Averaging in LTL

Patricia Bouyer, Nicolas Markey, and Raj Mohan Matteplackel

LSV – CNRS & ENS Cachan – France

Abstract. For the accurate analysis of computerized systems, powerful quantitative formalisms have been designed, together with efficient verification algorithms. However, verification has mostly remained boolean—either a property is true, or it is false. We believe that this is too crude in a context where quantitative information and constraints are crucial: correctness should be quantified!
In a recent line of works, several authors have proposed quantitative semantics for temporal logics, using e.g. *discounting* modalities (which give less importance to distant events). In the present paper, we define and study a quantitative semantics of LTL with *averaging* modalities, either on the long run or within an until modality. This, in a way, relaxes the classical Boolean semantics of LTL, and provides a measure of certain properties of a model. We prove that computing and even approximating the value of a formula in this logic is undecidable.

1 Introduction

Formal verification of computerized systems is an important issue that aims at preventing bugs in the developed computerized systems. The model-checking approach to verification consists in automatically checking that the model of a system satisfies a correctness property. The standard approach is therefore a yes/no (that is, boolean) approach: either the system satisfies the specified property, or the system does not satisfy the property. Model-checking has been widely developed and spread over the last 35 years and is a real success story.

In many applications, quantitative information is crucial; quantities can already appear at the functional level of the system (such as timing constraints between events, or bounds on various quantities like the energy consumption, ...), and many quantitative models like timed automata [4] and their weighted extension [5,7] have therefore been proposed and studied. But quantities can even have more impact on the quality of the system: how good is a system w.r.t. a property? In that case the standard boolean approach might appear as too crude: among those systems that are incorrect (in a boolean sense), some might still be better than others. In order to take this into account, the model-checking approach to verification has to be lifted to a more quantitative perspective [18]. This would allow to *quantify* the quality of systems, and to investigate their tolerance to slight perturbations.

Partly supported by ERC Starting Grant EQualIS and EU FP7 project Cassting.

There are three classical approaches for turning standard model checking to a quantitative perspective. A first approach, building on automata-based techniques to model checking, consists in defining quantitative semantics for finite state automata. This uses weighted automata [21,16], with different possible semantics. Quantitative decision problems for this setting are addressed in [13,15]. A second approach consists in defining distances between models, or between models and specifications, that can provide an accurateness measure of the model w.r.t. the specification. This approach has been developed e.g. in [12], and then extended into the *model measuring* problem [19]. A third approach is to define quantitative specification languages. For probabilistic systems, this approach is rather standard, and quantitative logics like CSL have been defined and used for model-checking [6]. More recently, this approach has been developed for quantitative but non-stochastic systems. We give more details on those approaches in the "related work" paragraph below.

Example 1 (Jobshop scheduling). Consider a finite set of machines, on which we want to schedule finitely many jobs with possibly dependencies between jobs. Standard analysis asks for the existence of a scheduler that satisfies some scheduling policy, or for optimal such schedulers. A more quality-oriented approach could consist in evaluating the average load along a schedule, or the least machine usage, or the average idle time of a given machine. Those cannot be expressed as a standard boolean model-checking question. ◁

Example 2 (Mobile-phone server). Consider a server that should acknowledge any request by some grant (representing the range of frequency—the bigger the range, the larger the grant). Then the quality of such a server could be expressed as the average over all requests of the range that is allocated in response. This cannot be expressed as a standard boolean model-checking question. ◁

In this paper, we propose quantitative measures of correctness based on the linear-time temporal logic LTL. More precisely, we propose a natural extension of LTL, called avgLTL, with two natural averaging modalities: a new average-until operator $\psi_1 \,\widetilde{\mathbf{U}}\, \psi_2$ that computes the average value of ψ_1 along the path until ψ_2 has a high value, and where the semantics of standard modalities are extended using a min-max approach; and a long-run average operator $\widetilde{\mathbf{G}}\, \psi$, which computes the limit of the values of ψ in the long run along the path. Developing the two examples above, we will show that this logic can express interesting properties.

We focus on the model-checking problem, which corresponds to computing the value of a run (or a Kripke structure) w.r.t. a given property, and on the corresponding decision (comparison with a threshold) and approximation problems. We show that all variants (*i.e.*, all kinds of thresholds, and both when the model is a single path and when it is a Kripke structure) of model-checking and approximation problems are undecidable. Such a robust undecidability is rather surprising (at least to us), given the positive results of [2] for a discounted semantics for LTL, of [22] for an extension of LTL with mean-payoff constraints. Despite the undecidability result for frequency-LTL (a boolean extension of LTL with frequency-constrained "until" modality) and for LTL with average assertions

over weighted Kripke structures [8,10], we had hope that some variants of our problem would be decidable.

However we believe these undecidability results are interesting in several respects. (*i*) First, up to now (see related work below), quantitative specification languages based on LTL have always involved discounting factors, which allows to only consider a bounded horizon; this helps obtaining decidability results. In several papers though, averaging in LTL is mentioned, but left as open research directions. (*ii*) Also, we prove robust undecidability results, in the sense that undecidability is proven both for model-checking over a path and model-checking a Kripke structure, and for all thresholds; note that many cases require a specific proof. (*iii*) Finally, our proof techniques are non-trivial and may be interesting in other contexts; we were not able to get a direct encoding of two-counter machines for proving the undecidability of the model-checking problem over Kripke structures, and had to use a diagonal argument; this is due to convergence phenomena that arise in the context of quantitative model-checking, and which have mostly been omitted so far in the rest of the literature.

Related work. Several recent papers have proposed quantitative-verification frameworks based on temporal logic. The authors of [14] were the first to suggest giving temporal logics a quantitative semantics: they extend CTL with various new modalities involving a discount on the future (the later the event, the smaller the impact on the value of the formula). In that framework, model-checking is proven decidable.

As regards linear-time temporal logics, a first attempt to define a quantitative semantics has been proposed in [17]. However, no modality is really quantitative, only the models are quantitative, yielding finitely non-boolean values. Still, the authors suggest discounting and long-run averaging as possible extensions of their work. Another approach is tackled in [1], where functions f are added to the syntax of LTL, with the value of $f(\psi_1, \ldots, \psi_k)$ on a path π being the result of applying f to the values of subformulas $\psi_1, \ldots, \psi_k$ on π. As explained in [1], this quantitative language is not that expressive: each formula only takes finitely many values. It follows that the verification problems are decidable.

Frequency-LTL , an extension of LTL with "frequency-until", has been studied in [9], and even though it has a boolean semantics, the frequency modality gives a quantitative taste to the logic: $\phi_1 \, \mathbf{U}^{\,c} \phi_2$ holds true along a path whenever there is a position along that path at which ϕ_2 holds, and the frequency of ϕ_1 along the prefix is at least c. This paper shows the undecidability of the satisfiability problem. We discuss this approach in more details in Section 8, since it shares some techniques with ours.

Finally the recent work [2] is the closest to ours. It studies LTL extended with a discounted until modality: roughly, the values of the subformulas are multiplied by a discount factor, which decreases and tends to zero with the distance to the evaluation point. This way, the further the witness, the lower the value. An automata-based algorithm is given to decide the threshold problem. Due to discounting, whether the value of a formula is larger than some threshold on a path can be checked on a bounded prefix of the path. On the other hand, adding

local average (*i.e.*, the average of finitely many subformulas) yields undecidability (for the existence of a path with value 1/2). We will discuss with more details this paper in Section 8.

2 Average-LTL

Let $\mathcal{P}$ be a finite set of atomic propositions. A *quantitative Kripke structure* over $\mathcal{P}$ is a 4-tuple $\mathcal{K} = \langle V, v_0, E, \ell \rangle$ where V is a finite set of vertices, $v_0 \in V$ is the initial vertex, $E \subseteq V \times V$ is a set of transitions (which we assume total, meaning that for each $v \in V$, there exists $v' \in V$ s.t. $(v, v') \in E$) and $\ell \colon V \to ([0,1] \cap \mathbb{Q})^{\mathcal{P}}$ is a labelling function, associating with each state the value of each atomic proposition in that state. The Kripke structure $\mathcal{K}$ is said *qualitative* whenever for every $v \in V$ and $p \in \mathcal{P}$, $(\ell(v))(p) \in \{0,1\}$. A run or path in a Kripke structure $\mathcal{K}$ from $v \in V$ is a finite or infinite sequence $\pi = (v_i)_{i \in I}$ (where I is a (bounded or unbounded) interval of $\mathbb{N}$ containing 0) s.t. $v_0 = v$ and $(v_{i-1}, v_i) \in E$ for all relevant $i \in I \setminus \{0\}$. The size $|\pi|$ of π is the cardinality of I. In the sequel, we will be interested in the sequence $\ell(\pi) = (\ell(v_i))_{i \in I}$, and we will often identify a run with the sequence in $(([0,1] \cap \mathbb{Q})^{\mathcal{P}})^I$ it defines. Given a run $\pi = (v_i)_{i \in I}$ and an integer j, we write $\pi_{\geq j}$ for the run $(v_{i+j})_{i \geq 0, i+j \in I}$.

We now introduce the logic average-LTL (avgLTL for short) and its interpretation over infinite runs. The syntax of avgLTL over $\mathcal{P}$ is given by:

$$\varphi ::= p \mid \neg p \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \mathbf{X}\,\varphi \mid \varphi\,\mathbf{U}\,\varphi \mid \mathbf{G}\,\varphi \mid \varphi\,\widetilde{\mathbf{U}}\,\varphi \mid \widetilde{\mathbf{G}}\,\varphi.$$

where $p \in \mathcal{P}$. Notice that negation is only allowed on atomic propositions. We write LTL for the fragment where $\widetilde{\mathbf{U}}$ and $\widetilde{\mathbf{G}}$ are not allowed.

Let $\pi = (v_i)_{i \in \mathbb{N}}$ be an infinite run, and φ be an avgLTL formula. The valuation $[\![\pi, \varphi]\!]$ is then given as follows:

$$\begin{aligned}
[\![\pi, p]\!] &= (\ell(v_0))(p) & [\![\pi, \neg p]\!] &= 1 - (\ell(v_0))(p) \\
[\![\pi, \psi_1 \vee \psi_2]\!] &= \max\{[\![\pi, \psi_1]\!], [\![\pi, \psi_2]\!]\} & [\![\pi, \mathbf{X}\,\psi]\!] &= [\![\pi_{\geq 1}, \psi]\!] \\
[\![\pi, \psi_1 \wedge \psi_2]\!] &= \min\{[\![\pi, \psi_1]\!], [\![\pi, \psi_2]\!]\} \\
[\![\pi, \mathbf{G}\,\psi]\!] &= \inf_{i \in \mathbb{N}} [\![\pi_{\geq i}, \psi]\!] \\
[\![\pi, \psi_1\,\mathbf{U}\,\psi_2]\!] &= \sup_{i \in \mathbb{N}} \min\big\{[\![\pi_{\geq i}, \psi_2]\!], \min_{0 \leq j < i}([\![\pi_{\geq j}, \psi_1]\!])\big\} \\
[\![\pi, \widetilde{\mathbf{G}}\,\psi]\!] &= \liminf_{i \to \infty} \big(\textstyle\sum_{j=0}^{j<i} [\![\pi_{\geq j}, \psi]\!]\big)/i \\
[\![\pi, \psi_1\,\widetilde{\mathbf{U}}\,\psi_2]\!] &= \sup\Big(\{[\![\pi, \psi_2]\!]\} \cup \big\{\min\big\{[\![\pi_{\geq i}, \psi_2]\!], \big(\textstyle\sum_{j=0}^{j<i} [\![\pi_{\geq j}, \psi_1]\!]\big)/i\big\} \mid i > 0\big\}\Big)
\end{aligned}$$

We recover the boolean semantics for the standard operators when all atomic propositions have either value 0 (false) or value 1 (true). Note that in that case we might abusively consider that $v_i \in 2^{\mathcal{P}}$, recording the set of atomic propositions with value 1 at each position. The first five rules are standard and natural in a quantitative setting. The semantics of the $\mathbf{U}$- and $\mathbf{G}$-modalities are also natural: they extends the standard equivalences $\psi_1\,\mathbf{U}\,\psi_2 \equiv \psi_2 \vee (\psi_1 \wedge \mathbf{X}\,(\psi_1\,\mathbf{U}\,\psi_2))$, and $\mathbf{G}\,\psi \equiv \psi \wedge \mathbf{X}\,\mathbf{G}\,\psi$ to a quantitative setting. The last two modalities are specific

to our setting: formula $\psi_1 \,\widetilde{\mathbf{U}}\, \psi_2$ computes the average of formula ψ_1 for the i first steps, and then compares the value with that of ψ_2 at the $(i+1)$-st step. The best choice of i (if it exists) is then selected, and gives the value to the formula. Formula $\widetilde{\mathbf{G}}\, \psi$ computes the average of ψ in the long-run.

We come back to our two illustrative examples given in the introduction, to show how our logic can be used to express natural properties.

Example 3 (Jobshop scheduling). We come back to Example 1, assuming a set of n machines. Let load be an atomic proposition having value k/n at state s if k machines are in use in that state. Notice that we could equivalently use the local averaging operator $\oplus$ of [2] in order to have load defined as the average of the atomic propositions indicating which machines are in use. Then formula $\varphi_1 = \mathsf{load} \,\widetilde{\mathbf{U}}\, \mathsf{stop}$ evaluated on a schedule computes the average machine use along that schedule, if stop is a boolean atomic proposition which holds true when all jobs are finished. A schedule assigning value 1 to φ_1 could be seen as an optimal schedule, where no computation power is lost. A schedule assigning a small value to formula φ_1 is a schedule with a large loss of computation power.

On the other hand formula $\varphi_2 = \mathsf{load}\, \mathbf{U}\, \mathsf{stop}$ will evaluate to the smallest instantaneous machine use along a schedule. Note that syntactically it is a standard until, but it evaluates differently in our quantitative framework. ◁

Example 4 (Mobile phone server). The quality of the server of Example 2 can be expressed as the average over all requests of the frequency allocated in response. We can write such a property as $\varphi_3 = \widetilde{\mathbf{G}}\ (\neg \mathsf{req} \vee \mathsf{no\_grant}\, \mathbf{U}\, \mathsf{grant})$, where req and $\mathsf{no\_grant}$ are boolean atomic propositions with the obvious meaning, and grant is an atomic proposition with value in $[0,1]$ representing the quality of the allocated range of frequencies (the closer to 1, the better). Larger values of φ_3 then indicate better frequency allocation algorithms. ◁

We also evaluate formulas of avgLTL over Kripke structures. If v is a state of the Kripke structure $\mathcal{K}$ and $\varphi \in \mathsf{avgLTL}$, then we define: $[\![(\mathcal{K}, v), \varphi]\!] = \sup\{ [\![\pi, \varphi]\!] \mid \pi \text{ is an infinite run of } \mathcal{K} \text{ from } v \}$. We simply write $[\![\mathcal{K}, \varphi]\!]$ when $v = v_0$ is the initial vertex of $\mathcal{K}$. Notice that considering the supremum here corresponds to the *existential* semantics of boolean LTL, where the aim is to find a path satisfying the formula.

Example 5. We develop a small toy example to illustrate how simple formulas can be evaluated in the (qualitative) Kripke structure depicted on Fig. 1.

Consider the avgLTL formulas $a \,\widetilde{\mathbf{U}}\, b$ and $c \,\widetilde{\mathbf{U}}\, b$. For the first formula we have $[\![a \cdot b \cdot c^\omega, a \,\widetilde{\mathbf{U}}\, b]\!] = 1$ (the supremum being reached at the second position along the run), and therefore $[\![\mathcal{K}, a \,\widetilde{\mathbf{U}}\, b]\!] = 1$.

Now, for the formula $c \,\widetilde{\mathbf{U}}\, b$ and the same run as above, we have $[\![a \cdot b \cdot c^\omega, c \,\widetilde{\mathbf{U}}\, b]\!] = 0$: indeed, the right-hand-side formula b has value zero everywhere except at position 1, but the average of c on the previous positions is zero. For the run $a \cdot (b \cdot c)^\omega$, considering all positions (but position 1) where b is non-zero, we get $[\![a \cdot (b \cdot c)^\omega, c \,\widetilde{\mathbf{U}}\, b]\!] = \sup\{n/(2n+1) \mid n \in \mathbb{N}_{>0}\} = 1/2$. Note that the value $1/2$ is not reached by any prefix. Now consider the run $\pi'_k = a \cdot b \cdot c^k \cdot (b \cdot c)^\omega$, for some

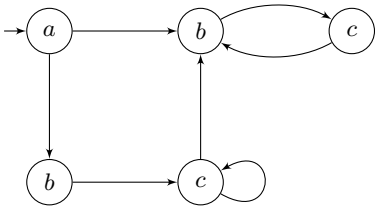

Fig. 1: A Kripke structure $\mathcal{K}$.

positive integer k. Then we have $[\![\pi'_k, c\,\widetilde{\mathbf{U}}\,b]\!] = \sup\{(k+n)/(k+2n+2) \mid n \in \mathbb{N}\}$. When $k \geq 3$, the supremum is $k/(k+2)$, which is reached for $n = 0$ (*i.e.*, at the second occurrence of b). From this we get that $[\![\mathcal{K}, c\,\widetilde{\mathbf{U}}\,b]\!] = 1$. However no run witnesses that value. ◁

3 The problems we consider

In this paper, we consider the following two problems:

Existence problem: given a Kripke structure $\mathcal{K}$, an avgLTL formula φ, and a threshold $\bowtie c$ (with $\bowtie \in \{<, \leq, =, \geq, >\}$ and $c \in [0,1] \cap \mathbb{Q}$), is there a path π in $\mathcal{K}$ such that $[\![\pi, \varphi]\!] \bowtie c$?

Value problem: given a Kripke structure $\mathcal{K}$, an avgLTL formula φ, and a threshold $\bowtie c$ (with $\bowtie \in \{<, \leq, =, \geq, >\}$ and $c \in [0,1] \cap \mathbb{Q}$), does $[\![\mathcal{K}, \varphi]\!] \bowtie c$?

Note that both problems are different since, as illustrated in Example 5, it can be the case that $[\![\mathcal{K}, \varphi]\!] = 1$ even though no path of $\mathcal{K}$ assigns value 1 to φ.

We also consider their approximation variants, defined as follows:

Approximate existence problem: given a Kripke structure $\mathcal{K}$, an avgLTL formula φ, a value $c \in [0,1] \cap \mathbb{Q}$ and $\varepsilon > 0$, is there a path π in $\mathcal{K}$ such that $c - \varepsilon < [\![\pi, \varphi]\!] < c + \varepsilon$?

Approximate value problem: given a Kripke structure $\mathcal{K}$, an avgLTL formula φ, a value $c \in [0,1] \cap \mathbb{Q}$ and $\varepsilon > 0$, does $c - \varepsilon < [\![\mathcal{K}, \varphi]\!] < c + \varepsilon$?

4 Model checking avgLTL is undecidable

In the sequel, we prove that avgLTL model-checking is *robustly* undecidable, in the sense that all the problems above are undecidable, for all threshold conditions considered. We would like to emphasize that different kinds of threshold give rise to different problems, and could have led to different decidability results. For instance, given a Kripke structure $\mathcal{K}$ and an avgLTL formula φ, $[\![\mathcal{K}, \varphi]\!] > 1/2$ iff there exists an infinite run π in $\mathcal{K}$ such that $[\![\pi, \varphi]\!] > 1/2$. On the other hand, $[\![\mathcal{K}, \varphi]\!] = 1/2$ iff there exists a sequence of infinite runs $(\pi^n)_{n \in \mathbb{N}}$ such that

$[\![\pi^n, \varphi]\!] \leq 1/2$ for every n, and $\lim_{n\to\infty}[\![\pi^n, \varphi]\!] = 1/2$. These remarks advocate for a clear and exhaustive study of the different problems with all the different thresholds.

Additionally, we believe that our original proof techniques (in particular the diagonal argument used to circumvent convergence phenomena for the model-checking of Kripke structures) are of particular interest and could be used in related settings. We discuss further these issues and related works in Section 8

We can now state the main results of the paper.

Theorem 6. *The existence problem is undecidable, for every threshold of the form* $\bowtie 1/2$*, with* $\bowtie \in \{<, \leq, =, \geq, >\}$.

Theorem 7. *The value problem is undecidable, for every threshold of the form* $\bowtie 1/2$*, with* $\bowtie \in \{<, \leq, =, \geq, >\}$.

We present these results as two distinct theorems, since proofs require very different techniques, even though a similar encoding is used.

Remark 8. Our proofs only involve qualitative Kripke structures. We present the results for $c = 1/2$, but our proofs could be adapted to handle any other rational value in $(0, 1)$ (e.g. by inserting fake actions in the encoding).

Now, if the approximate variants were decidable, then taking e.g. $c = 1$ and $\epsilon = 1/2$, we could decide e.g. whether a formula has value larger than $1/2$, contradicting the previous theorems. Hence:

Theorem 9. *The approximate existence and value problems are undecidable.*

The rest of the paper presents the main ideas of the proof. Due to lack of space, the full proofs could not be included here, but can be found in the research report [11] associated to this paper.

5 Proof of Theorem 6

We only give an explanation of the undecidability for the existence problem with threshold $\geq 1/2$ (the other types of thresholds require a twist in the construction, but no fundamental new argument).

The proof relies on an encoding of the halting problem for deterministic two-counter machines, which is well-known to be undecidable. A two-counter machine $\mathcal{M}$ is a finite-state machine, equiped with two kinds of transitions: *update*-transitions move from one state to another one while incrementing or decrementing one of the counters; *test*-transitions keep the counters unchanged, but may lead to two different states depending on the positiveness of one of the counters. The machine has a special state, called the *halting state*, from which no transitions is possible. We assume w.l.o.g. that all the other states have exactly one outgoing transition.

A configuration of $\mathcal{M}$ is given by the current state and the values of both counters. A run of $\mathcal{M}$ is a sequence of consecutive configurations *which might not properly update the counters*. It is said *valid* whenever the counters are properly

updated along the run. There is a unique maximal valid run in $\mathcal{M}$ from the initial configuration: it is either halting or infinite.

The idea of our reduction is to build a Kripke structure which generates the encodings of all (including invalid) runs of $\mathcal{M}$: it has to take care of the discrete structure of $\mathcal{M}$, but does not check that counters are properly updated along the run. Correct update of counter values will be checked using an avgLTL formula.

Description of the encoding. We first explain how we encode the runs of $\mathcal{M}$. We only give a simplified idea of the encoding. We write Q for the set of states of $\mathcal{M}$.

For $p \geq 2$, we write $\mathbb{B}_p$ for the set $\{0, 1, \ldots, p-1\}$. For $b \in \mathbb{B}_p$, we let $b^{+i} = b + i \mod p$. An element of $\mathbb{B}_p$ is abusively called a *bit*. These bits are used to distinguish between consecutive configurations. For the rest of this section, taking $p = 2$ would be sufficient, but the proof of Theorem 7 requires higher values for p. We encode configurations of $\mathcal{M}$ using the following finite set of atomic propositions: $\mathcal{P}_p = \big(Q \cup \{a_0, a_1\}\big) \times \mathbb{B}_p \cup \{\#\}$. The symbol $\#$ will be a marker for halting computations.

Exactly one atomic proposition from $\mathcal{P}_p$ will have value one at each position along the encoding (the other propositions having value zero). Given a bit b, a configuration $\gamma = (q, n_0, n_1)$ of $\mathcal{M}$ is encoded as the word $\mathsf{enc}_b(\gamma) = (q, b) \cdot (a_0, b)^{n_0} \cdot (a_1, b)^{n_1}$. For a halting configuration, we set $\mathsf{enc}_b(\gamma) = (q_{halt}, b)$.

The bit $b \in \mathbb{B}_p$ is incremented (modulo p) from one configuration to the next one. Let $\rho = \gamma_0 \cdot \gamma_1 \cdots$ be a (not necessary valid) run in $\mathcal{M}$. The p-encoding of ρ is then given by:

$$p\text{-}\mathsf{enc}(\rho) = \begin{cases} \mathsf{enc}_{b_0}(\gamma_0) \cdot \mathsf{enc}_{b_1}(\gamma_1) \cdot \mathsf{enc}_{b_2}(\gamma_2) \cdots & \text{if } \rho \text{ is infinite} \\ \mathsf{enc}_{b_0}(\gamma_0) \cdot \mathsf{enc}_{b_1}(\gamma_1) \cdots \mathsf{enc}_{b_{n-1}}(\gamma_{n-1}) \#^\omega & \text{if } \rho \text{ has length } n \end{cases}$$

with $b_j = j \mod p$ for every j. We write $\mathsf{enc}(\rho)$ if p is clear from the context.

We can easily construct a Kripke structure that generates the encodings of all possible (valid or invalid) runs of $\mathcal{M}$. For index p, we write $\mathcal{K}^p_{\mathcal{M}}$ for the corresponding Kripke structure. We now turn to the avgLTL formula, whose role is to check proper updates of the counters.

Definition of the formulas. We will define a formula $\mathtt{consec}^p_{\mathcal{M}}$, which will be used to check that each single consecution in the run properly updates the counters. Then we define formula

$$\mathtt{halt}^p_M = \mathbf{F}\, q_{halt} \wedge \mathbf{G}\, \mathtt{consec}^p_{\mathcal{M}}.$$

It is rather clear that if we can build such a formula $\mathtt{consec}^p_{\mathcal{M}}$, then the above formula will check that the unique maximal valid run of $\mathcal{M}$ is halting. Unfortunately, things are not that easy, and formula $\mathbf{G}\, \mathtt{consec}^p_{\mathcal{M}}$ will only be able to check the validity of *finite* runs

We now focus on defining $\mathtt{consec}^p_{\mathcal{M}}$, using the average-until modality. We only give an intuition (the full definition requires the complete encoding). Consider a

portion P of the p-encoding of a run ρ, which corresponds to a single-step of the computation of $\mathcal{M}$ where instruction q keeps both counter values unchanged:

$$\dots (q,b) \cdot (a_0,b)^{n_0} \cdot (a_1,b)^{n_1} \cdot (q',b^{+1}) \cdot (a_0,b^{+1})^{n'_0} \cdot (a_1,b^{+1})^{n'_1} (q'',b^{+2}) \dots$$

The formula has to enforce $n'_0 = n_0$ and $n'_1 = n_1$. This is the case if, and only if, for every $\alpha \in \{1+n_0+n_1, 1+n_0+n'_1, 1+n'_0+n_1, 1+n'_0+n'_1\}$,

$$\frac{\alpha}{1+n_0+n_1+1+n'_0+n'_1} = \frac{1}{2}.$$

The denominator is the length of the portion from (q,b) to the position just before (q'',b^{+2}), whereas the various values for α are the number of positions where some distinguished atomic proposition holds along this portion. For instance, $1+n'_0+n_1$ is the number of positions where formula $\psi = (q',b^{+1}) \vee (a_0,b^{+1}) \vee (a_1,b)$ holds along P. Computing the above quotient will be done using an $\widetilde{\mathbf{U}}$-formula: $[\![P, \psi\, \widetilde{\mathbf{U}}\, (q'',b^{+2})]\!]$ precisely equals $\frac{\alpha}{1+n_0+n_1+1+n'_0+n'_1}$

Using this idea, we are able to construct a formula $\mathtt{consec}^p_{\mathcal{M}}$ (as a conjunction of several $\widetilde{\mathbf{U}}$-formulas) whose value is $1/2$ along a single step of the computation if, and only if, this step is valid (that is, it correctly updates the counters).

Correctness of the reduction. Even though formula $\mathtt{consec}^p_{\mathcal{M}}$ properly checks the validity of a single step of the computation, it might be the case that $[\![p\text{-}\mathsf{enc}(\rho), \mathbf{G}\, \mathtt{consec}^p_{\mathcal{M}}]\!] = 1/2$, even though the whole computation is not valid: this is due to the definition of the semantics of $\widetilde{\mathbf{U}}$ as the supremum over all positions of the average; in particular, a single error in the computation can be hidden in the rest of the run. Consider for instance the counter machine in Fig. 2. The unique initial and maximal valid run of $\mathcal{M}$ halts. However, if the first transition increments counter a_0 twice, and all further transitions are properly taken, then the resulting (invalid) run will assign value $1/2$ to formula $\mathbf{G}\, \mathtt{consec}^p_{\mathcal{M}}$.

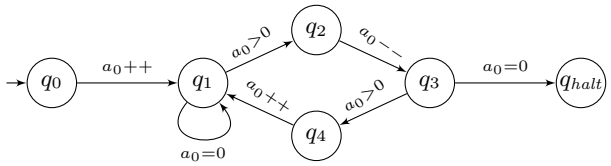

Fig. 2: There is an invalid infinite run ρ such that $[\![p\text{-}\mathsf{enc}(\rho), \mathbf{G}\, \mathtt{consec}^p_{\mathcal{M}}]\!] = 1/2$

Still, we are able to prove the following classification of runs of $\mathcal{M}$ in terms of the value of $\mathtt{halt}^p_{\mathcal{M}}$. It proves the fact that formula $\mathtt{consec}^p_{\mathcal{M}}$ properly checks the validity of a single step of the computation, provided the $\widetilde{\mathbf{U}}$-formulas cannot

benefit from the supremum semantics. This is the case when the run in the Kripke structure ends with $\#^\omega$, which corresponds to finite runs of $\mathcal{M}$.

Classification 1. *Fix $p \geq 2$. Let ρ be a maximal run in $\mathcal{M}$.*

- *if ρ is infinite, then $[\![p\text{-}\mathsf{enc}(\rho), \mathtt{halt}^p_{\mathcal{M}}]\!] = 0$;*
- *if ρ is finite and valid, then $[\![p\text{-}\mathsf{enc}(\rho), \mathtt{halt}^p_{\mathcal{M}}]\!] = 1/2$;*
- *if ρ is finite and invalid, then $[\![p\text{-}\mathsf{enc}(\rho), \mathtt{halt}^p_{\mathcal{M}}]\!] < 1/2$.*

Corollary 10. *Fix $p \geq 2$. The following five statements are equivalent:*

1. *$\mathcal{M}$ halts;*
2. *the unique initial and maximal valid run $\rho_{\mathcal{M}}$ of $\mathcal{M}$ is such that $[\![p\text{-}\mathsf{enc}(\rho_{\mathcal{M}}), \mathtt{halt}^p_{\mathcal{M}}]\!] = 1/2$;*
3. *there exists an initial maximal run ρ in $\mathcal{M}$ such that $[\![p\text{-}\mathsf{enc}(\rho), \mathtt{halt}^p_{\mathcal{M}}]\!] = 1/2$;*
4. *there exists an initial maximal path π in $\mathcal{K}^p_{\mathcal{M}}$ such that $[\![\pi, \mathtt{halt}^p_{\mathcal{M}}]\!] = 1/2$;*
5. *there exists an initial maximal path π in $\mathcal{K}^p_{\mathcal{M}}$ such that $[\![\pi, \mathtt{halt}^p_{\mathcal{M}}]\!] \geq 1/2$.*

This corollary allows to conclude the undecidability proof of Theorem 6.

6 Proof of Theorem 7

As already mentioned, whether $[\![\mathcal{K}, \varphi]\!] > 1/2$ (and dually, $[\![\mathcal{K}, \varphi]\!] \leq 1/2$) is equivalent to the existence of a path whose value is strictly more than $1/2$, which we just proved undecidable.

We now turn to the more interesting cases of $=$ (the result for $\geq$ and $<$ directly follows, as we explain at the end of this proof). We were not able to write a direct proof as previously, because we could not distinguish between counter machines that have a halting computation (whose encoding has value $1/2$ against formula $\mathtt{halt}^p_{\mathcal{M}}$ above) and counter machines that have sequences of computations whose encodings have values *converging* to $1/2$.

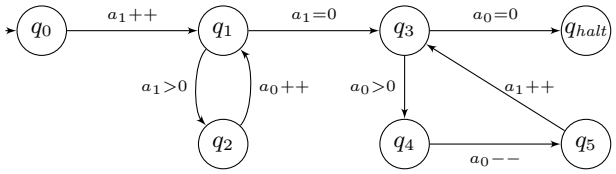

Fig. 3: A non-halting two-counter machine for which $[\![\mathcal{K}^p_{\mathcal{M}}, \mathtt{halt}^p_{\mathcal{M}}]\!] = 1/2$

Example 11. We consider the deterministic two-counter machine $\mathcal{M}$ of Fig. 3, having q_0 as its initial state. The unique initial and maximal valid run of $\mathcal{M}$ is infinite (it loops in $q_1 \leftrightarrows q_2$). A single error can make the transition from q_1 to q_3 available, from which valid consecutions lead to q_{halt}. The *weight* of this error can be arbitrarily small, as it can occur with an arbitrarily large value of a_0. It is not difficult to check that $[\![\mathcal{K}^p_{\mathcal{M}}, \mathtt{halt}^p_{\mathcal{M}}]\!] = 1/2$ (for any $p \geq 2$). ◁

Analysis of a non-halting two-counter machine. We consider a deterministic accept/reject two-counter machine $\mathcal{M}$: such machines have two halting states, now named q_{accept} and q_{reject}. Their computations may still be infinite. We consider formula $\mathtt{consec}^p_{\mathcal{M}}$ again, and define $\mathtt{accept}^p_{\mathcal{M}} \;=\; \mathbf{F}\, q_{accept} \wedge \mathbf{G}\, \mathtt{consec}^p_{\mathcal{M}}$.

We first analyse the impact of the first error along a finite run ρ of $\mathcal{M}$ onto the value of $\mathbf{G}\, \mathtt{consec}^p_{\mathcal{M}}$, and we are able to show the following surprising but crucial lemma (remember the example of Fig. 2) whose proof requires long technical developments. The condition imposed on p is a sufficient condition for "detecting" invalid consecutions along finite runs. The computation leading to this value is explained in the long version [11] of this work.

Lemma 12. *Fix $p \geq 927$. Let ρ be a finite invalid run of $\mathcal{M}$. Assume $\rho_i\rho_{i+1}$ is the first invalid consecution along ρ, and write step_i for the portion of p-$\mathsf{enc}(\rho)$ corresponding to that consecution. Pick $n \geq 30$ such that $[\![\mathsf{step}_i, \mathtt{consec}^p_{\mathcal{M}}]\!] \leq 1/2 - 1/n$. Then $[\![p\text{-}\mathsf{enc}(\rho), \mathbf{G}\, \mathtt{consec}^p_{\mathcal{M}}]\!] \leq 1/2 - 1/n$.*

This allows to prove the next fundamental result:

Lemma 13. *Fix $p \geq 927$, and assume that $[\![\mathcal{K}^p_{\mathcal{M}}, \mathtt{accept}^p_{\mathcal{M}}]\!] = 1/2$, but that no run ρ of $\mathcal{M}$ has $[\![p\text{-}\mathsf{enc}(\rho), \mathtt{accept}^p_{\mathcal{M}}]\!] = 1/2$. Then the unique initial and maximal valid run of $\mathcal{M}$ is infinite.*

We sketch the proof of this lemma, since it contains an interesting argument.

Sketch of proof. Let ρ be the unique initial and maximal valid run of $\mathcal{K}^p_{\mathcal{M}}$. Let $(\rho^n)_{n\in\mathbb{N}}$ be a sequence of initial and maximal runs such that $[\![p\text{-}\mathsf{enc}(\rho^n), \mathtt{accept}^p_{\mathcal{M}}]\!] > 1/2 - 1/n$ (such a sequence exists by hypothesis, but runs ρ^n might be invalid). Pick $n \geq 30$, and let $\rho^n_{i_n}\rho^n_{i_n+1}$ be the first invalid consecution of ρ^n. Write step_{i_n} for the portion of p-$\mathsf{enc}(\rho^n)$ corresponding to that consecution. Applying Lemma 12, we get that $[\![\mathsf{step}_{i_n}, \mathtt{consec}^p_{\mathcal{M}}]\!] > 1/2 - 1/n$. Since $\rho^n_{i_n}\rho^n_{i_n+1}$ is an invalid consecution, we also have that $[\![\mathsf{step}_{i_n}, \mathtt{consec}^p_{\mathcal{M}}]\!] < 1/2$. It follows that $1/(|\mathsf{step}_{i_n}| - 1) < 1/n$, which implies that $|\mathsf{step}_{i_n}| > n$. Now, the prefix of ρ of size i_n coincides with that of ρ^n, since $\rho^n_{i_n}\rho^n_{i_n+1}$ is the first invalid consecution. We conclude that ρ contains configurations of arbitrarily large size, so that the sum of the two counters is unbounded along ρ. Hence ρ is infinite. □

A diagonal argument. Any deterministic Turing machine can be simulated by a deterministic two-counter machine [20]. In particular, given a deterministic Turing machine B, we can build a deterministic two-counter machine $\mathcal{M}(B)$ whose computation mimics the run of B on input B. Then $\mathcal{M}(B)$ accepts (resp. rejects, does not halt) if, and only if, B accepts (resp. rejects, does not halt on) input B.

We fix $p \geq 927$, and define the following function $\mathcal{H}$, which takes as input a deterministic Turing machine B:

$$\mathcal{H}(B) = \begin{cases} \mathit{accept} & \text{if } [\![\mathcal{K}^p_{\mathcal{M}(B)}, \mathtt{accept}^p_{\mathcal{M}(B)}]\!] = 1/2 \\ \mathit{reject} & \text{otherwise} \end{cases}$$

Proposition 14. *The function $\mathcal{H}$ is not computable.*

Proof. Towards a contradiction, assume $\mathcal{H}$ is computable. Let $\mathcal{T}_{\mathcal{H}}$ be a deterministic Turing machine that computes $\mathcal{H}$. Notice in particular that $\mathcal{T}_{\mathcal{H}}$ halts on all its inputs; we assume that it ends in its state $q^{\mathcal{T}}_{accept}$ when $\mathcal{H}$ accepts the input, and in $q^{\mathcal{T}}_{reject}$ when $\mathcal{H}$ returns *reject*.

We now define the following deterministic Turing machine $\mathcal{C}$, which takes as input a deterministic Turing machine B:

$\mathcal{C}(B)$: Simulate $\mathcal{T}_{\mathcal{H}}$ on B;
If the simulation ends in $q^{\mathcal{T}}_{accept}$ then goto $q^{\mathcal{C}}_{reject}$, otherwise goto $q^{\mathcal{C}}_{accept}$.

The Turing machine $\mathcal{C}$ terminates on all its inputs, since so does $\mathcal{T}_{\mathcal{H}}$; also, $\mathcal{C}$ is deterministic, and we can therefore run $\mathcal{C}$ on input $\mathcal{C}$ itself.

Assume $\mathcal{C}$ accepts input $\mathcal{C}$. This means that $\mathcal{H}(\mathcal{C})$ rejects, which means that $[\![\mathcal{K}^p_{\mathcal{M}(\mathcal{C})}, \mathtt{accept}^p_{\mathcal{M}(\mathcal{C})}]\!] < 1/2$. This means that $\mathcal{M}(\mathcal{C})$ does not accept (by a straightforward extension of Corollary 10 to accept/reject two-counter machines), and therefore $\mathcal{C}$ does not accept $\mathcal{C}$, contradicting our hypothesis.

Hence $\mathcal{C}$ rejects input $\mathcal{C}$, so that $[\![\mathcal{K}^p_{\mathcal{M}(\mathcal{C})}, \mathtt{accept}^p_{\mathcal{M}(\mathcal{C})}]\!] = 1/2$. However, since $\mathcal{C}$ does not accept $\mathcal{C}$, the unique initial and maximal valid run of $\mathcal{M}_{\mathcal{C}}$ is either infinite or rejecting. Applying Lemma 13 to $\mathcal{M}_{\mathcal{C}}$, we get that it is actually infinite. This means that the simulation of $\mathcal{T}_{\mathcal{H}}$ on input $\mathcal{C}$ does not terminate. This contradicts the fact that $\mathcal{T}_{\mathcal{H}}$ terminates on every input. Therefore $\mathcal{H}$ is not computable. □

Theorem 7 is a direct consequence of this lemma for threshold $= 1/2$. Now, using Classification 1, for a deterministic two-counter machine $\mathcal{M}$, it holds that $[\![\mathcal{K}^p_{\mathcal{M}}, \mathtt{accept}^p_{\mathcal{M}}]\!] = 1/2$ iff $[\![\mathcal{K}^p_{\mathcal{M}}, \mathtt{accept}^p_{\mathcal{M}}]\!] \geq 1/2$. Hence the above proof applies to threshold $\geq 1/2$ as well. The case of $< 1/2$ is the dual of $\geq 1/2$: if $\mathcal{K}$ is a Kripke structure and φ an $\mathtt{avgLTL}$ formula, $[\![\mathcal{K}, \varphi]\!] < 1/2$ iff it is not the case that $[\![\mathcal{K}, \varphi]\!] \geq 1/2$, which proves the result for threshold $< 1/2$ as well.

7 Proof of Theorem 9

We now discuss the undecidability of the approximate variants. It relies on the same encoding as that for the existence problem and threshold $> 1/2$. For that threshold, we have a classification of the runs similar to Classification 1, for formula $\mathtt{halt}^{p,>}_{\mathcal{M}}$: for every maximal run ρ in $\mathcal{M}$:

- if ρ is infinite, then $[\![p\text{-}\mathtt{enc}(\rho), \mathtt{halt}^{p,>}_{\mathcal{M}}]\!] = 0$;
- if ρ is finite and valid, then $1/2 < [\![p\text{-}\mathtt{enc}(\rho), \mathtt{halt}^{p,>}_{\mathcal{M}}]\!] < 3/4$;
- if ρ is finite and invalid, then $[\![p\text{-}\mathtt{enc}(\rho), \mathtt{halt}^{p,>}_{\mathcal{M}}]\!] \leq 1/2$.

We deduce that $\mathcal{M}$ halts iff there exists an initial and maximal valid run π in $\mathcal{K}^p_{\mathcal{M}}$ with $1/2 < [\![\pi, \mathtt{halt}^{p,>}_{\mathcal{M}}]\!] < 3/4$. This shows undecidability of the approximate existence problem.

Now, we also have in this case the equivalence with $1/2 < [\![\mathcal{K}^p_{\mathcal{M}}, \mathtt{halt}^{p,>}_{\mathcal{M}}]\!] < 7/8$ (not $3/4$ since there might be some convergence phenomenon towards value $3/4$), which also shows the undecidability of the approximate value problem.

8 Discussion on related works

In this section, we would like to illustrate the difficulty of lifting temporal-logic model checking from the qualitative to the quantitative setting. As we saw in this paper, several new convergence phenomena do appear, which make the problem complex, but also make the proofs difficult. Our undecidability proofs in this paper involve difficult techniques to properly handle the convergence phenomena that appear in the semantics of the logic. This difficulty has led to several wrong arguments in the related litterature, as we now illustrate.

We first discuss the logic frequency-LTL of [9]. This logic has a boolean semantics, but extends LTL with a frequency-**U** modality, which gives it a quantitative taste: formula $\phi_1 \,\mathbf{U}^{\,c} \phi_2$ holds true along a path π whenever there is a position n along π at which ϕ_2 holds, and the number of previous positions where ϕ_1 holds is larger than or equal to $c \cdot n$ (hence c is a lower bound on the frequency of ϕ_1 on the prefix before ϕ_2 holds). Note that it need not be the case that the position n is the first position where ϕ_2 holds: for instance *abbcaaac* satisfies formula $a \,\mathbf{U}^{\,\frac{1}{2}} c$, but at the first occurrence of c, the frequency of a on the prefix is $1/3$, which is less than $1/2$; the correct witness position for $a \,\mathbf{U}^{\,\frac{1}{2}} c$ is the second occurrence of c, where the frequency of a becomes $4/7$. In frequency-LTL, there is no convergence phenomena, but some possibly unbounded search for some witnessing position. Then evaluating $b \,\mathbf{U}^{\,\frac{1}{2}} c$ on a path π is not equivalent to comparing formula $b \,\widetilde{\mathbf{U}}\, c$ to value $1/2$ on path π: first because of convergence phenomena (as illustrated in Example 5), and because in our quantitative setting, the value of the right-hand-side subformula could be less than $1/2$.

It is shown in [9] that the validity problem for frequency-LTL is undecidable, and our reduction shares similarities with that reduction (but we believe that our reduction is simpler, and the result stronger, since it uses no nested $\widetilde{\mathbf{U}}$). However the undecidability proof (as written in [9]) has a flaw: it relies on the claim that "[t]he formula $b \,\mathbf{U}^{\,\frac{1}{2}} l \wedge \hat{b} \,\mathbf{U}^{\,\frac{1}{2}} l$ enforces the pattern $b^m \hat{b}^m l$..." (the order of b's and $\hat{b}$'s is enforced by another LTL formula). This claim is wrong in general, since the $\mathbf{U}^{\,\frac{1}{2}}$-formulas might not refer to the same occurrences of l. The proof can be patched[1], and one way is to restrict to paths that end with $\#^\omega$ for some marker $\#$; in that way a backward argument can be used to check proper encoding of the execution of the two-counter machine (this is actually what we do in the proof of Theorem 6).

We now discuss the logic discounted-LTL of [2]. This logic gives a quantitative semantics to an extension of LTL, with a new discounted-**U** modality: given a discount function η, the value of formula $\phi_1 \,\mathbf{U}_\eta\, \phi_2$ along a path π is the supremum over all positions n along π of the minimum of the value of ϕ_2 at that position, discounted by $\eta(n)$, and of the values of ϕ_1 at every earlier position i, discounted by $\eta(i)$. Satisfiability is proven decidable; it is shown undecidable when adding the local average operator $\oplus$, which computes the average of two formulas.

[1] Personal communication with the authors.

Those results are then extended to the model-checking problem[2]. While the first result extends properly for threshold $< c$ (since the infimum over all paths is smaller than c if, and only if, there is a path that evaluates to a value smaller than c; hence convergence phenomena are avoided), it is not valid for Theorem 3 of [2] (which is stated with threshold $> c$). Also, undecidability of the model-checking problem with local-average operator (Theorem 6 of [2]) is not correct since it does not take convergence phenomena into account. A corrected version of the proof is available in [3]; while it does not use a diagonal argument as we do, the undecidability proof is not a direct encoding of a two-counter machine, but requires computing the value of two different formulas in order to encode the halting problem.

This all shows that extending temporal logics to a quantitative setting is more than a simple exercise: complex convergence phenomena come into play, which have to be understood and handled with extreme care. We hope that our work will provide new insights about these problems, and believe that our techniques can be useful for handling them.

9 Conclusion and future work

We believe that our logic avgLTL is a very relevant logic in many applications. It provides a way of *measuring* some properties, such as the average load of the CPUs in scheduling applications. We proved that the value of a formula can not be computed—and not even approximated. For the interesting case however (deciding whether $[\![\mathcal{K}, \phi]\!] \geq \eta$), we had to resort to an original diagonal argument to get around convergence phenomena.

Our negative results certainly echo back the fact, mentioned e.g. in [17], that averaging does not fit well with classical automata-based approaches for temporal logics. Indeed, averaging gives rise to new values that are not present in the original automaton. Discounting LTL instead of averaging has the same difficulty, but this is compensated by the fact that when discounting, the value of a formula can be approximated by considering only a finite prefix of a run [2].

We are currently investigating two directions in order to get decidability results: first by adding discounting on the right-hand-side formula (while keeping averaging on the left-hand-side); second, by considering the *qualitative* cases of avgLTL, namely whether a formula has value 0 or 1. One difficulty here is that in some cases the witnesses are a family of paths, instead of just a single path.

References

1. S. Almagor, U. Boker, and O. Kupferman. Formalizing and reasoning about quality. In ICALP'13, LNCS 7966, p. 15–27. Springer, 2013.

[2] Note that the value of a Kripke structure is defined there as the infimum over all paths, and not as the supremum as we do here; this corresponds to the duality between existential *vs* universal quantification in standard LTL

2. S. Almagor, U. Boker, and O. Kupferman. Discounting in LTL. In TACAS'14, LNCS, Lecture Notes in Computer Science. Springer, 2014. To appear.
3. S. Almagor, U. Boker, and O. Kupferman. Discounting in LTL. Research Report 1406.4249, arXiv, 2014. 21 pages.
4. R. Alur and D. L. Dill. A theory of timed automata. *Theor. Computer Science*, 126(2):183–235, 1994.
5. R. Alur, S. La Torre, and G. J. Pappas. Optimal paths in weighted timed automata. In HSCC'01, LNCS 2034, p. 49–62. Springer, 2001.
6. A. Aziz, K. Sanwal, V. Singhal, and R. K. Brayton. Model-checking continuous-time Markov chains. *ACM Transactions on Computational Logic*, 1(1):162–170, 2000.
7. G. Behrmann, A. Fehnker, T. Hune, K. G. Larsen, P. Pettersson, J. Romijn, and F. Vaandrager. Minimum-cost reachability for priced timed automata. In HSCC'01, LNCS 2034, p. 147–161. Springer, 2001.
8. U. Boker, K. Chatterjee, T. A. Henzinger, and O. Kupferman. Temporal specifications with accumulative values. In LICS'11, p. 43–52. IEEE Comp. Soc. Press, 2011.
9. B. Bollig, N. Decker, and M. Leucker. Frequency linear-time temporal logic. In TASE'12, p. 85–92. IEEE Comp. Soc. Press, 2012.
10. P. Bouyer, P. Gardy, and N. Markey. Quantitative verification of weighted kripke structures. Research Report LSV-14-08, Laboratoire Spécification et Vérification, ENS Cachan, France, 2014. 26 pages.
11. P. Bouyer, N. Markey, and R. M. Matteplackel. Quantitative verification of weighted kripke structures. Research Report LSV-14-02, Laboratoire Spécification et Vérification, ENS Cachan, France, 2014. 35 pages.
12. P. Černý, T. A. Henzinger, and A. Radhakrishna. Simulation distances. *Theor. Computer Science*, 413(1):21–35, 2012.
13. K. Chatterjee, L. Doyen, and T. A. Henzinger. Quantitative languages. *ACM Transactions on Computational Logic*, 11(4), 2010.
14. L. de Alfaro, M. Faella, T. A. Henzinger, R. Majumdar, and M. Stoelinga. Model checking discounted temporal properties. *Theor. Computer Science*, 345(1):139–170, 2005.
15. L. Doyen. *Games and Automata: From Boolean to Quantitative Verification.* Mémoire d'habilitation, ENS Cachan, France, 2012.
16. M. Droste, W. Kuich, and W. Vogler, editors. *Handbook of Weighted Automata.* Springer, 2009.
17. M. Faella, A. Legay, and M. Stoelinga. Model checking quantitative linear time logic. In QAPL'08, ENTCS 220, p. 61–77. Elsevier Science, 2008.
18. T. A. Henzinger. Quantitative reactive models. In MODELS'12, LNCS 7590, p. 1–2. Springer, 2012.
19. T. A. Henzinger and J. Otop. From model checking to model measuring. In CONCUR'13, LNCS 8052, p. 273–287. Springer, 2013.
20. M. L. Minsky. *Computation: Finite and Infinite Machines.* Prentice Hall, Inc., 1967.
21. M.-P. Schützenberger. On the definition of a family of automata. *Inf. & Cont.*, 4(2-3):245–270, 1961.
22. T. Tomita, S. Hiura, S. Hagihara, and N. Yonezaki. A temporal logic with mean-payoff constraints. In ICFEM'12, LNCS 7635, p. 249–265. Springer, 2012.