Time-Progress Evaluation for Dense-Time Automata with Concave Path Conditions*

Farn Wang

Dept. of Electrical Engineering
Graduate Institute of Electronic Engineering
National Taiwan University
farn@cc.ee.ntu.edu.tw; http://cc.ee.ntu.edu.tw/~farn

RED 7.0 is available at http://sourceforge.net/projects/redlib

Abstract. The evaluation of successor or predecessor state spaces through time progress is a central component in the model-checking algorithm of dense-time automata. The definition of the time progress operator takes into consideration of the path condition of time progress and usually results in high complexity in the evaluation. Previous algorithms in this aspect usually assume that the original location invariance conditions of an automaton are convex in the dense-time state space. Based on this assumption, efficient algorithms for convex path conditions can be designed for reachability analysis. However, it is not clear whether the path conditions are still convex in the general setting of TCTL model-checking. In this work, we discuss the concept of time-convexity that allows us to relax the restrictions on the application of time-progress evaluation algorithm for convex path conditions. Then we give examples in TCTL model-checking that engenders time-concave path conditions even when the original automaton location invariance conditions are time-convex. Then we present two techniques that allow us to apply the evaluation algorithms for time-convex path conditions to time-concave path conditions. Finally, we report our experiment with the techniques. For some benchmarks, our techniques may enhance the performance of model-checking by an order of magnitude.

Keywords: timed automaton, time progress, model-checking, TCTL, convex, concave

1 Introduction

In the last two decades, we have witnessed significant progress in both theory and applications of the model-checking technology of dense-time systems [1,4,8,11]. One popular framework in this regard is called *TCTL model-checking* [1] which assumes a given dense-time system description as a *timed automaton* (*TA*) [3] and

* The work is partially supported by NSC, Taiwan, ROC under grants NSC 95-2221-E-002-067 and NSC 95-2221-E-002-072.

a given specification formula in *Timed Computation Tree Logic* (*TCTL*) [1] and checks whether the TA satisfies the TCTL formula. The TCTL model-checking technology could be an attractive choice to the industry as long as the performance of the related verification algorithms could handle industrial projects. However, at this moment, many algorithms used in TCTL model-checking still suffer from low performance. To achieve the promise of TCTL model-checking, the performance of related algorithms has to be enhanced.

One important algorithm in TCTL model-checking is the time-progress evaluation algorithm. For simplicity, we focus on the backward time-progress operation. However, the ideas discussed in this work should also apply to the forward counterpart. Usually we are given a *path condition* ϕ and a *destination condition* ψ and want to compute the condition, $\texttt{Tbck}(\phi, \psi)$ in symbols, of *those states that can go to a state satisfying* ψ *through a time progression along which all states satisfying* ϕ. For convenience, given $t \in \mathbb{R}^{\geq 0}$, we let $\phi + t$ be the condition for states that satsify ϕ after the progression of t time units [1]. Then $\texttt{Tbck}(\phi, \psi)$ can be formulated as follows [7].

$$\begin{aligned} \texttt{Tbck}(\phi, \psi) &\overset{\text{def}}{=} \exists t \in \mathbb{R}^{\geq 0} \left(\psi + t \wedge \forall t' \in \mathbb{R}^{\geq 0} \left(t' \leq t \rightarrow \phi + t'\right)\right) \\ &\equiv \exists t \in \mathbb{R}^{\geq 0} \left(\psi + t \wedge \neg \exists t' \in \mathbb{R}^{\geq 0} \left(t' \leq t \wedge \neg\phi + t'\right)\right) \end{aligned}. \tag{T}$$

The outer quantification on t specifies the "*through a time progression*" part. The inner quantification specifies that every state along the finite computation also satisfies ϕ. As can be seen, $\texttt{Tbck}(\phi, \psi)$ incurs two existential quantifications (or Fourier-Motzkin elimination [6]), two complementations, and two conjunctions. Since the time-progress algorithm is fundamental to TCTL model-checking, such an involved formulation usually results in significant performance degradation.

One way to enhance the evaluation efficiency of $\texttt{Tbck}()$ is to make an assumption of the TAs. An observation is that if the path condition ϕ characterizes a *convex*[1] state space, then $\texttt{Tbck}(\phi, \psi)$ can be rewritten as follows.

$$\texttt{Tbck}'(\phi, \psi) \overset{\text{def}}{=} \exists t \in \mathbb{R}^{\geq 0} \left(\psi + t \wedge \phi \wedge \phi + t\right) \tag{T'}$$

The reason is that for two states ν and ν', that respectively represent the starting state and the destination state of a time progression, we know that the following two conditions are true.

- Both ν and ν' are in the convex space characterized by ϕ.
- All states that happen during this time progress actually form a straight line segment between ν and ν'.

According to the definition of convexity, then all states in this straight line segment (and time progression) must also be in the space characterized by ϕ. As can be seen from $\texttt{Tbck}'()$, one existential quantification and two complementations can be avoided with this assumption. It will be interesting to see to what extent in TCTL model-checking [1,10], we can use $\texttt{Tbck}'()$ in place of $\texttt{Tbck}()$. According

[1] A space is *convex* if for any two points in the space, any point in the straight line segment between the two points is also in the space. A space that is not convex is *concave*.

to our knowledge, there is no related work in this regard. In this work, we have the following contributions.

- We propose the idea of *time-convexity* to relax the applicability of `Tbck`$'()$ to concave path conditions.[2]
- We show that if the location invariance conditions of a TA are all time-convex, then all path conditions used in the reachability analysis are also time-convex.
- We show that there are examples in TCTL model-checking [1] that entail the computation of time progress through *time-concave* path conditions even when all the location invariance conditions of the TA are time-convex.
- We present two techniques that allow us to apply `Tbck`$'()$ for the time progress evaluation through time-concave path conditions. For several benchmarks, the techniques have significantly enhanced the performance of our TCTL model-checker.

We have the following presentation plan. Section 2 reviews the background theory. Section 3 explains the concept of time-convexity. Section 4 investigates the possibilities of time-concave and time-convex path conditions in reachability analysis and model-checking. Sections 5 and 6 respectively present a technique for efficient time progress evaluation with time-concave path conditions. Section 7 reports our implementation and experiment. Section 8 is the conclusion.

2 TCTL model-checking problem

2.1 Timed automata

Let $\mathbb{N}$ be the set of non-negative integers, $\mathbb{Z}$ the set of all integers, and $\mathbb{R}^{\geq 0}$ the set of non-negative reals. Also 'iff' means "if and only if." Given a set Q of atomic propositions and a set X of clocks, a *location predicate* is a Boolean combination of atoms of the forms q and $x \sim c$, where $q \in Q$, $x \in X$, '$\sim$' is one of $\leq, <, =, >, \geq$, and $c \in \mathbb{N}$. The set of all location predicates of Q and X is denoted as $\mathcal{L}(Q, X)$.

Definition 1. **Timed automaton (TA)** A *TA* is a tuple $\langle Q, X, I, H, E, \sigma, \delta, \tau, \pi \rangle$ with the following restrictions. Q is a finite set of control locations. X is a finite set of clocks. $I \in \mathcal{L}(Q, X)$ is the initial condition. $H \in \mathcal{L}(Q, X)$ is the location invariance condition. $E \subseteq Q \times Q$ is a finite set of transition rules. $\sigma : E \mapsto Q$ and $\delta : E \mapsto Q$ respectively specify the source and the destination locations of each transition. $\tau : E \mapsto \mathcal{L}(\emptyset, X)$ defines the triggering condition of each rule execution. For each $e \in E$, $\pi(e) \subseteq X$ specifies the set of clocks to reset during the transition. ■

For convenience, given a TA $A = \langle Q, X, I, H, E, \sigma, \delta, \tau, \pi \rangle$, we use $Q_A, X_A, I_A, H_A, E_A, \sigma_A, \delta_A, \tau_A$, and π_A to denote $Q, X, I, H, E, \sigma, \delta, \tau$, and π respectively.

Example 1. We have the transition diagrams of an example TA A in figure 1. The ovals represent control locations q_0, q_1, and q_2. Location q_0 is the initial one.

[2] For convenience, we say a condition is convex iff the state space that it characterizes is convex. A non-convex condition is concave.

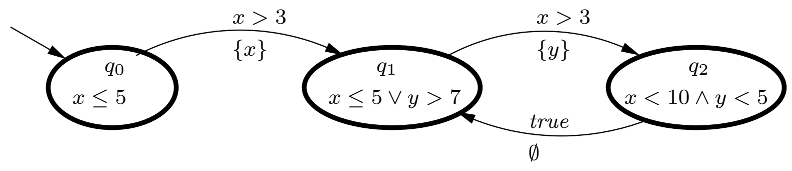

Fig. 1. An example TA

In each control location, we label the name and the constraint at that location. Thus the initial condition is $I_A \equiv q_0 \wedge x \leq 5$ and the location invariance condition is $H_A \equiv (q_0 \wedge x \leq 5) \vee (q_1 \wedge (x \leq 5 \vee y > 7)) \vee (q_2 \wedge x < 10 \wedge y < 5)$. The arcs represent transitions between locations. On each arc, we label the triggering condition and the clock reset set. ■

A *valuation* of a set Y *(domain)* is a mapping from Y to a *codomain.*

Definition 2. **States of a TA** A *clock valuation* of a TA A is a total valuation from X_A to $\mathbb{R}^{\geq 0}$. A *state* of A is a pair (q, ν) such that $q \in Q_A$ and ν is a clock valuation. Let V_A denote the set of states of A. ■

For any clock valuation ν of a TA A and $t \in \mathbb{R}^{\geq 0}$, $\nu + t$ is a valuation identical to ν except that for every $x \in X_A$, $(\nu + t)(x) = \nu(x) + t$. Given a set $X' \subseteq X_A$, we let $\nu X'$ be a valuation that is identical to ν except that all variables in X' are mapped to zero.

A state (q, ν) *satisfies* a location predicate η, in symbols $(q, \nu) \models \eta$, if η is evaluated true when q is interpreted true, all other location names are interpreted false, and all clock variables are interpreted according to ν. Given two states $(q, \nu), (q', \nu')$ and a transition e of a TA A, we say *A transits with e* from (q, ν) to (q', ν'), in symbols $(q, \nu) \xrightarrow{e} (q', \nu')$, if $\sigma_A(e) = q$, $\delta_A(e) = q'$, $(q, \nu) \models \tau_A(e) \wedge H_A$, $\nu \pi_A(e) = \nu'$, and $(q', \nu') \models H_A$.

Definition 3. **Runs** Given a TA A, a *run* of A is an infinite sequence of state-time pairs $((q_0, \nu_0), t_0)((q_1, \nu_1), t_1) \ldots ((q_k, \nu_k), t_k) \ldots\ldots$ with the following three restrictions. (1) $t_0 t_1 \ldots t_k \ldots\ldots$ is a monotonically, non-decreasing, and divergent real-number sequence. That is, $\forall k \geq 0 (t_{k+1} \geq t_k)$ and $\forall c \in \mathbb{N} \exists k > 1 (t_k > c)$. (2) For all $k \geq 0$, for all $t \in [0, t_{k+1} - t_k]$, $(q_k, \nu_k + t) \models H_A$. (3) For all $k \geq 0$, either $(q_k, \nu_k + t_{k+1} - t_k) = (q_{k+1}, \nu_{k+1})$ or there is an $e \in E_A$ such that $(q_k, \nu_k + t_{k+1} - t_k) \xrightarrow{e} (q_{k+1}, \nu_{k+1})$. The run is *initial* if $(q_0, \nu_0) \models I_A$. ■

2.2 Timed Computation Tree Logic (TCTL)

Given a set Q of atomic propositions, a set X of clocks, and a $b \in \mathbb{N}$, a *zone predicate* within bound b is a Boolean combination of atoms of the forms q and $x - y \sim c$, where $q \in Q$, $x, y \in X \cup \{0\}$, '$\sim$'$\in \{<, \leq, =, \neq, \geq, >\}$, and $c \in \mathbb{Z} \cap [-b, b]$. The set of all zone predicates of Q and X within bound b is denoted as $\mathcal{Z}_b(Q, X)$. The satisfaction of zone predicates by a state is defined similarly as that of location predicates.

TCTL is a language for the specification of timing behaviors with branching structure [1]. A TCTL formula ϕ is of the following syntax.

$$\phi ::= \eta \mid \phi_1 \vee \phi_2 \mid \neg\phi_1 \mid \exists\Box_{\sim c}\phi_1 \mid \exists\phi_1\mathcal{U}_{\sim c}\phi_2$$

Here η is a zone predicate in $\mathcal{Z}_\infty(Q,X)$[3], $\sim \in \{<,\leq,=,\neq,\geq,>\}$, and c is a non-negative integer constant. For modal formulas $\exists\Box_{\sim c}\phi_1$ and $\exists\phi_1\mathcal{U}_{\sim c}\phi_2$, ϕ_1 is called the *path condition* while ϕ_2 is called the *destination condition*. Standard shorthands like *true*, *false*, $\phi_1 \wedge \phi_2$, $\phi_1 \rightarrow \phi_2$, $\exists\Diamond_{\sim c}\phi_1$, $\forall\Box_{\sim c}\phi_1$, $\forall\Diamond_{\sim c}\phi_1$, and $\forall\phi_1\mathcal{U}_{\sim c}\phi_2$ are also adopted.

Note that, unlike the original definition of TCTL [1], we allow inequalities in $\mathcal{Z}_\infty(Q,X)$ to appear in TCTL formulas. The reason is that in the evaluation of nested modal formulas, the evaluation of inner modal formulas may yield predicates in $\mathcal{Z}_\infty(Q,X)$ anyway. Thus, in the general context of TCTL model-checking, it makes no difference to have zone predicates in TCTL formulas.

Given a state (q,ν) of a TA A and a TCTL formula ϕ, we use the notation $A,(q,\nu) \models \phi$ to mean that state (q,ν) *satisfies* ϕ in A. The definition of satisfaction of zone (location) predicates and Boolean formulas are straightforward. Those of satisfaction of the modal formulas are as follows.

- $A,(q,\nu) \models \exists\Box_{\sim c}\phi_1$ iff there is a run from (q,ν) such that for all states (q',ν') that is t time units from (q,ν) in the run with $t \sim c$, $A,(q',\nu') \models \phi_1$.
- $A,(q,\nu) \models \exists\phi_1\mathcal{U}_{\sim c}\phi_2$ iff there is a run from (q,ν) such that
  - there is a state (q',ν') that is t time units from (q,ν) in the run with $t \sim c$ and $A,(q',\nu') \models \phi_2$; and
  - for all states (q'',ν'') before (q',ν') in the run, $A,(q'',\nu'') \models \phi_1$.

The TCTL *model-checking problem* asks if all initial states of a TA satisfy a TCTL formula in the TA. Given a TA A and a TCTL formula ϕ, we use $[\![\phi]\!]_A$ to denote the state space characterized by ϕ in A. It is easy to see that for any state (q,ν), $(q,\nu) \in [\![\phi]\!]_A$ iff $A,(q,\nu) \models \phi$.

For convenience, given a condition η for a set of destination state and a transition e, we let $\mathtt{Xbck}_e(\eta)$ be the condition for states that can directly go to states in η through transition e. Formally speaking, $(q,\nu) \models \mathtt{Xbck}_e(\eta)$ iff there exists a $(q',\nu') \models \eta$ and $(q,\nu) \xrightarrow{e} (q',\nu')$. According to [7, 10], the formulation for the evaluation of a formula like $\exists\Box_{\sim c}\phi_1$ can be represented as follows.

$$\exists z \left(z = 0 \wedge \mathbf{gfp} Z \left(\bigvee_{e\in E_A} \mathtt{Tbck}\left(z \sim c \rightarrow (\phi_1 \wedge H_A), \mathtt{Xbck}_e(Z)\right)\right)\right) \qquad (\exists\Box)$$

Here **gfp** is the greatest fixpoint operator. The evaluation of the greatest fixpoint operator works by iteratively eliminating states from Z until we find that there is no more elimination possible. z is an auxiliary clock variable not used in ϕ_1, I_A, H_A, and the transitions of A. As can be seen, formula $z \sim c \rightarrow (\phi_1 \wedge H_A)$ appears in formula (T) as a path condition.

Also, the formulation for the evaluation of a formula like $\exists\phi_1\mathcal{U}_{\sim c}\phi_2$ can be represented as follows.

$$\exists z \left(z = 0 \wedge \mathbf{lfp} Z. \left((\phi_2 \wedge z \sim c \wedge H_A) \vee \bigvee_{e\in E} \mathtt{Tbck}(\phi_1 \wedge H_A, \mathtt{Xbck}_e(Z))\right)\right) \quad (\exists\mathcal{U})$$

[3] We abuse the notation $[-\infty,\infty]$ for $(-\infty,\infty)$.

Here **lfp** is the least fixpoint operator. It characterizes the space of those states that can reach states satisfying ϕ_2 through a run segment along which all states satisfy ϕ_1. The evaluation of the least fixpoint operator works by iteratively adding states to Z until we find that there is no more addition possible. z is an auxiliary clock variable not used in ϕ_1, ϕ_2, I_A, H_A, and the transitions of A. As can be seen, formula $\phi_1 \wedge H_A$ appears in formula (T) as a path condition.

3 Zones, convexity, and time-convexity

Given a TA A and a TCTL formula ϕ, we let C_A^ϕ be the biggest timing constant used in A and ϕ. A *clock zone* of A and ϕ is a set of clock valuations characterizable with a conjunctive[4] zone-predicate in $\mathcal{Z}_{C_A^\phi}(\emptyset, X_A)$. A clock zone is a convex space of clock valuations. Without loss of generality, we assume that the given characterization zone predicate for a clock zone is always *tight*. That is, for every inequality $x - y \sim c$ in the characterization zone predicate, we cannot change the value of c without changing the members of the corresponding clock zone. Such a tight zone predicate for a clock zone can be obtained with an all-pair shortest-path algorithm with cubic time complexity [5].

A *zone* of A and ϕ is a set of states in V_A characterizable with a conjunctive zone-predicate like $q \wedge \eta$ with a $q \in Q_A$ and $\eta \in \mathcal{Z}_{C_A^\phi}(\emptyset, X_A)$. The states in a zone share the same control location. According to [7], the state spaces of A that we need to manipulate in model-checking for ϕ are finite unions of zones. Such a union can be characterized with zone predicates in $\mathcal{Z}_{C_A^\phi}(Q_A, X_A \cup \{z\})$ where z is an auxiliary clock variable not used in A [7,10]. Many model-checkers for TAs are based on symbolic manipulation algorithms of *zone predicates* represented in various forms [4,8,11].

For convenience, we may also represent a zone as a pair like (q, η) with $q \in Q_A$ and $\eta \in \mathcal{Z}_{C_A^\phi}(\emptyset, X_A \cup \{z\})$. A set S of zones is *convex* if for each $q \in Q_A$, $\bigcup_{(q,\eta)\in S} [\![\eta]\!]_A$ is convex. If S is not convex, it is *concave*. The reachable state space of a TA is usually concave. Most state-spaces that we need to manipulate in TCTL model-checking are likely concave. For convenience, we say a formula ϕ is convex iff $[\![\phi]\!]_A$ is convex. If ϕ is not convex, then it is concave.

Example 2. The initial condition I_A of the TA in example 1 is convex while the location invariance condition H_A is concave. Specifically, the following subformula $\dot{H} \equiv q_1 \wedge (x \leq 5 \vee y > 7)$ is concave. For example, we may have two states (q_1, ν_1) and (q_1, ν_2) with $\nu_1(x) = \nu_1(y) = 4$ and $\nu_2(x) = \nu_2(y) = 8$. It is clear that $(q_1, \nu_1) \in [\![\dot{H}]\!]_A$ and $(q_1, \nu_2) \in [\![\dot{H}]\!]_A$. However, the middle point, say $(q_1, \nu_{3/2})$, between (q_1, ν_1) and (q_1, ν_2) with $\nu_{3/2}(x) = \nu_{3/2}(y) = 6$ is not in $[\![\dot{H}]\!]_A$.

Concavity may also happen with difference constraints between two clocks. For example, the following zone predicate $\ddot{H} \equiv q_1 \wedge (x - y < 3 \vee x - y > 7)$ is also concave. For example, we may have two states (q_1, ν_3) and (q_1, ν_4) with

[4] A conjunctive predicate does not have negation and disjunction in it.

$\nu_3(x) = 9$, $\nu_3(y) = 1$, $\nu_4(x) = 1$, and $\nu_4(y) = 9$. It is clear that (q_1, ν_3) and (q_1, ν_4) are both in $[\![\ddot{H}]\!]_A$. However the middle point, say $(q_1, \nu_{7/2})$, between (q_1, ν_1) and (q_1, ν_2) with $\nu_{7/2}(x) = \nu_{7/2}(y) = 5$ is not in $[\![\ddot{H}]\!]_A$. ■

It is known that procedure $\texttt{Tbck}'()$ for time progress evaluation can be applied to convex path conditions.

Example 3. In example 1, $\texttt{Tbck}'()$ is not applicable to H_A which is concave. For example, $\texttt{Tbck}(H_A, q_1 \wedge x = 10 \wedge y = 10)$ is $q_1 \wedge x > 7 \wedge y > 7 \wedge x \leq 10 \wedge y \leq 10 \wedge x - y = 0$. However, $\texttt{Tbck}'(H_A, q_1 \wedge x = 10 \wedge y = 10)$ is $q_1 \wedge x \geq 0 \wedge y \geq 0 \wedge x \leq 10 \wedge y \leq 10 \wedge x - y = 0$ which is incorrect. ■

Here we relax the restriction of the applicability of $\texttt{Tbck}'()$ with the following concept.

Definition 4. **Time-convexity** A union U of clock zones is *time-convex* iff for any $\nu \in U$ and $t \in \mathbb{R}^{\geq 0}$ with $\nu + t \in U$, then for any $t' \in [0, t]$, $\nu + t' \in U$. Otherwise, it is called *time-concave*. A set S of zones is time-convex if for each $q \in Q_A$, $\bigcup_{(q,\eta)\in S} [\![\eta]\!]_A$ is time-convex; it is time-concave else. ■

Example 4. In examples 1 and 2, I_A is time-convex while H_A is time-concave. Moreover, zone predicate $\ddot{H} \equiv q_1 \wedge (x - y < 3 \vee x - y > 7)$ is concave. But we have the following derivation for any state (q, ν) and real $t \in \mathbb{R}^{\geq 0}$.

$$\begin{array}{l}
(q_1, \nu) \models q_1 \wedge (x - y < 3 \vee x - y > 7) \\
\equiv (q_1, \nu + t) \models q_1 \wedge (x + t - y - t < 3 \vee x + t - y - t > 7) \\
\equiv (q_1, \nu + t) \models q_1 \wedge (x - y < 3 \vee x - y > 7)
\end{array}$$

Thus it is clear that $q_1 \wedge (x - y < 3 \vee x - y > 7)$ is time-convex. ■

Lemma 1. *Given a TA A, a time-convex path zone predicate ϕ, and a destination zone predicate ψ, $[\![\texttt{Tbck}(\phi, \psi)]\!]_A = [\![\texttt{Tbck}'(\phi, \psi)]\!]_A$.*

Proof : We can prove this lemma in two directions. First, we want to prove that $\texttt{Tbck}(\phi, \psi) \subseteq \texttt{Tbck}'(\phi, \psi)$. Given a state $(q, \nu) \models \texttt{Tbck}(\phi, \psi)$, we have the following derivation.

$$\begin{array}{l}
(q, \nu) \models \texttt{Tbck}(\phi, \psi) \\
\equiv (q, \nu) \models \exists t \in \mathbb{R}^{\geq 0} \left(\psi + t \wedge \forall t' \in \mathbb{R}^{\geq 0} (t' \leq t \rightarrow \phi + t')\right) \\
\Rightarrow (q, \nu) \models \exists t \in \mathbb{R}^{\geq 0} \left(\begin{array}{c} \psi + t \wedge \phi \wedge \phi + t \\ \wedge \forall t' \in \mathbb{R}^{\geq 0} (t' \leq t \rightarrow \phi + t') \end{array}\right) \text{, instantiating } t' \text{ with } t \\
\Rightarrow (q, \nu) \models \exists t \in \mathbb{R}^{\geq 0} (\psi + t \wedge \phi \wedge \phi + t) \text{, restriction relaxation.} \\
\equiv (q, \nu) \models \texttt{Tbck}'(\phi, \psi) \text{, defnition}
\end{array}$$

Now we prove $\texttt{Tbck}'(\phi, \psi) \subseteq \texttt{Tbck}(\phi, \psi)$ with the following derivation.

$$\begin{array}{l}
(q, \nu) \models \texttt{Tbck}'(\phi, \psi) \\
\equiv (q, \nu) \models \exists t \in \mathbb{R}^{\geq 0} (\psi + t \wedge \phi \wedge \phi + t) \\
\Rightarrow (q, \nu) \models \exists t \in \mathbb{R}^{\geq 0} \left(\begin{array}{c} \psi + t \wedge \phi \wedge \phi + t \\ \wedge \forall t' \in \mathbb{R}^{\geq 0} (t' \leq t \rightarrow \phi + t') \end{array}\right) \text{, since } \phi \text{ is time-convex} \\
\equiv (q, \nu) \models \exists t \in \mathbb{R}^{\geq 0} \left(\psi + t \wedge \forall t' \in \mathbb{R}^{\geq 0} (t' \leq t \rightarrow \phi + t')\right) \text{, since } t \leq t \wedge 0 \leq t \\
\equiv (q, \nu) \models \texttt{Tbck}(\phi, \psi) \text{, defnition}
\end{array}$$

With the proof for the two directions, we know the lemma is correct. ■

Lemma 1 implies that we can also apply the more efficient `Tbck`$'()$ to concave but time-convex path conditions.

Example 5. In example 2, `Tbck`$'()$ was not thought to be applicable to path zone predicate $\ddot{H}$ either. But now, `Tbck`$(\ddot{H}, q_1 \wedge x = 8 \wedge y = 8)$ and `Tbck`$'(\ddot{H}, q_1 \wedge x = 8 \wedge y = 8)$ both evaluate to $q_1 \wedge x \geq 0 \wedge y \geq 0 \wedge x \leq 8 \wedge y \leq 8 \wedge x - y = 0$. ■

.

4 Time-concavity and convexity in verification problems

In this section, we show two things for TCTL model-checking. First, time-convexity of the location invariance condition H_A is good enough to guarantee the time-convexity of all path conditions used in the reachability analysis of A. Second, time-convexity of H_A is not good enough to guarantee the time-convexity of all path conditions in the TCTL model-checking of A.

4.1 For reachability analysis

The most used verification framework is *reachability analysis*. In this framework, we are given a TA A and a safety predicate η and want to check whether there is an initial run of A along which some state satisfies $\neg\eta$. The system is *safe* iff A satisfies $\forall\Box\eta \equiv \neg\exists true\mathcal{U}\neg\eta$. According to formula $(\exists\mathcal{U})$ in page 5, we find that all the path conditions used in the time progress evaluation is exactly the H_A of a TA A. Thus we have the following lemma.

Lemma 2. *For reachability analysis of a TA A, if H_A is time-convex, then* `Tbck`$'()$ *can be used in place of* `Tbck`$()$ *in formula $(\exists\mathcal{U})$ without affecting the result of analysis.* ■

4.2 For TCTL model-checking

We have identified some generic cases in examples 6 through 10 that can cause time-concave path conditions in model-checking.

Example 6. **Disjunction in the path conditions in modal formulas.** We may have a formula: $\exists(q_1 \wedge (x \leq 5 \vee y > 7))\mathcal{U}q_2$. Given a TA A with a time-convex H_A, according to formula $(\exists\mathcal{U})$ in page 5, the path condition is $q_1 \wedge (x \leq 5 \vee y > 7) \wedge H_A$. As can be checked, the path condition is time-concave when $[\![q_1 \wedge x \leq 5 \wedge H_A]\!]_A \neq \emptyset$, $[\![q_1 \wedge y > 7 \wedge H_A]\!]_A \neq \emptyset$, and $[\![q_1 \wedge x > 5 \wedge y \leq 7 \wedge H_A]\!]_A \neq \emptyset$.

For another example, according to formulation $(\exists\Box)$ in page 5, formula $\exists\Box(q_1 \wedge (x \leq 5 \vee y > 7))$ also incurs time-concavity in path condition. ■

Example 7. **Complementation in the path conditions in modal formulas.** We have a formula: $\forall\Diamond(q_1 \rightarrow (x > 5 \wedge y \leq 7))$ which can be rewritten as $\neg\exists\Box(q_1 \wedge (x \leq 5 \vee y > 7))$. According to formulation $(\exists\Box)$ in page 5, the path

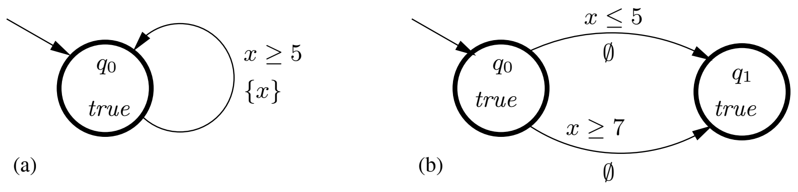

Fig. 2. Another example TA

condition $q_1 \wedge (x \leq 5 \vee y > 7) \wedge H_A$ is time-concave when $[\![q_1 \wedge x \leq 5 \wedge H_A]\!]_A \neq \emptyset$, $[\![q_1 \wedge y > 7 \wedge H_A]\!]_A \neq \emptyset$, and $[\![q_1 \wedge x > 5 \wedge y \leq 7 \wedge H_A]\!]_A \neq \emptyset$. ■

Note that in TCTL model-checking, we usually need to calculate the complement of time-convex state spaces and end up with time-concave state spaces.

Example 8. **Timing constraints with $\exists\Box$-formulas.** According to formula ($\exists\Box$) in page 5, formula: $\exists\Box_{\leq 7} x \leq 5$ incurs a path condition $z \leq 7 \rightarrow x \leq 5 \equiv z > 7 \vee x \leq 5$. Then following the argument in examples 2 and 4, it is easy to see that this path condition is also time-concave. ■

The following two examples show that path condition concavity may also happen due to the structures of TAs.

Example 9. **Time-concavities due to TA structures.** We may have the example TA A in figure 2(a) and want to check $A, (q_0, \nu) \models \exists\Box\exists x > 3\mathcal{U} x \leq 0$ with $\nu(x) = 0$. Note that the location invariance condition is time-convex. $(q_0, \nu) \in [\![\exists x > 3\mathcal{U} x \leq 0]\!]_A$ since $x \leq 0$ is immediately fulfilled at ν. Also $(q_0, \nu + 4) \in [\![\exists x > 3\mathcal{U} x \leq 0]\!]_A$ with the firing of the transition at $\nu + 5$. But it is clear that $(q_0, \nu + 1) \notin [\![\exists x > 3\mathcal{U} x \leq 0]\!]_A$. ■

According to the original definition of TCTL [1], only propositions may appear as atoms. Thus we may argue that the above-mentioned formulas in examples 6 to 9 may not happen in the original TCTL definition. The following example is interesting in this regard.

Example 10. **Nested $\exists\mathcal{U}$-formulas with modal timing constraints.** Now we may want to check the TA in figure 2(b) for a formula $\exists\Box\exists q_0 \mathcal{U}_{<1} q_1$ at a state (q_0, ν) with $\nu(x) = 5$. Then $(q_0, \nu) \in [\![\exists q_0 \mathcal{U}_{<1} q_1]\!]_A$ and $(q_0, \nu+2) \in [\![\exists q_0 \mathcal{U}_{<1} q_1]\!]_A$. However, it is clear that $(q_0, \nu + 1) \notin [\![\exists q_0 \mathcal{U}_{<1} q_1]\!]_A$. ■

5 Algorithm with cascading convexities

We have experimented with several techniques for performance enhancement of time progress evaluation for time-concave path conditions. We present one such technique that we have found useful. The technique breaks a time-concave zone predicate into time-convex ones and then applies `Tbck′()` on each time-convex ones for the evaluation of time progress.

Given a zone predicate ϕ that describes a time-concave state space, we want to characterizes those states in $[\![\phi]\!]_A$ that can first go through a state outside $[\![\phi]\!]_A$ through time progression and then again end up at a state in $[\![\phi]\!]_A$. To calculate the time progress operation to such states in $[\![\phi]\!]_A$, we cannot use $\texttt{Tbck}'()$ in place of $\texttt{Tbck}()$ since such states are evidence for the time-concavity of ϕ. Formally speaking, such states can be characterized as follows.

$$\mathit{TConcave}(\phi) \stackrel{\text{def}}{=} \phi \wedge \exists t \in \mathbb{R}^{\geq 0} \left(\phi + t \wedge \exists t' \in \mathbb{R}^{\geq 0}(t' < t \wedge (\neg\phi) + t')\right)$$

We have the following lemma that establishes some properties of the characterizaton useful for our performance-enhancing techniques.

Lemma 3. *Given a zone predicate ϕ for a TA A, $[\![\phi]\!]_A - [\![\mathrm{TConcave}(\phi)]\!]_A$ is time-convex.*

Proof : This is straightforward from the definition of $\mathit{TConcave}(\phi)$. ■

Given two state spaces S and S', we say S is *time-connected* to S' if there is a state $(q,\nu) \in S$ and a $t \in \mathbb{R}^{\geq 0}$ such that $(q, \nu + t) \in S'$ and for every $t' \in [0,t]$, $(q, \nu+t') \in S \cup S'$. If S is not *time-connected* to S', then it is *time-disconnected* to S'. The concept of time-connectivity is important for the correctness of piecewise evaluation of time progress.

Lemma 4. *Suppose we are given two zone predicates ϕ and ϕ' such that $[\![\phi]\!]_A$ is not time-connected to $[\![\phi']\!]_A$ and vice versa. Then for every zone predicate ψ, $[\![\texttt{Tbck}(\phi,\psi)]\!]_A \cup [\![\texttt{Tbck}(\phi',\psi)]\!]_A = [\![\texttt{Tbck}(\phi \vee \phi',\psi)]\!]_A$.*

Proof : It is easy to see that $[\![\texttt{Tbck}(\phi,\psi)]\!]_A \cup [\![\texttt{Tbck}(\phi',\psi)]\!]_A \subseteq [\![\texttt{Tbck}(\phi \vee \phi',\psi)]\!]_A$. On the other hand, $[\![\texttt{Tbck}(\phi,\psi)]\!]_A \cup [\![\texttt{Tbck}(\phi',\psi)]\!]_A \supseteq [\![\texttt{Tbck}(\phi \vee \phi',\psi)]\!]_A$ is false if either of the following two cases are true.

- There is a state $(q,\nu) \in [\![\phi]\!]_A$ and a $t \in \mathbb{R}^{\geq 0}$ such that $(q, \nu+t) \in [\![\phi']\!]_A$ and for every $t' \in [0,t]$, $(q, \nu+t') \in [\![\phi]\!]_A \cup [\![\phi']\!]_A$. But this exactly violates the assumption that $[\![\phi]\!]_A$ is not time-connected to $[\![\phi']\!]_A$.
- There is a state $(q,\nu) \in [\![\phi']\!]_A$ and a $t \in \mathbb{R}^{\geq 0}$ such that $(q, \nu+t) \in [\![\phi]\!]_A$ and for every $t' \in [0,t]$, $(q, \nu+t') \in [\![\phi]\!]_A \cup [\![\phi']\!]_A$. This is symmetric to the first case and violates the assumption that $[\![\phi']\!]_A$ is not time-connected to $[\![\phi]\!]_A$.

Since both of these cases are false, we know the lemma is proven. ■

Lemma 5. *Given a TA A and a zone predicate ϕ, $[\![\phi]\!]_A - [\![\mathrm{TConcave}(\phi)]\!]_A$ is not time-connected to $[\![\mathrm{TConcave}(\phi)]\!]_A$ and vice versa.*

Proof : We first assume that $[\![\phi]\!]_A - [\![\mathit{TConcave}(\phi)]\!]_A$ is time-connected to $[\![\mathit{TConcave}(\phi)]\!]_A$. This implies that there is a state $(q,\nu) \in [\![\phi]\!]_A - [\![\mathit{TConcave}(\phi)]\!]_A$ and a $t \in \mathbb{R}^{\geq 0}$ such that $(q,\nu+t) \in [\![\mathit{TConcave}(\phi)]\!]_A$ and for each $t' \in [0,t]$, $(q, \nu+t') \in [\![\phi]\!]_A$. According to the definition of $\mathit{TConcave}(\phi)$, there is a $t'' \in \mathbb{R}^{\geq 0}$ such that $(q, \nu+t+t'') \in [\![\phi]\!]_A$ while there is a $\bar{t}$ such that $t < \bar{t} < t+t''$ and $(q, \nu+\bar{t}) \notin [\![\phi]\!]_A$. This contradicts our assumption that $(q,\nu) \in [\![\phi]\!]_A - [\![\mathit{TConcave}(\phi)]\!]_A$.

We then assume that $[\![\mathit{TConcave}(\phi)]\!]_A$ is time-connected to $[\![\phi]\!]_A - [\![\mathit{TConcave}(\phi)]\!]_A$. This implies that there is a state $(q,\nu) \in [\![\mathit{TConcave}(\phi)]\!]_A$ and a $t \in \mathbb{R}^{\geq 0}$ such that $(q, \nu+t) \in [\![\phi]\!]_A - [\![\mathit{TConcave}(\phi)]\!]_A$ and for each $t' \in [0,t]$, $(q, \nu+t') \in [\![\phi]\!]_A$. According to the definition of $\mathit{TConcave}(\phi)$, there is a $t'' \in \mathbb{R}^{\geq 0}$ such that $(q, \nu + t'') \in [\![\phi]\!]_A$ while there is a $\bar{t}$ such that $0 < \bar{t} < t''$ and $(q, \nu + \bar{t}) \notin [\![\phi]\!]_A$. With the assumption on t, we know that $0 \leq t < \bar{t} < t''$. This implies that $(q, (\nu + t) + \bar{t} - t) \notin [\![\phi]\!]_A$ and $(q, (\nu + t) + t'' - t) \in [\![\phi]\!]_A$. This contradicts the assumption that $(q, \nu + t) \in [\![\phi]\!]_A - [\![TConcave(\phi)]\!]_A$. ■

Based on lemmas 3, 4, and 5, we present the following procedure that breaks a zone predicate ϕ into a finite set of zone predicates such that for each two state predicates ϕ_1, ϕ_2 in the set, $[\![\phi_1]\!]_A$ is not time-connected to $[\![\phi_2]\!]_A$.

```
CascadingConvexities($\phi$) /* $\phi$ is a zone predicate for a TA $A$. */ {
    Let $\Phi := \emptyset$.
    While $[\![\phi]\!]_A \neq \emptyset$, { $\eta := TConcave(\phi)$; $\Phi := \Phi \cup \{\phi \wedge \neg\eta\}$; $\phi := \eta$. }
    return $\Phi$.
}
```

Note that this procedure terminates since the number of zone sets is finite. Thus we repeatedly remove some zones from $[\![\phi]\!]_A$ and eventually reduce it to $\emptyset$. According to lemma 3, we know that every zone predicate in CascadingConvexities(ϕ) characterizes a time-convex state space. Moreover, with lemma 3, 4, and 5, we can establish the following lemma for a new formulation of backward time progress evaluation.

Lemma 6. *For every two zone predicates ϕ and ψ,* $[\![\texttt{Tbck}(\phi, \psi)]\!]_A = \bigcup_{\eta \in \texttt{CascadingConvexities}(\phi)} \texttt{Tbck}'(\eta, \psi)$. ■

6 Algorithm with approximate time-concavity checking

As can be seen in section 5, procedure $TConcave()$ can be executed many times in procdeure CascadingConvexities() and incur great computation cost. We want to investigate if it may pay off to use an alternative technique that avoids the evaluation of procedure $TConcave()$. Given a path zone predicate ϕ and a destination zone predicate ψ, this alternative technique works in the following two steps.

(1) Partition ϕ into two zone predicates ϕ_1 and ϕ_2 such that $[\![\phi_1]\!]_A \cap [\![\phi_1]\!]_A = \emptyset$, $[\![\phi_1]\!]_A \cup [\![\phi_1]\!]_A = [\![\phi]\!]_A$, $[\![\phi_1]\!]_A$ is time-convex, and $[\![\phi_1]\!]_A$ and $[\![\phi_2]\!]_A$ are time-disconnected to each other.
(2) Then we return $\texttt{Tbck}'(\phi_1, \psi) \vee \texttt{Tbck}(\phi_2, \psi)$ as the result of time progress evaluation.

The performance of the technique relies on the efficiency in carrying out step (1). We have the following lemma that helps us carrying out step (1). First, we need some notations for the convenience of discussion. Given a zone predicate ϕ, we assume that we can construct a set $ZoneSet(\phi)$ with zone elements of the form (q, η) such that $[\![\phi]\!]_A = [\![\bigvee_{(q,\eta) \in ZoneSet(\phi)} q \wedge \eta]\!]_A$. Depending on the implementation of ϕ, there are various ways to do this. If ϕ is implemented with DBMs [5], then ϕ should already have been represented as $ZoneSet(\phi)$ with η

represented as a DBM for each $(q,\eta) \in \mathit{ZoneSet}(\phi)$. If ϕ is implemented as a CRD (Clock-Restriction Diagram) [8], then each $(q,\eta) \in \mathit{ZoneSet}(\phi)$ corresponds to a path in the CRD.

Given a $q \in Q$, we let $\mathit{ZoneSet}_q(\phi) = \{(q,\eta) \mid (q,\eta) \in \mathit{ZoneSet}(\phi)\}$. Also, given a set Φ of zone predicates and two clocks $x, y \in X_A \cup \{0\}$, we let $\mathit{UB}_{x-y}(\Phi)$ be the minimum upper-bound for expression $x-y$ in all states $(q,\nu) \in [\![\bigvee_{(q,\eta)\in\Phi} q \wedge \eta]\!]_A$. That is, $\mathit{UB}_{x-y}(\Phi) = \min\{u \mid (q,\nu) \in [\![\bigvee_{(q,\eta)\in\Phi} q \wedge \eta]\!]_A, \nu(x) - \nu(y) \leq u\}$.[5] If $\mathit{UB}_{x-y}(\Phi)$ does not exist, then we denote $\mathit{UB}_{x-y}(\Phi) = \infty$.

Moreover, we define a predicate $\mathit{ConcavityNecessary}_\phi()$ of two zone representations that share the same control location. Specifically, given two such zone representations (q,η) and (q,η'), $\mathit{ConcavityNecessary}_\phi((q,\eta),(q,\eta'))$ is true if and only if there are two clocks $x, y \in X_A \cup \{z\}$ such that

- $\mathit{UB}_{x-0}(\{(q,\eta)\}) < \mathit{UB}_{x-0}(\mathit{ZoneSet}_q(\phi))$;
- $\mathit{UB}_{0-y}(\{(q,\eta')\}) < \mathit{UB}_{0-y}(\mathit{ZoneSet}_q(\phi))$; and
- for all $z, w \in X_A \cup \{z\}$, $\mathit{UB}_{z-w}(\{(q,\eta)\}) + \mathit{UB}_{w-z}(\{(q,\eta')\}) \geq 0$.

Lemma 7. *Given a zone predicate ϕ, if ϕ is time-concave, then there are (q,η), $(q,\eta') \in \mathrm{ZoneSet}(\phi)$ satisfying $\mathrm{ConcavityNecessary}_\phi((q,\eta),(q,\eta'))$.*

Proof : We assume there is a state (q,ν) and two reals $t' < t \in \mathbb{R}^{\geq 0}$ such that $(q,\nu), (q,\nu+t) \in [\![\phi]\!]_A$ while $(q,\nu+t') \notin [\![\phi]\!]_A$. Assume that there are $(q,\eta_1), (q,\eta_2) \in \mathit{ZoneSet}(\phi)$ such that $(q,\nu) \in [\![q \wedge \eta_1]\!]_A$ and $(q,\nu+t) \in [\![q \wedge \eta_2]\!]_A$. Note that for all clocks $z, w \in X_A$, $\nu(z) - \nu(w) = (\nu+t)(z) - (\nu+t)(w) = (\nu+t')(z) - (\nu+t')(w)$. This implies that there is a clock $x \in X_A$ such that $\mathit{UB}_{x-0}(\{(q,\eta_1)\}) < (\nu+t')(x) < (\nu+t)(x) \leq \mathit{UB}_{x-0}(\{(q,\eta_2)\}) \leq \mathit{UB}_{x-0}(\mathit{ZoneSet}_q(\phi))$. This implies that $\mathit{UB}_{x-0}(\{(q,\eta_1)\}) < \mathit{UB}_{x-0}(\mathit{ZoneSet}_q(\phi))$. This means that the first bullet is correct. Similarly, the second bullet is correct.

Note that for all clocks $z, w \in X_A$, $\nu(z)-\nu(w) = (\nu+t)(z)-(\nu+t)(w)$ which implies that $0 = \nu(z)-\nu(w)+(\nu+t)(w)-(\nu+t)(z)$. According to the definition of $\mathit{UB}_{z-w}()$ and $\mathit{UB}_{w-z}()$, we know that $\nu(z) - \nu(w) \leq \mathit{UB}_{z-w}(\{(q,\eta_1)\})$ and $(\nu+t)(w) - (\nu(z)+t) \leq \mathit{UB}_{w-z}(\{(q,\eta_2)\})$. Thus the third bullet is also proven for (q,η_1) and (q,η_2).

By letting η and η' be η_1 and η_2 respectively, the lemma is proven. ■

Based on lemma 7, for step (1) in the above, we let

$$\phi_2 = \bigvee_{(q,\eta),(q,\eta')\in \mathit{ZoneSet}(\phi),\, \mathit{ConcavityNecessary}_\phi((q,\eta),(q,\eta'))} (q \wedge \eta)$$

and $\phi_1 = \phi \wedge \neg\phi_2$. We have the following lemma that shows this is indeed what we can use in step (1).

Lemma 8. *With the ϕ_1 and ϕ_2 defined in the last paragraph, for any ψ, $[\![\mathtt{Tbck}(\phi,\psi)]\!]_A = [\![\mathtt{Tbck}'(\phi_1,\psi) \vee \mathtt{Tbck}(\phi_2,\psi)]\!]_A$.*

Proof : The lemma is true if ϕ_1 is time-convex and ϕ_1, ϕ_2 are time-disconnected to each other. If ϕ_1 is time-concave, then according to lemma 7, there are $(q,\eta), (q,\eta') \in \mathit{ZoneSet}(\phi_1)$ such that $\mathit{ConcavityNecessary}_\phi((q,\eta),(q,\eta'))$ is true. Then no states in $[\![(q\wedge\eta) \vee (q\wedge\eta')]\!]_A$ should be in $[\![\phi_1]\!]_A$. This is a contradiction.

[5] For convenience, we let $\nu(0) = 0$.

If ϕ_1 is time-connected to ϕ_2, then there are $(q, \eta) \in \mathit{ZoneSet}(\phi_1)$ and $(q, \eta'), (q, \eta'') \in \mathit{ZoneSet}(\phi_2)$ such that $[\![q \wedge \eta]\!]_A$ is time-connected to $[\![q \wedge \eta']\!]_A$ and $\mathit{ConcavityNecessary}_\phi((q, \eta'), (q, \eta''))$ is true. This implies $\mathit{ConcavityNecessary}_\phi((q, \eta), (q, \eta''))$ which is also a contradiction.

With an argument similar to the one in the last paragraph, we can also prove that ϕ_2 is not time-connected to ϕ_1. Thus the lemma is proven. ■

Finally, the technique has been realized with a symbolic manipulation algorithm for zone predicates represented with CRD.

7 Implementation and experiments

We have implemented our ideas in sections 5 and 6 in **RED** 7.0, a model-checker for TAs and a parametric safety analyzer for LHAs (linear hybrid automata) [2] based on CRD and HRD (Hybrid-Restriction Diagram) technology [8, 9]. We used the following two parameterized benchmarks from the literature.

1. *Fischer's timed mutual exclusion algorithm* [8]: The algorithm relies on a global lock and a local clock per process to control access to the critical section. Three timing constants used are 10, 19, and 30. The first formula that we check is the following.
$$\forall\Box\neg(\exists(\mathtt{critical}_1)\mathcal{U}((\neg\mathtt{critical}_1) \wedge \exists\Diamond_{<19}\mathtt{critical}_1)) \quad \text{(C)}$$
This formula intends to say that if process one leaves the critical section, then it cannot enter the critical section again in 19 time units. However, since the $\exists\mathcal{U}$-formula can be satisfied at a state that directly fulfills $(\neg\mathtt{critical}_1) \wedge \exists\Diamond_{<19}\mathtt{critical}_1$, the formula is not satisfied.
The second formula is the following.
$$\forall\Box\left(\begin{array}{l}\mathtt{critical}_1 \rightarrow \\ \exists\Diamond\exists\mathtt{idle}_1\mathcal{U}\exists\mathtt{ready}_1\mathcal{U}_{<10}\exists\mathtt{waiting}_1\mathcal{U}_{>19}\exists\mathtt{critical}_1\mathcal{U}\mathtt{idle}_1\end{array}\right) \quad \text{(D)}$$
It says that if process 1 is in the critical section, then it can go through an expected mode sequence with timing restrictions back to the idle mode.
The third formula is the following.
$$\forall\Box\left(\mathtt{ready}_1 \rightarrow \forall\Diamond_{<10}\left(\mathtt{waiting}_1 \wedge \forall\Box\left(\mathtt{critical}_1 \rightarrow \forall\Diamond_{<30}\mathtt{idle}_1\right)\right)\right) \quad \text{(E)}$$
The formula says that if process 1 is in the ready mode, it enters the waiting mode in 10 time units and from that point on, if it enters the critical section, it returns to the idle mode in 30 time units. Note that the formula is not satisfied with Zeno computations[6]. So we have to use option '-Z' for quantification for non-Zeno computations.
2. *CSMA/CD* [11]: This is the Ethernet bus arbitration protocol with collision-and-retry. The timing constants used are 26, 52, and 808. The first property that we want to check is the following.
$$\forall\Box((\mathtt{transm}_1 \wedge x_1 = 52) \rightarrow \forall\Box_{<756}\neg\mathtt{transm}_2) \quad \text{(F)}$$
It says that if sender 1 is in the transmission mode for 52 time units, then in all computations, sender 2 cannot be in the transmission mode for at least 756 time units.
The second formula is as follows.

[6] A Zeno computation runs forever without time converging to a finite value.

Table 1. Performance data of scalability w.r.t. various strategies

	TCTL		`Tbck`()		*Cascading*		*ATCC*		
benchmarks	spec's	m	time	mem	time	mem	time	mem	answer
Fischer's mutual exclusion (m processes)	(C)	3	0.024s	186k	0.004s	186k	0.012s	3.1M	violated
		4	0.42s	32M	0.044s	413k	0.044s	13M	
		5	8.38s	162M	1.80s	69M	1.83s	71M	
	(D)	3	0.008s	186k	0.008s	186k	0.012s	2.7M	satisfied
		4	0.060s	413k	0.044s	413k	0.036s	9.9M	
		5	4.18s	116M	0.588s	34M	0.536s	40M	
	(E)	3	0.036s	186k	0.032s	186k	0.032s	6.5M	satisfied with non-Zenoness
		4	1.06s	59M	0.048s	413k	0.128s	25k	
		5	26.9s	344M	5.36s	113M	4.25s	116M	
CSMA/CD (1 bus+m senders)	(F)	2	0.024s	168k	0.048s	168k	0.028s	2.5M	satisfied
		3	0.072s	333k	0.068s	333k	0.116s	20M	
		4	5.49s	100M	4.38s	90M	4.72s	109M	
	(G)	2	0.028s	168k	0.028s	168k	0.020s	1.2M	violated
		3	0.104s	333k	0.092s	333k	0.088s	4.1M	
		2	0.096s	168k	0.068s	168k	0.096s	166k	violated with non-Zenoess
		3	108s	662M	93.2s	584M	96.7s	605M	
	(H)	2	0.040s	168k	0.032s	168k	0.076s	1.6M	violated
		3	0.100s	333k	0.084s	333k	0.100s	5.6M	
		2	0.13s	168k	0.092s	168k	0.152s	24M	satisfied with non-Zenoess
		3	266s	1199M	214s	1036M	222s	1082M	

data collected on a Pentium 4 1.7GHz with 2G memory running LINUX;
s: seconds; k: kilobytes of memory in diagram data-structure;
M: megabytes of memory in diagram data-structure

$$\forall\Box\neg(\exists\mathtt{transm}_1\mathcal{U}(\mathtt{transm}_2 \wedge \exists\Box_{<26}\neg\mathtt{retry}_1)) \qquad \text{(G)}$$

It intends to say that it is not possible that sender 1 remains in the transmission mode until sender 2 also does so and sender 1 does not enter the retry mode in 26 time units. This formula is not satisfied since the $\exists\mathcal{U}$-formula can be satisfied immediately with a state that directly fulfills $\mathtt{transm}_2 \wedge \exists\Box_{<26}\neg\mathtt{retry}_1$. From that state on, there is a computation along which sender 2 stays in the transmission mode for 808 time units.
The final formula is the following.

$$\forall\Box\,(\mathtt{bus\_collision} \rightarrow \forall\,(\forall\Diamond_{<52}\,(\mathtt{wait}_1 \vee \mathtt{retry}_1))\,\mathcal{U}_{<52}\mathtt{bus\_idle}) \qquad \text{(H)}$$

Note that the formula is not satisfied with Zeno computations. So we have to use option '-Z' for quantification for non-Zeno computations.

We have collected data respectively with three tool configurations.

- `Tbck`() represents the one that uses `Tbck`() for all time progress evaluation.
- *Cascading* represents the one with the technique in section 5.
- *ATCC* represents the one with the technique in section 6.

The performance data is reported in table 1. The CPU time used, the total memory consumption for the data-structures in state-space representations, and the answers of model-checking are reported. As can be seen, our technique in section 5 always performs better than `Tbck`(). For some benchmark, the performance enhancement is one order of magnitude. This shows that our techniques could be useful in applying TCTL model-checking technology to industrial projects.

The technique in section 6 did not perform as well as we expected. Further investigation revealed that special arrangement for garbage CRD nodes might have slowed down the hash table operations and blown up the memory consumption. In our present implementation, garbage collection cannot be invoked inside the procedures for the technique. In the future, we may gain more performance with an implementation of a more powerful garbage collector.

8 Concluding remarks

In this work, we discuss how to improve the performance of an important component algorithm, the time progress evaluation algorithm, for the model-checking of TAs. Techniques in section 6 may worth further investiagtion for better precision in the approximation and more efficient algorithms.

References

1. R. Alur, C. Courcoubetis, D.L. Dill. Model Checking for Real-Time Systems, IEEE LICS, 1990.
2. R. Alur, C.Courcoubetis, T.A. Henzinger, P.-H. Ho. Hybrid Automata: an Algorithmic Approach to the Specification and Verification of Hybrid Systems. Workshop on Theory of Hybrid Systems, LNCS 736, Springer-Verlag, 1993.
3. R. Alur, D.L. Dill. A Theory of Timed Automata. Theoretical Computer Science, 126:183-235, 1994.
4. J. Bengtsson, K. Larsen, F. Larsson, P. Pettersson, Wang Yi. UPPAAL - a Tool Suite for Automatic Verification of Real-Time Systems. Hybrid Control System Symposium, 1996, LNCS, Springer-Verlag.
5. D.L. Dill. Timing Assumptions and Verification of Finite-state Concurrent Systems. CAV'89, LNCS 407, Springer-Verlag.
6. J.B. Fourier. (reported in:) Analyse des travaux de l'Académie Royale des Sciences pendant l'année 1824, Partie Mathématique, 1827.
7. T.A. Henzinger, X. Nicollin, J. Sifakis, S. Yovine. Symbolic Model Checking for Real-Time Systems, IEEE LICS 1992.
8. F. Wang. Efficient Verification of Timed Automata with BDD-like Data-Structures, STTT (Software Tools for Technology Transfer), Vol. 6, Nr. 1, June 2004, Springer-Verlag; special issue for the 4th VMCAI, Jan. 2003, LNCS 2575, Springer-Verlag.
9. F. Wang. Symbolic Parametric Safety Analysis of Linear Hybrid Systems with BDD-like Data-Structures. IEEE Transactions on Software Engineering, Volume 31, Issue 1 (January 2005), pp. 38-51, IEEE Computer Society. A preliminary version is in proceedings of 16th CAV, 2004, LNCS 3114, Springer-Verlag.
10. F. Wang, G.-D. Huang, F. Yu. TCTL Inevitability Analysis of Dense-Time Systems: From Theory to Engineering. IEEE Transactions on Software Engineering, Vol. 32, Nr. 7, July 2006, IEEE Computer Society.
11. S. Yovine. Kronos: A Verification Tool for Real-Time Systems. International Journal of Software Tools for Technology Transfer, Vol. 1, Nr. 1/2, October 1997.