# Quantitative Language Automata


**Thomas A. Henzinger**
Institute of Science and Technology Austria (ISTA), Klosterneuburg, Austria

**Pavol Kebis**
Institute of Science and Technology Austria (ISTA), Klosterneuburg, Austria

**Nicolas Mazzocchi**
Slovak University of Technology in Bratislava, Slovak Republic

**N. Ege Saraç**
Institute of Science and Technology Austria (ISTA), Klosterneuburg, Austria



## Abstract

A *quantitative word automaton* (QWA) defines a function from infinite words to values. For example, every infinite run of a limit-average QWA $\mathcal{A}$ obtains a mean payoff, and every word $w \in \Sigma^\omega$ is assigned the maximal mean payoff obtained by nondeterministic runs of $\mathcal{A}$ over $w$. We introduce *quantitative language automata* (QLAs) that define functions from language generators (i.e., implementations) to values, where a language generator can be nonprobabilistic, defining a set of infinite words, or probabilistic, defining a probability measure over infinite words. A QLA consists of a QWA and an aggregator function. For example, given a QWA $\mathcal{A}$, the infimum aggregator maps each language $L \subseteq \Sigma^\omega$ to the greatest lower bound assigned by $\mathcal{A}$ to any word in $L$. For boolean value sets, QWAs define boolean properties of traces, and QLAs define boolean properties of sets of traces, i.e., hyperproperties. For more general value sets, QLAs serve as a specification language for a generalization of hyperproperties, called *quantitative hyperproperties*. A nonprobabilistic (resp. probabilistic) quantitative hyperproperty assigns a value to each set (resp. distribution) $G$ of traces, e.g., the minimal (resp. expected) average response-time exhibited by the traces in $G$. We give several examples of quantitative hyperproperties and investigate three paradigmatic problems for QLAs: evaluation, nonemptiness, and universality. In the *evaluation* problem, given a QLA $\mathbb{A}$ and an implementation $G$, we ask for the value that $\mathbb{A}$ assigns to $G$. In the *nonemptiness* (resp. *universality*) problem, given a QLA $\mathbb{A}$ and a value $k$, we ask whether $\mathbb{A}$ assigns at least $k$ to some (resp. every) language. We provide a comprehensive picture of decidability for these problems for QLAs with common aggregators as well as their restrictions to $\omega$-regular languages and trace distributions generated by finite-state Markov chains.



**2012 ACM Subject Classification** Theory of computation → Automata over infinite objects; Theory of computation → Quantitative automata

**Keywords and phrases** Quantitative hyperproperties, quantitative automata, automata-based verification.

**Funding** This work was supported in part by the ERC-2020-AdG 101020093.


## 1 Introduction

The specification and verification of system properties traditionally take a boolean view. While this view is appropriate for correctness properties, it lacks the ability to reason about quantitative aspects of system behaviors, such as performance or robustness. Quantitative trace properties and quantitative word automata [15] address this gap: instead of partitioning the set of traces into correct and incorrect traces (as boolean properties do), they define functions from system executions to richer value domains, e.g., the real numbers. Using such a formalism, we can specify the maximal or average response time of a server's execution, or how badly a system behavior violates a desired boolean property [23].

Many interesting system properties lie beyond the trace setting; especially security properties often refer to multiple traces. In the boolean case, they can be specified by hyperproperties [16], which are interpreted over sets of traces rather than over individual traces. For example, a prominent hyperproperty is observational determinism, which requires every pair of executions with matching observable inputs to have matching observable outputs. In general, while trace properties specify which system behaviors are correct, hyperproperties specify which system implementations are correct. In a similar vein, while quantitative trace properties and quantitative word automata describe system properties on the level of individual executions, *quantitative hyperproperties* are needed to express quantitative aspects of system-wide properties. For example, a quantitative hyperproperty may measure the maximal or average response time of an implementation instead of a single execution, or how badly an implementation violates a desired boolean property. In this paper, we introduce *quantitative language automata* (QLAs), an automaton model for the specification and verification of quantitative hyperproperties. In contrast to quantitative word automata, quantitative language automata can measure system-wide properties.

Quantitative word automata (QWAs) extend boolean $\omega$-automata with weighted transitions. An infinite run yields an infinite sequence of weights, which are accumulated by a *run aggregator* (a.k.a. value function). Common run aggregators are $\mathsf{Sup}$ (the maximal weight along an infinite run), $\mathsf{LimSup}$ (the largest weight that occurs infinitely often), or $\mathsf{LimInfAvg}$ (the long-term average of an infinite sequence of weights). When a given infinite word yields more than one run, as is generally the case for nondeterministic automaton specifications, the possible run values are accumulated by a *word aggregator*. The most common word aggregator is $\mathsf{Sup}$ (the least upper bound of all values that can be achieved by resolving the nondeterministic choices), which generalizes the standard view that a single accepting run suffices to accept a word, but other word aggregators are possible. For example, the word aggregator $\mathsf{LimSup}$ assigns to each infinite word $w$ the l.u.b. of the values realized by infinitely many runs over $w$. When the specification is probabilistic (rather than nondeterministic), the word aggregator $\mathbb{E}$ assigns to $w$ an expected value for the automaton reading $w$.

Quantitative language automata extend quantitative word automata with a third kind of aggregator function, called *language aggregator*, which summarizes the values of all infinite words which are obtained from an implementation and defined by a so-called "language generator." A language generator can be nonprobabilistic, defining a set of infinite words, or probabilistic, defining a probability measure over infinite words. A QLA $\mathbb{A} = (h, \mathcal{A})$ consists of a language aggregator $h$ and a QWA $\mathcal{A}$, and maps each language generator $G$ to the value $h_{w\sim G}\mathcal{A}(w)$, where $\mathcal{A}(w)$ is the value assigned by the QWA $\mathcal{A}$ to the infinite word $w$, and $h_{w\sim G}$ denotes the accumulation over all words generated by $G$. When $G$ is nonprobabilistic, we interpret the aggregator $h$ over the words that belong to the language defined by $G$; when probabilistic, over the words generated with respect to the probability measure defined by $G$. Using the language aggregators $\mathsf{Inf}$ and $\mathsf{Sup}$, language automata can measure the best-case and the worst-case values that can be obtained by all executions of an implementation. With language aggregators $\mathsf{LimInf}$ and $\mathsf{LimSup}$, we can express the "almost" best- and worst-cases, considering only the values achieved by infinitely many words in the language. Finally, when $G$ is probabilistic, the language aggregator $\mathbb{E}$ captures the average-case with respect to the its probability measure. Using all three aggregator functions, we can specify, for example, the expected value (taken over all possible implementation traces) of the best case (realizable by a nondeterministic specification) of the average weight along a run of the specification.

The standard decision problems for automata extend naturally to our framework. Consider a language automaton $\mathbb{A}$ and a rational threshold $k$. Given a finite-state language generator $G$ (i.e., an $\omega$-regular automaton or a finite Markov chain), the evaluation problem asks to compute the value to which $\mathbb{A}$ maps $G$. The nonemptiness (resp. universality) problem asks whether $\mathbb{A}$ maps some (resp. every) language generator to a value of at least $k$, or strictly

| **(a)** | Evaluation | | | | | Nonemptiness | | |
|---|---|---|---|---|---|---|---|---|
| | $h = g$ | $h \neq g$ | $h = \mathbb{E}$ | $h \in \{\mathsf{Inf}, \mathsf{Sup}\}$ | $h = \mathbb{E}$ | $h \in \{\mathsf{Inf}, \mathsf{Sup}, \mathbb{E}\}$ | | |
| | $\in \{\mathsf{Inf}, \mathsf{Sup}\}$ | $\in \{\mathsf{Inf}, \mathsf{Sup}\}$ | $g \in \{\mathsf{Inf}, \mathsf{Sup}\}$ | $g = \mathbb{E}$ | $g = \mathbb{E}$ | $g = \mathsf{Sup}$ | $g = \mathsf{Inf}$ | $g = \mathbb{E}$ |
| $f \in \left\{ \begin{array}{c} \mathsf{Inf}, \\ \mathsf{Sup}, \\ \mathsf{LimInf}, \\ \mathsf{LimSup} \end{array} \right\}$ | PTime | PSpace | ExpTime | Undecidable | PTime | PTime | PSpace | Undecidable |
| $f \in \left\{ \begin{array}{c} \mathsf{LimInfAvg}, \\ \mathsf{LimSupAvg} \end{array} \right\}$ | PTime | Undecidable | Undecidable | Undecidable | PTime | PTime | Undecidable | Undecidable |
| $f = \mathsf{DSum}$ | PTime | Open-hard | Open-hard | Undecidable | PTime | PTime | Open-hard | Undecidable |

| **(b)** | Evaluation | Nonemptiness | Universality |
|---|---|---|---|
| $h, g, f \in \{\mathsf{Inf}, \mathsf{Sup}, \mathsf{LimInf}, \mathsf{LimSup}\}$ and at least one of $h$ or $g$ is in $\{\mathsf{LimInf}, \mathsf{LimSup}\}$ | PSpace | PSpace | PSpace |

**Table 1** Complexity of evaluation, nonemptiness, and universality for QLAs with language aggregator $h$, word aggregator $g$, and run aggregator $f$. **(a)** QLAs with $h, g \in \{\mathsf{Inf}, \mathsf{Sup}, \mathbb{E}\}$ (Sections 5 and 6). *Open-hard* means the problem is at least as hard as the universality problem for nondeterministic discounted-sum QWA [8, 15]. Universality of QLAs is dual to their nonemptiness, e.g., it is in PSpace when $h \in \{\mathsf{Inf}, \mathsf{Sup}, \mathbb{E}\}$ and $g = \mathsf{Sup}$ (see Remark 6.1). **(b)** QLAs with $h, g, f \in \{\mathsf{Inf}, \mathsf{Sup}, \mathsf{LimInf}, \mathsf{LimSup}\}$ with $g$ or $h$ in $\{\mathsf{LimInf}, \mathsf{LimSup}\}$ (Section 7).

above $k$. We investigate these problems for QLAs over unrestricted language generators, as well as with regard to natural subclasses of language generators such as finite-state ones.

**Contribution and Overview.** Our main contribution is the definition and systematic study of QLAs within a three-dimensional framework of aggregator functions (run, word, and language aggregators), which supports the specification and verification of quantitative hyperproperties over nonprobabilistic and probabilistic language generators.

Section 2 presents our definitional framework and Section 3 gives several examples of QLAs for specifying quantitative hyperproperties. Section 4 introduces the QLA evaluation, nonemptiness, and universality problems that we study. Section 5 focuses on the evaluation problem of QLAs with word and language aggregators $\mathsf{Inf}$, $\mathsf{Sup}$, or $\mathbb{E}$, and Section 6 on the nonemptiness and universality problems of these. Section 7 switches the focus to QLAs with word and language aggregators $\mathsf{LimInf}$ and $\mathsf{LimSup}$. Section 8 concludes with potential research directions. We summarize our decidability and complexity results in Table 1. We provide proof sketches in the main text; the details are deferred to the appendix.

To present a comprehensive picture, we overcome several technical challenges: First, for the evaluation problem (Section 5), we demonstrate that the value of an $\omega$-regular language $L$ may not be realized (i) by a lasso word in $L$ for limit-average (a.k.a. mean-payoff) QLAs and (ii) by any word in $L$ for discounted-sum QLAs. Although the evaluation problem for these automata is not always solvable, for its solvable cases we resolve these issues by proving that (i) in the limit-average case, the value of $L$ can still be computed by analyzing strongly connected components even if it is not realized by a lasso word, and (ii) in the discounted-sum case, the value of $L$ matches the value of its safety closure $L'$ and is realized by some word in $L'$. These results yield PTime algorithms for the evaluation of these QLAs. We complement these results with hardness proofs via reductions from universality problems of the underlying QWAs.

Second, for nonemptiness and universality (Section 6), we examine the behavior of common QWA classes regarding the greatest lower bound of their values—their so-called bottom values [6]. We show (i) discounted-sum QWAs always have a word achieving their bottom value, and (ii) the bottom value of limit-superior-average QWAs can be approximated by lasso words, contrasting sharply with limit-inferior-average QWAs [15]. This enables us to establish the hardness of nonemptiness and universality for certain classes of QLAs. Moreover, we prove that for most classes of QLAs the unrestricted versions of nonemptiness and universality coincide with their finite-state restrictions, indicating a finite-model property.

Third, for QLAs using $\mathsf{LimInf}$ and $\mathsf{LimSup}$ as word and language aggregators (Section 7), we study values realized by infinitely many runs of a word and infinitely many words of a given language. Using combinatorial arguments, we characterize structural patterns in $\omega$-automata that precisely capture when a value is infinitely realizable with the run aggregators $\mathsf{Inf}$, $\mathsf{Sup}$, $\mathsf{LimInf}$, and $\mathsf{LimSup}$.

**Related Work.** Our work builds on quantitative languages and quantitative word automata [13, 15], as well as on games with quantitative objectives [17]. There have been other definitions of quantitative hyperproperties [19, 33, 36].

While our motivations align, QLAs in their current form and the formalisms of [19, 33] are orthogonal. In [19, 33], a quantitative hyperproperty is based on counting the number of executions that appear in a given relation. Therefore, these formalisms cannot express, e.g., the worst-case server uptime, which QLAs can (see Section 3). Similarly, QLAs with the aggregators we considered cannot express properties that can only be defined using counting or cardinality constraints.

In [36], the authors provide a predicate-transformer theory for reasoning on finite executions of imperative programs. Their formalism can define quantitative hyperproperties that can be expressed as functions from a set or distribution of values at a program's final state to a single value, which is closer to our view compared to [19, 33]. For example, it can express the maximal value of a variable $x$ given the value of variable $y$ upon termination (akin to QLAs) or the variance of a random variable. However, it cannot express long-run aggregates such as limit superior or limit averages. Conversely, QLAs cannot express variances with the current aggregators, since it requires combining two expectations at the language-aggregator level. To the best of our knowledge, our work is the first to define quantitative hyperproperties as functions from sets or distributions of infinite words to quantitative values, and therefore the first to study the specification and verification of such properties through automata.

## 2 Definitional Framework

Let $\Sigma$ be a finite *alphabet* of letters. A *word* (or *trace*) over $\Sigma$ is a finite or infinite sequence of letters from $\Sigma$. We denote by $|w|$ the length of a finite word $w$. We denote by $\Sigma^*$ (resp. $\Sigma^\omega$) the set of all finite (resp. infinite) words over $\Sigma$. An infinite word $w$ is *ultimately periodic* (a.k.a. *lasso*) iff $w = uv^\omega$ for some $u, v \in \Sigma^*$ such that $|v| \geq 1$. A *language* is a set of infinite words. Given $u \in \Sigma^*$ and $w \in \Sigma^\omega$, we write $u \prec w$ when $u$ is a prefix of $w$. We denote by $\mathbb{N}$ the set of natural numbers (including 0), $\mathbb{Q}$ the set of rational numbers, $\mathbb{R}$ the set of real numbers. We further let $\overline{\mathbb{N}} = \mathbb{N} \cup \{\infty\}$ and $\overline{\mathbb{R}} = \mathbb{R} \cup \{-\infty, +\infty\}$. Consider a set $S$. An $S$-multiset is a function $M : S \to \overline{\mathbb{N}}$ that maps each element of $S$ to a value denoting its multiplicity. The support of a multiset $M$ is the set $supp(M) = \{x \in S \mid M(x) \geq 1\}$ of distinct elements in $M$. A *value domain* $\mathbb{D}$ is a nontrivial complete lattice. A *quantitative property* is a total function $\Phi : \Sigma^\omega \to \mathbb{D}$.

**Quantitative Word Automata.** A *weighted labeled transition system* is a tuple $\mathcal{T} = (\Sigma, Q, s, \delta, d)$, where: $\Sigma$ is a finite alphabet, $Q$ is a finite nonempty set of states, $s \in Q$ is the initial state, $\delta\colon Q \times \Sigma \to 2^{\mathbb{Q} \times Q}$ is a finite transition function over weight-state pairs, and $d : Q \times \Sigma \times \mathbb{Q} \times Q \to [0, 1]$ is a probability distribution such that for all $q \in Q$ and $a \in \Sigma$, (i) $d(q, a, x, q') > 0$ iff $(x, q') \in \delta(q, a)$, and (ii) $\sum_{(x,q') \in \delta(q,a)} d(q, a, x, q') = 1$. Given a transition system $\mathcal{T}$, the *dual* of $\mathcal{T}$ is a copy of $\mathcal{T}$ with the weights multiplied by $-1$.

A *transition* is a tuple $(q, \sigma, x, q') \in Q \times \Sigma \times \mathbb{Q} \times Q$ such that $(x, q') \in \delta(q, \sigma)$, denoted $q \xrightarrow{\sigma:x} q'$. Given a transition $t = q \xrightarrow{\sigma:x} q'$, we denote its weight by $\gamma(t) = x$. We say that $\mathcal{T}$ is *complete* (a.k.a. *total*) iff $|\delta(q, a)| \geq 1$ for every $q \in Q$ and $a \in \Sigma$, and *deterministic* iff $|\delta(q, a)| = 1$ for every $q \in Q$ and $a \in \Sigma$. Throughout the paper, we assume that weighted labeled transition systems are complete.

Although weighted labeled transition systems are probabilistic by definition, they can be viewed as *nondeterministic* by considering only the support of $d$, i.e., treating all transitions with positive probability as nondeterministic transitions.

A *run* of $\mathcal{T}$ is an infinite sequence $\rho = q_0 \xrightarrow{\sigma_0:x_0} q_1 \xrightarrow{\sigma_1:x_1} \ldots$ of transitions such that $q_0 = s$ and $(x_i, q_{i+1}) \in \delta(q_i, \sigma_i)$ for all $i \geq 0$. We write $\gamma(\rho)$ for the corresponding infinite sequence $x_0 x_1 \ldots$ of rational weights. Given a word $w \in \Sigma^\omega$, we denote by $R(\mathcal{T}, w)$ the set of runs of $w$ over $\mathcal{T}$. Let $\rho = t_0 t_1 \ldots$ be a run and $r = t_0 t_1 \ldots t_n$ be a finite prefix of $\rho$. The probability of $r$ is $\prod_{i=0}^{n} d(t_i)$. For each infinite word $w$, we define $\mu_{\mathcal{T},w}$ as the unique probability measure over Borel sets of infinite runs of $\mathcal{T}$ over $w$, induced by the transition probabilities $d$.

A *run aggregator* (a.k.a. value function) is a function $f : \mathbb{Q}^\omega \to \overline{\mathbb{R}}$ that accumulates an infinite sequence of weights into a single value. We consider the run aggregators below over an infinite sequence $x = x_0 x_1 \ldots$ of rational weights and a discount factor $\lambda \in \mathbb{Q} \cap (0, 1)$. We write $\mathsf{DSum}$ when the discount factor $\lambda \in \mathbb{Q} \cap (0, 1)$ is unspecified. The run aggregators $\mathsf{Inf}$ and $\mathsf{Sup}$ (resp., $\mathsf{LimInf}$ and $\mathsf{LimSup}$, $\mathsf{LimInfAvg}$ and $\mathsf{LimSupAvg}$) are duals. The dual of $\mathsf{DSum}_\lambda$ is itself.

- $\mathsf{Inf}(x) = \inf\{x_n \mid n \in \mathbb{N}\}$
- $\mathsf{Sup}(x) = \sup\{x_n \mid n \in \mathbb{N}\}$
- $\mathsf{LimInf}(x) = \liminf_{n\to\infty} x_n$
- $\mathsf{LimSup}(x) = \limsup_{n\to\infty} x_n$
- $\mathsf{LimInfAvg}(x) = \liminf_{n\to\infty} \left(\frac{1}{n}\sum_{i=0}^{n-1} x_i\right)$
- $\mathsf{LimSupAvg}(x) = \limsup_{n\to\infty} \left(\frac{1}{n}\sum_{i=0}^{n-1} x_i\right)$
- $\mathsf{DSum}_\lambda(x) = \sum_{i\geq 0} \lambda^i x_i$

A *word aggregator* is a function $g : \overline{\mathbb{N}}^{\overline{\mathbb{R}}} \to \overline{\mathbb{R}}$ that accumulates a multiset of values obtained from the runs of a word into a single value. We consider the word aggregators defined below where $M$ is an $\overline{\mathbb{R}}$-multiset and $\mu$ is a probability measure over $supp(M)$. The word aggregators $\mathsf{Inf}$ and $\mathsf{Sup}$ (resp., $\mathsf{LimInf}$ and $\mathsf{LimSup}$) are duals. The dual of $\mathbb{E}_\mu$ is itself.

- $\mathsf{Inf}(M) = \inf\{x \mid M(x) \geq 1\}$
- $\mathsf{Sup}(M) = \sup\{x \mid M(x) \geq 1\}$
- $\mathbb{E}_\mu(M) = \int_{x \in supp(M)} x d\mu(x)$
- $\mathsf{LimInf}(M) = \inf\{x \mid M(x) = \infty\}$
- $\mathsf{LimSup}(M) = \sup\{x \mid M(x) = \infty\}$

A *quantitative word automaton* (QWA) [15] is a tuple $\mathcal{A} = (g, f, \mathcal{T})$ where $\mathcal{T}$ is a complete weighted labeled transition system, $f$ is a run aggregator, and $g$ is a word aggregator. Given a word $w$, a transition system $\mathcal{T}$, and a run aggregator $f$, we define an $\overline{\mathbb{R}}$-multiset $M_{w,\mathcal{T},f}$ such that for every $x \in \overline{\mathbb{R}}$ the value $M_{w,\mathcal{T},f}(x)$ equals the number of runs $\rho$ of $\mathcal{T}$ over $w$ such that $f(\gamma(\rho)) = x$. For all $x \in supp(M_{w,\mathcal{T},f})$ we define $\mu(x) = \mu_{\mathcal{T},w}(\{\rho \in R(\mathcal{T}, w) \mid f(\gamma(\rho)) = x\})$. We let $\mathcal{A}(w) = g(M_{w,\mathcal{T},f})$ for all $w \in \Sigma^\omega$ and write $\mathbb{E}$ for $\mathbb{E}_\mu$ when $\mu$ is clear from the context.

A QWA is *nondeterministic* when its word aggregator is $\mathsf{Sup}$, and *universal* when $\mathsf{Inf}$. The *top value* of a word automaton $\mathcal{A}$ is $\top_\mathcal{A} = \sup_{w\in\Sigma^\omega} \mathcal{A}(w)$, and its *bottom value* is $\bot_\mathcal{A} = \inf_{w\in\Sigma^\omega} \mathcal{A}(w)$. We say that $\top_\mathcal{A}$ (resp. $\bot_\mathcal{A}$) is *achievable* iff there exists a word $w$ with $\mathcal{A}(w) = \top_\mathcal{A}$ (resp. $\mathcal{A}(w) = \bot_\mathcal{A}$). Given a word automaton $\mathcal{A} = (g, f, \mathcal{T})$ its *dual* is $\hat{\mathcal{A}} = (\hat{g}, \hat{f}, \hat{\mathcal{T}})$ where $\hat{g}$, $\hat{f}$, and $\hat{\mathcal{T}}$ are the duals of $g$, $f$, and $\mathcal{T}$, respectively. For a QWA $\mathcal{A} = (g, f, \mathcal{T})$ and a word $w$, if $M_{w,\mathcal{T},f}(x)$ is finite for all $x \in \overline{\mathbb{R}}$, then we let $\mathcal{A}(w) = \top_\mathcal{A}$ when $g = \mathsf{LimInf}$ and $\mathcal{A}(w) = \bot_\mathcal{A}$ when $g = \mathsf{LimSup}$.

Boolean word automata are a special case of QWAs with weights in $\{0, 1\}$ and $f \in \{\mathsf{Inf}, \mathsf{Sup}, \mathsf{LimInf}, \mathsf{LimSup}\}$. In particular, a nondeterministic Büchi automaton is a boolean word automaton with $g = \mathsf{Sup}$ and $f = \mathsf{LimSup}$. Given a boolean word automaton $\mathcal{B}$, we write $L(\mathcal{B}) = \{w \in \Sigma^\omega \mid \mathcal{B}(w) = 1\}$ for its language.

**Quantitative Language Automata.** A *language generator* is a function $G : 2^{\Sigma^\omega} \to [0, 1]$. A language generator $G$ is *nonprobabilistic* iff there is a language $L_G \subseteq \Sigma^\omega$ such that $G(L_G) = 1$ and $G(L) = 0$ for every other language $L \neq L_G$. A nonprobabilistic language generator $G$ is *nonempty* iff $L_G \neq \emptyset$. A language generator $G$ is *probabilistic* iff it defines a probability measure $\mu_G$ over $\Sigma^\omega$. A *nonprobabilistic quantitative hyperproperty* (resp. *probabilistic quantitative hyperproperty*) is a total function from the set of all nonprobabilistic (resp. probabilistic) language generators to a value domain $\mathbb{D}$.

A *language aggregator* is a function $h : \overline{\mathbb{N}}^{\overline{\mathbb{R}}} \to \overline{\mathbb{R}}$ that accumulates a multiset of values obtained from a set of words into a single value. We consider the language aggregators $\mathsf{Inf}$, $\mathsf{Sup}$, $\mathsf{LimInf}$, $\mathsf{LimSup}$, and $\mathbb{E}_\mu$, which are defined the same as for the word aggregators above.

A *quantitative language automaton* (QLA) is a pair $\mathbb{A} = (h, \mathcal{A})$ where $\mathcal{A} = (g, f, \mathcal{T})$ is a QWA and $h$ is a language aggregator. Consider a language generator $G$ and a QWA $\mathcal{A}$. If $G$ is nonprobabilistic (resp. probabilistic), we define $M_{G,\mathcal{A}}$ as an $\overline{\mathbb{R}}$-multiset such that for every $x \in \overline{\mathbb{R}}$ the value $M_{G,\mathcal{A}}(x)$ equals the number of words $w$ in $L_G \subseteq \Sigma^\omega$ (resp. in $\Sigma^\omega$) such that $\mathcal{A}(w) = x$. Moreover, if $G$ is probabilistic, we additionally let $\mu(x) = \mu_G(\{w \in \Sigma^\omega \mid \mathcal{A}(w) = x\})$. Then, we let $\mathbb{A}(G) = h(M_{G,\mathcal{A}})$ for every language generator $G$ where $\mu$ is as above. We write $\mathbb{E}$ for $\mathbb{E}_\mu$ when $\mu$ is clear from the context.

Below, we assume the input to a QLA is a nonprobabilistic (resp. probabilistic) language generator when $h \neq \mathbb{E}$ (resp. $h = \mathbb{E}$). We write $L$ to denote a nonprobabilistic language generator, and $\mu$ to denote a probabilistic one.

The *top value* of a QLA $\mathbb{A}$, denoted $\top_{\mathbb{A}}$, is the supremum of the values $\mathbb{A}(G)$ over all inputs $G$, and its *bottom value*, denoted $\bot_{\mathbb{A}}$, is the infimum. For a QLA $\mathbb{A} = (h, \mathcal{A})$ and a nonprobabilistic language generator $L$, if $M_{L,\mathcal{A}}(x) = 0$ for all $x \in \overline{\mathbb{R}}$ (i.e., when $L = \emptyset$), then we let $\mathbb{A}(L) = \top_{\mathbb{A}}$ when $h = \mathsf{Inf}$ and $\mathbb{A}(L) = \bot_{\mathbb{A}}$ when $h = \mathsf{Sup}$. Similarly, if $M_{L,\mathcal{A}}(x)$ is finite for all $x \in \overline{\mathbb{R}}$, then we let $\mathbb{A}(L) = \top_{\mathbb{A}}$ when $h = \mathsf{LimInf}$ and $\mathbb{A}(L) = \bot_{\mathbb{A}}$ when $h = \mathsf{LimSup}$.

▶ **Proposition 2.1.** *Consider a QLA $\mathbb{A} = (h, \mathcal{A})$ with $h \in \{\mathsf{Inf}, \mathsf{Sup}, \mathbb{E}\}$. Then, $\top_{\mathbb{A}} = \top_{\mathcal{A}}$ and $\bot_{\mathbb{A}} = \bot_{\mathcal{A}}$.*

Given a QLA $\mathbb{A} = (h, \mathcal{A})$ its *dual* is $\hat{\mathbb{A}} = (\hat{h}, \hat{\mathcal{A}})$ where $\hat{h}$ and $\hat{\mathcal{A}}$ are the duals of $h$ and $\mathcal{A}$.

▶ **Proposition 2.2.** *Consider a QLA $\mathbb{A}$ and its dual $\hat{\mathbb{A}}$. Then, $\mathbb{A}(G) = -\hat{\mathbb{A}}(G)$ for every language generator $G$.*

## 3 Applications of Language Automata

While QWAs describe system specifications *per execution*, QLAs describe *system-wide* specifications. We show several applications of QLAs for specifying quantitative hyperproperties. Notice that QLAs are an *operational* specification language—their underlying word automata serve as part of the specification itself.

**Server Uptime.** QWAs can model performance metrics for individual executions. Let $\Sigma = \{\mathtt{on}, \mathtt{off}\}$ be a finite alphabet of observations modeling a server's activity. Consider the weighted labeled transition system $\mathcal{T}_{\text{up}}$ given in Figure 1, and let $\mathcal{A}_{\text{up}} = (g, f, \mathcal{T}_{\text{up}})$ be a QWA where $g = \mathsf{Sup}$ and $f = \mathsf{LimInfAvg}$. The automaton $\mathcal{A}_{\text{up}}$ maps each infinite word $w$ to the long-run ratio of $\mathtt{on}$, i.e., the average uptime of the execution modeled by $w$. For example, if $w = \mathtt{on} \cdot (\mathtt{on} \cdot \mathtt{off})^\omega$, we have $\mathcal{A}(w) = 0.5$.

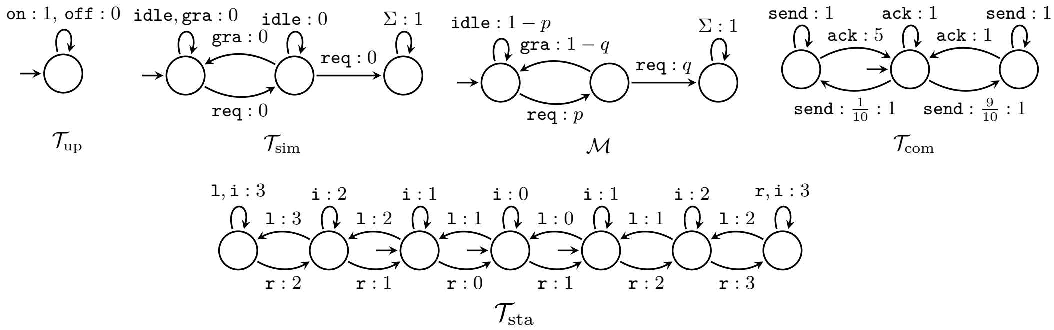


**Figure 1** Weighted labeled transition systems $\mathcal{T}_{\text{up}}$, $\mathcal{T}_{\text{sim}}$, $\mathcal{T}_{\text{com}}$, and $\mathcal{T}_{\text{sta}}$ can be used to specify QWAs and QLAs with various aggregators, as demonstrated in Section 3. Transitions are marked with $\sigma : p : x$ where $\sigma$ denotes a letter, $p$ the transition probability (dropped when $p = 1$), and $x$ the transition weight. The Markov chain $\mathcal{M}$ is a language generator where $0 \leq p, q \leq 1$ (its transitions are not weighted, and those with probability 0 are not shown).

QLAs can model *system-wide* performance metrics. Let $\mathbb{A}_{\text{up}} = (h, \mathcal{A}_{\text{up}})$. Setting $h = \mathsf{Inf}$ tunes $\mathbb{A}_{\text{up}}$ to map any given language $L$, capturing the implementation of a server, to the worst-case uptime over all the executions allowed by the implementation, i.e., $\mathbb{A}_{\text{up}}(L) = \inf_{w \in L} \mathcal{A}_{\text{up}}(w)$. For example, let $U = \{\texttt{on}^n \texttt{off}^k \mid n > k \geq 0\}$ and consider the language $L = \{u_1 u_2 \ldots \mid \forall i : u_i \in U\}$ of server executions where each block of $\texttt{on}$'s is followed by a strictly shorter block of $\texttt{off}$'s. Although for every $u \in U$ there are strictly more $\texttt{on}$'s than $\texttt{off}$'s, we have $w = s_1 s_2 \ldots \in L$ where $s_i = \texttt{on}^i \texttt{off}^{i-1} \in U$ for each $i \geq 1$, thus $\mathbb{A}_{\text{up}}(L) = 0.5$. In particular, notice that the value of $\mathbb{A}_{\text{up}}(L)$ cannot be achieved by any ultimately periodic word in $L$; we will later show that this may happen even for $\omega$-regular languages (Proposition 5.2). As another example of using QLAs to specify quantitative hyperproperties, setting $h = \mathsf{LimInf}$ tunes $\mathbb{A}_{\text{up}}$ to express the "almost" worst-case uptime by considering only the uptime values realized by infinitely many executions. For example, let $L = \Sigma^* \texttt{on}^\omega \cup \{(\texttt{on}^n \texttt{off})^\omega \mid n \geq 0\}$. The part $\Sigma^* \texttt{on}^\omega$ yields infinitely many executions with uptime 1, while each $(\texttt{on}^n \texttt{off})^\omega$ with uptime $\frac{n}{n+1}$ appears only once. Thus, even though the infimum over $L$ is 0, only uptime 1 occurs infinitely often, so $\mathbb{A}(L) = 1$.

**Implementation Distance.** QWAs can specify the distance of individual executions to a desired boolean trace property. Let $\Sigma = \{\texttt{req}, \texttt{gra}, \texttt{idle}\}$ be a finite alphabet of observations modeling a server receiving requests and issuing grants, and consider the boolean safety property $P$ requiring that no two requests are simultaneously open (i.e., there is a $\texttt{gra}$ between every two $\texttt{req}$'s). Consider the transition system $\mathcal{T}_{\text{sim}}$ given in Figure 1, and let $\mathcal{A}_{\text{sim}} = (g, f, \mathcal{T}_{\text{sim}})$ be a QWA where $f = \mathsf{DSum}_{0.5}$ and $g \in \{\mathsf{Inf}, \mathsf{Sup}\}$ (the choice of $g$ does not matter since $\mathcal{T}_{\text{sim}}$ is deterministic). The automaton $\mathcal{A}_{\text{sim}}$ maps each infinite word $w$ to the smallest Cantor distance between $w$ and some word $w' \in P$, i.e., $\mathcal{A}_{\text{sim}}(w) = \inf_{w' \in P} d(w, w')$. For example, if $w = \texttt{req} \cdot \texttt{idle} \cdot \texttt{req}^\omega$, we have $\mathcal{A}(w) = 0.25$.

QLAs can specify the distance of *systems* to a desired boolean trace property. Let $\mathbb{A}_{\text{sim}} = (h, \mathcal{A}_{\text{sim}})$. Setting $h = \mathsf{Sup}$ tunes $\mathbb{A}_{\text{sim}}$ to map any given language $L$, representing a server implementation, to the worst-case distance from $L$ to $P$. In other words, $\mathbb{A}_{\text{sim}}(L) = \sup_{w \in L} \inf_{w' \in P} d(w, w')$ where $d$ denotes the Cantor distance. For example, if $L = \Sigma^* \texttt{req}^2 \Sigma^\omega$, then $\mathbb{A}_{\text{sim}}(L) = 0.5$. Setting $h = \mathsf{Inf}$ tunes $\mathbb{A}_{\text{sim}}$ to map each language to the best-case distance from $L$ to $P$. For example, if $L$ is as above, the value of $L$ would be 0 although all the words in $L$ violate $P$. This is because for every $i \geq 0$ the word $w_i = \texttt{idle}^i \texttt{req}^2 \texttt{idle}^\omega$ belongs to $L$ and $\lim_{i \to \infty} \mathbb{A}_{\text{sim}}(w_i) = 0$, highlight a challenge for the evaluation of discounted-sum QLAs:

the value of $L$ may not be realized by any word in $L$, even if $L$ is omega-regular. Setting $h = \mathbb{E}$ tunes $\mathbb{A}_{\text{sim}}$ to map each language to the average-case distance to $P$. For example, consider the Markov chain $\mathcal{M}$ from Figure 1. The expected value can be computed by solving the corresponding system of linear equations, resulting in $\mathbb{A}_{\text{sim}}(\mu_{\mathcal{M}}) = \frac{2pq}{2+p(1+q)}$.

**Robot Stability.** QWAs can express stability constraints of individual executions. Let $\Sigma = \{\mathtt{l}, \mathtt{r}, \mathtt{i}\}$ be a finite alphabet representing robot movements—*left*, *right*, and *idle*—on a one-dimensional finite grid. An execution is $\varepsilon$-stable, for some $\varepsilon \geq 0$, if there exists $0 \leq \delta \leq \varepsilon$ such that whenever the system starts within a $\delta$-ball around the origin, it remains indefinitely within an $\varepsilon$-ball around the origin. For a fixed $\delta \geq 0$, QWAs can express the most permissive $\varepsilon$ associated with each execution. Consider the weighted labeled transition system $\mathcal{T}_{\text{sta}}$ given in Figure 1, and let $\mathcal{A}_{\text{sta}} = (\mathsf{Sup}, \mathsf{Sup}, \mathcal{T}_{\text{sta}})$. The automaton captures the scenario where $\delta = 1$, meaning the robot starts at most one step away from the origin. Transition weights indicate distances from the origin, and a run's value is the maximal distance reached. Thus, the automaton's value on a word is the worst-case distance (or most permissive $\varepsilon$) over all initial positions. For instance, the word $w = (\mathtt{l} \cdot \mathtt{r})^\omega$ has three runs with values 1, 1, and 2, so $\mathcal{A}_{\text{sta}}(w) = 2$.

QLAs can express stability constraints of *systems*. Let $\mathbb{A}_{\text{sta}} = (h, \mathcal{A}_{\text{sta}})$. If $h = \mathsf{Sup}$, then given a language $L$ modeling the robot's behavior, the automaton $\mathbb{A}_{\text{sta}}$ maps $L$ to least upper bound of the set of per-execution $\varepsilon$ values obtained from $L$ and $\mathcal{A}_{\text{sta}}$. In other words, $\mathbb{A}_{\text{sta}}(L) = \mathsf{Sup}_{w \in L} \mathbb{A}_{\text{sta}}(w)$ is the smallest $\varepsilon$ ensuring all executions in $L$ are $\varepsilon$-stable with $\delta = 1$. Alternatively, if $h = \mathsf{LimSup}$, then $\mathbb{A}_{\text{sta}}(L)$ captures the smallest $\varepsilon$ such that infinitely many executions are $\varepsilon$-stable for $\delta = 1$, allowing to discard "outlier" values achieved only by finitely many executions.

**Communication Channel Cost.** Probabilistic QWAs can specify the expected cost of individual executions. Let $\Sigma = \{\mathtt{send}, \mathtt{ack}\}$ be an alphabet modeling a communication channel. Consider the transition system $\mathcal{T}_{\text{com}}$ given in Figure 1, and let $\mathcal{A}_{\text{com}} = (\mathbb{E}, \mathsf{LimSup}, \mathcal{T}_{\text{com}})$ be a probabilistic QWA. Each run of the automaton is mapped to a long-term maximal cost, i.e., $\mathsf{LimSup}$ of the corresponding weight sequence. Then, each infinite word is mapped to the expected value over the distribution of its runs. For example, considering $w = (\mathtt{send} \cdot \mathtt{ack})^\omega$ gives us $\mathcal{A}_{\text{com}}(w) = 5$ because the set of runs with a $\mathsf{LimSup}$ value 5 (i.e., those in which the high-cost cycle occurs infinitely often) has probability 1.

QLAs can specify the aggregate cost of a communication channel. Let $\mathbb{A}_{\text{com}} = (h, \mathcal{A}_{\text{com}})$. If $h = \mathbb{E}$, the QLA $\mathbb{A}_{\text{com}}$ specifies the expected cost of the underlying probabilistic model of a communication channel. Consider a Markov chain defining a uniform probability measure $\mu$ over the alphabet $\Sigma = \{\mathtt{send}, \mathtt{ack}\}$. Then, $\mathbb{A}(\mu) = 1.05$ which can be computed by analyzing the product of the Markov chain defining $\mu$ and the underlying word automaton $\mathcal{A}_{\text{com}}$.

## 4 Problems on Language Automata

Let $\mathbb{A} = (h, \mathcal{A})$ be a QLA. We study the following problems for $\triangleright \in \{>, \geq\}$.

**Nonprobabilistic Evaluation ($h \neq \mathbb{E}$).** Given a Büchi automaton $\mathcal{B}$, compute $\mathbb{A}(\mathcal{B})$.
**Probabilistic Evaluation ($h = \mathbb{E}$).** Given a finite-state Markov chain $\mathcal{M}$, compute $\mathbb{A}(\mathcal{M})$.
**$\triangleright$-Nonemptiness.** Given $k \in \mathbb{Q}$, is $\mathbb{A}(G) \triangleright k$ for some (nonempty) language generator $G$?
**$\triangleright$-Universality.** Given $k \in \mathbb{Q}$, is $\mathbb{A}(G) \triangleright k$ for every (nonempty) language generator $G$?

For nonemptiness and universality, we also consider the following variants with restricted quantification over language generators.

**Borel.** Nonprobabilistic generators $G$ with $L_G$ Borel in $\Sigma^\omega$ under the Cantor topology.
**Markov.** Probabilistic generators $G$ with $\mu_G$ Markovian.
**Finite-state.** Nonprobabilistic generators $G$ with $L_G$ $\omega$-regular; probabilistic generators $G$ with $\mu_G$ finite-state Markovian.

We establish the relations between the questions about QLAs and the corresponding questions about their underlying QWA. For completeness, we provide these problem definitions here. Consider a QWA $\mathcal{A}$, a rational $k \in \mathbb{Q}$, and $\triangleright \in \{>, \geq\}$. Then, $\mathcal{A}$ is $\triangleright$-nonempty (resp -universal) for $k$ iff $\mathcal{A}(w) \triangleright k$ for some (resp. all) $w \in \Sigma^\omega$. Similarly as for the finite-state restriction for QLAs problems, the *lasso-word restriction* for these problems requires quantifying over lasso words instead of all words. Moreover, $\mathcal{A}$ is *approximate-nonempty* for $k$ iff $\top_{\mathcal{A}} \geq k$, i.e., if there exists $w \in \Sigma^\omega$ such that $\mathcal{A}(w) \geq k$ or if for every $\varepsilon > 0$ there exists $w$ such that $\mathcal{A}(w) > k - \varepsilon$. Hence, the achievability of $\top_{\mathcal{A}}$ implies that approximate-nonemptiness and $\geq$-nonemptiness are equivalent problems. Note that $\top_{\mathcal{A}} > k$ iff $\mathcal{A}(w) > k$ for some word $w$—it holds independently of the achievability of $\top_{\mathcal{A}}$. Dually, $\mathcal{A}$ is *approximate-universal* for $k$ iff $\perp_{\mathcal{A}} > k$.

For the classes of QLAs we consider, the Borel (resp. Markov) restrictions of the decision problems coincide with the unrestricted cases. Thus, for their decision problems we only focus on the unrestricted and finite-state cases.

▶ **Proposition 4.1.** *Consider a QLA $\mathbb{A} = (h, \mathcal{A})$ with $h \in \{\mathsf{Inf}, \mathsf{Sup}, \mathsf{LimSup}, \mathsf{LimInf}\}$ (resp. $h = \mathbb{E}$), a rational $k \in \mathbb{Q}$, and $\triangleright \in \{>, \geq\}$. Then, $\mathbb{A}$ is $\triangleright$-nonempty for $k$ iff $\mathbb{A}$ is Borel $\triangleright$-nonempty (resp. Markov $\triangleright$-nonempty) for $k$. The same holds for universality.*

For QLAs with language aggregators $\mathsf{Inf}$, $\mathsf{Sup}$, and $\mathbb{E}$, we demonstrate that their decision problems often reduce to those of their underlying QWAs. Moreover, we identify several cases in which QLAs enjoy a finite-model property.

▶ **Proposition 4.2.** *Consider a QLA $\mathbb{A} = (h, \mathcal{A})$ with $h \in \{\mathsf{Inf}, \mathsf{Sup}, \mathbb{E}\}$, a rational $k \in \mathbb{Q}$, and $\triangleright \in \{>, \geq\}$.*

**(a)** *Unrestricted variant*
- **(i)** *If $h \neq \mathsf{Sup}$ or $\triangleright = >$, then $\mathbb{A}$ is $\triangleright$-nonempty for $k$ iff $\mathcal{A}$ is $\triangleright$-nonempty for $k$.*
- **(ii)** *If $h = \mathsf{Sup}$, then $\mathbb{A}$ is $\geq$-nonempty for $k$ iff $\mathcal{A}$ is approximate-nonempty for $k$.*

**(b)** *Finite-state variant*
- **(i)** *If $h = \mathsf{Sup}$, then $\mathbb{A}$ is finite-state $\triangleright$-nonempty for $k$ iff $\mathbb{A}$ is $\triangleright$-nonempty for $k$.*
- **(ii)** *If $h = \mathsf{Inf}$, then $\mathbb{A}$ is finite-state $\triangleright$-nonempty for $k$ iff $\mathcal{A}$ is lasso-word $\triangleright$-nonempty for $k$.*
- **(iii)** *If $\top_{\mathcal{A}}$ is achievable by a lasso word, then $\mathbb{A}$ is finite-state $\triangleright$-nonempty for $k$ iff $\mathbb{A}$ is $\triangleright$-nonempty for $k$ iff $\mathcal{A}$ is lasso-word $\triangleright$-nonempty for $k$.*

*For universality, we have the duals where we exchange $\mathsf{Inf}$ with $\mathsf{Sup}$, $>$ with $\geq$ , nonempty with universal, approximate-nonempty with approximate-universal, and $\top_{\mathcal{A}}$ with $\perp_{\mathcal{A}}$.*

We show that approximating top or bottom values via lasso words lets us reduce certain unrestricted variants to their finite-state counterparts, establishing a finite-model property.

▶ **Proposition 4.3.** *Consider a QLA $\mathbb{A} = (h, \mathcal{A})$ with $h \in \{\mathsf{Inf}, \mathsf{Sup}, \mathbb{E}\}$ and $k \in \mathbb{Q}$. If for every $\varepsilon > 0$ there is a lasso word $w$ such that $\mathcal{A}(w) \geq \top_{\mathcal{A}} - \varepsilon$, then $\mathbb{A}$ is $>$-nonempty for $k$ iff $\mathbb{A}$ is finite-state $>$-nonempty for $k$. Dually, if for every $\varepsilon > 0$ there is a lasso word $w$ such that $\mathcal{A}(w) \leq \perp_{\mathcal{A}} + \varepsilon$, then $\mathbb{A}$ is $\geq$-universal for $k$ iff $\mathbb{A}$ is finite-state $\geq$-universal for $k$.*

# 5 Solving Evaluation

In this section, we investigate the evaluation problem for QLAs with language and word aggregators $h, g \in \{\mathsf{Inf}, \mathsf{Sup}, \mathbb{E}\}$, for which we provide a full picture of complexity results. First, in Section 5.1, we focus on the nonprobabilistic evaluation problem (where $h \in \{\mathsf{Inf}, \mathsf{Sup}\}$) and then, in Section 5.2, on the probabilistic one (where $h = \mathbb{E}$).

## 5.1 Nonprobabilistic Evaluation

First, we consider the evaluation problem for QLAs with $h \in \{\mathsf{Inf}, \mathsf{Sup}\}$. We start with QLAs whose underlying word automata are universal or nondeterministic (i.e., $g \in \{\mathsf{Inf}, \mathsf{Sup}\}$). We study various run aggregators $f$ separately and show that the problem is in PTime when the word aggregator $g$ and the language aggregator $h$ coincide. When they differ, the problem becomes harder: while it remains algorithmically solvable in PSpace for the "standard" run aggregators (i.e., $f \in \{\mathsf{Inf}, \mathsf{Sup}, \mathsf{LimInf}, \mathsf{LimSup}\}$), we show that it is not computable for limit-average and at least as hard as a long-standing open problem for discounted-sum. Finally, for QLAs whose underlying word automata are probabilistic (i.e., $g = \mathbb{E}$), we show that the problem is not computable. At their core, the easiness results rely on analyzing the extreme values of the underlying word automata. Similarly, we establish the hardness results by reductions from word automata problems.

**QLAs with Standard QWAs.** For QLAs with run aggregators $f \in \{\mathsf{Inf}, \mathsf{Sup}, \mathsf{LimInf}, \mathsf{LimSup}\}$, the nonprobabilistic evaluation problem can be solved by reasoning on lasso words since both top and bottom values of the underlying word automata are realized by lasso words.

▶ **Theorem 5.1.** *Consider a QLA $\mathbb{A} = (h, (g, f, \mathcal{T}))$ with $f \in \{\mathsf{Inf}, \mathsf{Sup}, \mathsf{LimInf}, \mathsf{LimSup}\}$ and $g, h \in \{\mathsf{Inf}, \mathsf{Sup}\}$. Let $S \subseteq \Sigma^\omega$ be an $\omega$-regular language given by a Büchi automaton. The value $\mathbb{A}(S)$ is computable in PTime when $g = h$ and in PSpace when $g \neq h$.*

**QLAs with Limit-Average QWAs.** QLAs with run aggregators $f \in \{\mathsf{LimInfAvg}, \mathsf{LimSupAvg}\}$ differ from the previous case in the sense that it is not sufficient to consider only the lasso words in a given $\omega$-regular language, even when the underlying word automaton is deterministic. To witness this, consider the best-case average uptime QLA $\mathbb{A} = (\mathsf{Sup}, (\mathsf{Sup}, \mathsf{LimInfAvg}, \mathcal{T}_{\mathrm{up}}))$ as in Section 3 and Figure 1. Having $L = (\Sigma^* \mathtt{off})^\omega$, we get $\mathbb{A}(L) = 1$ as there is a word in $L$ with infinitely many `off`'s but longer and longer blocks of `on`'s, but no lasso word in $L$ has an average uptime of 1.

▶ **Proposition 5.2.** *There is a QLA $\mathbb{A}$ with a run aggregator $f \in \{\mathsf{LimInfAvg}, \mathsf{LimSupAvg}\}$ and an $\omega$-regular language $S$ such that no lasso word in $S$ achieves the value $\mathbb{A}(S)$.*

Nonetheless, for QLAs with matching word and language aggregators, we show that the value of an $\omega$-regular language given by a Büchi automaton $\mathcal{B}$ is computable by analyzing the strongly connected components (SCCs) of the underlying word automaton's product with $\mathcal{B}$ as follows: Among all SCCs that are reachable from the initial state, we find the ones that contain at least one state whose Büchi component is accepting. Then, in each such SCC, we compute the maximum mean weight of its cycles by Karp's algorithm [26]. The largest among these mean values is exactly the value of the given language. Even though such a cycle may not involve an accepting state of $\mathcal{B}$, we can construct a run of the product that visits an accepting state infinitely often while going over this cycle with increasing frequency (hence the long-run average converges to the cycle's mean). When these aggregators differ, the problem is undecidable by reduction from the universality of limit-average QWAs [12, 17, 24].

▶ **Theorem 5.3.** *Consider a QLA $\mathbb{A} = (h, (g, f, \mathcal{T}))$ with $f \in \{\mathsf{LimInfAvg}, \mathsf{LimSupAvg}\}$ and $g, h \in \{\mathsf{Inf}, \mathsf{Sup}\}$. Let $S \subseteq \Sigma^\omega$ be an $\omega$-regular language given by a Büchi automaton. The value $\mathbb{A}(S)$ is computable in PTIME when $g = h$ and not computable when $g \neq h$.*

**QLAs with Discounted-Sum QWAs.** QLAs with the run aggregator $f = \mathsf{DSum}$ have the particular behavior that the value assigned to an $\omega$-regular language $L$ may be not achievable by any word in $L$, even when the underlying word automaton is deterministic. To witness this, consider $\mathbb{A} = (\mathsf{Sup}, \mathcal{A})$ with $\mathcal{A} = (\mathsf{Sup}, \mathsf{DSum}_{0.5}, \mathcal{T}_{\mathrm{up}})$ as in Figure 1. We have $\mathbb{A}((\Sigma^*\mathtt{off})^\omega) = 2$ since $\lim_{n\to\infty} \mathcal{A}(\mathtt{on}^n\mathtt{off}^\omega) = 2$. However, only for $w = \mathtt{on}^\omega \notin (\Sigma^*\mathtt{off})^\omega$ we have $\mathcal{A}(w) = 2$.

▶ **Proposition 5.4.** *There is a QLA $\mathbb{A}$ with the run aggregator $f = \mathsf{DSum}$ and an $\omega$-regular language $S$ such that no word in $S$ achieves the value $\mathbb{A}(S)$.*

We establish that such a behavior is not possible when the input language includes all its limit points, i.e., it is safe in the boolean sense [2, 28]: Consider a sequence of words in the safety language whose values approach the supremum. We build by a diagonalization argument an infinite word $w$ whose every finite prefix already appears in the language, so $w$ is in the safety language. Applying the same construction to the corresponding optimal runs yields an infinite run on $w$. This run's value equals the supremum since the contribution of the remaining tail is bounded by a vanishing geometric series due to discounting.

▶ **Proposition 5.5.** *Consider a QLA $\mathbb{A} = (\mathsf{Sup}, (\mathsf{Sup}, \mathsf{DSum}, \mathcal{T}))$. For every nonempty safety language $S \subseteq \Sigma^\omega$, the value $\mathbb{A}(S)$ is achievable by some run of a word in $S$.*

The value of a language $S$ matches that of its safety closure $S'$ (i.e., the smallest safety language containing it) because every word in the safety closure can be approximated arbitrarily closely by words from the original language: If $S'$ achieves a value on a word $w$, we can isolate a prefix of $w$ whose contribution is close enough to the value of $w$. By construction, the same prefix also occurs in a word of $S$, and completing the run along this word in $S$ can change the total value by at most an arbitrarily small amount due to discounting. Hence the maximal value in $S'$ can be approximated arbitrarily closely by words in $S$, and the two suprema coincide.

▶ **Proposition 5.6.** *Consider a QLA $\mathbb{A} = (\mathsf{Sup}, (\mathsf{Sup}, \mathsf{DSum}, \mathcal{T}))$. For every language $S \subseteq \Sigma^\omega$ we have $\mathbb{A}(S) = \mathbb{A}(S')$ where $S'$ is the safety closure of $S$.*

The observation above helps us provide a PTIME algorithm when the word and language aggregators match: We first construct the Büchi automaton's safety closure, so the optimal value is achieved by a run that never reaches the rejecting sink. Then, we compute the product of the underlying word automaton and the safety closure automaton. Computing the best (or worst) discounted sum over all sink-avoiding paths in the product can be done by solving a one-player discounted-payoff game [4]. When the two aggregators differ, the evaluation problem is at least as hard as the universality problem for nondeterministic discounted-sum automata, which is a long-standing open problem [8, 15].

▶ **Theorem 5.7.** *Consider a QLA $\mathbb{A} = (h, (g, \mathsf{DSum}, \mathcal{T}))$ with $g, h \in \{\mathsf{Inf}, \mathsf{Sup}\}$. Let $S \subseteq \Sigma^\omega$ be an $\omega$-regular language given by a Büchi automaton. The value $\mathbb{A}(S)$ is computable in PTIME when $g = h$. If $\mathbb{A}(S)$ is computable when $g \neq h$, then the $\geq$-universality of nondeterministic discounted-sum word automata is decidable.*

**QLAs with Probabilistic QWAs.** When the underlying word automaton is probabilistic, i.e., has the word aggregator $g = \mathbb{E}$, the nonprobabilistic evaluation problem has no algorithmic solution due to inapproximability of their top values [31].

▶ **Theorem 5.8.** *Consider a QLA* $\mathbb{A} = (h, (\mathbb{E}, f, \mathcal{T}))$ *with* $f \in \{\mathsf{Inf}, \mathsf{Sup}, \mathsf{LimInf}, \mathsf{LimSup}, \mathsf{LimInfAvg}, \mathsf{LimSupAvg}, \mathsf{DSum}\}$ *and* $h \in \{\mathsf{Inf}, \mathsf{Sup}\}$*. Let* $\mu$ *be a probability measure given by a finite-state Markov chain. The value* $\mathbb{A}(\mu)$ *is not computable.*

## 5.2 Probabilistic Evaluation

Now, we consider the evaluation problem for QLAs with $h = \mathbb{E}$ and follow the same structure as in Section 5.1: we start with the cases of $g \in \{\mathsf{Inf}, \mathsf{Sup}\}$ and study various run aggregators $f$ separately, and then look at the case of $g = \mathbb{E}$.

**QLAs with Standard QWAs.** First, we provide an ExpTime algorithm for QLAs with the run aggregators $f \in \{\mathsf{Inf}, \mathsf{Sup}, \mathsf{LimInf}, \mathsf{LimSup}\}$. Our proof builds on that of [32, Thm. 8], which considers only $f \in \{\mathsf{Inf}, \mathsf{Sup}\}$. The idea is to determinize the underlying word automaton, which leads to an exponential blow-up, and evaluate its product with the given Markov chain.

▶ **Theorem 5.9.** *Consider a QLA* $\mathbb{A} = (\mathbb{E}, (g, f, \mathcal{T}))$ *with* $f \in \{\mathsf{Inf}, \mathsf{Sup}, \mathsf{LimInf}, \mathsf{LimSup}\}$ *and* $g \in \{\mathsf{Inf}, \mathsf{Sup}\}$*. Let* $\mu$ *be a probability measure given by a finite-state Markov chain. The value* $\mathbb{A}(\mu)$ *is computable in* ExpTime*.*

**QLAs with Limit-Average QWAs.** Undecidability of probabilistic evaluation for limit-average QLAs was shown in [32] by a reduction from the universality problem of quantitative automata on finite words with the summation run aggregator and weights in $\{-1, 0, 1\}$, a.k.a. weighted automata over the tropical semiring of integers [1, 27].

▶ **Theorem 5.10.** **[32, Thm. 7].** *Consider a QLA* $\mathbb{A} = (\mathbb{E}, (g, f, \mathcal{T}))$ *with* $f \in \{\mathsf{LimInfAvg}, \mathsf{LimSupAvg}\}$ *and* $g \in \{\mathsf{Inf}, \mathsf{Sup}\}$*. Let* $\mu$ *be a probability measure given by a finite-state Markov chain. The value* $\mathbb{A}(\mu)$ *is not computable.*

**QLAs with Discounted-Sum QWAs.** Next, we show the hardness of probabilistic evaluation for DSum QLAs. As in the nonprobabilistic case, we provide a reduction from the universality problem of nondeterministic discounted-sum QWAs.

▶ **Theorem 5.11.** *Consider a QLA* $\mathbb{A} = (\mathbb{E}, (g, \mathsf{DSum}, \mathcal{T}))$ *with* $g \in \{\mathsf{Inf}, \mathsf{Sup}\}$*. Let* $\mu$ *be a probability measure given by a finite-state Markov chain. If* $\mathbb{A}(\mu)$ *is computable, then the* $\geq$*-universality of nondeterministic discounted-sum word automata is decidable.*

**QLAs with Probabilistic QWAs.** Finally, for QLAs with $h = g = \mathbb{E}$, the evaluation problem reduces to evaluating a Markov chain with rewards on infinite words, which is solved using linear programming [25].

▶ **Theorem 5.12.** **[25].** *Consider a QLA* $\mathbb{A} = (\mathbb{E}, (\mathbb{E}, f, \mathcal{T}))$ *with* $f \in \{\mathsf{Inf}, \mathsf{Sup}, \mathsf{LimInf}, \mathsf{LimSup}, \mathsf{LimInfAvg}, \mathsf{LimSupAvg}, \mathsf{DSum}\}$*. Let* $\mu$ *be a probability measure given by a finite-state Markov chain. The value* $\mathbb{A}(\mu)$ *is computable in* PTime*.*

# 6 Deciding Nonemptiness and Universality

In this section, we investigate the nonemptiness and universality problems for QLAs with language and word aggregators $h, g \in \{\mathsf{Inf}, \mathsf{Sup}, \mathbb{E}\}$. We give a complete picture of decidability results for the unrestricted cases. When the unrestricted cases are decidable, our algorithms provide a solution for the finite-state cases in the same complexity class. When they are undecidable or not known to be decidable, some finite-state cases remain open, which we make explicit in the corresponding statements.

Thanks to the duality between nondeterministic and universal automata, solving the nonemptiness problem solves the universality problem as well.

▶ Remark 6.1. Consider a QLA $\mathbb{A} = (h, \mathcal{A})$ and its dual $\hat{\mathbb{A}}$. Let $k \in \mathbb{Q}$. Thanks to Proposition 2.2, the QLA $\mathbb{A}$ is $\geq$-nonempty (resp. $\geq$-universal) for $k$ iff $\hat{\mathbb{A}}$ is not $>$-universal (resp. $>$-nonempty) for $-k$. The statement holds also for the finite-state restriction.

The rest of the section focuses on the nonemptiness problem and is organized by the type of the underlying QWAs. We first consider the case of nondeterministic automata (i.e., $g = \mathsf{Sup}$) and show that the problem is decidable in PTIME for these. Then, we consider universal automata (i.e., $g = \mathsf{Inf}$) with various run aggregators $f$ separately. Similarly as for the evaluation problem, nonemptiness is in PSPACE for the "standard" run aggregators, undecidable for limit-average, and at least as hard as a long-standing open problem for discounted sum. Finally, we consider probabilistic automata (i.e., $g = \mathbb{E}$) and show that the problem is undecidable.

**QLAs with Nondeterministic QWAs.** Let us first consider QLAs whose underlying word automata are nondeterministic, i.e., those where the word aggregator is $g = \mathsf{Sup}$. We show that the nonemptiness problems for such QLAs can be solved efficiently, independently of the choice of the remaining aggregators or additional restrictions on the problem. Intuitively, this is because their top values coincide with those of the underlying QWAs (Proposition 2.1), which are achievable by lasso words and can be computed efficiently.

▶ **Theorem 6.2.** *Consider a QLA $\mathbb{A} = (h, (\mathsf{Sup}, f, \mathcal{T}))$ with $f \in \{\mathsf{Inf}, \mathsf{Sup}, \mathsf{LimInf}, \mathsf{LimSup}, \mathsf{LimInfAvg}, \mathsf{LimSupAvg}, \mathsf{DSum}\}$ and $h \in \{\mathsf{Inf}, \mathsf{Sup}, \mathbb{E}\}$. Let $\triangleright \in \{>, \geq\}$. The $\triangleright$-nonemptiness of $\mathbb{A}$ is in PTIME. The statement holds also for the finite-state restriction.*

**QLAs with Universal Standard QWAs.** We turn our attention to QLAs whose underlying word automata are universal, i.e., those where the word aggregator is $g = \mathsf{Inf}$. For run aggregators $f \in \{\mathsf{Inf}, \mathsf{Sup}, \mathsf{LimInf}, \mathsf{LimSup}\}$, we show that computing the automaton's top value suffices, which can be done in PSPACE when $g = \mathsf{Inf}$.

▶ **Theorem 6.3.** *Consider a QLA $\mathbb{A} = (h, (\mathsf{Inf}, f, \mathcal{T}))$ with $f \in \{\mathsf{Inf}, \mathsf{Sup}, \mathsf{LimInf}, \mathsf{LimSup}\}$ and $h \in \{\mathsf{Inf}, \mathsf{Sup}, \mathbb{E}\}$. Let $\triangleright \in \{>, \geq\}$. The $\triangleright$-nonemptiness of $\mathbb{A}$ is in PSPACE. The statement holds also for the finite-state restriction.*

**QLAs with Universal Limit-Average QWAs.** Next, we focus on QLAs with the limit-average run aggregators. We first show that the bottom value of nondeterministic $\mathsf{LimSupAvg}$ QWAs (dually, top value of universal $\mathsf{LimInfAvg}$ QWAs) can be approximated arbitrarily closely by lasso words. This is in stark contrast with nondeterministic $\mathsf{LimInfAvg}$ QWAs, where an automaton may be lasso-word universal for some threshold $k$ while there exist non-lasso words with values below $k$ [15, Lem. 4]. The intuition behind this is that for $\mathsf{LimSupAvg}$, if $\mathcal{A}(w) = k$ for some word $w$ and value $k$, then for any $\varepsilon > 0$, there exists a length $\ell$ such that the average value of any run prefix on $w$ of length at least $\ell$ is below $k + \varepsilon$, which does not hold in general for $\mathsf{LimInfAvg}$. Consequently, we can find two distant-enough occurrences of the same state and pump the low-average segment between them to obtain a lasso whose overall value stays under the threshold.

▶ **Proposition 6.4.** *Consider a QWA $\mathcal{A} = (\mathsf{Sup}, \mathsf{LimSupAvg}, \mathcal{T})$. For every $\varepsilon > 0$ there is a lasso word $w$ such that $\mathcal{A}(w) < \bot_{\mathcal{A}} + \varepsilon$.*

In general, all variants of the nonemptiness problem for universal limit-average QWAs (dually, universality for the nondeterministic ones) are undecidable [12, 17, 24]. Combining these with our approximability result above, we obtain the following.

▶ **Theorem 6.5.** *Consider a QLA* $\mathbb{A} = (h, (\mathsf{Inf}, f, \mathcal{T}))$ *with* $f \in \{\mathsf{LimInfAvg}, \mathsf{LimSupAvg}\}$ *and* $h \in \{\mathsf{Inf}, \mathsf{Sup}, \mathbb{E}\}$*. Let* $\triangleright \in \{>, \geq\}$*. The* $\triangleright$*-nonemptiness of* $\mathbb{A}$ *is undecidable. The statement holds also for the finite-state restriction when (i)* $h \neq \mathbb{E}$ *or (ii)* $f = \mathsf{LimInfAvg}$ *and* $\triangleright = >$*.*

**QLAs with Universal Discounted-Sum QWAs.** Now, we consider QLAs with the discounted-sum run aggregators. We show that the bottom value of nondeterministic DSum QWAs (dually, top value of universal DSum QWAs) is achievable. At a high level, we repeatedly extend a finite prefix with a letter that does not increase the current infimum over continuation values. In the limit, this process results in an infinite word whose sequence of prefix infima converges to the automaton's bottom value. Since discounted-sum automata are co-safe in the quantitative sense [6, 7, 21], i.e., the value of every infinite word equals the limit of its prefix infima, this constructed word attains exactly the bottom value.

▶ **Proposition 6.6.** *Consider a QWA* $\mathcal{A} = (\mathsf{Sup}, \mathsf{DSum}, \mathcal{T})$*. There is a word* $w$ *such that* $\mathcal{A}(w) = \bot_{\mathcal{A}}$*.*

By combining the achievability result above with Proposition 4.2, we obtain a concise proof to establish that the nonemptiness problem for QLAs whose underlying QWAs are universal DSum QWAs is at least as hard as the universality problem for nondeterministic DSum QWAs, which is a long-standing open problem [8, 15].

▶ **Theorem 6.7.** *Consider a QLA* $\mathbb{A} = (h, (\mathsf{Inf}, \mathsf{DSum}, \mathcal{T}))$ *with* $h \in \{\mathsf{Inf}, \mathsf{Sup}, \mathbb{E}\}$*. Let* $\triangleright \in \{>, \geq\}$*. If the* $\triangleright$*-nonemptiness of* $\mathbb{A}$ *is decidable, then the universality of nondeterministic discounted-sum word automata is decidable. The statement holds also for the finite-state restriction when (i)* $h = \mathsf{Sup}$ *or (ii)* $\triangleright = >$*.*

**QLAs with Probabilistic QWAs.** Let us now focus on QLAs whose underlying QWA is probabilistic, i.e., where $g = \mathbb{E}$. We show that several variants of the nonemptiness problem for probabilistic QWAs are undecidable by building on the undecidability results of probabilistic word automata [31].

▶ **Theorem 6.8.** *Consider a QWA* $\mathcal{A} = (\mathbb{E}, f, \mathcal{T})$ *with* $f \in \{\mathsf{Inf}, \mathsf{Sup}, \mathsf{LimInf}, \mathsf{LimSup}, \mathsf{LimInfAvg}, \mathsf{LimSupAvg}, \mathsf{DSum}\}$*. Let* $\triangleright \in \{>, \geq\}$*. The* $\triangleright$*-nonemptiness of* $\mathcal{A}$ *is undecidable, whether we consider all words from* $\Sigma^\omega$ *or only lasso words. Moreover, its approximate-nonemptiness is also undecidable.*

Finally, putting together the above undecidability result with Propositions 4.2 and 4.3, we show the undecidability of the nonemptiness problem for QLAs whose underlying QWA is probabilistic.

▶ **Theorem 6.9.** *Consider a QLA* $\mathbb{A} = (h, (\mathbb{E}, f, \mathcal{T}))$ *with* $f \in \{\mathsf{Inf}, \mathsf{Sup}, \mathsf{LimInf}, \mathsf{LimSup}, \mathsf{LimInfAvg}, \mathsf{LimSupAvg}, \mathsf{DSum}\}$ *and* $h \in \{\mathsf{Inf}, \mathsf{Sup}, \mathbb{E}\}$*. Let* $\triangleright \in \{>, \geq\}$*. The* $\triangleright$*-nonemptiness of* $\mathbb{A}$ *is undecidable. The statement holds also for the finite-state restriction when (i)* $h \neq \mathbb{E}$ *or (ii)* $f = \mathsf{DSum}$ *and* $\triangleright = >$*.*

## 7 Language Automata with Limit Aggregators

We finally focus on QLAs with the aggregators $h, g, f \in \{\mathsf{Inf}, \mathsf{Sup}, \mathsf{LimInf}, \mathsf{LimSup}\}$ where at least one of the language aggregator $h$ or the word aggregator $g$ belongs to $\{\mathsf{LimInf}, \mathsf{LimSup}\}$.

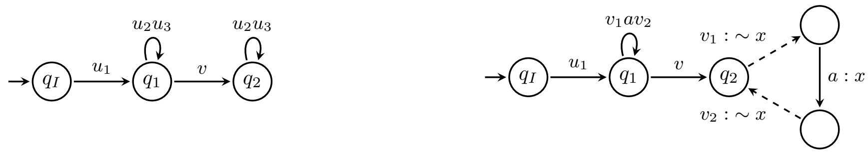


**Figure 2** **(a)** Pattern parameterized by $q_2$ and $n$ where $u_1, u_2, u_3, v \in \Sigma^+$ are such that $|u_2| = |v|$, $u_2 \neq v$, and $m|v| = |u_2u_3|$ for some $m \in \{1, \dots, 4n^4\}$. **(b)** Pattern parameterized by $x \in \mathbb{Q}$, $n \in \mathbb{N}$, $\sim \in \{\leq, \geq\}$, and $w \in \Sigma^\omega$, where $u_1, v, v_1, v_2 \in \Sigma^*$ and $a \in \Sigma$ are such that $q_1 \neq q_2$, $w = u(v_1av_2)^\omega$, and $(v)^m = v_1av_2$ for some $m \in \{1, \dots, 4n^4\}$. Dashed runs visit weights $y \in \mathbb{Q}$ satisfying $y \sim x$.

## 7.1 Deciding Infinite Achievability

To work with QLAs that use LimInf or LimSup as word or language aggregators, we need to reason about the values that appear infinitely often in a multiset. In this subsection, we develop the necessary tools for this purpose, focusing on QLAs with run aggregators $f \in \{\mathsf{Inf}, \mathsf{Sup}, \mathsf{LimInf}, \mathsf{LimSup}\}$.

We begin by characterizing the conditions under which a state in an automaton can be visited infinitely often by infinitely many words. The following lemma uses proof techniques similar to those used in the characterization of ambiguity in finite-state automata [29, 35].

▶ **Lemma 7.1.** *Let $\mathcal{T}$ be a (weighted) labeled transition system with $n$ states and $q_I$ its initial state. For every state $q_2$ of $\mathcal{T}$, there exist infinitely many words with a run from $q_I$ visiting $q_2$ infinitely often iff $\mathcal{T}$ complies with the pattern parameterized by $q_2$ and $n$ given in Figure 2a.*

It is well known that Büchi automata can be effectively complemented in ExpTime [20, 34]. Moreover, checking whether a finite-state automaton satisfies a given syntactic pattern, such as the one in Figure 2a, is in NLogSpace [18]. As a direct consequence of Lemma 7.1, keeping a symbolic representation of the complement provides the following PSpace procedure.

▶ **Corollary 7.2.** *Let $\mathcal{B}_1$ and $\mathcal{B}_2$ be two Büchi automata. We can decide in PSpace whether $L(\mathcal{B}_1) \setminus L(\mathcal{B}_2)$ is infinite.*

By interpreting the transitions of a run as input letters, we can use Figure 2a to reason about runs instead of words. Building on Lemma 7.1, we present a similar pattern that incorporates weights, enabling us to identify the words for which an automaton admits infinitely many runs of a given value.

▶ **Lemma 7.3.** *Consider a QWA $\mathcal{A} = (g, f, \mathcal{T})$ with $n$ states, $f \in \{\mathsf{LimInf}, \mathsf{LimSup}\}$, $g \in \{\mathsf{Inf}, \mathsf{Sup}, \mathsf{LimInf}, \mathsf{LimSup}\}$, and initial state $q_I$. Let $\sim = \leq$ if $f = \mathsf{LimSup}$ and $\sim = \geq$ if $f = \mathsf{LimInf}$. For every $x \in \mathbb{Q}$ and every lasso word $w \in \Sigma^\omega$, we have that $w$ admits infinitely many runs of value $x$ in $\mathcal{A}$ iff $\mathcal{A}$ complies with the pattern in Figure 2b parameterized by $x$, $n$, $\sim$, and $w$.*

As a corollary of Lemma 7.3, we can construct a boolean automaton accepting every lasso word with infinitely many runs of value $x$ in $\mathcal{A}$. To achieve this, we construct a Büchi or co-Büchi automaton that initially guesses a fixed pair of states $q_1$ and $q_2$ with the same properties as in Figure 2b. The automaton then guesses on-the-fly the segments $u_1$ and $v$ from Figure 2b. During the processing of each period $v$, it simulates $3 \times m$ runs, with $m \in \{1, \dots, 4n^4\}$, simultaneously verifying weights and connectivity. At the end of each period, the automaton resets (except for the fixed $q_1$ and $q_2$) to guess another period and visit an accepting state. Consequently, any accepted lasso word resets infinitely often and ultimately repeats a finite periodic segment $v$, complying with Figure 2b. Conversely, every word conforming to Figure 2b is accepted by appropriately selecting $v = v_1av_2$.

▶ **Corollary 7.4.** *Let $\mathcal{A} = (\mathsf{Sup}, f, \mathcal{T})$ be QWA with $f = \mathsf{LimSup}$ (resp. $f = \mathsf{LimInf}$). Given $x \in \mathbb{Q}$, we can construct in PTime a Büchi automaton (resp. a co-Büchi automaton) recognizing all lasso words admitting infinitely many runs of value $x$ in $\mathcal{A}$.*

## 7.2 Expressive Power of Limit Aggregators

This subsection investigates the expressive power of word and language aggregators $g, h \in \{\mathsf{Inf}, \mathsf{Sup}, \mathsf{LimInf}, \mathsf{LimSup}\}$. The following proposition generalizes the results of [6,15] to QLAs.

▶ **Proposition 7.5.** *Consider a QLA $\mathbb{A} = (h, (g, f, \mathcal{T}))$ with $f \in \{\mathsf{Inf}, \mathsf{Sup}\}$, $g \in \{\mathsf{Inf}, \mathsf{Sup}, \mathsf{LimInf}, \mathsf{LimSup}\}$, and $h$ any language aggregator function. We can construct in PTime a QLA $\mathbb{B} = (h, (g, f', \mathcal{T}'))$ with $f' \in \{\mathsf{LimInf}, \mathsf{LimSup}\}$ such that $\mathbb{A}(S) = \mathbb{B}(S)$ for all $S \subseteq \Sigma^\omega$.*

▶ Remark 7.6. The construction of [15] allowing to convert a nondeterministic QWA with run aggregator $\mathsf{LimInf}$ to one with $\mathsf{LimSup}$ can be extended to QLAs with word aggregator $\mathsf{Sup}$ but not with $\mathsf{LimSup}$. By duality, the conversion of a universal $\mathsf{LimSup}$ QWA to a $\mathsf{LimInf}$ one can be extended to QLAs with word aggregators $\mathsf{Inf}$ but not $\mathsf{LimInf}$. This is because deterministic $\mathsf{LimInf}$ QWA and deterministic $\mathsf{LimSup}$ QWA are expressively incomparable [15]. Hence, there is no conversion that preserves, for every word $w$, the number runs with the same value over $w$.

We now focus on word aggregators, first showing that $\mathsf{Inf}$ and $\mathsf{Sup}$ are at least as expressive as $\mathsf{LimInf}$ and $\mathsf{LimSup}$ when they are combined with the run aggregators $f \in \{\mathsf{Inf}, \mathsf{Sup}, \mathsf{LimInf}, \mathsf{LimSup}\}$. The construction relies on Corollary 7.4, the closure of QWAs under max and min operation [14], and an argument that equivalence of QWAs with run and word aggregators in $\{\mathsf{Inf}, \mathsf{Sup}, \mathsf{LimInf}, \mathsf{LimSup}\}$ can be determined by considering only lasso words.

▶ **Lemma 7.7.** *Let $\mathbb{A} = (h, (g, f, \mathcal{T}))$ be a QLA with $f \in \{\mathsf{Inf}, \mathsf{Sup}, \mathsf{LimInf}, \mathsf{LimSup}\}$, $g = \mathsf{LimSup}$ (resp. $g = \mathsf{LimInf}$), and $h$ any language aggregator function. We can construct in PSpace a QLA $\mathbb{B} = (h, (g', f', \mathcal{T}'))$ with $f' = \mathsf{LimSup}$ and $g' = \mathsf{Sup}$ (resp. $f' = \mathsf{LimInf}$ and $g' = \mathsf{Inf}$) such that $\mathbb{A}(S) = \mathbb{B}(S)$ for all $S \subseteq \Sigma^\omega$.*

Now, we prove the other direction: the word aggregators $\mathsf{LimInf}$ and $\mathsf{LimSup}$ are at least as expressive as $\mathsf{Inf}$ and $\mathsf{Sup}$ for any run aggregator and any language aggregator. The idea is to duplicate the transition system so that the automaton can switch between the two copies at any step, ensuring that every word yields infinitely many runs with the same value.

▶ **Lemma 7.8.** *Let $\mathbb{A} = (h, (g, f, \mathcal{T}))$ be a QLA with $f$ any run aggregator function, $g = \mathsf{Sup}$ (resp. $g = \mathsf{Inf}$), and $h$ any language aggregator function. We can construct in PTime a QLA $\mathbb{B} = (h, (g', f, \mathcal{T}')$ where $g' = \mathsf{LimSup}$ (resp. $g' = \mathsf{LimInf}$) such that $\mathbb{A}(S) = \mathbb{B}(S)$ for all $S \subseteq \Sigma^\omega$.*

The following comes as a consequence of Lemmas 7.7 and 7.8.

▶ **Theorem 7.9.** *For QLAs with run aggregators $f \in \{\mathsf{Inf}, \mathsf{Sup}, \mathsf{LimInf}, \mathsf{LimSup}\}$, the word aggregators $\mathsf{Inf}$ and $\mathsf{LimInf}$ (resp. $\mathsf{Sup}$ and $\mathsf{LimSup}$) are equally expressive.*

Finally, we turn our attention to language aggregators. In contrast to the case of word aggregators, even when combined with run aggregators $f \in \{\mathsf{Inf}, \mathsf{Sup}, \mathsf{LimInf}, \mathsf{LimSup}\}$, the language aggregators $\mathsf{Inf}$ and $\mathsf{Sup}$ are expressively incomparable with $\mathsf{LimInf}$ and $\mathsf{LimSup}$. Intuitively, when the top or the bottom value of the underlying QWA is achievable by a single word, a QLA with a limit language aggregator cannot achieve this value, while one with a non-limit language aggregator can. Conversely, if the extreme value of a QLA emerges only as the value of finite languages, a limit language aggregator can capture this behavior, while a non-limit aggregator cannot.

▶ **Proposition 7.10.** *QLAs with the language aggregators* Inf *and* LimInf *(resp.* Sup *and* LimSup*) are expressively incomparable.*

## 7.3 Decision Problems with Limit Aggregators

Finally, we consider the evaluation, nonemptiness, and universality problems for these classes of QLAs. We first provide a PSpace procedure for evaluation. Intuitively, each weight appearing on the underlying transition system are treated individually since the value of an input language must be one of them. For $h \in \{\mathsf{LimSup}, \mathsf{LimInf}\}$, we leverage Corollary 7.2 to identify the weights achievable by infinitely many words from the input languages in PSpace. The problem reduces to checking the nonemptiness of a Büchi automaton when $h \in \{\mathsf{LimSup}, \mathsf{Sup}\}$ and to checking their inclusion when $h \in \{\mathsf{LimInf}, \mathsf{Inf}\}$. We note that the proof of Theorem 7.11 also establishes the PTime cases presented in Theorem 5.1.

▶ **Theorem 7.11.** *Consider a QLA $\mathbb{A} = (h, (g, f, \mathcal{T}))$ with $f, g, h \in \{\mathsf{Inf}, \mathsf{Sup}, \mathsf{LimInf}, \mathsf{LimSup}\}$ and at least one $g$ and $h$ belong to $\{\mathsf{LimInf}, \mathsf{LimSup}\}$. Let $S \subseteq \Sigma^\omega$ be an $\omega$-regular language given by a Büchi automaton. The value $\mathbb{A}(S)$ is computable in PSpace.*

Next, we prove that QLAs with $f, g, h \in \{\mathsf{Inf}, \mathsf{Sup}, \mathsf{LimInf}, \mathsf{LimSup}\}$, checking whether a given QLA is an upper bound on another (namely, their inclusion problem) is decidable in PSpace. The proof starts by showing that this problem can be solved while reasoning exclusively on $\omega$-regular languages and then generalizes our algorithm for the evaluation problem (Theorem 5.1) to handle two QLAs.

▶ **Theorem 7.12.** *Consider two QLAs $\mathbb{A} = (h, (g, f, \mathcal{T}))$ and $\mathbb{B} = (h', (g', f', \mathcal{T}'))$ with $f, f', g, g', h, h' \in \{\mathsf{Inf}, \mathsf{Sup}, \mathsf{LimInf}, \mathsf{LimSup}\}$. Let $\triangleright \in \{>, \geq\}$. Deciding whether $\mathbb{A}(S) \triangleright \mathbb{B}(S)$ for every language $S \subseteq \Sigma^\omega$ is in PSpace. The same holds when $S$ ranges over $\omega$-regular languages.*

Using Theorem 7.12, we obtain a matching solution to the corresponding nonemptiness and universality problems. Note that Theorems 6.2 and 6.3 capture the cases of non-limit aggregators, including the PTime fragments.

▶ **Corollary 7.13.** *Consider a QLA $\mathbb{A} = (h, (g, f, \mathcal{T}))$ with $f, g, h \in \{\mathsf{Inf}, \mathsf{Sup}, \mathsf{LimInf}, \mathsf{LimSup}\}$ and at least one $g$ and $h$ belong to $\{\mathsf{LimInf}, \mathsf{LimSup}\}$. Let $\triangleright \in \{>, \geq\}$. The $\triangleright$-nonemptiness (resp. $\triangleright$-universality) of $\mathbb{A}$ is in PSpace. The statement holds also for the finite-state restriction.*

## 8 Conclusion

We introduced quantitative language automata (QLAs) as a uniform framework for specifying and verifying quantitative hyperproperties. Our framework extends beyond both the traditional boolean perspective of system properties and the "one-trace limitation" of traditional quantitative properties as system specifications, enabling reasoning about quantitative aspects of system-wide behaviors such as performance and robustness. We established a thorough foundation for QLAs by investigating the evaluation, nonemptiness, and universality problems, for which we provided a extensive picture of decidability results. In addition to closing the finite-state cases we left open, future research directions include exploring aggregators capable of specifying richer relational system properties, investigating decidable expressive fragments of QLAs, studying equivalent logical formalisms, and augmenting the software tool Quantitative Automata Kit (QuAK) [10, 11] with a support for QLAs.

## References


1 Shaull Almagor, Udi Boker, and Orna Kupferman. What's decidable about weighted automata? *Inf. Comput.*, 282:104651, 2022. URL: https://doi.org/10.1016/j.ic.2020.104651, doi:10.1016/J.IC.2020.104651.

2 Bowen Alpern and Fred B. Schneider. Defining liveness. *Inf. Process. Lett.*, 21(4):181–185, 1985. doi:10.1016/0020-0190(85)90056-0.

3 Bowen Alpern and Fred B. Schneider. Recognizing safety and liveness. *Distributed Comput.*, 2(3):117–126, 1987. doi:10.1007/BF01782772.

4 Daniel Andersson. An improved algorithm for discounted payoff games. In Janneke Huitink and Sophia Katrenko, editors, *Proceedings of the 11th ESSLLI Student Session*, pages 91–98, August 2006.

5 Christel Baier and Joost-Pieter Katoen. *Principles of model checking*. MIT Press, 2008.

6 Udi Boker, Thomas A. Henzinger, Nicolas Mazzocchi, and N. Ege Saraç. Safety and liveness of quantitative automata. In Guillermo A. Pérez and Jean-François Raskin, editors, *34th International Conference on Concurrency Theory, CONCUR 2023, September 18-23, 2023, Antwerp, Belgium*, volume 279 of *LIPIcs*, pages 17:1–17:18. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2023. URL: https://doi.org/10.4230/LIPIcs.CONCUR.2023.17, doi:10.4230/LIPICS.CONCUR.2023.17.

7 Udi Boker, Thomas A Henzinger, Nicolas Mazzocchi, and N Ege Saraç. Safety and liveness of quantitative properties and automata. *Logical Methods in Computer Science*, 21, 2025.

8 Udi Boker, Thomas A. Henzinger, and Jan Otop. The target discounted-sum problem. In *30th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2015, Kyoto, Japan, July 6-10, 2015*, pages 750–761. IEEE Computer Society, 2015. doi:10.1109/LICS.2015.74.

9 Udi Boker and Karoliina Lehtinen. History determinism vs. good for gameness in quantitative automata. In Mikolaj Bojanczyk and Chandra Chekuri, editors, *41st IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS 2021, December 15-17, 2021, Virtual Conference*, volume 213 of *LIPIcs*, pages 38:1–38:20. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021. URL: https://doi.org/10.4230/LIPIcs.FSTTCS.2021.38, doi:10.4230/LIPICS.FSTTCS.2021.38.

10 Marek Chalupa, Thomas A. Henzinger, Nicolas Mazzocchi, and N. Ege Saraç. Quak: Quantitative automata kit. In Tiziana Margaria and Bernhard Steffen, editors, *Leveraging Applications of Formal Methods, Verification and Validation. Software Engineering Methodologies - 12th International Symposium, ISoLA 2024, Crete, Greece, October 27-31, 2024, Proceedings, Part IV*, volume 15222 of *Lecture Notes in Computer Science*, pages 3–20. Springer, 2024. doi:10.1007/978-3-031-75387-9_1.

11 Marek Chalupa, Thomas A. Henzinger, Nicolas Mazzocchi, and N. Ege Saraç. Automating the analysis of quantitative automata with quak. In Arie Gurfinkel and Marijn Heule, editors, *Tools and Algorithms for the Construction and Analysis of Systems - 31st International Conference, TACAS 2025, Held as Part of the International Joint Conferences on Theory and Practice of Software, ETAPS 2025, Hamilton, ON, Canada, May 3-8, 2025, Proceedings, Part I*, volume 15696 of *Lecture Notes in Computer Science*, pages 303–312. Springer, 2025. doi:10.1007/978-3-031-90643-5_16.

12 Krishnendu Chatterjee, Laurent Doyen, Herbert Edelsbrunner, Thomas A. Henzinger, and Philippe Rannou. Mean-payoff automaton expressions. In Paul Gastin and François Laroussinie, editors, *CONCUR 2010 - Concurrency Theory, 21th International Conference, CONCUR 2010, Paris, France, August 31-September 3, 2010. Proceedings*, volume 6269 of *Lecture Notes in Computer Science*, pages 269–283. Springer, 2010. doi:10.1007/978-3-642-15375-4_19.

13 Krishnendu Chatterjee, Laurent Doyen, and Thomas A. Henzinger. Alternating weighted automata. In Miroslaw Kutylowski, Witold Charatonik, and Maciej Gebala, editors, *Fundamentals of Computation Theory, 17th International Symposium, FCT 2009, Wroclaw, Poland, September 2-4, 2009. Proceedings*, volume 5699 of *Lecture Notes in Computer Science*, pages 3–13. Springer, 2009. doi:10.1007/978-3-642-03409-1_2.

**14** Krishnendu Chatterjee, Laurent Doyen, and Thomas A. Henzinger. Expressiveness and closure properties for quantitative languages. *Log. Methods Comput. Sci.*, 6(3), 2010. URL: http://arxiv.org/abs/1007.4018.

**15** Krishnendu Chatterjee, Laurent Doyen, and Thomas A. Henzinger. Quantitative languages. *ACM Trans. Comput. Log.*, 11(4):23:1–23:38, 2010. doi:10.1145/1805950.1805953.

**16** Michael R. Clarkson and Fred B. Schneider. Hyperproperties. *J. Comput. Secur.*, 18(6):1157–1210, 2010. doi:10.3233/JCS-2009-0393.

**17** Aldric Degorre, Laurent Doyen, Raffaella Gentilini, Jean-François Raskin, and Szymon Torunczyk. Energy and mean-payoff games with imperfect information. In Anuj Dawar and Helmut Veith, editors, *Computer Science Logic, 24th International Workshop, CSL 2010, 19th Annual Conference of the EACSL, Brno, Czech Republic, August 23-27, 2010. Proceedings*, volume 6247 of *Lecture Notes in Computer Science*, pages 260–274. Springer, 2010. doi:10.1007/978-3-642-15205-4_22.

**18** Emmanuel Filiot, Nicolas Mazzocchi, and Jean-François Raskin. A pattern logic for automata with outputs. In Mizuho Hoshi and Shinnosuke Seki, editors, *Developments in Language Theory - 22nd International Conference, DLT 2018, Tokyo, Japan, September 10-14, 2018, Proceedings*, volume 11088 of *Lecture Notes in Computer Science*, pages 304–317. Springer, 2018. doi:10.1007/978-3-319-98654-8_25.

**19** Bernd Finkbeiner, Christopher Hahn, and Hazem Torfah. Model checking quantitative hyperproperties. In Hana Chockler and Georg Weissenbacher, editors, *Computer Aided Verification - 30th International Conference, CAV 2018, Held as Part of the Federated Logic Conference, FloC 2018, Oxford, UK, July 14-17, 2018, Proceedings, Part I*, volume 10981 of *Lecture Notes in Computer Science*, pages 144–163. Springer, 2018. doi:10.1007/978-3-319-96145-3_8.

**20** Vojtech Havlena, Ondrej Lengál, and Barbora Smahlíková. Complementing Büchi automata with ranker. In Sharon Shoham and Yakir Vizel, editors, *Computer Aided Verification - 34th International Conference, CAV 2022, Haifa, Israel, August 7-10, 2022, Proceedings, Part II*, volume 13372 of *Lecture Notes in Computer Science*, pages 188–201. Springer, 2022. doi:10.1007/978-3-031-13188-2_10.

**21** Thomas A. Henzinger, Nicolas Mazzocchi, and N. Ege Saraç. Quantitative safety and liveness. In Orna Kupferman and Pawel Sobocinski, editors, *Foundations of Software Science and Computation Structures - 26th International Conference, FoSSaCS 2023, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2023, Paris, France, April 22-27, 2023, Proceedings*, volume 13992 of *Lecture Notes in Computer Science*, pages 349–370. Springer, 2023. doi:10.1007/978-3-031-30829-1_17.

**22** Thomas A. Henzinger, Nicolas Mazzocchi, and N. Ege Saraç. Strategic dominance: A new preorder for nondeterministic processes. In Rupak Majumdar and Alexandra Silva, editors, *35th International Conference on Concurrency Theory, CONCUR 2024, September 9-13, 2024, Calgary, Canada*, volume 311 of *LIPIcs*, pages 29:1–29:20. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2024. URL: https://doi.org/10.4230/LIPIcs.CONCUR.2024.29, doi:10.4230/LIPICS.CONCUR.2024.29.

**23** Thomas A. Henzinger and Jan Otop. From model checking to model measuring. In Pedro R. D'Argenio and Hernán C. Melgratti, editors, *CONCUR 2013 - Concurrency Theory - 24th International Conference, CONCUR 2013, Buenos Aires, Argentina, August 27-30, 2013. Proceedings*, volume 8052 of *Lecture Notes in Computer Science*, pages 273–287. Springer, 2013. doi:10.1007/978-3-642-40184-8_20.

**24** Paul Hunter, Arno Pauly, Guillermo A. Pérez, and Jean-François Raskin. Mean-payoff games with partial observation. *Theor. Comput. Sci.*, 735:82–110, 2018. URL: https://doi.org/10.1016/j.tcs.2017.03.038, doi:10.1016/J.TCS.2017.03.038.

**25** Narendra Karmarkar. A new polynomial-time algorithm for linear programming. *Comb.*, 4(4):373–396, 1984. doi:10.1007/BF02579150.

**26** Richard M. Karp. A characterization of the minimum cycle mean in a digraph. *Discret. Math.*, 23(3):309–311, 1978. doi:10.1016/0012-365X(78)90011-0.

27 Daniel Krob. The equality problem for rational series with multiplicities in the tropical semiring is undecidable. *Int. J. Algebra Comput.*, 4(3):405–426, 1994. doi:10.1142/S0218196794000063.

28 Leslie Lamport. Proving the correctness of multiprocess programs. *IEEE Trans. Software Eng.*, 3(2):125–143, 1977. doi:10.1109/TSE.1977.229904.

29 Christof Löding and Anton Pirogov. On finitely ambiguous Büchi automata. In Mizuho Hoshi and Shinnosuke Seki, editors, *Developments in Language Theory - 22nd International Conference, DLT 2018, Tokyo, Japan, September 10-14, 2018, Proceedings*, volume 11088 of *Lecture Notes in Computer Science*, pages 503–515. Springer, 2018. doi:10.1007/978-3-319-98654-8_41.

30 Omid Madani. *Complexity results for infinite-horizon Markov decision processes*. University of Washington, 2000.

31 Omid Madani, Steve Hanks, and Anne Condon. On the undecidability of probabilistic planning and related stochastic optimization problems. *Artif. Intell.*, 147(1-2):5–34, 2003. doi:10.1016/S0004-3702(02)00378-8.

32 Jakub Michaliszyn and Jan Otop. Non-deterministic weighted automata evaluated over Markov chains. *J. Comput. Syst. Sci.*, 108:118–136, 2020. URL: https://doi.org/10.1016/j.jcss.2019.10.001, doi:10.1016/J.JCSS.2019.10.001.

33 Shubham Sahai, Pramod Subramanyan, and Rohit Sinha. Verification of quantitative hyperproperties using trace enumeration relations. In Shuvendu K. Lahiri and Chao Wang, editors, *Computer Aided Verification - 32nd International Conference, CAV 2020, Los Angeles, CA, USA, July 21-24, 2020, Proceedings, Part I*, volume 12224 of *Lecture Notes in Computer Science*, pages 201–224. Springer, 2020. doi:10.1007/978-3-030-53288-8_11.

34 Sven Schewe. Büchi complementation made tight. In Susanne Albers and Jean-Yves Marion, editors, *26th International Symposium on Theoretical Aspects of Computer Science, STACS 2009, February 26-28, 2009, Freiburg, Germany, Proceedings*, volume 3 of *LIPIcs*, pages 661–672. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, Germany, 2009. URL: https://doi.org/10.4230/LIPIcs.STACS.2009.1854, doi:10.4230/LIPICS.STACS.2009.1854.

35 Andreas Weber and Helmut Seidl. On the degree of ambiguity of finite automata. *Theor. Comput. Sci.*, 88(2):325–349, 1991. doi:10.1016/0304-3975(91)90381-B.

36 Linpeng Zhang, Noam Zilberstein, Benjamin Lucien Kaminski, and Alexandra Silva. Quantitative weakest hyper pre: Unifying correctness and incorrectness hyperproperties via predicate transformers. *CoRR*, abs/2404.05097, 2024. URL: https://doi.org/10.48550/arXiv.2404.05097, arXiv:2404.05097, doi:10.48550/ARXIV.2404.05097.

## A Proofs of Section 2

**Proposition 2.1.** Consider a QLA $\mathbb{A} = (h, \mathcal{A})$ with $h \in \{\mathsf{Inf}, \mathsf{Sup}, \mathbb{E}\}$. Then, $\top_{\mathbb{A}} = \top_{\mathcal{A}}$ and $\bot_{\mathbb{A}} = \bot_{\mathcal{A}}$.

**Proof.** Let $\mathbb{A} = (h, \mathcal{A})$ be a language automaton as in the statement. Assume $h = \mathsf{Sup}$. First, observe that for every language $L_1$ and $L_2$ with $L_1 \subseteq L_2$ we have $\mathbb{A}(L_1) \leq \mathbb{A}(L_2)$. We immediately obtain $\top_{\mathbb{A}} = \mathbb{A}(\Sigma^\omega)$. By definition, $\mathbb{A}(\Sigma^\omega) = \sup_{w \in \Sigma^\omega} \mathcal{A}(w) = \top_{\mathcal{A}}$. For the case of bottom values, we have $\bot_{\mathbb{A}} = \inf_{w \in \Sigma^\omega} \mathbb{A}(\{w\})$ by the observation above, which implies that for every word $w$ and language $L$ with $w \in L$ we have $\mathbb{A}(\{w\}) \leq \mathbb{A}(L)$. Then, as $\mathbb{A}(\{w\}) = \mathcal{A}(w)$, we conclude $\bot_{\mathbb{A}} = \bot_{\mathcal{A}}$. The case of $h = \mathsf{Inf}$ is similar.

Now, assume $h = \mathbb{E}$. Consider a sequence of words $w_1, w_2, \ldots$ such that $\lim_{i \to \infty} \mathcal{A}(w_i) = \top_{\mathcal{A}}$. Then, there is a sequence of probabilistic language generators $G_1, G_2, \ldots$ such that for each $i \geq 1$ the measure $\mu_i$ of $G_i$ assign probability 1 to $\{w_i\}$, which implies $\top_{\mathbb{A}} \geq \top_{\mathcal{A}}$. Moreover, thanks to monotonicity of expected value, we have $\top_{\mathbb{A}} \leq \top_{\mathcal{A}}$, which results in $\top_{\mathbb{A}} = \top_{\mathcal{A}}$. The case of bottom values is similar. ◀

**Proposition 2.2.** Consider a QLA $\mathbb{A}$ and its dual $\hat{\mathbb{A}}$. Then, $\mathbb{A}(G) = -\hat{\mathbb{A}}(G)$ for every language generator $G$.

**Proof.** Follows from the fact that $\inf X = -\sup -X$ and $\liminf X = -\limsup -X$. ◀

## B Proofs of Section 4

**Proposition 4.1.** Consider a QLA $\mathbb{A} = (h, \mathcal{A})$ with $h \in \{\mathsf{Inf}, \mathsf{Sup}, \mathsf{LimSup}, \mathsf{LimInf}\}$ (resp. $h = \mathbb{E}$), a rational $k \in \mathbb{Q}$, and $\triangleright \in \{>, \geq\}$. Then, $\mathbb{A}$ is $\triangleright$-nonempty for $k$ iff $\mathbb{A}$ is Borel $\triangleright$-nonempty (resp. Markov $\triangleright$-nonempty) for $k$. The same holds for universality.

**Proof.** We focus on the $\geq$-nonemptiness problem as the strict case is similar, and prove the $\Rightarrow$ implication as the other direction is immediate. (The proof is similar for the $\Leftarrow$ implication of universality.)

Let $h = \mathsf{Sup}$. Suppose there is a subset $L \subseteq \Sigma^\omega$ with $\mathbb{A}(L) \geq k$. Since for every $L_1, L_2 \subseteq \Sigma^\omega$ with $L_1 \subseteq L_2$ we have $\mathbb{A}(L_1) \leq \mathbb{A}(L_2)$. Then $\mathbb{A}(\Sigma^\omega) \geq k$, witnessing Borel nonemptiness of $\mathbb{A}$ as $\Sigma^\omega$ is a Borel set (it is both open and closed). For $h = \mathsf{LimSup}$, the same argument provides the witness. For $h = \mathsf{Inf}$, by the same argument, the witness for Borel nonemptiness is a singleton (thus a closed set). For $h = \mathsf{LimInf}$, we consider only the set $V$ of values achieved by infinitely many words in $L$, and map $L$ to $\inf V$. Since every infinite set has a countably infinite subset, take some $v \in V$ and such a subset $L'$ of words in $L$ with value $v$. As $L'$ can be viewed as a countable union of singletons, it is Borel.

Let $h = \mathbb{E}$. Suppose there is a measure $\mu$ with $\mathbb{A}(\mu) \geq k$. By monotonicity of $\mathbb{E}$, it implies the existence of a word $w$ with $\mathcal{A}(w) \geq k$. Then, the (potentially infinite-state) Markov chain generating $w$ with probability 1 witnesses the Markov nonemptiness. ◀

**Proposition 4.2.** Consider a QLA $\mathbb{A} = (h, \mathcal{A})$ with $h \in \{\mathsf{Inf}, \mathsf{Sup}, \mathbb{E}\}$, a rational $k \in \mathbb{Q}$, and $\triangleright \in \{>, \geq\}$.

**(a)** Unrestricted variant
- **(i)** If $h \neq \mathsf{Sup}$ or $\triangleright = >$, then $\mathbb{A}$ is $\triangleright$-nonempty for $k$ iff $\mathcal{A}$ is $\triangleright$-nonempty for $k$.
- **(ii)** If $h = \mathsf{Sup}$, then $\mathbb{A}$ is $\geq$-nonempty for $k$ iff $\mathcal{A}$ is approximate-nonempty for $k$.

**(b)** Finite-state variant
- **(i)** If $h = \mathsf{Sup}$, then $\mathbb{A}$ is finite-state $\triangleright$-nonempty for $k$ iff $\mathbb{A}$ is $\triangleright$-nonempty for $k$.
- **(ii)** If $h = \mathsf{Inf}$, then $\mathbb{A}$ is finite-state $\triangleright$-nonempty for $k$ iff $\mathcal{A}$ is lasso-word $\triangleright$-nonempty for $k$.

**(iii)** If $\top_{\mathcal{A}}$ is achievable by a lasso word, then $\mathbb{A}$ is finite-state $\triangleright$-nonempty for $k$ iff $\mathbb{A}$ is $\triangleright$-nonempty for $k$ iff $\mathcal{A}$ is lasso-word $\triangleright$-nonempty for $k$.

For universality, we have the duals where we exchange $\mathsf{Inf}$ with $\mathsf{Sup}$, $>$ with $\geq$ , nonempty with universal, approximate-nonempty with approximate-universal, and $\top_{\mathcal{A}}$ with $\bot_{\mathcal{A}}$.

**Proof.** **(a)** Unrestricted

**(i)** Suppose $h = \mathsf{Inf}$. If $\mathbb{A}(L) = \inf_{w \in L} \mathcal{A}(w) \triangleright k$ for some $L \subseteq \Sigma^\omega$, then $\mathcal{A}(w) \triangleright k$ for all $w \in L$. If $\mathcal{A}(w) \triangleright k$ for some $w \in \Sigma^\omega$, then $\mathbb{A}(\{w\}) \triangleright k$ as well.
Suppose $h = \mathbb{E}$. Assume $\mathbb{A}(\mu) \triangleright k$ for some probability measure $\mu$. By monotonicity of $\mathbb{E}$, there is a word $w$ with $\mathcal{A}(w) \triangleright k$. Now, assume $\mathcal{A}(w) \triangleright k$ for some $w \in \Sigma^\omega$. Then, construct a measure having $\mu(\{w\}) = 1$, which witnesses the $\triangleright$-nonemptiness.
Suppose $h = \mathsf{Sup}$ and $\triangleright = >$. If there exists $L \subseteq \Sigma^\omega$ such that $\mathbb{A}(L) > k$, then there is a word $w \in \Sigma^\omega$ such that $\mathcal{A}(w) > k$. For the other direction, let $w$ such that $\mathcal{A}(w) > k$, then $\mathbb{A}(\{w\}) > k$.

**(ii)** We have $\mathbb{A}(\Sigma^\omega) \triangleright \mathbb{A}(L)$ for any $L \subseteq \Sigma^\omega$ since for every $L_1, L_2 \subseteq \Sigma^\omega$ with $L_1 \subseteq L_2$ we have $\mathbb{A}(L_1) \leq \mathbb{A}(L_2)$. Using this fact, there exists $L \subseteq \Sigma^\omega$ such that $\mathbb{A}(L) \geq k$ iff $\mathbb{A}(\Sigma^\omega) \geq k$ iff $\top_{\mathbb{A}} \geq k$ iff $\top_{\mathcal{A}} \geq k$ iff $\mathcal{A}$ is approximate-nonempty.

**(b)** Finite-state

**(i)** There exists $L \subseteq \Sigma^\omega$ such that $\mathbb{A}(L) \triangleright k$ iff $\mathbb{A}(\Sigma^\omega) \triangleright k$ iff there exists an $\omega$-regular language $L$ such that $\mathbb{A}(L) \triangleright k$. The last equivalence follows since $\Sigma^\omega$ is an $\omega$-regular language.

**(ii)** Suppose $h = \mathsf{Inf}$. If $\mathbb{A}(L) = \inf_{w \in L} \mathcal{A}(w) \triangleright k$ for some $\omega$-regular $L \subseteq \Sigma^\omega$, then $\mathcal{A}(w) \triangleright k$ for all $w \in L$. In particular, $\mathcal{A}(w') \triangleright k$ for some lasso $w' \in \Sigma^\omega$ since every $\omega$-regular contains a lasso word. Suppose $\mathcal{A}(w) \triangleright k$ for some ultimately-periodic $w \in \Sigma^\omega$, then $\mathbb{A}(\{w\}) \triangleright k$ and $\{w\}$ is $\omega$-regular.

**(iii)** Suppose $h = \mathbb{E}$. For any measure $\mu$, we have $\top_{\mathcal{A}} \triangleright \mathbb{A}(\mu)$ as the expected value is never outside of the interval of numbers taking expected value from. Hence, if $\mathbb{A}(\mu) \triangleright k$ for some measure $\mu$, then $\top_{\mathcal{A}} \triangleright k$ and there is an ultimately-periodic word $w$ such that $\mathcal{A}(w) = k$ by our initial assumption.
Suppose $\top_{\mathcal{A}} \triangleright k$. Then, by our initial assumption, $\mathcal{A}(w) \triangleright k$ where $w$ is an ultimately-periodic word such that $\mathcal{A}(w) = \top_{\mathcal{A}}$. Then we can construct a Markov chain generating $w$ and thus witnessing finite-state $\triangleright$-nonemptiness of $\mathbb{A}$ for $k$ and also unrestricted $\triangleright$-nonemptiness of $\mathbb{A}$ for $k$.
Suppose $h = \mathsf{Inf}$. One can combine (a)(i) and (b)(ii) with the fact that there exists $w \in \Sigma^\omega$ such that $\mathcal{A}(w) \triangleright k$ iff there exists lasso $w' \in \Sigma^\omega$ such that $\mathcal{A}(w') \triangleright k$.
If $h = \mathsf{Sup}$, the argument works as for $h = \mathsf{Inf}$ since if $\top_{\mathcal{A}}$ is achievable, approximate-nonemptiness is equivalent to $\geq$-nonemptiness.

The duals follow from the same arguments. ◀

**Proposition 4.3.** Consider a QLA $\mathbb{A} = (h, \mathcal{A})$ with $h \in \{\mathsf{Inf}, \mathsf{Sup}, \mathbb{E}\}$ and $k \in \mathbb{Q}$. If for every $\varepsilon > 0$ there is a lasso word $w$ such that $\mathcal{A}(w) \geq \top_{\mathcal{A}} - \varepsilon$, then $\mathbb{A}$ is $>$-nonempty for $k$ iff $\mathbb{A}$ is finite-state $>$-nonempty for $k$. Dually, if for every $\varepsilon > 0$ there is a lasso word $w$ such that $\mathcal{A}(w) \leq \bot_{\mathcal{A}} + \varepsilon$, then $\mathbb{A}$ is $\geq$-universal for $k$ iff $\mathbb{A}$ is finite-state $\geq$-universal for $k$.

**Proof.** The case $h = \mathsf{Sup}$ follows from Proposition 4.2(b)(i). For $h = \mathsf{Inf}$, from Proposition 4.2(a)(i) we have that $\mathbb{A}$ is $>$-nonempty for $k$ iff there exists $w$ such that $\mathcal{A}(w) > k$. Using the assumption above, the latter is equivalent to: there exists ultimately-periodic $w$ such that $\mathcal{A}(w) > k$ which is equivalent to $\mathbb{A}$ being finite-state $>$-nonempty for $k$ using Proposition 4.2(b)(ii). For $h = \mathbb{E}$, if $\mathbb{A}$ is $>$-nonempty, then Proposition 4.2(a)(i) gives us that $\mathcal{A}(w) > k$ for some $w$ which implies, using the assumption, that there is an ultimately periodic word $w'$ such that $\mathcal{A}(w') > k$ for which we can construct a Markov chain $\mathcal{M}$ generating $w'$ such that $\mathbb{A}(\mathcal{M}) = \mathcal{A}(w')$. The other direction is trivial. The dual cases follow the same line of argument. ◀

## C Proofs of Section 5

**Theorem 5.1.** Consider a QLA $\mathbb{A} = (h, (g, f, \mathcal{T}))$ with $f \in \{\mathsf{Inf}, \mathsf{Sup}, \mathsf{LimInf}, \mathsf{LimSup}\}$ and $g, h \in \{\mathsf{Inf}, \mathsf{Sup}\}$. Let $S \subseteq \Sigma^\omega$ be an $\omega$-regular language given by a Büchi automaton. The value $\mathbb{A}(S)$ is computable in PTIME when $g = h$ and in PSPACE when $g \neq h$.

**Proof.** This statement is a special case in the proof of Theorem 7.11. ◀

**Proposition 5.2.** There is a QLA $\mathbb{A}$ with a run aggregator $f \in \{\mathsf{LimInfAvg}, \mathsf{LimSupAvg}\}$ and an $\omega$-regular language $S$ such that no lasso word in $S$ achieves the value $\mathbb{A}(S)$.

**Proof.** Consider a language automaton $\mathbb{A} = (\mathsf{Sup}, \mathcal{A})$ where $\mathcal{A} = (g, f, \mathcal{T})$ and $\mathcal{T}$ consists of a single state that loops over $a$ and $b$ with respectively weight 1 and 0. Since $\mathcal{T}$ is deterministic, the choice of $g \in \{\mathsf{Inf}, \mathsf{Sup}, \mathbb{E}\}$ is not relevant. Let $f \in \{\mathsf{LimInfAvg}, \mathsf{LimSupAvg}\}$ and $L = (\Sigma^* b)^\omega$. Observe that a lasso word $w$ fulfills $\mathcal{A}(w) = 1$ iff it has finitely many letters $b$. We define $w' = baba^2ba^4ba^8 \ldots$. Since $w' \in L$ and $\mathcal{A}(w') = 1$, we have $\mathbb{A}(L) = \mathsf{Sup}_{w \in L}\mathcal{A}(w) = 1$. Hence, although the value $\mathbb{A}(L)$ is achieved by some word, it cannot be achieved by an ultimately periodic word in $L$. ◀

**Theorem 5.3.** Consider a QLA $\mathbb{A} = (h, (g, f, \mathcal{T}))$ with $f \in \{\mathsf{LimInfAvg}, \mathsf{LimSupAvg}\}$ and $g, h \in \{\mathsf{Inf}, \mathsf{Sup}\}$. Let $S \subseteq \Sigma^\omega$ be an $\omega$-regular language given by a Büchi automaton. The value $\mathbb{A}(S)$ is computable in PTIME when $g = h$ and not computable when $g \neq h$.

**Proof.** Let $\mathcal{A}$ be the underlying QWA of $\mathbb{A}$. First, suppose $h = g = \mathsf{Sup}$. Let $\mathcal{B}$ be the input Büchi automaton. Construct the product automaton $\mathcal{A} \times \mathcal{B}$. Among all SCCs of $\mathcal{A} \times \mathcal{B}$ that are reachable from the initial state, find the ones that contain at least one state whose $\mathcal{B}$-component is accepting in $\mathcal{B}$. In each such SCC, compute the maximum mean of its cycles by Karp's algorithm [26]. The largest among these mean values is exactly $\mathbb{A}(S)$. Even though such a cycle $C$ may not involve an accepting state of $\mathcal{B}$, we can construct a run of the product that visits an accepting state infinitely often while going over $C$ with increasing frequency. Overall, $\mathbb{A}(S)$ is computable in PTIME. The case of $h = g = \mathsf{Inf}$ is dual thanks to Proposition 2.2. Now, suppose $h = \mathsf{Inf}$ and $g = \mathsf{Sup}$. Let $S = \Sigma^\omega$ and notice that $\mathbb{A}(\Sigma^\omega) = \bot_{\mathcal{A}}$. An algorithm to compute $\bot_{\mathcal{A}}$ would imply the decidability of the universality problem of nondeterministic limit-average automata since $\mathcal{A}(w) \geq k$ for every word $w$ iff $\bot_{\mathcal{A}} \geq k$. Since this universality problem is undecidable [12, 17], the value $\mathbb{A}(S)$ is not computable. The case of $h = \mathsf{Sup}$ and $g = \mathsf{Inf}$ is dual thanks to Proposition 2.2. ◀

**Proposition 5.4.** There is a QLA $\mathbb{A}$ with the run aggregator $f = \mathsf{DSum}$ and an $\omega$-regular language $S$ such that no word in $S$ achieves the value $\mathbb{A}(S)$.

**Proof.** Consider $L = (\Sigma^* b)^\omega$ and the automata $\mathbb{A}$ and $\mathcal{A}$ from Proposition 5.2 with $f = \mathsf{DSum}_{0.5}$. Observe that, $a^\omega$ is the only word achieving the value 2, i.e., $\mathcal{A}(w) < 2$ for all $w \neq a^\omega$. For each $i \geq 1$ we define $w_i = a^i b^\omega \in L$. Since $\lim_{i \to \infty} \mathcal{A}(w_i) = 2$ and $\top_{\mathbb{A}} = 2$, we have $\mathbb{A}(L) = 2$. Therefore, the value $\mathbb{A}(L)$ is not achievable by any word in $L$. ◀

**Proposition 5.5.** Consider a QLA $\mathbb{A} = (\mathsf{Sup}, (\mathsf{Sup}, \mathsf{DSum}, \mathcal{T}))$. For every nonempty safety language $S \subseteq \Sigma^\omega$, the value $\mathbb{A}(S)$ is achievable by some run of a word in $S$.

**Proof.** Let $\mathbb{A}$ be as in the statement and $\mathcal{A}$ be its underlying QWA. Since $\mathbb{A}(S) = \sup_{w \in S} \mathcal{A}(w)$, there exists a sequence of words $\{w_i\}_{i \in \mathbb{N}}$ all coming from $S$ such that $\lim_{i \to \infty} \mathcal{A}(w_i) = \mathbb{A}(S)$. As in the proof of [9, Thm. 9], for each $i \in \mathbb{N}$, we can choose a run $\pi_i$ on $w_i$ satisfying $\mathsf{DSum}(\gamma(\pi_i)) = \mathcal{A}(w_i)$, thus $\lim_{i \to \infty} \mathsf{DSum}(\gamma(\pi_i)) = \mathbb{A}(S)$. Since the alphabet $\Sigma$ is finite, for each $j \in \mathbb{N}$ there exists at least one finite word $u_j \in \Sigma^j$ (of length $j$) that occurs as a prefix of infinitely many $w_i$'s. By a diagonalization argument, we extract an infinite subsequence $(w'_j)_{j \in \mathbb{N}}$ (of $(w_i)_{i \in \mathbb{N}}$) such that $u_j \prec w'_j$ for each $j$, and $\lim_{j \to \infty} \mathcal{A}(w'_j) = \mathbb{A}(S)$.

Similarly, consider the corresponding runs $\rho'_j$ on $w'_j$ (with $\mathsf{DSum}(\gamma(\rho'_j)) = \mathcal{A}(w'_j)$). For each $j \in \mathbb{N}$ there exists a finite run fragment $\theta_j$ of length $j$ that is a prefix of infinitely many $\rho'_j$'s. Thus, we can extract an infinite subsequence (which we continue to index by $j$) such that $\theta_j \prec \rho'_j$ and $\theta_j \prec \theta_k$ for all $j < k$. Denote by $\rho_j$ the run $\rho'_j$ from the extracted subsequence. Let $w = \lim_{j \to \infty} u_j$. Since each $u_j$ is a prefix of some word in $S$ and $S$ is a safety language, we have $w \in S$. Next, define the infinite run $\rho$ on $w$ by letting the $j$th transition of $\rho$ equal the $j$th transition of $\rho_j$. The run $\rho$ is well defined because $\theta_j \prec \theta_k$ for every $j, k \in \mathbb{N}$ with $j < k$.

For each $j \in \mathbb{N}$, note that the difference between the values $\mathsf{DSum}(\gamma(\rho))$ and $\mathsf{DSum}(\gamma(\rho_j))$ comes from the tail of the run: $|\mathsf{DSum}(\gamma(\rho)) - \mathsf{DSum}(\gamma(\rho_j))| \leq \sum_{i=j}^{\infty} \lambda^i \ell = \frac{\ell \lambda^j}{1-\lambda}$ where $\ell$ is the maximal transition weight in $\mathcal{A}$ and $\lambda$ the discount factor. Since $\lim_{j \to \infty} \frac{\ell \lambda^j}{1-\lambda} = 0$, we have $\lim_{j \to \infty} \mathsf{DSum}(\gamma(\rho_j)) = \mathsf{DSum}(\gamma(\rho))$. Moreover, since $\lim_{j \to \infty} \mathsf{DSum}(\gamma(\rho_j)) = \lim_{j \to \infty} \mathsf{DSum}(\gamma(\rho'_j)) = \mathbb{A}(S)$, we conclude $\mathsf{DSum}(\gamma(\rho)) = \mathcal{A}(w) = \mathbb{A}(S)$. ◀

**Proposition 5.6.** Consider a QLA $\mathbb{A} = (\mathsf{Sup}, (\mathsf{Sup}, \mathsf{DSum}, \mathcal{T}))$. For every language $S \subseteq \Sigma^\omega$ we have $\mathbb{A}(S) = \mathbb{A}(S')$ where $S'$ is the safety closure of $S$.

**Proof.** Let $\mathbb{A}$ be as in the statement and $\mathcal{A}$ be its underlying QWA. Since $S \subseteq S'$ by definition, we immediately have $\mathbb{A}(S) \leq \mathbb{A}(S')$. We prove below $\mathbb{A}(S) \geq \mathbb{A}(S')$.

Suppose towards contradiction that $\mathbb{A}(S) < \mathbb{A}(S')$ and let $\varepsilon = \mathbb{A}(S') - \mathbb{A}(S) > 0$. Since $S'$ is safe, thanks to Proposition 5.5 there exists a word $w \in S'$ and a run $\rho$ of $\mathcal{A}$ over $w$ such that $\mathcal{A}(w) = \mathsf{DSum}(\gamma(\rho)) = \mathbb{A}(S')$. Moreover, since $\mathcal{A}$ is discounting, there exists $n \in \mathbb{N}$ such that $\sum_{i=n}^{\infty} \ell \lambda^i < \frac{\varepsilon}{2}$, where $\ell$ is the maximal transition weight in $\mathcal{A}$ and $\lambda$ the discount factor.

Let $u$ be the prefix of $w$ of length $n$ and $\theta$ the corresponding finite prefix of $\rho$. By definition of $S'$, there exists $w' \in S$ with $u \prec w'$. Let $\rho'$ be a run on $w'$ with the finite prefix $\theta$. By the discounting property mentioned above, we have $|\mathsf{DSum}(\gamma(\rho)) - \mathsf{DSum}(\gamma(\rho'))| \leq \sum_{i=n}^{\infty} \ell \lambda^i < \frac{\varepsilon}{2}$. Therefore, we obtain $\mathsf{DSum}(\gamma(\rho')) \geq \mathsf{DSum}(\gamma(\rho)) - \frac{\varepsilon}{2} = \mathbb{A}(S') - \frac{\varepsilon}{2}$. Since $\rho'$ is a run of a word in $S$, we also have $\mathbb{A}(S) \geq \mathsf{DSum}(\gamma(\rho'))$. It implies that $\mathbb{A}(S) \geq \mathbb{A}(S') - \frac{\varepsilon}{2}$, which contradicts the initial assumption that $\mathbb{A}(S') - \mathbb{A}(S) = \varepsilon > 0$. ◀

**Theorem 5.7.** Consider a QLA $\mathbb{A} = (h, (g, \mathsf{DSum}, \mathcal{T}))$ with $g, h \in \{\mathsf{Inf}, \mathsf{Sup}\}$. Let $S \subseteq \Sigma^\omega$ be an $\omega$-regular language given by a Büchi automaton. The value $\mathbb{A}(S)$ is computable in PTime when $g = h$. If $\mathbb{A}(S)$ is computable when $g \neq h$, then the $\geq$-universality of nondeterministic discounted-sum word automata is decidable.

**Proof.** We start with the case with $h = g = \mathsf{Sup}$. Let $\mathcal{B}$ be the Büchi automaton recognizing $S$. The Büchi automaton $\mathcal{B}'$ recognizing the safety closure of $S$ is computable in PTime thanks to [3]. By safety condition, we can assume without loss of generality that $\mathcal{B}'$ is complete and has only one rejecting state which is a sink, called $p$ hereafter. We construct the discounted-sum word automaton $\mathcal{C}$ as the cross-product of $\mathcal{B}'$ and $\mathcal{A}$ where the transition weights are taken from $\mathcal{A}$. By Proposition 5.6, we have $\mathbb{A}(L(\mathcal{B})) = \mathbb{A}(L(\mathcal{B}'))$, and by Proposition 5.5, the value $\mathbb{A}(L(\mathcal{B}'))$ is achievable by some run of a word of $L(\mathcal{B}')$. This implies that the value $\mathbb{A}(L(\mathcal{B}))$ is achieved by a run that never visits a pair with $p$. Hence, we can compute this value in polynomial time thanks to the algorithm solving the one-player discounted payoff objectives game of [4] on the arena defined by $\mathcal{C}$ where all pairs containing $p$ are removed. The case of $g = h = \mathsf{Inf}$ is dual thanks to Proposition 2.2.

Suppose $h = \mathsf{Inf}$ and $g = \mathsf{Sup}$. Considering $S = \Sigma^\omega$, we have $\mathbb{A}(S) = \bot_{\mathcal{A}}$. Notice that $\mathcal{A}$ is $\geq$-universal for $k$ iff $\bot_{\mathcal{A}} \geq k$. Therefore, the evaluation problem for this class of QLAs is at least as hard as the universality problem of nondeterministic discounted-sum word automata. The case of $h = \mathsf{Sup}$ and $g = \mathsf{Inf}$ is dual. ◀

**Theorem 5.8.** Consider a QLA $\mathbb{A} = (h, (\mathbb{E}, f, \mathcal{T}))$ with $f \in \{\mathsf{Inf}, \mathsf{Sup}, \mathsf{LimInf}, \mathsf{LimSup}, \mathsf{LimInfAvg}, \mathsf{LimSupAvg}, \mathsf{DSum}\}$ and $h \in \{\mathsf{Inf}, \mathsf{Sup}\}$. Let $\mu$ be a probability measure given by a finite-state Markov chain. The value $\mathbb{A}(\mu)$ is not computable.

**Proof.** For $h = \mathsf{Sup}$, we sketch a reduction from the approximate-nonemptiness problem of QWA with $f$ and $g$ as in the statement, which is undecidable as proved in Theorem 6.8. Evaluating $\mathbb{A}(\Sigma^\omega)$ is equivalent to evaluating $\sup_{w \in \Sigma^\omega} \mathcal{A}(w)$, which is equal to $\top_{\mathcal{A}}$. Recall that deciding approximate-nonemptiness for $k$ is equivalent to deciding whether $\top_{\mathcal{A}} \geq k$. For $h = \mathsf{Inf}$, the proof goes by a reduction to the approximate-universality problem, which is undecidable (see Theorem 6.8). Recall that deciding approximate-universality for $k$ is equivalent to deciding whether $\bot_{\mathcal{A}} > k$. ◀

**Theorem 5.9.** Consider a QLA $\mathbb{A} = (\mathbb{E}, (g, f, \mathcal{T}))$ with $f \in \{\mathsf{Inf}, \mathsf{Sup}, \mathsf{LimInf}, \mathsf{LimSup}\}$ and $g \in \{\mathsf{Inf}, \mathsf{Sup}\}$. Let $\mu$ be a probability measure given by a finite-state Markov chain. The value $\mathbb{A}(\mu)$ is computable in ExpTime.

**Proof.** This proof extends the results of [32]. Let $X = \{x_1, x_2, \dots, x_n\}$ be the set of weights of $\mathcal{A}$ such that $x_1 < x_2 < \dots < x_n$. Let $P_{\geq x}$ be the probability $P_{w \sim \mu}[\mathcal{A}(w) \geq x]$ for all $x \in X$. We define the probabilities $P_{w\sim\mu}[\mathcal{A}(w) = x_i] = P_{\geq x_i} - P_{\geq x_{i+1}}$ for $i < n$ and $P_{w\sim\mu}[\mathcal{A}(w) = x_n] = P_{\geq x_n}$. For all $w \in \Sigma^\omega$, we have $\mathcal{A}(w) \in X$, allowing us to express $\mathbb{A}(\mu) = \sum_{i=1}^{n} x_i P_{w\sim\mu}[\mathcal{A}(w) = x_i]$.

We only present the construction when $g = \mathsf{Sup}$. The duality property of Proposition 2.2 allows us to treat the case $g = \mathsf{Inf}$. When $f = \mathsf{LimSup}$ and $g = \mathsf{Sup}$, we can construct a Büchi automaton $\mathcal{A}_{\geq x}$ recognizing $L(\mathcal{A}_{\geq x}) = \{w \in L \mid \mathcal{A}(w) \geq x\}$ in linear time [15]. Thanks to [6, Prop. 2.1.] and [15, Thm 13. (iv)], we can assume that $f = \mathsf{LimSup}$ without loss of generality. From $\mathcal{A}_{\geq x}$, we can construct an equivalent deterministic Rabin automaton $\mathcal{B}_{\geq x}$ in ExpTime. Finally, we construct the Markov chain $\mathcal{C}$ as the cross-product of the Markov chain $M$ representing the measure $\mu$ with the automaton $\mathcal{B}_{\geq x}$. Observe that $P_{\geq x}$ corresponds to the probability for a random walk in $\mathcal{C}$ to fulfill the accepting condition of the Rabin automaton $\mathcal{B}_{\geq x}$. Let $Q_1$, $Q_2$, and $Q = Q_1 \times Q_2$ be the set of states of $M$, $\mathcal{B}_{\geq x}$, and $\mathcal{C}$, respectively. Given a pair of state sets $(G, B) \subseteq Q_2 \times Q_2$ of the accepting condition of the Rabin automaton $\mathcal{B}_{\geq x}$ and a bottom strongly connected component $S \subseteq Q$ of $\mathcal{C}$, we say that $S$ suits $(G, B)$ iff $S \subseteq Q_1 \times G$ and $S \cap (Q_1 \times B) = \emptyset$. Observe that, when $S$ suits $(G, B)$, every random walk in $S$ fulfills the accepting condition of the Rabin automaton $\mathcal{B}_{\geq x}$. Thus, $P_{\geq x}$ is the probability of reaching some bottom strongly connected component of $\mathcal{C}$ that suits some pair of the accepting condition of $\mathcal{B}_{\geq x}$. It can be computed in PTime thanks to [5, Chapter 10.3]. ◀

**Theorem 5.11.** Consider a QLA $\mathbb{A} = (\mathbb{E}, (g, \mathsf{DSum}, \mathcal{T}))$ with $g \in \{\mathsf{Inf}, \mathsf{Sup}\}$. Let $\mu$ be a probability measure given by a finite-state Markov chain. If $\mathbb{A}(\mu)$ is computable, then the $\geq$-universality of nondeterministic discounted-sum word automata is decidable.

**Proof.** Given a $\mathsf{DSum}$ word automaton $\mathcal{A}$ and a threshold $k$, we can decide whether $\mathcal{A}(w) \geq k$ holds for every $w \in \Sigma^\omega$ by reducing the evaluation of some language automaton $\mathbb{A}' = (\mathbb{E}, \mathcal{A}')$ where $\mathcal{A}$ has $g = \mathsf{Sup}$, and $f = \mathsf{DSum}$. We construct a word automaton $\mathcal{A}'$ such that $\mathcal{A}'(w) = \max(k, \mathcal{A}(w))$, for all $w \in \Sigma^\omega$. This is possible by introducing a new run from the initial state, for which the first transition is weighted by $k$ and all other transition is weighted by 0. The construction thus consists in adding a new state with self-loop weighed 0 that is reachable only from the initial state with weight $k$. Let $\mathbb{A} = (\mathbb{E}, \mathcal{A})$ and $\mathbb{A}' = (\mathbb{E}, \mathcal{A}')$ be two language automata, and $\mathcal{M}$ be the uniform Markov chain (i.e., single state with a uniform self loop). We show that $\mathbb{A}(\mathcal{M}) = \mathbb{A}'(\mathcal{M})$ if and only if $\mathcal{A}(w) \geq k$ for every $w \in \Sigma^\omega$. If $\mathcal{A}(w) \geq k$ holds for every $w \in \Sigma^\omega$, then both values are trivially equal. Otherwise, there exists $w \in \Sigma^\omega$ such that $\mathcal{A}(w) < k$, which by discounting implies that there exists a finite prefix $u \prec w$ such that $\mathcal{A}(uw') < k$ for all $w' \in \Sigma^\omega$. Since $\mathcal{M}$ is the uniform Markov chain, the measure of the set $u\Sigma^\omega$ is nonzero, and thus $\mathbb{A}(\mathcal{M}) \neq \mathbb{A}'(\mathcal{M})$. ◀

## D Proofs of Section 6

**Theorem 6.2.** Consider a QLA $\mathbb{A} = (h, (\mathsf{Sup}, f, \mathcal{T}))$ with $f \in \{\mathsf{Inf}, \mathsf{Sup}, \mathsf{LimInf}, \mathsf{LimSup}, \mathsf{LimInfAvg}, \mathsf{LimSupAvg}, \mathsf{DSum}\}$ and $h \in \{\mathsf{Inf}, \mathsf{Sup}, \mathbb{E}\}$. Let $\triangleright \in \{>, \geq\}$. The $\triangleright$-nonemptiness of $\mathbb{A}$ is in PTime. The statement holds also for the finite-state restriction.

**Proof.** Let $\mathcal{A}$ be the underlying QWA of $\mathbb{A}$. As $g = \mathsf{Sup}$, the value $\top_\mathcal{A}$ is achievable by some lasso word [15, Thm. 3]. Thus, using Proposition 4.2(b)(iii), we can reduce whether $\mathbb{A}$ is (finite-state) $\triangleright$-nonempty for $k$ to whether $\mathcal{A}$ is (lasso-word) $\triangleright$-nonempty for $k$, which is decidable in PTime. ◀

**Theorem 6.3.** Consider a QLA $\mathbb{A} = (h, (\mathsf{Inf}, f, \mathcal{T}))$ with $f \in \{\mathsf{Inf}, \mathsf{Sup}, \mathsf{LimInf}, \mathsf{LimSup}\}$ and $h \in \{\mathsf{Inf}, \mathsf{Sup}, \mathbb{E}\}$. Let $\triangleright \in \{>, \geq\}$. The $\triangleright$-nonemptiness of $\mathbb{A}$ is in PSpace. The statement holds also for the finite-state restriction.

**Proof.** For any QWA $\mathcal{A} = (g, f, \mathcal{T})$ with $f \in \{\mathsf{Inf}, \mathsf{Sup}, \mathsf{LimInf}, \mathsf{LimSup}\}$, the values $\top_\mathcal{A}$ and $\bot_\mathcal{A}$ are both achievable by some lasso words. This is because both values equal one of the weights in $\mathcal{A}$, and for each weight $x$ the language $\mathcal{A}_{=x}$ is $\omega$-regular. Thus, using Proposition 4.2(b)(iii), we can reduce whether $\mathbb{A}$ is (finite-state) $\triangleright$-nonempty for $k$ to whether $\mathcal{A}$ is $\triangleright$-nonempty for $k$, which is decidable in PSpace when $g = \mathsf{Inf}$ [13, Thm. 3]. ◀

**Proposition 6.4.** Consider a QWA $\mathcal{A} = (\mathsf{Sup}, \mathsf{LimSupAvg}, \mathcal{T})$. For every $\varepsilon > 0$ there is a lasso word $w$ such that $\mathcal{A}(w) < \bot_\mathcal{A} + \varepsilon$.

**Proof.** Let $\mathcal{A} = (\mathsf{Sup}, \mathsf{LimSupAvg}, \mathcal{T})$ be a QWA with the alphabet $\Sigma$ and the set of states $Q$. For all finite sequences of rational weights $x = x_0, x_1, \ldots$, we define $\mathsf{Sum}(x) = \sum_{i=0}^{|x|} x_i$ and $\mathsf{Avg}(x) = \frac{\mathsf{Sum}(x)}{|x|}$. Given $\varepsilon > 0$, let $w \in \Sigma^\omega$ be a word such that $\mathcal{A}(w) < \bot_\mathcal{A} + \frac{\varepsilon}{4}$. By definition of $\mathsf{LimSupAvg}$, there exists $\ell \in \mathbb{N}$ such that for all $u \in \Sigma^*$ if $|u| > \ell$ and $u \prec w$ then $\sup\{\mathsf{Avg}(\gamma(\pi)) \mid \pi \in R(\mathcal{T}, u)\} < \bot_\mathcal{A} + \frac{\varepsilon}{2}$ where $R(\mathcal{T}, u)$ denotes the set of runs of $u$ in $\mathcal{T}$. Otherwise, infinitely many prefix of $w$ achieve an average of at least $\bot_\mathcal{A} + \frac{\varepsilon}{2}$ which contradict that $\mathcal{A}(w) < \bot_\mathcal{A} + \frac{\varepsilon}{2}$. We define $x_{\max}$ and $x_{\min}$ as the respective maximal and minimal weight appearing on $\mathcal{A}$ together with $c = x_{\max} - x_{\min}$. Since $Q$ is finite, $w$ admits infinitely many prefixes allowing to reach the same set of states in $\mathcal{A}$. Accordingly, consider $u_1, u_2 \in \Sigma^*$ such that $u_1 \prec u_1 u_2 \prec w$, $\delta(s, u_1) = \delta(s, u_1 u_2)$, $|u_1 u_2| > \ell$ and $\frac{|u_1|}{|u_2|} < \frac{\varepsilon}{\varepsilon + 2c}$. In particular, we have that $\delta(s, u_1) = \delta(s, u_1 u_2^k)$ and $\frac{|u_1|}{k|u_2|} < \frac{\varepsilon}{\varepsilon + 2c}$ for all $k \in \mathbb{N}$. Let $k \in \mathbb{N}$, and let $\pi_1$ and $\pi_2$ be two runs on $u_1$ and $u_2^k$, respectively, such that $\pi_2$ starts from the state where

$\pi_1$ ends. Observe that $\mathsf{Sum}(\gamma(\pi_1)) \geq x_{\min}|u_1| \geq (\bot_{\mathcal{A}} - c)|u_1|$. Additionally, the following holds by construction:

$$\bot_{\mathcal{A}} + \tfrac{\varepsilon}{2} > \tfrac{\mathsf{Sum}(\gamma(\pi_1))+\mathsf{Sum}(\gamma(\pi_2))}{|u_1|+k|u_2|} \geq \tfrac{(\bot_{\mathcal{A}}-c)|u_1|+\mathsf{Sum}(\gamma(\pi_2))}{|u_1|+k|u_2|}$$
$$\left(\bot_{\mathcal{A}} + \tfrac{\varepsilon}{2}\right)\left(\tfrac{|u_1|}{k|u_2|} + 1\right) > \tfrac{|u_1|}{k|u_2|}\left(\bot_{\mathcal{A}} - c\right) + \tfrac{\mathsf{Sum}(\gamma(\pi_2))}{k|u_2|}$$
$$\bot_{\mathcal{A}} + \varepsilon > \mathsf{Avg}(\gamma(\pi_2))$$

Since for all $k \in \mathbb{N}$ every run $\pi$ on $u_1u_2^k$ fulfills $\bot_{\mathcal{A}} + \varepsilon > \mathsf{Avg}(\gamma(\pi))$, the lasso word $u_1u_2^\omega$ fulfills $\bot_{\mathcal{A}} + \varepsilon > \mathcal{A}(u_1u_2^\omega)$. ◀

▶ **Theorem D.1.** [12, 17, 24]. *Consider a QWA $\mathcal{A} = (\mathsf{Inf}, f, \mathcal{T})$ with $f \in \{\mathsf{LimInfAvg}, \mathsf{LimSupAvg}\}$. Let $\triangleright \in \{>, \geq\}$. The $\triangleright$-nonemptiness of $\mathcal{A}$ is undecidable, whether we consider all words from $\Sigma^\omega$ or only lasso words. Moreover, its approximate-nonemptiness is also undecidable.*

**Proof.** The undecidability of

- $>$-nonemptiness for $\mathsf{LimInfAvg}$ and $\mathsf{LimSupAvg}$ was proved in [17, Thm. 4],
- $\geq$-nonemptiness for $\mathsf{LimSupAvg}$ in [17, Thm. 5], and
- $\geq$-nonemptiness for $\mathsf{LimInfAvg}$ in [24, Thm. 1].

The undecidability of lasso-word $\geq$-nonemptiness for $\mathsf{LimInfAvg}$ and $\mathsf{LimSupAvg}$ was proved in [17, Thm. 6]. The lasso-word $>$-nonemptiness for $\mathsf{LimInfAvg}$ is undecidable since the problem is equivalent to $>$-nonemptiness for $\mathsf{LimInfAvg}$ by the dual version of Proposition 6.4. The corresponding problems for $\mathsf{LimSupAvg}$ are equivalent to those for $\mathsf{LimInfAvg}$ since for all lasso words the $\mathsf{LimSupAvg}$ and $\mathsf{LimInfAvg}$ values coincide.

To prove the undecidability of approximate-nonemptiness, notice that the proofs of [17, Thm. 5] and [24, Thm. 1] both actually show the undecidability of approximate-nonemptiness. In both proofs, either there is a word whose limit-average value is at least the threshold, or there is a gap between the potential limit-average values and the threshold. These gaps correspond to $\varepsilon$ in our definition of approximate-nonemptiness, which are of the size $\frac{1}{\ell}$ in [17, Thm. 5] and $\frac{1}{2N}$ in [24, Thm. 1], where both $\ell$ and $N$ encode the lengths of some runs. ◀

**Theorem 6.5.** Consider a QLA $\mathbb{A} = (h, (\mathsf{Inf}, f, \mathcal{T}))$ with $f \in \{\mathsf{LimInfAvg}, \mathsf{LimSupAvg}\}$ and $h \in \{\mathsf{Inf}, \mathsf{Sup}, \mathbb{E}\}$. Let $\triangleright \in \{>, \geq\}$. The $\triangleright$-nonemptiness of $\mathbb{A}$ is undecidable. The statement holds also for the finite-state restriction when (i) $h \neq \mathbb{E}$ or (ii) $f = \mathsf{LimInfAvg}$ and $\triangleright = >$.

**Proof.** Let $\mathcal{A}$ be the underlying QWA of $\mathbb{A}$. For the unrestricted cases, we use Proposition 4.2(a)(i-ii) to reduce the $\triangleright$-nonemptiness or approximate-nonemptiness of $\mathcal{A}$, both undecidable by Theorem D.1, to the $\triangleright$-nonemptiness of $\mathbb{A}$.

For the finite-state restrictions, we have the following. If $h = \mathsf{Inf}$, we use Proposition 4.2(b)(ii) to reduce the lasso-word $\triangleright$-nonemptiness of $\mathcal{A}$, which is undecidable by Theorem D.1, to the finite-state $\triangleright$-nonemptiness of $\mathbb{A}$. If $h = \mathsf{Sup}$, we use Proposition 4.2(b)(i) to reduce the unrestricted $\triangleright$-nonemptiness of $\mathbb{A}$, which we proved undecidable above, to the finite-state $\triangleright$-nonemptiness of $\mathbb{A}$. If $h = \mathbb{E}$ and $f = \mathsf{LimInfAvg}$, by Proposition 4.3 and the dual version of Proposition 6.4 (namely, for every QWA $\mathcal{A} = (\mathsf{Inf}, \mathsf{LimInfAvg}, \mathcal{T})$ and every $\varepsilon > 0$ there is a lasso word $w$ such that $\mathcal{A}(w) > \top_{\mathcal{A}} - \varepsilon$), the finite-state $>$-nonemptiness of $\mathbb{A}$ is equivalent to the unrestricted $>$-nonemptiness of $\mathbb{A}$, which we proved undecidable above. We leave the remaining finite-state cases open. ◀

**Proposition 6.6.** Consider a QWA $\mathcal{A} = (\mathsf{Sup}, \mathsf{DSum}, \mathcal{T})$. There is a word $w$ such that $\mathcal{A}(w) = \bot_{\mathcal{A}}$.

**Proof.** Let $\mathcal{A}$ be a QWA as in the statement. For every $u \in \Sigma^*$, let $P_u = \{\mathcal{A}(uw) \mid w \in \Sigma^\omega\}$. We claim that for every $u \in \Sigma^*$, there exists $\sigma \in \Sigma$ such that $\inf P_u = \inf P_{u\sigma}$. To prove, let $u$ be a finite word and observe the following: $P_u = \bigcup_{\sigma\in\Sigma} P_{u\sigma}$. Then, since $\inf(\bigcup_{i=1}^n X_i) = \min_{1\leq i\leq n} \inf X_i$ for every finite collection $X_1, \dots, X_n$ of sets and $\Sigma$ is finite, we have the following: $\inf P_u = \min_{\sigma\in\Sigma}(\inf P_{u\sigma})$. Finally, since some $\hat{\sigma} \in \Sigma$ achieves this minimum by definition, we have $\inf P_u = \inf P_{u\hat{\sigma}}$.

Now, let $u_0 = \epsilon$ be the empty word. For each $i \geq 0$, choose a letter $\sigma_0 \in \Sigma$ and set $u_{i+1} = u_i\sigma_i$. Then, define $w$ as the limit of this sequence, i.e., $w = \sigma_0\sigma_1\dots$. By the definition of $w$, we have $\lim_{i\to\infty} \inf P_{u_i} = \bot_{\mathcal{A}}$. Since the sequence of infima is nondecreasing, we can equivalently write $\sup_{u\prec w} \inf P_u = \bot_{\mathcal{A}}$. By [21, Definition 24] the left-hand side expresses the co-safety closure of $\mathcal{A}$. Moreover, since every nondeterministic discounted-sum word automaton is co-safe [6, Theorems 4.15 and 4.16], the left-hand side equals $\mathcal{A}(w)$. Therefore, $\mathcal{A}(w) = \bot_{\mathcal{A}}$. ◀

**Theorem 6.7.** Consider a QLA $\mathbb{A} = (h, (\mathsf{Inf}, \mathsf{DSum}, \mathcal{T}))$ with $h \in \{\mathsf{Inf}, \mathsf{Sup}, \mathbb{E}\}$. Let $\triangleright \in \{>, \geq\}$. If the $\triangleright$-nonemptiness of $\mathbb{A}$ is decidable, then the universality of nondeterministic discounted-sum word automata is decidable. The statement holds also for the finite-state restriction when (i) $h = \mathsf{Sup}$ or (ii) $\triangleright = >$.

**Proof.** Let $\hat{\triangleright} \in \{>, \geq\} \setminus \{\triangleright\}$ be the dual of $\triangleright$. Let $\mathcal{A} = (\mathsf{Inf}, \mathsf{DSum}, \mathcal{T})$ be the underlying QWA of $\mathbb{A}$. We show that if the $\triangleright$-nonemptiness of $\mathbb{A}$ is decidable, then the $\hat{\triangleright}$-universality of $\mathcal{A}$ is decidable. Observe that since $\top_{\mathcal{A}}$ is achievable thanks to the dual version of Proposition 6.6, the approximate-nonemptiness and the $\geq$-nonemptiness problems coincide. Then, using Proposition 4.2(a)(i-ii), we reduce the $\triangleright$-nonemptiness of $\mathcal{A}$, which is open and famously hard, to the $\triangleright$-nonemptiness of $\mathbb{A}$. For finite-state restriction, we have the following: If $h = \mathsf{Sup}$, we use Proposition 4.2(b)(i) to reduce the unrestricted $\triangleright$-nonemptiness of $\mathbb{A}$ to the finite-state $\triangleright$-nonemptiness of $\mathbb{A}$. If $\triangleright = >$, since $\top_{\mathcal{A}}$ is approximable by lasso words, we use Proposition 4.3 to reduce the unrestricted $>$-nonemptiness of $\mathbb{A}$ to the finite-state $>$-nonemptiness of $\mathbb{A}$. We leave the finite-state $\geq$-nonemptiness for $h \neq \mathsf{Sup}$ open. ◀

▶ **Theorem D.2.** **[31, Cor. 3.4].** *For any $\varepsilon$ such that $0 < \varepsilon < \frac{1}{2}$, the following problem is undecidable: Let $\mathcal{A} = (\mathbb{E}, \mathsf{Sup}, \mathcal{T})$ be a probabilistic word automaton with weights over $\{0, 1\}$ and accepting states, i.e., a subset of states $S \subseteq Q$ such that all transitions leading to $S$ have weight 1 and every other transition has weight 0. Given that either of the two cases hold:*

**(i)** *There exists a word $w \in \Sigma^\omega$ such that $\mathcal{A}(w) > 1 - \varepsilon$.*

**(ii)** *For all words $w \in \Sigma^\omega$, we have $\mathcal{A}(w) < \varepsilon$.*

*Decide whether (i) holds.*

**Theorem 6.8.** Consider a QWA $\mathcal{A} = (\mathbb{E}, f, \mathcal{T})$ with $f \in \{\mathsf{Inf}, \mathsf{Sup}, \mathsf{LimInf}, \mathsf{LimSup}, \mathsf{LimInfAvg}, \mathsf{LimSupAvg}, \mathsf{DSum}\}$. Let $\triangleright \in \{>, \geq\}$. The $\triangleright$-nonemptiness of $\mathcal{A}$ is undecidable, whether we consider all words from $\Sigma^\omega$ or only lasso words. Moreover, its approximate-nonemptiness is also undecidable.

**Proof.** When considering all words, the problems are generally known to be undecidable [31].

For the lasso-word variants, let us start with $f = \mathsf{DSum}$. Since the top value can be arbitrarily approximated by lasso words, then the lasso-word $>$-nonemptiness is equivalent to the unrestricted $>$-nonemptiness. For $\geq$-nonemptiness, we use [31, Thm. 4.8]: it is undecidable to determine whether the top value (which is known to be $\frac{1}{2}$) of a particular automaton is achieved by a lasso word. Since this problem can be encoded as a lasso-word $\geq$-nonemptiness (with threshold $\frac{1}{2}$), this nonemptiness problem is undecidable.

For other run aggregators, the undecidability follows from the inapproximability of the top value as stated by Theorem D.2. Let us first cover the case $f = \mathsf{Sup}$. If (i) holds for some $w \in \Sigma^\omega$ then there exists a long enough prefix $u \in \Sigma^*$ of $w$ such that $\mathcal{A}(uv^\omega) > \frac{1}{2}$ for any $v \in \Sigma^*$ since $\mathcal{A}(uv^\omega)$ is at least the probability of getting to an accepting state after reading $v$. If (ii) holds, there is no lasso word with value above $\frac{1}{2}$. For $f = \mathsf{Inf}$, the argument is the same but we also change all weights to 1 except the weights of sinks states, i.e., states from which the probability of getting to an accepting state is zero. For $f \in \{\mathsf{LimSup}, \mathsf{LimInf}, \mathsf{LimSupAvg}, \mathsf{LimInfAvg}\}$, the reduction is the same but we modify $\mathcal{A}$ such that every outgoing transition of an accepting state is a loop.

Regarding the approximate-nonemptiness problem, since $\mathsf{DSum}$ has $\top_{\mathcal{A}}$ value always achievable [30, Thm. 2.3.2], it is equivalent to the $\geq$-nonemptiness problem. For the remaining run aggregators $f$, the above construction proves undecidability of approximate-nonemptiness thanks to Theorem D.2 as $\varepsilon < \frac{1}{2}$. ◀

**Theorem 6.9.** Consider a QLA $\mathbb{A} = (h, (\mathbb{E}, f, \mathcal{T}))$ with $f \in \{\mathsf{Inf}, \mathsf{Sup}, \mathsf{LimInf}, \mathsf{LimSup}, \mathsf{LimInfAvg}, \mathsf{LimSupAvg}, \mathsf{DSum}\}$ and $h \in \{\mathsf{Inf}, \mathsf{Sup}, \mathbb{E}\}$. Let $\triangleright \in \{>, \geq\}$. The $\triangleright$-nonemptiness of $\mathbb{A}$ is undecidable. The statement holds also for the finite-state restriction when (i) $h \neq \mathbb{E}$ or (ii) $f = \mathsf{DSum}$ and $\triangleright = >$.

**Proof.** For the unrestricted cases, we use Proposition 4.2(a)(i-ii) to reduce the $\triangleright$-nonemptiness or approximate-nonemptiness of $\mathcal{A}$, both of which are undecidable by Theorem 6.8, to the $\triangleright$-nonemptiness of $\mathbb{A}$.

For the finite-state case, we have the following: If $h = \mathsf{Sup}$, we use Proposition 4.2(b)(i) to reduce the unrestricted $\triangleright$-nonemptiness of $\mathbb{A}$, which we proved undecidable above, to the finite-state $\triangleright$-nonemptiness of $\mathbb{A}$. If $h = \mathsf{Inf}$, we use Proposition 4.2(b)(ii) to reduce the lasso-word $\triangleright$-nonemptiness of $\mathcal{A}$, which is undecidable by Theorem 6.8, to the finite-state $\triangleright$-nonemptiness of $\mathbb{A}$. If $h = \mathbb{E}$ and $f = \mathsf{DSum}$, we use Proposition 4.3 to reduce the unrestricted $>$-nonemptiness of $\mathbb{A}$ to the finite-state $>$-nonemptiness of $\mathbb{A}$. We leave open the finite-state $\triangleright$-nonemptiness for (i) $h = \mathbb{E}$ and $f \neq \mathsf{DSum}$, and (ii) $h = \mathbb{E}$ and $f = \mathsf{DSum}$ and $\triangleright = \geq$. ◀

## E Proofs of Section 7

**Lemma 7.1.** Let $\mathcal{T}$ be a (weighted) labeled transition system with $n$ states and $q_I$ its initial state. For every state $q_2$ of $\mathcal{T}$, there exist infinitely many words with a run from $q_I$ visiting $q_2$ infinitely often iff $\mathcal{T}$ complies with the pattern parameterized by $q_2$ and $n$ given in Figure 2a.

**Proof.** We start with the backward implication. Given the state $q_2$, we assume that $\mathcal{T}$ complies with the pattern in Figure 2a exhibiting four finite runs of the form $\rho_1 : q_I \xrightarrow{u_1} q_1$, $\ell_1 : q_1 \xrightarrow{u_2 u_3} q_1$, $\rho_2 : q_1 \xrightarrow{v} q_2$, $\ell_2 \colon q_2 \xrightarrow{u_2 u_3} q_2$. Additionally we have that $|u_2| = |v|$ and $u_2 \neq v$. For all $i \in \mathbb{N}$, we define the run $\pi_i = \rho_1 (\ell_1)^i \rho_2 (\ell_2)^\omega$ from $q_I$ over $u_1 (u_2 u_3)^i v (u_2 u_3)^\omega$. Since $|u_2| = |v|$ and $u_2 \neq v$, we have that $\pi_i$ and $\pi_j$ read distinct words for all $i \neq j$. In particular, the set $\{u_1 (u_2 u_3)^i v (u_2 u_3)^\omega \mid i \in \mathbb{N}\}$ of ultimately periodic words with a run visiting $q_2$ infinitely often is infinite.

Next, we prove the forward implication having an infinite set $S \subseteq \Sigma^\omega$ of words admitting some run from the initial state $q_I$ that visits the state $q_2$ infinitely often. Consider a word $w_1 \in S$ together with one of its runs $\pi_1$ that visits $q_2$ infinitely. Let $w_0 \in \Sigma^*$ be a finite

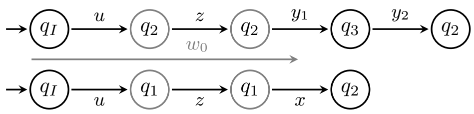


**Figure 3** Run synchronization in proof of Lemma 7.1, where $u, x, y_1, y_2, z \in \Sigma^+$ are such that $|x| = |y_1|$, $x \neq y_1$, $|z| = |y_1 y_2|$, and $m|x| = |z|$ for some $m \in \{1, \ldots, 4n^4\}$.

prefix of $w_1$ for which $\pi_1$ visits at least $n^2$ times the state $q_2$ while reading $w_0$, where $n$ refers to the number of states of $\mathcal{T}$. Assuming that $w_1$ mismatches with all the other words of $S$ exclusively over its $|w_0|$ first letters contradicts $|S| = \infty$. Hence, there exists another word $w_2 \in S$, distinct from $w_1$, but starting with the finite prefix $w_0$. Let $\pi_2$ be a run over $w_2$ that visits $q_2$ infinitely often. Since $\pi_1$ visits $n^2$ times the state $q_2$ while reading $w_0$, by the pigeon hole principle, there exists a state $q_1$ such that the pair $(q_1, q_2)$ is visited twice by $(\pi_1, \pi_2)$ while reading $w_0$. Figure 3 displays the two runs $\pi_1$ and $\pi_2$. The finite word $z$ refers the factor of $w_0$ corresponding to the synchronized loop, and the finite word $u$ stands for the prefix of $w_0$ before $z$. We ensure that $z \leq n^2$ by removing inner synchronized loops. We denote by $x$ and $y_1$ the synchronized section where $\pi_1$ and $\pi_2$ mismatch. In particular, $|x| = |y_1|$ and $x \neq y_1$. Since both $\pi_1$ and $\pi_2$ visit $q_2$ infinitely often, we can take $x$ and $y_1$ long enough so that $x$ leads to $q_2$. Then, we denote by $\hat{y}_2$ the continuation of $y_1$ ending in $q_2$. Additionally, we ensure that $|y_1 \hat{y}_2| \leq 3n^2 + 1$. Indeed, any synchronized loop in the factors $x$ and $y_1$ appearing before or after the mismatching letter can be removed without changing the reachability of $q_2$. Moreover, any loop in the factor $\hat{y}_2$ can be removed as well. Note that $z, x, y_1$ are nonempty by construction. By unfolding the synchronized loop, we can assume without loss of generality that all $u$ and $\hat{y}_2$ are nonempty. Hence, all words appearing in Figure 3 are nonempty. From $\hat{y}_2$, we define $y_2$ as $\hat{y}_2(y_1\hat{y}_2)^{|x|-1}$. This implies that $|y_1 y_2| = m|x|$ for some $m \in \{1, \ldots, 3n^2 + 1\}$. Furthermore, we can assume that $|z| = |y_1 y_2|$, otherwise we lengthen $y_2$ and $z$ by pumping the synchronized loop over $z$. In particular $|z| = |y_1 y_2| = m|x|$ for some $m \in \{1, \ldots, 4n^4\}$.

Now we distinguish the following two cases. First, if $z \neq y_1 y_2$, then we define $u_1 = u$, $u_2 = z$, $u_3 = \varepsilon$, and $v = y_1 y_2$. Second, if $z = y_1 y_2$, then we define $u_1 = u$, $u_2 = y_1$, $u_3 = y_2$, and $v = x$. Either way, $u_1, u_2, u_3, v$ allow $\mathcal{T}$ to comply with the pattern in Figure 2a. ◀

▶ **Lemma E.1.** *Consider a QWA $\mathcal{A} = (g, f, \mathcal{T})$ with $n$ states, $f \in \{\mathsf{LimInf}, \mathsf{LimSup}\}$, $g \in \{\mathsf{Inf}, \mathsf{Sup}, \mathsf{LimInf}, \mathsf{LimSup}\}$. Given $x \in \mathbb{Q}$, we can construct in polynomial time a Büchi automaton $\mathcal{B}$ such that (i) $L(\mathcal{B}) = \{w \in \Sigma^\omega \mid \mathcal{A}(w) = x\}$ and (ii) for every word $w$, the automaton $\mathcal{B}$ admits infinitely many accepting runs over $w$ iff $\mathcal{A}$ admits infinitely many runs of value $x$ over $w$.*

**Proof.** We start by constructing the Büchi automaton $\mathcal{B}_1$ that captures the runs visiting infinitely a transition of weight $x$. To do so, we consider two copies of $\mathcal{T}$ (the transition system of $\mathcal{A}$) such that: (1.i) the states of the first copy are all accepting, (1.ii) firing any transition from the first copy leads to the second copy, (2.i) the states of the second copy are all non-initial and non-accepting, (2.ii) firing a transition from the second copy leads to the first copy if and only if its weight is $x$. Essentially, $\mathcal{B}_1$ visits an accepting state only when a transition weighted $x$ is fired.

Then, from $\mathcal{B}_1$, we construct the Büchi automaton $\mathcal{B}_2$ that captures the runs of value exactly $x$. To do so, we consider three copies of $\mathcal{B}_1$ such that: (1.i) the states of the first copy are all non-accepting, (1.ii) firing any transition from the first copy leads nondeterministically to the second or the third copy, (2.i) the states of the second copy are all non-initial and non-accepting, (2.ii) firing a transition from the second copy leads to the first copy if and only if its weight is respectively larger or lower than $x$ when $f = \mathsf{LimSup}$ or $f = \mathsf{LimInf}$ (otherwise it stays in the second copy), (3.i) the states of the third copy are all accepting and non-initial, (3.ii) all transitions weighted more than $x$ are removed from the third copy, and the others are unchanged. Essentially, $\mathcal{B}_2$ reaches the copy with accepting states (the third one) only once no more transitions dismissing the value $x$ can be fired.

Observe that there is a bijection between the accepting runs of $\mathcal{B}_2$ and the runs of value $x$ in $\mathcal{A}$. It follows from the fact that $\mathcal{B}_1$ jumps between the copies of $\mathcal{A}$ deterministically and the nondeterministic choices of $\mathcal{B}_2$ in its first copy are unambiguous (in the following sense for $f = \mathsf{LimSup}$: jumping to the second copy and not seeing a weight larger than $x$ in $\mathcal{A}$ and jumping to the third copy and seeing a weight larger than $x$ both result in a rejecting run). Note that this does not hold for rejecting runs due to the nondeterminism of $\mathcal{B}_2$. ◀

**Lemma 7.3.** Consider a QWA $\mathcal{A} = (g, f, \mathcal{T})$ with $n$ states, $f \in \{\mathsf{LimInf}, \mathsf{LimSup}\}$, $g \in \{\mathsf{Inf}, \mathsf{Sup}, \mathsf{LimInf}, \mathsf{LimSup}\}$, and initial state $q_I$. Let $\sim\, = \,\leq$ if $f = \mathsf{LimSup}$ and $\sim\, = \,\geq$ if $f = \mathsf{LimInf}$. For every $x \in \mathbb{Q}$ and every lasso word $w \in \Sigma^\omega$, we have that $w$ admits infinitely many runs of value $x$ in $\mathcal{A}$ iff $\mathcal{A}$ complies with the pattern in Figure 2b parameterized by $x$, $n$, $\sim$, and $w$.

**Proof.** We first prove the backward implication. Given a weight $x$ and a lasso word $w \in \Sigma^\omega$, we assume that $\mathcal{A}$ complies with the pattern given in Figure 2b exhibiting four runs of the form $\rho_1 : q_I \xrightarrow{u_1} q_1$, $\ell_1 : q_1 \xrightarrow{v_1 a v_2} q_1$, $\rho_2 : q_1 \xrightarrow{u_2} q_2$ and $\ell_2 : q_2 \xrightarrow{v_1 a v_2} q_2$ where $q_1 \neq q_2$, $w = u(v_1 a v_2)^\omega$, and $(u_2)^m = v_1 a v_2$ for some $m \in \{1, \dots, 4n^4\}$. For all $i \in \mathbb{N}$, we define the run $\pi_i = \rho_1 (\ell_1)^i \rho_2 (\ell_2)^\omega$ in $\mathcal{A}$. Since $(u_2)^m = v_1 a v_2$, the run $\pi_i$ is over $w$ and achieves the value $x$ thanks to the definition of $\sim$. As a direct consequence of $q_1 \neq q_2$, we have that $\pi_i \neq \pi_j$ for all $i \neq j$. In particular, $w$ have infinitely many runs of value $x$ in $\mathcal{A}$.

Next, we prove the forward implication. Given a weight $x$ and a lasso word $w \in \Sigma^\omega$, we assume that $w$ has infinitely many runs of value $x$ in $\mathcal{A}$. Let $\tilde{u} \in \Sigma^*$ and $\tilde{v} \in \Sigma^* \setminus \{\varepsilon\}$ be such that $w = \tilde{u}(\tilde{v})^\omega$.

In the first step, we construct a Büchi automaton $\mathcal{B}$ that captures the runs over $w$ of value $x$ in $\mathcal{A}$. Thanks to Lemma E.1, we construct the Büchi automaton $\mathcal{B}'$ such that (i) $L(\mathcal{B}') = \{w \in \Sigma^\omega \mid \mathcal{A}(w) = x\}$ and (ii) for every word $w$, the automaton $\mathcal{B}'$ admits infinitely many accepting runs over $w$ iff $\mathcal{A}$ admits infinitely many runs of value $x$ over $w$. Finally, from $\mathcal{B}'$, we construction the Büchi automaton $\mathcal{B}$ that captures the runs over $w$ of value $x$ in $\mathcal{A}$. To do so, we consider a deterministic Büchi automaton $\mathcal{B}''$ accepting only $w$ and we construct $\mathcal{B}$ such that $L(\mathcal{B}) = L(\mathcal{B}') \cap L(\mathcal{B}'')$. Observe that there is a bijection between the accepting runs of $\mathcal{B}$ and the runs over $w$ of value $x$ in $\mathcal{A}$. This does not hold for non-accepting runs due to the nondeterminism of $\mathcal{B}'$.

The second step consists of lifting runs to words and leveraging Lemma 7.1 to get the desired pattern. We thus construct a Büchi automaton $\mathcal{C}$ from $\mathcal{B}$ by re-labeling each transition $p_1 \xrightarrow{\sigma} p_2$ by $p_1 \xrightarrow{(p_1,\sigma,p_2)} p_2$. Since $w$ has infinitely many runs in $\mathcal{A}$ and $w \in L(\mathcal{B})$, it has infinitely many accepting runs in $\mathcal{C}$. By Lemma 7.1, some accepting state of $\mathcal{C}$ allows $\mathcal{A}$ to comply with the pattern given in Figure 2a. Consequently there exist $u_1, u_2, u_3, v \in \Sigma^+$ such that $|u_2| = |v|$, $u_2 \neq v$, and $m|v| = |u_2 u_3|$ for some $m \in \{1, \dots, 4n^4\}$, together with four runs of the form $\rho_1 : q_I \xrightarrow{u_1} q_1$, $\ell_1 : q_1 \xrightarrow{u_2 u_3} q_1$, $\rho_2 = q_1 \xrightarrow{v} q_2$, $\ell_2 : q_2 \xrightarrow{u_2 u_3} q_2$ in $\mathcal{C}$. Observe that $u_1, u_2, u_3, v$ are not over $\Sigma^*$ but over the alphabet consisting of the transitions of $\mathcal{B}$, called $\Gamma$ here after. For all finite words $w_0 \in \Gamma^*$ we denote $\Sigma(w_0) \in \Sigma^*$ the projection of all letters of $\Gamma$ to $\Sigma$, e.g., $\Sigma((p_1, \sigma_1, p_2)(p_2, \sigma_2, p_3)) = \sigma_1 \sigma_2$. We established that $\mathcal{C}$ admits infinitely many ultimately periodic words of $\Gamma^\omega$ visiting $q_2$ infinitely often, and thus $\mathcal{B}$ admits infinitely many ultimately periodic runs (over its singleton language $\{w\}$). In particular, the set $\{u_1(u_2u_3)^i v(u_2u_3)^\omega \mid i \in \mathbb{N}\}$ of ultimately periodic words with a run visiting $q_2$ infinitely often is infinite. However, $\{\Sigma(u_1(u_2u_3)^i v)(\Sigma(u_2u_3))^\omega \mid i \in \mathbb{N}\} = \{w\}$. It is worth emphasizing that despite $u_2 \neq v$ we have $\Sigma(u_2) = \Sigma(v)$ because $L(\mathcal{B}) = \{w\}$.

In the third step, we prove how to ensure $q_1 \neq q_2$. If in the above construction we obtain $q_1 = q_2$, called $q$ here after, then we can construct $u_1', u_2', u_2'', u_3', u_3'', v'$ to get the desired property as follows. We identified $\rho_1 : q_I \xrightarrow{u_1} q$, $\rho_2 : q \xrightarrow{v} q$ and $\ell : q \xrightarrow{u_2u_3} q$ in $\mathcal{C}$. We recall that Lemma 7.1 ensures $|u_2| = |v|$ and $u_2 \neq v$. Since $\Sigma(u_2) = \Sigma(v)$, the mismatch is caused by the states of some letter of $\Gamma$. Thus, there are two distinct states $q_1' \neq q_2'$ of $\mathcal{C}$ such that $q \xrightarrow{y_1} q_1'$, $q_1' \xrightarrow{y_2} q$, $q \xrightarrow{z_1} q_2'$, $q_2' \xrightarrow{z_2} q$ in $\mathcal{C}$, where $v = y_1y_2$, $u_2 = z_1z_2$, $|y_1| = |z_1|$, and $|y_2| = |z_2|$. Let $u_1' = u_1y_1$, $u_2' = y_2u_2u_3$, $u_2'' = z_2u_3v$, $u_3' = y_1$, $u_3'' = z_1$, and $v' = y_2vz_1$ together with the four runs of the form $\rho_1' : q_I \xrightarrow{u_1'} q_1'$, $\ell_1' : q_1' \xrightarrow{u_2'u_3'} q_1'$, $\rho_2' : q_1' \xrightarrow{v'} q_2'$, $\ell_2' : q_2' \xrightarrow{u_2''u_3''} q_2'$. For all $i \in \mathbb{N}$, we define the run $\pi_i = \rho_1'(\ell_1')^i \rho_2'(\ell_2')^\omega$ over $w$ in $\mathcal{C}$. Observe that $\pi_i = \rho_1(\rho_2\ell)^i \rho_2\rho_2(\ell\rho_2)^\omega$ because $q_1 = q_2 = q$. In particular, the set $\{u_1'(u_2'u_3')^i v'(u_2''u_3'')^\omega \mid i \in \mathbb{N}\}$ of ultimately periodic words with a run visiting $q_2'$ infinitely often is infinite. We now prove that new pattern complies with Figure 2b. Since $\Sigma(u_2) = \Sigma(v)$, $|y_1| = |z_1|$, and $|y_2| = |z_2|$, we have that $\Sigma(y_1) = \Sigma(z_1)$ and $\Sigma(y_2) = \Sigma(z_2)$. Hence, $\Sigma(u_2'u_3') = \Sigma(u_2''u_3'')$. Additionally, $\Sigma(v)^m = \Sigma(u_2'u_3')$ for some $m \in \{1, \ldots, 4n^4\}$ comes as a consequence of $m|v| = |u_2u_3|$ for some $m \in \{1, \ldots, 4n^4\}$.

Finally, we decompose $u_2u_3$ into $v_1av_2$. We identified $\rho_1 : q_I \xrightarrow{u_1} q_1$, $\ell_1 : q_1 \xrightarrow{u_2u_3} q_1$, $\rho_2 : q_1 \xrightarrow{v} q_2$, and $\ell_2 : q_2 \xrightarrow{u_2u_3} q_2$ where $q_1 \neq q_2$ in $\mathcal{B}$. Recall that there is a bijection between the accepting runs of $\mathcal{B}$ and the runs over $w$ of value $x$ in $\mathcal{A}$. By construction of $\mathcal{B}$, the accepting run $u_1v(u_2u_3)^\omega$ of $\mathcal{B}$ (which is a word accepted by $\mathcal{C}$) corresponds to a run over $w$ of value $x$ in $\mathcal{A}$. Therefore, the loop $\ell_2 : q_2 \xrightarrow{u_2u_3} q_2$ corresponds to a loop visiting the weight $x$ and other weights of value at most (resp. at least) $x$ when $f = \mathsf{LimSup}$ (resp. $f = \mathsf{LimInf}$). To conclude, there exist $v_1, v_2 \in \Sigma^*$ and $a \in \Sigma$ such that $u_2u_3 = v_1av_2$ together with the runs $q_2 \xrightarrow{v_1:\sim x} q_3$, $q_4 \xrightarrow{v_2:\sim x} q_2$ and the transition $q_3 \xrightarrow{a:x} q_4$. ◀

**Proposition 7.5.** Consider a QLA $\mathbb{A} = (h, (g, f, \mathcal{T}))$ with $f \in \{\mathsf{Inf}, \mathsf{Sup}\}$, $g \in \{\mathsf{Inf}, \mathsf{Sup}, \mathsf{LimInf}, \mathsf{LimSup}\}$, and $h$ any language aggregator function. We can construct in PTime a QLA $\mathbb{B} = (h, (g, f', \mathcal{T}'))$ with $f' \in \{\mathsf{LimInf}, \mathsf{LimSup}\}$ such that $\mathbb{A}(S) = \mathbb{B}(S)$ for all $S \subseteq \Sigma^\omega$.

**Proof.** A careful observation shows that the constructions of [6, 15] allowing to convert a word automaton to a $\mathsf{LimSup}$ one extends to language automata. This proof provides the correctness of the construction from [6] in the context of language automata.

Consider a language automaton $\mathbb{A} = (h, (g, \mathsf{Sup}, \mathcal{T}))$ where $\mathcal{T} = (\Sigma, Q, q_I, \delta)$ with $\gamma$ the weight function of $\mathcal{T}$, the word aggregator function $g \in \{\mathsf{Inf}, \mathsf{Sup}, \mathsf{LimInf}, \mathsf{LimSup}\}$, and $h$ can be any language aggregator function. Let $\mathcal{A} = (g, \mathsf{Sup}, \mathcal{T})$. The idea is to construct an equivalent word automaton $\mathcal{A}' = (g, f', \mathcal{T}')$ where $f' \in \{\mathsf{LimInf}, \mathsf{LimSup}\}$ and transition system $\mathcal{T}'$ memorizes the maximal visited weight, and thus be used to construct a language automaton where runs are aggregated indifferently with $\mathsf{Sup}$, $\mathsf{LimInf}$, or $\mathsf{LimSup}$. Let $X$ be the set of weights of $\mathcal{A}$. Since $|X| < \infty$, we can fix the minimal weight $X_0 = \min X$. We construct $\mathcal{T}' = (\Sigma, Q \times X, (q_I, X_0), \delta')$ where $(q_1, v) \xrightarrow{a:x} (q_2, x)$ is a transition in $\mathcal{T}'$ iff $q_1 \xrightarrow{a:v'} q_2$ in $\mathcal{T}$ where $x = \max\{v, v'\}$. The construction of $\mathcal{T}'$ can be done in PTime in the size of $|\mathcal{T}|$. Observe that, by construction, every run $\pi$ of $\mathcal{T}'$ yields a non-decreasing weight sequence for which there exists $i \in \mathbb{N}$ such that for all $j \geq i$ we have $\gamma(\pi[i]) = \gamma(\pi[j]) = \sup(\gamma(\pi))$. Therefore, $\mathcal{A}$ and $\mathcal{A}'$ are equivalent regardless of the choice of $f' \in \{\mathsf{LimInf}, \mathsf{LimSup}\}$. Moreover, again by construction, there is a bijection between their runs, implying that for all $w \in \Sigma^\omega$ the number of runs over $w$ is the same in $\mathcal{A}$ and $\mathcal{A}'$. Now, let $\mathbb{A}' = (h, \mathcal{A}')$ and observe that since $\mathcal{A}$ and $\mathcal{A}'$ are equivalent, so are $\mathbb{A}$ and $\mathbb{A}'$. The construction for a given automaton with $f = \mathsf{Inf}$ is dual as it consists in memorizing the minimal visited weight, therefore the weight sequences are non-increasing. ◀

**Lemma 7.7.** Let $\mathbb{A} = (h, (g, f, \mathcal{T}))$ be a QLA with $f \in \{\mathsf{Inf}, \mathsf{Sup}, \mathsf{LimInf}, \mathsf{LimSup}\}$, $g = \mathsf{LimSup}$ (resp. $g = \mathsf{LimInf}$), and $h$ any language aggregator function. We can construct in PSPACE a QLA $\mathbb{B} = (h, (g', f', \mathcal{T}'))$ with $f' = \mathsf{LimSup}$ and $g' = \mathsf{Sup}$ (resp. $f' = \mathsf{LimInf}$ and $g' = \mathsf{Inf}$) such that $\mathbb{A}(S) = \mathbb{B}(S)$ for all $S \subseteq \Sigma^\omega$.

**Proof.** Consider a language automaton $\mathbb{A} = (h, \mathcal{A})$ where $\mathcal{A} = (g, f, \mathcal{T})$ is as in the statement. We show the case $g = \mathsf{LimSup}$, as the case $g = \mathsf{LimInf}$ can be solved by duality (Proposition 2.2). Thanks to Proposition 7.5, we can assume that $f \in \{\mathsf{LimInf}, \mathsf{LimSup}\}$. The proof describes the construction of a QWA $\mathcal{B} = (\mathsf{Sup}, f, \mathcal{T}')$ such that $\mathcal{A}(w) = \mathcal{B}(w)$ for every ultimately periodic word $w \in \Sigma^\omega$, and argue why this equivalence generalizes to all words.

Recall that, by definition (when $g = \mathsf{LimSup}$), $\bot_\mathcal{A}$ is the smallest weight $x$ in $\mathcal{T}$ such that there is a word $w$ with infinitely many runs of value $x$ in $\mathcal{T}$ and at most finitely many runs of value larger than $x$ in $\mathcal{T}$. Therefore, we can compute $\bot_\mathcal{A}$ as follows: Take a weight $x$ in $\mathcal{T}$ and construct in PTIME a Büchi automaton $\mathcal{D}_x$ from the transition system $\mathcal{T}$, recognizing exactly the set of words with some run of value $x$ in $\mathcal{T}$. Then, using [18,29], we can decide in PTIME if $\mathcal{D}_x$ has an infinite degree of ambiguity, i.e., some word in its language has infinitely many accepting runs (namely, runs of value $x$ in $\mathcal{T}$). Let $Y$ be the set of weights $x$ such that $\mathcal{D}_x$ has an infinite degree of ambiguity. Then, $\bot_\mathcal{A}$ is the largest $y \in Y$ such that the automaton $\mathcal{D} = (\mathsf{Sup}, f, \mathcal{T})$ is $\geq$-universal for $y$, which can be computed in PSPACE. We can compute $\bot_\mathcal{A}$ similarly when $g = \mathsf{LimInf}$. Now, let $X$ be the set of all weights in $\mathcal{T}$. By Corollary 7.4, for all $x \in X$, we can construct in PTIME a Büchi (if $f = \mathsf{LimSup}$) or a co-Büchi (if $f = \mathsf{LimInf}$) automaton $\mathcal{C}_x$ recognizing all ultimately periodic words admitting infinitely many runs of value $x$ in $\mathcal{A}$. From $\mathcal{C}_x$ we construct in PTIME the QWA $\mathcal{A}'_x = (\mathsf{Sup}, f, \mathcal{T}_x)$ such that $\mathcal{A}'_x(w) = x$ when $w \in L(\mathcal{C}_x)$ and $\mathcal{A}'_x(w) = \bot_{\mathcal{A}'}$ when $w \notin L(\mathcal{C}_x)$. The construction consists in changing the transition weight 1 to $x$ and 0 to $\bot_\mathcal{A}$. The QWA $\mathcal{B} = (\mathsf{Sup}, f, \mathcal{T}')$ is defined by $\mathcal{B} = \max_{x \in X} \mathcal{A}'_x$, effectively computable in PTIME [14].

For every ultimately periodic word $w \in \Sigma^\omega$, the value $\mathcal{A}(w)$ corresponds to the maximal value $x \in X$ such that there exist infinitely many runs of value $x$ for $w$ in $\mathcal{T}$. Let $X_w \subseteq X$ denote the set of values $x$ for which $w$ admits infinitely many runs in $\mathcal{T}$; equivalently, $x \in X_w$ iff $w \in L(C_x)$. By construction, we have $\mathcal{A}'_x(w) = x$ for each $x \in X_w$ and $\mathcal{A}'_x(w) = \bot_{\mathcal{A}'}$ for each $x \in X \setminus X_w$. If $X_w \neq \emptyset$, it follows that $\mathcal{A}(w) = \max X' = \max_{x \in X'} \mathcal{A}'_x(w) = \mathcal{B}(w)$. Otherwise, if $X_w = \emptyset$, we have $\mathcal{A}(w) = \bot_{\mathcal{A}'} = \mathcal{B}(w)$.

Next, we argue that if $\mathcal{A}(w) = \mathcal{B}(w)$ for every ultimately periodic word $w \in \Sigma^\omega$, then $\mathcal{A}(w) = \mathcal{B}(w)$ for every word $w \in \Sigma^\omega$. We prove the contrapositive. Suppose the existence of a word $\hat{w}$ such that $\mathcal{A}(\hat{w}) \neq \mathcal{B}(\hat{w})$. Clearly, we have $\mathcal{A}(\hat{w}) \in X$ and $\mathcal{B}(\hat{w}) \in X$. Using Lemma E.1, we construct a Büchi automaton $\mathcal{A}'_{\hat{w}}$ such that (i) $L(\mathcal{A}'_{\hat{w}}) = \{w \in \Sigma^\omega \mid \mathcal{A}'(w) = \mathcal{A}'(\hat{w})\}$ and (ii) for every word $w$, the automaton $\mathcal{A}'_{\hat{w}}$ admits infinitely many accepting runs over $w$ iff $\mathcal{A}'$ admits infinitely many runs of value $\mathcal{A}'(\hat{w})$ over $w$. Similarly, we construct a Büchi automaton $\mathcal{B}_{\hat{w}}$ such that (i) $L(\mathcal{B}_{\hat{w}}) = \{w \in \Sigma^\omega \mid \mathcal{B}(w) = \mathcal{B}(\hat{w})\}$ and (ii) for every word $w$, the automaton $\mathcal{B}_{\hat{w}}$ admits infinitely many accepting runs over $w$ iff $\mathcal{B}$ admits infinitely many runs of value $\mathcal{B}(\hat{w})$ over $w$. Thanks to [29], we can construct from $\mathcal{B}_{\hat{w}}$ an unambiguous Büchi automaton $\mathcal{B}'_{\hat{w}}$ recognizing the same language. The cross product between $\mathcal{A}'_{\hat{w}}$ and $\mathcal{B}'_{\hat{w}}$ yields a Büchi automaton $\mathcal{C}$ such that (i) $L(C) = L(\mathcal{B}_{\hat{w}}) \cap L(\mathcal{A}'_{\hat{w}})$ and (ii) for every word $w$, the automaton $\mathcal{C}$ admits infinitely many accepting runs over $w$ iff $\mathcal{A}'$ admits infinitely many runs of value $\mathcal{A}'(\hat{w})$ over $w$. Observe that $\hat{w} \in L(\mathcal{C})$ and for all $w \in L(\mathcal{C})$ we have $\mathcal{A}'(w) = \mathcal{A}'(\hat{w}) \neq \mathcal{B}(\hat{w}) = \mathcal{B}(w)$. By [29,35], we can decide whether $\mathcal{C}$ admits infinitely many accepting runs on some word. Furthermore, it follows from [29,35] that if $\mathcal{C}$ admits infinitely many accepting runs on a word, then it recognizes an ultimately periodic word $\tilde{w}$ with infinitely many accepting runs. In this case, $\mathcal{A}'$ admits infinitely many runs of value $\mathcal{A}'(\hat{w})$ over $\tilde{w}$, implying that $\mathcal{A}(\tilde{w}) = \mathcal{A}'(\tilde{w}) = \mathcal{A}'(\hat{w})$. Otherwise, we have $\mathcal{A}(\hat{w}) = \bot_{\mathcal{A}'} = \mathcal{A}'(\hat{w})$, and since $L(\mathcal{C}) \neq \emptyset$, there exists an ultimately periodic word $\tilde{w}$ for which $\mathcal{A}'(\tilde{w}) = \bot_{\mathcal{A}'}$. In either scenario, since $\tilde{w} \in L(\mathcal{C})$, we have $\mathcal{B}(\tilde{w}) = \mathcal{B}(\hat{w})$. Therefore, we conclude that there is an ultimately periodic word $\tilde{w}$ satisfying $\mathcal{A}'(\tilde{w}) \neq \mathcal{B}(\tilde{w})$. ◀

**Lemma 7.8.** Consider a QLA $\mathbb{A} = (h, (g, f, \mathcal{T}))$ with $f$ any run aggregator function, $g = \mathsf{Sup}$ (resp. $g = \mathsf{Inf}$), and $h$ any language aggregator function. We can construct in PTime a QLA $\mathbb{B} = (h, (g', f, \mathcal{T}')$ with $g' = \mathsf{LimSup}$ (resp. $g' = \mathsf{LimInf}$) such that $\mathbb{A}(S) = \mathbb{B}(S)$ for all $S \subseteq \Sigma^\omega$.

**Proof.** The transition system $\mathcal{T}'$ consists of two copies of $\mathcal{T}$ that are connected by nondeterministic transitions, allowing for every run in $\mathbb{A}$ infinitely many runs in $\mathbb{B}$. In particular, for every transition $q \xrightarrow{\sigma:x} p$ in $\mathcal{T}$, there are three transitions in $\mathcal{T}'$ as follows: $q \xrightarrow{\sigma:x} p$ copying the original transition, $q \xrightarrow{\sigma:x} p'$ allowing to jump from the first copy of the transition system to the second, and $q' \xrightarrow{\sigma:x} p'$ allowing to stay in the second copy. Evidently, for every run $\rho$ in $\mathcal{T}$ and every $i \in \mathbb{N}$, there is a run $\rho'_i$ in $\mathcal{T}'$ such that $\rho'_i$ imitates $\rho$ in the first copy in the first $i$ transitions and in the second copy afterwards. Moreover, the values of $\rho'_i$ and $\rho$ coincide as they yield the same weight sequences. Since the word automata $(g, f, \mathcal{T})$ and $(g', f, \mathcal{T}')$ are equivalent, so are $\mathbb{A}$ and $\mathbb{B}$. ◀

**Proposition 7.10.** QLAs with the language aggregators $\mathsf{Inf}$ and $\mathsf{LimInf}$ (resp. $\mathsf{Sup}$ and $\mathsf{LimSup}$) are expressively incomparable.

**Proof.** Consider a QLA $\mathbb{A} = (\mathsf{Inf}, \mathcal{A})$ over the value domain $\{0, 1\}$ where $\mathcal{A}$ maps a single word $w$ to 0 and every other word to 1. Then, for every language $L$, we have $\mathbb{A}(L) = 0$ iff $w \in L$. Assume the existence of a QLA $\mathbb{B} = (\mathsf{LimInf}, \mathcal{B})$ that is equivalent to $\mathbb{A}$. By construction, $\mathbb{A}(\{w\}) = 0$ but since the language is finite, we have $\mathbb{B}(\{w\}) = 1$, yielding a contradiction.

Consider a QLA $\mathbb{A} = (\mathsf{LimInf}, \mathcal{A})$ over the value domain $\{0, 1\}$ where $\mathcal{A}$ maps all words to 0. Then, for every language $L$, we have $\mathbb{A}(L) = 0$ iff $|L| = \infty$. Assume the existence of a QLA $\mathbb{B} = (\mathsf{Inf}, \mathcal{B})$ that is equivalent to $\mathbb{A}$. By definition, for every language $L$ we have $\mathbb{B}(L) = \mathsf{Inf}_{w \in L} \mathcal{B}(w)$, meaning that $\mathcal{B}$ maps all words $w \in L$ to 0 iff $L$ is infinite. Intuitively, this is not possible since $\mathcal{B}$ only has a "local" view, i.e., values of words do not depend on other words they are given together with. Formally, consider the infinite language $\Sigma^\omega$. By assumption, we have $\mathbb{A}(\Sigma^\omega) = \mathbb{B}(\Sigma^\omega) = 0$. Notice that $\mathbb{B}(\Sigma^\omega) = \mathsf{Inf}_{w \in L} \mathcal{B}(w) = 0$ implies the existence of a word $w$ with $\mathcal{B}(w) = 0$. Then, $\mathbb{B}(\{w\}) = 0$ although $\mathbb{A}(\{w\}) = 1$, yielding a contradiction.

The case of $\mathsf{Sup}$ and $\mathsf{LimSup}$ can be proved similarly. ◀

**Theorem 7.11.** Consider a QLA $\mathbb{A} = (h, (g, f, \mathcal{T}))$ with $f, g, h \in \{\mathsf{Inf}, \mathsf{Sup}, \mathsf{LimInf}, \mathsf{LimSup}\}$ and at least one $g$ and $h$ belong to $\{\mathsf{LimInf}, \mathsf{LimSup}\}$. Let $S \subseteq \Sigma^\omega$ be an $\omega$-regular language given by a Büchi automaton. The value $\mathbb{A}(S)$ is computable in PSpace.

**Proof.** Consider a QLA $\mathbb{A} = (h, (g, f, \mathcal{T}))$ with $f, g, h \in \{\mathsf{Inf}, \mathsf{Sup}, \mathsf{LimInf}, \mathsf{LimSup}\}$. Let $X$ be the set of weights of $\mathcal{T}$. We assume that $g \in \{\mathsf{Inf}, \mathsf{Sup}\}$ thanks to Lemma 7.7. Further, we only consider the case when $g = \mathsf{Sup}$. By the duality property of Proposition 2.2, the other case can be solved by evaluating $\hat{\mathbb{A}}$ instead and multiplying the result by $-1$. Thanks to [15, Thm 13. (iv)], we can assume that $f = \mathsf{LimSup}$. It is worth emphasizing that the transformation of [15] must be performed once we have $g = \mathsf{Sup}$. Since $f = \mathsf{LimSup}$, we can compute $\top_{\mathcal{A}}$ in PTime [15] and $\bot_{\mathcal{A}}$ in PSpace [22]. Let $\mathcal{B}_S$ be the Büchi automaton recognizing $S \subseteq \Sigma^\omega$. Next, we define $L(\mathcal{A}_{\sim x}) = \{w \in \Sigma^\omega \mid \mathcal{A}(w) \sim x\}$ where $x \in \mathbb{Q}$ is a rational threshold and $\sim \in \{=, >, \geq\}$. When $f \in \{\mathsf{Inf}, \mathsf{Sup}, \mathsf{LimInf}, \mathsf{LimSup}\}$ and $g = \mathsf{Sup}$, we can construct a Büchi automaton recognizing $L(\mathcal{C}_{>x})$ (resp., $L(\mathcal{C}_{\geq x})$) in linear time [15]. Thus, we leverage Corollary 7.2 to compute in PSpace the set of weights $Y \subseteq X$ such that $x \in Y$ if and only if $(L(\mathcal{B}_S) \cap L(\mathcal{A}_{\geq x})) \setminus L(\mathcal{A}_{>x})$ is infinite. We define the set of weights $Z$ as $Z = X$ if $h \in \{\mathsf{Inf}, \mathsf{Sup}\}$, as $Z = Y \cup \{\bot_{\mathcal{A}}\}$ if $h = \mathsf{LimSup}$, and as $Z = Y \cup \{\top_{\mathcal{A}}\}$ if $h = \mathsf{LimInf}$.

Now, we distinguish two cases. When $h \in \{\mathsf{Sup}, \mathsf{LimSup}\}$, the evaluation procedure finds the largest weight of $\tilde{x} \in X$ such that for some $w \in L(\mathcal{B}_S)$ we have $\mathbb{A}(w) = \tilde{x}$. Equivalently, $\tilde{x}$ is the largest weight of $x \in Z$ for which $L(\mathcal{B}_S) \cap L(\mathcal{A}_{\geq x})$ is not empty. Hence, $\tilde{x}$ can be identified in PSpace $h = \mathsf{LimSup}$ and in PTime when $h = \mathsf{Sup}$ since the set $Y$ then does not need to be computed and the nonemptiness check above is in PTime. When $h \in \{\mathsf{Inf}, \mathsf{LimInf}\}$, the evaluation procedure finds the smallest weight of $\tilde{x} \in X$ such that for all $w \in L(\mathcal{B}_S)$ we have $\mathbb{A}(w) = \tilde{x}$. Equivalently, $\tilde{x}$ is the smallest weight of $x \in Z$ for which $L(\mathcal{B}_S) \subseteq L(\mathcal{A}_{\geq x})$ holds. Hence, $\tilde{x}$ can be identified in PSpace in this case. ◀

**Theorem 7.12.** Consider two QLAs $\mathbb{A} = (h, (g, f, \mathcal{T}))$ and $\mathbb{B} = (h', (g', f', \mathcal{T}'))$ with $f, f', g, g', h, h' \in \{\mathsf{Inf}, \mathsf{Sup}, \mathsf{LimInf}, \mathsf{LimSup}\}$. Let $\triangleright \in \{>, \geq\}$. Deciding whether $\mathbb{A}(S) \triangleright \mathbb{B}(S)$ for every language $S \subseteq \Sigma^\omega$ is in PSpace. The same holds when $S$ ranges over $\omega$-regular languages.

**Proof.** The inclusion problem of QWA with $f_{\mathbb{A}}, f_{\mathbb{B}} \in \{\mathsf{Inf}, \mathsf{Sup}, \mathsf{LimInf}, \mathsf{LimSup}\}$ and $g \in \{\mathsf{Inf}, \mathsf{Sup}\}$ reduces to that of $\omega$-regular languages [13, 15], and thus can be decided by reasoning exclusively about lasso words. This property extends to language automata with $g_{\mathbb{A}}, g_{\mathbb{B}} \in \{\mathsf{Inf}, \mathsf{Sup}, \mathsf{LimInf}, \mathsf{LimSup}\}$ thanks to Lemma 7.7. We now argue that it also extends to language automata with $h_{\mathbb{A}}, h_{\mathbb{B}} \in \{\mathsf{Inf}, \mathsf{Sup}, \mathsf{LimInf}, \mathsf{LimSup}\}$. Consider two QLAs $\mathbb{A}$ and $\mathbb{B}$ as in the statement. Let $X_{\mathbb{A}}, X_{\mathbb{B}}$ be the sets of weights of $\mathbb{A}$ and $\mathbb{B}$, respectively. We assume that $g_{\mathbb{A}}, g_{\mathbb{B}} \in \{\mathsf{Inf}, \mathsf{Sup}\}$ thanks to Lemma 7.7. Suppose the existence of a non-empty language $S \subseteq \Sigma^\omega$ such that $\mathbb{A}(S) \not\triangleright \mathbb{B}(S)$. We have $\mathbb{A}(S) \in X_{\mathbb{A}}$ and $\mathbb{B}(S) \in X_{\mathbb{B}}$. Let $L_{\mathbb{A}} = \{w \in \Sigma^\omega \mid \mathcal{A}(w) \leq \mathbb{A}(S)\}$, $L_{\mathbb{B}} = \{w \in \Sigma^\omega \mid \mathcal{B}(w) \geq \mathbb{B}(S)\}$ both $\omega$-regular, together with they respective complements $L_{\mathbb{A}}^{\complement}$ and $L_{\mathbb{B}}^{\complement}$. Intuitively, we aim at constructing an $\omega$-regular language $L$ that allows $\mathbb{A}$ and $\mathbb{B}$ to achieve their respective image of $S$.

The simpler cases are when exactly one of $h_{\mathbb{A}}$ and $h_{\mathbb{B}}$ belongs to $\{\mathsf{Inf}, \mathsf{Sup}\}$ and the other to $\{\mathsf{LimInf}, \mathsf{LimSup}\}$ since the value of the limit language aggregator is not affected by the addition of a single word. For instance, having $h_{\mathbb{A}} = \mathsf{LimSup}$ and $h_{\mathbb{B}} = \mathsf{Inf}$, we define $L = L_{\mathbb{A}} \cup \{w\}$ where $w$ is an ultimately periodic word witnessing that $L_{\mathbb{B}}$ is not empty.

The cases when both $h_{\mathbb{A}}$ and $h_{\mathbb{B}}$ belong to $\{\mathsf{LimInf}, \mathsf{LimSup}\}$ go as follows. If $h_{\mathbb{A}} = \mathsf{LimInf}$ and $h_{\mathbb{B}} = \mathsf{LimSup}$, we define $L = \Sigma^\omega$. Otherwise, if (i) $h_{\mathbb{A}} = h_{\mathbb{B}} = \mathsf{LimInf}$, (ii) $h_{\mathbb{A}} = h_{\mathbb{B}} = \mathsf{LimSup}$, or (iii) $h_{\mathbb{A}} = \mathsf{LimSup}$ and $h_{\mathbb{B}} = \mathsf{LimInf}$, we continue as follows. If $S$ is finite, then both automata output their respective "extreme value" and thus defining $L$ with any finite language will do. Otherwise, we let $L = L_{\mathbb{A}} \cap L_{\mathbb{B}}$ and prove that $L$ is infinite. Indeed, the infiniteness of $S$ implies the infiniteness of both $L_{\mathbb{A}} \cap S$ and $L_{\mathbb{B}} \cap S$. Also, because $h_{\mathbb{A}} = \mathsf{LimSup}$ or $h_{\mathbb{B}} = \mathsf{LimInf}$, either $S \cap L_{\mathbb{A}}^{\complement}$ or $S \cap L_{\mathbb{B}}^{\complement}$ is finite, allowing us to conclude. For instance, having $h_{\mathbb{B}} = \mathsf{LimInf}$ implies that $S \cap L_{\mathbb{B}}^{\complement}$ is finite. Since $L_{\mathbb{A}} \cap S$ is infinite but $S \cap L_{\mathbb{B}}^{\complement}$ is finite, then $L_{\mathbb{A}} \cap S = (L_{\mathbb{A}} \cap S \cap L_{\mathbb{B}}^{\complement}) \cup (L_{\mathbb{A}} \cap S \cap L_{\mathbb{B}})$ is infinite, as well as $L_{\mathbb{A}} \cap S \cap L_{\mathbb{B}}$. In particular, $L_{\mathbb{A}} \cap L_{\mathbb{B}}$ is infinite.

Finally, the cases both $h_{\mathbb{A}}$ and $h_{\mathbb{B}}$ belongs to $\{\mathsf{Inf}, \mathsf{Sup}\}$ uses a similar reasoning, replacing infinite by non-empty. If $h_{\mathbb{A}} = \mathsf{Inf}$ and $h_{\mathbb{B}} = \mathsf{Sup}$, we define $L = \Sigma^\omega$. Otherwise, if (i) $h_{\mathbb{A}} = h_{\mathbb{B}} = \mathsf{Inf}$, (ii) $h_{\mathbb{A}} = h_{\mathbb{B}} = \mathsf{Sup}$, or (iii) $h_{\mathbb{A}} = \mathsf{Sup}$ and $h_{\mathbb{B}} = \mathsf{Inf}$, we define $L = L_{\mathbb{A}} \cap L_{\mathbb{B}}$

and prove that it is non-empty. Clearly, both $L_{\mathbb{A}} \cap S$ and $L_{\mathbb{B}} \cap S$ are non-empty. Also, because $h_{\mathbb{A}} = \mathsf{Sup}$ or $h_{\mathbb{B}} = \mathsf{Inf}$, either $S \cap {L_{\mathbb{A}}}^{\complement}$ or $S \cap L_{\mathbb{B}}^{\complement}$ is empty, allowing us to conclude. For instance, having $h_{\mathbb{B}} = \mathsf{Inf}$ implies that $L_{\mathbb{B}}^{\complement}$ is empty. Since $L_{\mathbb{A}} \cap S$ is non-empty but $L_{\mathbb{B}}^{\complement}$ is empty, then $L_{\mathbb{A}} \cap S = (L_{\mathbb{A}} \cap S \cap L_{\mathbb{B}}^{\complement}) \cup (L_{\mathbb{A}} \cap S \cap L_{\mathbb{B}})$ is non-empty, as well as $L_{\mathbb{A}} \cap S \cap L_{\mathbb{B}}$. In particular, $L_{\mathbb{A}} \cap L_{\mathbb{B}}$ is non-empty.

Next we provide a decision procedure for the inclusion problem. Consider two QLAs $\mathbb{A} = (h_{\mathbb{A}}, (g_{\mathbb{A}}, f_{\mathbb{A}}, \mathcal{T}_{\mathbb{A}}))$ and $\mathbb{B} = (h_{\mathbb{B}}, (g_{\mathbb{B}}, f_{\mathbb{B}}, \mathcal{T}_{\mathbb{B}}))$ with $f_{\mathbb{A}}, f_{\mathbb{B}} \in \{\mathsf{Inf}, \mathsf{Sup}, \mathsf{LimInf}, \mathsf{LimSup}\}$. Let $X_{\mathbb{A}}, X_{\mathbb{B}}$ be the sets of weights of $\mathbb{A}$ and $\mathbb{B}$, respectively. We assume that $g_{\mathbb{A}}, g_{\mathbb{B}} \in \{\mathsf{Inf}, \mathsf{Sup}\}$ thanks to Lemma 7.7. Further, we use the duality properties of Proposition 2.2 to get an equation where all language automata have a word aggregator $\mathsf{Sup}$. The problem is then of one of the following forms.

- $\forall S : \mathbb{A}(S) \triangleright \mathbb{B}(S)$
- $\forall S : \mathbb{A}(S) + \hat{\mathbb{B}}(S) \triangleright 0$
- $\forall S : 0 \triangleright \hat{\mathbb{A}}(S) + \mathbb{B}(S)$
- $\forall S : \hat{\mathbb{B}}(S) \triangleright \hat{\mathbb{A}}(S)$

For simplicity, we reason about automata without duality and assume $g_{\mathbb{A}} = g_{\mathbb{B}} = \mathsf{Sup}$ while solving the four problems above. Thanks to [15, Thm 13. (iv)], we can assume that $f_{\mathbb{A}} = f_{\mathbb{B}} = \mathsf{LimSup}$. It is worth emphasizing that, the transformation of [15] must be performed once we have $g_{\mathbb{A}} = g_{\mathbb{B}} = \mathsf{Sup}$. Since $f_{\mathbb{A}} = f_{\mathbb{B}} = \mathsf{LimSup}$, we can compute $\top_{\mathcal{A}}$ and $\top_{\mathcal{B}}$ in PTIME [15], $\bot_{\mathcal{A}}$ and $\bot_{\mathcal{B}}$ in PSPACE [22]. Next, we define $L(\mathcal{C}_{\sim x}) = \{w \in \Sigma^\omega \mid \mathcal{C}(w) \sim x\}$ where $\mathcal{C}$ is QWA, $x \in \mathbb{Q}$ is a rational threshold, and $\sim \in \{=, >, \geq\}$. When $f \in \{\mathsf{Inf}, \mathsf{Sup}, \mathsf{LimInf}, \mathsf{LimSup}\}$ and $g = \mathsf{Sup}$, we can construct a Büchi automaton recognizing $L(\mathcal{C}_{>x})$ (respectively $L(\mathcal{C}_{\geq x})$) in linear time [15]. Thus, we leverage Corollary 7.2 to compute in PSPACE the set of weights $Y_{\mathbb{A}} \subseteq X_{\mathbb{A}}$ such that $x \in Y_{\mathbb{A}}$ if and only if $L(\mathcal{A}_{\geq x}) \setminus L(\mathcal{A}_{>x})$ is infinite. We compute $Y_{\mathbb{B}}$ similarly.

The procedure iterates over all pair of weights $x \in X_{\mathbb{A}}$ and $y \in X_{\mathbb{B}}$. First, we check whether some word $w \in \Sigma^\omega$ is such that $\mathcal{A}(w) = x$ and $\mathcal{B}(w) = y$, i.e., whether $L(\mathcal{A}_{=x}) \cap L(\mathcal{B}_{=y})$ is not empty. This can be equivalently encoded as the inclusion of Büchi automata $L(\mathcal{A}_{\geq x}) \cap L(\mathcal{B}_{\geq y}) \nsubseteq L(\mathcal{A}_{>x}) \cup L(\mathcal{B}_{>y})$ which is decidable in PSPACE. If $L(\mathcal{A}_{=x}) \cap L(\mathcal{B}_{=y}) \neq \varnothing$, we consider its the two values $x' = \mathbb{A}(L(\mathcal{A}_{=x}))$ and $y' = \mathbb{B}(L(\mathcal{B}_{=y}))$. Thus the following holds.

- $x' = x$ iff $x \in Y_{\mathbb{A}} \vee h_{\mathbb{A}} \in \{\mathsf{Inf}, \mathsf{Sup}\}$
- $x' = \top_{\mathcal{A}}$ iff $x \notin Y_{\mathbb{A}} \wedge h_{\mathbb{A}} = \mathsf{LimInf}$
- $x' = \bot_{\mathcal{A}}$ iff $x \notin Y_{\mathbb{A}} \wedge h_{\mathbb{A}} = \mathsf{LimSup}$
- $y' = y$ iff $y \in Y_{\mathbb{B}} \vee h_{\mathbb{B}} \in \{\mathsf{Inf}, \mathsf{Sup}\}$
- $y' = \top_{\mathcal{B}}$ iff $y \notin Y_{\mathbb{A}} \wedge h_{\mathbb{B}} = \mathsf{LimInf}$
- $y' = \bot_{\mathcal{B}}$ iff $y \notin Y_{\mathbb{B}} \wedge h_{\mathbb{B}} = \mathsf{LimSup}$

Additionally, we observe that for all language automaton $\mathbb{C} = (h_{\mathbb{C}}, \mathcal{C})$ have $\top_{\mathbb{C}} = \top_{\mathcal{C}}$ for $h_{\mathcal{C}} = \mathsf{LimInf}$ and $\bot_{\mathbb{C}} = \bot_{\mathcal{C}}$ for $h_{\mathcal{C}} = \mathsf{LimSup}$. This is because $h_{\mathbb{C}} = \mathsf{LimInf}$ may only increase the values of languages compared to using $\mathsf{Inf}$, while $h_{\mathbb{C}} = \mathsf{LimSup}$ may only decrease compared to $\mathsf{Sup}$. Finally, we respectively check that $x' \triangleright y$, $x' + y' \triangleright 0$, $0 \triangleright x' + y'$, or $y' \triangleright x'$ holds. Clearly, if for some $\tilde{x} \in X_{\mathbb{A}}$ and all $\tilde{y} \in X_{\mathbb{B}}$, we have $S_{\tilde{x},\tilde{y}} = L(\mathcal{A}_{=\tilde{x}}) \cap L(\mathcal{B}_{=\tilde{y}}) \neq \varnothing$ but $\tilde{x}', \tilde{y}'$ do not satisfy the equation, then $S_{\tilde{x},\tilde{y}}$ is a non-inclusion witness. Otherwise, if for all $x \in X_{\mathbb{A}}$ and all $y \in X_{\mathbb{B}}$ having $S_{x,y} = L(\mathcal{A}_{=x}) \cap L(\mathcal{B}_{=y}) \neq \varnothing$ implies that $x', y'$ satisfy the equation, then the inclusion holds for all $\bigcup_{x \in Z_{\mathbb{A}}, y \in Z_{\mathbb{B}}} S_{x,y}$ where $Z_{\mathbb{A}} \subseteq X_{\mathbb{A}}$ and $Z_{\mathbb{B}} \subseteq X_{\mathbb{B}}$, and thus for all $S \subseteq \Sigma^\omega$. ◀

**Corollary 7.13.** Consider a QLA $\mathbb{A} = (h, (g, f, \mathcal{T}))$ with $f, g, h \in \{\mathsf{Inf}, \mathsf{Sup}, \mathsf{LimInf}, \mathsf{LimSup}\}$ and at least one $g$ and $h$ belong to $\{\mathsf{LimInf}, \mathsf{LimSup}\}$. Let $\triangleright \in \{>, \geq\}$. The $\triangleright$-nonemptiness (resp. $\triangleright$-universality) of $\mathbb{A}$ is in PSPACE. The statement holds also for the finite-state restriction.

**Proof.** We provide a direct proof as an alternative.

Let $\mathbb{A} = (h, (g, f, \mathcal{T}))$ be a QLA with $f, g, h \in \{\mathsf{Inf}, \mathsf{Sup}, \mathsf{LimInf}, \mathsf{LimSup}\}$ and let $k \in \mathbb{Q}$. Thanks to Proposition 7.5 and Lemma 7.7, we can assume w.l.o.g. that $g = \mathsf{Sup}$ and $f = \mathsf{LimSup}$. Let $\mathcal{A}$ be the underlying QWA of $\mathbb{A}$.

First, we show that $\top_{\mathbb{A}} \triangleright k$ iff $\mathbb{A}$ is $\triangleright$-nonempty for $k$. For the left-to-right direction, notice that there are only finitely many values to which a language $L$ can be mapped by $\mathbb{A}$, and the value $\top_{\mathbb{A}}$ is the maximum among these values. So, if $\mathbb{A}(L) \not\triangleright k$ for all $L \subseteq \Sigma^\omega$, then $\top_{\mathbb{A}} \not\triangleright k$ as well. For the right-to-left direction, simply observe that $\top_{\mathbb{A}} \geq \mathbb{A}(L)$ by definition where $L$ is a nonemptiness witness.

Second, we describe an algorithm to compute $\top_{\mathbb{A}}$. For $h \in \{\mathsf{Inf}, \mathsf{Sup}\}$, we have $\top_{\mathbb{A}} = \top_{\mathcal{A}}$ thanks to Proposition 2.1, which can be computed in PTime when $g = \mathsf{Sup}$ and in PSpace otherwise. (Note that translating the underlying QWA to one with $g = \mathsf{Sup}$ can be also done in PSpace.) Let us now consider $h \in \{\mathsf{LimInf}, \mathsf{LimSup}\}$. Notice that $\top_{\mathbb{A}}$ is exactly the largest weight $x$ in $\mathcal{A}$ such that $L(\mathcal{A}_{=x})$ is infinite, which we can compute in PSpace thanks to Corollary 7.2.

Third, we show that the unrestricted and the finite-state nonemptiness problems coincide in this setting. To achieve this, we prove the following claim: For every language $S$, there is an $\omega$-regular language $L$ such that $\mathbb{A}(S) = \mathbb{A}(L)$. If $h \in \{\mathsf{Inf}, \mathsf{LimInf}\}$, let $L = \{w \in \Sigma^\omega \mid \mathcal{A}(w) \geq \mathbb{A}(S)\}$. For $h = \mathsf{Inf}$, it is easy to see that $\mathbb{A}(L) = \mathbb{A}(S)$—there is a word $w \in L$ with value $\mathbb{A}(S)$ since $\mathcal{A}$ has $g = \mathsf{Sup}$ and $f = \mathsf{LimSup}$, and all words in $L$ have at least this value. For $h = \mathsf{LimInf}$, the argument goes as follows. If $S$ is not finite, $\mathbb{A}(S)$ is the least weight $x$ of $\mathcal{A}$ such that there are infinitely many words $w \in S$ with $\mathcal{A}(w) = x$. By definition, $L$ contains at least all such words, so it has infinitely many words with value $x$. Moreover, $L$ does not contain any word with a smaller value, therefore $\mathbb{A}(L) = x$. If $S$ is finite, then $\mathbb{A}(S) = \top_{\mathbb{A}}$. Recall that $\top_{\mathbb{A}}$ is exactly the largest weight $x$ in $\mathcal{A}$ such that $L(\mathcal{A}_{=x})$ is infinite, so $L$ is nonempty and satisfies $\mathbb{A}(L) = \top_{\mathbb{A}}$. For $h \in \{\mathsf{Sup}, \mathsf{LimSup}\}$, let $L = \{w \in \Sigma^\omega \mid \mathcal{A}(w) \leq \mathbb{A}(S)\}$ instead. Having this, we finally conclude that $\mathbb{A}(S) \triangleright k$ for some language $S$ iff $\mathbb{A}(L) \triangleright k$ for some $\omega$-regular language $L$. ◀