sarac}@ist.ac.at

Abstract. We introduce the monitoring of trace properties under assumptions. An assumption limits the space of possible traces that the monitor may encounter. An assumption may result from knowledge about the system that is being monitored, about the environment, or about another, connected monitor. We define monitorability under assumptions and study its theoretical properties. In particular, we show that for every assumption A, the boolean combinations of properties that are safe or co-safe relative to A are monitorable under A. We give several examples and constructions on how an assumption can make a non-monitorable property monitorable, and how an assumption can make a monitorable property monitorable with fewer resources, such as integer registers.

1 Introduction

Monitoring is a run-time verification technique that checks, on-line, if a given trace of a system satisfies a given property [3]. The trace is an infinite sequence of observations, and the property defines a set of "good" traces. The monitor watches the trace, observation by observation, and issues a positive verdict as soon as all infinite extensions of the current prefix are good, and a negative verdict as soon as all infinite extensions of the current prefix are bad. The property is *monitorable* if every prefix of every trace has a finite extension that allows a verdict, positive or negative [17]. All safety and co-safety properties, and their boolean combinations, are monitorable [5,10].

The above definition of monitorability assumes that the system may generate any trace. Often a stronger assumption is possible: in predictive monitoring, the monitor has partial knowledge of the system and, therefore, can partly predict the future of a trace [7,18,8,16]; in real-time monitoring, the monitor can be certain that every trace contains infinitely many clock ticks [12]; in composite monitoring, a secondary monitor can rely on the result of a primary monitor. In all these scenarios, the monitor can assume that the observed trace comes from a limited set A of admissible traces. We say that the given property is *monitorable under assumption* A if every prefix of every trace in A has a finite extension in A that allows a verdict relative to A, that is, either all further, infinite extensions in A are good, or they all are bad.

[⋆] This research was supported in part by the Austrian Science Fund (FWF) under grant Z211-N23 (Wittgenstein Award).

Assumptions can make non-monitorable properties monitorable. Consider the finite alphabet $\{req, ack, other\}$ of observations, and the response property

$$P = \Box(req \to \Diamond ack)$$

that "every *req* is followed by *ack*." The property P is not monitorable because every finite trace can be extended in two ways: by the infinite extension ack^ω which makes the property true, and by the infinite extension req^ω which makes the property false. Now suppose that the monitor can assume that "if any *req* is followed by another *req* without an intervening *ack*, then there will not be another *ack*," or formally:

$$A = \Box(req \to \bigcirc((\neg req)\,\mathcal{W}\,(ack \vee \Box(\neg ack)))).$$

The property P is monitorable under A because every finite prefix in A has the admissible extension $req \cdot req$ which makes the property false[1].

In Section 2, we study the boolean closure and entailment properties of monitoring under assumptions. In Section 3, we study safety and co-safety under assumptions, following [12]. We show that for every assumption A, every property that is safe relative to A, every property that is co-safe relative to A, and all their boolean combinations are monitorable under A. The results of both sections hold also if the universe of properties and assumptions is limited to the ω-regular or the counter-free ω-regular languages, i.e., those properties which can be specified using finite automata over infinite words or linear temporal logic, respectively.

In Section 4, we show that assumptions can reduce the resources needed for monitoring. Following [11], we define *k-register monitorability* for monitors that use a fixed number k of integer registers. A register that is operated by increments, decrements, and tests against zero is called a *counter*. It is known that the k-counter monitorability hierarchy is strict, that is, strictly more properties are $(k+1)$-counter monitorable than are k-counter monitorable, for all $k \geq 0$ [11]. We present a property which requires k counters for monitoring, but can be monitored with $k - \ell$ counters under an assumption that can be monitored with ℓ counters.

Finally, in Section 5, we construct for every property P three assumptions that make P monitorable: first, a liveness assumption A_S that makes P safe relative to A_S, and therefore monitorable under A_S; second, a liveness assumption A_C that makes P co-safe relative to A_C, and therefore monitorable under A_C; and third, a co-safety assumption A_M so that P is monitorable under A_M. We use topological tools for our constructions, most notably the characterization of monitorable properties as those sets, in the Cantor topology on infinite words, whose boundary is nowhere dense [9].

[1] We follow the notation of [13] for temporal logic, where $\mathcal{U}$ is the (strong) *until* operator, and $\mathcal{W}$ is the *unless* (or weak until) operator.

2 Monitorability and Assumptions

Let $\Sigma = \{a, b, \ldots\}$ be a finite alphabet of observations. A *trace* is a finite or infinite sequence of observations. We usually denote finite traces by $s, r, t, u \in \Sigma^*$, and infinite traces by $f \in \Sigma^\omega$. A *property* $P \subseteq \Sigma^\omega$ is a set of infinite traces, and so is an *assumption* $A \subseteq \Sigma^\omega$. For traces $f \in \Sigma^\omega$ and $s \in \Sigma^*$, we write $s \prec f$ iff s is a finite prefix of f, and denote by $\mathit{pref}(f)$ the set of finite prefixes of f. For trace sets $P \subseteq \Sigma^\omega$, we define $\mathit{Pref}(P) = \bigcup_{f \in P} \mathit{pref}(f)$. We denote by $\overline{P}$ the complement of P in Σ^ω.

Intuitively, an assumption limits the universe of possible traces. When there are no assumptions, the system can produce any trace in Σ^ω. However, under an assumption A, all observed traces come from the set A. We extend the classical definition of monitorability [17] to account for assumptions as follows.

Definition 1. *Let P be a property, A an assumption, and $s \in \mathit{Pref}(A)$ a finite trace. The property P is* positively determined under A *by s iff, for all f, if $sf \in A$, then $sf \in P$. Similarly, P is* negatively determined under A *by s iff, for all f, if $sf \in A$, then $sf \notin P$.*

Definition 2. *The property P is* s-monitorable under the assumption A*, where $s \in \mathit{Pref}(A)$ is a finite trace, iff there is a finite continuation r such that $sr \in \mathit{Pref}(A)$ positively or negatively determines P under A. The property P is* monitorable under A *iff it is s-monitorable under A for all finite traces $s \in \mathit{Pref}(A)$. We denote the set of properties that are monitorable under A by* $\mathsf{Mon(A)}$.

For a property P and an assumption A, if $P \cap A \neq \emptyset$, we say that P *specifies under* A the set $P \cap A$. The monitorability of P under A may seem related to the monitorability of $P \cap A$. However, the two concepts are independent as we show in the following remark.

Remark 1. In general, $P \in \mathsf{Mon(A)}$ does not imply $P \cap A \in \mathsf{Mon}(\Sigma^\omega)$. Consider $A = \Box\Diamond c$ and $P = a \vee ((\neg b)\,\mathcal{U}\,(a \wedge \Box\Diamond c))$. The property P specifies $((\neg b)\,\mathcal{U}\,a) \wedge \Box\Diamond c$ under A. Observe that every finite trace $s \in \mathit{Pref}(A)$ can be extended to $sr \in \mathit{Pref}(A)$ which satisfies or violates $((\neg b)\,\mathcal{U}\,a)$. Then, since every infinite extension of sr in A satisfies $\Box\Diamond c$, the finite trace sr positively or negatively determines P under A. Therefore, $P \in \mathsf{Mon(A)}$. However, $P \cap A$ is not s-monitorable under Σ^ω for $s = a$ because for every finite extension r we have $src^\omega \in P \cap A$ and $sra^\omega \notin P \cap A$.

Conversely, $P \cap A \in \mathsf{Mon}(\Sigma^\omega)$ does not imply $P \in \mathsf{Mon(A)}$ either. Consider $A = \Box\neg a$ and $P = \Diamond\Box c$. We have $P \cap A \in \mathsf{Mon}(\Sigma^\omega)$ because, for every $s \in \Sigma^*$, the finite trace sa negatively determines P. However, $P \notin \mathsf{Mon(A)}$ because for every finite trace $s \in \mathit{Pref}(A)$, we have $sc^\omega \in P$ and $sb^\omega \notin P$. We will discuss the upward and downward preservation of monitorability later in this section.

As in the case of monitorability in Σ^ω, the set of monitorable properties under a fixed assumption enjoy the following closure properties.

Theorem 1. *For every assumption A, the set* $\mathsf{Mon}(\mathsf{A})$ *is closed under boolean operations.*

Proof. Let $P, Q \in \mathsf{Mon}(\mathsf{A})$ be two monitorable properties under assumption A.

- $\overline{P} \in \mathsf{Mon}(\mathsf{A})$: If P is positively (resp. negatively) determined under A by a finite trace $s \in \mathit{Pref}(A)$, then s negatively (resp. positively) determines $\overline{P}$ under A.
- $P \cap Q \in \mathsf{Mon}(\mathsf{A})$: Let $s \in \mathit{Pref}(A)$ be a finite trace. Since $P \in \mathsf{Mon}(\mathsf{A})$, there is an extension r such that $sr \in \mathit{Pref}(A)$ positively or negatively determines P under A. Moreover, since $Q \in \mathsf{Mon}(\mathsf{A})$, there exists t such that $srt \in \mathit{Pref}(A)$ positively or negatively determines Q under A. If both properties are positively determined under A by given finite traces, then $P \cap Q$ is positively determined under A by srt. Otherwise, the intersection is negatively determined under A by srt.
- $P \cup Q \in \mathsf{Mon}(\mathsf{A})$: Follows from above points since $P \cup Q = \overline{\overline{P} \cap \overline{Q}}$. □

Next, we switch our focus from boolean operations on properties to boolean operations on assumptions. The following examples demonstrate that monitorability is not preserved under complementation, intersection, nor under union of assumptions.

Example 1. Let $A = \Box\Diamond b$ be an assumption, and $P = \Box\Diamond a \vee (\Box\Diamond b \wedge \Diamond c)$ be a property. Under assumption A, the property P specifies $(\Box\Diamond a \vee \Diamond c) \wedge \Box\Diamond b$. For every $s \in \mathit{Pref}(A)$, the finite trace sc positively determines P under A because every infinite extension of sc in A satisfies the property. Therefore, we have $P \in \mathsf{Mon}(\mathsf{A})$. However, under assumption $\overline{A}$, the property P specifies $\Box\Diamond a \wedge (\neg\Box\Diamond b)$, and $P \notin \mathsf{Mon}(\overline{\mathsf{A}})$. This is because every finite trace $s \in \mathit{Pref}(\overline{A})$ can be extended to either satisfy or violate P under $\overline{A}$, as illustrated by $sa^\omega \in P$ and $sc^\omega \notin P$.

Example 2. Let $A = \Box\neg a$ and $B = \Box\neg b$ be assumptions, and $P = \Box a \vee \Box b \vee (\Box(\neg a) \wedge \Box(\neg b) \wedge \Diamond\Box d)$ be a property. We have $P \in \mathsf{Mon}(\mathsf{A})$ because for every finite prefix $s \in \mathit{Pref}(A)$, the finite trace sbc negatively determines P under A. Similarly, $P \in \mathsf{Mon}(\mathsf{B})$ because for every $s \in \mathit{Pref}(B)$, the finite trace sac negatively determines P under B. However, $P \notin \mathsf{Mon}(\mathsf{A} \cap \mathsf{B})$. If it were, then for every finite $s \in \mathit{Pref}(A \cap B)$ there would exist a finite continuation r such that $sr \in \mathit{Pref}(A \cap B)$ positively or negatively determines P under $A \cap B$. In either case, consider $src^\omega \notin P$ and $srd^\omega \in P$ to reach a contradiction.

Example 3. Let $A = \Box(c \to \Box\Diamond a)$ and $B = (\neg\Box\Diamond b) \wedge \Box(c \to (\neg\Box\Diamond a))$ be assumptions, and $P = \Box\Diamond a \vee \Box\Diamond b$ be a property. We have $P \in \mathsf{Mon}(\mathsf{A})$ because for every $s \in \mathit{Pref}(A)$, the finite trace sc positively determines P under A. Similarly, $P \in \mathsf{Mon}(\mathsf{B})$ because for every $s \in \mathit{Pref}(B)$, the finite trace sc negatively determines P under B. Consider the assumption $A \cup B$, and let $s \in \mathit{Pref}(A \cup B)$ be a finite trace containing c. We know that for each continuation f, either (i) sf has infinitely many a's by assumption A, or (ii) sf has finitely many

a's and finitely many b's by assumption B. If (i) holds, the trace s positively determines P under $A \cup B$. If (ii) holds, the trace s negatively determines P under $A \cup B$. However, we cannot distinguish between the two cases by looking at finite prefixes. Therefore, for every $s \in \mathit{Pref}(A \cup B)$ that contains c, property P is not s-monitorable under $A \cup B$, which implies $P \notin \mathsf{Mon}(\mathsf{A} \cup \mathsf{B})$.

The union is arguably the most interesting boolean operation on assumptions. It is relatively easy to discover strong assumptions that make a given property monitorable. However, in practice, we are interested in assumptions that are as weak as possible, and taking the union of assumptions can be a way to construct such assumptions. Next, we define a relation between two assumptions and a property, in order to capture a special case in which monitorability is preserved under the union of assumptions.

Definition 3. *Let A and B be two assumptions, and P be a property such that $P \in \mathsf{Mon}(\mathsf{A})$ and $P \in \mathsf{Mon}(\mathsf{B})$. The assumptions A and B are* compatible with respect to P *iff for every finite trace $s \in \mathit{Pref}(A)$ that positively (resp. negatively) determines P under A, there is no finite extension r such that $sr \in \mathit{Pref}(B)$ and sr negatively (resp. positively) determines P under B, and vice versa.*

Intuitively, the notion of compatibility prevents contradictory verdicts as in Example 3. Under the supposition of compatibility with respect to a given property, we show that monitorability is preserved under the union of assumptions.

Theorem 2. *Let A and B be assumptions, and P be a property such that $P \in \mathsf{Mon}(\mathsf{A})$ and $P \in \mathsf{Mon}(\mathsf{B})$. If A and B are compatible with respect to P, then $P \in \mathsf{Mon}(\mathsf{A} \cup \mathsf{B})$.*

Proof. Let $s \in \mathit{Pref}(A \cup B)$. We want to show that P is s-monitorable under $A \cup B$. Observe that either $s \in \mathit{Pref}(A)$ or $s \in \mathit{Pref}(B)$. Suppose $s \in \mathit{Pref}(A)$. Since $P \in \mathsf{Mon}(\mathsf{A})$, there is an extension r such that $sr \in \mathit{Pref}(A)$ positively or negatively determines P under A. Suppose sr positively determines P under A.

Observe that either $sr \in \mathit{Pref}(A) \setminus \mathit{Pref}(B)$ or $sr \in \mathit{Pref}(A) \cap \mathit{Pref}(B)$. If $sr \in \mathit{Pref}(A) \setminus \mathit{Pref}(B)$, then sr also positively determines P under $A \cup B$ because all possible continuations of sr come from assumption A. If $sr \in \mathit{Pref}(A) \cap \mathit{Pref}(B)$, since $P \in \mathsf{Mon}(\mathsf{B})$ and the two assumptions are compatible with respect to P, there is an extension t such that srt positively determines P under B, and either $srt \in \mathit{Pref}(B) \setminus \mathit{Pref}(A)$ or $srt \in \mathit{Pref}(A) \cap \mathit{Pref}(B)$. If $srt \in \mathit{Pref}(B) \setminus \mathit{Pref}(A)$, then srt also positively determines P under $A \cup B$ because all possible continuations of srt come from B. If $srt \in \mathit{Pref}(A) \cap \mathit{Pref}(B)$, since sr and srt positively determine P under A and under B, respectively, srt also positively determines P under $A \cup B$.

Cases for $s \in \mathit{Pref}(B)$ and negative determinacy follow from similar arguments. Therefore, $P \in \mathsf{Mon}(\mathsf{A} \cup \mathsf{B})$ since P is s-monitorable under $A \cup B$ for every finite trace $s \in \mathit{Pref}(A \cup B)$. □

Next, we explore the preservation of monitorability under the strengthening and weakening of assumptions. We show that, in general, monitorability is neither downward nor upward preserved. However, for each direction, we identify a special case in which monitorability is preserved. The following is an example of a property that is monitorable under an assumption, but becomes non-monitorable under a stronger assumption.

Example 4. Let $A = \Sigma^\omega$ and $B = \Box\neg a$ be assumptions, and $P = \Box(\neg a) \land \Diamond\Box c$ be a property. Observe that $P \subseteq B \subseteq A$. We have $P \in \mathsf{Mon(A)}$ because for every finite prefix $s \in \mathit{Pref}(A)$, the finite trace sa negatively determines P under A. We claim that $P \notin \mathsf{Mon(B)}$. If it were, then for every finite $s \in \mathit{Pref}(B)$ there would exist a finite continuation r such that $sr \in \mathit{Pref}(B)$ positively or negatively determines P under B. Consider $srb^\omega \notin P$ and $src^\omega \in P$ to reach a contradiction in either case.

In the example above, the stronger assumption removes all prefixes that enable us to reach a verdict. We formulate a condition to avoid this problem, and enable downward preservation as follows.

Theorem 3. *Let A and B be assumptions, and P be a property such that $B \subseteq A$ and $P \cap A = P \cap B$. If $P \in \mathsf{Mon(A)}$ and $B \in \mathsf{Mon(A)}$ such that every prefix that negatively determines B under A has a proper prefix that negatively determines P under A, then $P \in \mathsf{Mon(B)}$.*

Proof. Let $s \in \mathit{Pref}(A)$ be a finite trace and $r, t \in \Sigma^*$ be extensions such that $sr, srt \in \mathit{Pref}(A)$ positively or negatively determine P or B under A.

- Suppose sr positively determines P under A. Then, sr also positively determines P under B since $B \subseteq A$ and $P \cap A = P \cap B$.
- Suppose sr positively determines B under A, and srt positively determines P under A. Then, srt positively determines P under B.
- Suppose sr positively determines B under A, and srt negatively determines P under A. Then, srt negatively determines P under B.
- Suppose sr negatively determines P under A, and srt positively determines B under A. Then, sr negatively determines P under B.
- Suppose sr negatively determines P under A, and srt negatively determines B under A. If we have $t \neq \varepsilon$, then we have $sr \in \mathit{Pref}(B)$ and therefore negatively determines P under B. Otherwise, there is a shortest proper prefix u of sr that negatively determines P under A, and $u \in \mathit{Pref}(B)$, therefore u negatively determines P under B.
- Suppose sr negatively determines B under A, then there is a proper prefix of sr that negatively determines P under A. We can resolve this case as above.

These imply that P is s-monitorable under B for every finite trace $s \in \mathit{Pref}(B)$. Therefore, $P \in \mathsf{Mon(B)}$. □

Next, we move on to the upward preservation of monitorability. We give an example of a property that is monitorable under an assumption, but becomes non-monitorable under a weaker assumption.

Example 5. Let $A = \Sigma^\omega$ and $B = \Box(b \to \Diamond c)$ be assumptions, and $P = \Diamond a \wedge \Box(b \to \Diamond c)$ be a property. Observe that $P \subseteq B \subseteq A$. We have $P \in \mathsf{Mon(B)}$ because for each finite prefix $s \in \mathit{Pref}(B)$, the finite trace sa positively determines P under B. One can verify that $P \notin \mathsf{Mon(A)}$ by supposing that a finite trace $s \in \mathit{Pref}(A)$ positively or negatively determines P under A, and considering $sb^\omega \notin P$ and $sa(bc)^\omega \in P$.

Intuitively, the weaker assumption in the previous example introduces prefixes that prevents us from reaching a verdict. The following theorem provides a condition to ensure that all new prefixes can be extended to reach a verdict.

Theorem 4. *Let A and B be assumptions, and P be a property such that $B \subseteq A$ and $P \cap A = P \cap B$. If $P \in \mathsf{Mon(B)}$ and $B \in \mathsf{Mon(A)}$, then $P \in \mathsf{Mon(A)}$.*

Proof. Let s be a finite trace and r be a finite continuation such that $sr \in \mathit{Pref}(A)$ positively or negatively determines B under A. If sr negatively determines B under A, then it also negatively determines P under A because $B \subseteq A$ and $P \cap A = P \cap B$. Suppose sr positively determines B under A. Since $P \in \mathsf{Mon(B)}$, there is a finite extension t such that $srt \in \mathit{Pref}(B)$ positively or negatively determines P under B. Then, srt also positively or negatively determines P under A. It yields that P is s-monitorable under A for every finite trace $s \in \mathit{Pref}(A)$, hence $P \in \mathsf{Mon(A)}$. □

For many problems in runtime verification, the set of ω-regular and LTL-expressible properties deserve special attention due to their prevalence in specification languages. Therefore, we remark that the results presented in this section still hold true if we limit ourselves to ω-regular or to LTL-expressible properties and assumptions.

3 Safety and Co-safety Properties Under Assumptions

In this section, we extend the notion of relative safety from [12] to co-safety properties, and to general boolean combinations of safety properties, with a focus on monitorability.

Definition 4. *A property P is a* safety property under assumption *A iff there is a set $S \subseteq \Sigma^*$ of finite traces such that, for every trace $f \in A$, we have $f \in P$ iff every finite prefix of f is contained in S. Formally,*

$$\exists S \subseteq \Sigma^* : \forall f \in A : f \in P \iff \mathit{pref}(f) \subseteq S.$$

Equivalently, P is a safety property under assumption *A iff every $f \notin P$ has a finite prefix $s \prec f$ that negatively determines P under A. We denote by $\mathsf{Safe(A)}$ the set of safety properties under assumption A.*

Definition 5. *A property P is a* co-safety property under assumption A *iff there is a set $S \subseteq \Sigma^*$ of finite traces such that, for every trace $f \in A$, we have $f \in P$ iff some finite prefix of f is contained in S. Formally,*

$$\exists S \subseteq \Sigma^* : \forall f \in A : f \in P \iff \mathit{pref}(f) \cap S \neq \emptyset.$$

Equivalently, P is a co-safety property under assumption A *iff every $f \in P$ has a finite prefix $s \prec f$ that positively determines P under A. We denote by* $\mathsf{CoSafe(A)}$ *the set of co-safety properties under assumption A.*

One can observe from these definitions that, for every assumption A and property P, we have $P \in \mathsf{Safe(A)}$ iff $\overline{P} \in \mathsf{CoSafe(A)}$.

Definition 6. *A property P is an* obligation property under assumption A *iff $P = \bigcap_{i=1}^{k}(S_i \cup C_i)$ for some finite $k \geq 0$, where $S_i \in \mathsf{Safe(A)}$ and $C_i \in \mathsf{CoSafe(A)}$ for all $1 \leq i \leq k$. We denote by* $\mathsf{Obl(A)}$ *the set of obligation properties under assumption A.*

The set $\mathsf{Obl(A)}$ is exactly the boolean combinations of properties from $\mathsf{Safe(A)}$ and $\mathsf{CoSafe(A)}$. Therefore, we have $\mathsf{Safe(A)} \subseteq \mathsf{Obl(A)}$ and $\mathsf{CoSafe(A)} \subseteq \mathsf{Obl(A)}$ for every assumption A. Note also that when $A = \Sigma^\omega$, our definitions are equivalent to the classical definitions of safety, co-safety, and obligation properties. Next, we present examples of non-monitorable properties that become safe or co-safe under an assumption.

Example 6. Let $P = ((\neg a)\,\mathcal{U}\,b) \vee \Box\Diamond c$. The property P is not monitorable, thus not safe, because the finite trace a has no extension that positively or negatively determines P. Let $A = \neg\Box\Diamond c$. Then, P specifies $((\neg a)\,\mathcal{U}\,b) \wedge \neg\Box\Diamond c$ under A. Observe that every $f \notin P$ has a finite prefix $s \prec f$ that negatively determines P under A because every such infinite trace in A must have a finite prefix that violates $((\neg a)\,\mathcal{U}\,b)$. Therefore, we get $P \in \mathsf{Safe(A)}$.

Example 7. Let $P = (\neg\Box\Diamond a) \wedge \Diamond b$. The property P is not monitorable, thus not co-safe, because the finite trace b has no extension that positively or negatively determines P. Let $A = \neg\Box\Diamond a$. Then, every $f \in P$ has a finite prefix $s \prec f$ that contains b, which positively determines P under A. Therefore, $P \in \mathsf{CoSafe(A)}$.

For the sets of safety and co-safety properties relative to a given assumption, the following closure properties hold.

Theorem 5. *For every assumption A, the set* $\mathsf{Safe(A)}$ *is closed under positive boolean operations.*

Proof. Let $P, Q \in \mathsf{Safe(A)}$ be two safety properties under assumption A. Let $f \notin (P \cup Q)$ be a trace. Since we also have $f \notin P$, there is a finite prefix $s \prec f$ that negatively determines P under A. Similarly, we have $r \prec f$ that negatively determines Q under A. Assume without loss of generality that s is a prefix of r. Then, r negatively determines $P \cup Q$ under A, and thus $P \cup Q \in \mathsf{Safe(A)}$.

Now, let $f \notin (P \cap Q)$. By a similar argument, we have a prefix $s \prec f$ that negatively determines P under A or Q under A. Then, one can verify that s also negatively determines $P \cap Q$ under A. Therefore, $P \cap Q \in \mathsf{Safe(A)}$. □

Theorem 6. *For every assumption A, the set* $\mathsf{CoSafe(A)}$ *is closed under positive boolean operations.*

Proof. Let $P, Q \in \mathsf{CoSafe(A)}$ be two co-safety properties under assumption A. Observe that $P \cup Q = \overline{\overline{P} \cap \overline{Q}}$ and $P \cap Q = \overline{\overline{P} \cup \overline{Q}}$ where $\overline{P}, \overline{Q} \in \mathsf{Safe(A)}$, and apply Theorem 5. □

By combining Theorems 5 and 6 with the definition of $\mathsf{Obl(A)}$, we obtain the following corollary.

Corollary 1. *For every assumption A, the set* $\mathsf{Obl(A)}$ *is closed under all boolean operations.*

Next, we show that relative safety, co-safety, and obligation properties enjoy downward preservation. In other words, if P is a safety, co-safety, or obligation property under an assumption, then it remains a safety, co-safety, or obligation property under all stronger assumptions.

Theorem 7 ([12]). *Let A and B be assumptions such that $B \subseteq A$. For every property P, if $P \in \mathsf{Safe(A)}$, then $P \in \mathsf{Safe(B)}$.*

Theorem 8. *Let A and B be assumptions such that $B \subseteq A$. For every property P, if $P \in \mathsf{CoSafe(A)}$, then $P \in \mathsf{CoSafe(B)}$.*

Proof. Since $P \in \mathsf{CoSafe(A)}$, we have $\overline{P} \in \mathsf{Safe(A)}$. Then, by Theorem 7, we get $\overline{P} \in \mathsf{Safe(B)}$, which implies that $P \in \mathsf{CoSafe(B)}$. □

Theorem 9. *Let A and B be assumptions such that $B \subseteq A$. For every property P, if $P \in \mathsf{Obl(A)}$, then $P \in \mathsf{Obl(B)}$.*

Proof. By definition, $P = \bigcap_{i=1}^{k} (S_i \cup C_i)$ for some finite $k > 1$, where $S_i \in \mathsf{Safe(A)}$ and $C_i \in \mathsf{CoSafe(A)}$ for each $1 \leq i \leq k$. Theorems 7 and 8 imply that $S_i \in \mathsf{Safe(B)}$ and $C_i \in \mathsf{CoSafe(B)}$ for every $1 \leq i \leq k$. Therefore, $P \in \mathsf{Obl(B)}$. □

Finally, we show that every safety, co-safety, and obligation property relative an assumption A is monitorable under A.

Theorem 10. *For every assumption A, we have* $\mathsf{Safe(A)} \subseteq \mathsf{Mon(A)}$.

Proof. Let $P \in \mathsf{Safe(A)}$ be a property and $s \in \mathit{Pref}(A)$ be a finite trace. If there is a continuation f such that $sf \notin P$, then there is a finite prefix $r \prec sf$ that negatively determines P under A. Otherwise, s itself positively determines P under A. In either case, P is s-monitorable under A for an arbitrary finite trace $s \in \mathit{Pref}(A)$, and thus $P \in \mathsf{Mon(A)}$. □

Theorem 11. *For every assumption A, we have* $\mathsf{CoSafe(A)} \subseteq \mathsf{Mon(A)}$.

Proof. The proof idea is the same as in Theorem 10. Let $P \in \mathsf{CoSafe(A)}$ be a property and $s \in \mathit{Pref}(A)$ be a finite trace. If there is a continuation f such that $sf \in P$, then there is a finite prefix $r \prec sf$ that positively determines P under A. Otherwise, s itself negatively determines P under A. In either case, P is s-monitorable under A for an arbitrary finite trace $s \in \mathit{Pref}(A)$, and thus $P \in \mathsf{Mon(A)}$.

Theorem 12. *For every assumption A, we have* $\mathsf{Obl(A)} \subseteq \mathsf{Mon(A)}$.

Proof. Let $P \in \mathsf{Obl(A)}$ be a property. We can rewrite P as $\bigcap_{i=1}^{k}(S_i \cup C_i)$ for some finite $k > 0$ such that $S_i \in \mathsf{Safe(A)}$ and $C_i \in \mathsf{CoSafe(A)}$. By Theorems 10 and 11, each S_i and C_i is in $\mathsf{Mon(A)}$. By Theorem 1, each $S_i \cap C_i$ and their union is in $\mathsf{Mon(A)}$. Therefore, $P \in \mathsf{Mon(A)}$. □

We note that, as in Section 2, the results of this section still hold when restricted to the ω-regular or to the LTL-expressible properties and assumptions.

4 Register Monitorability

In this section, we study monitorability under assumptions for an operational class of monitors, namely, register machines. We follow [11] to define register machines. Let X be a set of *registers* storing integer variables, and consider an *instruction set* of integer-valued and boolean-valued expressions over X. An *update* is a mapping from registers to integer-valued expressions, and a *test* is a boolean-valued expression. We denote the set of updates and tests over the set X of registers by $\Gamma(X)$ and $\Phi(X)$, respectively. We define a *valuation* as a mapping $v : X \to \mathbb{Z}$ from the set of registers to integers. For every update $\gamma \in \Gamma(X)$, we define the *updated valuation* $v[\gamma] : X \to \mathbb{Z}$ by letting $v[\gamma](x) = v(\gamma(x))$ for every $x \in X$. A test $\phi \in \Phi(X)$ is true under the valuation v iff $v \models \phi$.

Definition 7. *A* register machine *is a tuple $M = (X, Q, \Sigma, \Delta, q_0, \Omega)$ where X is a finite set of registers, Q is a finite set of states, Σ is a finite alphabet, $\Delta \subseteq Q \times \Sigma \times \Phi(X) \times \Gamma(X) \times Q$ is a set of edges, $q_0 \in Q$ is the initial state, and $\Omega \subseteq Q^\omega$ is a set of accepting runs, such that for every state $q \in Q$, letter $\sigma \in \Sigma$, and valuation v, there is one and only one outgoing edge $(q, \sigma, \phi, \gamma, r) \in \Delta$ with $v \models \phi$, i.e., the machine is deterministic.*

Let $M = (X, Q, \Sigma, \Delta, q_0, \Omega)$ be a register machine. A *configuration* of M is a pair (q, v) consisting of a state $q \in Q$ and a valuation $v : X \to \mathbb{Z}$. A *transition* $\xrightarrow{\sigma}$ between two configurations of M is defined by the relation $(q, v) \xrightarrow{\sigma} (q', v')$ iff $v' = v[\gamma]$ and $v \models \phi$ for some edge $(q, \sigma, \phi, \gamma, q') \in \Delta$. A *run* of M over a word $w = \sigma_1\sigma_2\ldots$ is an infinite sequence of transitions $(q_0, v_0) \xrightarrow{\sigma_1} (q_1, v_1) \xrightarrow{\sigma_2} \cdots$ where $v_0(x) = 0$ for all $x \in X$. The word $w \in \Sigma^\omega$ is *accepted* by M iff its (unique) run over w yields an infinite sequence $q_0q_1q_2\ldots$ of states which belongs to Ω. The set of infinite words accepted by M is called the *language* of M, and denoted $\mathcal{L}(M)$.

Register machines are a more powerful specification language for traces than finite-state monitors. Even when confined to safety, this model can specify many interesting properties beyond ω-regular, as explored in [11]. Formally, we can limit our model to safety properties as follows: let $q_{sink} \in Q$ be a *rejecting state* such that there are no edges from q_{sink} to any state in $Q \setminus \{q_{sink}\}$, and let Ω be the set of infinite state sequences that do not contain q_{sink}. Under these conditions, $\mathcal{L}(M)$ is a safety property monitored by M. Next, we introduce assumptions to register monitorability.

Definition 8. *Let P be a property, A an assumption, and $s \in \mathit{Pref}(A)$ a finite trace. The property P is* positively k-register determined under *A by s iff there is a register machine M with k registers such that, for all $f \in \Sigma^\omega$, if $sf \in \mathcal{L}(M) \cap A$, then $sf \in P$. Similarly, P is* negatively k-register determined under *A by s iff there is a register machine M with k registers such that, for all $f \in \Sigma^\omega$, if $sf \in A \setminus \mathcal{L}(M)$, then $sf \notin P$.*

Definition 9. *A property P is* k-register monitorable under assumption *A iff for every finite trace $s \in \mathit{Pref}(A)$ there is a finite extension r such that P is positively or negatively k-register determined under A by $sr \in \mathit{Pref}(A)$. We denote the set of properties that are k-register monitorable under A by* $\mathsf{k\text{-}RegMon(A)}$.

In the following, we restrict ourselves to a simple form of register machines in order to demonstrate how assumptions help for monitoring non-regular properties.

Definition 10. *A* counter machine *is a register machine with the instructions $x+1$, $x-1$, and $x=0$ for all registers $x \in X$. We write* $\mathsf{k\text{-}CtrMon(A)}$ *for the set of properties that are monitorable by k-counter machines under assumption A.*

Computational resources play an important role in register monitorability. As proved in [11], for every $k \geq 0$ there is a safety property that can be monitored with k counters but not with $k-1$ counters, that is, the set $\mathsf{k\text{-}CtrMon(\Sigma^\omega)} \setminus \mathsf{(k-1)\text{-}CtrMon(\Sigma^\omega)}$ is non-empty. We now show that assumptions can be used to reduce the number of counters needed for monitoring.

Theorem 13. *Let $\Sigma_k = \{0, 1, \ldots, k\}$. For every $k \geq 1$ and $1 \leq \ell \leq k$, there exist a safety property $P_k \in \mathsf{k\text{-}CtrMon(\Sigma^\omega)} \setminus \mathsf{(k-1)\text{-}CtrMon(\Sigma^\omega)}$ and a safety assumption $A_\ell \in \ell\text{-}\mathsf{CtrMon(\Sigma_k^\omega)}$ such that $P_k \in \mathsf{(k-\ell)\text{-}CtrMon(A_\ell)}$.*

Proof. For every letter $\sigma \in \Sigma$ and finite trace $s \in \Sigma^*$, let $|s|_\sigma$ denote the number of occurrences of σ in s. Let $P_k = \{f \in \Sigma_k^\omega \mid \forall 0 \leq i < k : \forall s \prec f : |s|_i \leq |s|_{i+1}\}$. We can construct a k-counter machine M that recognizes P_k as follows. For each $0 \leq i < k$, the counter x_i of M tracks the difference between $|s|_i$ and $|s|_{i+1}$ by decrementing with letter i and incrementing with $i+1$. The machine keeps running as long as every counter value is non-negative, and rejects otherwise. Notice that we can rewrite $P_k = \bigcap_{i=0}^{k-1} S_i$ where $S_i = \{f \in \Sigma_k^\omega \mid \forall s \prec f : |s|_i \leq |s|_{i+1}\}$ and $S_i \in \mathsf{1\text{-}CtrMon(\Sigma_k^\omega)}$. Then, for each $1 \leq \ell \leq k$, we can construct an assumption $A_\ell = \bigcap_{i=0}^{\ell-1} S_i$ where $A_\ell \in \ell\text{-}\mathsf{CtrMon(\Sigma_k^\omega)}$. Since the properties S_0 to $S_{\ell-1}$ are true under assumption A_ℓ, we only need to monitor the remaining conditions S_ℓ to S_{k-1}. Therefore, it is not hard to verify that $P_k \in \mathsf{(k-\ell)\text{-}CtrMon(A_\ell)}$. □

5 Using Topology to Construct Assumptions

Let X be a topological space, and $S \subseteq X$ be a set. The set S is *closed* iff it contains all of its limit points. The complement of a closed set is *open*. The *closure* of S is the smallest closed set containing S, denoted $cl(S)$. Similarly, the *interior* of S is the largest open set contained in S, denoted $int(S)$. The *boundary* of S contains those points in the closure of S that do not belong to the interior of S, that is, $bd(P) = cl(P) \setminus int(P)$. The set S is *dense* iff every point in X is either in S or a limit point of S, that is, $cl(S) = X$. Similarly, S is *nowhere dense* iff $int(cl(S)) = \emptyset$. For the operations in relative topology induced by X on a subspace $Y \subseteq X$, we use $cl_Y(S)$, $int_Y(S)$, and $bd_Y(S)$ where $S \subseteq Y$.

The safety properties correspond to the closed sets in the Cantor topology on Σ^ω, and the liveness properties correspond to the dense sets [1]. Moreover, the co-safety properties are the open sets [6], and the monitorable properties are the sets whose boundary is nowhere dense [9]. Since these topological characterizations extend to subsets of Σ^ω through relativization [12,9], we use them to construct assumptions under which properties become safe, co-safe, or monitorable.

Theorem 14. *For every property P, there is a liveness assumption A such that $P \in \mathsf{Safe(A)}$ [12]. Moreover, if P is not live, then $P \subset A$; and if P is not safe, then for every assumption B such that $A \subset B$, we have $P \notin \mathsf{Safe(B)}$.*

Proof. Using the standard construction, we can rewrite P as an intersection of a safety property and a liveness property. Formally, $P = P_S \cap P_L$ where $P_S = cl(P)$ is the smallest safety property that contains P, and $P_L = \overline{P_S \setminus P}$ is a liveness property [1]. Let $A = P_L$. We know by Theorem 7 that $P_S \in \mathsf{Safe(A)}$. Since $P_S \cap A = P \cap A$, we also have $P \in \mathsf{Safe(A)}$. Also, if P is not live, we have $P_S \subset \Sigma^\omega$, and $A = \overline{P_S} \cup P$ strictly contains P.

Now, let B be an assumption such that $A \subset B$. Then,

$$cl_B(P \cap B) = cl(P) \cap B = P_S \cap B$$

strictly contains $P \cap B$ because there is a trace $f \in (P_S \setminus P) \cap B$ by construction. It implies that $P \cap B$ is not closed in B, therefore $P \notin \mathsf{Safe(B)}$. □

Intuitively, the construction in the proof of Theorem 14 removes all traces in $\overline{P}$ which have no prefix that negatively determines P. We can alternatively exclude the traces in P which have no prefix that positively determine P, in order to turn P into a relative co-safety property.

Theorem 15. *For every property P, there is a liveness assumption A such that $P \in \mathsf{CoSafe(A)}$. Moreover, if $\overline{P}$ is not live, then $P \cap A \neq \emptyset$; and if P is not co-safe, then for every assumption B such that $A \subset B$, we have $P \notin \mathsf{CoSafe(B)}$.*

Proof. Let $P_C = int(P)$ be the largest co-safety property contained in P, and $A = \overline{P \setminus P_C}$ be an assumption. The assumption A is live since $int(P \setminus P_C) \subseteq int(P) \setminus int(P_C) = \emptyset$. We know by Theorem 8 that $P_C \in \mathsf{CoSafe(A)}$. Then, because $P_C \cap A = P \cap A$, we also have $P \in \mathsf{CoSafe(A)}$. Also, if $\overline{P}$ is not live, we have $P_C \neq \emptyset$, and thus $P \cap A \neq \emptyset$ by construction.

Now, let B be an assumption such that $A \subset B$. Then,

$$int_B(P \cap B) = int((P \cap B) \cup \overline{B}) \cap B = int(P) \cap B = P_C$$

is strictly contained in $P \cap B$ since there is a trace $f \in (P \setminus P_C) \cap B$ by construction. It implies that $P \cap B$ is not open in B, therefore $P \notin \mathsf{CoSafe(B)}$. □

Notice that we removed elements from $cl(P) \setminus P$ and $P \setminus int(P)$ in the above constructions. The union of these two regions corresponds to $bd(P)$, and a property P is monitorable iff $bd(P)$ is nowhere dense [9], that is, $int(cl(bd(P))) = \emptyset$. Since boundary sets are closed in general, and $cl(S) = S$ for every closed set S, this condition is equivalent to $int(bd(P)) = \emptyset$. Now, we describe a construction to make any property monitorable by removing a subset of $bd(P)$ from Σ^ω.

Theorem 16. *For every property P, there is a co-safety assumption A such that $P \in \mathsf{Mon(A)}$. Moreover, if $\overline{P}$ is not live, then $P \cap A \neq \emptyset$.*

Proof. We want to construct a subspace $A \subseteq \Sigma^\omega$ such that $int_A(bd_A(P \cap A)) = \emptyset$. Note that $bd_A(P \cap A) \subseteq bd(P \cap A) \cap A$ and $int_A(P \cap A) = int((P \cap A) \cup \overline{A}) \cap A$. Then, we have

$$int_A(bd_A(P \cap A)) \subseteq int((bd(P \cap A) \cap A) \cup \overline{A}) \cap A.$$

Since union of interiors is contained in interior of unions and we want the expression on the right-hand side to be empty, we have

$$int(bd(P \cap A) \cap A) \cup int(\overline{A}) \subseteq int((bd(P \cap A) \cap A) \cup \overline{A}) \subseteq int(\overline{A}).$$

It implies that $int(bd(P \cap A) \cap A) \subseteq int(\overline{A})$, and since $bd(P \cap A) \cap A$ and $\overline{A}$ are disjoint, we get $int(bd(P \cap A) \cap A) = \emptyset$. Then,

$$\begin{aligned} int(bd(P \cap A) \cap A) &\subseteq int((bd(P) \cup bd(A)) \cap A) \\ &= int((bd(P) \cap A) \cup (bd(A) \cap A)). \end{aligned}$$

Now, we can pick A to be open to have $bd(A) \cap A = \emptyset$, which yields

$$\begin{aligned} int((bd(P) \cap A) \cup (bd(A) \cap A)) &= int(bd(P) \cap A) \\ &= int(bd(P)) \cap A \end{aligned}$$

since interior of finite intersection equals intersection of interiors and A is open. At this point, we want $int(bd(P)) \cap A = \emptyset$ such that A is open. It is equivalent to choosing A such that $\overline{A}$ is a closed set containing $int(bd(P))$, for which the smallest such choice is $\overline{A} = cl(int(bd(P)))$. Therefore, we let $A = \overline{cl(int(bd(P)))}$. Observe that A is indeed open, i.e., a co-safety assumption. Since we obtained that $int_A(bd_A(P \cap A)) = \emptyset$ if $int(bd(P)) \cap A = \emptyset$ and A is open, we have $P \in \mathsf{Mon(A)}$.

Finally, given that $\overline{P}$ is not live, we get $int(P) \neq \emptyset$. It implies that $bd(P) \subset \Sigma^\omega$. Then, since $int(P) \subseteq \overline{bd(P)} \subseteq A$ and $int(P) \subseteq P$, we obtain that $P \cap A \neq \emptyset$. □

Since both ω-regular and LTL-definable languages are closed under the topological closure [2,15], the constructions presented in this section can be performed within the restricted universe of ω-regular or LTL-definable languages. In other words, given an ω-regular (resp. LTL-definable) property, the constructions from the proofs of Theorems 14, 15, and 16 produce ω-regular (resp. LTL-definable) assumptions. Note also that, if P is safe, co-safe, or monitorable under Σ^ω, respectively, then all three constructions yield $A = \Sigma^\omega$.

As pointed out in the previous theorems, the constructions are useful only for certain classes of properties. To demonstrate this, consider a liveness property P such that $\overline{P}$ is also live, that is, P is both live and co-live. Such properties are said to have zero monitoring information [14]. For example, $P = \Box\Diamond a$ is a property with zero monitoring information because there is no finite prefix s such that P is s-monitorable under Σ^ω. Since P is live, we have $cl(P) = \Sigma^\omega$, and since $\overline{P}$ is live, we have $int(P) = \emptyset$. It follows that $bd(P) = \Sigma^\omega$. Therefore, if we let A_S, A_C, and A_M be assumptions as constructed in Theorems 14, 15, and 16, respectively, we obtain $A_S = P$, $A_C = \Sigma^\omega \setminus P$, and $A_M = \emptyset$.

Next, we present an example of a non-monitorable property that is neither live nor co-live, and apply the constructions described in this section.

Example 8. Let $P = (a \vee \Box\Diamond a) \wedge \bigcirc b$. One can verify that $cl(P) = \bigcirc b$ and $int(P) = a \wedge \bigcirc b$ by constructing the corresponding Büchi automata. Then, we also get $bd(P) = (\neg a) \wedge \bigcirc b$. We now apply the constructions described above. If we let $A_S = \overline{cl(P)} \cup P$, we get $P \in \mathsf{Safe}(\mathsf{A_S})$ because every finite trace in A_S that satisfies $\bigcirc\neg b$ negatively determines P under A_S. If we let $A_C = \overline{P} \cup int(P)$, we get $P \in \mathsf{CoSafe}(\mathsf{A_C})$ because every finite trace in A_C that satisfies $a \wedge \bigcirc b$ positively determines P under A_C. Now, observe that $cl(int(bd(P))) = bd(P)$. Then, we have $A_M = a \vee \bigcirc(\neg b)$, which yields that P specifies $a \wedge \bigcirc b$ under A_M, and therefore $P \in \mathsf{Mon}(\mathsf{A_M})$. Note that both A_S and A_C are live, while A_M is co-safe.

Finally, we apply the construction from the proof of Theorem 16 to make a non-monitorable liveness property monitorable.

Example 9. Let $\Sigma = \{req, ack, reboot, other\}$ be a finite set of observations, and consider the property

$$P = (\Box(req \rightarrow \Diamond ack) \vee (\neg ack)\,\mathcal{U}\ req) \wedge \Diamond reboot.$$

The property P is live because $cl(P) = \Sigma^\omega$. We can compute its boundary as $bd(P) = \overline{int(P)} = cl(\overline{P})$. Constructing the Büchi automaton for $\overline{P}$ and taking its closure gives us $bd(P) = (ack\,\mathcal{R}\,(\neg req)) \vee \Box\neg reboot$. One can similarly compute $int(bd(P)) = ack\mathcal{R}(\neg req)$, and observe that $cl(int(bd(P))) = int(bd(P))$. Therefore, we have $A = (\neg ack)\mathcal{U}\,req$, which is indeed a co-safety assumption. The property P specifies $((\neg ack)\,\mathcal{U}\ req) \wedge \Diamond reboot$ under A, therefore $P \in \mathsf{Mon}(\mathsf{A})$.

Since the assumption A constructed in the proof of Theorem 16 is co-safe, we get by Theorem 4 that $P \cap A$ is also monitorable. However, as explained in

Section 2, this is not necessarily the case in general for monitorability under assumptions. We can look for other ways of constructing an assumption A such that P is monitorable under A, but $P \cap A$ is not necessarily monitorable. For this, Theorem 4 may prove useful, and we aim to explore it in future work.

6 Conclusion

Inspired by the notion of relative safety [12], we defined the concepts of co-safety and monitorability relative to an assumption. Assumptions may result from knowledge about the system that is being monitored (as in *predictive* monitoring [18]), knowledge about the environment (e.g., time always advances), or knowledge about other, connected monitors. In further work, we plan to develop a theory of composition and refinement for monitors that use assumptions, including assume-guarantee monitoring, where two or more monitors are connected and provide information to each other (as in *decentralized* monitoring [4]). We gave several examples and constructions on how an assumption can make a non-monitorable property monitorable. In the future, we intend to study the structure of the *weakest* assumptions that make a given property monitorable, particularly the conditions under which such assumptions are unique. Finally, we showed how an assumption can make a monitorable property monitorable with fewer integer registers. More generally, carefully chosen assumptions can make monitors less costly (use fewer resources), more timely (reach verdicts quicker), and more precise (in the case of quantitative verdicts). In further work, we will study all of these dimensions to provide a theoretical foundation for the practical design of a network of monitors with assumptions.

References

1. Alpern, B., Schneider, F.B.: Defining liveness. Information Processing Letters **21**(4), 181 – 185 (1985). https://doi.org/https://doi.org/10.1016/0020-0190(85)90056-0, `http://www.sciencedirect.com/science/article/pii/0020019085900560`
2. Alpern, B., Schneider, F.B.: Recognizing safety and liveness. Distributed computing **2**(3), 117–126 (1987)
3. Bartocci, E., Falcone, Y., Francalanza, A., Reger, G.: Introduction to runtime verification. In: Bartocci, E., Falcone, Y. (eds.) Lectures on Runtime Verification - Introductory and Advanced Topics, Lecture Notes in Computer Science, vol. 10457, pp. 1–33. Springer (2018). https://doi.org/10.1007/978-3-319-75632-5_1, `https://doi.org/10.1007/978-3-319-75632-5_1`
4. Bauer, A., Falcone, Y.: Decentralised ltl monitoring. In: Giannakopoulou, D., Méry, D. (eds.) FM 2012: Formal Methods. pp. 85–100. Springer Berlin Heidelberg, Berlin, Heidelberg (2012)
5. Bauer, A., Leucker, M., Schallhart, C.: Runtime verification for ltl and tltl. ACM Trans. Softw. Eng. Methodol. **20**(4) (Sep 2011). https://doi.org/10.1145/2000799.2000800, `https://doi.org/10.1145/2000799.2000800`
6. Chang, E., Manna, Z., Pnueli, A.: The safety-progress classification. In: Logic and Algebra of Specification, pp. 143–202. Springer (1993)

7. Chen, F., Serbanuta, T., Rosu, G.: jpredictor. In: 2008 ACM/IEEE 30th International Conference on Software Engineering. pp. 221–230 (2008)
8. Cimatti, A., Tian, C., Tonetta, S.: Assumption-based runtime verification with partial observability and resets. In: Finkbeiner, B., Mariani, L. (eds.) Runtime Verification. pp. 165–184. Springer International Publishing, Cham (2019)
9. Diekert, V., Leucker, M.: Topology, monitorable properties and runtime verification. Theoretical Computer Science **537**, 29–41 (2014)
10. Falcone, Y., Fernandez, J.C., Mounier, L.: What can you verify and Enforce at Runtime? Sotfware Tools for Technology Transfer p. Online First (Apr 2011), https://hal.archives-ouvertes.fr/hal-00497350
11. Ferrère, T., Henzinger, T.A., Saraç, N.E.: A theory of register monitors. In: Proceedings of the 33rd Annual ACM/IEEE Symposium on Logic in Computer Science. pp. 394–403 (2018)
12. Henzinger, T.A.: Sooner is safer than later. Information Processing Letters **43**(3), 135 – 141 (1992). https://doi.org/https://doi.org/10.1016/0020-0190(92)90005-G, http://www.sciencedirect.com/science/article/pii/002001909290005G
13. Manna, Z., Pnueli, A.: The temporal logic of reactive and concurrent systems - specification. Springer (1992). https://doi.org/10.1007/978-1-4612-0931-7, https://doi.org/10.1007/978-1-4612-0931-7
14. Peled, D., Havelund, K.: Refining the safety–liveness classification of temporal properties according to monitorability. In: Models, Mindsets, Meta: The What, the How, and the Why Not?, pp. 218–234. Springer (2019)
15. Petric Maretić, G., Torabi Dashti, M., Basin, D.: Ltl is closed under topological closure. Information Processing Letters **114**(8), 408 – 413 (2014). https://doi.org/https://doi.org/10.1016/j.ipl.2014.03.001, http://www.sciencedirect.com/science/article/pii/S0020019014000386
16. Pinisetty, S., Jéron, T., Tripakis, S., Falcone, Y., Marchand, H., Preoteasa, V.: Predictive runtime verification of timed properties. Journal of Systems and Software **132**, 353 – 365 (2017). https://doi.org/https://doi.org/10.1016/j.jss.2017.06.060, http://www.sciencedirect.com/science/article/pii/S0164121217301310
17. Pnueli, A., Zaks, A.: Psl model checking and run-time verification via testers. In: Misra, J., Nipkow, T., Sekerinski, E. (eds.) FM 2006: Formal Methods. pp. 573–586. Springer Berlin Heidelberg, Berlin, Heidelberg (2006)
18. Zhang, X., Leucker, M., Dong, W.: Runtime verification with predictive semantics. In: NASA Formal Methods Symposium. pp. 418–432. Springer (2012)