Monitoring Discounted Sum Properties

Filip Cano
Institute of Science and Technology Austria, Klosterneuburg, Austria

Thomas A. Henzinger
Institute of Science and Technology Austria, Klosterneuburg, Austria

Konstantin Kueffner
Institute of Science and Technology Austria, Klosterneuburg, Austria

N. Ege Saraç
CISPA Helmholtz Center for Information Security, Saarbrücken, Germany

Abstract

Runtime monitoring of quantitative signals faces a fundamental trade-off between volatility and over-aggregation: instantaneous observations are noisy, while long-run averages obscure local structure. Localisation measures such as discounted averages offer a principled middle ground, yet remain poorly understood in runtime verification. This paper studies discounted sums from a monitoring perspective, in both deterministic and stochastic settings. We formalize the discounted monitoring problem and show that exact, sound monitoring of discounted sums cannot be achieved with finite memory. To overcome this impossibility, we introduce ε-approximately sound monitoring, deriving explicit bounds on memory and observation requirements. We then extend the framework to stochastic processes via expected discounted sums, defining pointwise and uniform (ε, δ)-soundness notions, establishing statistical optimality, and proving impossibility beyond a precision threshold. We also formalize the resource complexity of deterministic discounted monitoring via affine register machines and prove a tight worst-case lower bound. Finally, we present a specification language for arithmetic expressions over multiple discounted sums with synchronous and asynchronous semantics, and evaluate our approach on practical scenarios including algorithmic fairness.

2012 ACM Subject Classification Software and its engineering → Dynamic analysis; Mathematics of computing → Stochastic processes; Software and its engineering → Specification languages; Theory of computation → Quantitative automata

Keywords and phrases Runtime Verification, Probabilistic Systems, Quantitative Verification, Approximate Monitoring

Digital Object Identifier 10.4230/LIPIcs.CONCUR.2026.22

Related Version *Full Version*: `https://doi.org/10.48550/arXiv.2606.25979`

Supplementary Material *Software (Source Code)*: `https://github.com/filipcano/monitoring-discounted-sum-properties` [10]
archived at `swh:1:dir:cb99396415f279c985cd05206b44320227f64245`

Funding This work has been supported by the European Research Council under Grant No.: ERC-2020-AdG 101020093 and ERC-2021-AdG 101055412.
Filip Cano: ERC-2020-AdG 101020093
Thomas A. Henzinger: ERC-2020-AdG 101020093
Konstantin Kueffner: ERC-2020-AdG 101020093
N. Ege Saraç: ERC-2021-AdG 101055412

1 Introduction

Runtime monitoring is a central problem in modern computer science, with applications across systems, networks, machine learning, security, and cyber-physical systems [5, 35]. Traditionally, it has focused either on instantaneous properties or long-run behaviors. However, for quantitative signals, such as CPU temperature, model accuracy, or algorithmic fairness, there is a tension: single observations are noisy, while long-run aggregates can obscure local structure. We illustrate this trade-off with the following motivating examples.

(i) *CPU temperature:* monitoring the temperature of a CPU is essential to prevent hardware damage; the conundrum: a single spike that dissipates quickly may raise an undue alarm, yet aggregating over long stretches of time can hide sustained overheating.

(ii) *Model accuracy:* monitoring a model's accuracy is required to ensure sustained performance as distributions shift; the conundrum: one bad prediction is not indicative of the model's performance, yet a long-run average may fail to adapt to distribution shifts.

(iii) *Algorithmic fairness:* monitoring the fairness of an arbiter may uncover uneven issuing of grants between two clients: in isolation, every grant issued to one client is maximally unfair to the other, yet a limit average can hide the deliberate starvation of a client.

A natural way to resolve this conundrum is to use *localisation measures*, such as the window average, discounted average, or, more generally, kernel-weighted averages. These are essentially convolutions that smooth volatile signals by weighting values around a given time point inversely with their distance. While common in time series analysis, signal processing, and image classification, localisation measures are underexplored in runtime verification. In this paper, we study a (seemingly) trivial and widely used localisation measure – the discounted average – from a runtime verification perspective. Discounted monitoring faces two fundamentally different sources of uncertainty: epistemic uncertainty from unseen past and future values, and statistical uncertainty from noisy observations. This paper treats both explicitly and shows that each leads to distinct impossibility thresholds.

Deterministic Discounted Monitoring. We assume a bi-infinite stream of numerical values $w = (x_i)_{i\in\mathbb{Z}}$. Centered at time index $t \in \mathbb{Z}$, we consider a bi-directional discounted sum, with discount factor $r \in [0, 1)$ for the past and $s \in [0, 1)$ for the future:

$$S_t^{r,s}(w) := \underbrace{\textstyle\sum_{i=1}^{\infty} r^i \cdot x_{t-i}}_{\text{past}} + \underbrace{x_t}_{\text{present}} + \underbrace{\textstyle\sum_{i=1}^{\infty} s^i \cdot x_{t+i}}_{\text{future}}. \tag{1}$$

▶ **Example 1** (Decision Fairness)**.** The stream w is a sequence of binary decisions, where $x_t = 1$ indicates that client 1 was granted access to the resource at time t, while $x_t = 0$ indicates the same for client 0. Suppose we are at time t and client 0 was granted the resource (so $x_t = 0$). Normalising the discounted sum by $\lambda^{r,s} := \frac{r}{1-r} + \frac{s}{1-s} + 1$, we obtain the discounted average. Then $S_t^{r,s}(w)/\lambda^{r,s}$ quantifies how fairly the arbiter treated the clients around time t, where 0.5 indicates perfect fairness. Setting $s = 0$ yields a purely backward-looking notion of fairness, whereas setting $r = 0$ yields a forward-looking notion. In particular, for forward-looking fairness, if the arbiter initially favours client 1, the discount factor dictates how much it must compensate later to remain fair, if possible at all.

In discounted monitoring, time progresses and we slide the discounted sum across the value stream, yielding the sequence of discounted property values. Without loss of generality, our monitors start observing the stream at time 0. After each new observation at time n, the monitor must assess, for every past index $t \in [0; n]$, whether the discounted sum lies inside or outside a target interval $\mathcal{I}$:

$$\cdots,\ S_{t-2}^{r,s}(w),\ S_{t-1}^{r,s}(w),\ S_t^{r,s}(w),\ S_{t+1}^{r,s}(w),\ S_{t+2}^{r,s}(w),\ \cdots$$

$$\uparrow \text{bound error}$$

$$w_0, w_1, \cdots, w_{n-1}, w_n \ \to \mathcal{M}\text{onitor.} \ \to \ (\in \mathcal{I}?)\text{-verdict}$$

The fundamental challenge is *epistemic uncertainty*: the monitor must reason about a function over a bi-infinite stream after observing only n values. Within this setting, we propose a monitor that maintains, for each time index t, an *uncertainty set* containing all possible values the full discounted sum at time t could take. Although the uncertainty sets shrink as more information becomes available, we show that the time it takes to shrink enough to be completely inside or outside the target interval is unbounded. As a solution, we study monitors with a tolerance of ε in their verdict, and obtain an upper bound on the number of observations required to reach a decisive verdict, enabling monitoring with bounded resources at the cost of precision. We call these ε-approximate monitors.

Statistical Discounted Monitoring. In many applications, the value stream is a realisation of a stochastic process $W = (X_i)_{i\in\mathbb{Z}}$. In such settings, we are often interested in expected quantities rather than noisy observations, e.g., the latent temperature or the arbiter's underlying propensity. This leads to our second object of study: the *expected discounted sum*, defined as the bi-directional discounted sum evaluated over conditional expectations,

$$\mathbb{S}_t^{r,s}(W) \coloneqq \underbrace{\textstyle\sum_{i=1}^{\infty} r^i \mathbb{E}_{t-i-1}(X_{t-i})}_{\text{past}} + \underbrace{\mathbb{E}_{t-1}(X_t)}_{\text{present}} + \underbrace{\textstyle\sum_{i=1}^{\infty} s^i \mathbb{E}_{t+i-1}(X_{t+i})}_{\text{future}}. \tag{2}$$

▶ **Example 2** (Bias Fairness)**.** Consider a stochastic arbiter that generates its decision X_t at time t by tossing a coin with probability P_t, where P_t is chosen based on the history up to time t. This yields two processes: the hidden bias process $U = (P_t)_{t\in\mathbb{Z}}$ and the observed outcome process $W = (X_t)_{t\in\mathbb{Z}}$, with $\mathbb{E}_{t-1}(X_t) = P_t$. If we want to assess the arbiter around time t independently of chance, we should evaluate discounted sums (or averages) with respect to the bias process, e.g., $S_t^{r,s}(U) = \mathbb{S}_t^{r,s}(W)$ rather than $S_t^{r,s}(W)$.

In statistical discounted monitoring, we slide both the expected and observed discounted sums over the stochastic stream, yielding expected and observed discounted property values. As before, monitors start observing the realised sequence at time 0. After each new observation at time n, the monitor must assess, for every past index $t \in [0; n]$, whether the *expected* discounted sum lies inside or outside a target interval $\mathcal{I}$ with high probability.

$$\begin{aligned}
&\cdots,\ \mathbb{S}_{t-2}^{r,s}(W),\ \mathbb{S}_{t-1}^{r,s}(W), \mathbb{S}_t^{r,s}(W),\ \mathbb{S}_{t+1}^{r,s}(W),\ \mathbb{S}_{t+2}^{r,s}(W),\ \cdots \\
&\qquad\qquad\qquad\qquad \uparrow \text{bound deviation} \\
&\cdots,\ S_{t-2}^{r,s}(W),\ S_{t-1}^{r,s}(W), S_t^{r,s}(W),\ S_{t+1}^{r,s}(W),\ S_{t+2}^{r,s}(W),\ \cdots \\
&\qquad\qquad\qquad\qquad \uparrow \text{bound error} \\
&W_0, W_1, \cdots, W_{n-1}, W_n \ \to \mathcal{M}\text{onitor.} \ \to\ (\in \mathcal{I}?)\text{-verdict}
\end{aligned}$$

Apart from epistemic uncertainty, in this setting the extra challenge is *statistical inference*: the monitor has to estimate also the deviation between the observed and expected discounted sums. Because of the probabilistic nature of the monitored quantities, the monitor's verdict is accompanied by an error probability δ. Within this setting, we extend ε-approximate monitoring to the probabilistic setting, and define three notions of (ε, δ)-soundness with increasing strength of guarantees: pointwise, local, and uniform; depending on whether the probability of an error is given for each point after a fixed release time (pointwise), for each point with flexible release time (local), or for the whole execution (uniform). As the strength of the guarantee increases, the monitor's precision decreases. We show that the deviation bounds used by our pointwise sound monitors are minimax optimal, which establishes that statistical discounted monitoring is impossible beyond a certain precision.

Generalization to Arithmetic Expressions. Many relevant quantitative properties are naturally expressed as relations between discounted sums rather than as single aggregates. To support discounted versions of such properties, we introduce a richer specification language in which a monitor observes multiple outcome streams and monitors an arithmetic expression over their discounted sums. We distinguish between synchronous and asynchronous semantics for combining discounted sums.

▶ **Example 3** (Demographic Parity). So far we have assumed that both clients always request access to the resource. We model the more general setting using the binary sequences representing: w^{r_1} the requests of client 1, w^{g_1} the grants client 1, w^{r_2} the requests client 2, and w^{g_2} the grants client 2. All indexed by the same global clock. Then we can express the discounted acceptance rate, i.e., a discounted form of demographic parity [16], as the difference between the ratios $\mathrm{S}_t^{r,s}(w^{g_1})/\mathrm{S}_t^{r,s}(w^{r_1})$ and $\mathrm{S}_t^{r,s}(w^{g_2})/\mathrm{S}_t^{r,s}(w^{r_2})$.

Register Complexity. The ε-approximate monitor from the deterministic setting yields a finite observation horizon; let τ denote the number of additional observations after which each monitored position receives a sound verdict up to the ε boundary region. This horizon is first a delay bound: a verdict for time t may require waiting until time $t + \tau$. It also serves as a memory bound: during this waiting period, several positions may remain unresolved simultaneously, and the monitor maintains a separate running discounted sum for each of them. The delay bound, however, only says when verdicts are guaranteed to arrive; it does not characterize how much information must be stored before those verdicts are reached. This leads to the register-complexity question: can the overlapping discounted sums in general be represented more compactly than by one running sum per pending position?

To make this question precise, we formalize monitors as *affine register machines (ARMs)*: finite-state monitors equipped with real-valued registers, affine updates, and linear guards. In this model, we answer this question negatively by proving a worst-case lower bound already for future-only monitoring. For every k, there is a future-only monitor with horizon k that cannot be monitored by any ARM with fewer than k registers.

Experimental Evaluation. Finally, we implement our monitors on scenarios mirroring the motivating examples, and investigate empirically how actual register use compares to our theoretical upper bounds, and how register usage differs between synchronous and asynchronous interpretations of arithmetic properties.

Contributions. The main contributions of this paper are as follows:

1. A formalization of the discounted monitoring problem, and a proof that exact, sound discounted monitoring cannot be achieved with finite memory.
2. An ε-approximately sound notion of discounted monitoring, accompanied by explicit bounds on the required number of observations and registers.
3. An extension of discounted monitoring to stochastic processes via expected discounted sums, including pointwise and uniform (ε, δ)-soundness notions, as well as results on statistical optimality and impossibility beyond a precision threshold.
4. A specification language for arithmetic expressions over multiple discounted sums, supporting both synchronous and asynchronous semantics.
5. A formalization of monitors as affine register machines and a tight bound on the number of registers required for sound ε-approximate monitoring in the future-only case.
6. An implementation and empirical evaluation of the proposed monitors on realistic case studies, analyzing performance and efficiency relative to the proven worst-case guarantees.

2 Monitoring Discounted Sums

2.1 Discounted Sum Property

The main object of study in this paper is the discounted sum operator $S_t^{r,s}$, where $r, s \in [0, 1)$ are the past and future discount factors and $t \in \mathbb{Z}$ is a time index. It acts on bi-infinite sequences $w = (x_t)_{t\in\mathbb{Z}}$ taking values in a nonempty bounded set $\mathcal{R} \subset \mathbb{R}$ of diameter $d_{\mathcal{R}} \coloneqq \sup \mathcal{R} - \inf \mathcal{R}$. We call a tuple $(\mathcal{R}, r, s)$ an *input setting*. Intuitively, $S_t^{r,s}$ is centered at t, discounting past values ($i < t$) by r and future values ($j > t$) by s. Formally, for $n_l \leq t \leq n_u$, let $w_{n_l:n_u} = (x_{n_l}, \dots, x_{n_u})$ and define $S_t^{r,s}(w_{n_l:n_u}) \coloneqq \sum_{i=1}^{t-n_l} r^i x_{t-i} + x_t + \sum_{i=1}^{n_u-t} s^i x_{t+i}$. For a bi-infinite sequence w, define $S_t^{r,s}(w) \coloneqq \sum_{i=1}^{\infty} r^i x_{t-i} + x_t + \sum_{i=1}^{\infty} s^i x_{t+i}$.

Basic properties. We prove some basic properties of discounted sums. First, the discounted sum converges, and consecutive sums differ by at most $d_{\mathcal{R}}$. This follows from the definition.

▶ **Lemma 4.** *Let $(\mathcal{R}, r, s)$ be an input setting, $w \in \mathcal{R}^{\mathbb{Z}}$, and $t \in \mathbb{Z}$. Then $S_t^{r,s}(w)$ exists and $\left|S_t^{r,s}(w) - S_{t-1}^{r,s}(w)\right| \leq d_{\mathcal{R}}$.*

Second, we characterize the values a discounted sum can take. Formally, let $\mathrm{Sums}_{\mathcal{R}}^{r,s} \coloneqq \{S_t^{r,s}(w) \,:\, w \in \mathcal{R}^{\mathbb{Z}}\}$; note that this set does not depend on t. Its shape depends on whether $\mathcal{R}$ has gaps that are too large relative to the discount factors.

▶ **Definition 5** (Proper input setting)**.** *Let $(\mathcal{R}, r, s)$ be an input setting. The* normalization factor *$\lambda^{r,s}$ and the* maximum gap *$\Delta_{\mathcal{R}}$ are defined as*

$$\lambda^{r,s} \coloneqq 1 + \frac{r}{1-r} + \frac{s}{1-s}, \qquad \Delta_{\mathcal{R}} \coloneqq \sup\Big(\{0\} \cup \{\, b - a : \ [a, b] \subseteq [\inf \mathcal{R}, \sup \mathcal{R}] \setminus \mathcal{R}\,\}\Big). \tag{3}$$

We say that $(\mathcal{R}, r, s)$ is proper *if $\Delta_{\mathcal{R}} \leq \min\left\{\frac{r}{1-r}, \frac{s}{1-s}\right\} d_{\mathcal{R}}$.*

The convention $\sup\{0\} = 0$ ensures $\Delta_{\mathcal{R}} = 0$ whenever $\mathcal{R}$ has no gaps in its interval hull, so interval-valued domains are automatically proper. For non-interval domains such as $\mathcal{R} = \{0, 1\}$, properness is a substantive condition: it requires the discount factors to be large enough to interpolate across the gaps.

▶ **Lemma 6.** *Let $(\mathcal{R}, r, s)$ be an input setting.*

(1) $\mathrm{Sums}_{\mathcal{R}}^{r,s} \subseteq \lambda^{r,s} \cdot [\inf \mathcal{R}, \sup \mathcal{R}]$.
(2) *If $(\mathcal{R}, r, s)$ is proper, then* $\mathrm{Closure}(\mathrm{Sums}_{\mathcal{R}}^{r,s}) = \lambda^{r,s} \cdot [\inf \mathcal{R}, \sup \mathcal{R}]$.
(3) *If $(\mathcal{R}, r, s)$ is proper and $\mathcal{R}$ is compact, then* $\mathrm{Sums}_{\mathcal{R}}^{r,s} = \lambda^{r,s} \cdot [\inf \mathcal{R}, \sup \mathcal{R}]$.

The three statements isolate the role of each assumption. Boundedness of $\mathcal{R}$ gives the interval over-approximation in (1). Properness ensures that the closure of the set of sums fills this interval, as stated in (2). Compactness lifts closure equality to equality in (3). For a non-closed domain, the endpoints need not be attainable; under compactness they are attained by the constant sequences at $\inf \mathcal{R}$ and $\sup \mathcal{R}$.

Sums vs. Averages. In some settings it is more natural to think of a discounted average than a discounted sum. A weighted average is a weighted sum whose weights are non-negative and sum to 1. Since the weights of a discounted sum add up to $\lambda^{r,s}$, an input setting $(\mathcal{R}, r, s)$ can be reframed for averages by considering the equivalent setting $(\mathcal{R}/\lambda^{r,s}, r, s)$ and rescaling each sequence $w = (x_t)_{t\in\mathbb{Z}} \in \mathcal{R}^{\mathbb{Z}}$ to $w' = (x_t/\lambda^{r,s})_{t\in\mathbb{Z}} \in (\mathcal{R}/\lambda^{r,s})^{\mathbb{Z}}$.

2.2 Monitoring Problem

A monitor observes a bi-infinite stream of values $w = (x_i)_{i\in\mathbb{Z}}$ from time 0 onward. After observing a finite prefix $w_{0:n}$, it is queried only on observed positions $t \in \{0,\ldots,n\}$. Formally, a monitor computes a function $\mathcal{M}\colon \mathcal{R}^* \times \mathbb{N} \to \{\bot, ?, \top\}$. Only pairs (u,t) with $t < |u|$ are queried. The verdicts $\bot$ and $\top$ indicate that $S_t^{r,s}(w)$ lies outside or inside the target interval $\mathcal{I}$, respectively, and ? indicates that the verdict is inconclusive.

2.2.1 Sound Monitors

▶ **Definition 7** (Sound monitor). *Let $(\mathcal{R}, r, s)$ be an input setting and let $\mathcal{I}$ be a target interval. A monitor is* sound *if, for every $w \in \mathcal{R}^{\mathbb{Z}}$, $n \in \mathbb{N}$, and $t \in \{0,\ldots,n\}$,*

$$\mathcal{M}(w_{0:n}, t) = \bot \implies S_t^{r,s}(w) \notin \mathcal{I} \quad \text{and} \quad \mathcal{M}(w_{0:n}, t) = \top \implies S_t^{r,s}(w) \in \mathcal{I}. \tag{4}$$

Uncertainty. Since the monitor receives values incrementally, it typically returns an inconclusive verdict (?) until the observed prefix contains enough information to commit to a final $\top$ or $\bot$. For $0 \le t \le n$, the *uncertainty set* of a finite prefix $w_{0:n}$ collects the values $S_t^{r,s}$ can take across all extensions of $w_{0:n}$: $U_t(w_{0:n}) \coloneqq \left\{S_t^{r,s}(w') : w' \in \mathcal{R}^{\mathbb{Z}},\ w'_{0:n} = w_{0:n}\right\}$. A sound monitor therefore satisfies $\mathcal{M}(w_{0:n}, t) = \top \implies U_t(w_{0:n}) \subseteq \mathcal{I}$ and $\mathcal{M}(w_{0:n}, t) = \bot \implies U_t(w_{0:n}) \cap \mathcal{I} = \emptyset$. Similarly as in Lemma 6, for proper input settings the closure of the uncertainty set is an interval; the uncertainty set itself is that interval when $\mathcal{R}$ is compact.

▶ **Lemma 8.** *Let $(\mathcal{R}, r, s)$ be an input setting, let $0 \le t \le n$, and let $\gamma_{t,n}^{r,s} \coloneqq \frac{r^{t+1}}{1-r} + \frac{s^{n-t+1}}{1-s}$. Then, $U_t^{r,s}(w_{0:n}) \subseteq S_t^{r,s}(w_{0:n}) + \gamma_{t,n}^{r,s}[\inf\mathcal{R}, \sup\mathcal{R}]$. If $(\mathcal{R}, r, s)$ is proper, then* $\mathrm{Closure}(U_t^{r,s}(w_{0:n})) = S_t^{r,s}(w_{0:n}) + \gamma_{t,n}^{r,s}[\inf\mathcal{R}, \sup\mathcal{R}]$. *If, additionally, $\mathcal{R}$ is compact, then equality holds without taking closures.*

A natural quantity is then how long a sound monitor must wait before it can commit to a final verdict at a given time.

▶ **Definition 9.** *Let $\mathcal{I}$ be a target interval, let $t \in \mathbb{N}$, and let $w \in \mathcal{R}^{\mathbb{Z}}$. The* minimum required observation time *is*

$$\mu_{\mathcal{I}}(t, w) \coloneqq \min\left(\{n \in \mathbb{N} : n \ge t \text{ and } (U_t(w_{0:n}) \subseteq \mathcal{I} \text{ or } U_t(w_{0:n}) \cap \mathcal{I} = \emptyset)\} \cup \{\infty\}\right).$$

For a fixed time t, future observations eliminate the unobserved future contribution, but they do not reveal the values before time 0. Therefore, an irreducible uncertainty of diameter $\frac{r^{t+1}}{1-r} d_{\mathcal{R}}$ may remain, and an exact verdict need not ever become possible. Even when exact verdicts are possible, their delay has no uniform finite bound in general.

▶ **Theorem 10** (No uniform exact verdict delay). *Let $(\mathcal{R}, r, s)$ be a proper input setting such that $\mathcal{R}$ is compact and $r + s > 0$, and let $J \coloneqq \lambda^{r,s}[\inf\mathcal{R}, \sup\mathcal{R}]$. Let $\mathcal{I}$ be a target interval satisfying $J \cap \mathcal{I} \neq \emptyset$ and $J \setminus \mathcal{I} \neq \emptyset$. Then, for every $t \in \mathbb{N}$ and every $n \ge t$, there exists $w \in \mathcal{R}^{\mathbb{Z}}$ such that $\mu_{\mathcal{I}}(t, w) > n$.*

The proof shows that for any proposed finite delay, one can construct a prefix whose remaining uncertainty crosses a boundary of the target interval: one completion yields a discounted sum in the target, while another yields one outside it A sound exact monitor must therefore remain inconclusive, which motivates approximate soundness.

2.2.2 Approximately Sound Monitors

▶ **Definition 11** (ε-approximately sound monitor)**.** *Let $(\mathcal{R}, r, s)$ be an input setting, let $\mathcal{I}$ be a target interval, and let $\varepsilon > 0$. A monitor $\mathcal{M}$ is* ε-approximately sound *if, for every $w \in \mathcal{R}^{\mathbb{Z}}$, $n \in \mathbb{N}$, and $t \in \{0, \ldots, n\}$,*

$$\mathcal{M}(w_{0:n}, t) = \bot \implies S_t^{r,s}(w) \notin \mathcal{I}_{-\varepsilon} \quad \textit{and} \quad \mathcal{M}(w_{0:n}, t) = \top \implies S_t^{r,s}(w) \in \mathcal{I}_{+\varepsilon}. \tag{5}$$

where, for $\mathcal{I} = (L, U)$, $\mathcal{I}_{+\varepsilon} := (L - \varepsilon, U + \varepsilon)$ and $\mathcal{I}_{-\varepsilon} := (L + \varepsilon, U - \varepsilon)$.

Although $\mathcal{I}$, $\mathcal{I}_{+\varepsilon}$, and $\mathcal{I}_{-\varepsilon}$ are open, we note that the results of this section hold for arbitrary intervals. We next give a tight bound on the minimum required observation time for ε-approximately sound monitors, and show how to realise them with finite resources.

▶ **Theorem 12.** *Let $(\mathcal{R}, r, s)$ be an input setting, $\mathcal{I}$ a target interval, $\varepsilon > 0$, and $T \in \mathbb{N}$. With*

$$\tau^*(\mathcal{R}, r, s, \varepsilon, T) := \min\left(\left\{\tau \in \mathbb{N} : d_{\mathcal{R}}\left(\frac{r^{T+1}}{1-r} + \frac{s^{\tau+1}}{1-s}\right) \leq 2\varepsilon\right\} \cup \{\infty\}\right). \tag{6}$$

1. *If $\tau^* < \infty$, there exists an ε-approximately sound monitor that produces a verdict for every $t \geq T$ within τ^* steps.*
2. *Suppose that $\mathcal{R} = [m, M]$ with $m < M$, and write $J = \lambda^{r,s}[m, M]$. If $J \cap \mathcal{I}_{-\varepsilon} \neq \emptyset$ and $J \setminus \mathcal{I}_{+\varepsilon} \neq \emptyset$, then, for every ε-approximately sound monitor and every $\tau < \tau^*$, there exists $w \in \mathcal{R}^{\mathbb{Z}}$ on which the monitor does not produce a verdict for time T after τ steps.*

For the usual case $d_{\mathcal{R}} > 0$ and $0 < s < 1$, let $B_T := \left(\frac{2\varepsilon}{d_{\mathcal{R}}} - \frac{r^{T+1}}{1-r}\right)(1-s)$. If $B_T > 0$, then $\tau^* = \max\{0, \lceil \log_s B_T \rceil - 1\}$. If $B_T \leq 0$, then $\tau^* = \infty$. The semantic definition in Eq. (6) also covers the exceptional cases: $d_{\mathcal{R}} = 0$ gives $\tau^* = 0$, while for $s = 0$ we have $\tau^* = 0$ if $d_{\mathcal{R}} \frac{r^{T+1}}{1-r} \leq 2\varepsilon$ and $\tau^* = \infty$ otherwise. Unless stated otherwise, we assume $0 < \tau^* < \infty$.

For every bounded input domain, Lemma 8 provides an interval enclosure whose diameter at time t, after τ further observations, is $d_{\mathcal{R}}\left(\frac{r^{t+1}}{1-r} + \frac{s^{\tau+1}}{1-s}\right)$. A diameter of at most 2ε forces an approximately sound verdict, which proves the upper bound. For interval-valued input domains, the finite observed contribution can be positioned continuously; under the two nontriviality conditions in point (2), this gives matching accepting and rejecting completions for every smaller delay.

2.3 Monitor Construction

Algorithm 1 constructs an ε-approximately sound monitor. With $\tau = \tau^*(\mathcal{R}, r, s, \varepsilon, T)$ from Eq. (6), it maintains:

- a global variable `Psum` storing the present-inclusive discounted sum of all past observations;
- an array of τ tuples $(\texttt{RunningSum}_i, \texttt{pos}_i)$, one per active monitor;
- a Boolean array `in_use`$[1{:}\tau]$ marking which tuples are active.

Each active tuple tracks a candidate time $\texttt{pos}_i$ whose discounted sum is being monitored. The invariant is that after observing x_n, every active tuple with $p = \texttt{pos}_j$ satisfies $\texttt{RunningSum}_j = S_p^{r,s}(w_{0:n}) = \sum_{i=1}^{p} r^i x_{p-i} + \sum_{i=0}^{n-p} s^i x_{p+i}$.

Algorithm 1 ε-Approximately Sound Discounted-Sum Monitor.

Require: Input setting $(\mathcal{R}, r, s)$, tolerance ε, target interval $\mathcal{I}$, start time T

$\tau \leftarrow \tau^*(\mathcal{R}, r, s, \varepsilon, T)$ ▷ Eq. (6)
$\mathtt{Psum} \leftarrow 0$
for $i = 1$ to τ **do**
  $\mathtt{in\_use}[i] \leftarrow \bot$
end for
for $t = 0, 1, 2, \dots$ **do**
  observe x_t; $\mathtt{Psum} \leftarrow x_t + r \cdot \mathtt{Psum}$ ▷ now $\mathtt{Psum} = x_t + \sum_{i=1}^{t} r^i x_{t-i}$
  for $j = 1$ to τ **do** ▷ Update all active registers
    if $\mathtt{in\_use}[j] = \top$ **then**
      $\mathtt{RunningSum}_j \leftarrow \mathtt{RunningSum}_j + x_t \cdot s^{\,t-\mathtt{pos}_j}$ ▷ future contribution for time $\mathtt{pos}_j$
      compute $C^{r,s}_{\mathtt{pos}_j} = \mathtt{RunningSum}_j + \left(\frac{r^{\mathtt{pos}_j+1}}{1-r} + \frac{s^{t-\mathtt{pos}_j+1}}{1-s}\right) \cdot [\inf \mathcal{R}, \sup \mathcal{R}]$
      if $C^{r,s}_{\mathtt{pos}_j} \subseteq \mathcal{I}_{+\varepsilon}$ **then**
        output verdict $\top$ for time $\mathtt{pos}_j$; $\mathtt{in\_use}[j] \leftarrow \bot$
      else if $C^{r,s}_{\mathtt{pos}_j} \cap \mathcal{I}_{-\varepsilon} = \emptyset$ **then**
        output verdict $\bot$ for time $\mathtt{pos}_j$; $\mathtt{in\_use}[j] \leftarrow \bot$
      end if
    end if
  end for
  if $t \geq T$ **then** ▷ Instantiate new register for current t
    select i with $\mathtt{in\_use}[i] = \bot$ ▷ Guaranteed to exist by Thm. 12
    $\mathtt{RunningSum}_i \leftarrow \mathtt{Psum}$; ▷ initially $S^{r,s}_t(w_{0:t})$ $\mathtt{pos}_i \leftarrow t$; $\mathtt{in\_use}[i] \leftarrow \top$
  end if
end for

Special cases. For future-only sums ($r = 0$), the past uncertainty vanishes. For past-only sums ($s = 0$), later observations cannot refine the uncertainty of an earlier position. Accordingly, the uniform bound in Eq. (6) is either 0 or ∞: each position is tested once at creation time, and if its enclosure is still inconclusive, no later observation can change that.

General bounded input domains. Neither the upper bound in Theorem 12(1) nor Algorithm 1 requires properness. Both use only the interval enclosure from Lemma 8. The interval-domain assumption in Theorem 12(2) is used only to position the observed finite contribution continuously. For input domains with gaps, τ^* remains a valid upper bound but need not be tight for a fixed target interval.

3 Statistical Discounted Monitor

In statistical discounted monitoring, we assume that the observation sequence is generated by a stochastic process and the monitors objective is to estimate the value of the expected discounted sum. The statistical monitors are similar to the monitors of Section 2, but for an additional statistical error term which impacts the release condition (and time τ^*).

Setting. Assume that the value sequence w is a realisation of the stochastic process $W = (X_t)_{t\in\mathbb{Z}}$ in $\mathcal{R}$. Let $\mathcal{F} := (\mathcal{F}_t)_{t\in\mathbb{Z}}$ be its canonical filtration, i.e., $\mathcal{F}_t$ is the sigma-algebra generated by $(X_i)_{i\leq t}$. *No further assumptions* are placed on W.

Expected discounted sum. We are interested in the expected discounted sum, in infinite and finite form. We define the conditional expectation for the integrable random variable Y as $\mathbb{E}_t(Y) \coloneqq \mathbb{E}(Y \mid \mathcal{F}_t)$. We define $\mathbb{S}_t^{r,s}(W) \coloneqq \lim_{n_l \to -\infty} \lim_{n_u \to +\infty} \mathbb{S}_t^{r,s}(W_{n_l:n_u})$

$$\mathbb{S}_t^{r,s}(W_{n_l:n_u}) \coloneqq \sum_{i=1}^{t-n_l} r^i\, \mathbb{E}_{t-i-1}(X_{t-i}) + \sum_{i=0}^{n_u-t} s^i\, \mathbb{E}_{t+i-1}(X_{t+i}) \text{ for } n_l \le t \le n_u,\ r, s \in [0,1).$$

3.1 Monitoring

Statistical monitors compute the observed discounted sum and bound its deviation from the expected discounted sum. This permits three natural notions of statistical soundness, which differ in how many possible verdicts are protected by the same probability guarantee.

▶ **Definition 13** (Statistical soundness). *For $(\mathcal{R}, r, s)$, let $\mathcal{I} = (L, U)$ be a target interval, $\varepsilon \ge 0$ a precision, and $\delta \in (0,1)$ an error probability. For $t \le n \in \mathbb{N}$ define the success event*

$$E_t(n) \coloneqq \{\mathcal{M}(W_{0:n}, t) = \bot \Rightarrow \mathbb{S}_t^{r,s}(W) \notin \mathcal{I}_{-\varepsilon} \wedge \mathcal{M}(W_{0:n}, t) = \top \Rightarrow \mathbb{S}_t^{r,s}(W) \in \mathcal{I}_{+\varepsilon}\},$$

where $\mathcal{I}_{+\varepsilon} \coloneqq (L - \varepsilon, U + \varepsilon)$ and $\mathcal{I}_{-\varepsilon} \coloneqq (L + \varepsilon, U - \varepsilon)$. A statistical monitor is

(i) pointwise (ε, δ)-approximately sound if $\forall t \in \mathbb{N}\, \forall n \ge t \colon \mathbb{P}(E_t(n)) \ge 1 - \delta$;
(ii) locally (ε, δ)-approximately sound if $\forall t \in \mathbb{N} \colon \mathbb{P}(\forall n \ge t \colon E_t(n)) \ge 1 - \delta$;
(iii) uniformly (ε, δ)-approximately sound if $\mathbb{P}(\forall t \in \mathbb{N}\, \forall n \ge t \colon E_t(n)) \ge 1 - \delta$.

The three notions express increasingly stronger guarantees. *Pointwise soundness* protects one fixed verdict at a fixed time t and observation horizon n. Hence, flexible register release is not possible and an error probability of δ must be tolerated for the verdict at each t. *Local soundness* protects one fixed monitored time t for all observation horizons $n \ge t$. Hence, flexible register release is possible, but an error probability of δ remains for the verdict at each t. *Uniform soundness* protects the entire run, i.e., the invariant "every verdict issued for every time index and at every observation horizon is correct" holds with probability at least $1 - \delta$. This is the right notion when the correctness of verdicts is imperative.

3.2 Monitor Construction

We modify the deterministic monitor by adding a statistical error term to the uncertainty interval (which affects the register release bound τ^*). The resulting statistical uncertainty interval for the expected discounted sum for time t and observations n is

$$\mathcal{U}_t^{r,s}(\delta; W_{0:n}) \coloneqq \mathcal{I}_t^{r,s}(\delta; W_{0:n}) + \gamma_{t,n}^{r,s} \cdot \Gamma \quad \text{where} \quad \mathcal{I}_t^{r,s}(\delta; W_{0:n}) \coloneqq \mathrm{S}_t^{r,s}(W_{0:n}) \pm \beta_{t,n}^{r,s}(\delta). \tag{7}$$

where $\beta_{t,n}^{r,s}(\delta)$ is a statistical error term and $\gamma_{t,n}^{r,s}$ is a tail error as in Lemma 8, and $\Gamma \coloneqq [\inf \mathcal{R}, \sup \mathcal{R}]$. To obtain soundness it suffices to ensure that $\mathcal{I}_t^{r,s}(\delta; W_{0:n})$ covers the finite expected sum $\mathbb{S}_t^{r,s}(W_{0:n})$ with the desired probability guarantee. Then the deterministic tail term $\gamma_{t,n}^{r,s} \cdot \Gamma$ lifts this to coverage of the infinite expected sum.

Error bounds. For simplicity we focus on data-independent deviation bounds, but the results can be extended to variance adaptive bounds [25]. Hence, the width of the statistical error term is dictated by the squared discount factors, i.e., for every $n \in \mathbb{N}$ and $t \le n$ as

$$\omega_{t,n}^{r,s} \coloneqq \sum_{i=1}^{t} r^{2i} + \sum_{i=0}^{n-t} s^{2i} = \frac{r^2(1 - r^{2t})}{1 - r^2} + \frac{1 - s^{2(n-t+1)}}{1 - s^2},$$

and a uniform upper bound σ on the conditional sub-Gaussian norm σ_t of $X_t - \mathbb{E}_{t-1}(X_t)$, which is trivially given by $d_{\mathcal{R}}/2$, i.e., $\sigma_t \leq \sigma \leq d_{\mathcal{R}}/2$ [39]. We define the pointwise, local, and uniform error bound, respectively, for $\delta \in (0,1)$ as

$$\mathrm{PE}^{r,s}_{t,n}(\delta) := \sqrt{2\,\sigma^2\,\omega^{r,s}_{t,n}\,\log(2/\delta)} \quad \mathrm{LE}^{r,s}_{t,n}(\delta) := k_1\sqrt{V^{r,s}_{t,n}\left(2\log\left(\log_2(V^{r,s}_{t,n})+1\right)+\log\left(\tfrac{2\pi^2}{6\delta}\right)\right)}$$
$$\mathrm{UE}^{r,s}_{t,n}(\delta) := \mathrm{LE}^{r,s}_{t,n}\left(\frac{6\delta}{\pi^2(t+1)^2}\right) \quad \text{where } V^{r,s}_{t,n} := \max(1, \sigma^2\,\omega^{r,s}_{t,n}) \text{ and } k_1 := 2^{1/4} + 2^{-1/4}/\sqrt{2}.$$

The pointwise sound monitor leverages the fixed-sample Hoeffding–Azuma deviation bound [3]. The locally sound monitor leverages the anytime-valid deviation bounds from Howard et al. [25]. The uniformly sound monitor leverages the locally sound statistical error bounds with error level $6\delta/(\pi^2(t+1)^2)$ for time t, because a union bound over all monitored times results in a uniform guarantee, i.e., $\sum_{t=0}^{\infty} 6\delta/(\pi^2(t+1)^2) = \delta$. Unsurprisingly, we can observe that as the guarantees become stronger, the interval width becomes wider.

▶ **Theorem 14.** *Let $(\mathcal{R}, r, s)$ be an input setting, $\mathcal{I} = (L, U)$ a target interval, $\varepsilon \geq 0$, and $\delta \in (0,1)$. Consider Algorithm 1 where a register for time t is released at a fixed (n) or at a flexible ($\mathcal{U}^{r,s}_t(\delta; W_{0:n}) \subseteq \mathcal{I}_{+\varepsilon}$ or $\mathcal{U}^{r,s}_t(\delta; W_{0:n}) \cap \mathcal{I}_{-\varepsilon} = \emptyset$) observation horizon. Then*
(i) *if $\beta^{r,s}_{t,n}(\delta) = \mathrm{PE}^{r,s}_{t,n}(\delta)$ the monitor is pointwise (ε,δ)-approx. sound for fixed release;*
(ii) *if $\beta^{r,s}_{t,n}(\delta) = \mathrm{LE}^{r,s}_{t,n}(\delta)$, the monitor is locally (ε,δ)-approx. sound for flexible release;*
(iii) *if $\beta^{r,s}_{t,n}(\delta) = \mathrm{UE}^{r,s}_{t,n}(\delta)$, the monitor is uniformly (ε,δ)-approx. sound for flexible release.*

Impossibility. If we consider the statistical uncertainty intervals as $n \to \infty$, we observe that the tail error $\gamma^{r,s}_{t,n}$ converges to 0 from above and the statistical error $\beta^{r,s}_{t,n}(\delta)$ converges to a non-zero constant from below. In particular, for the pointwise bound we have

$$\mathrm{PE}^{r,s}_{t,n}(\delta) \to \sqrt{2\sigma^2\left(\tfrac{r^2(1-r^{2t})}{1-r^2} + \tfrac{1}{1-s^2}\right)\log(2/\delta)} \quad \text{and} \quad d_{\mathcal{R}}\,\gamma^{r,s}_{t,n} \to d_{\mathcal{R}}\,\tfrac{r^{t+1}}{1-r} \quad \text{as } n \to \infty.$$

Without additional assumptions this error is unavoidable. Specifically, the pointwise statistical term (the smallest among the soundness notions) is minimax optimal. Intuitively, this is because the variance of the discounted average converges to a positive constant and to 0 only as the discount factors approach 1.

▶ **Theorem 15.** *Given an input setting $(\mathcal{R}, r, s)$ s.t. $\mathcal{R}$ contains an interval of length at least 2σ. Fix $n \in \mathbb{N}$ and $t \leq n$. Consider the class $\mathcal{D}_\sigma$ of product measures on $\mathcal{R}^{n+1}$ s.t. $X_i - \mathbb{E}_{i-1}(X_i)$ is conditionally σ-sub-Gaussian. We define: the set of all $(1-\delta)$ confidence intervals $\mathsf{CI}_\sigma(\delta)$ for $\mathbb{S}^{r,s}_t(W_{0:n})$, i.e., all $I : \mathcal{R}^{n+1} \to \mathrm{Interval}(\mathbb{R})$ satisfying $\sup_{P\in\mathcal{D}_\sigma} \mathbb{P}_{W\sim P}\left(\mathbb{S}^{r,s}_t(W_{0:n}) \notin I(W_{0:n})\right) \leq \delta$, $|I| = \sup I - \inf I$ as interval length, and $\eta^{r,s}_{t,n} = \sum_{i=1}^{t} r^i + \sum_{i=0}^{n-t} s^i$. There exist universal constants $0 < c_0 < C_0 < \infty$ s.t. $\delta \in (0, 1/4)$*

$$c_0 \cdot \min\left(\sigma\eta^{r,s}_{t,n}, \mathrm{PE}^{r,s}_{t,n}(\delta)\right) \leq \inf_{I\in\mathsf{CI}_\sigma(\delta)} \sup_{P\in\mathcal{D}_\sigma} \mathbb{E}_P\left[|I(W_{0:n})|\right] \leq C_0 \cdot \min\left(d_{\mathcal{R}}\eta^{r,s}_{t,n}, \mathrm{PE}^{r,s}_{t,n}(\delta)\right).$$

4 General Discounted Properties

Many quantitative properties involve arithmetic relations between multiple discounted aggregates. A canonical example is (group) fairness, where one compares *acceptance rates* across groups; such rates are quotients of discounted counts of accepted and total events. This section shows how to use the monitors of Section 2 as building blocks for monitoring multi-aggregate expressions.

Setting. Let $\mathcal{N} \coloneqq [n]$, $n \in \mathbb{N}^+$, be the set of event types, and let $(\mathcal{R}, r, s)$ be an input setting. Let $\mathcal{R}_\square \coloneqq \mathcal{R} \cup \{\square\}$ extend $\mathcal{R}$ with a distinguished empty symbol $\square$. An *event* is a vector $\boldsymbol{e} \in \mathcal{E} \coloneqq (\mathcal{R}_\square)^n$, where the k-th component $e^{(k)} \in \mathcal{R}_\square$ is the *atomic event of type* k. We consider bi-infinite event streams $w = (\boldsymbol{e}_t)_{t\in\mathbb{Z}} \in \mathcal{E}^{\mathbb{Z}}$.

4.1 Specification Language

We introduce a simple expression language whose atoms are (bi-directional) discounted sums over individual event types and whose connectives are the standard arithmetic operations. We give two interpretations – *synchronous* and *asynchronous* – differing in how empty atomic events affect discounting.

Syntax. Formulas are built from discounted-sum atoms by scalar multiplication, addition, multiplication, and division:

$$\varphi ::= \mathcal{S}^{(k)}_{r,s} \mid c \cdot \varphi \mid \varphi + \varphi \mid \varphi \cdot \varphi \mid \varphi \div \varphi, \qquad \text{where } k \in \mathcal{N},\ c \in \mathbb{R}. \tag{8}$$

Semantics: atoms. Consider an event stream $w = (\boldsymbol{e}_t)_{t\in\mathbb{Z}}$ and time $t \in \mathbb{Z}$. We map empty atomic events to zero via $x^{(k)}_t \coloneqq \mathbb{1}[e^{(k)}_t \neq \square] \cdot e^{(k)}_t$ for $t \in \mathbb{Z}$, $k \in \mathcal{N}$. The value of $\mathcal{S}^{(k)}_{r,s}$ at time t is a bi-directional discounted sum over the k-typed atomic values, with discount exponents depending on the interpretation:

$$[\![\mathcal{S}^{(k)}_{r,s}]\!]_{sync/async}(w,t) \coloneqq \underbrace{\textstyle\sum_{i=1}^{\infty} r^{\tau^{(k)}_{t,-i}}\, x^{(k)}_{t-i}}_{\text{past}} + \underbrace{x^{(k)}_t}_{\text{present}} + \underbrace{\textstyle\sum_{i=1}^{\infty} s^{\tau^{(k)}_{t,+i}}\, x^{(k)}_{t+i}}_{\text{future}}. \tag{9}$$

The discount exponents $\tau^{(k)}_{t,\pm i}$ are defined as $\tau^{(k)}_{t,\pm i} \coloneqq i$ in the synchronous interpretation, and $\tau^{(k)}_{t,\pm i} \coloneqq \sum_{j=1}^{i} \mathbb{1}[e^{(k)}_{t\pm j} \neq \square]$ in the asynchronous one. Thus, in the synchronous interpretation, every time step advances discounting, even if the k-typed atomic event is empty. In the asynchronous interpretation, only *non-empty* atomic events advance discounting for type k; empty entries are ignored for the purpose of discount progression.

Note that, if $0 \in \mathcal{R}$, any stream $w \in \mathcal{E}^{\mathbb{Z}}$ can be converted to $w' \in \mathcal{E}^{\mathbb{Z}}$ such that $[\![\mathcal{S}^{(k)}_{r,s}]\!]_{sync}(w,t) = [\![\mathcal{S}^{(k)}_{r,s}]\!]_{async}(w',t)$.

Semantics: expressions. The semantics extends naturally to arithmetic operations: for a stream w, time t, and expressions ψ, χ,

$$[\![c \cdot \psi]\!](w,t) = c \cdot [\![\psi]\!](w,t), \qquad [\![\psi \circ \chi]\!](w,t) = [\![\psi]\!](w,t) \circ [\![\chi]\!](w,t) \quad (\circ \in \{+, \cdot, \div\}).$$

Note that division is undefined when the denominator evaluates to 0.

4.2 Uncertainty Propagation Through Interval Arithmetic

Algorithm 1 naturally extends to arbitrary formulas by instantiating a register for each atomic discounted sum and propagating uncertainty intervals via standard interval arithmetic.

Theorem 12 bounds the observation time needed for an ε-approximately sound verdict on a single discounted sum. A natural question is whether an analogous bound holds for arbitrary formulas. The proof of Theorem 12 relies on the fact that, for fixed t, the uncertainty interval of a discounted sum has length $\gamma^{r,s}_{t,t+\tau}(\sup \mathcal{R} - \inf \mathcal{R})$ and depends only on the observation count τ, allowing to identify the earliest τ at which uncertainty falls below 2ε.

The first observation is that this type of reasoning can only be applied for the synchronous interpretation, as in the asynchronous interpretation, a stream can contain arbitrary many empty events for each atom, which make it impossible to give a minimum required observation time. For the synchronous interpretation, we have to develop a theory for different fragments of the whole language. In a nutshell, analogous results to Thm. 12 (1) exist for expressions that contain no divisions, by considering how the uncertainty intervals of sums and products grow through interval arithmetic. This avoids the undefined divisions by 0. When $\mathcal{R}$ is an interval and φ contains only sums and scalar products, the analogous result to Thm. 12 (2) also holds, as monitoring such an expression is equivalent to monitoring an atom on a larger input setting.

We start by defining set arithmetic as usual: for sets S_1, S_2 and operator $\circ \in \{+, -, \cdot, \div\}$, $S_1 \circ S_2 \coloneqq \{s_1 \circ s_2 \;:\; s_1 \in S_1,\ s_2 \in S_2\}$. For a scalar $c \in \mathbb{R}$ and a set $S \subseteq \mathbb{R}$, we use $c \cdot S$ to denote $\{c\} \cdot S$. Notably, for $I_1 = [L_1, U_1]$, and $I_2 = [L_2, U_2]$, and $c \in \mathbb{R}$:

- $c \cdot I_1 = [cL_1, cU_1]$ if $c \geq 0$, $c \cdot I_1 = [cU_1, cL_1]$ if $c < 0$.
- $I_1 + I_2 = [L_1 + L_2, U_1 + U_2]$.
- $I_1 \cdot I_2 = [\min\{L_1 \cdot L_2, L_1 \cdot U_2, U_1 \cdot L_2, U_1 \cdot U_2\}, \max\{L_1 \cdot L_2, L_1 \cdot U_2, U_1 \cdot L_2, U_1 \cdot U_2\}]$.
- If $0 \notin I_2$, then
  $I_1/I_2 = [\min\{L_1/L_2, L_1/U_2, U_1/L_2, U_1/U_2\}, \max\{L_1/L_2, L_1/U_2, U_1/L_2, U_1/U_2\}]$.

▶ **Definition 16** (Linear and multiplicative properties). *Let φ be a general discounted property as defined in Eq. 8. We say that φ is* linear *if it contains only atoms, additions, and scalar multiplications. We say that φ is* multiplicative *if it contains only atoms, additions, scalar multiplication, and products.*

We define the spread of a formula as the range of values it can take at each timestep.

▶ **Definition 17** (Spread). *Let $(\mathcal{R}, r, s)$ be an input setting. The spread of a formula φ is defined recursively as:*

- *If φ is an atom,* $\mathrm{Spread}_{\mathcal{R}}(\varphi) = \mathcal{R}$.
- *If $\varphi = c \cdot \psi$ for $c \in \mathbb{R}$,* $\mathrm{Spread}_{\mathcal{R}}(\varphi) = c \cdot \mathrm{Spread}_{\mathcal{R}}(\psi)$.
- *If $\varphi = \psi \circ \chi$, then* $\mathrm{Spread}_{\mathcal{R}}(\varphi) = \mathrm{Spread}_{\mathcal{R}}(\psi) \circ \mathrm{Spread}_{\mathcal{R}}(\chi)$*, for $\circ \in \circ \in \{+, \cdot, \div\}$.*

If, in any of the divisions, the denominator interval contains 0*, the spread is undefined.*

Note that the spread is always defined for multiplicative properties. For properties containing divisions, it may not be defined.

▶ Remark 18. Because of the interval arithmetic, the expression of $\mathrm{Spread}(\varphi)$ is generally convoluted. A special case is when φ is linear. If φ is linear it can be written as

$$\varphi = \textstyle\sum_{i=1}^{m^-} \alpha_i^- \cdot x_i^- + \sum_{i=1}^{m^+} \alpha_i^+ \cdot x_i^+.$$

Therefore, for $\mathcal{R} = [\inf \mathcal{R}, \sup \mathcal{R}]$, its spread can be written as

$$\mathrm{Spread}_{\mathcal{R}}(\varphi) = \left[\sup \mathcal{R} \cdot \textstyle\sum_{i=1}^{m^-} \alpha_i^- + \inf \mathcal{R} \cdot \sum_{i=1}^{m^+} \alpha_i^+, \inf \mathcal{R} \cdot \sum_{i=1}^{m^-} \alpha_i^- + \sup \mathcal{R} \cdot \sum_{i=1}^{m^+} \alpha_i^+\right].$$

Using synchronous semantics, monitoring a linear expression φ on an input setting $(\mathcal{R}, r, s)$ is equivalent to monitoring an atom on the input setting $(\mathcal{R}_\varphi, r, s)$ where $\mathcal{R}_\varphi = \mathrm{Spread}_{\mathcal{R} \cup \{0\}}(\varphi)$. Therefore, we have the following corollary to Theorem 12.

▶ **Corollary 19.** *Let $(\mathcal{R}, r, s)$ be an input setting, $\mathcal{I}$ a target interval, $\varepsilon > 0$, $T \in \mathbb{N}$, and φ a linear formula. Let $\mathcal{R}_\varphi = \mathrm{Spread}_{\mathcal{R} \cup \{0\}}(\varphi)$. With*

$$\tau^*(\mathcal{R}, r, s, \varepsilon, T, \varphi) \coloneqq \min\left(\left\{\tau \in \mathbb{N} : d_{\mathcal{R}_\varphi}\left(\frac{r^{T+1}}{1-r} + \frac{s^{\tau+1}}{1-s}\right) \leq 2\varepsilon\right\} \cup \{\infty\}\right). \tag{10}$$

1. *If $\tau^* < \infty$, there exists an ε-approximately sound monitor that produces a verdict for every $t \geq T$ within τ^* steps.*
2. *Suppose that $\mathcal{R}_\varphi = [m, M]$ with $m < M$, and write $J = \lambda^{r,s}[m, M]$. If $J \cap \mathcal{I}_{-\varepsilon} \neq \emptyset$ and $J \setminus \mathcal{I}_{+\varepsilon} \neq \emptyset$, then, for every ε-approximately sound monitor of φ and every $\tau < \tau^*$, there exists $w \in \mathcal{R}^{\mathbb{Z}}$ on which the monitor does not produce a verdict for time T after τ steps.*

▶ Remark 20. Corollary 19 cannot be extended expressions with the asynchronous interpretation. If φ is interpreted asynchronously, then it can contain arbitrary many empty events for each atom, which make it impossible to give a minimum required observation time.

5 Register Complexity of Discounted-Sum Monitoring

The quantity τ plays two roles in Algorithm 1: it bounds both the *verdict delay*, i.e., the number of steps required to resolve the uncertainty interval for a position, and the *register count*, i.e., the number of running sums maintained in parallel. These roles have different origins. The delay is inherent to discounted sums: contributions from unobserved values decay geometrically, and τ is the first horizon where this is resolved for sure. The register count, by contrast, is an implementation cost. Since the running sums are related, it is natural to ask whether we can track the τ pending positions with less memory.

We show that no such reduction is possible in the worst case in the future-only setting. To make the question precise, we formalize monitors as *affine register machines* (ARMs): finite-state machines equipped with real-valued registers, affine updates, and strict affine guards [28, 2, 18]. By Theorem 12, every monitored position admits an ε-approximately sound verdict by its horizon, so we encode these deadline verdicts as a safety language over $\mathcal{R}^\omega$. We build, for every $\tau \geq 1$, a monitoring instance with horizon τ whose language cannot be recognized by any ARM with fewer than τ registers.

Safety Formulation of Approximate Monitoring. We first recast approximate monitoring as a safety-language recognition problem. Throughout this section we work over the normalized input domain $\mathcal{R} = [0, 1]$. This is without loss of generality for interval domains: any bounded interval $[m, M]$ can be mapped to $[0, 1]$ by an affine transformation, and discounted sums, target intervals, and tolerances rescale accordingly.

A *monitoring instance* is a tuple $\mathcal{P} = (\mathcal{R}, r, s, \mathcal{I}, \varepsilon)$, where $(\mathcal{R}, r, s)$ is a proper input setting with $\mathcal{R} = [0, 1]$ and $r, s \in [0, 1)$, where $\mathcal{I} = (L, U)$ is a target interval and $\varepsilon > 0$. The *horizon* of $\mathcal{P}$ is $\tau(\mathcal{P}) := \tau^*(\mathcal{R}, r, s, \varepsilon, 0)$, as in Theorem 12. We say that $\mathcal{P}$ is *future only* if $r = 0$ and $s > 0$, and *past only* if $r > 0$ and $s = 0$.

▶ **Definition 21** (Monitoring Language). *The* language *of a monitoring instance $\mathcal{P} = (\mathcal{R}, r, s, \mathcal{I}, \varepsilon)$ is $L(\mathcal{P}) := \{x \in \mathcal{R}^\omega \mid \forall t \geq 0 : U_t^{r,s}(x_{0:t+\tau(\mathcal{P})}) \subseteq \mathcal{I}_{+\varepsilon}\}$.*

By Lemma 8 and Theorem 12, the uncertainty interval at time $t + \tau(\mathcal{P})$ has diameter at most 2ε. Hence, if it is not contained in $\mathcal{I}_{+\varepsilon}$, then it is disjoint from $\mathcal{I}_{-\varepsilon}$. Therefore, the rule that returns $\top$ exactly when $U_t^{r,s}(x_{0:t+\tau(\mathcal{P})}) \subseteq \mathcal{I}_{+\varepsilon}$ and returns $\bot$ otherwise is always defined and ε-approximately sound.

Since non-membership in $L(\mathcal{P})$ is witnessed by a finite prefix, $L(\mathcal{P})$ is a safety language. Indeed, $U_t^{r,s}(x_{0:t+\tau(\mathcal{P})})$ depends only on $x_{0:t+\tau(\mathcal{P})}$, so any violation at time t is shared by every extension of this prefix. By Lemma 8 and since $\mathcal{R} = [0, 1]$, the condition $U_t^{r,s}(x_{0:t+\tau(\mathcal{P})}) \subseteq \mathcal{I}_{+\varepsilon}$ is equivalent to $L - \varepsilon < S_t^{r,s}(x_{0:t+\tau(\mathcal{P})}) < U + \varepsilon - \gamma^{r,s}_{t,t+\tau(\mathcal{P})}$.

Affine Register Machines. Let $Y = \{y_1, \dots, y_k\}$ be a finite set of registers. A *valuation* is a map $\nu : Y \to \mathbb{R}$. An *update* over Y is a parallel assignment $y_i \leftarrow \sum_{j=1}^{k} a_{i,j} y_j + b_i\varsigma + c_i$ for all $1 \le i \le k$, where $\varsigma \in \mathcal{R}$ is the current input and all coefficients are real. We write $\Gamma(Y)$ for the set of such updates. A *guard* over Y is a finite conjunction of strict affine inequalities $\sum_{j=1}^{k} a_j y_j + b\varsigma \bowtie c$, where $\bowtie \in \{<, >\}$. We write $(\nu, \varsigma) \models \varphi$ if the guard φ holds under valuation ν and input ς, and we write $\Phi(Y)$ for the set of guards over Y.

▶ **Definition 22** (Affine Register Machine). *A (deterministic)* affine register machine *(ARM) is a tuple $\mathcal{M} = (\mathcal{R}, Y, Q, q_0, \nu_0, \Delta)$ where $\mathcal{R} \subset \mathbb{R}$ is a bounded input set, Y is a finite set of registers, Q is a finite set of control locations, $q_0 \in Q$ is the initial location, $\nu_0 \colon Y \to \mathbb{R}$ is the initial valuation, and $\Delta \subseteq Q \times \Phi(Y) \times \Gamma(Y) \times Q$ is a finite transition relation. We write transitions as $(q, \varphi, \gamma, q') \in \Delta$ and require the following determinism condition: for every location $q \in Q$, every valuation ν, and every input $\varsigma \in \mathcal{R}$, there is at most one transition $(q, \varphi, \gamma, q') \in \Delta$ such that $(\nu, \varsigma) \models \varphi$.*

A configuration is a pair (q, ν). On input ς, the machine follows the unique enabled transition (q, φ, γ, q') to $(q', \gamma(\nu, \varsigma))$. If no transition is enabled, the run terminates. A run on $x \in \mathcal{R}^\omega$ is *accepting* if infinite. The language $L(\mathcal{M})$ is the set of streams with an accepting run; hence every ARM recognizes a safety language.

Sufficiency: An ARM Construction. We show that $L(\mathcal{P})$ can be recognized by an ARM with $\tau(\mathcal{P}) + 1$ registers. The construction follows the monitor from Algorithm 1. The ARM keeps one register for each pending position and updates all pending truncated sums in parallel. If a register y_a stores the truncated sum for the pending position of age a, then on input x the affine updates are $y_0 \leftarrow r y_0 + x$ and $y_a \leftarrow y_{a-1} + s^a x$ for all $1 \le a \le \tau(\mathcal{P}) - 1$. When a position reaches age $\tau(\mathcal{P})$, its completed truncated sum is $y_{\tau(\mathcal{P})-1} + s^{\tau(\mathcal{P})} x$, which the machine checks by a guard before applying these updates. One additional register tracks the geometric factor needed to express the time-dependent past residual $r^{t+1}/(1 - r)$.

▶ **Theorem 23** (General Upper Bound). *For every monitoring instance $\mathcal{P}$, there is an ARM $\mathcal{M}$ with $\tau(\mathcal{P}) + 1$ registers such that $L(\mathcal{M}) = L(\mathcal{P})$.*

In the future-only case the past residual is zero. The auxiliary register is therefore unnecessary, and the same construction uses exactly one register per pending future window.

▶ **Corollary 24** (Future-Only Upper Bound). *For every future-only monitoring instance $\mathcal{P}$, there is an ARM $\mathcal{M}$ with $\tau(\mathcal{P})$ registers such that $L(\mathcal{M}) = L(\mathcal{P})$.*

Necessity: A Register Lower Bound. We now show that the future-only upper bound is optimal in the worst case. For each $k \ge 1$, consider the future-only monitoring instance $\mathcal{P}_k := ([0, 1],\ 0,\ \frac{1}{2},\ (0, 1),\ 2^{-(k+1)})$. For $x \in [0, 1]^\omega$ and $t \ge 0$, write $V_t^k(x) := \sum_{i=0}^{k} 2^{-i} x_{t+i}$ and $T_k := 1 - 2^{-(k+1)}$. For this family, the safety condition has a particularly simple form. The horizon is k, the residual future uncertainty is 2^{-k}, and the lower bound is vacuous because all inputs are nonnegative. Thus, the monitoring language is exactly the set of streams whose discounted windows of length $k + 1$ remain below the threshold T_k.

▶ **Lemma 25** (Characterization of the hard instances). *For every $k \ge 1$, the future-only monitoring instance $\mathcal{P}_k$ has horizon $\tau(\mathcal{P}_k) = k$, and its monitoring language is $L(\mathcal{P}_k) = \{x \in [0, 1]^\omega \mid \forall t \ge 0 : \sum_{i=0}^{k} 2^{-i} x_{t+i} < 1 - 2^{-(k+1)}\}$.*

The lower-bound proof is a finite-dimensional linear-algebra argument. Suppose an ARM with only $k - 1$ registers recognizes the language above. Consider the first k transitions taken on a suitable accepted anchor word. On all length-k prefixes inducing this same transition sequence, the register valuation after k steps is an affine map from a k-dimensional prefix space to $\mathbb{R}^{k-1}$. Therefore, this map has a nonzero kernel direction: a perturbation that is invisible to the registers after the first k inputs.

The separating prefixes are chosen from an explicit one-parameter family. For θ just below T_k, the word $u(\theta) \cdot \alpha \cdot 0^\omega$ is constructed so that its first k discounted windows all have value θ. Moving slightly in the invisible kernel direction preserves the first k transitions, the register valuation after k steps, and the entire suffix. However, by choosing the last nonzero coordinate of the kernel direction, one discounted window is pushed above T_k. Determinism then forces the ARM to treat two identical configurations with the same suffix identically, although one stream satisfies the monitoring condition and the other violates it. Note that $\mathcal{R}$ being an interval is crucial here, as it ensures that the kernel direction (an arbitrary vector in $\mathbb{R}^k$) can be realized as a small perturbation inside $\mathcal{R}^k$.

▶ **Theorem 26** (Future-Only Lower Bound). *For every $k \geq 1$, there is a future-only monitoring instance $\mathcal{P}_k$ such that for every ARM $\mathcal{M}$ with $\tau(\mathcal{P}_k) - 1$ registers we have $L(\mathcal{M}) \neq L(\mathcal{P}_k)$.*

Special Cases. For the past-only case ($s = 0$ and $r > 0$), it suffices to use just two registers: one for the observed past, which follows the rule $p_t = x_t + rp_{t-1}$, and one for the uncertainty term $r^{t+1}/(1-r)$. The register complexity for a general $s > 0$ and $r > 0$ remains open.

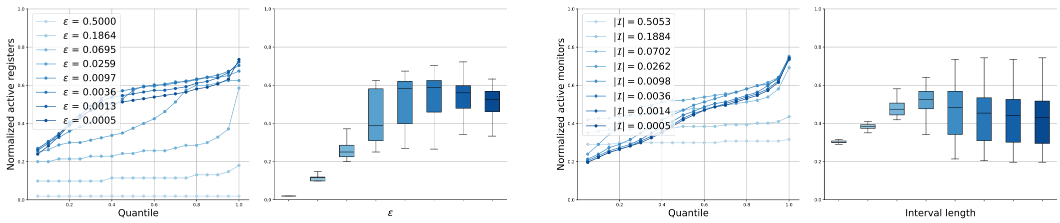
Over finite alphabets, the future-only lower bound disappears. For $r = 0$, monitoring reduces to checking a sliding window $x_t, \ldots, x_{t+\tau}$, so a finite-state monitor suffices, at the cost of exponential state space in τ. With past discounting, however, this is no longer possible: although a finite alphabet collapses the τ parallel future registers into finite control, the recurrence $p_t = x_t + rp_{t-1}$ still can take infinitely many values, which may require an infinite-state monitor.

6 Experiments

We evaluate our monitors on a collection of real-world and synthetic scenarios with the objective of demonstrating the practical effectiveness of the proposed monitors and to examine how tight the theoretical bounds on resource usage and accuracy are in practice. In the main paper our evaluation is guided by the research questions: **(RQ1)** How closely does the observed register usage of Algorithm 1 match the theoretical bound of Thm. 12?, and **(RQ2)** How does register usage vary under synchronous and asynchronous semantics? We address research questions beyond (RQ1) and (RQ2) in the full version.

We evaluate our monitors on traces from two benchmarks: `PowerData`, a dataset of power usage traces from Google data centers [34]; and `Adult`, demographic parity traces produced by a fairness-aware neural network trained on the Adult dataset [7].

Register Usage on `PowerData`. To address RQ1, we study register usage on traces from `PowerData`. We report *normalized register usage*, defined as the number of active registers divided by the theoretical upper bound τ^* from Thm. 12. The input range is $\mathcal{R} = [0, 1]$ and we fix the discount factors to $r = s = 0.9$. In Fig. 1 (a), we vary the tolerance ε logarithmically between 0.5 and $5 \cdot 10^{-4}$, while fixing the target interval to $\mathcal{I} = [0.703, 0.728]$, centered at the empirical mean with width equal to one standard deviation of the dataset. In Fig. 1 (b), we fix $\varepsilon = 5 \cdot 10^{-4}$ and vary the width of $\mathcal{I}$ between 10 and 0.01 times the standard deviation, again centered at the mean. In all experiments, the initial monitoring time T is chosen such that $r^{T+1}/(1-r)$ is closest to $\varepsilon/(\sup \mathcal{R} - \inf \mathcal{R})$. In both Fig. 1 (a) and (b), the left graph represents the cumulative distribution of normalized active monitors,



(a) Variable monitoring tolerance factor ε.

(b) Variable width of the target interval $|\mathcal{I}|$.

Figure 1 Register usage in terms of normalized active registers, `PowerData` dataset.

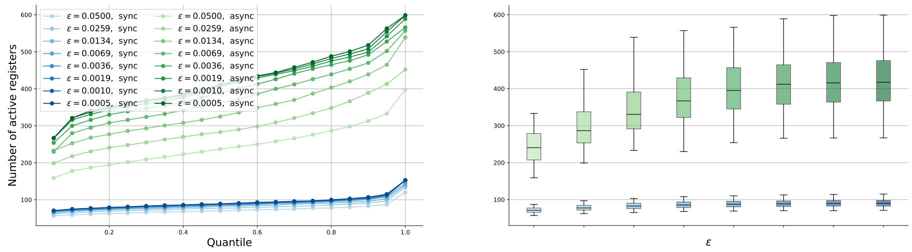


Figure 2 Active registers when demographic parity, variable ε. `Adult` dataset.

with one line representing each parameter setting – ε in (a), $|\mathcal{I}|$ in (b) –, while the right graph represents each distribution as a boxplot. Each boxplot corresponds to a parameter setting, color matched with the corresponding left graph.

Across all configurations, most verdicts are produced with fewer than 60% of the theoretically available registers active, and register usage never exceeds 80%. This result indicates that we could actually implement a monitor with, for example, 70% of the theoretically required registers, and still output a sound verdict most of the time.

Monitoring Demographic Parity. To investigate RQ2, we monitor demographic parity on the `Adult` dataset. The input set is $\mathcal{R} = \{0, 1\}$ and we again fix $r = s = 0.95$. We consider decreasing tolerance values with a fixed target interval $\mathcal{I} = [-0.1, 0.1]$. We report the results in Fig. 2, in the same format as the previous figure, for the demographic parity property (Ex. 3) with the synchronous (blue) and asynchronous (green) interpretation. In contrast to the previous experiments, we report absolute register counts rather than normalized usage. This is because Thm. 12 does not provide a uniform bound in this setting, as the denominators can be arbitrarily close to zero. As expected, synchronous interpretations are significantly more efficient than asynchronous ones, since discounting is applied uniformly across streams rather than independently.

7 Related Work

Discounted-sum objectives are well studied in verification, automata, games, and logic as quantitative specifications where recent events outweigh distant ones [11, 12, 8, 37, 19, 1]. The nontrivial structure of achievable values underlies the difficulty of the target discounted-sum problem [9], motivating approximation: nondeterministic discounted-sum automata cannot be determinized exactly, but approximate determinization is always possible [8] and approximate inclusion is decidable [4]. We study online monitoring under bounded memory and show a parallel dichotomy: exact sound monitoring requires unbounded memory, whereas approximate monitoring admits tight bounds.

Runtime verification provides lightweight formal guarantees by checking specifications against executions online [5]. Quantitative monitoring replaces boolean verdicts with real-valued measures; STL robustness semantics, for instance, quantify how strongly a signal satisfies a specification [31, 17, 15, 14, 13]. Recent work [24] establishes precision-cost tradeoffs for quantitative monitoring. We focus on discounted-sum monitoring and extend the framework to stochastic processes via confidence sequences.

Discounted sums can be viewed as convolution with an exponential kernel: the exponentially weighted moving average (EWMA) computes a normalized discounted sum [33], equivalent to a first-order IIR filter [36]. EWMA control charts detect mean shifts under i.i.d. inputs [30, 32], and exponential smoothing in forecasting balances responsiveness and noise reduction [20, 26]. We complement these by studying memory requirements and approximation guarantees for arbitrary input streams.

Standard fairness metrics such as demographic parity and equalized odds [22, 16] are static population statistics. Existing fairness monitors use only cumulative statistics [23] or stream-based specifications [6]. Recent work highlights temporal aspects of fairness [21]: discounting appears in fair reinforcement learning [27], decision making [38], and resource allocation [29], but targets policy design rather than monitoring with formal guarantees.

References

1 Shaull Almagor, Udi Boker, and Orna Kupferman. Discounting in ltl. In *International conference on tools and algorithms for the construction and analysis of systems*, pages 424–439. Springer, 2014. doi:10.1007/978-3-642-54862-8_37.

2 Rajeev Alur, Loris D'Antoni, Jyotirmoy V. Deshmukh, Mukund Raghothaman, and Yifei Yuan. Regular functions and cost register automata. In *28th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2013, New Orleans, LA, USA, June 25-28, 2013*, pages 13–22. IEEE Computer Society, 2013. doi:10.1109/LICS.2013.65.

3 Kazuoki Azuma. Weighted sums of certain dependent random variables. *Tohoku Mathematical Journal, Second Series*, 19(3):357–367, 1967.

4 Suguman Bansal, Swarat Chaudhuri, and Moshe Y Vardi. Comparator automata in quantitative verification. *Logical Methods in Computer Science*, 18, 2022. doi:10.46298/LMCS-18(3:13)2022.

5 Ezio Bartocci, Yliès Falcone, Adrian Francalanza, and Giles Reger. Introduction to runtime verification. In *Lectures on Runtime Verification: Introductory and Advanced Topics*, pages 1–33. Springer, 2018. doi:10.1007/978-3-319-75632-5_1.

6 Jan Baumeister, Bernd Finkbeiner, Frederik Scheerer, Julian Siber, and Tobias Wagenpfeil. Stream-based monitoring of algorithmic fairness. In *International Conference on Tools and Algorithms for the Construction and Analysis of Systems*, pages 60–81. Springer, 2025. doi:10.1007/978-3-031-90643-5_4.

7 Barry Becker and Ronny Kohavi. Adult. UCI Machine Learning Repository, 1996. doi:10.24432/C5XW20.

8 Udi Boker and Thomas A Henzinger. Approximate determinization of quantitative automata. *Leibniz International Proceedings in Informatics*, 18, 2012. doi:10.4230/LIPIcs.FSTTCS.2012.362.

9 Udi Boker, Thomas A Henzinger, and Jan Otop. The target discounted-sum problem. In *2015 30th Annual ACM/IEEE Symposium on Logic in Computer Science*, pages 750–761. IEEE, 2015. doi:10.1109/LICS.2015.74.

10 Filip Cano, Thomas A. Henzinger, Konstantin Kueffner, and N. Ege Saraç. filipcano/monitoring-discounted-sum-properties. Software, swhId: swh:1:dir:cb99396415f279c985cd05206b44320227f64245 (visited on 2026-08-14).

URL: https://github.com/filipcano/monitoring-discounted-sum-properties, doi:10.4230/artifacts.27673.

11 Krishnendu Chatterjee, Laurent Doyen, and Thomas A Henzinger. Quantitative languages. *ACM Transactions on Computational Logic (TOCL)*, 11(4):1–38, 2010. doi:10.1145/1805950.1805953.

12 Luca De Alfaro, Thomas A Henzinger, and Rupak Majumdar. Discounting the future in systems theory. In *International Colloquium on Automata, Languages, and Programming*, pages 1022–1037. Springer, 2003. doi:10.1007/3-540-45061-0_79.

13 Jyotirmoy V Deshmukh, Alexandre Donzé, Shromona Ghosh, Xiaoqing Jin, Garvit Juniwal, and Sanjit A Seshia. Robust online monitoring of signal temporal logic. *Formal Methods in System Design*, 51(1):5–30, 2017. doi:10.1007/S10703-017-0286-7.

14 Alexandre Donzé, Thomas Ferrere, and Oded Maler. Efficient robust monitoring for stl. In *International conference on computer aided verification*, pages 264–279. Springer, 2013.

15 Alexandre Donzé and Oded Maler. Robust satisfaction of temporal logic over real-valued signals. In *International conference on formal modeling and analysis of timed systems*, pages 92–106. Springer, 2010. doi:10.1007/978-3-642-15297-9_9.

16 Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard Zemel. Fairness through awareness. In *Proceedings of the 3rd innovations in theoretical computer science conference*, pages 214–226, 2012. doi:10.1145/2090236.2090255.

17 Georgios E Fainekos and George J Pappas. Robustness of temporal logic specifications for continuous-time signals. *Theoretical Computer Science*, 410(42):4262–4291, 2009. doi:10.1016/J.TCS.2009.06.021.

18 Thomas Ferrère, Thomas A. Henzinger, and N. Ege Saraç. A theory of register monitors. In Anuj Dawar and Erich Grädel, editors, *Proceedings of the 33rd Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2018, Oxford, UK, July 09-12, 2018*, pages 394–403. ACM, 2018. doi:10.1145/3209108.3209194.

19 Jerzy Filar and Koos Vrieze. *Competitive Markov decision processes.* Springer Science & Business Media, 2012.

20 Everette S Gardner Jr. Exponential smoothing: The state of the art – Part II. *International journal of forecasting*, 22(4):637–666, 2006.

21 Usman Gohar, Zeyu Tang, Jialu Wang, Kun Zhang, Peter Spirtes, Yang Liu, and Lu Cheng. Long-term fairness inquiries and pursuits in machine learning: A survey of notions, methods, and challenges. *Trans. Mach. Learn. Res.*, 2025. URL: https://openreview.net/forum?id=mYi6EWvFlR.

22 Moritz Hardt, Eric Price, and Nati Srebro. Equality of opportunity in supervised learning. *Advances in neural information processing systems*, 29, 2016.

23 Thomas A Henzinger, Mahyar Karimi, Konstantin Kueffner, and Kaushik Mallik. Monitoring algorithmic fairness. In *International Conference on Computer Aided Verification*, pages 358–382. Springer, 2023. doi:10.1007/978-3-031-37703-7_17.

24 Thomas A Henzinger and N Ege Saraç. Quantitative and approximate monitoring. In *2021 36th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*, pages 1–14. IEEE, 2021. doi:10.1109/LICS52264.2021.9470547.

25 Steven R Howard, Aaditya Ramdas, Jon McAuliffe, and Jasjeet Sekhon. Time-uniform, nonparametric, nonasymptotic confidence sequences. *The Annals of Statistics*, 49(2):1055–1080, 2021.

26 Rob J Hyndman and Yeasmin Khandakar. Automatic time series forecasting: The forecast package for R. *Journal of statistical software*, 27:1–22, 2008.

27 Shahin Jabbari, Matthew Joseph, Michael Kearns, Jamie Morgenstern, and Aaron Roth. Fairness in reinforcement learning. In *International conference on machine learning*, pages 1617–1626. PMLR, 2017. URL: http://proceedings.mlr.press/v70/jabbari17a.html.

28 Michael Kaminski and Nissim Francez. Finite-memory automata. *Theor. Comput. Sci.*, 134(2):329–363, 1994. doi:10.1016/0304-3975(94)90242-9.

29 Ashwin Kumar and William Yeoh. Remember, but also, forget: Bridging myopic and perfect recall fairness with past-discounting. *arXiv preprint arXiv:2504.01154*, 2025. doi:10.48550/arXiv.2504.01154.
30 James M Lucas and Michael S Saccucci. Exponentially weighted moving average control schemes: properties and enhancements. *Technometrics*, 32(1):1–12, 1990.
31 Oded Maler and Dejan Nickovic. Monitoring temporal properties of continuous signals. In *International symposium on formal techniques in real-time and fault-tolerant systems*, pages 152–166. Springer, 2004. doi:10.1007/978-3-540-30206-3_12.
32 Douglas C Montgomery. *Introduction to statistical quality control*. John wiley & sons, 2020.
33 Stuart W Roberts. Control chart tests based on geometric moving averages. *Technometrics*, 42(1):97–101, 2000. doi:10.1080/00401706.2000.10485986.
34 Varun Sakalkar, Vasileios Kontorinis, David Landhuis, Shaohong Li, Darren De Ronde, Thomas Blooming, Anand Ramesh, James Kennedy, Christopher Malone, Jimmy Clidaras, and Parthasarathy Ranganathan. Data center power oversubscription with a medium voltage power plane and priority-aware capping. In *Proceedings of the Twenty-Fifth International Conference on Architectural Support for Programming Languages and Operating Systems*, pages 497–511, 2020. Data available: https://github.com/google/cluster-data/blob/master/PowerData2019.md. doi:10.1145/3373376.3378533.
35 César Sánchez, Gerardo Schneider, Wolfgang Ahrendt, Ezio Bartocci, Domenico Bianculli, Christian Colombo, Yliès Falcone, Adrian Francalanza, Srđan Krstić, João M Lourenço, et al. A survey of challenges for runtime verification from advanced application domains (beyond software). *Formal Methods in System Design*, 54(3):279–335, 2019. doi:10.1007/S10703-019-00337-W.
36 Ronald William Schafer and Alan V Oppenheim. *Discrete-time signal processing*, volume 5. Prentice Hall Englewood Cliffs, New Jersey, 1989.
37 Lloyd S Shapley. Stochastic games. *Proceedings of the national academy of sciences*, 39(10):1095–1100, 1953.
38 Manuel R. Torres, Parisa Zehtabi, Michael Cashmore, Daniele Magazzeni, and Manuela Veloso. Temporal fairness in decision making problems. In *European Conference of Artificial Intelligence (ECAI)*, Frontiers in Artificial Intelligence and Applications, pages 1132–1139. IOS Press, 2024. doi:10.3233/FAIA240606.
39 Roman Vershynin. *High-dimensional probability: An introduction with applications in data science*, volume 47. Cambridge university press, 2018.