Monitor-Based Statistical Model Checking for Weighted Metric Temporal Logic *

Peter Bulychev[1], Alexandre David[1], Kim G. Larsen[1], Axel Legay[2], Guangyuan Li[3], Danny Bøgsted Poulsen[1], Amelie Stainer[4]

[1] Computer Science, Aalborg University, Denmark
[2] INRIA/IRISA, Rennes Cedex, France
[3] State Key Laboratory of Computer Science, Institute of Software, Chinese Academy of Sciences, Beijing, P.R. of China
[4] University of Rennes 1, Rennes, France

Abstract. We present a novel approach and implementation for analysing weighted timed automata (WTA) with respect to the weighted metric temporal logic ($\mathrm{WMTL}_{\leq}$). Based on a stochastic semantics of WTAs, we apply statistical model checking (SMC) to estimate and test probabilities of satisfaction with desired levels of confidence. Our approach consists in generation of deterministic monitors for formulas in $\mathrm{WMTL}_{\leq}$, allowing for efficient SMC by run-time evaluation of a given formula. By necessity, the deterministic observers are in general approximate (over- or under-approximations), but are most often exact and experimentally tight. The technique is implemented in the new tool Casaal that we seamlessly connect to Uppaal-smc in a tool chain. We demonstrate the applicability of our technique and the efficiency of our implementation through a number of case-studies.

1 Introduction

Model checking (MC) [14] is a widely used approach to guarantee correctness of a system by checking that its model satisfies a given property. A typical model checking algorithm explores a state space of a model and tries to prove or disprove that the property holds on the model.

Despite a large and growing number of successful applications in industrial case studies, the MC approach still suffers from the so-called state explosion problem. This problem manifests itself in the form of unmanageably large state spaces of models with large number of components (i.e. number of variables, parallel components, etc). The situation is even worse when a system under analysis is hybrid (i.e. it possesses both continuous and discrete behaviors), because a state space of such models may lack finite representation [2]. Another challenge for MC is to analyze stochastic systems, i.e. systems with probabilistic assumptions for their behavior.

* The paper is supported by the Danish National Research Foundation, the National Natural Science Foundation of China (Grant No.61061130541) for the Danish-Chinese Center for Cyber Physical Systems and VKR Center of Excellence MT-LAB.

One of the ways to avoid these complexity and undecidability issues is to use statistical model checking (SMC) approach [19]. The main idea of the latter is to observe a number of simulations of a model and then use results from statistics (e.g. sequential analysis) to get an overall estimate of a system behavior.

In the present paper we consider a problem of computing the probability that a random run of a given weighted timed automaton (WTA) satisfies a given weighted metric temporal logic formula (WMTL$_{\leq}$). Solving this problem is of great practical interest since WTA are as expressive as general linear hybrid automata [2], a formalism which has proved to be very useful for modeling real-world hybrid and real-time systems. Moreover, WMTL$_{\leq}$ [7] is not only a weighted extension of the well established LTL but can also be seen as an extension of MTL [15] to hybrid systems. However, the model checking problem for WMTL$_{\leq}$ is known to be undecidable [7], and in our paper we propose an approximate approach that computes a confidence interval for the probability. In most of the cases this confidence interval can be made arbitrary small.

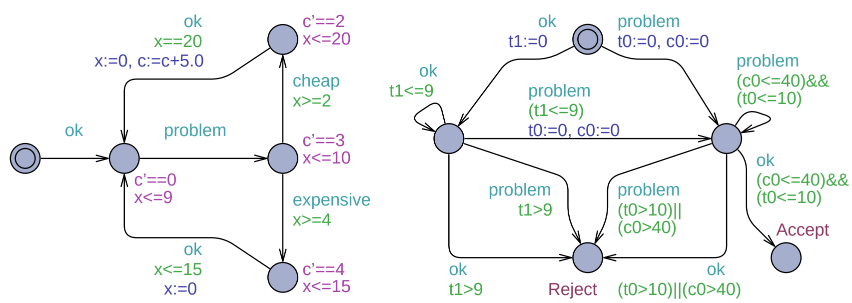


Fig. 1: A model (left) and deterministic monitor (right) for the repair problem

As an example consider a never-ending process of repairing problems [7], whose Weighted Timed Automata model are depicted at Fig. 1 (left). The repair of a problem has a certain cost, captured in the model by the clock `c`[1]. As soon as a problem occurs (modeled by the transition labeled by action `problem`) the value of `c` grows with rate 3, until actual `cheap` (rate 2) or `expensive` (rate 4) repair is taking place. Clock `x` grows with rate 1 (it's default behavior unless other rate is specified). Being a Weighted Timed Automata, this model is equipped with a natural stochastic semantics [10] with a uniform choice on possible discrete transitions and uniformly selected delays in locations.

Now consider that we want to express the property that a path goes from `ok` back to itself in time less than 10 time units and cost less than 40. This can be formalized by the following WMTL$_{\leq}$ formula:

$$\texttt{ok}\,\mathsf{U}^{\tau}_{\leq 9}(\texttt{problem} \wedge (\neg\texttt{ok}\ \vee \mathsf{U}^{\tau}_{\leq 10}\texttt{ok}) \wedge (\neg\texttt{ok}\mathsf{U}^{c}_{\leq 40}\texttt{ok}))$$

[1] we will (mis)use the term "clock" from timed automata, though in the setting of WTAs the clocks are really general real-valued variables

Here, the MITL$_\leq$-formula $\varphi_1 \mathsf{U}^c_{\leq d} \varphi_2$ is satisfied by a run if φ_1 is satisfied on the run until φ_2 is satisfied, and this will happen before the value of the clock c increases with more than d starting from the beginning of the run (τ is a special clock that always grows with rate 1).

In order to estimate the probability that a random run of a model satisfies a given property, our approach will first construct deterministic monitoring weighted timed automata for this property. In fact, it is not always possible to construct an *exact* deterministic observer for a property, thus our tool can result in deterministic under- and over-approximations. For our example, the tool constructed the exact deterministic monitor presented in Fig. 1 (right). Here rates of a monitoring automaton are defined by the rates of the automaton being monitored, i.e. the rate of `c0` is equal to the rate of `c`.

The constructed monitoring WTA permits the SMC engine of UPPAAL to use run-time evaluation of the property in order to efficiently estimate the probability that runs of the models satisfy the given property. In our example the UPPAAL-SMC returns the 95% confidence interval $[0.215, 0.225]$. If none of the under- and over-approximation monitors are exact, then we use both of them to compute the confidence interval.

Our contribution is twofold. First, we are the first to extend statistical model checking to the WMTL$_\leq$ logic. The closest logic that has been studied so far is the strictly less expressive MITL$_\leq$, that does not allow to use energy clocks in the U operator. Second, our monitor-based approach works on-the-fly and can terminate a simulation as soon as it may conclude that a formula will be satisfied (or violated) by the simulation. Other statistical model checking algorithms that deal with linear-time properties (cf. [1,18,19,20]) require a posterior (and expensive) check after a complete simulation of a fixed duration has been generated.

2 Weighted Timed Automata & Metric Temporal Logic

In this section we describe weighted timed automata (WTA) and weighted metric temporal logic (WMTL$_\leq$) as our modeling and specification formalisms. A notion of monitoring weighted timed automata (MWTA) is used to define automatically constructed (deterministic) observers for WMTL$_\leq$ properties.

2.1 Weighted Timed Automata

Let C be a set of clocks. A clock bound over C has the form $c \sim n$ where $c \in C$, $\sim \in \{<, \leq, \geq, >\}$ and $n \in \mathbb{Z}_{\geq 0}$. We denote the set of all possible clock bounds over C by $\mathcal{B}(C)$. A valuation over C is a function $v : C \to \mathbb{R}_{\geq 0}$, and a rate vector is a function $r : C \to \mathbb{Q}$. We let $\mathcal{V}(C)$ ($\mathcal{R}(C)$, respectively) to be all clock valuations (rates) over C.

Definition 1. *A* Weighted Timed Automaton[2] *(WTA) over alphabet $\mathcal{A}$ is a tuple $(L, \ell_0, C_i, C_o, E, W, I, R)$ where:*

- *L is a finite set of locations,*
- *$\ell_0 \in L$ is the initial location,*
- *C_i and C_o are finite set of real-valued variables called* internal *clocks and* observable *clocks, respectively,*
- *$E \subseteq L \times \mathcal{A} \times 2^{\mathcal{B}(C_i \cup C_o)} \times 2^{C_i} \times L$ is a finite set of edges,*
- *$W : E \to \mathcal{R}(C_i \cup C_o)$ assigns weights to edges, weights of observable clocks should be non-negative (i.e. $W(e)(c) \geq 0$ for any $e \in E$ and $c \in C_o$),*
- *$I : L \to 2^{\mathcal{B}(C_i \cup C_o)}$ assigns an invariant to each location,*
- *$R : L \to \mathcal{R}(C_i \cup C_o)$ assigns rates to the clocks in each location, rates of observable clocks should be non-negative.*

If $\delta \in \mathbb{R}_{\geq 0}$, then we define $v + \delta$ to be equal to the valuation v' such, that for all $c \in C$ we have $v'(c) = v(c) + \delta$. If r is a rate vector, then $v + r \cdot \delta$ is the valuation v' such that for all clocks c in C, $v'(c) = v(c) + r(c) \cdot \delta$. The valuation that assigns zero to all clocks is denoted by $\vec{0}$. Given $Y \subseteq C$, $v[Y = 0]$ is the valuation equal to $\vec{0}$ over Y and equal to v over $C \setminus Y$. We say, that a valuation v satisfies a clock bound $b = c \sim n$ (denoted $v \vDash b$), iff $v(c) \sim n$. A valuation satisfies a set of clock bounds if it satisfies all of them or this set is empty. A state (l, v) of a WTA consists of a location $l \in L$ and a valuation $v \in \mathcal{V}(C_i \cup C_o)$. In particular, the initial state of the WTA is $(\ell_0, \vec{0})$. From a state a WTA can either delay for some time δ or it can perform a discrete action a, the rules are given below:

- $(\ell, v) \xrightarrow{\delta} (\ell, v')$ if $v' = v + R(\ell) \cdot \delta$ and $v' \vDash I(\ell)$.
- $(\ell, v) \xrightarrow{a} (\ell', v')$ if $v \vDash g$ and there exists an edge $e \in E$ such that $e = (\ell, g, a, Y, r, \ell')$, $v' = v[Y = 0] + W(e) \cdot 1$ and $v' \vDash I(\ell')$.

An (infinite) weighted word over actions $\mathcal{A}$ and clocks C is a sequence $w = (a_0, v_0)(a_1, v_1)\ldots$ of pairs of actions $a_i \in \mathcal{A}$ and valuations $v_i \in \mathcal{V}(C)$. For $i \geq 0$, we denote by w^i the weighted word $w^i = (a_i, v_i)(a_{i+1}, v_{i+1})\ldots$.

A WTA $A = (L, \ell_0, C_i, C_o, E, W, I, R)$ over $\mathcal{A}$ generates a weighted word $w = (a_0, v_0)(a_1, v_1)\ldots$ over actions $\mathcal{A}$ and *observable* clocks C_o, iff $v_0 = \vec{0}$ and there exists a sequence of transitions

$$(\ell_0, v'_0) \xrightarrow{\delta_0} (\ell_0, v''_0) \xrightarrow{a_0} (\ell_1, v'_1) \xrightarrow{\delta_1} \ldots \xrightarrow{a_n} (\ell_{n+1}, v'_{n+1}) \ldots,$$

and for any i the valuation v_i is a projection of v'_i to C_o, i.e. $v_i(c)$ is equal to $v'_i(c)$ for any observable clock $c \in C_o$.

Note, that since *observable* clocks are never reset and grow only with positive rates, the values of observable clocks can not decrease in a word generated by a

[2] In the classical notion of priced timed automata [6,4] cost-variables (e.g. clocks where the rate may differ from 1) may not be referenced in guards, invariants or in resets, thus making e.g. optimal reachability decidable. This is in contrast to our notion of WTA, which is as expressive as linear hybrid systems [8].

WTA. In fact, we restrict ourselves to WTAs that generate cost-divergent words (i.e. for any observable clock c and constant $k \in \mathbb{R}_{\geq 0}$ there is v_i such, that $v_i(c) > k$). If we consider that the WTA in Fig. 1(left) has only one observable clock `c`, then this WTA can generate a weighted word $(\texttt{ok}, \{\texttt{c} \mapsto 2.0\})$, $(\texttt{problem}, \{\texttt{c} \mapsto 3.1\})$, $(\texttt{cheap}, \{\texttt{c} \mapsto 4.2\}), \ldots$.

We let $\mathcal{L}(A)$ denote the set of all weighted words generated by an WTA A and refer to it as the language of A.

A network of Weighted Timed Automata is a parallel composition of several WTA that have disjoint set of clocks and same set of actions $\mathcal{A}$. The automata are synchronized regarding discrete transitions such that if one automata performs a transition $\xrightarrow{a}$ all other also must perform an $\xrightarrow{a}$ transition. The notion of language recognized by WTA is naturally extended to the networks of Weighted Timed Automata.

In [10] we proposed a stochastic semantics for WTA, i.e. a probability measure over the set of accepted weighted words $\mathcal{L}(A)$. The non-determinism regarding discrete transitions for a single WTA is resolved using a uniform probabilistic choice among the possible transitions. Non-determinism regarding delays from a state (ℓ, v) of a single WTA is resolved using a density function $\mu_{(\ell,v)}$ over delays in $\mathbb{R}_{\geq 0}$ being either a uniform or an exponential distribution depending on whether the invariant of ℓ is empty or not.
The stochastic semantics for networks of WTA is then given in terms of repeated races between the component WTAs of the network: before a discrete transition each WTA chooses a delay according to its delay density function; then the WTA with a smallest delay wins the race and chooses probabilistically the action that the network must perform.

2.2 Monitoring Weighted Timed Automata

A monitoring weighted timed automaton (MWTA) A_M is a special kind of WTA used to define allowed behavior of a given WTA A (or a network of WTAs): a weighted word generated by A is fed as input to A_M for acceptance. For this, the actions of A and A_M coincide and there is a correspondence between the monitoring clocks of A_M and the observable clocks A ensuring that corresponding clocks grow with the same rate.

Definition 2. *A* Monitoring Weighted Timed Automaton *(MWTA) over the clocks C and the actions $\mathcal{A}$ is a tuple $(L, \ell_0, \ell_a, C_M, E, m)$ where:*

- *L is a finite set of locations,*
- *$\ell_0 \in L$ is the initial location,*
- *$\ell_a \in L$ is the accepting location,*
- *C_M is a finite set of* local *clocks,*
- *$E \subseteq (L \setminus \{l_a\}) \times \mathcal{A} \times 2^{\mathcal{B}(C_M)} \times 2^{C_M} \times L$ is a finite set of edges,*
- *$m : C_M \rightarrow C$ gives the correspondence of local clocks and C.*

An MWTA is called deterministic if for any location $l \in L \setminus \{\ell_a\}$, action $a \in \mathcal{A}$ and valuation $v \in \mathcal{V}(C_M)$ there exist not more than one edge $(l, a, g, Y, l') \in E$ such that $v \models g$.

An MWTA $A_M = (L, \ell_0, \ell_a, C_M, E, m)$ over clocks C and actions $\mathcal{A}$ accepts a weighted word $(a_0, v_0)(a_1, v_1)\ldots$ over the same C and $\mathcal{A}$, iff there exists a finite sequence $(l_0, v'_0), (l_1, v'_1), \ldots, (l_n, v'_n)$ of states of A_M such, that:

- $v'_0(c) = v_0(m(c))$ for any clock $c \in C_M$,
- for any i there exists an edge $(l_i, a_i, g_i, Y_i, l_{i+1}) \in E$ such, that:
  - $v'_i \models g_i$ and
  - for every clock $c \in C_M$, if $c \in Y_i$ then $v'_{i+1}(c) = 0$, and otherwise $v'_{i+1}(c) = v'_i(c) + (v_{i+1}(m(c)) - v_i(m(c)))$,
- $l_n = l_a$ is the accepting location of A.

Thus, after reading an element of an input weighted word, a *local* clock c the MWTA either reset, or it grows with the same rate as the corresponding clock $m(c)$ in the input word.

2.3 Weighted Metric Temporal logic WMTL$_\leq$

Definition 3. *[7] A WMTL$_\leq$ formula φ over atomic propositions P and clocks C is defined by the grammar*

$$\varphi ::= p \,|\, \neg\varphi \,|\, \varphi_1 \wedge \varphi_2 \,|\, O\varphi \,|\, \varphi_1 \mathsf{U}^c_{\leq d} \varphi_2$$

where $p \in P$, $d \in \mathbb{N}$, and $c \in C$.

Let *false* be an abbreviation for $(p \wedge \neg p)$, and *true* be an abbreviation for $\neg false$. The other commonly used operators in WMTL$_\leq$ can be defined by the following abbreviations: $(\varphi_1 \vee \varphi_2) = \neg(\neg\varphi_1 \wedge \neg\varphi_2)$, $(\varphi_1 \rightarrow \varphi_2) = (\neg\varphi_1) \vee \varphi_2$, $\Diamond^c_{\leq d}\varphi = true\, \mathsf{U}^c_{\leq d}\varphi$, $\Box^c_{\leq d}\varphi = \neg\Diamond^c_{\leq d}\neg\varphi$, and $\varphi_1 \mathsf{R}^c_{\leq d}\varphi_2 = \neg(\neg\varphi_1 \mathsf{U}^c_{\leq d}\neg\varphi_2)$, where R is the "release" operator. We also assume, that there always exists a special clock $\tau \in C$ (that grows with a rate 1 in an automaton being monitored).

Assuming that P are atomic propositions over actions $\mathcal{A}$, WMTL$_\leq$ formulas are interpreted over weighted words. For a given weighted word $w = (a_0, v_0)(a_1, v_1)(a_2, v_2)\ldots$ over $\mathcal{A}$ and C and WMTL$_\leq$ formula φ over P and C, the satisfaction relation $w^i \models \varphi$ is defined inductively:

1. $w^i \models p$ iff $a_i \models p$
2. $w^i \models \neg\varphi$ iff $w^i \not\models \varphi$
3. $w^i \models O\varphi$ iff $w^{i+1} \models \varphi$
4. $w^i \models \varphi_1 \wedge \varphi_2$ iff $w^i \models \varphi_1$ and $w^i \models \varphi_2$
5. $w^i \models \varphi_1 \mathsf{U}^c_{\leq d}\varphi_2$ iff there exists j such that $j \geq i$, $w^j \models \varphi_2$, $v_j(c) - v_i(c) \leq d$, and $w^k \models \varphi_1$ for all k with $i \leq k < j$.

We say, that a weighted word w satisfies φ, iff $w^0 \models \varphi$, and denote by $\mathcal{L}(\varphi)$ the set of all weighted words that are satisfied by φ. φ_1 and φ_2 are equivalent if they are satisfied by the same weighted words, in which case we write $\varphi_1 \equiv \varphi_2$.

Given the stochastic semantics of a WTA A, and semantics of WMTL$_\leq$ formula φ, we can define $Pr[A \models \varphi]$ to be the probability that a random run of A satisfies φ. This probability is well-defined because $\mathcal{L}(A) \cap \mathcal{L}(\varphi)$ is a countable union and intersection of measurable sets and thus it is measurable itself.

3 From Formulas to Monitors

In this section we present a novel procedure for translating $\mathrm{WMTL}_{\leq}$ formulas into equivalent MWTA monitors, providing an essential and efficient component of our tool-chain. However, to enable monitor-based, statistical model checking it is essentially that the generated MWTA is deterministic. Unfortunately, this might not always be possible as there are $\mathrm{WMTL}_{\leq}$ formulas for which no equivalent deterministic MWTA exist[3]. As a remedy, we describe how basic syntactic transformations prior to translation allow us to obtain deterministic over- and under-approximating MWTAs for any given formula φ. In Section 5, we shall see that these approximations are tight and often exact.

3.1 Closures & Extended Formulas

In this section, we assume that φ is $\mathrm{WMTL}_{\leq}$ formula over propositions P and (observable) clocks C and has been transformed into negative normal form (NNF), i.e. an equivalent formula in which negations are applied to the atomic propositions only. We use $\mathrm{Sub}(\varphi)$ to denote all the sub-formulas of φ.

In order to further expand φ into a disjunctive normal form, we introduce for each $\phi_1 \mathsf{U}^c_{\leq d} \phi_2 \in \mathrm{Sub}(\varphi)$ and each $\phi_1 \mathsf{R}^c_{\leq d} \phi_2 \in \mathrm{Sub}(\varphi)$, one local clock x and two timing constraints $x \leq d$ and $x > d$ to express some timing information related to $\phi_1 \mathsf{U}^c_{\leq d} \phi_2$ and $\phi_1 \mathsf{R}^c_{\leq d} \phi_2$. Also, we introduce auxiliary formulas $\phi_1 \mathsf{U}^c_{\leq d-x} \phi_2$ and $\phi_1 \mathsf{R}^c_{\leq d-x} \phi_2$ to express some requirements that should be satisfied in the future when we try to guarantee $\phi_1 \mathsf{U}^c_{\leq d} \phi_2 \in \mathrm{Sub}(\varphi)$ or $\phi_1 \mathsf{R}^c_{\leq d} \phi_2 \in \mathrm{Sub}(\varphi)$ is true in the current state.

We define $X_\varphi = \{x_{\phi_1 \mathsf{U}^c_{\leq d} \phi_2} | \phi_1 \mathsf{U}^c_{\leq d} \phi_2 \in \mathrm{Sub}(\varphi)\} \cup \{x_{\phi_1 \mathsf{R}^c_{\leq d} \phi_2} | \phi_1 \mathsf{R}^c_{\leq d} \phi_2 \in \mathrm{Sub}(\varphi)\}$ to be the set of all local clocks for φ, where $x_{\phi_1 \mathsf{U}^c_{\leq d} \phi_2}$ is the clock assigned to $\phi_1 \mathsf{U}^c_{\leq d} \phi_2$ and $x_{\phi_1 \mathsf{R}^c_{\leq d} \phi_2}$ is the local clock assigned to $\phi_1 \mathsf{R}^c_{\leq d} \phi_2$. We call $x_{\phi_1 \mathsf{U}^c_{\leq d} \phi_2}$ a local clock of $\mathsf{U}_{\leq}$-type, and $x_{\phi_1 \mathsf{R}^c_{\leq d} \phi_2}$ a local clock of $\mathsf{R}_{\leq}$-type. The mapping m from local clocks X_φ to observable clocks C is defined by $m(x_{\phi_1 \mathsf{U}^c_{\leq d} \phi_2}) = c$ and $m(x_{\phi_1 \mathsf{R}^c_{\leq d} \phi_2}) = c$. The closure of φ, write as $\mathrm{CL}(\varphi)$, is now defined by the following rules:

1. $\mathit{true} \in \mathrm{CL}(\varphi)$, $\mathrm{Sub}(\varphi) \subseteq \mathrm{CL}(\varphi)$
2. If $\phi_1 \mathsf{U}^c_{\leq d} \phi_2 \in \mathrm{Sub}(\varphi)$ and x is the local clock assigned to $\phi_1 \mathsf{U}^c_{\leq d} \phi_2$, then $x \leq d$, $x > d$, $\phi_1 \mathsf{U}^c_{\leq d-x} \phi_2 \in \mathrm{CL}(\varphi)$
3. If $\phi_1 \mathsf{R}^c_{\leq d} \phi_2 \in \mathrm{Sub}(\varphi)$ and x is the local clock assigned to $\phi_1 \mathsf{R}^c_{\leq d} \phi_2$, then $x \leq d$, $x > d$, $\phi_1 \mathsf{R}^c_{\leq d-x} \phi_2 \in \mathrm{CL}(\varphi)$
4. If $\Phi_1, \Phi_2 \in \mathrm{CL}(\varphi)$, then $\Phi_1 \wedge \Phi_2$, $\Phi_1 \vee \Phi_2 \in \mathrm{CL}(\varphi)$

Obviously, $\mathrm{CL}(\varphi)$ has only finitely many different non-equivalent formulas.

For a local clock x, we use $rst(x)$ to represent that x will be reset at current step and $unch(x)$ to represent that x will not be reset at current step. The set of extended formulas for φ, write as $\mathrm{Ext}(\varphi)$, is now defined by the following rules:

[3] For instance, $\Diamond^{\tau}_{\leq 1}(p \wedge \Box^{\tau}_{\leq 1}(\neg r) \wedge \Diamond^{\tau}_{\leq 1}(q))$ is an example of a formula not equivalent to any deterministic MWTA.

1. If $\Phi \in \mathrm{CL}(\varphi)$, then Φ, $\mathrm{O}\Phi \in \mathrm{Ext}(\varphi)$
2. If $x \in X_\varphi$ is a local clock of $\mathsf{U}_{\leq}$-type, then $\mathrm{unch}(x) \in \mathrm{Ext}(\varphi)$
3. If $x \in X_\varphi$ is a local clock of $\mathsf{R}_{\leq}$-type, then $\mathrm{rst}(x) \in \mathrm{Ext}(\varphi)$
4. If Φ_1, $\Phi_2 \in \mathrm{Ext}(\varphi)$, then $\Phi_1 \wedge \Phi_2$, $\Phi_1 \vee \Phi_2 \in \mathrm{Ext}(\varphi)$

Extended formulas can be interpreted using extended weighted words. An *extended weighted word* $\omega = (a_0, v_0, \nu_0)(a_1, v_1, \nu_1)(a_2, v_2, \nu_2)\ldots$ is a sequence where $w = (a_0, v_0)(a_1, v_1)(a_2, v_2)\ldots$ is a weighted word over 2^P and C, and for every $i \in \mathbb{N}$, ν_i is a clock valuation over X_φ such that for all $x \in X_\varphi$, either $\nu_{i+1}(x) = v_{i+1}(m(x)) - v_i(m(x))$ or $\nu_{i+1}(x) = \nu_i(x) + v_{i+1}(m(x)) - v_i(m(x))$.

The semantics for extended formulas is naturally induced by the semantics of $\mathrm{WMTL}_{\leq}$ formulas:

Definition 4. *Let $\omega = (a_0, v_0, \nu_0)(a_1, v_1, \nu_1)(a_2, v_2, \nu_2)\ldots$ be an extended weighted word and $\Phi \in Ext(\varphi)$. The satisfaction relation $\omega^i \models_e \Phi$ is inductively defined as follows:*

1. $\omega^i \models_e x \sim d$ *iff* $\nu_i(x) \sim d$
2. $\omega^i \models_e rst(x)$ *iff* $\nu_{i+1}(x) = v_{i+1}(m(x)) - v_i(m(x))$
3. $\omega^i \models_e unch(x)$ *iff* $\nu_{i+1}(x) = \nu_i(x) + v_{i+1}(m(x)) - v_i(m(x))$
4. $\omega^i \models_e \phi$ *iff* $w^i \models \phi$, *if* $\phi \in Sub(\varphi)$
5. $\omega^i \models_e \varphi_1 \mathsf{U}^c_{\leq d-x} \varphi_2$ *iff there exists j such that* $j \geq i$, $w^j \models \varphi_2$, $v_j(c) - v_i(c) \leq d - \nu_i(x)$, *and* $w^k \models \varphi_1$ *for all k with* $i \leq k < j$
6. $\omega^i \models_e \varphi_1 \mathsf{R}^c_{\leq d-x} \varphi_2$ *iff for all* $j \geq i$ *such that* $v_j(c) - v_i(c) \leq d - \nu_i(x)$, *either* $w^j \models \varphi_2$ *or there exists k with* $i \leq k < j$ *and* $w^k \models \varphi_1$
7. $\omega^i \models_e \Phi_1 \wedge \Phi_2$ *iff* $\omega^i \models_e \Phi_1$ *and* $\omega^i \models_e \Phi_2$
8. $\omega^i \models_e \Phi_1 \vee \Phi_2$ *iff* $\omega^i \models_e \Phi_1$ *or* $\omega^i \models_e \Phi_2$
9. $\omega^i \models_e O\Phi$ *iff* $\omega^{i+1} \models_e \Phi$

ω^i *is a* model *of Φ if $\omega^i \models_e \Phi$ and two extended $WMTL_{\leq}$-formulas are said* equivalent *if they have exactly the same models.*

3.2 Constructing Non-deterministic Monitors

As in the construction of Büchi automata from LTL formulas, we will break a formula into a disjunction of several conjunctions [9]. Each of the disjuncts corresponds to a transition of a resulting observer automaton and specifies the requirements to be satisfied in the current and in the next states. In the rest of this section, we use $rst(\{x_1, x_2, \ldots, x_n\})$ and $unch(\{y_1, y_2, \ldots, y_n\})$ to denote the formula of $rst(x_1) \wedge rst(x_2) \wedge \ldots \wedge rst(x_n)$ and the formula of $unch(y_1) \wedge unch(y_2) \wedge \ldots \wedge unch(y_n)$ respectively. A *basic conjunction* is an extended formula of the form:

$$\alpha \wedge g \wedge rst(X) \wedge unch(Y) \wedge O(\Psi),$$

where α is a conjunction of literals (a literal is a proposition or its negation), g is a conjunction of clock constraints, X is a set of local clocks with $\mathsf{R}_{\leq}$-type, Y is a set of local clocks with $\mathsf{U}_{\leq}$-type, and Ψ is a formula in $\mathrm{CL}(\varphi)$. $\alpha \wedge g \wedge rst(X) \wedge unch(Y)$

specifies the requirements to be satisfied in the current state and Ψ specifies the requirements in the next-state. The next Lemma 1 and main Theorem 1 provides the construction of a monitor from a formula.

Lemma 1. *Each formula in $CL(\varphi)$ can be transformed into a disjunction of several basic conjunctions by using the following rules and Boolean equivalences.*

1. $f\,\mathsf{U}^{c}_{\leq d}\,g = g \vee (f \wedge O((x \leq d) \wedge (f\,\mathsf{U}^{c}_{\leq d-x} g)))$, *where x is the clock assigned to* $f\,\mathsf{U}^{c}_{\leq d}\,g$
2. $f\,\mathsf{U}^{c}_{\leq d-x}\,g = g \vee (f \wedge unch(x) \wedge O((x \leq d) \wedge (f\,\mathsf{U}^{c}_{\leq d-x} g)))$
3. $f\,\mathsf{R}^{c}_{\leq d}\,g = g \wedge (f \vee (rst(x) \wedge O(((x \leq d) \wedge (f\,\mathsf{R}^{c}_{\leq d-x} g)) \vee (x > d))))$, *where x is the clock assigned to* $f\,\mathsf{R}^{c}_{\leq d}\,g$
4. $f\,\mathsf{R}^{c}_{\leq d-x}\,g = g \wedge (f \vee O(((x \leq d) \wedge (f\,\mathsf{R}^{c}_{\leq d-x} g)) \vee (x > d)))$
5. $(Of) \wedge (Og) = O(f \wedge g)$
6. $(Of) \vee (Og) = O(f \vee g)$

Theorem 1. *Let φ be a $WMTL_{\leq}$-formula over the propositions P and the clocks C and is in NNF. Let the MWTA $A_{\varphi} = (L, \ell_0, \ell_a, C_M, E, m)$ over the clocks C and the actions $\mathcal{A} = 2^P$ be defined as follows:*

- $L = \{\{\phi\} \mid \phi \in CL(\varphi)\ \}$ *is a finite set of locations, and* $\ell_0 = \{\varphi\}$ *is the initial location;*
- $\ell_a = \{true\}$ *is the accepting location;*
- $C_M = X_{\varphi}$ *is the set of all local clocks for* φ*;*
- $(\{f_1\}, a, g, \lambda, \{f_2\}) \in E$ *iff* $\alpha \wedge g \wedge rst(X) \wedge unch(Y) \wedge O(f_2)$ *is a basic conjunction of* f_1 *and that a satisfies* α*, and for each* $x \in X_{\varphi}$ *of* $\mathsf{U}_{\leq}$*-type,* $x \in \lambda$ *iff* $x \notin Y$*, and for each* $x \in X_{\varphi}$ *of* $\mathsf{R}_{\leq}$*-type,* $x \in \lambda$ *iff* $x \in X$*;*
- m *is defined by* $m(x_{\phi_1 \mathsf{U}^{c}_{\leq d} \phi_2}) = c$ *and* $m(x_{\phi_1 \mathsf{R}^{c}_{\leq d} \phi_2}) = c$.

Then $\mathcal{L}(\varphi) = \mathcal{L}(A_{\varphi})$.

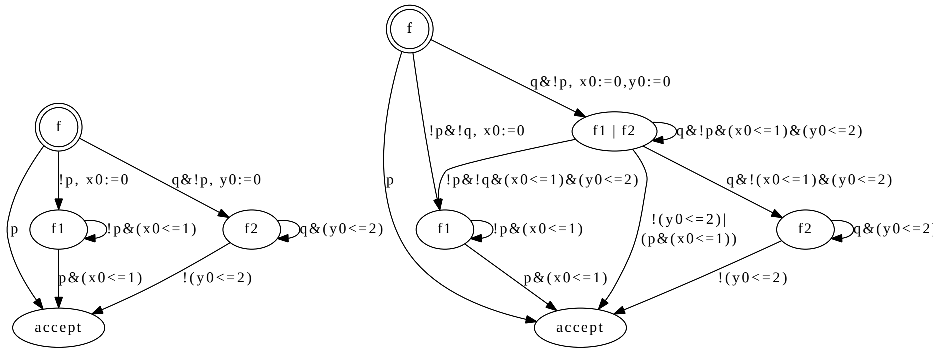
Example 1. Fig.2a is a MWTA obtained with our approach for $f = (\Diamond^{x}_{\leq 1} p) \vee (\Box^{y}_{\leq 2} q) = (true\,\mathsf{U}^{x}_{\leq 1} p) \vee (false\,\mathsf{R}^{y}_{\leq 2} q)$.

3.3 Constructing Deterministic Monitors

The construction of the section 3.2 might produce non-deterministic automata. In fact, as stated earlier, there exist $\text{WMTL}_{\leq}$ formulas for which no equivalent deterministic MWTA. To get deterministic MWTA for $\text{WMTL}_{\leq}$-formulas, we further translate formulas in disjunctive into the following *deterministic* form by repeated use of the logical equivalence $p \Leftrightarrow (p \wedge q) \vee (p \wedge \neg q)$.

$$F = \bigvee_{i=1}^{n} \left(\alpha_i \wedge g_i \wedge \bigvee_{k=1}^{m_i} (rst(X_{ik}) \wedge unch(Y_{ik}) \wedge O(\Psi_{ik}))\right)$$

where for all $i \in \{1, \ldots, n\}$: m_i is a positive integer, $X_{ik} \subseteq X_{\varphi}$ is a set of local clocks of type $\mathsf{R}_{\leq}$ and $Y_{ik} \subseteq X_{\varphi}$ is a set of local clocks of type $\mathsf{U}_{\leq}$, and for all $i \neq j$: $\alpha_i \wedge g_i \wedge \alpha_j \wedge g_j$ is *false*.



(a) Non-deterministic monitor (b) Deterministic under-approximation monitor

Fig. 2: Monitoring WTA for $f \equiv (\Diamond^{x}_{\leq 1} p) \vee (\Box^{y}_{\leq 2} q)$, with $f1 \equiv (x_0 \leq 1) \wedge (\mathit{true}\ \mathsf{U}^{x}_{\leq 1-x_0} p)$ and $f2 \equiv ((y_0 \leq 2) \wedge (\mathit{false}\ \mathsf{R}^{y}_{\leq 2-y_0} q)) \vee (y_0 > 2)$.

Using the facts that O distributes over $\vee$, and $\mathit{rst}(X)$ and $\mathit{unch}(X)$ are monotonic in X, the following formulas are obviously strengthened (F^u) respectively weakened (F^o) versions of F:

$$F^u = \bigvee_{i=1}^{n} \big(\alpha_i \wedge g_i \wedge \mathit{rst}(\bigcup_{k=1}^{m_i} X_{ik}) \wedge \mathit{unch}(\bigcup_{k=1}^{m_i} Y_{ik}) \wedge O(\bigvee_{k=1}^{m_i} \Psi_{ik})\big)$$
$$F^o = \bigvee_{i=1}^{n} \big(\alpha_i \wedge g_i \wedge \mathit{rst}(\bigcap_{k=1}^{m_i} X_{ik}) \wedge \mathit{unch}(\bigcap_{k=1}^{m_i} Y_{ik}) \wedge O(\bigvee_{k=1}^{m_i} \Psi_{ik})\big)$$

Interestingly, by simply applying the construction of Theorem 1 to F^u (F^o) we immediately obtain a deterministic under-approximating (over-approximating) MWTA A^u_φ (A^o_φ) for φ. Moreover, if during the construction of A^u_φ we see that F^u is always semantically equivalent to F, then A^u_φ is an exact determinization of φ, i.e. $\mathcal{L}(A^u_\varphi) = \mathcal{L}(\varphi)$ (the same is true for overapproximation).

Example 2. (continued) Fig.2b is the under-approximation deterministic MWTA for $f = (\Diamond^{x}_{\leq 1} p) \vee (\Box^{y}_{\leq 2} q)$.

4 The Tool Chain

Figure 3 provides an architectural view of our tool chain. The tool chain takes as input an MITL$_{\leq}$ formula φ, a WTA model M, as well as statistical parameters ϵ, α for controlling precision and confidence level. As a result a confidence interval for the probability $Pr[M \models \varphi]$ with the desired precision and confidence level is returned.

Casaal The tool chain includes the new tool component Casaal for generating monitors. The tool is implemented in C++ and is build on top of the Spot[4]

[4] http://spot.lip6.fr/wiki/

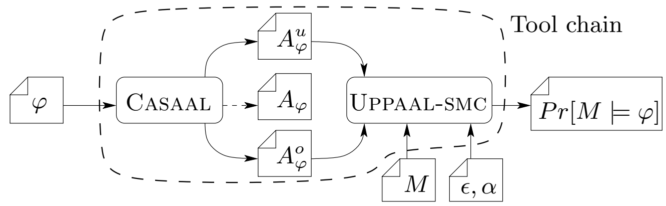


Fig. 3: Tool chain architecture

open-source library for LTL to Büchi automata translation. We also use Buddy[5] BDD package to handle operations over Boolean formulas. Given a $\text{WMTL}_{\leq}$ formula φ, CASAAL may construct an exact monitoring WTA A_φ, as well as two – possibly approximating – monitoring WTAs, A^u_φ and A^o_φ. The tool also reports if one of these approximations is exact (i.e. recognizes exactly the language of φ). Table 1 demonstrates some experimental results for CASAAL. The formulas were also used in [13] and for comparison we list their results as well.

formula	automaton	states	trans	time(s)
$p\mathsf{U}^\tau_{\leq 1}(q\mathsf{U}^\tau_{\leq 1}(\mathsf{R}^\tau_{\leq 1}s))$	nondet	5	14	0.02
	under	9	58	0.02
	over	9	56	0.04
	Geilen	14	30	
$(p \rightarrow \diamond^\tau_{\leq 5}q)\mathsf{U}^\tau_{\leq 100}\square^\tau_{\leq 5}\neg p$	nondet	7	19	0.01
	under	9	32	0.01
	over	9	32	0.01
	Geilen	21	64	
$(((p\mathsf{U}^\tau_{\leq 4}q)\mathsf{U}^\tau_{\leq 3}r)\mathsf{U}^\tau_{\leq 2}s)\mathsf{U}^\tau_{\leq 1}t)$	nondet	17	121	0.02
	under	17	121	0.03
	over	17	121	0.03
	Geilen	60	271	

Table 1: Experimental results for $\text{WMTL}_{\leq}$ formulas.

Uppaal-smc [10,11] is a tool that allows to estimate and test $Pr[M \models \phi]$, i.e. the probability that a random run of a given WTA model M satisfies ϕ, where ϕ is a $\text{WMTL}_{\leq}$ formula restricted to the form $\diamond^c_{\leq d}\psi$ and ψ is a state predicate. Estimation is performed by generating a number of random simulations of M, where each simulation stops when either it reaches a state when ψ is satisfied, or $c \leq d$ is violated.

Combining Casaal and Uppaal-smc Let us describe how we use UPPAAL-SMC together with the CASAAL tool to estimate the probability that a random run of a WTA model M satisfies a general $\text{WMTL}_{\leq}$ property ϕ, i.e. $Pr[M \models \phi]$.

Let us first assume, that one of two deterministic approximations for φ returned by CASAAL is exact. This means, that we have MWTA $A^{det}_\varphi = (L, \ell_0, \ell_a, C_M, E, m)$ such that $\mathcal{L}(A^{det}_\varphi) = \mathcal{L}(\varphi)$. First, we turn A^{det}_φ into input-enabled automaton by introducing a *rejecting* location l_r and adding complementary

[5] http://sourceforge.net/projects/buddy/develop

transitions to l_r from all other locations. Then we augment MWTA A_φ^{det} with a clock $c^\dagger$ that will grow with rate 1 in rejecting location ℓ_r, and with rate 0 in all other locations. Additionally, for every clock $c \in C_M$ we duplicate all rates and transition weights from the corresponding clock $m(c)$ to make sure, that the clocks of A_φ^{det} grow with the same rate as the corresponding clocks of the automaton M being monitored. Forming a parallel composition of M and A_φ^{det}, we may now use UPPAAL-SMC to estimate the probability $p = Pr[M||A_\varphi^{det} \models \Diamond_{\leq 1}^{c^\dagger}(\ell_a)]$. This can be done because of the following theorem:

Theorem 2. *If M produces cost-divergent runs only, then each simulation of $M||A_\varphi^{det}$ will end up in accepting or rejecting location of A_φ^{det} after finite number of steps.*

If none of the two MWTAs A_φ^o and A_φ^u are exact determinization of A_φ (i.e. $\mathcal{L}(A_\varphi^u) \subsetneq \mathcal{L}(\varphi) \subsetneq \mathcal{L}(A_\varphi^o)$), then we use both of them to compute upper (using A_φ^o) and lower (using A_φ^u) bounds for $Pr[M \models \varphi]$. Indeed, if n_1 (n_2, correspondingly) out of m random simulations of $M||A_\varphi^u$ ($M||A_\varphi^o$, correspondingly) ended in accepting location l_a^u (l_a^o, correspondingly), then with significance level of α we can accept a hypothesis H_1 (H_2, correspondingly) that $Pr[M \models \varphi] \geq n_1/m - \varepsilon$ ($Pr[M \models \varphi] \leq n_2/m + \varepsilon$). By combining hypothesis H_1 and H_2 we can obtain a confidence interval $[n_1/m - \varepsilon, n_2/m + \varepsilon]$ for $Pr[M \models \varphi]$ with significance level of $1 - (1-\alpha)^2 = 2\alpha - \alpha^2$.

5 Case Studies

We performed several case studies to demonstrate the applicability of our tool chain. In the first case study we analyze the performance of CASAAL on a set of randomly generated WMTL$_\leq$ formulas. In the second case study we use a model of a robot moving on a two-dimensional grid, this model was first analyzed in [5] using the manually constructed monitoring timed automaton.

5.1 Automatically Generated Formulas

In the first case study we analyze the performance of CASAAL on a set of randomly generated WMTL$_\leq$ formulas. We generated 1000 formulas with 2, 3 and 4 actions, and created deterministic over and approximations for these formulas. Each of the formulas have 15 connectives (release, until, conjunction or disjunction) and four clocks.

	# exact				Avg. time (s)		Avg. size		Stochastic difference	
Actions	under	over	none	one	under	over	under	over	no exact	one exact
2	831	542	169	289	0.24	1.01	6.35	6.35	0.27	0.15
3	706	370	294	336	1.42	2.75	12.29	12.29	0.05	0.03
4	586	233	414	353	8.66	13.05	22.97	22.97	0.01	0.02

Table 2: Results for the random generated formula test.

For the formulas where only one or none of the approximations was exact (i.e. $\mathcal{L}(A^u_\varphi) \neq \mathcal{L}(A_\varphi)$ or $\mathcal{L}(A^o_\varphi) \neq \mathcal{L}(A_\varphi)$), we measured the "stochastic difference" between approximations by generating a number of random weighted words and estimating the probability that the over approximation accepts a random word, when the under approximation does not.

Table 2 reports the amount of formulas for which the under or over approximation was exact and the amount of formulas where none of them was exact. It also contains the average time spent for generating the monitors and the average number of locations, and the stochastic difference.

5.2 Robot Control

We consider the case of a robot moving on a two-dimensional grid that was explored in e.g. [5]. Each field of the grid is either `normal`, on `fire`, cold as `ice` or it is a wall which that cannot be passed. Also, there is a `goal` field that the robot must reach. The robot is moving in a random fashion i.e. it stays in a field for some time, and then randomly moves to one of the neighboring fields (if it is not a wall). Fig. 5 shows a robot controller implementing this along with the grid we use.

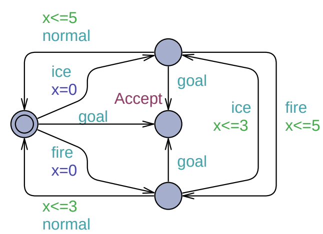


Fig. 4: Observer automaton used in [5]

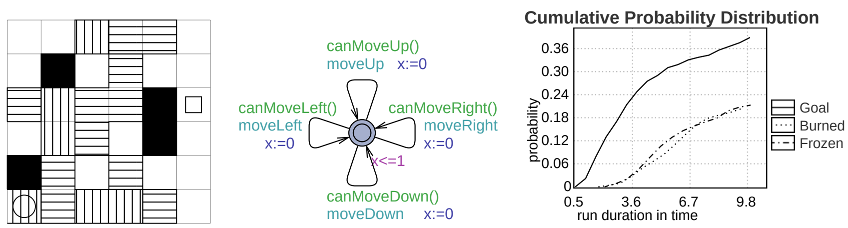
We are interested in the probability that the robot reaches its goal location without staying on consecutive fire fields for more than one time units and on consecutive ice fields for more than two time units.

In [5] the authors solved this problem by manually constructing a monitoring automaton to operate in parallel with the model of the robot. The automaton they used is depicted in Figure 4. Using WMTL$_\leq$ we can express the same requirement more easily as $\varphi \equiv (\varphi_1 \wedge \varphi_2) \mathsf{U}^\tau_{\leq 10} \texttt{goal}$, where:

$$\varphi_1 \equiv \texttt{ice} \implies \Diamond^\tau_{\leq 2}(\texttt{fire} \vee \texttt{normal} \vee \texttt{goal})$$
$$\varphi_2 \equiv \texttt{fire} \implies \Diamond^\tau_{\leq 1}(\texttt{ice} \vee \texttt{normal} \vee \texttt{goal})$$

Casaal produces an MWTA (6 locations, 55 edges) that is an exact under-approximation for φ. Based on this MWTA, our tool chain estimates the probability that the random behavior of the robot satisfies φ to lie in the interval $[0.373, 0.383]$ with a confidence of 95%. Fig. 5c shows how we can visualize and compare the different distributions using the plot composer of Uppaal-smc.

Energy We extend the model by limiting the energy of the robot that will stop moving when it runs out of energy. Furthermore, it can regain energy while staying on fire fields and use additional energy while staying on ice fields. Let c be the clock accumulating the amount of consumed energy. Now, we can express the property $\varphi \equiv (\varphi_1 \wedge \varphi_2 \wedge \neg \texttt{noEnergy}) \mathsf{U}^c_{\leq 10} \texttt{goal}$ that the robot should not use more than 5 units of energy while obeying the requirements from before. The tool chain estimates the probability that the robot satisfies this requirement to lie in $[0.142; 0.152]$ with a confidence of 95%.



(a) (b) (c)

Fig. 5: (a) A 6×6 grid. The black fields are walls, the fields with vertical lines are on fire and the fields with horizontal lines contain ice. The circle indicates the robot's starting position and the square the goal.
(b) WTA implementing the random movement of the robot.
(c) Cumulative distribution of the robot reaching the goal, staying too long in the fire or too long on the ice.

6 Related and Future Work

To our knowledge, we are the first to propose and implement an algorithm for translation of $\mathrm{WMTL}_{\leq}$ formulas into monitoring automata. However, if we level down to $\mathrm{MITL}_{\leq}$, there are several translation procedures described in the literature that are dealing with this logic. First, Rajeev Alur in [3] presents a procedure that is mostly theoretical and is not intended to be practically implemented. Second, Oded Maler et al. [16] proposed a procedure to translate MITL into temporal testers (not the classic timed automata), their procedure also has not been implemented. Nir Piterman et al. [17] proposed an approach how to translate MTL to deterministic timed automata under *finite variability* assumption (this assumption is not valid for the WTA stochastic semantics that we use). Finally, Marc Geilen[12] has implemented a procedure to translate $\mathrm{MITL}_{\leq}$ to timed automata, but his approach works in continuous time semantics.

For future work we aim at extending our monitor- and approximate determinization constructions to $\mathrm{WMTL}_{[a,b]}$ with (non-singleton) cost interval-bounds on the U modality in order to allow for SMC for this more expressive logic. Here a challenge will be how to bound the length of the random runs to be generated.

References

1. G. Agha, J. Meseguer, and K. Sen. Pmaude: Rewrite-based specification language for probabilistic object systems. *Electronic Notes in Theoretical Computer Science*, 153(2):213–239, 2006.
2. R. Alur, C. Courcoubetis, N. Halbwachs, T. A. Henzinger, P. Ho, X. Nicollin, A. Olivero, J. Sifakis, and S. Yovine. The algorithmic analysis of hybrid systems. *Theoretical Computer Science*, 138(1):3–34, 1995.
3. R. Alur, T. Feder, and T. A. Henzinger. The benefits of relaxing punctuality. *J. ACM*, 43:116–146, January 1996.

4. R. Alur, S. L. Torre, and G. J. Pappas. Optimal paths in weighted timed automata. In *HSCC'01*, pages 49–62. Springer, 2001.
5. B. Barbot, T. Chen, T. Han, J.-P. Katoen, and A. Mereacre. Efficient ctmc model checking of linear real-time objectives. In P. A. Abdulla and K. R. M. Leino, editors, *TACAS*, volume 6605 of *Lecture Notes in Computer Science*, pages 128–142. Springer, 2011.
6. G. Behrmann, A. Fehnker, T. Hune, K. G. Larsen, P. Pettersson, J. Romijn, and F. W. Vaandrager. Minimum-cost reachability for priced timed automata. In M. D. D. Benedetto and A. L. Sangiovanni-Vincentelli, editors, *HSCC*, volume 2034 of *Lecture Notes in Computer Science*, pages 147–161. Springer, 2001.
7. P. Bouyer, K. G. Larsen, and N. Markey. Model checking one-clock priced timed automata. *Logical Methods in Computer Science*, 4(2), 2008.
8. F. Cassez and K. G. Larsen. The impressive power of stopwatches. In C. Palamidessi, editor, *CONCUR*, volume 1877 of *Lecture Notes in Computer Science*, pages 138–152. Springer, 2000.
9. J.-M. Couvreur. On-the-fly verification of linear temporal logic. In *FM '99*, pages 253–271, 1999.
10. A. David, K. G. Larsen, A. Legay, M. Mikucionis, D. Poulsen, J. van Vliet, and Z. Wang. Statistical model checking for networks of priced timed automata. In *Proceedings of Formal Modeling and Analysis of Timed Systems*, pages 80–96, Aalborg, 2011. LNCS.
11. A. David, K. G. Larsen, A. Legay, M. Mikucionis, and Z. Wang. Time for statistical model checking of real-time systems. In *Proceedings of the 23^{rd} International Conference on Computer Aided Verification (CAV)*, LNCS. Springer Verlag, 2011.
12. M. Geilen. An improved on-the-fly tableau construction for a real-time temporal logic. In *In International Conference on Computer Aided Verification*, pages 276–290. Springer, 2003.
13. M. Geilen and D. Dams. An on-the-fly tableau construction for a real-time temporal logic. In *FTRTFT*, pages 276–290, 2000.
14. E. M. C. Jr., O. Grumberg, and D. A. Peled. *Model Checking*. The MIT Press, 1999.
15. R. Koymans. Specifying real-time properties with metric temporal logic. *Real-Time Syst.*, 2:255–299, October 1990.
16. O. Maler, D. Nickovic, and A. Pnueli. From mitl to timed automata. In *FORMATS'06*, pages 274–289. Springer, 2006.
17. D. Nickovic and N. Piterman. From mtl to deterministic timed automata. In *8th International Conference on Formal Modelling and Analysis of Timed Systems*, volume 6246 of *LNCS*, pages 152–167. Springer, 2010.
18. K. Sen, M. Viswanathan, and G. Agha. On statistical model checking of stochastic systems. In *In Etessami, K., Rajamani, S.K., eds.: CAV. Volume 3576 of Lecture Notes in Computer Science*, pages 266–280. Springer, 2005.
19. H. L. S. Younes. *Verification and Planning for Stochastic Processes with Asynchronous Events*. PhD thesis, Carnegie Mellon University, 2005.
20. P. Zuliani, A. Platzer, and E. M. Clarke. Bayesian statistical model checking with application to simulink/stateflow verification. In *HSCC '10*, pages 243–252, New York, NY, USA, 2010. ACM.