Available online at www.sciencedirect.com

ScienceDirect

Electronic Notes in Theoretical Computer Science 220 (2008) 61–77

Electronic Notes in Theoretical Computer Science

www.elsevier.com/locate/entcs

Model Checking Quantitative Linear Time Logic

Marco Faella

Università di Napoli "Federico II", Italy

Axel Legay

University of Liège, Belgium

Mariëlle Stoelinga

University of Twente, The Netherlands

Abstract

This paper considers QLtl, a quantitative analagon of Ltl and presents algorithms for model checking QLtl over quantitative versions of Kripke structures and Markov chains.

Keywords: Linear temporal logic, Quantitative verification, Automata.

1 Introduction

Quantitative properties, such as real-time and resource consumption, are essential in embedded system design. Hence, a wide variety of verification frameworks have been developed for the verification and validation of quantitative system aspects; we mention timed automata [3], probabilistic CTL [18], hybrid bisimilarity [19].

However, the analysis within these frameworks is still Boolean: either a timed automaton satisfies a property or not; two hybrid automata are either bisimilar or they are not. A Boolean approach to quantitative system analysis suffers from the drawback of being fragile: small perturbations in the values within the system description may lead to opposite truth values for the satisfaction of a property. This is problematic, since the system values are usually only known approximately, because they are often obtained by measurement, learning or educated guesses.

To circumvent this problem, quantitative methods for quantitative system analysis have been proposed [5,6,8,14]. These approaches are based on quantitative

doi:10.1016/j.entcs.2008.11.019

logics and quantitative system relations: whereas Boolean logics indicate whether a property holds for a system or not, quantitative logics express to what extend a property hold for a system; whereas (Boolean) bisimulations indicate whether or not two systems are equivalent, quantitative system relations (or distances) measure how similar two systems are. More specifically, [6] introduces QCtl, a quantitative analogon of Ctl, together with model checking procedures for it. [5] considers QLtl, a quantitative analogon of Ltl, and a quantitative μ-calculus and characterizes these by quantitative versions of trace equivalence and bisimilarity, respectively.

This paper continues the quest for quantitative verification and provides model checking procedures for QLtl over quantitative transition systems (QTSs) and quantitative Markov chains (QMCs). QTSs and QMCs are resp. Kripke structures and Markov chains whose atomic propositions have values in $[0, 1]$, rather than in $\{0, 1\}$. We also extend QLtl with a quantitative until operator, which is not present in [5]. Our logic is a particular instance of the general logic χLtl of [9], except that QLtl allows atomic propositions to be interpreted over an infinite and uncountable domain. Our model checking algorithm, although similar to the one of [9], is more direct, does not involve the complementation of the formula and exploits the separatedness of the automaton corresponding to the formula. Finally, our treatment of stochastic systems and the discussion on possible extensions to the logic are novel.

Our model checking procedure generalizes the classical Ltl model checking algorithms and constructs a Büchi automaton A_φ for each QLtl formula φ. Our construction of A_φ bears many similarities to the Ltl case: Recall that, for Ltl, each state q in A_φ is a subset of the *closure* of φ, which, roughly speaking, contains all subformulas of φ. In our case, each state q in A_φ assigns a value $\gamma(\psi)$ (from a finite subset of $[0, 1]$) to each formula ψ in the closure of φ. The correctness of this construction heavily relies on the fact that the Büchi conditions for the Boolean until operator immediately generalize to the quantitative case: for each formula $\psi_1 \mathcal{U} \psi_2$ in the closure of φ, we require that a trace accepted by A_φ hits infinitely many times a state where $\gamma(\psi_1 \mathcal{U} \psi_2) = \gamma(\psi_2)$. On the other hand, there are also several striking differences in the model checking algorithms for Ltl and QLtl. As an example, when model checking φ over a QTS $\mathcal{S}$, we consider $\mathcal{S} \times A_\varphi$, rather than $\mathcal{S} \times A_{\neg\varphi}$, hence avoiding any complementation operation. Indeed, contrary to the LTL case, we do not test whether all the executions satisfy the property, but rather compute the minimal value for which the property is satisfied. The latter can be done by combining the automaton and the system, looking for the minimal among all the accepting executions.

We show that for Markov chains, the model checking problem for QLtl reduces to the one for the Ltl case, and that it has no additional cost. One interesting aspect of our approach is that the automaton A_φ we build is separated, i.e. all states accept disjoint languages. Following [12], this allows us to avoid the use of Rabin automata, matching the well-known single exponential complexity bound proposed in [13].

Finally, we conclude the paper with several open problems and extensions. First, we state that the model checking procedure for QLTL over quantitative Markov decision processes is still partially open. Indeed, as for Markov chains, we show that this problem can be reduced to the model checking problem for LTL over Markov decision process. However, contrary to the Markov chain case, this reduction has an exponential cost. We also consider three extensions of QLTL. First, we deal with the logic QCTL*, which is obtained by adding path quantifiers $\exists$ and $\forall$ to QLTL. By interpreting the path quantifiers as in [6], we obtain that, on QMCs, model checking QCTL* is directly equivalent to model checking QLTL, while on QTSs it can be reduced to model checking QLTL. Further extensions include one where temporal operators are equipped with discount factors, and the another one featuring a long-run average operator. Model checking procedures for those two extensions is open, and the paper clearly states where the difficulties are.

Organization of the paper. In Section 2, we briefly recall some theory on automata over infinite words. Section 3 introduces QTS and QMC models, Section 4 presents the logic QLTL, while Section 5 treats our model checking algorithms. Then, we present in Section 6 several extensions to the theory and in Section 7 some conclusions.

2 Background on Infinite-Word Automata

We suppose the reader familiar with the theory of finite-word automata. We recall basic notions and definitions concerning infinite words and infinite-word automata. An *infinite word* (or ω*-word*) w over an alphabet Σ is a mapping $w : \mathbb{N} \to \Sigma$. The set of infinite words over Σ is denoted Σ^ω.

We consider sets of infinite words that can be represented by automata. Formally, an *infinite-word automaton* is a tuple $A = (\Sigma, Q, Q_0, \rho, F)$, where Σ is a finite alphabet, Q is a finite set of states, $Q_0 \subseteq Q$ is a non-empty set of initial states, $\rho : Q \times \Sigma \to 2^Q$ is a *nondeterministic* transition function, and F is an acceptance condition. The automaton A is said to be *deterministic* iff $|\rho(q, a)| = 1$ for each $q \in Q$ and $a \in \Sigma$. A *run* π of A on an infinite word w is a mapping $\pi : \mathbb{N} \to Q$, with $\pi_0 \in Q_0$, and for all $i \geq 0$, $\pi_{i+1} \in \rho(\pi_i, w_i)$.

Acceptance of a run π is defined in terms of the set $\mathit{inf}(\pi)$ of states that occur infinitely often in π, defined by $\mathit{inf}(\pi) = \{a \in \Sigma \mid \forall i > 0 \,.\, \exists j > i \,.\, \pi_j = a\}$. We consider the following types of acceptance conditions.

- A *Büchi condition* is a set $F \subseteq Q$ of accepting states. A set $T \subseteq Q$ is accepting for the Büchi condition if $T \cap F \neq \emptyset$.
- A *generalized Büchi condition* is a subset F of 2^Q. A set $T \subseteq Q$ is accepting for the generalized Büchi condition iff for each $F_i \in F$, $T \cap F_i \neq \emptyset$.
- A *Rabin condition* is a subset F of $2^Q \times 2^Q$, i.e., it is a collection of pairs of sets of states, written $[(L_1, U_1) \ldots (L_n, U_n)]$. A set $T \subseteq Q$ is accepting for the Rabin condition if $T \cap L_i \neq \emptyset$ and $T \cap U_i = \emptyset$ for some i.

A Büchi (resp. generalized Büchi, Rabin) automaton A is an automaton on infinite words with a Büchi (resp. a generalized Büchi, a Rabin) acceptance condition. A word w is accepted by A if there exists a run π on w such that the set $\mathit{inf}(\pi)$ is accepting with respect to the Büchi (resp. generalized Büchi, Rabin) condition. The set of infinite words accepted by A is called the language of A and is denoted by $L(A)$. For $Q' \subseteq Q$, we write $L_{Q'}(A)$ for the language accepted by A if Q' is the set of initial states. We write $L_q(A)$ for $L_{\{q\}}(A)$. The automaton A is *separated* if each word is accepted from at most one state; that is if $L_q(A) \cap L_{q'}(A) = \emptyset$ for all $q, q' \in Q$ with $q \neq q'$.

A Büchi condition is a special case of both the generalized Büchi and Rabin conditions. Hence, Büchi automata are not more expressive than generalized Büchi and Rabin automata. The opposite direction also holds. Moreover, unlike finite-word automata, infinite-word automata are not always closed under determinization: Büchi and generalized Büchi automata are not, while Rabin automata are closed under determinization. These facts yield the following result, which plays a significant role in many automata-based model checking algorithms.

Theorem 2.1 *Given a Büchi automaton A, there is a* deterministic *Rabin automaton A' such that $L(A) = L(A')$.*

Theorem 2.1 was first stated in [22], where a doubly exponential construction was provided. This construction was improved in [24], where a singly exponential, with an almost linear exponent, construction was provided: if A has n states, then A' has $2^{O(n \log n)}$ states and $O(n)$ pairs in its acceptance condition.

3 Quantitative Transition Systems and Markov Chains

3.1 Basic Definitions

For two real numbers u_1 and u_2, we write $u_1 \sqcup u_2$ for $\max\{u_1, u_2\}$, and $u_1 \sqcap u_2$ for $\min\{u_1, u_2\}$. Given a set E and a sequence $\pi = e_0 e_1 e_2 \cdots \in E^\omega$, we write π_i for the $i-$th element e_i of π, and we write $\pi^i = e_i e_{i+1} e_{i+2} \ldots$ for the (infinite) suffix of π starting from π_i. Let Σ be a finite set and $X \subseteq [0,1]$, we denote by $\mathit{vals}(\Sigma, X)$ the set of all functions from Σ to X. All elements of $\mathit{vals}(\Sigma, X)$ are called Σ-*valuations*; elements of $\mathit{vals}(\Sigma, \{0,1\})$ are called *boolean* Σ-valuations. We denote by $\mathit{trac}(\Sigma, X)$ the set of infinite sequences of valuations from $\mathit{vals}(\Sigma, X)$. All elements of $\mathit{trac}(\Sigma, X)$ are called Σ-*traces*. The *boolean* traces are the elements of $\mathit{trac}(\Sigma, \{0,1\})$.

3.2 Quantitative Transition Systems

A *quantitative transition system* (QTS for short) is an LTS whose atomic propositions, rather than being true of false, take values in $[0,1]$. Formally, a QTS $\mathcal{S} = (\Sigma, S, \delta, [\cdot])$ consists of a set Σ of atomic propositions, a finite set S of states, a transition relation $\delta \subseteq S \times S$, and a function $[\cdot]\colon S \to (\Sigma \to [0,1])$ which assigns to each state $s \in S$ and proposition $r \in \Sigma$ a real value $[s](r)$. A QTS is boolean if, for all s, $[s]$ is boolean. The *size* of $\mathcal{S}$ is given by its number of transitions, i.e.

$|\mathcal{S}| = |\delta|$.

A *path* in $\mathcal{S}$ is an infinite sequence $\pi = s_0 s_1 s_2 \dots$ of states such that $(s_i, s_{i+1}) \in \delta$ for all $i \in \mathbb{N}$. Given a state s, we write $\mathit{pts}(s)$ for the set of all paths starting in s. Every path π in $\mathcal{S}$ induces the Σ-trace $[\pi] = [\pi_0][\pi_1][\pi_2]\dots$. With an abuse of notation, we write $\mathit{trac}(s) = \{[\pi] \mid \pi \in \mathit{pts}(s)\}$ for the set of Σ-traces from $s \in S$. Notice that $\mathit{trac}(s) \subseteq \mathit{trac}(\Sigma, [0,1])$.

3.3 Quantitative Markov Chains

Given a finite set S, a *probability distribution* on S is a function $\mu : S \to [0,1]$ such that $\sum_{s \in S} \mu(s) = 1$. We denote by $\mathcal{D}(S)$ the set of all probability distributions on S.

A *quantitative Markov chain* (QMC for short) is a Markov chain whose propositions take values in $[0,1]$. Thus, in each state of a QMC, the successor state is choosen probabilistically. Formally, a QMC $\mathcal{S} = (\Sigma, S, \Delta, [\cdot])$ consists of a set Σ of atomic propositions, a finite set S of states, a transition relation $\Delta : S \to \mathcal{D}(S)$, and a function $[\cdot] : S \to (\Sigma \to [0,1])$. The *size* of $\mathcal{S}$ is given by $|S|^2$. A QMC $(\Sigma, S, \Delta, [\cdot])$ induces a QTS $(\Sigma, S, \delta, [\cdot])$, where $\delta = \{(s,t) \in S^2 \mid \Delta(s)(t) > 0\}$. Definitions for paths and traces in a QMC are identical to those for the corresponding QTS.

A quantitative Markov chain together with an initial state s gives rise to a *probability space* $(\mathit{trac}(s), \mathcal{B}, \Pr_s)$, where $\mathcal{B}$ is the set of measurable subsets of $\mathit{trac}(s)$, and $\Pr_s$ is the uniquely induced probability measure (see [11] for an introduction). Given a *random variable* X over this probability space, we denote its expected value by $\mathrm{E}_s[X]$.

When discussing the complexity of algorithms taking a QMC as input, we assume that transition probabilities are encoded as fixed-precision numbers, and therefore that arithmetic operations and comparisons take constant time.

4 Quantitative LTL

In this section we introduce *Quantitative Linear Temporal Logic* (QLtl for short), a quantitative version of the *Linear Temporal Logic* (Ltl for short) introduced in [23].

4.1 Syntax

Let Σ be a set of atomic propositions. The QLtl formulas over Σ are generated by the following grammar:

$$\varphi \ ::= \ r \mid \mathrm{T} \mid \mathrm{F} \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \neg\varphi \mid \varphi \mathcal{U} \varphi \mid \varphi \tilde{\mathcal{U}} \varphi \mid \bigcirc\varphi \mid \Diamond\varphi \mid \Box\varphi$$

where $r \in \Sigma$. The operators $\mathcal{U}$, $\tilde{\mathcal{U}}$, $\Diamond$, and $\Box$ are the *temporal operators*. The syntax of QLtl is therefore the same as the one of Ltl.

4.2 Semantics

Here $r \in \Sigma$ is an atomic proposition. A QLtl formula φ over Σ assigns a real value $[\![\varphi]\!](\sigma) \in [0,1]$ to each $\Sigma-$trace σ as follows.

$$\begin{array}{llllll}
[\![r]\!](\sigma) = & \sigma_0(r) & [\![\neg\varphi]\!](\sigma) = & \mathbf{1} - [\![\varphi]\!](\sigma) & [\![\bigcirc\varphi]\!](\sigma) = & [\![\varphi]\!](\sigma^1) \\
[\![\mathrm{T}]\!](\sigma) = & \mathbf{1} & [\![\varphi \vee \psi]\!](\sigma) = & [\![\varphi]\!](\sigma) \sqcup [\![\psi]\!](\sigma) & [\![\Diamond\varphi]\!](\sigma) = & \sup_{i \geq 0} [\![\varphi]\!](\sigma^i) \\
[\![\mathrm{F}]\!](\sigma) = & \mathbf{0} & [\![\varphi \wedge \psi]\!](\sigma) = & [\![\varphi]\!](\sigma) \sqcap [\![\psi]\!](\sigma) & [\![\Box\varphi]\!](\sigma) = & \inf_{i \geq 0} [\![\varphi]\!](\sigma^i)
\end{array}$$

$$\begin{aligned}
[\![\varphi \mathcal{U} \psi]\!](\sigma) &= \sup_{i \geq 0} \big([\![\varphi]\!](\sigma^0) \sqcap \cdots \sqcap [\![\varphi]\!](\sigma^{i-1}) \sqcap [\![\psi]\!](\sigma^i)\big) \\
[\![\varphi \tilde{\mathcal{U}} \psi]\!](\sigma) &= \inf_{i \geq 0} \big([\![\varphi]\!](\sigma^0) \sqcup \cdots \sqcup [\![\varphi]\!](\sigma^{i-1}) \sqcup [\![\psi]\!](\sigma^i)\big).
\end{aligned}$$

The semantics of $\mathcal{U}$ can be understood as follows. In the Boolean case, $\varphi\mathcal{U}\psi$ holds on the boolean trace σ there exists an i such that $[\![\varphi]\!](\sigma^0) \wedge \ldots [\![\varphi]\!](\sigma^{i-1}) \wedge [\![\psi]\!](\sigma^i)$. In the quantitative case, $\wedge$ corresponds to $\sqcap$ and the existential quantification of i corresponds to the supremum over all i. The semantics of $\tilde{\mathcal{U}}$ is obtained in a similar way.

The following result is an immediate consequence of the fact that each finite set has a minimum.

Lemma 4.1 *Let φ, ψ be* QLtl *formulas, let $\sigma \in trac(\Sigma, X)$ where X is a finite set. Then*

$$\begin{aligned}
[\![\varphi \mathcal{U} \psi]\!](\sigma) &= \max_{i \geq 0} \big([\![\varphi]\!](\sigma^0) \sqcap \cdots \sqcap [\![\varphi]\!](\sigma^{i-1}) \sqcap [\![\psi]\!](\sigma^i)\big) \\
[\![\varphi \tilde{\mathcal{U}} \psi]\!](\sigma) &= \min_{i \geq 0} \big([\![\varphi]\!](\sigma^0) \sqcup \cdots \sqcup [\![\varphi]\!](\sigma^{i-1}) \sqcup [\![\psi]\!](\sigma^i)\big).
\end{aligned}$$

Evaluation over quantitative transition systems. A QLtl formula φ assigns a real value $[\![\varphi]\!](s) \in [0,1]$ to each state s of a given QTS, according to the rule $[\![\varphi]\!](s) = \inf\{[\![\varphi]\!](\sigma) \mid \sigma \in trac(s)\}$. Taking the infimum over all σ corresponds to the fact that an Ltl formula holds in a state if it hold for all traces from that state.

Evaluation over quantitative Markov chains. Given a QMC $\mathcal{S}$, a state s, and a QLtl formula φ, the function $[\![\varphi]\!]$, which assigns a real value to each Σ-trace, is a random variable over the probability space $(trac(s), \mathcal{B}, \Pr_s)$. Accordingly, we define the value of φ on state s to be $[\![\varphi]\!](s) = \mathrm{E}_s[[\![\varphi]\!]]$.

4.3 Properties of QLtl

Extension of Ltl**.** The semantics of QLtl is a proper extension of the one of Ltl: for Boolean traces σ, we have that $[\![\varphi]\!](\sigma) = 1$ if $\sigma \models \varphi$ and $[\![\varphi]\!](\sigma) = 0$ otherwise. Here, $\models$ is the classical LTL satisfaction relation.

Equivalences and positive normal forms. Two QLtl formulas φ and ψ are equivalent, notation $\varphi \equiv \psi$, if they assign the same value to each trace. Formally, we have $\varphi \equiv \psi$ iff $[\![\varphi]\!](\sigma) = [\![\psi]\!](\sigma)$ for all traces σ. Obviously, if two formalas agree on all traces, they will also agree on all QLTSs and QMCs.

As for LTL, each QLTL formula can be transformed an equivalent formula in positive normal form where negations only appear in front of the atomic propositions. This transformation relies on the following equivalences.

$$\neg(\varphi_1 \mathcal{U} \varphi_2) \equiv (\neg\varphi_1)\tilde{\mathcal{U}}(\neg\varphi_2)$$
$$\neg(\varphi_1 \tilde{\mathcal{U}} \varphi_2) \equiv (\neg\varphi_1)\mathcal{U}(\neg\varphi_2)$$
$$\neg(\bigcirc\varphi) \equiv \bigcirc(\neg\varphi).$$

Moreover, the following classical equivalences also hold for QLTL:

$$\Box\varphi \equiv \mathrm{F}\tilde{\mathcal{U}}\varphi \qquad\qquad \Diamond\varphi \equiv \mathrm{T}\mathcal{U}\varphi.$$

From now on, we only consider formulas in positive normal form without occurence of $\Box$ and $\Diamond$. Notice that we cannot remove both T and F from the logic, since $p \vee \neg p \not\equiv \mathrm{T}$ (i.e. the Law of the Excluded Middle does not hold) and $p \wedge \neg p \not\equiv \mathrm{F}$.

5 Evaluating QLTL

In this section, we extend the automata-based technique by [28,29] to determine the valuation of a QLTL formula on a QTS or a QMC. First, we prove that if along a trace all atomic propositions only take a finite number of different values, any QLTL formula assigns to that trace either one of the values occurring in the trace, or $1 - x$, where x is a value occurring in the trace. The formula can also directly assign values 0 and 1 using constants T and F. As a corollary, when evaluated on a QTS $\mathcal{S}$, a QLTL formula can only assume value in $\mathcal{V}(\mathcal{S})$.

Theorem 5.1 *Let $\mathcal{V}$ be a finite subset of $[0, 1]$ and let $\sigma \in trac(\Sigma, \mathcal{V})$. Then, for all* QLTL *formulas φ, we have $[\![\varphi]\!](\sigma) \in \{0, 1\} \cup \mathcal{V} \cup \{1 - x \mid x \in \mathcal{V}\}$.*

Proof. By induction on φ, using Lemma 4.1. □

Corollary 5.2 *Given a QTS $\mathcal{S}$, a state $s \in S$, and a* QLTL *formula φ, we have $[\![\varphi]\!](s) \in \mathcal{V}(\mathcal{S})$.*

5.1 Closure labeling

Next,we consider the two following definitions.

Definition 5.3 The *closure* of a QLTL formula φ is the smallest set $clos(\varphi)$ of

QLTL formulas such that:

$$\begin{aligned}
\varphi &\in \mathit{clos}(\varphi) \\
\psi_1 \vee \psi_2 \in \mathit{clos}(\varphi) &\implies \psi_1, \psi_2 \in \mathit{clos}(\varphi) \\
\psi_1 \wedge \psi_2 \in \mathit{clos}(\varphi) &\implies \psi_1, \psi_2 \in \mathit{clos}(\varphi) \\
\bigcirc\psi_1 \in \mathit{clos}(\varphi) &\implies \psi_1 \in \mathit{clos}(\varphi) \\
\psi_1 \mathcal{U} \psi_2 \in \mathit{clos}(\varphi) &\implies \psi_1, \psi_2 \in \mathit{clos}(\varphi) \\
\psi_1 \tilde{\mathcal{U}} \psi_2 \in \mathit{clos}(\varphi) &\implies \psi_1, \psi_2 \in \mathit{clos}(\varphi).
\end{aligned}$$

We denote by $|\varphi|$ the number of temporal operators, Boolean connectives and propositions found in the formula φ. Notice that $|\mathit{clos}(\varphi)| = O(|\varphi|)$.

Definition 5.4 A *closure-valuation* for a QLTL formula φ is a function $v : \mathit{clos}(\varphi) \to [0,1]$. A closure-valuation is *consistent* if the following conditions hold.

(i) If $\mathrm{T} \in \mathit{clos}(\varphi)$, then $v(\mathrm{T}) = 1$.

(ii) If $\mathrm{F} \in \mathit{clos}(\varphi)$, then $v(\mathrm{F}) = 0$.

(iii) If $\psi_1 \vee \psi_2 \in \mathit{clos}(\varphi)$, then $v(\psi_1 \vee \psi_2) = v(\psi_1) \sqcup v(\psi_2)$.

(iv) If $\psi_1 \wedge \psi_2 \in \mathit{clos}(\varphi)$, then $v(\psi_1 \wedge \psi_2) = v(\psi_1) \sqcap v(\psi_2)$.

(v) If both r and $\neg r$ belong to $\mathit{clos}(\varphi)$, then $v(\neg r) = 1 - v(r)$.

A *closure-trace* is an infinite sequence of consistent closure-valuations.

To determine the value of a QLTL formula on a Σ-trace one can proceed by building a closure-trace in a way that is compatible with QLTL semantics. Consider a closure-trace γ for a formula φ defined over a set of atomic propositions Σ. For a Σ-trace σ, we say that γ is *valid for* σ if, for each $i{\geq}0$, it satisfies the rules (i) - (vi) below. (adapted from [29]):

(i) For each $r \in \Sigma$, if $r \in \mathit{clos}(\varphi)$ then $\gamma_i(r) = \sigma_i(r)$, and if $\neg r \in \mathit{clos}(\varphi)$ then $\gamma_i(\neg r) = 1 - \sigma_i(r)$.

(ii) If $\gamma_i(\bigcirc\psi_1) = u$ then $\gamma_{i+1}(\psi_1) = u$.

For the $\mathcal{U}$ and $\tilde{\mathcal{U}}$ operators, the semantics rules refer to a possibly infinite set of points of the sequence. The solution is first to notice that the following identities hold for each $i{\geq}0$:

$$\begin{aligned}
[\![\psi_1 \mathcal{U} \psi_2]\!](\sigma^i) &= [\![\psi_2]\!](\sigma^i) \sqcup ([\![\psi_1]\!](\sigma^i) \;\sqcap\; [\![\bigcirc(\psi_1 \mathcal{U} \psi_2)]\!](\sigma^i)) \\
[\![\psi_1 \tilde{\mathcal{U}} \psi_2]\!](\sigma^i) &= [\![\psi_2]\!](\sigma^i) \sqcap ([\![\psi_1]\!](\sigma^i) \;\sqcup\; [\![\bigcirc(\psi_1 \mathcal{U} \psi_2)]\!](\sigma^i)).
\end{aligned}$$

These identities suggest the following labeling rules for each $i \geq 0$:

(iii) If $\gamma_i(\psi_1 \mathcal{U} \psi_2) = u$, then $u = \gamma_i(\psi_2) \sqcup (\gamma_i(\psi_1) \sqcap \gamma_{i+1}(\psi_1 \mathcal{U} \psi_2))$.

(iv) If $\gamma_i(\psi_1 \tilde{\mathcal{U}} \psi_2) = u$, then $u = \gamma_i(\psi_2) \sqcap (\gamma_i(\psi_1) \sqcup \gamma_{i+1}(\psi_1 \tilde{\mathcal{U}} \psi_2))$.

5.2 The acceptance conditions

However, as is illustrated by the following example, those conditions are not sufficient for the closure-trace to be valid.

Example 5.5 Consider the QLTL formula $p\mathcal{U}q$ with $p, q \in \Sigma$. We have $clos(\varphi) = \{p\mathcal{U}q, p, q\}$. Consider now a closure-trace that constantly assigns values 0.6, 0.7, and 0.3 to $p\mathcal{U}q$, p, and q, respectively. This trace is valid for any Σ-trace that always assigns the value 0.7 to p and 0.3 to q. However the evaluation of $p\mathcal{U}q$ on such a trace would be 0.3, and thus not 0.6 as it is suggested by the closure-trace.

The problem in the example above is that when only considering rules (iii) and (iv), the evaluation of $p\mathcal{U}q$ can always be postponed to the next element in the sequence. The solution is to observe that since the systems on which QLTL formulas are evaluated are finite-state systems, we can restrict ourselves to a finite subset of $[0, 1]$. In this setting, we obtain the following result.

Theorem 5.6 *Consider a* QLTL *formula of the form* $\varphi_1\mathcal{U}\varphi_2$ *(resp.* $\varphi_1\tilde{\mathcal{U}}\varphi_2$*). Let* $\mathcal{V}$ *be a finite subset of* $[0, 1]$ *and let* $\sigma \in trac(\Sigma, \mathcal{V})$*. For all* $i{\geq}0$ *there exists* $j{\geq}i$ *such that* $[\![\varphi_1\mathcal{U}\varphi_2]\!](\sigma^j) = [\![\varphi_2]\!](\sigma^j)$ *(resp.* $[\![\varphi_1\tilde{\mathcal{U}}\varphi_2]\!](\sigma^j) = [\![\varphi_2]\!](\sigma^j)$*).*

Proof. Let $i \in \mathbb{N}$. Let $j = \text{argmax}_{k>i}[\![\varphi_2]\!](\sigma^k)$, that is, j is an index where a maximum φ_2 value is attained, i.e. j is such that $[\![\varphi_2]\!](\sigma^j) = \max_{k>i}[\![\varphi_2]\!](\sigma^k)$. Such an index j exists because $\mathcal{V}$ is finite and therefore so is the set of possible values of φ_2. We claim that $[\![\varphi_1\mathcal{U}\varphi_2]\!](\sigma^j) = [\![\varphi_2]\!](\sigma^j)$. On the one hand

$$[\![\varphi\mathcal{U}\psi]\!](\sigma^j) \ = \sup\nolimits_{k\geq 0}\left([\![\varphi]\!](\sigma^j) \sqcap \cdots \sqcap [\![\varphi]\!](\sigma^{j+k-1}) \sqcap [\![\psi]\!](\sigma^{j+k})\right) \geq [\![\psi]\!](\sigma^j).$$

On the other hand, using Lemma 4.1, there is a k such that

$$\begin{aligned}[\![\varphi\mathcal{U}\psi]\!](\sigma^j) \ &= [\![\varphi]\!](\sigma^j) \sqcap \cdots \sqcap [\![\varphi]\!](\sigma^{j+k-1}) \sqcap [\![\psi]\!](\sigma^{j+k}) \\ &\leq [\![\psi]\!](\sigma^{j+k}) \\ &\leq [\![\psi]\!](\sigma^j).\end{aligned}$$

The $\tilde{\mathcal{U}}$ case is proved similarly. □

As a consequence of Theorem 5.6, we add the following labeling rules, which only have sense when considering $\mathcal{V}$ to be a finite subset of $[0, 1]$:

(v) For each $i{\geq}0$, there exists $j{\geq}i$ such that $\gamma_j(\psi_1\mathcal{U}\psi_2) = \gamma_j(\psi_2)$.

(vi) For each $i{\geq}0$, there exists $j{\geq}i$ such that $\gamma_j(\psi_1\tilde{\mathcal{U}}\psi_2) = \gamma_j(\psi_2)$.

The following result states that the six labeling rules (i)-(vi) completely characterize the semantics of a QLTL formula. Its proof is an easy induction on the structure of the subformulas of φ.

Theorem 5.7 *Consider a* QLTL *formula* φ*, a finite set* $\mathcal{V} \subseteq [0, 1]$ *and a* Σ*-trace* $\sigma \in trac(\Sigma, \mathcal{V})$*. Then* γ *is a valid closure-trace for* σ *if and only if* $[\![\psi]\!](\sigma^i) = \gamma_i(\psi)$ *for all* $\psi \in clos(\varphi)$ *and* $i \geq 0$*.*

Corollary 5.8 *Consider a* QLTL *formula* φ*, a finite set* $\mathcal{V} \subseteq [0,1]$ *and a* Σ*-trace* $\sigma \in \mathit{trac}(\Sigma, \mathcal{V})$*. We have that* $[\![\varphi]\!](\sigma) = u$ *if and only if there exists a valid closure-trace* γ *for* σ *such that* $\gamma_0(\varphi) = u$*.*

5.3 The Büchi automaton for QLTL *model checking*

Given a QLTL formula φ, we now build a generalized Büchi automaton that describes a possibly infinite set of Σ-traces and whose states are consistent closure-valuations of φ. More precisely, the automaton is built in such a way that for each formula ψ_1 in the closure of φ, for each state q, and for each accepting Σ-trace σ from s, the valuation $[\![\psi_1]\!](\sigma)$ is given by $q(\psi_1)$.

Definition 5.9 Let Σ be a set of atomic propositions and let $\mathcal{V}$ be a finite subset of $[0,1]$ such that, for all $x \in \mathcal{V}$, $1 - x \in \mathcal{V}$. We define the QLTL-*automaton* for φ and $\mathcal{V}$ as the tuple $A_\varphi^\mathcal{V} = (\mathit{vals}(\Sigma, \mathcal{V}), Q, Q_0, \rho, F)$, where:

- The alphabet of the automaton is $\mathit{vals}(\Sigma, \mathcal{V})$.
- The set of states Q is the set of closure-valuations in $\mathit{vals}(\mathit{clos}(\varphi), \mathcal{V})$ which are consistent.
- We choose $Q_0 = Q$.
- The transition function is such that for each $q, q' \in Q$ and $a \in \mathit{vals}(\Sigma, \mathcal{V})$, we have $q' \in \rho(q, a)$ iff

(i) For all $r \in \Sigma$, if $r \in \mathit{clos}(\varphi)$ (resp. $\neg r \in \mathit{clos}(\varphi)$), then $q(r) = a(r)$ (resp. $q(\neg r) = 1 - a(r)$).

(ii) If $\bigcirc\psi_1 \in \mathit{clos}(\varphi)$, then $q(\bigcirc\psi_1) = q'(\psi_1)$.

(iii) If $\psi_1 \mathcal{U} \psi_2 \in \mathit{clos}(\varphi)$, then $q(\psi_1 \mathcal{U} \psi_2) = q(\psi_2) \sqcup (q(\psi_1) \sqcap q'(\psi_1 \mathcal{U} \psi_2))$.

(iv) If $\psi_1 \tilde{\mathcal{U}} \psi_2 \in \mathit{clos}(\varphi)$, then $q(\psi_1 \tilde{\mathcal{U}} \psi_2) = q(\psi_2) \sqcap (q(\psi_1) \sqcup q'(\psi_1 \mathcal{U} \psi_2))$.

- If $\mathit{clos}(\varphi)$ contains no formula with $\mathcal{U}$ and $\tilde{\mathcal{U}}$ operators, then $F = \{Q\}$. Otherwise, for each formula of the form $\psi_1 \mathcal{U} \psi_2$ (resp. $\psi_1 \tilde{\mathcal{U}} \psi_2$) in $\mathit{clos}(\varphi)$, F contains the set $Q_{\psi_1 \mathcal{U} \psi_2}$ (resp. $Q_{\psi_1 \tilde{\mathcal{U}} \psi_2}$), where $q \in Q_{\psi_1 \mathcal{U} \psi_2}$ iff $q(\psi_1 \mathcal{U} \psi_2) = q(\psi_2)$ (resp. $q(\psi_1 \tilde{\mathcal{U}} \psi_2) = q(\psi_2)$).

Observe that the number of states of $A_\varphi^\mathcal{V}$ is bounded by $|\mathcal{V}|^{|\mathit{clos}(\varphi)|}$. In practice the bound is not reached since one only considers consistent closures. The set of initial states is defined arbitrarily and will be discussed in the next section. Corollary 5.11 states the correctness of the construction of $A_\varphi^\mathcal{V}$. The key to this correctness property is given by Theorem 5.10.

Theorem 5.10 *Let* φ *be a* QLTL *formula and let* $\mathcal{V}$ *be a finite subset of* $[0,1]$ *such that if* $x \in \mathcal{V}$*, then also* $1 - x \in \mathcal{V}$*. Furthermore, let* $\pi : \mathbb{N} \to Q$ *be an infinite sequence of states in* $A_\varphi^\mathcal{V}$*. Then* π *is an accepting run from state* q *in* $A_\varphi^\mathcal{V}$ *if and only if the following conditions hold.*

(i) π *is a valid closure-trace for* σ*, which is given by* $\sigma_i(r) = \pi(r)$ *for all* $r \in \Sigma$*.*

(ii) $\pi(\psi) = q(\psi)$ *for all* $\psi \in \mathit{clos}(\varphi)$*.*

Corollary 5.11 *Let φ be a* QLtl *formula and let $\mathcal{V}$ be a finite subset of* $[0, 1]$ *such that if $x \in \mathcal{V}$, then also $1 - x \in \mathcal{V}$. Furthermore, let $\sigma \in vals(\Sigma, \mathcal{V})$ be an Σ-trace. Then*

(i) *$\sigma \in L_q(A^{\mathcal{V}}_\varphi)$ if an only if $[\![\psi]\!](\sigma) = q(\psi)$ for all $\varphi \in clos(\varphi)$;*

(ii) *for all $v \in \mathcal{V}$ and all ψ in the closure of φ we have*

$$\bigcup_{q \in Q, q(\psi)=v} L_q(A^{\mathcal{V}}_\varphi) = \{\sigma \in vals(\Sigma, \mathcal{V}) \mid [\![\psi]\!](\sigma) = v\}.$$

The automaton $A^{\mathcal{V}}_\varphi$ also satisfies the following property, which will be of particular interest for the results that will be presented in Section 5.5.

Theorem 5.12 *The automaton $A^{\mathcal{V}}_\varphi$ is separated.*

Proof. For each $q, q' \in Q$ with $q \neq q'$, there exists $\varphi_1 \in clos(\varphi)$ such that $q(\varphi_1) \neq q'(\varphi_1)$. Since it is not possible that a Σ-trace assigns two different values to the same formula φ_1, we have $L_q(A^{\mathcal{V}}_\varphi) \cap L_{q'}(A^{\mathcal{V}}_\varphi) = \emptyset$. □

5.4 Evaluating QLtl *on Quantitative Transition Systems*

Consider a quantitative transition system $\mathcal{S} = (\Sigma, S, \delta, [\cdot])$ and a QLtl formula φ. We aim at computing $[\![\varphi]\!](s)$ for a state $s \in S$. We first propose the following definition.

Definition 5.13 Consider a QTS $\mathcal{S} = (\Sigma, S, \delta, [\cdot])$ and a QLtl formula φ. Let $A_\varphi = (vals(\Sigma, \mathcal{V}(\mathcal{S})), Q, Q_0, \rho, F)$ be the QLtl-automaton for φ and $\mathcal{V}(\mathcal{S})$. For a state $\bar{s} \in S$, the *$\bar{s}$-product* of $\mathcal{S}$ and A_φ, denoted $\mathcal{S} \times A_\varphi$, is the automaton $(\{\emptyset\}, Q', Q'_0, \rho', F')$, where:

- The alphabet contains only the symbol $\emptyset$.
- The set of states Q' contains all pairs $(s, q) \in S \times Q$ which are synchronized w.r.t. the value of the atomic propositions. Formally, for all $r \in clos(\varphi)$, $[s](r) = q(r)$.
- The set of initial states is given by $Q'_0 = (\{\bar{s}\} \times Q_0) \cap Q'$.
- The set of final states is given by $F' = (S \times F) \cap Q'$.
- We have $(s', q') \in \rho'((s, q), \emptyset)$ iff $(s, s') \in \delta$.

Our approach to computing $[\![\varphi]\!](\bar{s})$ consists in the following three steps:

(i) We first build the $\bar{s}$-product $\mathcal{S} \times A_\varphi$ between the system $\mathcal{S}$ and the QLtl-automaton for φ and $\mathcal{V}(\mathcal{S})$.

(ii) We then compute the set of states $Q'' = \{(\bar{s}, q) \in Q'_0 \mid L_{\{(\bar{s},q)\}}(\mathcal{S} \times A_\varphi) \neq \emptyset\}$.

(iii) Finally, $[\![\varphi]\!](\bar{s}) = \min_{(\bar{s},q) \in Q''} q(\varphi)$.

As far as the complexity of the above procedure is concerned, it is easy to see that step (ii) dominates the others. Such step consists in determining the set Q'' of states of the product automaton which, used as initial states, give rise to a non-empty language. The classical algorithm for the emptiness of a generalized Büchi automaton can be easily adapted to compute the set Q'' in time linear in the size of the product (precisely, in the number of edges in the product). We thus obtain the following theorem which states that the model checking procedure for QLTL is not more expensive than the one for model checking LTL.

Theorem 5.14 *Given a* QLTL*-formula* φ*, a QTS* $\mathcal{S} = (\Sigma, S, \delta, [\cdot])$*, and a state* $s \in S$*, the value* $[\![\varphi]\!](s)$ *can be computed in time* $O(|\delta| \cdot |\mathcal{V}(\mathcal{S})|^{|clos(\varphi)|})$*.*

Notice that, unlike the LTL case, our evaluation procedure does not need to complement a Büchi automaton or a QLTL formula.

5.5 *Evaluating* QLTL *on Quantitative Markov Chains*

In this section, we consider the model checking problem for QLTL over quantitative Markov chains. We will show that this problem can be reduced to the model checking problem for LTL over Markov chains.

Consider a QMC $\mathcal{S} = (\Sigma, S, \Delta, [\cdot])$ and a QLTL formula φ on Σ. We aim at computing $E_s[[\![\varphi]\!]]$ for a state $s \in S$. Assuming that $\mathcal{V}(\mathcal{S}) = \{b_1, b_2, \ldots, b_n\}$, recall that we denote by $\mathrm{Pr}_s[[\![\varphi]\!] = b_i]$ the probability for the value of the random variable $[\![\varphi]\!]$ to be b_i on the probability space generated by the traces starting at s. We have

$$[\![\varphi]\!](s) = \mathrm{E}_s[[\![\varphi]\!]] = \sum_{i=1}^{n} b_i \cdot \mathrm{Pr}_s[[\![\varphi]\!] = b_i].$$

Consequently, to compute $[\![\varphi]\!](s)$, it is sufficient to compute for each value $b_i \in \mathcal{V}(\mathcal{S})$ the probability for the value of the random variable to be b_i. More precisely, given the set of Σ-traces $T^{b_i} = \{\sigma \in \mathit{trac}(\mathcal{S}) \mid [\![\varphi]\!](\sigma) = b_i\}$ and the probability space $(\mathit{trac}(s), \mathcal{B}, \mathrm{Pr}_s)$ given by $\mathcal{S}$ and s, we aim at computing $\mathrm{Pr}_s(T^{b_i})$. For this, we recall the following theorem (see [11] for a proof).

Theorem 5.15 *Consider a QMC* $\mathcal{S} = (\Sigma, S, \Delta, [\cdot])$*, a state* $s \in S$*, and the probability space* $(\mathit{trac}(s), \mathcal{B}, \mathrm{Pr}_s)$ *given by* $\mathcal{S}$ *and* s*. Let* T *be a set of* Σ*-traces. If* T *can be represented by a deterministic Rabin automaton with* n *states, then one can compute* $\mathrm{Pr}_s(T)$ *in time polynomial in* $|S| \cdot n$*.*

We thus need to provide a deterministic Rabin automaton $A_{\varphi=b_i}$ accepting T^{b_i}, for each b_i. Working with a deterministic Rabin automaton is needed not to break the deterministic behavior of Markov chain (see [11]). The automaton $A_{\varphi=b_i}$ can easily be obtained from the automaton A_φ. Indeed, it suffices to remove from the set of initial states of A_φ all the states that do not assign the value b_i to φ. We obtain a generalized Büchi automaton, which can be turned into a deterministic Rabin one whose size is exponentially larger (see Theorem 2.1).

The result above involves a double exponential, which is due to the fact that we build a generalized Büchi automaton for the formula (whose size is exponential in the size of the formula), and then turn it into a deterministic Rabin one (whose size is again exponential in the size of the Büchi). However, in [12], it is showed that one can avoid the exponential blow-up needed to compute the deterministic

Rabin automaton, when the generalized Büchi automaton representing the formula is separated. Observe that since the automaton A_φ is separated, any automaton $A_{\varphi=b_i}$ will also be separated. We can thus use the result from [12] to avoid one exponential blow-up. The resulting algorithm is polynomial in the size of the QMC and singly exponential w.r.t. the formula.

Remark 5.16 The result in [12] additionally requires automata to be *unambiguous*. An automaton is unambiguous if two transitions that start in the same state and have the same label reach different destinations. This property is satisfied by our automata by definition. In conclusion, the automaton A_φ is separated and unambiguous. Moreover, this property does not depend on the set of initial states.

6 Extensions and Open Problems

This section discusses several extensions of QLTL model checking. First, we describe how our QLTL model checking algorithm can be extended to an algorithm for QCTL*. Then, we present a partial solution to the model checking problem for a discounted version of QLTL. Finally, model checking the long-run average operator and quantitative Markov decision processes remain completely open.

6.1 From QLTL *to* QCTL*

Having considered the branching logic DCTL in [6] and the linear logic QLTL in this paper, it is natural to consider logic QCTL*, which extends QLTL with path quantifiers $\exists$ and $\forall$. The syntax of QCTL* is the same as the one of CTL*. The semantics of a QCTL* formula is defined with respect to the system on which it is evaluated. Consider a QCTL* formula φ.

- If the formula is evaluated for a state s of a quantitative transition system $\mathcal{S}$, then the operators $\forall$ and $\exists$ represent the inf-evaluation-over-all and the sup-evaluation-over-all traces, respectively. Formally, $[\![\forall\varphi]\!](s) = \inf\{[\![\varphi]\!](\sigma) \mid \sigma \in \mathit{trac}(s)\}$, and $[\![\exists\varphi]\!](s) = \sup\{[\![\varphi]\!](\sigma) \mid \sigma \in \mathit{trac}(s)\}$. Observe also that $[\![\exists\varphi]\!](s) = 1 - [\![\forall\neg\varphi]\!](s)$.
- When considering quantitative Markov chains, following [6], we interpret both $\forall$ and $\exists$ as the expected value operator. Therefore, on QMCs, QCTL* essentially coincides with QLTL.

Evaluating QCTL* formulas with only one path quantifier is immediate. Indeed, the automata-based algorithm presented in Section 5.4 allows us to immediately evaluate formulas of the form $\exists\varphi$. Observing that $[\![\exists\varphi]\!](s) = 1 - [\![\forall\neg\varphi]\!](s)$, we get the result. When considering formulas with several path quantifiers, one recursively replaces each quantified subformula with a new atomic proposition that represents its value (using again the automata-based algorithm). We thus have the following complexity result.

Theorem 6.1 *Consider a QTS $\mathcal{S} = (\Sigma, S, \delta, [\cdot])$, a state $s \in S$, and a* QCTL**-formula φ. The value $[\![\varphi]\!](s)$ can be computed in time $O(|\varphi| \cdot |\mathcal{S}| \cdot |\mathcal{V}(\mathcal{S})|^{|clos(\varphi)|})$.*

Recall from Theorem 5 of [6] that a formula φ from the logic DCTL can be evaluated on a QTS $\mathcal{S}$ in time $O(|S|^2 \cdot |\varphi|)$. It follows that QLTL formulas which do not contain nesting of linear operators can be evaluated in the same time. As a side note, it should be noted that DCTL does not feature an until operator. However, it is our belief that its addition would not increase the complexity of model checking the logic. We therefore observe, as expected, that ad-hoc algorithms for dealing directly with $\Box$ and $\Diamond$ operators are definitely more efficient than the present automata-based algorithms, which on the other hand is capable of treating arbitrary nesting of temporal operators.

6.2 Discounting

The logics in [6] use discounting, meaning that values in the near future weigh more than values in the far future. Given a *discount factor* $\alpha \in [0,1]$, discounted versions $\bigcirc_\alpha$, $\Diamond_\alpha$, and $\Box_\alpha$ of the next, eventually, and always operator are defined below. There is a second next operator $\widehat{\bigcirc}_\alpha$, which is the dual of $\bigcirc_\alpha$

$$\begin{aligned} [\![\bigcirc_\alpha \varphi]\!](\sigma) &= \ \alpha[\![\varphi]\!](\sigma^1) & [\![\Diamond_\alpha \varphi]\!](\sigma) &= \ \sup_{i\geq 0} \alpha^i [\![\varphi]\!](\sigma^i) \\ [\![\widehat{\bigcirc}_\alpha \varphi]\!](\sigma) &= 1-\alpha+\alpha[\![\varphi]\!](\sigma^1) & [\![\Box_\alpha \varphi]\!](\sigma) &= \ \inf_{i\geq 0} 1-\alpha^i(1-[\![\varphi]\!](\sigma^i)). \end{aligned}$$

Just as for the next operator, one should consider two discounted variants $\mathcal{U}_\alpha$ and $\widehat{\mathcal{U}}_\alpha$ of $\mathcal{U}$ (and also two for $\tilde{\mathcal{U}}$)

$$\begin{aligned} [\![\varphi \mathcal{U}_\alpha \psi]\!](\sigma) =& [\![\psi]\!](\sigma) \sqcup \\ & \sup_{i>0} \alpha^0 [\![\varphi]\!](\sigma^0) \sqcap \alpha^1 [\![\varphi]\!](\sigma^1) \sqcap \cdots \sqcap \alpha^{i-1}[\![\varphi]\!](\sigma^{i-1}) \sqcap \alpha^i [\![\psi]\!](\sigma^i) \\ [\![\varphi \widehat{\mathcal{U}}_\alpha \psi]\!](\sigma) =& [\![\psi]\!](\sigma) \sqcup \sup_{i>0} 1-\alpha^0(1-[\![\varphi]\!](\sigma^0)) \sqcap 1-\alpha^1(1-[\![\varphi]\!](\sigma^1)) \sqcap \dots \\ & \sqcap 1-\alpha^{i-1}(1-[\![\varphi]\!](\sigma^{i-1})) \sqcap 1-\alpha^i(1-[\![\psi]\!](\sigma^i)). \end{aligned}$$

For $\psi_1 \mathcal{U}_\alpha \psi_2$, we have $[\![\psi_1 \mathcal{U}_\alpha \psi_2]\!](\sigma^i) = [\![\psi_2]\!](\sigma^i) \sqcup ([\![\psi_1]\!](\sigma^i) \sqcap [\![\bigcirc_\alpha(\psi_1 \mathcal{U}_\alpha \psi_2)]\!](\sigma^i))$ and thus the following labeling rule.

$$\gamma_i(\psi_2) \ \sqcup \ (\gamma_i(\psi_1) \sqcap \alpha \cdot \gamma_{i+1}(\psi_1 \mathcal{U} \psi_2)).$$

The other until operators can be treated similarly. It is important to realize that no Büchi conditions are needed for $\alpha < 1$: in the undiscounted case, the recursive characterization $\psi_1 \mathcal{U} \psi_2 \equiv \psi_2 \vee (\psi_1 \wedge \bigcirc(\psi_1 \mathcal{U} \psi_2))$ for $\mathcal{U}$ has two fixed points and one needs the smallest. If $\alpha < 1$, then the underlying operators are contractions and have unique fixed points.

The analogon of Theorem 5.1 does not hold in the case of discounting. Simple examples show that, given a QTS $\mathcal{S}$, the set of values $\mathcal{V}_\alpha(\mathcal{S}) = \{[\![\varphi]\!](\mathcal{S}) \mid \varphi \text{ is a QLTL formula with discount factor } \alpha\}$ is in general infinite. However, by performing the construction of Definition 5.9 with $\mathcal{V}$ being an *infinite* subset of $[0,1]$, one can build an infinite-state Büchi automaton with the property of Theorem 5.11. In other words, the Büchi construction works for discounting, but we cannot use it for model checking, since it yields an infinite-state automaton. Alternative model-checking methods should therefore be investigated, e.g. based on approximation.

6.3 Long-run Average Operator

The branching logic DCTL of [6] also contains the path operator $\triangle$ ("triangle"). This operator stands for the long-run average of a quantitative proposition and is defined by:

$$[\![\triangle\varphi]\!](\sigma) = \lim_{n\to\infty} \frac{[\![\varphi]\!](\sigma^0) + [\![\varphi]\!](\sigma^1) + \cdots + [\![\varphi]\!](\sigma^{n-1})}{n}.$$

Such operator does not fit well with the finite automata-based approach, since the value of a $\triangle p$ formula in general does not coincide with the value of the proposition p in any state of the system: for instance, any number in $[0,1]$ can be obtained as the long run average of a sequence whose propositional values are $\{0,\ 0.1,\ 0.2 \ldots 0.9\}$.

Thus, it remains open whether the $\triangle$ operator can be evaluated on a system by automata-theoretic means.

6.4 Model checking QLTL *over Quantitative Markov Decision Processes*

A Quantitative Markov Decision Process (QMDP) is a Markov decision process (MDP) with quantitative values in the states. Thus, a QMDP can be viewed as a QMC combined with nondeterminism, i.e. each QMDP state enables one or more transitions whose target state is determined probabilistically. Formally, a QMDP $\mathcal{S} = (\Sigma, S, \Delta, [\cdot])$ contains the same ingredients as a QMC, except that the transition relation is a function $\Delta : S \to 2^{\mathcal{D}(S)}$ such that $\Delta(s) \neq \emptyset$ for each $s \in S$. Each QMDP induces a QTS $(\Sigma, S, \delta, [\cdot])$, where $\delta = \{(s,t) \in S^2 \mid \exists \mu \in \Delta(s) \,.\, \mu(t) > 0\}$. Definitions for paths and traces in a QMDP are identical to those for the corresponding QTS.

A *scheduler* for $\mathcal{S}$ resolves the non-deterministic choices in $\mathcal{S}$. Schedulers can be (1) history-dependent, i.e. they may base their decisions on the history of the system, and (2) randomized, i.e. they may make a probabilistic choice over the outgoing transitions in each state. More precisely, a scheduler for $\mathcal{S}$ in a state s_0 is a function $\pi : \mathit{pts}(s_0) \to \mathcal{D}(\mathcal{D}(S))$ such that if $\pi(s_0 s_1 \ldots s_n)(\mu) > 0$, then $\mu \in \Delta(s_n)$. We denote the set of all schedulers in s_0 by $\mathit{Sched}(s_0)$. Each scheduler π in state s defines a probability space over $\mathcal{P}^\pi_s = (\mathit{trac}^\pi(s), \mathcal{B}^\pi, \Pr^\pi_s)$, where $\mathcal{B}^\pi$ is the set of measurable subsets of $\mathit{trac}^\pi(s)$, and $\Pr^\pi_s$ is the uniquely induced probability measure over $\mathcal{B}^\pi$. We denote the expected value of a random variable X over $\mathcal{P}^\pi_s$ by $\mathrm{E}^\pi_s[X]$.

We interpret QLTL over QMDPs by taking the minimum expected value over all schedulers, i.e. we set $[\![\varphi]\!](s) = \inf_{\pi \in \mathit{Sched}(s)} \mathrm{E}^\pi_s[[\![\varphi]\!]]$.

Unfortunately, contrary to the case of Markov chains, we cannot directly extend the algorithm for LTL model checking over MDPs [1,2,25]: we could, for each value b_i run an LTL-inspired algorithm that finds the minimum probability $\min_{\pi \in \mathit{Sched}} \Pr^\pi_s[[\![\varphi]\!] = b_i]$ with which the value b_i is attained. However, the QMDP model checking problem asks for the global minimum, i.e. $\min_{\pi \in Sched} \sum_{i=1}^{n} b_i \cdot \Pr_s^\pi[[\![\varphi]\!] = b_i]$, which cannot be found by solving the model checking problem for the b_i's separately [1] . We did not find a way to solve this global minimization problem based on LTL model checking. Thus, the QLTL model checking problem over QMDPs is open.

7 Conclusion and Future Work

In this paper, we extended the work done in [6], by presenting a quantitative linear temporal logic and showing how such logic can be model-checked (i.e., evaluated) over non-deterministic or probabilistic systems, by using a classical automata-based approach. We have provided partial solutions to the model checking problem for QCTL*, and over quantitative Markov decision processes. Model checking of the long run average operator and the discounted version of QLTL is open.

Apart from the directions mentioned in Section 6, it is also worthwhile to investigate an extension of the results presented in this paper (and in [6]) to continuous time or interval Markov chains. Another promising research direction consists in extending the abstract probabilistic frameworks of [16,20] to quantitative logics. We could also investigate whether the alternating automata based construction of [21] extends to the case of QCTL*. Finally, it would also be of interest to see whether one can reduce the size of the automata we construct following techniques similar to those proposed in [17].

References

[1] L. de Alfaro, *Formal Verification of Probabilistic Systems*, Phd Thesis, Stanford University, 1997.

[2] L. de Alfaro and A. Bianco. *Model Checking of Probabilistic and Nondeterministic Systems*, Proc. Int. Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS), Lecture Notes in Computer Science, Volume 1026, 1995, pages 499–513.

[3] R. Alur and D. L. Dill, *A theory of timed automata*, Theoretical Computer Science, 126(2):183–235, 1994.

[4] L. de Alfaro and T. A. Henzinger and R. Majumdar, *Discounting the future in Systems Theory*, Proc of ICALP, Lecture Notes in Computer Science, Volume 2719, 2003, pages 1022–1037.

[5] L. de Alfaro and M. Faella and M. Stoelinga, *Linear and Branching Metrics for Quantitative Transition Systems*, "Proc. Int. Colloquium on Automata, Languages and Programming (ICALP), Lecture Notes in Computer Science, Volume 3142, 2004, pages 97–109.

[6] L. de Alfaro and M. Faella and T. A. Henzinger and R. Majumdar and M. Stoelinga, *Model checking discounted temporal properties*, Theoretical Computer Science, volume 345, number 1, 2005, pages 139–170.

[7] L. de Alfaro and R. Majumdar and V. Raman and M. Stoelinga, *Game Relations and Metrics*, Proc. IEEE Symposium on Logic in Computer Science (LICS), IEEE, 2007, pages 99–108.

[8] F. van Breugel and J. Worrel, *Towards quantitative verification of probabilistic systems*, Proc. 28th Int. Colloq. Aut. Lang. Prog., volume 2076 of Lect. Notes in Comp. Sci., pages 421–432. Springer-Verlag, 2001.

[9] M. Chechik and B. Devereux and A. Gurfinkel, *Model-Checking Infinite State-Space Systems with Fine-Grained Abstractions Using SPIN*, Proc. of SPIN Workshop on Model-Checking Software, 2001.

[1] Indeed, since the minimal probability for each of the b_i's could be computed with a different scheduler, the sum of all the probabilities could be greater than 1, which breaks the definition of the expected value.

[10] F. Ciesinski and C. Baier, *LiQuor: A tool for Qualitative and Quantitative Linear Time analysis of Reactive Systems*, Proc. Int. Conference on the Quantitative Evaluaiton of Systems (QEST), IEEE, 2006, pages 131–132.

[11] F. Ciesinski and M. Größer, *On Probabilistic Computation Tree Logic*, Validation of Stochastic Systems - A guide to Current Research, Lecture Notes in Computer Science, volume 2925, 2004, pages 147–188.

[12] J-M. Couvreur and N. Saheb and G. Sutre, *An Optimal Automata Approach to LTL Model Checking of Probabilistic Systems*, Proc. Int. Conference on Logic for Programming, Artificial Intelligence, and Reasoning (LPAR), LNAI, Volume 2850, 2003, pages 361–375.

[13] C. Courcoubetis and M. Yannakakis, *The Complexity of Probabilistic Verification*, Journal of the ACM, Volume 42(4), 1995, pages 857–907.

[14] J. Desharnais, V. Gupta, R. Jagadeesan, and P. Panangaden, *Approximating labelled markov processes*, Information and Computation, 2002.

[15] J. Esparza and A. Kucera and R. Mayr, *Model Checking Probabilistic Pushdown Automata*, Proc. 5th Symposium on Logic in Computer Science (LICS), IEEE, 2004, pages 12–21.

[16] H. Fesher and M. Leucker and V. Wolf. *Don't know in Probabilistic Systems*, Proc. Int. Spin Workshp, Lecture Notes in Computer Science, Volume 3925, 2006.

[17] R. Gerth and D. Peled and M. Y. Vardi and P. Wolper, *Simple on-the-fly automatic verification of linear temporal logic*, Proc. Int. Symposium on Protocol Specification, Testing and Verification, IFIP Conference Proceedings, Volume 38, 1995, pages 3–18.

[18] H. Hansson and B. Jonsson, *A logic for reasoning about time and reliability*, Formal Aspects of Computing, 6(5):512–535, 1994.

[19] T. A. Henzinger, *The theory of hybrid automata*, Proc. IEEE Symposium on Logic in Computer Science (LICS), New Brunswick, New Jersey, 1996.

[20] M.Z. Kwiatkowska and G. Norman and D. Parker, *Game-based Abstraction for Markov Decision Processes*, Proc. Int. Conference on the Quantitative Evaluaiton of Systems (QEST), IEEE, 2006, pages 157–166.

[21] O. Kupferman and M. Y. Vardi and P. Wolper, *An automata-theoretic approach to branching-time model checking*, Journal of the ACM, volume 47, number 2, 2000, pages 312–360.

[22] R. McNaughton, *Testing and Generating infinite sequences by a finite automaton*, Information and control, 1966, pages 521–530.

[23] A. Pnueli, *The Temporal Logic of Programs*, Proc. Annual Symposium on Foundations of Computer Science (FOCS), 1977, pages 46–57.

[24] S. Safra, *Complexity of Automata on Infinite Objects*, Phd Thesis, Weizmann Institute of Science, 1989.

[25] M. Y. Vardi, *Automatic Verification of Probabilistic Concurrent Finite-State Programs*, Proc of FOCS, IEEE, 1985, pages 327–338.

[26] M. Y. Vardi, *Probabilistic Linear-Time Model Checking: An Overview of the Automata-Theoretic Approach*, Proc of Int. AMAST Workshop, Lecture Notes in Computer Science, Volume 1601, 1999, pages 265–276.

[27] M. Y. Vardi, *The Büchi Complementation Saga*, Proc. Int. Annual Symposium on Theoretical Aspects of Computer Science, Lecture Notes in Computer Science, Volume 4393, 2007, pages 12-22.

[28] M. Y. Vardi and P. Wolper, *An Automata-Theoretic Approach to Automatic Program Verification (Preliminary Report)*, Proc. IEEE Symposium on Logic in Computer Science (LICS), IEEE, 1986, pages 332–344.

[29] P. Wolper, *Constructing Automata from Temporal Logic Formulas: A Tutorial*, Proc of European Educational Forum: School on Formal Methods and Performance Analysis, Lecture Notes in Computer Science, Volume 2090, 2000, pages 261–277.