Theoretical Computer Science 534 (2014) 69–85

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

Parameterized model checking of weighted networks

Ingmar Meinecke [1], Karin Quaas [*,2]

Institut für Informatik, Universität Leipzig, D-04109 Leipzig, Germany

ARTICLE INFO

Keywords:
Model checking
Weighted automata
Parameterized model checking
LTL

ABSTRACT

We consider networks of weighted processes that consist of an arbitrary number of weighted automata equipped with weights taken from a monoid. We investigate parameterized model checking problems used to verify whether certain qualitative and quantitative properties hold independently of the number of processes. We prove that model checking properties expressed in a weighted extension of LTL is decidable if the monoid satisfies some simple properties. We further present decision procedures for checking global properties of weighted networks.

1. Introduction

In this paper, we consider networks of many identical processes. Such a network consists of one control process and an arbitrary number of user processes that behave according to the same finite-state process definition. A single process can either execute an internal transition or communicate with another process via handshake synchronization. Each transition may be assigned some value that represents, e.g., emission, energy, money or time that is produced or consumed during the transition. The resulting model, which we call *weighted networks*, can thus be used to model concurrent systems with resources.

We are interested in verifying *qualitative* and *quantitative* properties of weighted networks. For instance, we would like to decide whether the control process can always reach a certain state without breaking a given local budget. Or we may ask for global properties, e.g.: If the controller is in some critical state, do all user processes keep their energy level within a certain interval? Hereby, a certain property should hold *independently* of the number of instantiated user processes, i.e., the number of processes is not fixed.

German and Sistla [14] investigated such parameterized model checking problems for unweighted networks by using Linear Temporal Logic (LTL) as a specification language. To incorporate quantitative aspects, we extend LTL to a quantitative version, called *weighted LTL* (wLTL) by assigning intervals from the weight structure as a constraint to the temporal operators. In this way, wLTL is syntactically very similar to the well-established real-time logic MTL [15,5,20,21] and its variants [4,9,17], or extensions of MTL with several cost functions [8]. However, the interval constraints in our logic wLTL may stem from more general structures than in MTL, similarly to the way it is done for weighted automata and weighted MSO logic over semirings, cf. [11]. But here we do not need a second operation (like the addition of a semiring) to resolve non-determinism because we are only interested in the computation of a weight along a path. Thus, we allow interval constraints from *monotone ordered monoids*, i.e., monoids with a linear order such that the operation is monotone with respect to the order. Also the transitions of the networks are equipped with weights from these structures. Monotone ordered monoids cover the

* Corresponding author.
E-mail addresses: meinecke@informatik.uni-leipzig.de (I. Meinecke), quaas@informatik.uni-leipzig.de (K. Quaas).

[1] The first author was supported by *Deutsche Forschungsgemeinschaft (DFG)*, project DR 202/11-1.
[2] The second author was supported by *Deutsche Forschungsgemeinschaft (DFG)*, project QU 316/1-1.

http://dx.doi.org/10.1016/j.tcs.2014.02.037
0304-3975/

non-negative reals with addition, used in MTL, as well as the non-negative rationals with maximum or the positive integers with multiplication.

The semantics of wLTL formulas are languages of *weighted words*, i.e., every letter of the infinite word is made up of a set of atomic propositions and a weight. Whenever we restrict the weights that may occur in a weighted word to a finite set, we can translate *weighted* LTL formulas into *unweighted* Büchi automata recognizing the semantics of the formula over the restricted alphabet. This translation works for every monotone ordered monoid that is *bounded locally finite*, a property that ensures that finitely many values can generate, up to a certain bound, only finitely many values. The size of the Büchi automaton can be optimized depending on the underlying monoid.

As a main result, we show decidability of the *controller model checking problem* in a quantitative setting, i.e., it is decidable whether each execution of the control process satisfies a wLTL formula *independently* of the number of user processes. An analogue result holds for user model checking. Using the translation of a weighted formula into an unweighted automaton, we can treat the weights of the processes as mere labels. This way, we can reuse significant concepts and results of [14] for our quantitative model checking procedure.

Weighted LTL formulas express *local* properties of the controller or a user. We also investigate *global* properties specifying both control and user processes of weighted networks at the same time. Model checking networks against formulas from an extension of LTL with universal process quantifiers is undecidable already in the unweighted setting [14]. In our setting, simultaneous satisfaction of a wLTL formula by the controller and all user processes also turns out to be undecidable. Thus, we restrict to problems concerning the feasibility of computations for which the overall weight for every process stays within certain bounds. By reduction to the controller model checking problem, we can show decidability of several of those bounded weight problems. However, if we turn to a more global weight model, then global model checking problems turns out to be undecidable, cf. Theorem 5.5.

Related work. As already noted, German and Sistla [14] showed decidability of local LTL model checking and of several global properties for unweighted networks. In [1], Abdulla and Jonsson introduce *timed* networks, in which the behavior of the user processes is defined by a single-clock timed automaton. In contrast to our result, for timed networks the controller model checking problem is undecidable [1]. However, safety properties can be decided. For a weighted extension of a similar model, interesting optimal reachability problems are investigated in recent works [2,3].

Our contribution profits from the elaborated work done for a timed setting. In [17], the authors explore durational Kripke structures and model checking with TCTL and TLTL. There, transitions are labeled with intervals of non-negative integers. The authors show that TLTL model checking is EXPSPACE-complete whereas a fragment of TLTL can be decided in PSPACE. We rewin these results for tight durational Kripke structures, i.e., those with singleton intervals at the transitions, by our translation of wLTL into Büchi automata, cf. Theorem 4.4. The result of [17] for TCTL is mainly based on former work on MTL [15] over both a discrete [5,6] and dense time domain [4].

The work on MTL model checking over a dense time domain was taken up again both for a state-based [9] and an event-based semantics [20,21]. Ouaknine and Worrell [21] could show decidability of MTL model checking over finite timed words and of a safety fragment for infinite words. However, general MTL model checking and satisfiability turn out to be undecidable for infinite timed words [20]. A weighted extension of MTL was explored in [8] where now several cost functions are allowed for the constraints of the temporal operators. However, decidability is limited to one clock and an additional simple stopwatch cost variable.

In [7], the authors explore temporal logics with prefix-accumulation assertions as atomic formulas. Whereas such an assertion may speak about several weight variables with values in $\mathbb{Q}$, we restrict here in wLTL to one weight variable (which has to take non-negative values). In contrast to our setting, the weight variables are not reset in [7] but the weight is accumulated from the beginning. However, the use of several weight variables over $\mathbb{Q}$ leads to undecidability of the model checking problem for most temporal logics with prefix-accumulation assertions. Only the EF fragment turns out to be decidable [7].

In a broader spectrum, quantitative LTL-model checking of a single system over more general weight structures was studied, e.g., for distributive lattices [16], and the max-plus-semiring with discounting parameters [18]. For bounded lattices, a recent work generalizes classical results on the coincidence of aperiodicity, star-freeness, and lattice-valued first-order- and LTL-definability [12].

2. Weighted networks

A *weighted network* represents a family of weighted systems, each consisting of a controller process and a finite, but arbitrary number of user processes. Both the controller and the users are modeled by finite-state transition systems communicating by handshake, i.e., by synchronized actions. Moreover, each transition is assigned some weight.

Let $\mathbb{Z}$, $\mathbb{Q}$, $\mathbb{N}$ and $\mathbb{Q}_{\geqslant 0}$ denote the set of integers, rationals, non-negative integers, and non-negative rationals, respectively. Let Σ_+ be a finite alphabet and $\Sigma_- = \{\bar{a} \mid a \in \Sigma_+\}$ a disjoint copy of Σ_+. We will use the two alphabets Σ_+ and Σ_- for modeling handshake communications: a handshake of two processes is realized by the synchronized execution of a letter $a \in \Sigma_+$ and its copy $\bar{a} \in \Sigma_-$. Put $\Sigma = \Sigma_+ \cup \Sigma_- \cup \{\varepsilon\}$ where $\varepsilon \notin \Sigma_+$. The letter ε represents internal actions of a single process. By Σ^ω we denote the set of infinite words over Σ. Let *AP* be a finite set of atomic propositions.