On model checking durational Kripke structures (Extended abstract)

F. Laroussinie, N. Markey, and Ph. Schnoebelen

Lab. Spécification & Vérification
ENS de Cachan & CNRS UMR 8643
61, av. Pdt. Wilson, 94235 Cachan Cedex France
email: {fl,markey,phs}@lsv.ens-cachan.fr

Abstract. We consider quantitative model checking in *durational Kripke structures* (Kripke structures where transitions have integer durations) with timed temporal logics where subscripts put quantitative constraints on the time it takes before a property is satisfied.
We investigate the conditions that allow polynomial-time model checking algorithms for timed versions of *CTL* and exhibit an important gap between logics where subscripts of the form "$= c$" (exact duration) are allowed, and simpler logics that only allow subscripts of the form "$\leq c$" or "$\geq c$" (bounded duration).
A surprising outcome of this study is that it provides the second example of a Δ_2^p-complete model checking problem.

1 Introduction

Model checking (the automatic verification that a model fulfills temporal logic specifications) is widely used when designing and debugging critical reactive systems [Eme90,CGP99]. During the last decade, model checking has been extended to *real-time systems*, where quantitative information about timings is required.

Real-time model checking has been mostly studied and developed in the framework of Alur and Dill's *Timed Automata* [ACD93]. There now exists a large body of theoretical knowledge and practical experience for this class of systems, and it is agreed that their main drawback is the complexity blowup induced by timing constraints: all model checking problems are at least PSPACE-hard over Timed Automata [Alu91,CY92,ACD93,AL99].

However, there exist simpler families of timed models, for which polynomial-time model checking is possible. Usually, these are based on classical, discrete, Kripke structures (KS). Here there is no inherent concept of time (contrary to clocks in Timed Automata) and the elapsing of time is encoded by events. For example, in [EMSS92] each transition of a KS is viewed as taking exactly one time unit, and in [LST00] a "tick" proposition labels states where the clock is incremented. This framework is less expressive than Timed Automata, but it is conceptually simpler, it allows efficient model checking algorithms, and is convenient in many situations.

There are two main popular approaches for extending temporal logics with the ability to express timing aspects of computations (see [AH92] for a survey).

First, the use of *freeze variables* (also *formula clocks*) in temporal formulae allows the comparison of delays between events. The resulting logics are very expressive but often have hard model checking problems (because they make it possible to combine the timings of several different events in arbitrary ways).

A simpler approach is the use of timing constraints tagging temporal modalities. For example, the formula $\mathsf{EF}_{<10}\ A$ states that it is possible to reach a state verifying A ("$\mathsf{EF}\, A$") in less than 10 time units. These constraints are less expressive than freeze variables but they lead to more readable formulae, and sometimes allow easier model checking.

Timing constraints can have three main forms: "$\leq c$" and "$\geq c$" set a lower or upper bound for durations, while "$= c$" requires a precise value. *TCTL* is the extension of *CTL* with all three types of constraints, while $TCTL_{\leq,\geq}$ is the fragment of *TCTL* where the "$= c$" constraints are forbidden. Other classical temporal logics can be extended in the same way, and we call $TCTL^*$, $TLTL_{\leq,\geq}$, etc. the resulting formalisms.

Model checking *TCTL* over Kripke structures can be done in time[1] $O(|S|^3 \cdot |\varphi|)$ [EMSS92]. This is in sharp contrast with model checking over Timed Automata (PSPACE-complete [ACD93]) and with model checking *CTL* extended by freeze variables (PSPACE-complete over KSs [LST00]).

Thus it appears that polynomial-time model checking of timed properties is possible if one picks the right logic (e.g. *TCTL*) and the right models (e.g. KSs).

Our contribution. In this paper, we propose and study *durational Kripke structures* (DKSs), a very natural extension of KSs. As illustrated in Fig. 1, a DKS is a KS where transitions have possible durations specified by an interval of integers. Such structures generalize the models of [EMSS92] or [LST00] and provide a higher-level viewpoint. For example, steps having long durations can be modeled without long sequences of transitions. Also, the size of a DKS is mostly insensitive to a change of time scale. Still, the model does not allow anything like the synchronization of several clocks in Timed Automata: the main limitation of DKSs is that there is no simple way to combine several components.

We show that model checking DKSs can be done in polynomial time when $TCTL_{\leq,\geq}$ is considered, i.e. when exact durations are not allowed as subscripts of modalities. This extends the positive results from [EMSS92,LST00] to a more expressive class of models.

Allowing exact duration constraints increases the complexity of model checking: we show that model checking *TCTL* over DKSs is Δ_2^p-complete. This last result is technically involved, and it is also quite surprising since Δ_2^p, the class

[1] In such statements, $|S|$ denotes the size of the structure, and $|\varphi|$ the length of the temporal formula.

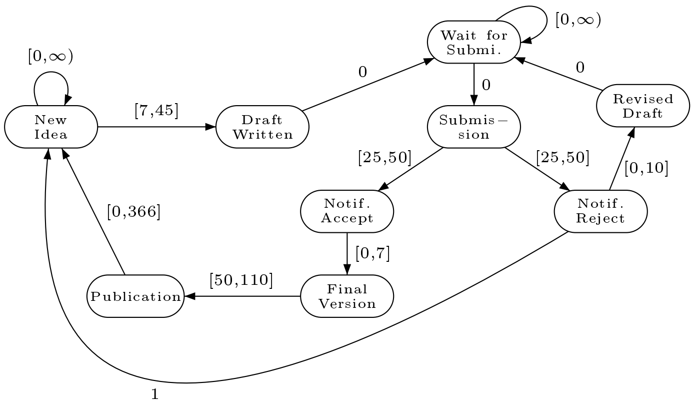


Fig. 1. A DKS modeling publications by one author (time in days)

P^{NP} of problems that can be solved by a deterministic polynomial-time Turing machine that has access to an NP oracle [Sto76,Pap94], does not contain many natural complete problems [Pap84,Wag87,Kre88]. Indeed, the only known Δ_2^p-complete problem from the field of temporal model checking has only been recently identified [LMS01].

Finally, we show that exact duration constraints induce a similar complexity blowup when model checking DKSs with other logics like $TLTL$ and $TCTL^*$.

Related work. Quantitative logics *for Timed Automata* are now well-known and many results are available regarding their expressive power, or the complexity of satisfiability and model checking [AH94,ACD93,AH93,AFH96,Hen98]. That exact durations may induce harder model checking complexity was already observed in the case of $TLTL$ and Timed Automata [AFH96].

The literature contains several models that are close to DKSs but mostly linear-time logics were considered [Ost90,AH94] and this makes model checking at least PSPACE-hard.

Emerson *et al.* give polynomial time algorithms for model checking $TCTL$ over discrete KSs in [EMSS92] and $TCTL_{\leq}$ over "tight DKS" (all intervals are singletons) in [ET99, section 4]. They also study model checking for quantitative logics with more complex constraints in [ET97,ET99]. Model checking $TCTL$ over "small-step DKS" (see section 2) is considered in [LST00] where the expressive power of constraints is investigated. [Lew90] describes a quantitative CTL over discrete timed structures but does not investigate complexity of model checking.

2 Durational Kripke structures

We write $\mathbb{N}$ for the set of natural numbers, and $\mathcal{I}_{\mathbb{N}}$ (or just $\mathcal{I}$) for the set of intervals over $\mathbb{N}$. An interval $\rho \in \mathcal{I}$ is either finite (of the form "$[n, m]$") or right-open and infinite (of the form "$[n, \infty)$").

Assume a countable set $AP = \{P_1, P_2, \ldots\}$ of *atomic propositions.*

Definition 2.1. *A* durational Kripke structure *(DKS) is a 3-tuple* $S = \langle Q, R, l\rangle$ *where* Q *is a set of* states, $R \subseteq Q \times \mathcal{I} \times Q$ *is a total* transition relation with duration *and* $l : Q \to 2^{AP}$ *labels every state with a subset of* AP.

Below we only consider finite DKSs, such that Q, R and all $l(q)$ are finite sets.

Graphically, a DKS is just a directed graph where a triple $(q, \rho, q') \in R$ is depicted as a ρ-labeled edge from q to q'.

Semantics. The intended meaning of an edge (q, ρ, q') is that it is possible to move from q to q' with any *integer* duration d belonging to the interval ρ. We write $q \xrightarrow{d} q'$ when $d \in \rho$ for some $(q, \rho, q') \in R$. A sequence $\pi = q_0 \xrightarrow{d_0} q_1 \xrightarrow{d_1} q_2 \ldots$ with $q_i \xrightarrow{d_i} q_{i+1} \in R$ for all i is called a *path* if it is finite and a *run* if it is infinite. A *simple* path is a path where no state is visited twice (a loop-free path). For a run π, $\pi_{|n}$ is the path obtained by only considering the first n steps in π. For $q \in Q$, we let $\mathrm{Exec}(q)$ denote the set of runs starting from q: because R is total, any state is the start of at least one run.

The *size* (or *length*) of a path $\pi = q_0 \xrightarrow{d_0} q_1 \xrightarrow{d_1} q_2 \cdots q_n$ is n (the number of steps), and its *duration*, denoted $\mathsf{Time}(\pi)$, is $d_0 + \cdots + d_{n-1}$.

Example 2.2. The DKS of Fig. 1 models the publication process of one busy researcher, assuming time in counted in days. (This example does not distinguish between the name of the states and their labeling by propositions. Also, singleton intervals $[n, n]$ are written simply n.) A property we would like to express (and model-check) is "*whenever a notification is received, either publication or submission occurs in less than 150 days*".

Restricted DKSs. There are several natural restrictions one can put on the general model of DKS:

- A *tight DKS* is a DKS where all intervals are singletons. The Timed Kripke structures of [ET99, section 4] are tight DKSs and the Timed State Graphs considered in [AH94] are equivalent to tight DKS. Below we show that, in general, restricting to tight DKSs does not make model checking easier.
- A *small-step DKS* (a ssDKS) is a tight DKS where all steps have duration 0 or 1. The model used in [LST00] is very close to small-step DKSs, but the duration information, "0 or 1 time unit?", is carried by the nodes.
- A KS is a small-step DKS where all steps have duration 1. This is the model assumed in [EMSS92].

There are fundamental differences between ssDKSs and DKSs. First, if there is a path connecting some q to some q', then the shortest such path has duration at most $|Q| - 1$ in a ssDKS, while it can have exponential duration in DKSs.

Moreover, in ssDKSs time progresses smoothly along paths: a path π of duration c can always be decomposed into two subpaths $\pi = \pi'.\pi''$ with $\mathsf{Time}(\pi') = \lfloor \frac{c}{2} \rfloor$ and $\mathsf{Time}(\pi'') = \lceil \frac{c}{2} \rceil$. This property plays a crucial rôle in efficient *TCTL* model checking algorithms over ssDKSs [EMSS92,LST00].

3 Quantitative temporal logic

TCTL is the quantitative extension of *CTL* where temporal modalities are subscripted with constraints on duration [ACD93]. Here it is interpreted over DKSs states.

Definition 3.1 (Syntax of *TCTL*). *TCTL formulae are given by the following grammar:*

$$\varphi, \psi \ ::= \ P_1 \mid P_2 \mid \ldots \mid \neg\varphi \mid \varphi \wedge \psi \mid \mathsf{EX}\varphi \mid \mathsf{E}\varphi\mathsf{U}_{\sim c}\,\psi \mid \mathsf{A}\varphi\mathsf{U}_{\sim c}\,\psi$$

where $\sim$ can be any comparator in $\{<, \leq, =, \geq, >\}$ and c any natural number.

Standard abbreviations include $\top, \bot, \varphi \vee \psi, \varphi \Rightarrow \psi, \ldots$ as well as $\mathsf{AX}\varphi$ (for $\neg\mathsf{EX}\neg\varphi$), $\mathsf{EF}_{\sim c}\,\varphi$ (for $\mathsf{E}\top\mathsf{U}_{\sim c}\,\varphi$), $\mathsf{AF}_{\sim c}\,\varphi$ (for $\mathsf{A}\top\mathsf{U}_{\sim c}\,\varphi$), $\mathsf{EG}_{\sim c}\,\varphi$ (for $\neg\mathsf{AF}_{\sim c}\,\neg\varphi$) and $\mathsf{AG}_{\sim c}\,\varphi$ (for $\neg\mathsf{EF}_{\sim c}\,\neg\varphi$). Further, the modalities U, F and G without subscripts are shorthand for $\mathsf{U}_{\geq 0}$, etc. The size $|\varphi|$ of a formula φ is defined in the standard way, with constants written in binary notation.

Definition 3.2 (Semantics). *The following clauses define when a state q of some DKS S, satisfies a TCTL formula φ, written $q \models \varphi$, by induction over the structure of φ (semantics of boolean operators is omitted).*

$$\begin{array}{ll}
q \models \mathsf{EX}\varphi & \text{iff there is a } q \xrightarrow{d} q' \text{ s.t. } q' \models \varphi, \\
q \models \mathsf{E}\varphi\mathsf{U}_{\sim c}\,\psi & \text{iff there is a run } \pi \text{ of the form } q = q_0 \xrightarrow{d_0} q_1 \xrightarrow{d_1} q_2 \cdots \text{ and a } n \\
 & \quad \text{s.t. } \mathsf{Time}(\pi_{|n}) \sim c,\ q_n \models \psi, \text{ and } q_i \models \varphi \text{ for all } 0 \leq i < n, \\
q \models \mathsf{A}\varphi\mathsf{U}_{\sim c}\,\psi & \text{iff for all runs } \pi \text{ of the form } q = q_0 \xrightarrow{d_0} q_1 \xrightarrow{d_1} q_2 \cdots \text{ there is a } n \\
 & \quad \text{s.t. } \mathsf{Time}(\pi_{|n}) \sim c,\ q_n \models \psi, \text{ and } q_i \models \varphi \text{ for all } 0 \leq i < n.
\end{array}$$

Thus, in $\mathsf{E}\varphi\mathsf{U}_{\sim c}\,\psi$, the classical until is extended by requiring that ψ be satisfied within a duration (from the current state) verifying the constraint "$\sim c$".

Here are some examples of *TCTL* formulae stating expected properties for the DKS from Figure 1:

$$\mathsf{AG}(\texttt{New\_Idea} \Rightarrow \neg\mathsf{EF}_{<100}\,\texttt{Publication})$$
$$\mathsf{AG}(\texttt{Submission} \Rightarrow \mathsf{AF}_{<40}\,(\texttt{Publication} \vee \texttt{Revised\_Draft} \vee \texttt{New\_Idea}))$$

The first formula states that a new idea is never followed by a publication in less than 100 days. The second formula states that any submission is inevitably followed by a notification of acceptance, a revised draft, or a new idea, in less than 40 days.

Equivalent formulae. We write $\varphi \equiv \psi$ when φ and ψ are *equivalent* (every state of every DKS satisfies $\varphi \Leftrightarrow \psi$) and $\varphi \equiv_{\text{ss}} \psi$ when the equivalence only holds in states of small-step DKSs.

The following equivalences hold:

$$\mathsf{A}\,\varphi\,\mathsf{U}_{\leq c}\,\psi \equiv \mathsf{AF}_{\leq c}\,\psi \;\wedge\; \neg\mathsf{E}(\neg\psi)\mathsf{U}(\neg\varphi \wedge \neg\psi) \tag{1}$$

$$\mathsf{A}\,\varphi\,\mathsf{U}_{\geq c}\,\psi \equiv \mathsf{AG}_{< c}\,\Big(\varphi \;\wedge\; \mathsf{A}\,\varphi\,\mathsf{U}_{>0}\,\psi\Big) \tag{2}$$

Some seemingly natural equivalences only hold for small-step DKSs (but do not hold for DKSs in general). For example:

$$\mathsf{E}\,\varphi\,\mathsf{U}_{\leq c}\,\psi \equiv_{ss} \mathsf{E}\,\varphi\,\mathsf{U}_{\leq 1}\,(\psi \vee \mathsf{E}\,\varphi\,\mathsf{U}_{\leq c-1}\,\psi) \tag{3}$$

$$\mathsf{E}\,\varphi\,\mathsf{U}_{=c}\,\psi \equiv_{ss} \mathsf{E}\,\varphi\,\mathsf{U}_{=1}\,(\mathsf{E}\,\varphi\,\mathsf{U}_{=c-1}\,\psi) \tag{4}$$

$$\mathsf{E}\,\varphi\,\mathsf{U}_{\geq c}\,\psi \equiv_{ss} \mathsf{E}\,\varphi\,\mathsf{U}_{=c}\,(\mathsf{E}\,\varphi\,\mathsf{U}\,\psi) \tag{5}$$

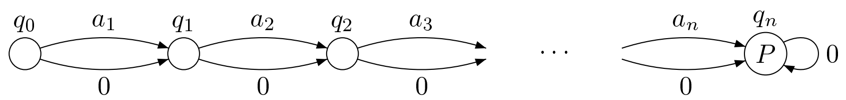
4 Model checking *TCTL* over DKSs

The model checking problem we consider in this section is, given some DKS S, some state q, and some $TCTL$ formula φ, to decide whether $q \models \varphi$.

Model checking algorithms for $TCTL$ have to deal with the timing constraints carried by the modalities. Constraints of the form "$= c$" (exact durations) are usually more difficult than inequality constraints. Furthermore, when dealing with DKSs, the durations associated with the transitions make the problem even harder. Indeed NP-hard problems appear for simple formulae:

Proposition 4.1 (Hardness of reachability with exact duration). *Model checking formulae of the form $\mathsf{EF}_{=c}\,P$ over DKS is NP-hard.*

Proof. By reduction from SUBSET-SUM [GJ79, p. 223]: an instance is a finite set $A = \{a_1, \dots, a_n\}$ of natural numbers and some number D. One asks whether there exists a subset A' of A such that $D = \sum_{a \in A'} a$. This is the case iff $q_0 \models \mathsf{EF}_{=D}\,P$ in the following DKS:



□

Therefore model checking $TCTL$ over DKSs is NP-hard and coNP-hard. Now, the problem is clearly in PSPACE: one can easily adapt algorithms for model checking $TCTL$ over Timed Automata (a PSPACE-complete problem [ACD93]) to handle DKSs.

4.1 Polynomial-time model checking for restricted cases

It turns out that polynomial-time model checking remains possible as long as equality constraints *or* durations on transitions are forbidden, as we now show.

Let $TCTL_{\leq,\geq}$ denote the fragment of $TCTL$ where equality constraints on modalities are not allowed:

Theorem 4.2.
Model checking $TCTL_{\leq,\geq}$ over DKSs can be done in time $O\big(|S|^2 \cdot |\varphi|\big)$.

Proof. It is enough to extend the classical *CTL* algorithm with labeling procedures running in time $|S|^2.\lceil \log c \rceil$ for each modality $\mathsf{E}\,P_1\,\mathsf{U}_{\sim c}\,P_2$ and $\mathsf{A}\,P_1\,\mathsf{U}_{\sim c}\,P_2$.

φ is $\mathsf{E}\,P_1\,\mathsf{U}_{\leq c}\,P_2$: We restrict to the subgraph where only states satisfying $\mathsf{E}\,P_1\,\mathsf{U}\,P_2$ have been kept, and where we only consider the left extremity of intervals ρ on edges. Then for every state q we compute the smallest duration (call it c_q) such that $q \models \mathsf{E}P_1\mathsf{U}_{\leq c_q}P_2$. This can be done in time $O(|S^2|)$ using a classical *single-source shortest paths* algorithm [CLR90]. Then $q \models \varphi$ iff $c_q \leq c$.

φ is $\mathsf{E}\,P_1\,\mathsf{U}_{\geq c}\,P_2$: We start with some preprocessing of S: let us introduce a new proposition $P_{\mathsf{SCC}^+(\psi)}$ and use it to label every node belonging to a strongly connected set of nodes satisfying ψ and where at least one edge allows a strictly positive duration. That is, $q \models P_{\mathsf{SCC}^+(\psi)}$ iff it is possible to loop on ψ-states around q with ever increasing durations. Labeling states for $P_{\mathsf{SCC}^+(\psi)}$ can be done in time $O(|S|)$ once they are labeled for ψ.
We can now solve the original problem. There are two ways a state can satisfy φ. Either a simple path is enough, or a path with loops is required so that a long enough duration is reached. We check the existence of a path of the first kind with a variant of the earlier shortest paths method, this times geared towards *longest acyclic paths*. We check for the existence of a path of the second kind by model checking the *CTL* formula $\mathsf{E}P_1\mathsf{U}(P_{\mathsf{SCC}^+(P_1)} \wedge \mathsf{E}P_1\mathsf{U}P_2)$.

φ is $\mathsf{A}\,P_1\,\mathsf{U}_{\leq c}\,P_2$: We reduce to the previous cases using equivalence (1) and $\mathsf{AF}_{\leq c}\,\psi \equiv \neg\mathsf{E}\neg\psi\,\mathsf{U}_{>c}\top \wedge \neg\mathsf{E}\neg\psi\mathsf{U}\,P_{\mathsf{SCC}^0(\neg\psi)}$. Here $P_{\mathsf{SCC}^0(\neg\psi)}$ labels strongly connected components where one can loop on $\neg\psi$-states using only transitions allowing for zero durations.

φ is $\mathsf{A}\,P_1\,\mathsf{U}_{\geq c}\,P_2$: We reduce to the previous cases using equivalence (2) and $\mathsf{AG}_{<c}\,\varphi \equiv \neg\mathsf{EF}_{<c}\,\neg\varphi$. ☐

Note that this result extends Theorems 3 and 4 of [ET99] on model checking $\mathsf{E}\,P_1\,\mathsf{U}_{\leq c}\,P_2$ and $\mathsf{EG}_{\leq c}\,P$ over tight DKSs.

When equality constraints are allowed but arbitrary large durations in DKSs are forbidden, we rely on the following result:

Theorem 4.3 ([EMSS92,LST00]).
Model checking TCTL over small-step DKSs can be done in polynomial-time.

Now, since model checking is already P-hard for *CTL* over KSs, model checking $TCTL_{\leq,\geq}$ over DKSs, or *TCTL* over ssDKSs, is PTIME-complete.

4.2 Δ_2^p model checking for *TCTL* over DKSs

Allowing both exact durations *and* general DKSs makes model checking harder (Prop. 4.1) but this is not enough to make the problem PSPACE-complete[2]. In fact, we have:

[2] This sentence assumes there is no collapse in the polynomial-time hierarchy.

Proposition 4.4. *Model checking TCTL over DKSs is in Δ_2^p.*

Proof. A natural Δ_2^p algorithm is to use the natural *CTL*-like labeling algorithm, accessing an NP oracle for the basic modalities. Theorem 4.2 provides deterministic polynomial-time solutions for modalities where exact duration is not used. Therefore it remains to provide NP routines for modalities of the form $\mathsf{E}P_1 \mathsf{U}_{=c} P_2$ and $\mathsf{A}\, P_1\, \mathsf{U}_{=c}\, P_2$. We deal with the $\mathsf{E}\, P_1\, \mathsf{U}_{=c}\, P_2$ modalities in Lemma 4.5, and with the $\mathsf{A}\, P_1\, \mathsf{U}_{=c}\, P_2$ modalities in Lemma 4.6. □

Lemma 4.5. *Model checking the formula $\mathsf{E}P_1 \mathsf{U}_{=c}\, P_2$ over DKSs is in* NP.

Proof. Let $S = \langle Q, R, l\rangle$ be a DKS. We first deal with the simpler case where S is tight (all intervals labeling R are singletons).

Assume there exists a path $\pi = q_0 \xrightarrow{d_0} q_1 \xrightarrow{d_1} q_2 \cdots q_n$ in S witnessing $q_0 \models \mathsf{E}\, P_1\, \mathsf{U}_{=c}\, P_2$. We can assume $n < c \cdot |Q|$ since any null duration loop can be removed from π, but this is not enough to guarantee that π has size polynomial in $|S| + \lceil \log c\rceil$.

With π we associate the Parikh image of its transitions, that is, the map $\Phi_\pi \colon R \mapsto \mathbb{N}$ that counts the number of times each transition appears in π. Such a Φ also counts the number of times each node is entered and left: $\Phi^{\mathrm{i}}(q) = \sum\{\Phi(t) \mid t \text{ enters } q\}$ and $\Phi^{\mathrm{o}}(q) = \sum\{\Phi(t) \mid t \text{ leaves } q\}$.

Obviously, Φ_π satisfies the following properties:

1. $\Phi^{\mathrm{i}}_\pi(q) = \Phi^{\mathrm{o}}_\pi(q)$ for any q different from q_0 and q_n. Furthermore, if $q_0 = q_n$, then $\Phi^{\mathrm{i}}_\pi(q_0) = \Phi^{\mathrm{o}}_\pi(q_0)$, otherwise $\Phi^{\mathrm{o}}_\pi(q_0) - \Phi^{\mathrm{i}}_\pi(q_0) = 1 = \Phi^{\mathrm{i}}_\pi(q_n) - \Phi^{\mathrm{o}}_\pi(q_n)$.
2. The subgraph of S induced by the transitions $t \in R$ with $\Phi_\pi(t) > 0$ is connected.
3. Φ_π has duration c, i.e. $c = \sum\{d \cdot \Phi(t) \mid t = (q, [d, d], q') \in R\}$.
4. $q_n \models P_2$ and $q \models P_1$ for any state q such that $\Phi^{\mathrm{o}}_\pi(q) > 0$.

Conversely, if some Φ (with q_0, q_n) fulfills conditions 1. and 2., then by Euler circuit theorem, Φ is Φ_π for some path π from q_0 to q_n in S. If conditions 3. and 4. also hold, then π proves that $q_0 \models \mathsf{E}\, P_1\, \mathsf{U}_{=c}\, P_2$. If we assume $n < c \cdot |Q|$, then Φ can be encoded in polynomial-size, conditions 1 to 4 can be checked in polynomial-time, and Φ (with q_n) can be used as the polynomial-size witness we need for an NP algorithm.

Now, if we remove the assumption that S is tight, it is enough to replace condition 3. by

$$\sum_{t=(q,\rho,q')} \min(\rho) \cdot \Phi(t) \;\leq\; c \;\leq \sum_{t=(q,\rho,q')} \max(\rho) \cdot \Phi(t)$$

□

Lemma 4.6. *Model checking the formula $\mathsf{A}\, P_1\, \mathsf{U}_{=c}\, P_2$ over DKSs is in co*NP.

Proof (Sketch). Since $\mathsf{A}\, P_1\, \mathsf{U}_{=c}\, P_2 \;\equiv\; \mathsf{A}\, P_1\, \mathsf{U}_{\geq c}\, P_2 \wedge \neg \mathsf{EG}_{=c}\, \neg P_2$, it is enough to show that model checking formulae of the form $\mathsf{EG}_{=c}\, P$ can be done in NP. This is done using techniques similar to the previous Lemma. (One difference is that we have to consider two cases: the path visits duration c, or it avoids it.) □

4.3 Δ_2^p-hardness of *TCTL* model checking

We now show that model checking $TCTL$ over DKSs is Δ_2^p-hard, and hence Δ_2^p-complete. This means that there is no essentially better way for model checking $TCTL$ over DKS than the labeling algorithm used in Prop. 4.4.

Proving Δ_2^p-hardness is difficult in part because there exist very few natural problems that are Δ_2^p-complete and that could be used in reductions to $TCTL$ model checking. Here we capitalize on our recent proof that model-checking $FCTL$ is Δ_2^p-complete [LMS01] and follow its pattern. However, this pattern must be altered and we have to encode boolean problems in numerical problems. Since model-checking $TCTL$ becomes polynomial-time when the numerical constants are written in unary, the Δ_2^p-hardness proof has to encode information in the bits of the numbers used in the DKS and the $TCTL$ formula.

A Δ_2^p-complete problem. We start by briefly recalling SNSAT (for *sequentially nested* satisfiability), the Δ_2^p-complete satisfiability problem we reduce from. An instance $\mathcal{I}$ of SNSAT has the form

$$\mathcal{I} \;=\; \begin{bmatrix} x_1 := \exists Z_1\ F_1(Z_1), \\ x_2 := \exists Z_2\ F_2(x_1, Z_2), \\ \quad\vdots \\ x_n := \exists Z_n\ F_n(x_1, \dots, x_{n-1}, Z_n) \end{bmatrix}$$

where each F_i is a boolean expression, each Z_i is a set of (auxiliary) boolean variables, and the x_i are the main variables. We write X for $\{x_1, \dots, x_n\}$, Z for $Z_1 \cup \dots \cup Z_n$, and assume the sets $X, Z_1, \dots, Z_n$ are pairwise disjoint. Var denotes $X \cup Z$ and $p = |Z|$.

W.l.o.g., we assume every F_i is a 3-CNF of the form $\bigwedge_l \bigvee_{m=1}^3 \alpha_{i,l,m}$ where the $\alpha_{i,l,m}$ are literals. With every disjunct $\bigvee_m \alpha_{i,l,m}$ we associate a clause $C_{i,l}$ of the form $\overline{x_i} \vee \bigvee_m \alpha_{i,l,m}$ and write $Cl = \{C_1, \dots, C_r\}$ for the resulting set of clauses.

$\mathcal{I}$ defines a unique valuation $v_\mathcal{I}$ of the variables in X where $v_\mathcal{I}(x_i) = \top$ iff $F_i(v_\mathcal{I}(x_1), \dots, v_\mathcal{I}(x_{i-1}), Z_i)$ is satisfiable. The computational problem called SNSAT is, given an instance $\mathcal{I}$ as above, to decide whether $v_\mathcal{I}(x_n) = \top$. Therefore $\mathcal{I}$ can be seen as a sequence of n satisfiability problems where the ith problem depends on the answers of the earlier problems.

With this in mind, we say a valuation w of Var is:

safe: if, for all $i = 1, \dots, n$, $w(x_i)$ implies $F_i(w(x_1), \dots, w(x_{i-1}), w(Z_i))$,
correct: if, for all $i = 1, \dots, n$, $w(x_i) = F_i(w(x_1), \dots, w(x_{i-1}), w(Z_i))$,
admissible: if w is correct and coincide with $v_\mathcal{I}$ over X.

A correct valuation is safe and is also consistent for negative values assigned to some x_i. Still, this does not guarantee that the values of variables in Z are best possible, i.e. that w is admissible. An arbitrary valuation over Z extends into a correct valuation in a unique way, and checking that a given w is correct can be done in polynomial-time.

An admissible valuation is just a valuation for Z that yields $v_\mathcal{I}$ for X. Hence it is optimal over Z. Clearly, admissible valuations exist for any SNSAT instance, positive ($v_{\mathcal{I}}(x_n) = \top$) or negative, but checking that a given w is admissible is Δ_2^p-complete.

Reducing SNSAT to TCTL model checking. Fix some $K \in \mathbb{N}$. To variables $u \in \mathit{Var}$ and clauses $C \in \mathit{Cl}$ we assign weights $s(u)$ and $s(C)$ given by:

$$s(x_i) = K^i \qquad s(z_i) = K^{n+i} \qquad s(C_i) = K^{n+p+i}$$

A multiset $\mathcal{M}$ of variables and clauses ($\mathcal{M} \in \mathbb{N}^{\mathit{Var} \cup \mathit{Cl}}$) has weight $s(\mathcal{M}) = \sum_x s(x) \times \mathcal{M}(x)$. Now if $\mathcal{M}(x) < K$ and $\mathcal{M}'(x) < K$ for all $x \in \mathit{Var} \cup \mathit{Cl}$, then $s(\mathcal{M}) = s(\mathcal{M}')$ iff $\mathcal{M} = \mathcal{M}'$. Therefore, by picking K large enough, we can reduce the equality of small multisets to the equality of their weights.

We now build $\mathcal{S}_{\mathcal{I}}$, a DKS associated with $\mathcal{I}$. See Fig. 2. Nodes in $\mathcal{S}_{\mathcal{I}}$ are of

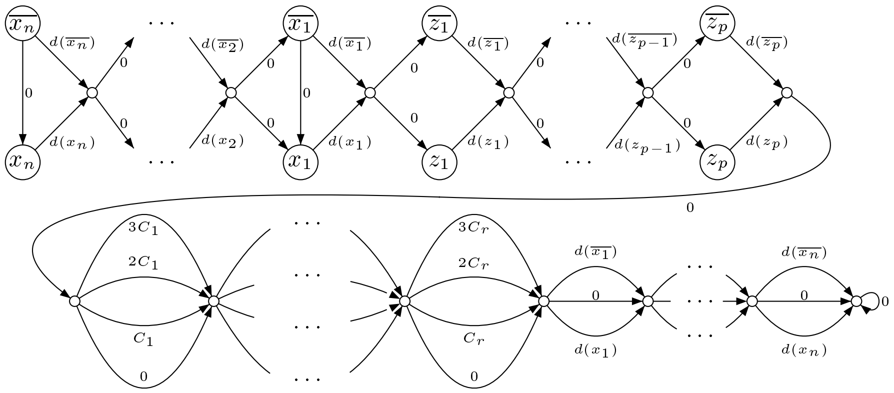


Fig. 2. Kripke structure $S_{\mathcal{I}}$ associated with SNSAT instance $\mathcal{I}$

two kinds: literal nodes (in the upper part of the figure) and filling nodes (in the lower part). With a path through the literal nodes that avoids the vertical "$\overline{x_i} \to x_i$" edges one associates a valuation of Var in the obvious way. The filling nodes are there for accounting purposes (see below).

For a literal α of the form $\pm u$, the duration $d(\alpha)$ is defined as $s(u) + \sum\{s(C) \mid C \in \mathit{Cl}, \alpha \Rightarrow C\}$. Therefore a path through the literal nodes will collect in its duration the weight of all the variables it visits plus the weight of all the clauses these literals satisfy (each clause being counted up to four times since it may be satisfied thanks to four different literals). Then the path visits the filling nodes where it can gather further clause or literal weights.

Now define

$$K' \stackrel{\text{def}}{=} \sum_{u \in \mathit{Var}} s(u) + 4 \times \sum_{C \in \mathit{Cl}} s(C)$$

and assume K is large enough (here $K > 11$ suffices). Then, for any $u \in \mathit{Var}$, a path π of weight K' must collect $d(u)$ or $d(\overline{u})$ once and only once. Thus π defines a valuation of Var. Furthermore π has to gather 4 times the weight of all clauses from Cl. Since, for $C \in \mathit{Cl}$, we can only collect $3s(C)$ via filling nodes, π must visit at least one literal that satisfies C.

Hence paths of length K' correspond to valuations that satisfy all the clauses. We rely on this and introduce the following TCTL formulae:

$$\varphi_0 \stackrel{\text{def}}{=} \top,$$
$$\text{and, for } k > 0, \quad \varphi_k \stackrel{\text{def}}{=} \mathsf{E}\Big[P_{\overline{x}} \Rightarrow \mathsf{EX}\big(P_x \land \neg\varphi_{k-1}\big)\Big]\mathsf{U}_{=K'}\top,$$

where P_x (resp. $P_{\overline{x}}$) is an atomic proposition that labels the n positive x_i nodes (resp. the $\overline{x_i}$ nodes).

We can now link $v_{\mathcal{I}}$ and the φ_k by:

Lemma 4.7. *For $k \in \mathbb{N}$ and $r = 1, \ldots, n$:*
(a) if $k \geq 2r - 1$ then $(v_{\mathcal{I}}(x_r) = \top$ iff $\mathcal{S}_{\mathcal{I}}, x_r \models \varphi_k)$,
(b) if $k \geq 2r$ then $(v_{\mathcal{I}}(x_r) = \bot$ iff $\mathcal{S}_{\mathcal{I}}, \overline{x_r} \models \varphi_k)$.

Proof. By induction on k. The case $k = 0$ holds vacuously. We now assume that $k > 0$ and that the Lemma holds for $k - 1$.
i. We prove the "$\Rightarrow$" direction of both "iff"s.

Let w be an admissible valuation. We use w to build a path π that starts at x_r (or $\overline{x_r}$ if $w(x_r) = \bot$), has total duration K', and only visit literals true under w (such a π exists because w is admissible). We claim π proves $x_r \models \varphi_k$ (or $\overline{x_r} \models \varphi_k$). This only requires that all nodes visited by π satisfy $P_{\overline{x}} \Rightarrow \mathsf{EX}(P_x \land \neg\varphi_{k-1})$ but on $\mathcal{S}_{\mathcal{I}}$ this translates into "$w(x_i) = \bot$ for $i \leq r$ implies $x_i \models \neg\varphi_{k-1}$" and is given by the induction hypothesis.
ii. We now prove the "$\Leftarrow$" direction of both "iff"s.

Assume $k \geq 2r - 1$ and $x_r \models \varphi_k$ (or $k \geq 2r$ and $\overline{x_r} \models \varphi_k$). Thus there is a path π starting from x_r (or $\overline{x_r}$), with duration K', and only visiting states satisfying $P_{\overline{x}} \Rightarrow \mathsf{EX}(P_x \land \neg\varphi_{k-1})$. Since $\mathsf{Time}(\pi) = K'$ the valuation w induced by π satisfies all $C \in \mathit{Cl}$. We further claim that $w(x_i) = v_{\mathcal{I}}(x_i)$ for $i = 1, \ldots, r$ and prove this by induction over i:
iia. If $w(x_i) = \top$ then $\bigwedge_l \bigvee_m w(\alpha_{i,l,m}) = \top$, so that $F_i(w(x_1), \ldots, w(x_{i-1}), Z_i)$ is satisfiable. By ind. hyp. we get that $F_i(v_{\mathcal{I}}(x_1), \ldots, v_{\mathcal{I}}(x_{i-1}), Z_i)$ is satisfiable, so that $v_{\mathcal{I}}(x_i) = \top$.
iib. If $w(x_i) = \bot$ then $\overline{x_i} \models \mathsf{EX}(P_x \land \neg\varphi_{k-1})$, implying $x_i \models \neg\varphi_{k-1}$. If $i < r$ we have $k - 1 \geq 2i - 1$ and, by ind. hyp., $v_{\mathcal{I}}(x_i) = \bot$. If $i = r$ then we are dealing with the case $k \geq 2r$ and $\overline{x_k} \models \varphi_k$, so that $k - 1 \geq 2i - 1$ and again $v_{\mathcal{I}}(x_i) = \bot$ by ind. hyp. □

Proposition 4.8. *Model checking TCTL over DKSs is Δ_2^p-hard.*

Proof. By Lemma 4.7, $\mathcal{I}$ is a positive instance iff $S_{\mathcal{I}}, x_n \models \varphi_{2n-1}$. Observe that $\mathcal{S}_{\mathcal{I}}$ and φ_{2n-1} can be built in logspace from $\mathcal{I}$. Thus SNSAT, a Δ_2^p-complete [LMS01], reduces to TCTL model checking. □

Theorem 4.9. *Model checking TCTL over DKSs is Δ_2^p-complete.*

Proof. Combine Props 4.4 and 4.8. □

Remark 4.10. Theorem 4.9 can be strengthened in various ways. E.g. note that $\mathcal{S}_\mathcal{I}$ is a *tight DKS*. Further, we used the EX modality in φ_k but this is not necessary (and could be replaced by $\mathsf{EF}_{\leq 0}$). Moreover $\mathcal{S}_\mathcal{I}$ contains transitions with null duration but it is easy to adapt the construction and show that Theorem 4.9 still holds over tight DKSs with strictly positive durations.

5 When are exact durations harder over DKSs?

Comparing Theorems 4.2 and 4.9 shows that allowing subscripts "$= c$" makes model checking *TCTL* over DKSs significantly harder. There exist other situations where exact duration subscripts make problems harder. For example:

- In small-step DKSs, model checking *TCTL* and $TCTL_{\leq,\geq}$ are both PTIME-complete[3] but satisfiability is harder for *TCTL* than for $TCTL_{\leq,\geq}$ [EMSS92].
- Over Timed Automata, exact duration subscripts do not make model checking harder for *TCTL* (PSPACE-complete for *TCTL* [ACD93]), but they do for the linear time temporal logic *MITL* [AFH96].

In this section we consider how exact duration subscripts do or do not increase the cost of model checking when the models are DKSs and the logic is a timed variant of classic temporal logics like *LTL* or CTL^*.

We will write *TLTL*, $TCTL^*$ and $TCTL^+$ for the timed variants of the logics *LTL*, CTL^* and CTL^+ (definitions omitted, see [Eme90]) and will let $TLTL_{\leq,\geq}$, etc., denote the fragments where exact duration is not allowed.

5.1 Model checking *TLTL* over DKSs

TLTL formulae are *path* formulae and are interpreted over runs in a DKS. As usual in this case, we consider *existential model checking*, that is the problem of deciding for a DKS S, a state q and a formula φ, whether there exists a path from q verifying φ.

Theorem 5.1.
1. Model checking TLTL over DKSs (and ssDKSs) is EXPSPACE*-complete.*
2. Model checking $TLTL_{\leq,\geq}$ over DKSs (and ssDKSs) is PSPACE*-complete.*

Proof. 1. EXPSPACE-hardness: it is possible to describe with a *TLTL* formula the accepting runs of a Turing Machine that runs in space 2^n. As usual, a run of the TM is seen as a sequence of instantaneous descriptions (i.d.). Here each i.d. has length 2^n. One easily writes that any two consecutive i.d.'s agree with the TM rules by means of the $F_{=2^n}$ modality, a modality of size $O(n)$.

[3] More precisely, model checking full *TCTL* can be done in time $O(|S|^3 \cdot |\varphi|)$ while model checking $TCTL_{\leq,\geq}$ can be done in time $O(|S| \cdot |\varphi|)$ [LST00].

Membership in EXPSPACE: this can be seen as a special case of the EXPSPACE upper bound for TPTL [AH94], a logic more expressive than *TLTL* interpreted over “timed state graphs” (a model in which one can encode DKSs).

2. PSPACE-hardness: comes from PSPACE-hardness of *LTL* model checking.

Membership in PSPACE: [AFH96] shows that model checking $MITL_{<,>}$ (a logic equivalent to $TLTL_{\leq,\geq}$) over Timed Automata can be done in PSPACE. Since Timed Automata easily encode DKSs, the upper bound follows. ☐

5.2 Model checking $TCTL^*$ over DKSs

Theorem 5.2.
1. Model checking $TCTL^*$ over DKSs (and ssDKSs) is EXPSPACE*-complete.*
2. Model checking $TCTL^*_{\leq,\geq}$ over DKSs (and ssDKSs) is PSPACE*-complete.*

Proof. A direct consequence of Theorem 5.1: the techniques from [EL87] produce an algorithm for $TCTL^*$ under the form of a simple polynomial-time labeling algorithm that calls an oracle for *TLTL* model checking. Hence model checking belongs to $\mathrm{P}^{\mathrm{EXPSPACE}}$, that is EXPSPACE. The same reasoning applies to $TCTL^*_{\leq,\geq}$ and yields a a $\mathrm{P}^{\mathrm{PSPACE}}$, that is a PSPACE algorithm. ☐

5.3 Model checking $TCTL^+$ over DKSs

CTL^+ is the extension of *CTL* in which *boolean combinations* of path formulae are allowed to appear under a path quantifier [Eme90]. For example, $\mathsf{A}(\mathsf{F}_{<3}\,\mathtt{req}_1 \Rightarrow \mathsf{F}_{<5}\,\mathtt{req}_2)$ is a $TCTL^+$ formula.

Theorem 5.3.
Model checking $TCTL^+$ and $TCTL^+_{\leq,\geq}$ over DKSs (and ssDKSs) is Δ^p_2-complete.

Proof. Δ^p_2-hardness comes from Δ^p_2-hardness of (untimed) CTL^+ model checking [LMS01]. Membership in Δ^p_2 is a consequence on Lemma 5.4, an extension of Lemma 4.5. ☐

Lemma 5.4. *Model checking formulae of the form* $\mathsf{E}\big(\bigwedge_i P_i \;\mathsf{U}_{\sim c_i}\; P'_i \wedge \bigwedge_j \neg(P_j \;\mathsf{U}_{\sim c_j} P'_j)\big)$ *over DKSs is* NP*-complete.*

Proof (Idea). Only membership in NP needs be proved. This is done by combining ideas from Lemmas 4.5 and 4.6, where we show how it is possible to witness the existence of a path through its Parikh image (and a few additional bits of information), and ideas from CTL^+ model checking, where a witness simply indicates in what order the different $P_i \mathsf{U} P'_i$ modalities are eventually satisfied along the path, with the choice of nodes that appear at these satisfaction points. ☐

Therefore, unlike the classical untimed case where model checking is harder for CTL^+ than for *CTL*, there is essentially no complexity cost for using $TCTL^+$ instead of *TCTL* over DKSs.

6 Conclusion

Figure 3 summarizes our results on the complexity of model checking quantitative temporal logics over DKSs. A general pattern is that exact durations make model checking harder. Polynomial-time model checking is possible if one considers *TCTL* and forbids exact durations or restricts to small-step DKSs.

		ssDKSs ($\xrightarrow{0/1}$)	tight DKSs ($\xrightarrow{n}$), DKSs ($\xrightarrow{\rho}$)
TCTL	$\leq, \geq$	PTIME-complete	PTIME-complete (Th. 4.2)
	$\leq, \geq, =$		Δ_2^p-complete (Th. 4.9)
TLTL	$\leq, \geq$	PSPACE-complete (Th. 5.1.*2*)	
	$\leq, \geq, =$	EXPSPACE-complete (Th. 5.1.*1*)	
$TCTL^*$	$\leq, \geq$	PSPACE-complete (Th. 5.2.*2*)	
	$\leq, \geq, =$	EXPSPACE-complete (Th. 5.2.*1*)	
$TCTL^+$	$\leq, \geq$	Δ_2^p-complete (Th. 4.9)	
	$\leq, \geq, =$		

Fig. 3. The complexity of model checking over DKSs

We see two main directions for future work. First, it appears that DKSs can be seen as Timed Automata where *only one clock is used* and *this clock is reset with every transition*. It would be interesting to see if our results generalize to Timed Automata with one clock but without the reset restriction.

The second direction is to investigate alternate semantics for DKSs. Our assumption that steps in a DKS are all $q \xrightarrow{d} q'$ for $d \in \rho$ has practical, not philosophical, motivations: it makes technicalities and notations simpler. But DKSs could be given alternate semantics.

For example, one could assume that an edge $q \xrightarrow{d} q'$ is a succinct description of d edges between q and q', visiting $d-1$ intermediary states. Here time flows more smoothly. The behavior is the same but the temporal logic can refer to more observation points (so that its meaning is modified). Another possibility is to consider that an interval $\rho = [n, n']$ in a transition $(q, \rho, q') \in R$ states that the system must wait at least n time units and at most n' *before making the choice to move from q to q'*. This impacts the branching behavior of S.

These two variant semantics may or may not be preferable. The complexity of model checking remains PTIME-complete for $TCTL_{\leq,\geq}$, but becomes PSPACE-complete for *TCTL* (proofs will appear in a full version of this paper).

References

[ACD93] R. Alur, C. Courcoubetis, and D. Dill. Model-checking in dense real-time. *Information and Computation*, 104(1):2–34, 1993.

[AFH96] R. Alur, T. Feder, and T. A. Henzinger. The benefits of relaxing punctuality. *Journal of the ACM*, 43(1):116–146, 1996.

[AH92] R. Alur and T. A. Henzinger. Logics and models of real time: A survey. In *Real-Time: Theory in Practice, Proc. REX Workshop, Mook, NL, June 1991*, volume 600 of *Lecture Notes in Computer Science*, pages 74–106. Springer, 1992.

[AH93] R. Alur and T. A. Henzinger. Real-time logics: Complexity and expressiveness. *Information and Computation*, 104(1):35–77, 1993.

[AH94] R. Alur and T. A. Henzinger. A really temporal logic. *Journal of the ACM*, 41(1):181–203, 1994.

[AL99] L. Aceto and F. Laroussinie. Is your model checker on time? In *Proc. 24th Int. Symp. Math. Found. Comp. Sci. (MFCS'99), Szklarska Poreba, Poland, Sep. 1999*, volume 1672 of *Lecture Notes in Computer Science*, pages 125–136. Springer, 1999.

[Alu91] R. Alur. *Techniques for Automatic Verification of Real-Time Systems*. PhD thesis, Stanford Univ., August 1991. Available as Tech. Report STAN-CS-91-1378.

[CGP99] E. M. Clarke, O. Grumberg, and D. A. Peled. *Model Checking*. MIT Press, 1999.

[CLR90] T. H. Cormen, C. E. Leiserson, and R. L. Rivest. *Introduction to algorithms*. MIT Press, 1990.

[CY92] C. Courcoubetis and M. Yannakakis. Minimum and maximum delay problems in real-time systems. *Formal Methods in System Design*, 1(4):385–415, 1992.

[EL87] E. A. Emerson and Chin-Laung Lei. Modalities for model checking: Branching time logic strikes back. *Science of Computer Programming*, 8(3):275–306, 1987.

[Eme90] E. A. Emerson. Temporal and modal logic. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science, vol. B*, chapter 16, pages 995–1072. Elsevier Science, 1990.

[EMSS92] E. A. Emerson, A. K. Mok, A. P. Sistla, and J. Srinivasan. Quantitative temporal reasoning. *Real-Time Systems*, 4(4):331–352, 1992.

[ET97] E. A. Emerson and R. J. Trefler. Generalized quantitative temporal reasoning: An automata-theoretic approach. In *Proc. 7th Int. Joint Conf. Theory and Practice of Software Development (TAPSOFT'97), Lille, France, Apr. 1997*, volume 1214 of *Lecture Notes in Computer Science*, pages 189–200. Springer, 1997.

[ET99] E. A. Emerson and R. J. Trefler. Parametric quantitative temporal reasoning. In *Proc. 14th IEEE Symp. Logic in Computer Science (LICS'99), Trento, Italy, July 1999*, pages 336–343. IEEE Comp. Soc. Press, 1999.

[GJ79] M. R. Garey and D. S. Johnson. *Computers and Intractability. A Guide to the Theory of NP-Completeness*. Freeman, 1979.

[Hen98] T. A. Henzinger. It's about time: real-time logics reviewed. In *Proc. 9th Int. Conf. Concurrency Theory (CONCUR'98), Nice, France, Sep. 1998*, volume 1466 of *Lecture Notes in Computer Science*, pages 439–454. Springer, 1998.

[Kre88] M. W. Krentel. The complexity of optimization problems. *Journal of Computer and System Sciences*, 36(3):490–509, 1988.

[Lew90] H. R. Lewis. A logic of concrete time intervals (extended abstract). In *Proc. 5th IEEE Symp. Logic in Computer Science (LICS'90), Philadelphia, PA, USA, June 1990*, pages 380–389. IEEE Comp. Soc. Press, 1990.

[LMS01] F. Laroussinie, N. Markey, and Ph. Schnoebelen. Model checking CTL^+ and $FCTL$ is hard. In *Proc. 4th Int. Conf. Foundations of Software Science and Computation Structures (FOSSACS'2001), Genova, Italy, Apr. 2001*, volume 2030 of *Lecture Notes in Computer Science*, pages 318–331. Springer, 2001.

[LST00] F. Laroussinie, Ph. Schnoebelen, and M. Turuani. On the expressivity and complexity of quantitative branching-time temporal logics. In *Proc. 4th Latin American Symposium on Theoretical Informatics (LATIN'2000), Punta del Este, Uruguay, Apr. 2000*, volume 1776 of *Lecture Notes in Computer Science*, pages 437–446. Springer, 2000.

[Ost90] J. S. Ostroff. Deciding properties of timed transition models. *IEEE Transactions on Parallel and Distributed Systems*, 1(2):170–183, 1990.

[Pap84] C. H. Papadimitriou. On the complexity of unique solutions. *Journal of the ACM*, 31(2):392–400, 1984.

[Pap94] C. H. Papadimitriou. *Computational Complexity.* Addison-Wesley, 1994.

[Sto76] L. J. Stockmeyer. The polynomial-time hierarchy. *Theoretical Computer Science*, 3(1):1–22, 1976.

[Wag87] K. W. Wagner. More complicated questions about maxima and minima, and some closures of NP. *Theoretical Computer Science*, 51(1–2):53–80, 1987.