# Quantified Linear Temporal Logic over Probabilistic Systems with an Application to Vacuity Checking

Jakob Piribauer, Christel Baier, Nathalie Bertrand, Ocan Sankur

# Quantified Linear Temporal Logic over Probabilistic Systems with an Application to Vacuity Checking

**Jakob Piribauer**
Technische Universität Dresden, Germany

**Christel Baier**
Technische Universität Dresden, Germany

**Nathalie Bertrand**
Université Rennes, Inria, CNRS, IRISA, France

**Ocan Sankur**
Université Rennes, Inria, CNRS, IRISA, France

## Abstract

Quantified linear temporal logic (QLTL) is an $\omega$-regular extension of LTL allowing quantification over propositional variables. We study the model checking problem of QLTL-formulas over Markov chains and Markov decision processes (MDPs) with respect to the number of quantifier alternations of formulas in prenex normal form. For formulas with $k-1$ quantifier alternations, we prove that all qualitative and quantitative model checking problems are k-EXPSPACE-complete over Markov chains and k+1-EXPTIME-complete over MDPs.

As an application of these results, we generalize vacuity checking for LTL specifications from the non-probabilistic to the probabilistic setting. We show how to check whether an LTL-formula is affected by a subformula, and also study inherent vacuity for probabilistic systems.

**2012 ACM Subject Classification** Theory of computation → Verification by model checking

**Keywords and phrases** Quantified linear temporal logic, Markov chain, Markov decision process, vacuity

**Digital Object Identifier** 10.4230/LIPIcs.CONCUR.2021.7

**Related Version** *Extended Version*: `https://wwwtcs.inf.tu-dresden.de/ALGI/PUB/CONCUR21/` [19]

**Funding** This work was funded by DFG grant 389792660 as part of TRR 248 – CPEC (see `https://perspicuous-computing.science`), the Cluster of Excellence EXC 2050/1 (CeTI, project ID 390696704, as part of Germany's Excellence Strategy), DFG-projects BA-1679/11-1 and BA-1679/12-1, and the Research Training Group QuantLA (GRK 1763).

## 1 Introduction

In the formal verification of probabilistic systems, a central problem is the *model-checking problem*: Given a system model $\mathcal{M}$ and a specification $\varphi$, decide whether the probability $\Pr_{\mathcal{M}}(\varphi)$ that $\varphi$ holds on an execution of $\mathcal{M}$ is 1 or whether it is positive, respectively, (qualitative model checking) or compute the probability $\Pr_{\mathcal{M}}(\varphi)$ (quantitative model checking). In case the system exhibits non-deterministic behavior, the model-checking problems address the worst- or best-case resolution of the non-determinism, i.e., the minimal or maximal satisfaction probability among all possible resolutions of the non-deterministic choices. Common probabilistic system models are finite-sate Markov chains that are purely probabilistic and Markov decision processes (MDPs) that also model non-deterministic behavior. Specifications can be formulated in temporal logics, such as linear temporal logic (LTL) as an important example, or be given by automata, such as non-deterministic Büchi automata (NBA). The choice of the specification formalism is a balancing act between expressive power, succinctness,


and the complexity of the respective model-checking problems. Additionally, the formalism should allow one to describe desired system behaviors in a way comprehensible to a human user as writing down the specification is itself an error-prone process in practice.

*Quantified linear temporal logic* (QLTL), introduced by Sistla [21], is an extension of LTL with quantification over propositional variables lifting the expressive power from star-free to all $\omega$-regular languages. A formula of the form $\exists x.\varphi$ holds on a word $w$ if one can choose a set of positions at which $x$ holds such that the word $w$ extended with this choice satisfies $\varphi$. The quantification hence ranges over all sets of positions, i. e., sets of natural numbers. In QLTL, LTL-formulas can be extended with the quantification over propositions that, for example, capture hidden variables or encode annotations of a trace. This can be useful if we want to define properties not expressible in LTL in a context in which one often works with LTL. Examples include definitions of refinement relations in which internal variables are quantified to express equivalence of two specifications with respect to the observable variables [14], a necessary and sufficient condition expressed as a QLTL-formula on the serializability of histories in concurrent database scheduling produced by a scheduler whose behavior is expressed by an LTL-formula [13], or the QLTL-expressible existence of finite counterexamples witnessing the unrealizability of an LTL-specification for distributed fault-tolerant systems [8]. Furthermore, the vacuous satisfaction of a specification in a transitions system indicating that parts of the specification are irrelevant for the satisfaction has been defined using QLTL [1]. We transfer this definition of vacuous satisfaction to the probabilistic setting in this paper and explain the notion of vacuity in more detail below. In all of these successful applications of QLTL to questions in formal verification, the necessary QLTL-formulas require only few quantifier alternations; often even a single block of quantifiers without alternation is sufficient.

The full logic, however, is not suitable for practical applications: the non-probabilistic model-checking problem of QLTL on transition systems has non-elementary complexity [22]. The lower bounds can be pinpointed to the different levels of quantifier alternation of formulas in prenex normal form. Model-checking of QLTL-formulas with $k-1$ quantifier alternations in transition systems is k-EXPSPACE-complete. Distinguishing whether the first block of quantifiers is existential ($\Sigma_k^{QLTL}$) or universal ($\Pi_k^{QLTL}$) refines the result as for $\Pi_k^{QLTL}$-formulas the complexity of model-checking drops to k−1-EXPSPACE-completeness [22]. The increase of the complexity by one exponential per quantifier alternation is theoretically intriguing on the one hand, and on the other hand leads to reasonable complexity results for properties that can be expressed succinctly with the use of few quantifier alternations. A similar complexity hierarchy is observed in other settings. The complexity of model checking quantified computation tree logic (CTL) with $k$ quantifier alternations is k-EXPTIME-complete, and it is $k+1$-EXPTIME-complete for quantified CTL$^*$ in the tree semantics; while in the structure semantics, these problems span the polynomial hierarchy [17]. The hardness of the fragments of QLTL [22] was used to show that model checking strategy logic is k-EXPSPACE-hard when restricted to $k$ quantifier alternations.

In this paper, we study the model-checking problem of QLTL in probabilistic systems. Our main result is that the complexity of the model-checking problems on Markov chains and MDPs match the upper bounds obtained via straight-forward automata constructions: For Markov chains and $\Sigma_k^{QLTL}$- and $\Pi_k^{QLTL}$-formulas, all model-checking problems are k-EXPSPACE-complete, while for MDPs the problems are $k+1$-EXPTIME-complete. These complexity results are summarized in Table 1. As the upper bounds are easily obtained, the main contribution lies in proving the lower bounds. The hardness proofs for Markov chains, encode a tiling problem of a k-exponentially wide rectangle with arbitrary height. For the

hardness proofs for MDPs, we encode the computation of an alternating k-exponentially space-bounded Turing machine. The alternation can be mimicked in an MDP by letting one player in the acceptance game of the alternating Turing machine be played randomly, while the scheduler takes the role of the other player. We obtain the result that the complexities of the model-checking problems for $\Sigma_k^{QLTL}$ and $\Pi_k^{QLTL}$ coincide in the probabilistic setting in contrast to the asymmetry known for the non-probabilistic setting. It is remarkable that the complexities of $\Sigma_1^{QLTL}$- and $\Pi_1^{QLTL}$-model checking in MDPs are the same as the complexity of LTL-model checking. For each further quantifier alternation, the complexity increases by one exponential. In contrast, we see an exponential increase in complexity already for the first block of quantifiers in $\Sigma_1^{QLTL}$ and $\Pi_1^{QLTL}$ compared to LTL-model checking in Markov chains.

**Table 1** Complexity results for the model-checking problems of fragments of QLTL. All entries state completeness results.

| | transition system | Markov chain | MDP |
|---|---|---|---|
| LTL | PSPACE [23, 25] | PSPACE [7] | 2-EXPTIME [7] |
| $\Pi_1^{QLTL}$ | PSPACE [22] | **EXPSPACE** | **2-EXPTIME** |
| $\Sigma_1^{QLTL}$ | EXPSPACE [22] | **EXPSPACE** | **2-EXPTIME** |
| $\Pi_k^{QLTL}$ | k-1-EXPSPACE [22] | **k-EXPSPACE** | **k+1-EXPTIME** |
| $\Sigma_k^{QLTL}$ | k-EXPSPACE [22] | **k-EXPSPACE** | **k+1-EXPTIME** |

On the one hand, knowledge of the precise complexities of the model-checking problems for $\Sigma_k^{QLTL}$- and $\Pi_k^{QLTL}$-formulas over probabilistic systems might be useful to determine the complexity of other problems in the formal verification of probabilistic systems – in particular, by using the new lower bounds provided in this paper for new hardness results. On the other hand, the upper bounds are obtained via the construction of automata. It follows easily that all investigated model-checking problems can be solved in time polynomial in the size of the model, i.e., the Markov chain or the MDP. This means that efficient model checking for low levels of the quantifier alternation hierarchy of QLTL might be possible in many application areas despite the high complexities of the model-checking problems because formulas are typically small compared to the size of the models.

As an application of our main results, we extend the definition of vacuous satisfaction of a specification from [1] to the probabilistic setting. For an illustration of vacuous satisfaction, consider the specification: "Whenever a request is sent, it is eventually granted." If in a system model no requests are ever sent, the specification is satisfied and a model checker would report this result. However, something is obviously wrong with either the specification or – in this case more likely – the system model. We say the specification is vacuously true. The formal definition of vacuity that we generalize to the probabilistic setting captures the fact that the truth values of the grants in the specification do not influence the satisfaction of the specification at all. We could replace "it is eventually granted" with any arbitrary requirement or even choose an arbitrary set of positions at which that part of the specification should be true and the specification would still hold in the system model. We say that this subformula does not *affect* the satisfaction of the specification. Perturbing the truth values

in arbitrary ways is expressed by a universal quantification over a proposition in the formal definition. A vacuity check during the model checking process can be an invaluable help as it can detect such severe errors in the design of the model or the specification in an early stage of the development that would otherwise stay undetected if the model checker returns the desired result.

We provide a generalization of the definition of affection that is suitable for the probabilistic setting. We prove that $\Pi_1^{\mathrm{QLTL}}$-model checking is inter-reducible with the question whether a subformula affects a formula in a probabilistic system. Hence, an additional vacuity check according to this definition does not increase the complexity of model checking in MDPs. For Markov chains, however, the additional vacuity check would lead to an exponential blow-up of the procedure as shown by our new lower bound for $\Pi_1^{\mathrm{QLTL}}$-model checking over Markov chains. Consequently, we turn our attention to the notion of inherent vacuity introduced in [9]. This notion captures that a specification is vacuous, i.e. not affected by some subformula, in all models. So, while disregarding the interplay between model and specification, inherent vacuity indicates a severe error in the specification. For all natural variants of this definition for Markov chains and MDPs that can be obtained using our notion of affection, we obtain the result that inherent vacuity of a specification can be checked by a (non-probabilistic) validity check of a $\Pi_1^{\mathrm{QLTL}}$-formula. Therefore, inherent vacuity for Markov chains and MDPs can be checked in polynomial space rendering the addition of a check for inherent vacuity to the model checking procedure potentially useful and reasonable in practice.

### Related Work

Closest to our main complexity hierarchy result is the complexity hierarchy result for QLTL in the non-probabilistic setting [22]. Over probabilistic systems, the model-checking problems for Wolper's ETL [26], another $\omega$-regular extension of LTL, which uses automata operators, is investigated in [7] and shown to lie in EXPTIME. We are not aware of any explicit investigations of QLTL or further $\omega$-regular extensions of LTL, such as Gabbay's USF [10], an extension with fixed-point operators, over probabilistic systems.

Concerning vacuity checking, various notions have been studied for non-probabilistic systems. In [3] and [16], a notion of *formula vacuity* for fragments of CTL$^*$ is investigated in which the underlying notion of non-affection means that a subformula can be replaced by any other formula without affecting the truth of the formula in a model. *Trace vacuity* for LTL, which we generalize to the probabilistic setting, was introduced in [1]. The authors argue that trace vacuity has advantages over formula vacuity as it is more robust with respect to changes of the model or the specification language. Based on this definition, the notion of inherent vacuity, which we adapt to the probabilistic setting, was introduced in [9]. Trace vacuity has been extended to various other logics such as CTL$^*$ [11] relying on a propositionally quantified version of the logic, or to the logic RELTL, an extension of LTL with regular layers, by universally quantifying interval variables [4]. In [12], a variety of degrees to which a formula can be vacuous is defined and analyzed in the context of CTL-model checking. For a survey covering different approaches of vacuity checking, we refer the reader to [15].

## 2 Preliminaries

We suppose familiarity with basic concepts of discrete Markovian models, LTL, and $\omega$-automata, and only provide a brief summary of the notions and our notation. Details can be found in textbooks, e.g., [2, 6, 20]. Furthermore, we provide definitions regarding QLTL and state basic results.

## 2.1 Basic definitions

### Markov decision processes (MDPs)

An MDP is a tuple $\mathcal{M} = (S, Act, P, s_{init}, AP, L)$ where $S$ is a finite state space, $Act$ a finite set of actions, $P : S \times Act \times S \to [0,1] \cap \mathbb{Q}$ the transition probability function satisfying $\sum_{t \in S} P(s,\alpha,t) \in \{0,1\}$ for all $(s,\alpha) \in S \times Act$, $s_{init} \in S$ the initial state, $AP$ a finite set of atomic propositions, and $L : S \to 2^{AP}$ a labeling function. The triples $(s,\alpha,t)$ with $P(s,\alpha,t) > 0$ are called transitions of $\mathcal{M}$. The actions enabled in $s$ form $Act(s) = \{\alpha \in Act : \sum_{t \in S} P(s,\alpha,t) = 1\}$. The *size* of an MDP is the number of states and actions plus the sum of the logarithmic lengths of the transition probabilities. Intuitively, when $\mathcal{M}$ is at a state $s$, then an action $\alpha$ of $Act(s)$ is selected nondeterministically; afterwards the next state is obtained by probabilistically choosing one of the potential successor states according to the probability distribution $P(s,\alpha,\cdot)$. Paths in MDP are alternating sequences of states and actions: $\pi = s_0\,\alpha_0\,s_1\,\alpha_1\ldots$ where $\alpha_i \in Act(s_i)$ and $P(s_i,\alpha_i,s_{i+1}) > 0$ for all $i \geqslant 0$. We write $\pi_{[i\ldots]}$ for the suffix starting from $s_i$. The *trace* of $\pi$ is the word $L(\pi) = L(s_0)\,L(s_1)\,L(s_2)\ldots$ over $2^{AP}$ obtained by projecting states to their labels. We do not distinguish between a path and its trace when the intended meaning is clear from context. A *scheduler* for $\mathcal{M}$ is a function $\mathfrak{S}$ that maps a finite path $\zeta$ to a probability distribution over $Act(last(\zeta))$ where $last(\zeta)$ is the last state of $\zeta$. The function $\Pr^{\mathfrak{S}}_{\mathcal{M},s}$ denotes the probability measure induced by $\mathfrak{S}$, when $s$ is the initial state. It is well-known that all $\omega$-regular path properties $\varphi$ are measurable and there exist schedulers maximizing or minimizing the probability for $\varphi$ (see, e.g., [2]). This justifies the notations $\Pr^{\max}_{\mathcal{M},s}(\varphi) = \max_{\mathfrak{S}} \Pr^{\mathfrak{S}}_{\mathcal{M},s}(\varphi)$ and analogously $\Pr^{\min}_{\mathcal{M},s}(\varphi)$ for $\omega$-regular properties.

A *Markov chain* is a tuple $\mathcal{M} = (S, P, s_{init}, AP, L)$ which can be seen as an MDP with only one action. The transition probability function $P : S \times S \to [0,1] \cap \mathbb{Q}$ does not include the action anymore and satisfies $\sum_{t \in S} P(s,t) \in \{0,1\}$ for all $s \in S$. There are no non-deterministic choices and $\Pr_{\mathcal{M},s}$ denotes the induced probability measure on maximal paths starting in $s$.

### $\omega$-automata

A *non-deterministic Büchi automaton (NBA)* is a tuple $\mathcal{A} = (Q, \Sigma, \delta, Q_0, F)$ where $Q$ is a finite set of states, $\Sigma$ an alphabet, $\delta \subseteq S \times \Sigma \times S$ the transition relation, $Q_0 \subseteq Q$ the set of initial states and $F \subseteq Q$ the set of final states. A word $w = w_0\,w_1\,\ldots$ in $\Sigma^\omega$ is accepted by $\mathcal{A}$ if there is a run $q_0\,w_0\,q_1\,w_1\,q_2\,\ldots$ such that $q_0 \in Q_0$, $(q_i, w_i, q_{i+1}) \in \delta$ for all $i$, and for infinitely many $i$, $q_i \in F$. The language $\mathcal{L}(\mathcal{A})$ is the set of words accepted by $\mathcal{A}$.

## 2.2 Quantified linear temporal logic (QLTL)

Let $AP$ be a finite set of atomic propositions. The syntax of linear temporal logic (LTL) is given by

$$\varphi ::= a \mid \varphi \wedge \varphi \mid \neg\varphi \mid \bigcirc \varphi \mid \varphi\,U\,\varphi$$

where $a \in AP$. The semantics is given on words in $(2^{AP})^\omega$: For a word $w = w_0, w_1, \ldots$, we have $w \vDash a$ if $a \in w_0$; $w \vDash \bigcirc\varphi$ if $w_1, w_2, \cdots \vDash \varphi$; and $w \vDash \varphi\,U\,\psi$ if there is a $j \in \mathbb{N}$ such that $w_j, w_{j+1}, \cdots \vDash \psi$ and $w_i, w_{i+1}, \cdots \vDash \varphi$ for all $i < j$. The semantics of the Boolean connectives is defined as usual. For more details, consult, e.g., [2]. The logic QLTL is an extension of LTL with quantification over atomic propositions. We extend the syntax of LTL by allowing existential quantification $\exists x.\varphi$ over additional, fresh atomic propositions $x \notin AP$ where $\varphi$ is an LTL-formula over $AP \cup \{x\}$. We further allow the common abbreviations $\top$ for

true, $\bot$ for false, $\vee$, $\rightarrow$, $\leftrightarrow$, $\Diamond$, $\Box$, and $\forall x$. For a word $w \in (2^{AP})^\omega$, we define that $w \vDash \exists x.\varphi$ if and only if there is a set $X \subseteq \mathbb{N}$ such that the word $w'$ with $w'[i] = w[i]$ if $i \notin X$ and $w'[i] = w[i] \cup \{x\}$ if $i \in X$ satisfies $w' \vDash \varphi$. Consider the following example to illustrate the semantics of QLTL:

$$\begin{array}{lllll} \{a\} & \{b\} & \{a\} & \{c\} & \dots \quad \vDash \exists x.\Box(x \leftrightarrow \neg a) \text{ because} \\ \{a\} & \{b,x\} & \{a\} & \{c,x\} & \dots \quad \vDash \Box(x \leftrightarrow \neg a). \end{array}$$

For a QLTL-formula $\vartheta$ over $AP$, we allow arbitrarily many additional atomic propositions but require that all atomic propositions not in $AP$ are quantified. We distinguish QLTL-formulas in prenex normal form according to the number of quantifier alternations. For $k \geqslant 1$, let $\Sigma_k^{QLTL}$ be the set of QLTL-formulas of the form

$$\underbrace{\exists^*\forall^*\exists^* \dots}_{k \text{ blocks of quantifiers}} \quad \varphi \equiv \quad \underbrace{\exists^*\neg\exists^*\neg\exists^* \dots}_{k \text{ blocks of quantifiers}} \quad (\neg)\varphi$$

where $\varphi$ is quantifier-free, i.e. an LTL-formula. Likewise, let $\Pi_k^{QLTL}$ be the set of QLTL-formulas starting with $k$ blocks of quantifiers followed by a quantifier-free formula such that the first block is $\forall^*$. The negation of a $\Sigma_k^{QLTL}$-formula is equivalent to a $\Pi_k^{QLTL}$-formula.

QLTL, and in particular $\Sigma_1^{QLTL}$, can express exactly all $\omega$-regular properties. In fact, the existence of an accepting run on a word in an NBA $\mathcal{A}$ with states $Q$ can be expressed by a $\Sigma_1^{QLTL}$-formula with $|Q|$-many existential quantifications followed by an LTL-formula [21].

Conversely, for a $\Sigma_k^{QLTL}$-formula $\vartheta = \exists^*\neg\exists^* \dots (\neg)\varphi$, we can build an NBA of $k$-exponential size accepting exactly the words satisfying $\vartheta$: For the LTL-part $(\neg)\varphi$, we first construct an NBA of exponential size (see [25]). Existential quantification on the NBA-level is easy as it corresponds to standard projection onto the non-quantified variables; a quantified variable $x$ is simply removed from the labels of the transition relation. This introduces new non-deterministic choices between the options to take a transition requiring a letter, i.e. a set of atomic propositions, $P$ or a transition requiring $P \cup \{x\}$ when reading $P$. The quantifier prefix contains $k-1$ negations in addition to the existential quantifiers. Each of these negations requires a complementation of the automaton constructed so far before we can use projection again to account for the next block of quantifiers. Each complementation increases the size by one further exponential. Hence, the procedure produces an NBA for $\vartheta$ of $k$-exponential size in $k$-exponential time (see [22] for more details).

## 3 QLTL model checking in probabilistic systems

This section is devoted to proving the complexity hierarchy results in terms of the quantifier alternation for the model-checking problem of QLTL in probabilistic systems. More precisely, our goal is to pinpoint the complexities of the following problems, for $\Pi_k^{QLTL}$- or $\Sigma_k^{QLTL}$-formulas $\varphi$:

- Qualitative model-checking problems:
  - For a Markov chain $\mathcal{M}$, decide whether $\Pr_{\mathcal{M},s_{init}}(\varphi) = 1$, or whether $\Pr_{\mathcal{M},s_{init}}(\varphi) > 0$, respectively.
  - For an MDP $\mathcal{M}$, decide whether $\Pr^{max}_{\mathcal{M},s_{init}}(\varphi) = 1$, whether $\Pr^{max}_{\mathcal{M},s_{init}}(\varphi) > 0$, whether $\Pr^{min}_{\mathcal{M},s_{init}}(\varphi) = 1$, or whether $\Pr^{min}_{\mathcal{M},s_{init}}(\varphi) > 0$, respectively.
- Quantitative model-checking problems:
  - For a Markov chain $\mathcal{M}$, compute $\Pr_{\mathcal{M},s_{init}}(\varphi)$. For hardness results, we consider the decision versions whether $\Pr_{\mathcal{M},s_{init}}(\varphi) \bowtie \vartheta$ for a given $\vartheta \in \mathbb{Q}$ and $\bowtie \in \{\leqslant, <, >, \geqslant\}$.

- For an MDP $\mathcal{M}$, compute $\Pr^{\text{opt}}_{\mathcal{M},s_{init}}(\varphi)$ for $\text{opt} \in \{\max, \min\}$. For hardness results, we consider the decision versions whether $\Pr^{\text{opt}}_{\mathcal{M},s_{init}}(\varphi) \bowtie \vartheta$ for a given $\vartheta \in \mathbb{Q}$, $\bowtie \in \{\leqslant, <, >, \geqslant\}$, and $\text{opt} \in \{\max, \min\}$.

We restrict our attention to QLTL-formulas in prenex normal form. While we have seen that all QLTL-formulas are equivalent to a $\Sigma_1^{\text{QLTL}}$-formula, the transformation from arbitrary QLTL-formulas to $\Sigma_1^{\text{QLTL}}$-formulas has non-elementary complexity. The lower bound for this transformation is a direct consequence of the complexity hierarchy result for the non-probabilistic model-checking problem mentioned above. However, there is a polynomial-time transformation to prenex normal form for QLTL-formulas: After renaming all quantified variables such that each quantifier quantifies a unique variable not occurring outside the scope of this quantifier, we can pull out quantifiers using the following equivalences for arbitrary QLTL-formula $\varphi$ and $\psi$ where $\psi$ does not contain the atomic proposition $x$ and both formulas do not contain $t$:

1. $(\exists x \varphi)\, U\, \psi \equiv \forall t \exists x((t\, U(\neg t \wedge \varphi)) \vee (t\, U\, \psi))$.
2. $(\forall x \varphi)\, U\, \psi \equiv \forall x(\varphi\, U\, \psi)$.
3. $\psi\, U(\exists x \varphi) \equiv \exists x(\psi\, U\, \varphi)$.
4. $\psi\, U(\forall x \varphi) \equiv \exists t \forall x((\psi \wedge t)\, U(\varphi \wedge \neg t))$.

Note that in the first and last equivalence where $t$ is quantified, only the first position where $\neg t$ holds is important for the subsequent formulas. In this way, the quantification over $t$ corresponds to the quantification over positions in the semantics of the U-operator. For $Q \in \{\exists, \forall\}$, we further have $\bigcirc Q x \varphi \equiv Q x \bigcirc \varphi$ and moving quantifiers to the front over Boolean connectives can be done as usual. So, we can transform a QLTL-formula to prenex normal form in polynomial time while introducing new quantifiers to account for the implicit quantification over positions of the U-operator.

In applications of QLTL in formal verification, however, quantified variables are mostly used to describe possible annotations of a trace or traces of hidden variables. Hence, the quantified traces are supposed to be constant once chosen and not to be reassigned when evaluating subformulas on different suffixes. Thus, these formulas often are already in prenex normal form.

Our main result concerning QLTL-model checking over probabilistic systems is the following complexity hierarchy result:

▶ **Theorem 1** (Main Result). *All qualitative and quantitative model-checking problems for $\Sigma_k^{\text{QLTL}}$ and $\Pi_k^{\text{QLTL}}$ in Markov chains are $k$-EXPSPACE-complete and can be solved in time polynomial in the size of the Markov chain.*

*All qualitative and quantitative model-checking problems for $\Sigma_k^{\text{QLTL}}$ and $\Pi_k^{\text{QLTL}}$ with $k \geqslant 1$ in MDPs are $k+1$-EXPTIME-complete and can be solved in time polynomial in the size of the MDP.*

The upper bounds are obtained by the straight-forward construction of NBAs as described above (Section 2.2). The main contribution hence is the proof of the lower bounds. For Markov chains, we provide a reduction from a tiling problem that simultaneously shows hardness for all qualitative model-checking problems (Theorem 2). We afterwards conclude the same complexity result for all quantitative model-checking problems (Corollary 3). For MDPs, the result requires two different hardness proofs (Theorem 4): The hardness results for model-checking problems regarding the maximal satisfaction probability of $\Pi_k^{\text{QLTL}}$-formulas (or analogously the minimal satisfaction probability of $\Sigma_k^{\text{QLTL}}$-formulas) are somewhat simpler. We encode computations of an alternating Turing machine that is $k$-exponentially space

bounded and can directly use sequences of k-exponentially many extended tape symbols for the encoding. For the hardness proof concerning the minimal satisfaction probability of $\Pi_k^{\mathrm{QLTL}}$-formulas, we have to include a binary counter of $k-1$-exponential length separating two successive tape symbols in the encoding. In the hardness proof for Markov chains, we use a similar counter. So, the final hardness proof combines the ideas behind the first hardness proof for MDPs and the hardness proof for Markov chains. The same complexity results for all quantitative model-checking problems in MDPs can be concluded afterwards (Corollary 5).

## 3.1 Markov chains

We first address the qualitative model-checking problems in Markov chains. We provide a proof sketch for the hardness proof. The full proof can be found in [19].

▶ **Theorem 2.** *For any* $k$*, all qualitative model-checking problems for* $\Sigma_k^{\mathrm{QLTL}}$ *and* $\Pi_k^{\mathrm{QLTL}}$ *in Markov chains are* $k$*-EXPSPACE-complete and can be solved in time polynomial in the size of the Markov chain.*

**Proof sketch.** The upper bounds are obtained by building NBAs of k-exponential size for $\Sigma_k^{\mathrm{QLTL}}$-formulas as described in Section 2. The negation of a $\Pi_k^{\mathrm{QLTL}}$-formula is equivalent to a $\Sigma_k^{\mathrm{QLTL}}$-formula of the same length. As all qualitative model-checking problems for NBAs in Markov chains are PSPACE-complete and can be solved in time polynomial in the size of the Markov chain [7], we obtain the upper bounds.

For the hardness results, we use a reduction from k-exponential tiling problems. We define the following function $h\colon \mathbb{N}^2 \to \mathbb{N}$: Let $h(0,n) = n$ for all $n$ and $h(k+1,n) = 2^{h(k,n)} \cdot h(k,n)$ for all $k$. So, $h(k,n)$ is k-exponential in $n$. The following k-exponential tiling problem is known to be k-EXPSPACE-complete [24]:

*Given:* a finite set of tiles $T$, two relations $H \subseteq T^2$ and $V \subseteq T^2$, an initial tile $t_0 \in T$ and a final tile $t_f \in T$ as well as a natural number $n$ in unary.

*Question:* Is there an $m \in \mathbb{N}$ such that the $2^{h(k-1,n)} \times (m+1)$-grid $\{0,\dots,2^{h(k-1,n)}-1\} \times \{0,\dots,m\}$ can be tiled, i. e., is there a function $f\colon \{0,\dots,2^{h(k-1,n)}-1\} \times \{0,\dots,m\} \to T$, such that:

1. the tile at position $(0,0)$ is the initial tile $t_0$ and the tile at position $(0,m)$ is the final tile $t_f$; in other words, $f(0,0) = t_0$ and $f(0,m) = t_f$,
2. two tiles placed next to each other horizontally satisfy the relation $H$; more precisely, for any $0 \leqslant i < 2^{h(k-1,n)} - 1$ and $0 \leqslant j \leqslant m$, the pair $(f(i,j), f(i+1,j)) \in H$, and
3. two tiles placed next to each other vertically satisfy the relation $V$; more precisely, for any $0 \leqslant i \leqslant 2^{h(k-1,n)} - 1$ and $0 \leqslant j < m$, the pair $(f(i,j), f(i,j+1)) \in V$?

Given an instance of the k-exponential tiling problem, we construct a Markov chain $\mathcal{M}$ and a $\psi$ in $\Pi_k^{\mathrm{QLTL}}$ such that $\Pr_{\mathcal{M}}(\psi) = 1$ iff $\Pr_{\mathcal{M}}(\psi) > 0$ iff there is a valid tiling. This establishes k-EXPSPACE-hardness for both qualitative model checking problems for $\Pi_k^{\mathrm{QLTL}}$. As the negation of $\psi$ is in $\Sigma_k^{\mathrm{QLTL}}$ and k-EXPSPACE is closed under complementation, the same result holds for $\Sigma_k^{\mathrm{QLTL}}$.

Let $T = \{t_0,\dots,t_\ell\}$ be the set of tiles that we also use as atomic propositions and let $\{\mathit{start}, \mathit{end}, 0, 1\}$ be further atomic propositions. We construct a simple Markov chain $\mathcal{M}$, depicted in Figure 1, that almost surely produces a concatenation of infinitely many words from $\mathit{start}(T \cup \{0,1\})^+\mathit{end}$ that contains each of the finite words in $\mathit{start}(T \cup \{0,1\})^+\mathit{end}$. Some of these finite words will encode potential tilings. Namely, we encode a function $f\colon \{0,\dots,2^{h(k-1,n)}-1\} \times \{0,\dots,m\} \to T$ in the word

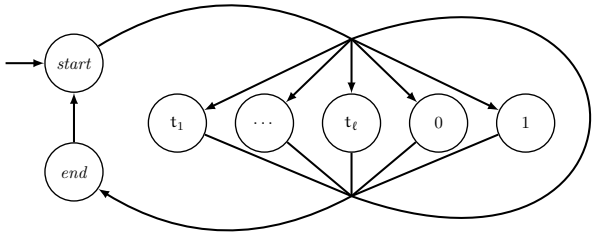


**Figure 1** The Markov chain $\mathcal{M}$.

$$\begin{aligned} &\mathit{start},\ f(0,0),\ \overbrace{0,0,0,\ldots,0}^{h(k-1,n)\text{ steps}},f(1,0),\ \overbrace{1,0,0,\ldots,0}^{h(k-1,n)\text{ steps}},\ldots,f(2^{h(k-1,n)}-1,0),\ \overbrace{1,1,1,\ldots,1}^{h(k-1,n)\text{ steps}},\\ &\quad f(0,1),\ \ldots,\\ &f(0,m),\overbrace{0,0,0,\ldots,0}^{h(k-1,n)\text{ steps}},f(1,m),\overbrace{1,0,0,\ldots,0}^{h(k-1,n)\text{ steps}},\ldots,f(2^{h(k-1,n)}-1,m),\ \overbrace{1,1,1,\ldots,1}^{h(k-1,n)\text{ steps}},\mathit{end}. \end{aligned}$$

For a valid encoding, the blocks of $h(k-1,n)$ bits have to encode a correct binary counter modulo $2^{h(k-1,n)}$, where the first bit is the least significant one, starting with $0\ldots0$ after *start* and ending in $1\ldots1$ before *end*. The encoding of the counter makes sure that indeed a function from a rectangle $\{0,\ldots,2^{h(k-1,n)}-1\}\times\{0,\ldots,m\}$ for some $m$ is encoded.

Further, we construct a $\Pi_k^{QLTL}$-formula *valid_tiling* that expresses that at some point a valid tiling is encoded in a run. Several of the conditions including the initial, final and horizontal condition can easily be expressed. As tiles that are vertically adjacent in a tiling are separated by $h(k,n)=h(k-1,n)\cdot 2^{h(k-1,n)}$ steps, however, we have to employ additional ideas to express that all conditions on a valid encoding of a valid tiling are satisfied at some point. An important ingredient for our reduction is the collection of $\Sigma_{k-1}^{QLTL}$-formulas $\varphi_{k-1,n}(p,q)$ from [22]. For each $n$ and $k$ from $\mathbb{N}$, the formula $\varphi_{k-1,n}(p,q)$ holds on a word if $p$ and $q$ occur exactly once and, if the position at which $p$ occurs is $i$, the position at which $q$ occurs is $i+h(k-1,n)$. In addition to the use of these formulas, we use universally quantified propositions that mark potential violations of the conditions. To illustrate this idea, we sketch a formula that expresses that a run of $\mathcal{M}$ eventually contains a finite word starting with *start* and ending in *end* in which tiles are followed by exactly $h(k-1,n)$-many bits. The atomic proposition *tile* holds if the current letter encodes a tile.

$$\begin{aligned} \forall d.\Big(\big[\forall p\forall q\big(\varphi_{k-1,n}(p,q)\rightarrow\Box[(d\wedge \mathit{tile}\wedge p)\rightarrow\\ \text{next occurrence of } \mathit{tile} \text{ or } \mathit{end} \text{ not one step after } q]\big)\big]\rightarrow\Diamond(\mathit{start}\wedge(\neg(d\,U\,\mathit{end}))\Big). \end{aligned}$$

The quantified proposition $d$ can be used to mark any tiles for which the next tile or *end* does not follow exactly $h(k-1,n)+1$ steps later. The quantified variables $p$ and $q$ and the formula $\varphi_{k-1,n}(p,q)$ are used to check that the markers are placed correctly, i.e., that indeed the next occurrence of *tile* or *end* after the marked position is not exactly $h(k-1,n)+1$ steps later. If the markers $d$ are not placed correctly, the formula holds. Otherwise, it holds if a finite word between *start* and *end* is contained in the run in which no tile is marked

by $\mathtt{d}$. As $\mathtt{d}$ is universally quantified, the formula hence holds on a run of $\mathcal{M}$ iff it contains a finite word starting with *start* and ending in *end* in which tiles are followed by exactly $h(k-1,n)$-many bits. Note that $\varphi_{k-1,n}(p,q)$ occurs in the scope of two negations due to the implications while $\forall p \forall q$ occurs in the scope of one negation. So, the formula is in $\Pi_k^{\mathrm{QLTL}}$.

The correctness of the counter can be expressed using the same idea of marking bits that violate the correctness of the counter with a universally quantified variable and the fact that a bit in a binary counter changes during an increment of the counter if and only if all less significant bits are 1. The vertical condition of the tiling is checked by using universally quantified markers $\nu_1$ and $\nu_2$ that have to be placed on vertically adjacent tiles. The correct placement of the markers is checked by stating that there exists a proposition $\mathtt{b}$ that encodes a correct binary counter with $h(k-1,n)$-many bits that starts with $0\dots0$ after $\nu_1$ and counts up to $1\dots1$ right before $\nu_2$. The correctness of the counter is checked as for the counter using the bits 0 and 1. The additional existential quantification over $\mathtt{b}$ does not yield an additional quantifier alternation. The resulting formula *valid_tiling* is in $\Pi_k^{\mathrm{QLTL}}$ and holds on a run of $\mathcal{M}$ if an encoding of a valid tiling is produced. As a run of $\mathcal{M}$ almost surely contains all words in $\mathit{start}(T \cup \{0,1\})^+\mathit{end}$, the formula *valid_tiling* holds with probability 1 iff it holds with positive probability iff there is a valid tiling for the given instance of the k-exponential tiling problem. ◀

As the upper bounds are obtained via the construction of NBAs for the QLTL-formulas, we can conclude the same results for the quantitative model-checking problems over Markov chains.

▶ **Corollary 3** (Quantitative model checking). *Given a $\Sigma_k^{\mathrm{QLTL}}$- or $\Pi_k^{\mathrm{QLTL}}$-formula $\varphi$ and a Markov chain $\mathcal{M}$, the probability $\Pr_{\mathcal{M}}(\varphi)$ can be computed in k-exponential space and in time polynomial in the size of $\mathcal{M}$. Given a rational $\vartheta \in [0,1]$ and $\bowtie \in \{\leqslant, <, >, \geqslant\}$, deciding whether $\Pr_{\mathcal{M}}(\varphi) \bowtie \vartheta$ is k-EXPSPACE-complete.*

**Proof.** The lower bounds follow directly from the previous theorem. The upper bound follows from the fact that, given a Markov chain $\mathcal{M}$ and an NBA $\mathcal{A}$, the probability $\Pr_{\mathcal{M}}(\mathcal{A})$ that a word produced by $\mathcal{M}$ is accepted by $\mathcal{A}$ can be computed in time polynomial in $\mathcal{M}$ [7] and in space polynomial in the total size of the input. We sketch a proof of the latter claim: In the algorithm provided by Courcoubetis and Yannakakis in [7] to compute this probability, an exponentially large Markov chain $\mathcal{N}$ is constructed from $\mathcal{M}$ and $\mathcal{A}$. The states of $\mathcal{N}$ have a polynomial representation in the size of $\mathcal{M}$ and $\mathcal{A}$ and one can compute the transition probabilities between any two states in polynomial time. The probability $\Pr_{\mathcal{M}}(\mathcal{A})$ now equals the probability to reach a *recurrent* state in $\mathcal{N}$ – as it is called in [7], but which we do not define here. It is only important to us that one can decide whether a state is recurrent in polynomial space polynomial in the size of $\mathcal{A}$ (and polylogarithmic in the size of $\mathcal{M}$) as shown in [7]. The probability to reach a recurrent state in $\mathcal{N}$ can be computed by solving a linear equation system. As transition probabilities and whether states are recurrent in $\mathcal{N}$ can be computed in space polynomial in $\mathcal{A}$, each entry of the matrix and vector representing this linear equation system, which is of size exponential in $\mathcal{A}$ and polynomial in $\mathcal{M}$, can be computed in space polynomial in $\mathcal{A}$. Using the fact that solving linear equation systems lies in the complexity class NC and can hence be done in polylogarithmic space (see, e.g., [18, Section 15]) and standard results on the composition of space-bounded transductions (see, e.g., [18, Section 8]), we can conclude that the probability $\Pr_{\mathcal{M}}(\mathcal{A})$ can be computed in space polynomial in the size of $\mathcal{A}$. Applied to the k-exponentially sized NBAs for $\Sigma_k^{\mathrm{QLTL}}$-formulas, this result leads to the claim of the corollary. ◀

## 3.2 Markov decision processes

We now provide the complexity results for QLTL-model checking over MDPs.

▶ **Theorem 4.** *Given an MDP* $\mathcal{M}$*, a* $\Pi_k^{\mathrm{QLTL}}$*-formula* $\varphi$*, and* $\mathrm{opt} \in \{\max, \min\}$*, deciding whether* $\Pr_{\mathcal{M}}^{\mathrm{opt}}(\varphi) = 1$ *and deciding whether* $\Pr_{\mathcal{M}}^{\mathrm{opt}}(\varphi) > 0$ *are* $k+1$*-EXPTIME-complete for any* $k \geqslant 1$*. The problems are solvable in time polynomial in the size of* $\mathcal{M}$*.*

As $\Pi_k^{\mathrm{QLTL}}$ is not closed under negation, the model-checking problems in MDPs concerning the maximal and minimal satisfaction probability, respectively, require different hardness proofs. We sketch the two proof ideas in the sequel. The full proofs can be found in [19].

**Proof sketch.** The upper bounds are obtained via the straight-forward construction of *deterministic* automata (e.g., deterministic Rabin automata; see, e.g., [2]). This requires the determinization of the k-exponentially large NBAs for $\Sigma_k^{\mathrm{QLTL}}$-formulas, which are computable in k-exponential time, and leads to a $k+1$-exponential-time procedure.

For the lower bounds, first consider the problems with $\mathrm{opt} = \max$. We prove $k+1$-EXPTIME-hardness by encoding the computation of k-exponentially space-bounded alternating Turing machines (ATM). It is well-known that the class of problems decidable by such ATMs coincides with $k+1$-EXPTIME [5]. So, given a k-exponentially space-bounded ATM $\mathcal{T}$ and an input word $w$, we construct an MDP $\mathcal{M}$ and a $\Pi_k^{\mathrm{QLTL}}$-formula $\varphi$ such that $\Pr_{\mathcal{M}}^{\max}(\varphi) > 0$ iff $\Pr_{\mathcal{M}}^{\max}(\varphi) = 1$ iff $w$ is accepted by $\mathcal{T}$. Recall that acceptance in an ATM can be specified in terms of a game between a universal player choosing the next move in universal states and an existential player choosing the next move in existential state. A word is accepted if the existential player has a strategy that ensures that an accepting state is reached from the initial configuration with the input word on the tape.

The idea for the reduction is to construct an MDP $\mathcal{M}$ in which the scheduler can produce a sequence of (k-exponentially long) configurations of $\mathcal{T}$. The sequence of configurations in turn represents a sequence of infinitely many finite computations. The first configuration of each computation has to be the initial configuration with $w$ on the tape. After each configuration, the scheduler has to specify whether the universal or existential player has to choose the next move, or whether the computation ended and a new computation is about to start. If it is the existential player's turn, the scheduler chooses a move and has to construct the successor configuration accordingly. If it is the universal player's turn, the successor move is specified by a random choice and again the scheduler has to construct the correct successor configuration. The constructed MDP is sketched in Figure 2.

The $\Pi_k^{\mathrm{QLTL}}$-formula $\varphi$ we construct, on the one hand, expresses that the sequence produced by the scheduler obeys all these requirements. Checking that the successor configurations are constructed correctly is possible with the use of the $\Sigma_k^{\mathrm{QLTL}}$-formulas $\varphi_{k,n}(p, q)$ from [22] that express that the positions at which $p$ and $q$ are a fixed k-exponentially large number of steps apart. On the other hand, the formula $\varphi$ expresses that all (infinitely many) encoded computations end in an accepting state. If $w$ is accepted by $\mathcal{T}$, the scheduler can construct correct accepting computations no matter what moves are chosen by the universal player and so $\Pr_{\mathcal{M}}^{\max}(\varphi) = 1$. If $w$ is not accepted, however, the universal player will play according to a winning strategy in any of the encoded computations with positive probability. So, almost surely at some point any scheduler has to violate one of the requirements or construct a rejecting computation. In this case, $\Pr_{\mathcal{M}}^{\max}(\varphi) = 0$.

In contrast to the case just discussed, the statement $\Pr_{\mathcal{M}}^{\min}(\varphi) = 1$ is a statement about all schedulers. So, we cannot let a scheduler construct sequences of computations anymore. Instead, we construct an MDP $\mathcal{M}'$ that randomly generates sequences that potentially encode

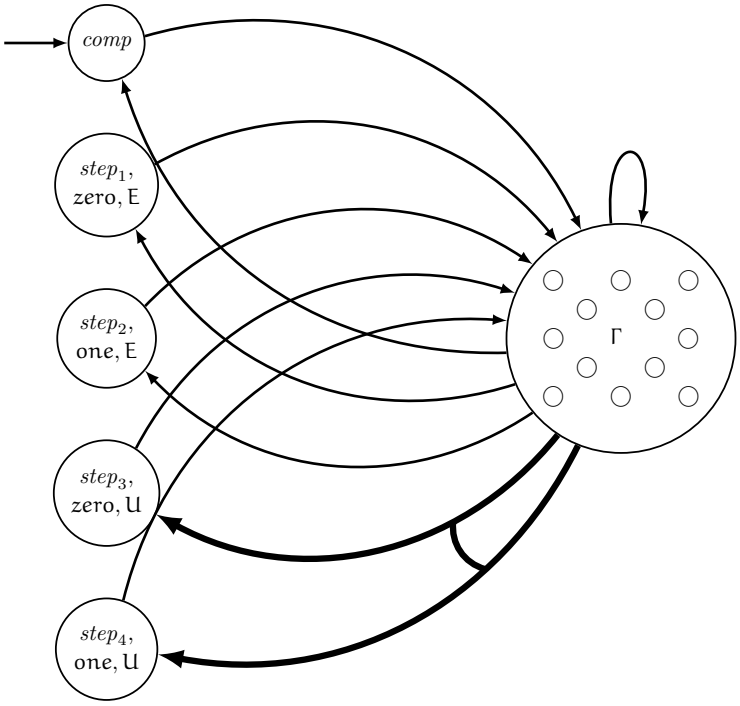


**Figure 2** The MDP $\mathcal{M}$. The state depicted as $\Gamma$ represents the behavior of each state $\gamma \in \Gamma$. I.e. from each state, there is one action to each state in $\Gamma$ with probability 1. Further, from all states in $\Gamma$, there are actions leading to states *comp*, $step_1$ and $step_2$ with probability 1, as well as a randomized action (bold lines) leading to states $step_3$ and $step_4$ with probability 1/2 each. The labels zero and one indicate which successor move was chosen according to which the new configuration has to be constructed. The labels E and U indicate which player has chosen the move and are used to check whether the successor move was indeed chosen randomly iff it is the universal player's move.

correct computations. In the acceptance game of the given ATM, we also switch roles and let the choices of the existential player be made randomly while the scheduler can specify which successor move should be chosen in universal states. With positive probability, the correct successor configuration will be generated afterwards. Hence, if the existential player has a winning strategy, a correct accepting computation will eventually be produced randomly with probability 1 no matter what successor moves a scheduler chooses. Otherwise, there is a scheduler that prohibits this.

In order to express that eventually a correct accepting computation is generated in $\Pi_k^{\mathrm{QLTL}}$, however, it turns out that we cannot use the $\Sigma_k^{\mathrm{QLTL}}$-formulas $\varphi_{k,n}(p, q)$ from [22] as before. This is in part due to the implicit existential quantification in the eventually-modality. For this reason, we do not encode the computations simply as concatenations of configurations. Instead, we employ the ideas that were also used in the hardness proof for Markov chains (Theorem 2): We separate the symbols of the configurations by $k-1$-exponentially long binary counters to check that configurations have the correct length and use universally quantified variables to mark violations to any of the requirements of a valid encoding of an accepting computation. The blocks of the potential binary counter values are also randomly generated as sketched in Figure 3. An existentially quantified proposition encoding a further binary counter with $k-1$-exponentially many bits is then used to compare tape cells at the same position in two successive configurations, which are $k$-exponentially many steps apart

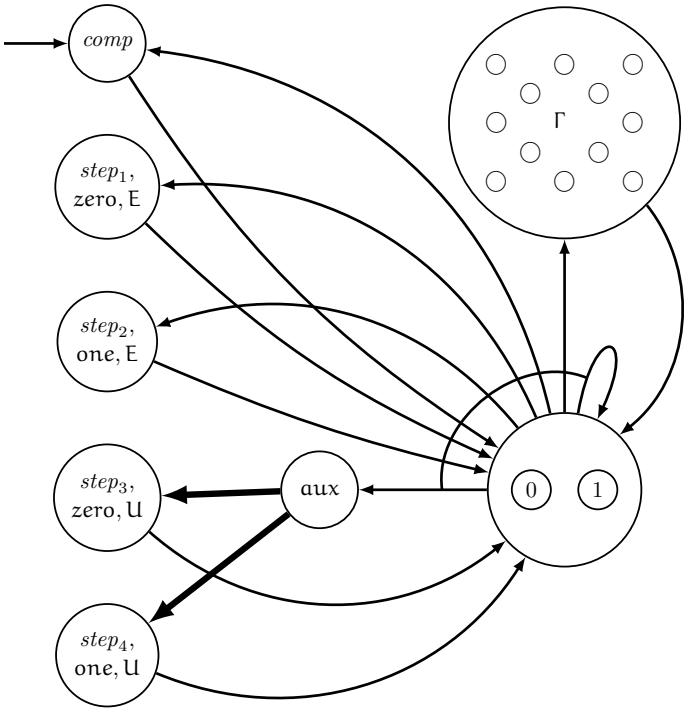


**Figure 3** The MDP $\mathcal{M}'$. The behavior is probabilistic except for the choice in the state aux. When entering the cluster of states Γ or the cluster with the two bits 0 and 1, one of the states in the cluster is chosen randomly. Further all states in Γ have only the outgoing transition randomly moving to 0 or 1. The state aux is only an auxiliary state for the graphical representation. That means that in states 0 and 1 two actions are enabled. The first moving randomly to any state except for $step_3$; the second moving randomly to any state except for $step_4$.

in the encoding. Under any scheduler, the resulting $\Pi_k^{\mathrm{QLTL}}$-formula $\varphi$ holds on an execution of $\mathcal{M}$ almost surely if $w$ is accepted by $\mathcal{T}$. Similar to before, each of the randomly generated potential computations is correct with positive probability and in each of these computations the randomly chosen moves of the existential player are in accordance with a winning strategy with positive probability against any scheduler, which chooses the moves of the universal player. If $w$ is not accepted by $\mathcal{T}$, however, there is a strategy for the universal player and hence a scheduler that makes sure that no correct accepting computation is generated. In this case, $\Pr^{\min}_{\mathcal{M}}(\varphi) = 0$. ◀

These results allow us to conclude that all qualitative model-checking problems for $\Sigma_k^{\mathrm{QLTL}}$-formulas in MDPs are $k+1$-EXPTIME-complete for any $k \geqslant 1$, too, as the negation of a $\Sigma_k^{\mathrm{QLTL}}$-formula is a $\Pi_k^{\mathrm{QLTL}}$-formula. Furthermore, as the upper bounds are obtained via the naive construction of deterministic automata, also the quantitative model checking problems have the same complexity as the minimal and maximal probabilities that an execution of an MDP is accepted by a suitable deterministic automaton (such as a deterministic Rabin automaton) can be computed in polynomial time (for details see, e.g., [2]).

▶ **Corollary 5** (Quantitative model checking). *Given a* $\Sigma_k^{\mathrm{QLTL}}$- *or* $\Pi_k^{\mathrm{QLTL}}$*-formula* $\varphi$ *and an MDP* $\mathcal{M}$*, the probabilities* $\Pr^{\min}_{\mathcal{M}}(\varphi)$ *and* $\Pr^{\max}_{\mathcal{M}}(\varphi)$ *can be computed in time* $k+1$*-exponential in the size of* $\varphi$ *and polynomial in the size of* $\mathcal{M}$*. Given a rational* $\vartheta \in [0,1]$, $\bowtie \in \{\leqslant, <, >, \geqslant\}$ *and* $\mathrm{opt} \in \{\min, \max\}$*, deciding whether* $\Pr^{\mathrm{opt}}_{\mathcal{M}}(\varphi) \bowtie \vartheta$ *is* $k+1$*-EXPSPACE-complete.*

# 4 Trace Vacuity in Probabilistic Systems

Vacuity notions have been studied for non-probabilistic systems in order to express, roughly, that the truth of a formula is not affected by the truth of one of its subformulae [1, 3, 16]. Among the existing definitions of vacuity in the literature, *trace vacuity* is the strongest.

▶ **Definition 6.** *Let* $\varphi$ *be an LTL-formula and* $\psi$ *a subformula. Let* $\mathcal{T}$ *be a transition system. We say that* $\psi$ does not affect $\varphi$ *in* $\mathcal{T}$ *if for every execution* $\pi$ *in* $\mathcal{T}$*:*

$$\pi \vDash \forall x.\varphi[\psi \leftarrow x] \quad \iff \quad \pi \vDash \exists x.\varphi[\psi \leftarrow x].$$

*We say that* $\varphi$ *holds vacuously in* $\mathcal{T}$ *if there is a subformula that does not affect* $\varphi$ *in* $\mathcal{T}$*.*

The above definition of non-affection generalizes the one from [1] by relaxing the hypothesis that $\varphi$ holds on $\mathcal{T}$. For any execution $\pi$, $\pi \vDash \forall x.\varphi[\psi \leftarrow x] \Rightarrow \pi \vDash \varphi \Rightarrow \pi \vDash \exists x.\varphi[\psi \leftarrow x]$. We thus merely require that the three sets of executions that satisfy, $\forall x.\varphi[\psi \leftarrow x]$, $\varphi$, and $\exists x.\varphi[\psi \leftarrow x]$ respectively, coincide. Also, this generalisation allows us to naturally extend the notions of non-affection and vacuity to probabilistic systems. In the remainder of this section, we introduce trace vacuity for probabilistic systems, and establish tight complexity bounds for checking probabilistic vacuity. As in the non-probabilistic case, vacuity checking reduces to checking a $\Pi_1^{\mathrm{QLTL}}$-formula. Conversely, one can reduce the qualitative model checking of $\Pi_1^{\mathrm{QLTL}}$ to probabilistic vacuity.

## 4.1 Probabilistic trace vacuity

▶ **Definition 7.** *Let* $\varphi$ *be an LTL-formula and* $\psi$ *a subformula. Let* $\mathcal{M}$ *be an MDP or a Markov chain. We say that* $\psi$ does not affect $\varphi$ *in* $\mathcal{M}$ *iff*

$$\Pr^{\min}_{\mathcal{M}}(\forall x.(\varphi[\psi \leftarrow x] \leftrightarrow \varphi)) = 1.$$

*We say that* $\varphi$ *is vacuous in* $\mathcal{M}$ *if there is a subformula that does not affect* $\varphi$ *in* $\mathcal{M}$*.*

Note that it does make sense for Markov chains and MDPs to consider that a formula is vacuous if its satisfaction probability (under any scheduler) is not affected when replacing a subformula, even if the global formula does not hold almost-surely. In MDPs, the definition of non-affection generalizes the non-probabilistic definition. This is made more precise in the following proposition. Paths in a transition system correspond to schedulers not making use of randomization when we view a transition system as an MDP.

▶ **Proposition 8.** *A subformula* $\psi$ *does not affect a formula* $\varphi$ *in an MDP (or a Markov chain)* $\mathcal{M}$ *if and only if for all schedulers* $\mathfrak{S}$*,* $\Pr^{\mathfrak{S}}_{\mathcal{M}}(\forall x.\varphi[\psi \leftarrow x]) = \Pr^{\mathfrak{S}}_{\mathcal{M}}(\varphi) = \Pr^{\mathfrak{S}}_{\mathcal{M}}(\exists x.\varphi[\psi \leftarrow x])$.

**Proof.** Let us rewrite $\forall x.(\varphi[\psi \leftarrow x] \leftrightarrow \varphi)$ as $(\forall x.(\varphi \rightarrow \varphi[\psi \leftarrow x])) \wedge (\forall x.(\varphi[\psi \leftarrow x] \rightarrow \varphi))$. The latter is equivalent to $(\varphi \rightarrow \forall x.\varphi[\psi \leftarrow x]) \wedge (\forall x.\neg\varphi[\psi \leftarrow x] \vee \varphi)$. Rewritten as implications, we obtain $(\varphi \rightarrow \forall x.\varphi[\psi \leftarrow x]) \wedge (\exists x.\varphi[\psi \leftarrow x] \rightarrow \varphi)$. As the two implications $(\varphi \leftarrow \forall x.\varphi[\psi \leftarrow x])$ and $(\exists x.\varphi[\psi \leftarrow x] \leftarrow \varphi)$ are tautologies, the claim follows easily considering that the minimal probability in Definition 7 can be read as a universal quantification over schedulers. ◀

▶ **Example 9.** We provide a short example of non-affection in Markov chains, also to shed light on the difference with the non-probabilistic setting. Consider the Markov chain on Fig. 4, where we assume arbitrary non-zero probabilities on edges, and the following

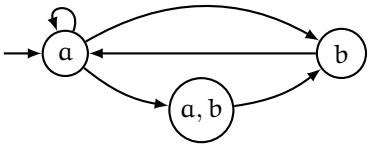


**Figure 4** A Markov chain to illustrate the notion of affection.

formulas: $\varphi = \Box\Diamond(\mathfrak{a} \wedge \mathfrak{b}) \vee \Box(\mathfrak{a} \vee \mathfrak{b})$ and $\psi = \Box(\mathfrak{a} \vee \mathfrak{b})$. Clearly enough, $\Pr_{\mathcal{M}}(\varphi) = \Pr_{\mathcal{M}}(\exists x.\varphi[\psi \leftarrow x]) = \Pr_{\mathcal{M}}(\forall x.\varphi[\psi \leftarrow x]) = 1$ so that $\psi$ does not affect $\varphi$, and $\varphi$ is vacuous in this Markov chain. However, if one views the graph as a transition system $\mathcal{T}$, then $\mathcal{T} \vDash \varphi$ and $\mathcal{T} \nvDash \forall x.\varphi[\psi \leftarrow x]$. So, $\psi$ affects $\varphi$.

Armoni *et al.* [1] observed that if $\psi$ appears only positively in $\varphi$, for every execution $\pi$ in the transition system $\mathcal{T}$ then: $\mathcal{T}, \pi \vDash \forall x.\varphi[\psi \leftarrow x] \iff \mathcal{T}, \pi \vDash \varphi[\psi \leftarrow \bot]$. As a consequence, a pure polarity subformulas $\psi$ does not affect $\varphi$ if and only if $\Pr^{\min}_{\mathcal{M}}(\varphi[\psi \leftarrow \top] \leftrightarrow \varphi[\psi \leftarrow \bot]) = 1$. Therefore, checking whether a pure polarity subformula affects a formula reduces to quantitative model checking of LTL formulas and can be done in PSPACE for Markov chains and in 2-EXPTIME for MDPs.

As also argued in [1], restricting attention to subformulas with pure polarity or to consider single occurrences of subformulas separately is insufficient for a satisfactory vacuity check. For example, a formula like $\Box(p \rightarrow p) \equiv \Box(p \vee \neg p)$, in which $p$ occurs positively and negatively, should be rendered vacuous in any system. Restricting attention to only one of the two occurences of $p$, however, would in general lead to the insight that each of the two occurrences on its own does affect the formula. Beyond pure polarity formulas, checking affectation is harder for Markov chains. Indeed, hardness of $\Pi_1^{\mathrm{QLTL}}$ model checking transfers to hardness of vacuity checking. As stated in the next theorem, for MDPs affection checking has the same complexity as quantitative LTL model checking, whereas Markov chains exhibit an exponential complexity blowup.

▶ **Theorem 10.** *Checking whether a subformula $\psi$ affects an LTL-formula $\varphi$ in a Markov chain $\mathcal{M}$ is EXPSPACE-complete. In MDPs, the problem is 2-EXPTIME-complete.*

**Proof.** The upper bounds follow directly from the upper bounds of qualitative model-checking of $\Pi_1^{\mathrm{QLTL}}$ in Markov chains and MDPs. For the lower bound, we first concentrate on MDPs. We provide a reduction from the problem whether a $\Pi_1^{\mathrm{QLTL}}$-formula $\vartheta = \forall x.\varphi$ satisfies $\Pr^{\min}_{\mathcal{M}}(\vartheta) = 1$ in an MDP $\mathcal{M}$. A proof that the restriction to one quantified variable does not influence the complexity is given in [19]. So, let $\mathcal{M}$ be labeled with atomic propositions from $\mathsf{AP}$. Let $\vartheta = \forall x.\varphi$ where $\varphi$ is an LTL-formula over $\mathsf{AP} \cup \{x\}$ with $x \notin \mathsf{AP}$ be given. We construct the MDP $\mathcal{M}'$ by adding a new initial state $s'_{init}$ from which the original initial state $s_{init}$ is reached in one step with probability 1. Further, we let $\beta$ be an LTL-formula that is valid and does not occur in $\varphi$. Finally, we define $\varphi'$ to be the LTL-formula $\varphi' = \beta \vee \bigcirc\varphi[x \leftarrow \beta]$. Of course, $\Pr^{\min}_{\mathcal{M}',s'_{init}}(\varphi') = 1$ as $\beta$ is valid. We claim that $\beta$ does not affect $\varphi'$ in $\mathcal{M}'$ if and only if $\Pr^{\min}_{\mathcal{M}}(\forall x.\varphi) = 1$. The subformula $\beta$ does not affect $\varphi'$ in $\mathcal{M}'$ iff $\Pr^{\min}_{\mathcal{M}',s'_{init}}(\forall x.(x \vee \bigcirc\varphi)) = 1$ by definition and the fact that $\beta$ does not occur anywhere else in $\varphi$. But $\Pr^{\min}_{\mathcal{M}',s'_{init}}(\forall x.(x \vee \bigcirc\varphi)) = 1$ holds if and only if $\Pr^{\min}_{\mathcal{M}',s'_{init}}(\forall x.\bigcirc\varphi) = 1$ because the universal quantifier can choose $x$ not to hold in the first position of any trace produced by $\mathcal{M}'$. After the first step $\mathcal{M}'$ behaves exactly like $\mathcal{M}$ and hence $\Pr^{\min}_{\mathcal{M}',s'_{init}}(\forall x.\bigcirc\varphi) = 1$ if and only if $\Pr^{\min}_{\mathcal{M},s_{init}}(\forall x.\varphi) = 1$. So, checking affection in MDPs is as hard as the respective qualitative model-checking problem for $\Pi_1^{\mathrm{QLTL}}$ and hence 2-EXPTIME-complete.

For Markov chains, the argument goes analogously. Note that the constructed MDP $\mathcal{M}'$ is a Markov chain if $\mathcal{M}$ is a Markov chain. So, checking affection in Markov chains is also as hard as the respective qualitative model-checking problem for $\Pi_1^{\mathrm{QLTL}}$ and hence EXPSPACE-complete. ◀

In Markov chains, the exponential blow-up in complexity of non-affection checking compared to LTL-model checking constitutes a major obstacle for vacuity checking. To provide a possibility to check that a specification is not obviously faulty without such an exponential blow-up, we turn our attention to the notion of inherent vacuity.

## 4.2 Inherent vacuity in probabilistic systems

Inherent vacuity for transition systems expresses whether a formula holds vacuously in every model in which it holds [9]. Using our generalized definition, we do not restrict ourselves to the models in which the formula holds anymore and provide an analogous definition for probabilistic systems. As in [9], we consider two natural variants of the definition and investigate how to check whether a formula is inherently vacuous.

▶ **Definition 11.** *Let $\varphi$ be an LTL-formula. Let $\mathcal{C}$ be the class of all transition systems, all Markov chains, or all MDPs, respectively. We say that $\varphi$ is inherently vacuous over $\mathcal{C}$, if $\varphi$ is vacuous in all models $\mathcal{M} \in \mathcal{C}$. For a subformula $\psi$ of $\varphi$ we say that $\psi$ inherently does not affect $\varphi$ over $\mathcal{C}$, if for every $\mathcal{M} \in \mathcal{C}$, $\psi$ does not affect $\varphi$ in $\mathcal{M}$. If there is a subformula that inherently does not affect $\varphi$ over $\mathcal{C}$, we say that $\varphi$ is uniformly inherently vacuous.*

In [9], it is shown that inherent vacuity and uniform inherent vacuity coincide for transition systems. Dropping the restriction to models in which a formula $\varphi$ holds, the results of [9] show that the notions are equivalent to the existence of a subformula $\psi$ such that $\forall x.(\varphi[\psi \leftarrow x] \leftrightarrow \varphi)$ is valid. We prove that inherent and uniform inherent vacuity for Markov chains and MDPs are also equivalent to this condition and hence to inherent vacuity in transition systems. First, we show that uniform inherent vacuity coincides with inherent vacuity.

▶ **Proposition 12.** *Let $\varphi$ be an LTL-formula and let $\mathcal{C}$ be the class of all Markov chains or all MDPs. The formula $\varphi$ is uniformly inherently vacuous over $\mathcal{C}$ if and only if it is inherently vacuous over $\mathcal{C}$.*

**Proof.** One direction is clear. For the other direction, suppose that $\varphi$ is inherently vacuous over $\mathcal{C}$, but not uniformly inherently vacuous. Hence, for each subformula $\psi$ of $\varphi$, there is a model $\mathcal{M}_\psi \in \mathcal{C}$ such that $\psi$ affects $\varphi$ over $\mathcal{M}_\psi$. Let $\mathcal{N}$ be the disjoint union of the models $\mathcal{M}_\psi$ for all subformulas $\psi$ with an initial uniform probability distribution over the initial states of these models. We claim that $\varphi$ is not vacuous in $\mathcal{M}$. For each subformula $\psi$, there is a positive probability that $\mathcal{M}_\psi$ is chosen. As there is a scheduler $\mathfrak{S}$ (for Markov chains, the unique scheduler) with $\Pr^{\mathfrak{S}}_{\mathcal{M}_\psi}(\forall x.(\varphi[\psi \leftarrow x] \leftrightarrow \varphi) < 1$, the same holds in $\mathcal{N}$. This is a contradiction to the inherent vacuity of $\varphi$. ◀

The following proposition establishes that all variants of inherent vacuity considered coincide:

▶ **Proposition 13.** *Let $\varphi$ be an LTL-formula and $\psi$ a subformula. Then, $\psi$ inherently does not affect $\varphi$ over Markov chains or MDPs, respectively, if and only if the formula $\forall x.(\varphi \leftrightarrow \varphi[\psi \leftarrow x])$ is valid.*

**Proof.** Only the left-to-right implication deserves a proof, and we prove the contrapositive. Assume the formula $\chi = \forall x.(\varphi \leftrightarrow \varphi[\psi \leftarrow x])$ is not valid. Since $\chi$ expresses a regular property, there exists an ultimately periodic word $w$ that violates $\chi$. It suffices to consider the Markov chain or MDP $\mathcal{M}$ that has only one path, and produces $w$ with probability 1, and observe that $\psi$ does affect $\varphi$ in $\mathcal{M}$. ◀

As a consequence, checking inherent vacuity for probabilistic systems is as simple as in the non-probabilistic case, and can be done in polynomial space. In particular for Markov chains, an inherent vacuity check might be an interesting option for practical applications as it avoids the exponential blow-up in complexity over LTL-model checking.

## 5 Conclusion

We determined the precise complexities of the model-checking problems for the different levels of the quantifier alternation hierarchy of QLTL over probabilistic systems. The knowledge of the precise complexities, in particular the established lower bounds, has the potential to serve as the basis for hardness proofs for other questions in the formal verification of probabilistic systems. Despite the high complexities that we obtained, efficient model checking for formulas with few quantifier alternations might still be possible because all problems are solvable in time polynomial in the size of the system and typically formulas are small compared to the size of the models.

These results have been applied to the notion of trace vacuity known from the non-probabilistic setting that we adapted to the probabilistic setting. It turned out that checking whether a formula is affected by a subformula in a system is inter-reducible with $\Pi_1^{\mathrm{QLTL}}$-model checking. For Markov chains, our new lower bounds allowed us to conclude that affection checking is EXPSPACE-complete and hence exponentially harder than LTL-model checking, while the complexity of affection checking and LTL-model checking are the same in MDPs. Furthermore, we showed that the notion of inherent vacuity – expressing that a formula is vacuous in a class of system models – is invariant under the switch from non-probabilistic to probabilistic models, and hence, known polynomial-space algorithms are applicable for Markov chains and MDPs. In addition to the vacuity notions we studied here, an interesting direction for future research is the investigation of "more probabilistic" notions of vacuity that express that a perturbation of a subformula does not influence the satisfaction probability of a formula in a system.

## References


1 Roy Armoni, Limor Fix, Alon Flaisher, Orna Grumberg, Nir Piterman, Andreas Tiemeyer, and Moshe Y. Vardi. Enhanced vacuity detection in linear temporal logic. In *Proceedings of the 15th International Conference on Computer Aided Verification (CAV'03)*, volume 2725 of *Lecture Notes in Computer Science*, pages 368–380. Springer, 2003.

2 Christel Baier and Joost-Pieter Katoen. *Principles of Model Checking*. MIT Press, 2008.

3 Ilan Beer, Shoham Ben-David, Cindy Eisner, and Yoav Rodeh. Efficient detection of vacuity in temporal model checking. *Formal Methods in System Design*, 18(2):141–163, 2001.

4 Doron Bustan, Alon Flaisher, Orna Grumberg, Orna Kupferman, and Moshe Y. Vardi. Regular vacuity. In *Proceedings of the 13th IFIP WG 10.5 Advanced Research Working Conference on Correct Hardware Design and Verification Methods (CHARME'05)*, volume 3725 of *Lecture Notes in Computer Science*, pages 191–206. Springer, 2005.

5 Ashok K. Chandra, Dexter C. Kozen, and Larry J. Stockmeyer. Alternation. *Journal of the ACM*, 28(1):114–133, 1981. doi:10.1145/322234.322243.

6 Edmund Clarke, Orna Grumberg, and Doron Peled. *Model Checking*. MIT Press, 2000.

7 Costas Courcoubetis and Mihalis Yannakakis. The complexity of probabilistic verification. *Journal of the ACM*, 42(4):857–907, 1995.

8 Bernd Finkbeiner and Leander Tentrup. Detecting unrealizability of distributed fault-tolerant systems. *Logical Methods in Computer Science*, 11(3):1–31, 2015. doi:10.2168/LMCS-11(3:12)2015.

9 Dana Fisman, Orna Kupferman, Sarai Sheinvald-Faragy, and Moshe Y. Vardi. A framework for inherent vacuity. In *Proceedings of the 4th International Haifa Verification Conference (HVC'08)*, volume 5394 of *Lecture Notes in Computer Science*, pages 7–22. Springer, 2008. doi:10.1007/978-3-642-01702-5_7.

10 Dov Gabbay. The declarative past and imperative future. In B. Banieqbal, H. Barringer, and A. Pnueli, editors, *Temporal Logic in Specification*, pages 409–448, Berlin, Heidelberg, 1989. Springer Berlin Heidelberg.

11 Arie Gurfinkel and Marsha Chechik. Extending extended vacuity. In *Proceedings of the 5th International Conference on Formal Methods in Computer-Aided Design (FMCAD'04)*, volume 3312 of *Lecture Notes in Computer Science*, pages 306–321. Springer, 2004.

12 Arie Gurfinkel and Marsha Chechik. How vacuous is vacuous? In *Proceedings of the 10th International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS'04)*, pages 451–466, Berlin, Heidelberg, 2004. Springer Berlin Heidelberg.

13 Walter Hussak. Serializable histories in quantified propositional temporal logic. *International Journal of Computer Mathematics*, 81(10):1203–1211, 2004. doi:10.1080/00207160412331284051.

14 Yonit Kesten and Amir Pnueli. Complete proof system for QPTL. *Journal of Logic and Computation*, 12(5):701–745, 2002.

15 Orna Kupferman. Sanity checks in formal verification. In *Proceedings of the 17th International Conference on Concurrency Theory (CONCUR'06)*, volume 4137 of *Lecture Notes in Computer Science*, pages 37–51. Springer, 2006.

16 Orna Kupferman and Moshe Y. Vardi. Vacuity detection in temporal model checking. *International Journal on Software Tools for Technology Transfer*, 4(2):224–233, 2003.

17 François Laroussinie and Nicolas Markey. Quantified CTL: Expressiveness and Complexity. *Logical Methods in Computer Science*, 10(4):1–45, 2014. doi:10.2168/LMCS-10(4:17)2014.

18 Christos H. Papadimitriou. *Computational complexity*. Addison-Wesley, 1994.

19 Jakob Piribauer, Christel Baier, Nathalie Bertrand, and Ocan Sankur. Quantified linear temporal logic over probabilistic systems with an application to vacuity checking (extended version). Technical report, TU Dresden, Dresden, Germany, 2021. See https://wwwtcs.inf.tu-dresden.de/ALGI/PUB/CONCUR21/.

20 Martin L. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, Inc., New York, NY, 1994.

21 A. Prasad Sistla. *Theoretical Issues in the Design and Verification of Distributed Systems*. PhD thesis, Carnegie-Mellon University, 1983.

22 A. Prasad Sistla, Moshe Y. Vardi, and Pierre Wolper. The complementation problem for Büchi automata with applications to temporal logic. *Theoretical Computer Science*, 49(2–3):217–237, 1987.

23 Aravinda P. Sistla and Edmund M. Clarke. The complexity of propositional linear temporal logics. *Journal of the ACM*, 32(3):733–749, 1985. doi:10.1145/3828.3837.

24 Peter van Emde Boas. The convenience of tilings. *Lecture Notes in Pure and Applied Mathematics*, pages 331–363, 1997.

25 Moshe Y. Vardi and Pierre Wolper. An automata-theoretic approach to automatic program verification (preliminary report). In *Proceedings of the 1st Symposium on Logic in Computer Science (LICS'86)*, pages 332–344. IEEE Computer Society Press, 1986.

26 Pierre Wolper. Temporal logic can be more expressive. *Information and Control*, 56(1):72–99, 1983. doi:10.1016/S0019-9958(83)80051-5.