Efficient Model-Checking of Dense-Time Systems with Time-Convexity Analysis *

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

Proof of some lemmas can be found in the appendices.

Abstract

The evaluation of successor or predecessor state spaces through time progress is a central component in the model-checking algorithms of dense-time automata. The time progress operator takes the concavity of a path condition into consideration and usually results in high complexity in the evaluation. Previous algorithms in this aspect usually assume that the location invariance condition of an automaton are convex in the dense-time state space and use a more efficient algorithm for time progress evaluation. In fact, the restriction of location invariance condition convexity can be further relaxed to that of time-convexity for a broader range of application of the more efficient algorithm. In this work, we present techniques for the efficient model-checking of dense-time automata by taking the time-convexity of path conditions into consideration. We first identify a class of TCTL formulas that only characterize time-convex state spaces. The class includes several important types of TCTL formulas, including some timed inevitabilities (of the form like $\forall\Diamond_{[0,d]}\phi$) with deadlines. We then present a new formulation for the efficient evaluation of timed inevitabilities with time-convex path conditions. The new formulation also leads to a new technique for the approximate evaluation of timed inevitabilities with better precision. Finally, we report our implementation and experiment.

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

1 Introduction

A popular framework for the verification of embedded systems [1, 4, 8, 13] is the TCTL model-checking problem [1]. In this framework, we are given a dense-time system description as a *timed automaton* (*TA*) [3] and a specification formula in *Timed Computation Tree Logic* (*TCTL*) [1] and checks whether the TA satisfies the TCTL formula. To achieve the promise of TCTL model-checking, the importance of performance enhancement of related algorithms cannot be over-emphasized. 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, $\mathtt{Tbck}(\phi,\psi)$ in symbols, of *those states that can go to a state satisfying* ψ *through a time progression along which all states satisfy* ϕ. For convenience, given $t \in \mathbb{R}^{\geq 0}$ (the set of non-negative reals), we let $\phi + t$ be the condition for states that satsify ϕ after the progression of t time units [1]. Then $\mathtt{Tbck}(\phi,\psi)$ can be formulated as in table 1 [7]. The outer quantification on t specifies the "*through a time progression of* t *time units*" part. The inner quantification specifies that every state along the time progression satisfies ϕ. As can be seen, this formulation (T) of $\mathtt{Tbck}(\phi,\psi)$ incurs two existential quantifications [6], one complementation, 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 formulation (T) is to take the shape of the path condition ϕ into consideration. An observation is that if the path condition ϕ characterizes a *convex*[1] state space, then formulation (T)

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

[1] A space is *convex* if for any two points in the space, any point in

names	formulas	formulations
(T)	$\texttt{Tbck}(\phi,\psi)$	$\exists t\in\mathbb{R}^{\geq 0}\left(\psi+t\wedge\forall t'\in\mathbb{R}^{\geq 0}\left(t'\leq t\rightarrow\phi+t'\right)\right)$ $\equiv\quad\exists t\in\mathbb{R}^{\geq 0}\left(\psi+t\wedge\neg\exists t'\in\mathbb{R}^{\geq 0}\left(t'\leq t\wedge\neg\phi+t'\right)\right)$
(T')	$\texttt{Tbck}'(\phi,\psi)$	$\exists t\in\mathbb{R}^{\geq 0}\left(\psi+t\wedge\phi\wedge\phi+t\right)$

Table 1. Formulations (T) and (T') of time progress evaluation.

can be rewritten as formulation (T') in table 1. The reason is that for two states ν and ν', that respectively represent the starting state and the destination state of a time progression along path condition ϕ, 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' in the state space.

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.

Example 1 *Suppose* we are in a state space of two clocks x and y of readings in $\mathbb{R}^{\geq 0}$. For a state with $x=3$ and $y=3$, we may use the pair $(x=3,y=3)$ to represent the state. We have a path condition $\phi\equiv x\leq 5\vee y>7$. The condition is concave since states $(x=3,y=3)$ and $(x=9,y=9)$ both satisfy it but their middle point $(x=6,y=6)$ does not. Intuitively, there is a gap between $(x=5,y=5)$ and $(x=7,y=7)$ that cannot be stepped into according to ϕ. With the formulations in table 1, $\texttt{Tbck}(\phi,x=8\wedge y=8)$ is $7<x\leq 8\wedge 7<y\leq 8\wedge x=y$. However, $\texttt{Tbck}'(\phi,x=8\wedge y=8)$ is $0\leq x\leq 8\wedge 0\leq y\leq 8\wedge x=y$ which extends across the gap and is incorrect. ■

As can be seen from the new formula (T'), one existential quantification and one complementation can be avoided with convex path conditions. It will be interesting to see to what extent in TCTL model-checking [1, 12], we can use formulation (T') in place of (T) for better model-checking performance. Specifically, one important class of TCTL formulas, called *timed inevitabilities*, are of the form $\forall\Diamond_{\langle c,d\rangle}\phi$, where $\langle c,d\rangle$ is an interval in $\mathbb{R}^{\geq 0}$, and are important in specifying that some good behavior should happen within a deadline.

Example 2 *We* may want to specify that after a fire is detected, an alarm is signaled in 5 to 10 time units. In TCTL, such a property can be written as follows.

$$\forall\Box\left(\texttt{fire}\rightarrow\forall\Diamond_{[5,10]}\texttt{alarm}\right)$$

Here propositions `fire` and `alarm` respectively specify that fire is detected and that alarm is on. This property is usually evaluated as the following equivalent formula: $\neg\exists\mathit{true}\bigcup\left(\texttt{fire}\wedge\exists\Box_{[5,10]}\neg\texttt{alarm}\right)$.

However, the traditional formulation [7, 12] for the evaluation of such a timed inevitability is like $\neg\exists\mathit{true}\bigcup(\texttt{fire}\wedge\exists m\exists\Box(m<5\vee m>10\vee\neg\texttt{alarm}))$, where m is a clock variable not used in the model. It is obvious that such a formulation creates a concave path condition. [2] ■

Thus it would be interesting to see whether we can avoid the time progress evaluation along concave path conditions in TCTL model-checking. In [11], the concept of *time-convexity* of path conditions was discussed to relax the applicability of the more efficient formulation (T') to concave path conditions. In this work, we have the following contributions.

- We identify a class, called $\texttt{TCTL}^{tc}$, of TCTL formulas that only characterize time-convex state spaces. The syntax structures of formulas in this class can be relatively efficiently checked. $\texttt{TCTL}^{tc}$ itself may not be general enough for writing full specifications. But its design purpose is to help us efficiently identifying those subformulas in a full specification that can induce efficient time progress evaluations.
- We then propose an adaptive algorithm for the time progress evaluation. The algorithm uses several techniques, including off-line analysis to recognize the $\texttt{TCTL}^{tc}$ path conditions in a full TCTL specification, in order to avoid time-concavity checking and to avoid evaluation with formulation (T).
- We propose a new formulation for the evaluation of timed inevitabilities. The new formulation breaks a time progress path into at most three run segments and allows us to take the convexity of each segment into consideration for efficient time progress evaluation.
- We extend the just-mentioned new formulation for a new formulation for the approximate evaluation of

the straight line segment between the two points is also in the space. A space that is not convex is *concave*. For convenience, we say a condition is convex iff the state space that it characterizes is convex. A non-convex condition is concave.

[2] Suppose we use a pair (q,ν) to represent a state in this system. Here q is a location name and ν is a valuation of m. Suppose we are given three valuations ν_1,ν_2,ν_3 with $\nu_1(m)=0,\nu_2(m)=6$, and $\nu_3(m)=12$. It is true that $(\texttt{alarm},\nu_2)$ is the middle point between $(\texttt{alarm},\nu_1)$ and $(\texttt{alarm},\nu_3)$. The path condition is concave since it is true that $(\texttt{alarm},\nu_1)$ and $(\texttt{alarm},\nu_3)$ both satisfy $m<5\vee m>10\vee\neg\texttt{alarm}$ while $(\texttt{alarm},\nu_2)$ does not.

timed inevitabilities. This new approximate formulation offers a better precision than the previous approach [12] in both theory and experiment.

- Finally, we have implemented our techniques and report our experiment with our TA model-checker RED, version 7.0. The result shows significant performance enhancement against many timed inevitabilities.

We have the following presentation plan. Section 2 is for related work. Section 3 defines TAs and the TCTL model-checking problem and reviews a symbolic algorithm for the TCTL model-checking problem. Section 4 explains how to use the concept of time-convexity for the efficient evaluation of time progress. Section 5 introduces TCTL^{tc}. Section 6 presents our new formulations for evaluating timed inevitabilities. Section 7 reports our implementation and experiment. Section 8 is the conclusion.

2 Related work

In [7], formulation (T) was proposed for the calculation of time progress precondition for TAs through concave path conditions. Various tools for reachability analysis are now available with formulation (T') based on the convexity assumption of the location invariance condition of TAs [4, 8, 13].

In [12], performance-enhancing techniques for timed inevitabilities were presented. They also presented an early decision technique for the evaluation of timed inevitabilities. They also discussed how to pick the time length value for the efficient evaluation of states that start non-Zeno runs.

In [9], a model-checking algorithm for timed inevitabilities with event constraints and weak/strong fairness assumptions was presented and implemented.

In [11], the concept of time-convexity was discussed.

3 TA and TCTL model-checking

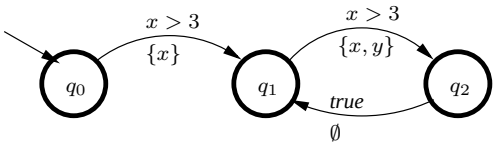
3.1 Timed automata

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. Given two sets X and Y, $X \subsetneq Y$ means $X \subset Y$ and $X \neq Y$. For convenience, we may write a constraint like $c \leq A \leq d$ as a shorthand for $c \leq A \wedge A \leq d$. 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 3 **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, σ_A, δ_A, τ_A, and π_A to denote Q, X, I, H, E, σ, δ, τ, and π respectively.



$$
\begin{array}{lcl}
I_A & \equiv & q_0 \wedge x \leq 5 \\
H_A & \equiv & (q_0 \wedge x \leq 5) \\
 & \vee & (q_1 \wedge (x \leq 5 \vee y > 7)) \\
 & \vee & (q_2 \wedge x < 10 \wedge y < 5)
\end{array}
$$

Figure 1. An example TA

Example 4 *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. The initial location is q_0. 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 5 **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 of A. 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 $e \in E_A$, 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 6 **Runs** Given a TA A, a *run* of A is an infinite sequence of state-time pairs

$((q_0,\nu_0),t_0)((q_1,\nu_1),t_1)\ldots((q_k,\nu_k),t_k)\ldots\ldots$ such that for all $k \geq 0$, the following three restrictions hold. (1) $t_k \leq t_{k+1}$. (2) For all $t \in [0, t_{k+1}-t_k]$, $(q_k,\nu_k+t) \models H_A$. (3) 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 $(q_0,\nu_0) \models I_A$. It is *Zeno* if there is a $c \in \mathbb{N}$ such that for every $\forall k \geq 0 (t_k \leq c)$. ■

3.2 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 can be defined similarly as that of location predicates.

TCTL (*Timed Computation Tree Logic*) is a language for the specification of timing behaviors with branching structures [1]. Here we adopt TCTL formulas, say ϕ, with the following extended syntax.

$$\phi ::= \eta \mid \phi_1 \vee \phi_2 \mid \neg\phi_1 \mid \exists\Box_{\langle c,d\rangle}\phi_1 \mid \exists\phi_1 \bigcup_{\langle c,d\rangle}\phi_2$$

Here η is a zone predicate in $\mathcal{Z}_\infty(Q, X)$[3], '$\langle$' $\in \{[, (\}$, and '$\rangle$' $\in \{],)\}$. $c \in \mathbb{N}$, $d \in \mathbb{N} \cup \{\infty\}$, $c \leq d$, and $d = \infty \longrightarrow$ '$\rangle$' $=$ ')'. Standard shorthands like *true*, *false*, $\phi_1 \wedge \phi_2$, $\phi_1 \rightarrow \phi_2$, $\exists\Diamond_{\langle c,d\rangle}\phi_1$, $\forall\Box_{\langle c,d\rangle}\phi_1$, $\forall\Diamond_{\langle c,d\rangle}\phi_1$, and $\forall\phi_1 \bigcup_{\langle c,d\rangle}\phi_2$ are also adopted. Also interval $[0, \infty)$ an be conveniently omitted.

For modal formulas $\exists\Box_{\langle c,d\rangle}\phi$ and $\exists\phi \bigcup_{\langle c,d\rangle}\psi$, ϕ is called the *path condition* while ψ is called the *destination condition*.

Given a TA A and a TCTL formula ϕ, we let C_A^ϕ be the biggest timing constant used in A and ϕ. Note that our TCTL definition is a little extended from [1]. First, we allow intervals instead of inequalities as the subscripts to modal operators. Computationally, this does not affect much in the related algorithms in model-checking. Second, unlike the original definition in [1], we allow inequalities in $\mathcal{Z}_\infty(Q, X)$ to appear in formulas. The reason is that according to [7], in the evaluation of nested modal formulas, the evaluation of inner modal formulas may yield predicates in $\mathcal{Z}_{C_A^\phi}(Q_A, X_A \cup \{m, z\})$ where m, z are two auxiliary clock variables not used in A. Thus, for the investigation of concave path conditions in time progress evaluation, 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 satisfaction of zone (location) predicates and Boolean formulas in A are defined straightforwardly. The satisfaction of the modal formulas is defined as follows.

- $A,(q,\nu) \models \exists\Box_{\langle c,d\rangle}\phi$ iff there is a non-Zeno run from (q,ν) such that for all states (q',ν') that is t time units from (q,ν) in the run with $t \in \langle c, d\rangle$, $A,(q',\nu') \models \phi$.
- $A,(q,\nu) \models \exists\phi \bigcup_{\langle c,d\rangle}\psi$ iff there is a non-Zeno run from (q,ν) such that
  - there is a state (q',ν') that is t time units from (q,ν) in the run with $t \in \langle c, d\rangle$ and $A,(q',\nu') \models \psi$; and
  - for all states (q'',ν'') before (q',ν') in the run, $A,(q'',\nu'') \models \phi$.

For a detailed definition, please check [1]. The TCTL *model-checking problem* instance of a TA A and a TCTL formula ϕ asks if all initial states of A satisfy ϕ.

We also use $\langle\!\langle\phi\rangle\!\rangle_A$ to denote the set of states $(q,\nu) \in V_A$ with $A,(q,\nu) \models \phi$. It is clear that for any $(q,\nu) \in V_A$, $A,(q,\nu) \models \phi$ if and only if $(q,\nu) \in \langle\!\langle\phi\rangle\!\rangle_A$.

3.3 Symbolic TCTL model-checking

A *zone* of a proposition set Q and a clock set X within a bound b is a set of states characterizable with a conjunctive zone-predicate like $q \wedge \eta$ with a $q \in Q$ and $\eta \in \mathcal{Z}_b(\emptyset, X)$. The states in a zone share the same control location. A zone is a convex space of states [5, 7]. Without loss of generality, we assume that the given characterization zone predicate for a non-empty zone is always *tight*. That is, for every inequality $x - y \sim c$ in the characterization zone predicate, we cannot lower the value of c without changing the members of the corresponding zone. Such a tight zone predicate for a zone can be obtained with an all-pair shortest-path algorithm with cubic time complexity [5, 8].

According to [7, 12], the state spaces of A that we need to manipulate in model-checking for ϕ are finite unions of zones characterizable with zone predicates in $\mathcal{Z}_{C_A^\phi}(Q_A, X_A \cup \{m, z\})$. Specifically, m and z are clock variables used respectively for the evaluation of timing constraints of modal formulas and the non-Zeno requirement of runs. Many model-checkers for TAs are based on symbolic manipulation algorithms of *zone predicates* represented in various forms [4, 8, 13].

For convenience, given a formula ϕ and a set $X = \{x_1, \ldots, x_n\}$ of variables, we use $\exists X(\phi)$ as the shorthand for $\exists x_1 \ldots \exists x_n(\phi)$. In table 2, we list the formulations for the symbolic evaluation algorithm of TCTL formulas from the literature [7, 12]. Given a TA A and a TCTL formula ϕ, the symbolic algorithm inductively constructs zone predicates that characterize states satisfying subformulas of ϕ. We use $[\![\phi]\!]_A$ to denote the zone predicate obtained from the algorithm in tables 1 and 2 for a TCTL formula ϕ.

In table 2, given a condition ψ for a set of destination state and a transition e, $\texttt{Xbck}_e(\psi)$ denotes the condition

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

names	formulas	non-Zeno runs only?	formulations
(q)	q	N/A	q
$(x-y\sim c)$	$x-y\sim c$	N/A	$x-y\sim c$
(Xbck)	$\texttt{Xbck}_e(\psi)$	N/A	$\sigma_A(e)\wedge H_A\wedge(\bigwedge_{q\in Q_A,q\neq\sigma_A(e)}\neg q)\wedge\tau_A(e)$ $\wedge\exists\pi_A(e)\cup\{\sigma_A(e),\delta_A(e)\}(\llbracket\psi\rrbracket_A\wedge\delta_A(e)\wedge H_A\wedge\bigwedge_{x\in\pi_A(e)}x=0)$
$(\vee)$	$\phi\vee\psi$	N/A	$\llbracket\phi\rrbracket_A\vee\llbracket\psi\rrbracket_A$
$(\neg)$	$\neg\phi$	N/A	$\neg\llbracket\phi\rrbracket_A$
$(\exists)$	$\exists x(\phi)$	N/A	$\exists x(\llbracket\phi\rrbracket_A)$
$(\rightsquigarrow)$	$\phi\rightsquigarrow\psi$	no	$\textbf{lfp}Z.\,(\llbracket\psi\rrbracket_A\vee\bigvee_{e\in E}\texttt{Tbck}(\llbracket\phi\rrbracket_A,\texttt{Xbck}_e(Z)))$
(NZ)	$\mathrm{NZ}(\phi)$	yes	$\textbf{gfp}Z.\,(\exists z\,(z=0\wedge(\llbracket\phi\rrbracket_A\rightsquigarrow(Z\wedge z\geq k))))$, with $k\geq 1$.
$(\exists\bigcup^{NZ})$	$\exists\phi\bigcup_{\langle c,d\rangle}\psi$	yes	$\exists m\,(m=0\wedge(\llbracket\phi\rrbracket_A\rightsquigarrow(m\in\langle c,d\rangle\wedge\llbracket\psi\rrbracket_A\wedge\mathrm{NZ}(\textit{true}))))$
$(\exists\Box^{NZ})$	$\exists\Box_{\langle c,d\rangle}\phi$	yes	$\exists m\,(m=0\wedge\mathrm{NZ}(m\in\langle c,d\rangle\rightarrow\llbracket\phi\rrbracket_A))$

N/A: not applicable.

Table 2. Traditional formulation for the symbolic evaluation of TCTL formulas

for states that can directly go to states in $\langle\!\langle\psi\rangle\!\rangle_A$ through transition e. Formally speaking, $(q,\nu)\models\texttt{Xbck}_e(\psi)$ iff there exists a $(q',\nu')\models\psi$ with $(q,\nu)\xrightarrow{e}(q',\nu')$.

Also in table 2, formulation $(\rightsquigarrow)$ calculates the backward reachability to states in $\langle\!\langle\psi\rangle\!\rangle_A$ through a path along which all states are in $\langle\!\langle\phi\rangle\!\rangle_A$. **lfp** is the least fixpoint operator. The evaluation of the least fixpoint operator works by iteratively adding states to Z until we find that there is no more addition possible. **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.

Formulation (NZ), with $k\geq 1$, characterizes those states that starts a non-Zeno run along which all states satisfy ϕ. Parameter k can be any chosen integer no less than one. In [12], it was reported that $k=\max(1,C_A^\phi)$ usually yields reasonably good performance.

To check a TA A against a TCTL formula ϕ, usually we check if $I_A\wedge\llbracket\neg\phi\rrbracket_A$ is satisfiable. A satisfies ϕ if and only if $I_A\wedge\llbracket\neg\phi\rrbracket_A$ is unsatisfiable.

According to [7, 12], we have the following lemma.

Lemma 7 *Given a TA A and a TCTL formula ϕ, for any state $(q,\nu)\in V_A$, $(q,\nu)\models\llbracket\phi\rrbracket_A$ iff $A,(q,\nu)\models\phi$.* ■

It is known that formulation $(\exists\Box^{NZ})$ is needed for the correct evaluation of timed inevitabilities so that the inevitabilities are not refuted by Zeno runs [12]. But for many verification tasks, the formulation (NZ) nested inside $(\exists\Box^{NZ})$ is expensive to carry out. It involves a double fixpoint calculation, i.e., a least fixpoint for formulation $(\rightsquigarrow)$ nested inside a greatest fixpoint. According to [12], the formulations in table 3 can be used for the approximate evaluation of formulations $(\exists\bigcup^{NZ})$ and $(\exists\Box^{NZ})$ if computation resources is limited. The approximation skips the evaluation with formulation (NZ) and directly uses H_A instead. Given a TA A and a TCTL formula ϕ, we use $\llbracket\phi\rrbracket_A^{app}$ to denote the zone predicate obtained for ϕ from the algorithm in tables 1 and 2 except that formulations $(\exists\Box)$ and $(\exists\bigcup)$ in table 3 are respectively used in place of formulations $(\exists\Box^{NZ})$ and $(\exists\bigcup^{NZ})$ in table 2. The following lemma shows that this approximation is strictly safe for the evaluation of timed inevitabilities.

Lemma 8 *Given a TA A, a zone predicate η, and an interval $\langle c,d\rangle\subseteq\mathbb{R}^{\geq 0}$, $\langle\!\langle\llbracket\forall\Diamond_{\langle c,d\rangle}\eta\rrbracket_A^{app}\rangle\!\rangle_A\subsetneq\langle\!\langle\forall\Diamond_{\langle c,d\rangle}\eta\rangle\!\rangle_A$.* ■

Note that in example 2, the inner modal formula, i.e.,

$$\exists\Box_{[5,10]}\neg\texttt{alarm}, \tag{A}$$

cannot be correctly evaluated with formulation $(\exists\Box)$ in table 3 which does not take non-Zenoness into consideration. For example, the property can be violated with a Zeno run that does not progress more than 5 time units after `fire` is true. Thus, to correctly check such properties, we cannot use formulation $(\exists\Box)$ in table 3. Later in section 6, we shall present a new approximate formulation that with enough precision to correctly evaluate many timed inevitabilities.

4 Time-convexity

In this section, we first review the concept of time-convexity [11]. Given a TA A and a space S of states with $S\subseteq V_A$, S is *convex* if for each $q\in Q_A$, $\{(q,\nu)\mid(q,\nu)\in S\}$ is convex. A state space that is not convex 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. We say a TCTL formula ϕ is convex iff $\langle\!\langle\phi\rangle\!\rangle_A$ is convex. If ϕ is not convex, then it is concave.

names	formulas	non-Zeno runs only?	formulations
($\exists\bigcup$)	$\exists\phi\bigcup_{\langle c,d\rangle}\psi$	no	$\exists m\,(m=0\wedge([\![\phi]\!]_A \rightsquigarrow (m\in\langle c,d\rangle\wedge[\![\psi]\!]_A)))$
($\exists\Box$)	$\exists\Box_{\langle c,d\rangle}\phi$	no	$\exists m\,(m=0\wedge\mathbf{gfp}Z\,(\bigvee_{e\in E_A}\mathtt{Tbck}\,(m\in\langle c,d\rangle\rightarrow[\![\phi]\!]_A,\mathtt{Xbck}_e(Z))))$

Table 3. A formulation for the approximate evaluation of $\exists\Box$-formulas

Example 9 *The* initial condition I_A of the TA in example 4 is convex. On the other hand, the corresponding 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 $\nu_1(x)=\nu_1(y)=3$ and $\nu_2(x)=\nu_2(y)=9$. It is clear that (q_1,ν_1) and (q_1,ν_2) are both in $\langle\!\langle\dot{H}\rangle\!\rangle_A$. 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 $\langle\!\langle\dot{H}\rangle\!\rangle_A$.

Concavity may also happen with difference constraints between two clocks. For example, the following zone predicate $\ddot{H}\equiv q_1\wedge(x-y<-3\vee x-y>3)$ is also concave. For example, we may have two states (q_1,ν_3) and (q_1,ν_4) with $\nu_3(x)=9$, $\nu_3(y)=3$, $\nu_4(x)=3$, and $\nu_4(y)=9$. It is clear that (q_1,ν_3) and (q_1,ν_4) are both in $\langle\!\langle\ddot{H}\rangle\!\rangle_A$. However the middle point, say $(q_1,\nu_{7/2})$, between (q_1,ν_3) and (q_1,ν_4) with $\nu_{7/2}(x)=\nu_{7/2}(y)=6$ is not. ■

It is known that formula (T') for time progress evaluation can be applied to convex path conditions.

Example 10 *In* example 4, formula (T') is not applicable to H_A which is concave. For example, $\mathtt{Tbck}(H_A,q_1\wedge x=8\wedge y=8)$ is $q_1\wedge 7<x\leq 8\wedge 7<y\leq 8\wedge x=y$. However, $\mathtt{Tbck}'(H_A,q_1\wedge x=8\wedge y=8)$ is $q_1\wedge 0\leq x\leq 8\wedge 0\leq y\leq 8\wedge x=y$ which is incorrect. ■

According to [11], the restriction of the applicability of formula (T') can be relaxed with the following concept.

Definition 11 **Time-convexity** A state space S is *time-convex* iff for any $(q,\nu)\in S$ and $t\in\mathbb{R}^{\geq 0}$ with $(q,\nu+t)\in S$, then for any $t'\in[0,t]$, $(q,\nu+t')\in S$. If S is not time-convex, it is *time-concave*. ■

Example 12 *In* examples 4 and 9, I_A is time-convex while H_A is time-concave. Moreover, zone predicate $\ddot{H}\equiv q_1\wedge(x-y<-3\vee x-y>3)$ is concave and time-convex. ■

We restate the following lemma from [11].

Lemma 13 *Given a TA A and a time-convex path zone predicate ϕ and a destination zone predicate ψ, $\langle\!\langle\mathtt{Tbck}(\phi,\psi)\rangle\!\rangle_A=\langle\!\langle\mathtt{Tbck}'(\phi,\psi)\rangle\!\rangle_A$.* ■

Lemma 13 implies that we can also apply the more efficient formula of (T') to concave but time-convex path conditions.

Example 14 *In* example 9, $\mathtt{Tbck}(\ddot{H},q_1\wedge 10\leq x\leq 20\wedge 10\leq y\leq 20)$ and $\mathtt{Tbck}'(\ddot{H},q_1\wedge x=8\wedge y=8)$ both evaluate to $q_1\wedge 0\leq x\leq 20\wedge 0\leq y\leq 20\wedge(x-y<-3\vee x-y>3)$. ■

.

According to [11], given a zone predicate ϕ, we can use the following predicate, denoted $\mathtt{time\_concave}(\phi)$, to check if ϕ is time-concave.

$$\phi\wedge\exists t\in\mathbb{R}^{\geq 0}\left(\phi+t\wedge\exists t'\in\mathbb{R}^{\geq 0}(t'<t\wedge(\neg\phi)+t')\right)$$

Lemma 15 *Given a zone predicate ϕ, $\mathtt{time\_concave}(\phi)$ is unsatisfiable if and only if $\langle\!\langle\phi\rangle\!\rangle_A$ is time-convex.* ■

5 Time-convexity checking from the syntax of TCTL formulas

Note that at the end of the last section, we present predicate $\mathtt{time\_concave}()$ to check if a zone predicate is concave. This checking incurs a complementation, three conjunctions, and two existential quantifications and could still be costly. In this section, we present techniques for avoiding this checking. The first idea is to recognize the syntax structures in TCTL that yield only time-convex zone predicates. This syntax checking could be much less expensive than invoking $\mathtt{time\_concave}()$.

Given a TA A, two TCTL formulas ϕ and ψ are *location-disjoint* if there is a $Q'\subseteq Q_A$ such that $\langle\!\langle\phi\rangle\!\rangle_A\subseteq\langle\!\langle\bigvee_{q\in Q'}q\rangle\!\rangle_A$ and $\langle\!\langle\psi\rangle\!\rangle_A\subseteq\langle\!\langle\bigvee_{q\in Q_A-Q'}q\rangle\!\rangle_A$.

The following lemmas characterize classes of TCTL that only yield time-convex zone predicates.

Lemma 16 *Assume we have a TA A, an interval $\langle c,\infty)\subseteq\mathbb{R}^{\geq 0}$, and two time-convex TCTL formulas ϕ and ψ such that ϕ and ψ are location-disjoint. $\langle\!\langle\exists\phi\bigcup_{\langle c,\infty)}\psi\rangle\!\rangle_A$ is time-convex.* ■

Corollary 17 *Assume we have a TA A, an interval $\langle c,\infty)\subseteq\mathbb{R}^{\geq 0}$, and a TCTL formula ϕ such that*

- *$H_A\wedge\phi$ and $H_A\wedge\neg\phi$ are location-disjoint; and*
- *ϕ and $\neg\phi$ are both time-convex.*

Then $\langle\!\langle\exists\Diamond_{\langle c,\infty)}\phi\rangle\!\rangle_A$ is time-convex.

Proof : We can prove that $\langle\!\langle\exists(\neg\phi)\bigcup_{\langle c,\infty)}\phi\rangle\!\rangle_A=\langle\!\langle\exists\Diamond_{\langle c,\infty)}\phi\rangle\!\rangle_A$. Then the corollary follows lemma 16. ■

Lemma 18 *Given a TA A, a time-convex TCTL formula ϕ, a TCTL formula ψ, and an interval $\langle c, \infty) \subseteq (0, \infty)$, $\langle\langle \exists\phi \bigcup_{\langle c,\infty)} \psi \rangle\rangle_A$ is time-convex.* ■

Lemma 19 *Assume we have a TA A, an interval $[0, d\rangle \subseteq \mathbb{R}^{\geq 0}$, and two time-convex TCTL formulas ϕ and ψ such that ϕ and ψ are location-disjoint. $\langle\langle \forall\phi \bigcup_{[0,d\rangle} \psi \rangle\rangle_A$ is time-convex.* ■

The following corollary is important in that it captures the time-convexity of the important class of timed inevitabilities in TCTL formulas.

Corollary 20 *Assume we have a TA A, an interval $[0, d\rangle \subseteq \mathbb{R}^{\geq 0}$, and a TCTL formula ϕ such that*

- *$H_A \wedge \phi$ and $H_A \wedge \neg\phi$ are location-disjoint; and*
- *ϕ and $\neg\phi$ are both time-convex.*

Then $\langle\langle \forall \Diamond_{[0,d\rangle} \phi \rangle\rangle_A$ is time-convex.

Proof : We can prove that $\langle\langle \forall(\neg\phi) \bigcup_{[0,d\rangle} \phi \rangle\rangle_A = \langle\langle \forall \Diamond_{[0,d\rangle} \phi \rangle\rangle_A$. Then the corollary follows lemma 19. ■

Example 21 *For* the TA in example 9, suppose we want to specify that location q_2 will always happen in 10 time units. The property can be written as $\forall \Diamond_{[0,10]} q_2$. According to corollary 20, the space of the property is time-convex. ■

Lemma 22 *Given a TA A, a time-convex TCTL formula ϕ, and an interval $\langle c, \infty) \subseteq \mathbb{R}^{\geq 0}$, $\langle\langle \forall \Box_{\langle c,\infty)} \phi \rangle\rangle_A$ is time-convex.* ■

Lemma 23 *Given a TA A, a time-convex TCTL formula ϕ, and an interval $[0, d\rangle \subseteq \mathbb{R}^{\geq 0}$, $\langle\langle \exists \Box_{[0,d\rangle} \phi \rangle\rangle_A$ is time-convex.* ■

Based on the lemmas in the above, we can define a new class of TCTL formulas that yield only time-convex state spaces. This class is called *TCTL with timed-convexity*, denoted $\texttt{TCTL}^{tc}$, and is defined inductively as follows.

Definition 24 $\underline{\texttt{TCTL}^{tc}}$ Only TCTL formulas of the following forms are members of TCTL^{tc}.

- η, a zone predicate such that $\texttt{time\_concave}(\eta)$ is unsatisfiable.
- $\phi_1 \wedge \phi_2$, with $\phi_1 \in \text{TCTL}^{tc} \wedge \phi_2 \in \text{TCTL}^{tc}$.
- $\exists\phi \bigcup_{\langle c,\infty)} \psi$ with the following two properties.
  - ϕ and ψ are in $\texttt{TCTL}^{tc}$.
  - ϕ and ψ are location-disjoint.
- $\exists\phi \bigcup_{\langle c,\infty)} \psi$ with $\phi \in \texttt{TCTL}^{tc}$ and $\langle c, \infty) \subseteq (0, \infty)$.
- $\forall\phi \bigcup_{[0,d\rangle} \psi$ with the following properties.
  - $\phi \in \texttt{TCTL}^{tc}$ and $\psi \in \texttt{TCTL}^{tc}$.
  - ϕ and ψ are location-disjoint.
- $\forall \Box_{\langle c,\infty)} \phi$ with $\langle c, \infty) \subseteq \mathbb{R}^{\geq 0}$ and $\phi \in \texttt{TCTL}^{tc}$.
- $\exists \Box_{[0,d\rangle} \phi$ with $[0, d\rangle \subseteq \mathbb{R}^{\geq 0}$ and $\phi \in \texttt{TCTL}^{tc}$. ■

Note that $\texttt{TCTL}^{tc}$ itself may not support the full specification of some properties. But it helps identifying subformulas in a full specification that yield only time-convex path conditions for efficient time progress evaluation. For example, in benchmark (H) in table 7 in page 10, the whole property is not in $\texttt{TCTL}^{tc}$. But the inner path condition $\forall \texttt{transm}_1 \bigcup_{[0,52)} \texttt{retry}_1$ is in $\texttt{TCTL}^{tc}$ according to lemma 19. This implies that in the evaluation of the outer $\forall\bigcup$-formula, we do not have to evaluate $\texttt{time\_concave}([\![\forall \texttt{transm}_1 \bigcup_{[0,52)} \texttt{retry}_1]\!]_A)$ in order to use the more efficient formulation (T') in place of (T).

Then we can use the following procedure for the adaptive evaluation of time progress.

Initially $\Phi = \emptyset$. /* for known convexities */
Initially $\Psi = \emptyset$. /* for known concavities */
$\texttt{Adaptive\_Tbck}(\eta_\phi, \phi, \eta_\psi)$ {
  If H_A is time-concave, return $\texttt{Tbck}(\eta_\phi, \eta_\psi)$.
  Else if ϕ is a TCTL^{tc} formula, return $\texttt{Tbck}'(\eta_\phi, \eta_\psi)$.
  Else if $\phi \in \Phi$, return $\texttt{Tbck}'(\eta_\phi, \eta_\psi)$.
  Else if $\phi \in \Psi$, return $\texttt{Tbck}(\eta_\phi, \eta_\psi)$.
  Else if $\texttt{time\_concave}(\eta_\phi)$ is unsatisfiable, {
    $\Phi = \Phi \cup \{\phi\}$; return $\texttt{Tbck}'(\eta_\phi, \eta_\psi)$.
  }
  Else {
    $\Psi = \Psi \cup \{\phi\}$; return $\texttt{Tbck}(\eta_\phi, \eta_\psi)$.
} }

Note that the procedure is designed to return a zone predicate characterizing the same state space as $\texttt{Tbck}(\eta_\phi, \eta_\psi)$. There is one extra argument, ϕ, which is the subformula whose evaluation yields η_ϕ. This extra argument is used in the first "Else if" statement to help us checking if the path condition is generated from a $\texttt{TCTL}^{tc}$ formula and can be evaluated with $\texttt{Tbck}'()$.

Also, we use two set variables Φ and Ψ to record the results of previous invocations of $\texttt{time\_concave}()$. If it has already been recorded, then we also avoid the repeated invocations of $\texttt{time\_concave}()$ on the same argument path condition. Repeated time progress evaluations along the same path condition can actually happen a lot in the least fixpoint and greatest fixpoint evaluation.

The following lemma shows that the procedure is correct.

Lemma 25 *Given a TA A and two zone predicates η_ϕ, η_ψ and a TCTL formula ϕ such that $\langle\langle \phi \rangle\rangle_A = \langle\langle \eta_\phi \rangle\rangle_A$, $\langle\langle \texttt{Tbck}(\eta_\phi, \eta_\psi) \rangle\rangle_A = \langle\langle \texttt{Adaptive\_Tbck}(\eta_\phi, \phi, \eta_\psi) \rangle\rangle_A$.* ■

Without loss of generality, from now on, we assume that in the model-checking, we use $\texttt{Adaptive\_Tbck}()$ in place of $\texttt{Tbck}()$. To check a TCTL formula ϕ against a TA, the only extra cost of using $\texttt{Adaptive\_Tbck}()$ is to feed a subformula ϕ' in ϕ to $\texttt{Adaptive\_Tbck}()$ as the second argument when we want to invoke $\texttt{Adaptive\_Tbck}()$ with the first argument being an (approximate) evaluation of ϕ'.

$\exists m\,(m = 0 \wedge \mathbf{gfp}Z.\,(\exists z\,(z = 0 \wedge ((m < 5 \vee m > 10 \vee \neg\texttt{alarm}) \rightsquigarrow (Z \wedge z \geq k)))))$
$\equiv \exists m\,\big(m = 0 \wedge \mathbf{gfp}Z \exists z\,\big(z = 0 \wedge \mathbf{lfp}Z'.\,\big(Z \wedge z > k \vee \bigvee_{e\in E} \texttt{Tbck}(m < 5 \vee m > 10 \vee \neg\texttt{alarm}, \texttt{Xbck}_e(Z'))\big)\big)\big)$

Table 4. Traditional formulation for the evaluation of $\exists\Box_{[5,10]}\neg\texttt{alarm}$.

6 New formulation of timed inevitabilities

With formulation ($\exists\Box^{NZ}$), formula (A) in example 2 is evaluated with the formula in table 4. As can be seen in table 4, the path condition in time progress evaluation is $m < 5 \vee m > 10 \vee \neg\texttt{alarm}$ which is time-concave.[4] This means that we cannot use $\texttt{Tbck}'()$ in place of $\texttt{Tbck}()$ with formulation ($\exists\Box^{NZ}$).

However, we can see that for a non-Zeno run ρ to satisfy $\Box_{\langle c,d\rangle}\phi$, it means ρ can be decomposed into three run segments ρ_1, ρ_2, ρ_3 such that

- states in ρ_1 happen at time in $[0, d\rangle - \langle c, d\rangle$ and satisfy H_A,
- states in ρ_2 happen at time in $\langle c, d\rangle$ and satisfy $\phi \wedge H_A$, and
- ρ_3 is a non-Zeno run with all states happening at time in $\langle c, \infty) - \langle c, d\rangle$ and satisfy H_A.

This observation leads to the new formulation for a formula like $\exists\Box_{\langle c,d\rangle}\phi$ in table 5. The advantage of the new formulation is that it may happen that for each segment of ρ_1, ρ_2, ρ_3, its individual path condition could be time-convex and the corresponding time progress evaluation can be done with the more efficient formulation (T') in place of (T). Specifically, for example, for $\exists\Box_{[c,d]}\phi$, we have the following mapping between the syntax structure of its new formulation in table 5 and the three run segments.

- The path condition of the outer $\exists\bigcup$-formula characterizes the time progress along ρ_1. It can be evaluated with formulation (T') if $H_A \wedge m < c$ is time-convex.
- The destination condition of the outer $\exists\bigcup$-formula and the path condition of the inner $\exists\bigcup$-formula together characterize the time progress along ρ_2. It can be evaluated with formulation (T') if $\phi \wedge H_A \wedge m \geq c \wedge m \leq d$ is time-convex.
- The destination condition of the inner $\exists\bigcup$-formula and $\mathrm{NZ}(\mathit{true})$ together characterize the time progress along ρ_3. It can be evaluated with formulation (T') if H_A is time-convex.

Given a TA A and a TCTL formula ϕ, we use $[\![\phi]\!]_A^{new}$ to denote the zone predicate obtained for ϕ from the algorithm in tables 1 and 2 except that formulation ($\exists\hat{\Box}^{NZ}$) in table 5 is used in place of formulation ($\exists\Box^{NZ}$) in table 2. The following lemma establishes the correctness of the new formulation in table 5.

Lemma 26 *Given a TA A and a TCTL formula ϕ, $\langle\!\langle\phi\rangle\!\rangle_A = \langle\!\langle[\![\phi]\!]_A^{new}\rangle\!\rangle_A$.* ■

In section 7, we will see that for some benchmarks, formulation ($\exists\hat{\Box}^{NZ}$) indeed leads to much better verification performance. However, there is an additional advantage of formulation ($\exists\hat{\Box}^{NZ}$) in table 5 in that it links to a safe approximation of timed inevitabilities with better precision. Intuitively, we may replace $\mathrm{NZ}(\phi)$ in table 5 directly with ϕ as an approximation for the efficient evaluation of $[\![\mathrm{NZ}(\phi)]\!]_A$. Specifically, we propose the approximate formulation ($\exists\hat{\Box}$) in table 6. Similarly, we use $[\![\phi]\!]_A^{new-app}$ to denote the zone predicate obtained for ϕ from the algorithm in tables 1 and 2 except that formulation ($\exists\Box^{NZ}$) in table 2 for $\exists\Box$-formulas are changed to ($\exists\hat{\Box}$) in table 6. We then have the following lemma for the desired property of our new formulation for approximate model-checking.

Lemma 27 *Given a TA A, a zone predicate η, and an interval $\langle c, d\rangle \subseteq \mathbb{R}^{\geq 0}$. $\langle\!\langle[\![\forall\Diamond_{\langle c,d\rangle}\eta_2]\!]_A^{app}\rangle\!\rangle_A \subsetneq \langle\!\langle[\![\forall\Diamond_{\langle c,d\rangle}\eta]\!]_A^{new-app}\rangle\!\rangle_A \subsetneq \langle\!\langle\forall\Diamond_{\langle c,d\rangle}\eta\rangle\!\rangle_A$.* ■

This lemma shows that the new approximate formulation in table 6 is not only safe but also is strictly more precise than the one in table 3 for evaluating timed inevitabilities. Our experiment report in the next section corroborates this lemma in that against all our timed inevitability benchmarks, the approximate formulation in table 6 yields correct evaluation while the one in table 3 does not.

7 Implementation and experiments

We have implemented procedure $\texttt{Adaptive\_Tbck}()$ in section 5 and the two formulations ($\exists\hat{\Box}^{NZ}$) and ($\exists\hat{\Box}$) in sections 6 in RED 7.0, a model-checker for TAs and a parametric safety analyzer for LHAs (linear hybrid automata) [2] based on the CRD (Clock-Restriction Diagram) and HRD (Hybrid-Restriction Diagram) technology [8, 10]. We used the following two parameterized benchmarks from the literature in our experiment.

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 5, 10, and 19. The formulas that we check are formulas (C), (D), and (E) in table 7. Formula (C) says that if process 1 is in the ready mode,

[4] The path condition is time-concave since for example, given a state $(\texttt{alarm}, \nu)$ with $\nu(m) = 0$, it is true that both $(\texttt{alarm}, \nu)$ and $(\texttt{alarm}, \nu{+}12)$ satisfy $m < 5 \vee m > 10 \vee \neg\texttt{alarm}$. But $(\texttt{alarm}, \nu{+}7)$ does not.

name	formulas	new formulations
$(\exists\hat{\Box}^{NZ})$	$\exists\Box_{[c,d]}\phi$	$\exists m(m=0 \wedge (m<c \rightsquigarrow ((\phi \wedge m \geq c \wedge m \leq d) \rightsquigarrow (m>d \wedge \mathrm{NZ}(\mathit{true})))))$
	$\exists\Box_{(c,d]}\phi$	$\exists m(m=0 \wedge (m\leq c \rightsquigarrow ((\phi \wedge m > c \wedge m \leq d) \rightsquigarrow (m>d \wedge \mathrm{NZ}(\mathit{true})))))$
	$\exists\Box_{[c,d)}\phi$	$\exists m(m=0 \wedge (m<c \rightsquigarrow ((\phi \wedge m \geq c \wedge m < d) \rightsquigarrow (m\geq d \wedge \mathrm{NZ}(\mathit{true})))))$
	$\exists\Box_{(c,d)}\phi$	$\exists m(m=0 \wedge (m\leq c \rightsquigarrow ((\phi \wedge m > c \wedge m < d) \rightsquigarrow (m\geq d \wedge \mathrm{NZ}(\mathit{true})))))$
	$\exists\Box_{[0,d]}\phi$	$\exists m(m=0 \wedge ((\phi \wedge m \leq d) \rightsquigarrow (m>d \wedge \mathrm{NZ}(\mathit{true}))))$
	$\exists\Box_{[0,d)}\phi$	$\exists m(m=0 \wedge ((\phi \wedge m < d) \rightsquigarrow (m\geq d \wedge \mathrm{NZ}(\mathit{true}))))$
	$\exists\Box_{[c,\infty)}\phi$	$\exists m(m=0 \wedge (m<c \rightsquigarrow (m\geq c \wedge \mathrm{NZ}(\phi))))$
	$\exists\Box_{(c,\infty)}\phi$	$\exists m(m=0 \wedge (m\leq c \rightsquigarrow (m> c \wedge \mathrm{NZ}(\phi))))$

Table 5. A new formulation for $\exists\Box$-formulas.

name	formulas	new formulations
$(\exists\hat{\Box})$	$\exists\Box_{[c,d]}\phi$	$\exists m(m=0 \wedge (m<c \rightsquigarrow ((\phi \wedge m \geq c \wedge m \leq d) \rightsquigarrow m>d)))$
	$\exists\Box_{(c,d]}\phi$	$\exists m(m=0 \wedge (m\leq c \rightsquigarrow ((\phi \wedge m > c \wedge m \leq d) \rightsquigarrow m>d)))$
	$\exists\Box_{[c,d)}\phi$	$\exists m(m=0 \wedge (m<c \rightsquigarrow ((\phi \wedge m \geq c \wedge m < d) \rightsquigarrow m\geq d)))$
	$\exists\Box_{(c,d)}\phi$	$\exists m(m=0 \wedge (m\leq c \rightsquigarrow ((\phi \wedge m > c \wedge m < d) \rightsquigarrow m\geq d)))$
	$\exists\Box_{[0,d]}\phi$	$\exists m(m=0 \wedge ((\phi \wedge m \leq d) \rightsquigarrow m>d))$
	$\exists\Box_{[0,d)}\phi$	$\exists m(m=0 \wedge ((\phi \wedge m < d) \rightsquigarrow m\geq d))$
	$\exists\Box_{[c,\infty)}\phi$	$\exists m(m=0 \wedge (m<c \rightsquigarrow (m\geq c \wedge \phi)))$
	$\exists\Box_{(c,\infty)}\phi$	$\exists m(m=0 \wedge (m\leq c \rightsquigarrow (m> c \wedge \phi)))$

Table 6. A new formulation for the approximate evaluation of $\exists\Box$-formulas.

then it will be in either the critical mode or the waiting mode in 19 time units and it cannot stay in the ready mode in the next 5 to 10 time units. This formula consists of the conjunction of two timed inevitabilities.

Formula (D) says that after process 1 enters the ready mode, in 24 to 34 time units, it will be in either the idle mode or the critical mode. This timed inevitabilities of this formula is fulfilled by first crossing the ready mode and then the waiting mode.

Formula (E) says that after process 1 enters the ready mode, it enters the waiting mode in 5 to 10 time units and then enters either the critical section or the idle mode in 19 to 24 time units. The formula consists of a timed inevitability nesting another.

2. *CSMA/CD* [13]: This is the Ethernet bus arbitration protocol with collision-and-retry. The timing constants used are 26, 52, and 808. The properties that we want to check are formulas (F), (G), and (H) in table 7.

   Formula (F) 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. This formula is a safety property, not a timed inevitability.

   Formula (G) says that if senders 1 and 2 are in the transmission mode at the same time, then sender 2 will enter the retry mode in 26 time units. This is a simple timed inevitability.

   Formula (H) uses an outer $\forall\bigcup$-formula that nests a timed inevitability (expressed as another $\forall\bigcup$-formula) in its path condition. The outer $\forall\bigcup$-formula is also a timed inevitability that says if the bus is in the collision mode and sender 1 is in the transmission mode, then the bus will enter the idle mode in 52 time uints. Also before the outer inevitability is fulfilled, sender 1 will inevitably go to the retry mode in 52 time units.

We have collected performance data for our implementation configurations for formulations $(\exists\Box)$, $(\exists\hat{\Box})$, $(\exists\Box^{NZ})$, and $(\exists\hat{\Box}^{NZ})$ for the evaluation of timed inevitabilities. The performance data is reported in table 8. The CPU time used, the total memory consumption for the data-structures in state-space representations, and the answers of model-checking are reported. Following the suggestion in [12], we use $k = C_A^{\phi}$ for all the confugurations. Also we use the early decision techniques for the greatest fixpoint evaluation [12]. In general, our new formulations $(\exists\hat{\Box})$ and $(\exists\hat{\Box}^{NZ})$ for timed ineviatbilities yield better performance than the traditional formulation $(\exists\Box)$ and $(\exists\Box^{NZ})$ do.

As for the approximation techniques, for the five benchmarks for timed inevitabilities, formulation $(\exists\Box)$ also sometimes yield performance comparable with formulation $(\exists\hat{\Box})$. However, the traditional formulation $(\exists\Box)$ does not have the precision to correctly evaluate the timed inevitabilities. In contrast, our new approximate formulation $(\exists\hat{\Box})$ is precise and efficient enough to correctly check the five

(C)	$\forall\Box((\texttt{ready}_1 \rightarrow ((\forall\Diamond_{[0,19]}(\texttt{critical}_1 \vee \texttt{waiting}_1)) \wedge \neg(\exists\Box_{[5,10]}\texttt{ready}_1)))$
(D)	$\forall\Box\left((\texttt{ready}_1 \wedge x_1 = 0) \rightarrow \forall\Diamond_{[24,34]}(\texttt{critical}_1 \vee \texttt{idle}_1)\right)$
(E)	$\forall\Box\left((\texttt{ready}_1 \wedge x_1 = 0) \rightarrow \forall\Diamond_{[5,10]}(\texttt{waiting}_1 \wedge \forall\Diamond_{(19,24]}(\texttt{idle}_1 \vee \texttt{critical}_1))\right)$
(F)	$\forall\Box((\texttt{transm}_1 \wedge x_1 = 52) \rightarrow \forall\Box_{[0,756)}\neg\texttt{transm}_2)$
(G)	$\forall\Box\left((\texttt{transm}_1 \wedge \texttt{transm}_2) \longrightarrow \forall\Diamond_{[0,26)}\texttt{retry}_2\right)$
(H)	$\forall\Box\left((\texttt{bus\_collision} \wedge \texttt{transm}_1) \rightarrow \forall\left(\forall\texttt{transm}_1 \bigcup_{[0,52)}\texttt{retry}_1\right)\bigcup_{[0,52)}\texttt{bus\_idle}\right)$

Table 7. Six formulas used in the experiment

Table 8. Performance data of scalability w.r.t. various strategies

benchmarks	formulas	non-Zeno runs only ?	m	Previous formulation time	Previous formulation memory	Previous formulation satisfied ?	New formulation time	New formulation memory	New formulation satisfied ?
Fischer's mutual exclusion algorithm with m processes	(C)	no	4	7.29s	140M	no	0.112s	886k	yes
			5	N/A			6.76s	134M	
			6				107s	701M	
		yes	4	22.7s	254M	yes	0.088s	896k	yes
			5	N/A			3.16s	79M	
			6				58.5s	482M	
	(D)	no	4	1.22s	44M		0.524s	29M	yes
			5	29.4s	296M	no	6.93s	139M	
			6	N/A			104s	689M	
		yes	4	5.22s	117M		0.092s	886k	yes
			5	140s	919M	yes	2.41s	77M	
			6	N/A			45.2s	387M	
	(E)	no	4	0.076s	407k		0.096s	407k	yes
			5	6.98s	97M	no	3.30s	93M	
			6	N/A			38.8s	356M	
		yes	4	3.78s	102M		0.028s	407k	yes
			5	85.0s	664M	yes	0.092s	1.1M	
			6	N/A			2.38s	76M	
CSMA/CD with 1 bus and m senders	(F)	no	3	0.108s	325k		0.112s	325k	yes
			4	6.17s	87M	yes	2.65s	71M	
			5	69.8s	448M		50.0s	359M	
		yes	3	0.152s	325k		0.092s	325k	yes
			4	6.17s	87M	yes	2.57s	71M	
			5	69.8s	448M		49.7s	359M	
	(G)	no	3	0.112s	325k		0.064s	325k	yes
			4	5.05s	76M	no	0.108s	746k	
			5	50.5s	367M		2.08s	65M	
		yes	3	81.3s	479M	yes	56.0s	414M	yes
	(H)	no	3	0.128s	325k		0.072s	325k	yes
			4	2.32s	77M	no	0.168s	746k	
			5	48.0s	391M		2.20s	65M	
		yes	3	250s	1031M	yes	53.8s	413M	yes

data collected on a Pentium 4 1.7GHz with 2G memory running LINUX;
N/A: not available; s: seconds;
k: kilobytes of memory in diagram data-structure; M: megabytes of memory in diagram data-structure

timed inevitabilities. This may imply that when computing budget is a concern, formulation $(\exists\hat{\Box})$ seems a good choice for economic and effective verification configuration.

Also, we have not tuned the garbage collection capabilities of RED in the experiment. In our present implementation, the garbage collection in RED is invoked regularly with procedure calls. As a result, garbage collection may consume excessive computation time. In the future, we may gain more performance with a better implementation of the garbage collector.

8 Concluding remarks

In this work, we discuss how to improve the performance of TCTL model-checking algorithm with several techniques. In our experiment, our techniques sometimes have enhanced the performance of TCTL model-checking dramatically against several benchmarks. We feel hopeful that such techniques could be used as a foundation in our future endeavor to make TCTL model-checking an industrial strength.

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. Model-Checking Distributed Real-Time Systems with States, Events, and Multiple Fairness Assumptions. AMAST'2004, LNCS 3116, Springer-Verlag.

[10] 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.

[11] F. Wang. Time-Progress Evaluation for Dense-Time Automata with Concave Path Conditions. ATVA 2008, LNCS, Springer-Verlag.

[12] 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.

[13] 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.

APPENDICES

A Proof for lemma 8

Lemma 8 Given a TA A, a zone predicate η, and an interval $\langle c,d\rangle \subseteq \mathbb{R}^{\geq 0}$. $\langle\!\langle [\![\forall\Diamond_{\langle c,d\rangle}\eta]\!]^{app}_A \rangle\!\rangle_A \subsetneq \langle\!\langle \forall\Diamond_{\langle c,d\rangle}\eta\rangle\!\rangle_A$.
Proof : Note that for the timed inevitability of $\forall\Diamond_{\langle c,d\rangle}\eta_2$, the set runs considered in formulation ($\exists\Box$) is a superset of those considered in formulation ($\exists\Box^{NZ}$). Thus it is clear that the lemma is true with the semantics of $\forall\Diamond$-formulas. ■

B Proof for lemma 16

Lemma 16 Assume we have a TA A, an interval $\langle c,\infty) \subseteq \mathbb{R}^{\geq 0}$, and two time-convex TCTL formulas ϕ and ψ such that ϕ and ψ are location-disjoint. $\langle\!\langle \exists\phi\bigcup_{\langle c,\infty)}\psi\rangle\!\rangle_A$ is time-convex.
Proof : We assume there is a $Q' \subseteq Q_A$ such that $\langle\!\langle \phi\rangle\!\rangle_A \subseteq \langle\!\langle \bigvee_{q\in Q'} q\rangle\!\rangle_A$ and $\langle\!\langle \psi\rangle\!\rangle_A \subseteq \langle\!\langle \bigvee_{q\in Q_A-Q'} q\rangle\!\rangle_A$. We assume that there is a state (q,ν) and a $t \in \mathbb{R}^{\geq 0}$ such that (q,ν) and $(q,\nu+t)$ are both in $\langle\!\langle \exists\phi\bigcup_{\langle c,\infty)}\psi\rangle\!\rangle_A$. There are two cases to analyze.

- $\underline{q \in Q'}$: This implies that both (q,ν) and $(q,\nu+t)$ are in $\langle\!\langle \phi\rangle\!\rangle_A$. Since $\langle\!\langle \phi\rangle\!\rangle_A$ is time-convex, we know that for every $t' \in [0,t]$, $(q,\nu+t') \in \langle\!\langle \phi\rangle\!\rangle_A$ also. Then the concatenation of the time progression from $(q,\nu+t')$ to $(q,\nu+t)$ and a non-Zeno run from $(q,\nu+t)$ for the proof of $(q,\nu+t) \in \langle\!\langle \exists\phi\bigcup_{\langle c,\infty)}\psi\rangle\!\rangle_A$ can be used to support that $(q,\nu+t') \in \langle\!\langle \exists\phi\bigcup_{\langle c,\infty)}\psi\rangle\!\rangle_A$.
- $\underline{q \in Q_A - Q'}$: This is possible only when $\langle c,\infty)$ is $[0,\infty)$ since (q,ν) does not satisfy the path condition of $\exists\phi\bigcup_{\langle c,\infty)}\psi$ in A. In this case, we know that (q,ν) and $(q,\nu+t)$ are both in $\langle\!\langle \psi\rangle\!\rangle_A$. Since $\langle\!\langle \psi\rangle\!\rangle_A$ is time-convex, we know that for every $t' \in [0,t]$, $(q,\nu+t') \in \langle\!\langle \psi\rangle\!\rangle_A$ also. This implies that $A, (q,\nu+t') \in \langle\!\langle \exists\phi\bigcup_{[0,\infty)}\psi\rangle\!\rangle_A$.

Thus the lemma is proven. ■

C Proof for lemma 18

Lemma 18 Given a TA A, a time-convex TCTL formula ϕ, a TCTL formula ψ, and an interval $\langle c,\infty) \subseteq (0,\infty)$, $\langle\!\langle \exists\phi\bigcup_{\langle c,\infty)}\psi\rangle\!\rangle_A$ is time-convex.
Proof : We assume that there is a state (q,ν) and a $t \in \mathbb{R}^{\geq 0}$ such that (q,ν) and $(q,\nu+t)$ are both in $\langle\!\langle \forall\phi\bigcup_{\langle c,\infty)}\psi\rangle\!\rangle_A$. Since $\langle c,\infty) \neq [0,\infty)$, we know that $0 \notin \langle c,\infty)$ which implies that (q,ν) and $(q,\nu+t)$ are both in $\langle\!\langle \phi\rangle\!\rangle_A$. Since $\langle\!\langle \phi\rangle\!\rangle_A$ is time-convex, we know that for every $t' \in [0,t]$, $(q,\nu+t') \in \langle\!\langle \phi\rangle\!\rangle_A$ also. Suppose there is a non-Zeno run ρ that satisfy $\phi\bigcup_{\langle c,\infty)}\psi$ in A with a state fulfilling ψ at a state $\hat{t}$ time units from $(q,\nu+t')$ with $\hat{t} \in \langle c,\infty)$. Then the concatenation of the time progression from $(q,\nu+t')$ to $(q,\nu+t)$ and ρ can be used to support that $(q,\nu+t') \in \langle\!\langle \exists\phi\bigcup_{\langle c,\infty)}\psi\rangle\!\rangle_A$ since $t-t'+\hat{t} \geq \hat{t}$ and $t-t'+\hat{t} \in \langle c,\infty)$ also. ■

D Proof for lemma 19

Lemma 19 Assume we have a TA A, an interval $[0,d\rangle \subseteq \mathbb{R}^{\geq 0}$, and two time-convex TCTL formulas ϕ and ψ such that ϕ and ψ are location-disjoint. $\langle\!\langle \forall\phi\bigcup_{[0,d\rangle}\psi\rangle\!\rangle_A$ is time-convex.
Proof : We assume there is a $Q' \subseteq Q_A$ such that $\langle\!\langle \phi\rangle\!\rangle_A \subseteq \langle\!\langle \bigvee_{q\in Q'} q\rangle\!\rangle_A$ and $\langle\!\langle \psi\rangle\!\rangle_A \subseteq \langle\!\langle \bigvee_{q\in Q_A-Q'} q\rangle\!\rangle_A$. We assume that there is a state (q,ν) and a $t \in \mathbb{R}^{\geq 0}$ such that (q,ν) and $(q,\nu+t)$ are both in $\langle\!\langle \forall\phi\bigcup_{[0,d\rangle}\psi\rangle\!\rangle_A$. There are two cases to analyze.

- $\underline{q \in Q'}$: This implies that both (q,ν) and $(q,\nu+t)$ are in $\langle\!\langle \phi\rangle\!\rangle_A$. Moreover, both (q,ν) and $(q,\nu+t)$ are not in $\langle\!\langle \psi\rangle\!\rangle_A$. This further implies that $t \in [0,d\rangle$. Otherwise, the time progression from (q,ν) to $(q,\nu+t)$ refutes $(q,\nu) \in \langle\!\langle \forall\phi\bigcup_{[0,d\rangle}\psi\rangle\!\rangle_A$. Since $\langle\!\langle \phi\rangle\!\rangle_A$ is time-convex, we know that for every $t' \in [0,t]$, $(q,\nu+t') \in \langle\!\langle \phi\rangle\!\rangle_A$ and $(q,\nu+t') \notin \langle\!\langle \psi\rangle\!\rangle_A$ also. With some arithmetic on the time length to the fulfillment of ψ, we know that for every $t' \in [0,t]$, $(q,\nu+t') \in \langle\!\langle \forall\phi\bigcup_{[0,\lfloor d-t'\rfloor\rangle}\psi\rangle\!\rangle_A$. Since $[0,\lfloor d-t'\rfloor\rangle \subseteq [0,d\rangle$, we know that $(q,\nu+t') \in \langle\!\langle \forall\phi\bigcup_{[0,d\rangle}\psi\rangle\!\rangle_A$ according to the semantics of TCTL.
- $\underline{q \in Q_A - Q'}$: In this case, we know that (q,ν) and $(q,\nu+t)$ are both in $\langle\!\langle \psi\rangle\!\rangle_A$. Since $\langle\!\langle \psi\rangle\!\rangle_A$ is time-convex, we know that for every $t' \in [0,t]$, $(q,\nu+t') \in \langle\!\langle \psi\rangle\!\rangle_A$ also. This implies that $A, (q,\nu+t') \in \langle\!\langle \forall\phi\bigcup_{[0,d\rangle}\psi\rangle\!\rangle_A$.

Thus the lemma is proven. ■

E Proof for lemma 22

Lemma 22 Given a TA A, a time-convex TCTL formula ϕ, and an interval $\langle c,\infty) \subseteq \mathbb{N}$, $\langle\!\langle \forall\Box_{\langle c,\infty)}\phi\rangle\!\rangle_A$ is time-convex.
Proof : We assume that there is a state (q,ν) and a $t \in \mathbb{R}^{\geq 0}$ such that (q,ν) and $(q,\nu+t)$ are both in $\langle\!\langle \forall\Box_{\langle c,\infty)}\phi\rangle\!\rangle_A$. Since every run from $(q,\nu+t')$ is a tail run from (q,ν). There are two cases to analyze.

- $\underline{t' \notin \langle c,\infty)}$: This implies that $(q,\nu+t') \in \langle\!\langle \forall\Box_{\langle \lceil c-t'\rceil,\infty)}\phi\rangle\!\rangle_A$. It is easy to see that $\langle c,\infty) \subseteq \langle \lceil c-t'\rceil,\infty)$. Thus we know $(q,\nu+t') \in \langle\!\langle \forall\Box_{\langle c,\infty)}\phi\rangle\!\rangle_A$ according to the semantics of TCTL.
- $\underline{t' \in \langle c,\infty)}$: This implies $(q,\nu+t') \in \langle\!\langle \forall\Box_{[0,\infty)}\phi\rangle\!\rangle_A$. This further implies that $(q,\nu+t') \in \langle\!\langle \forall\Box_{\langle c,\infty)}\phi\rangle\!\rangle_A$ according to the semantics of TCTL.

Thus the lemma is proven. ■

F Proof for lemma 23

Lemma 23 Given a TA A, a time-convex TCTL formula ϕ, and an interval $[0, d\rangle \subseteq \mathbb{R}^{\geq 0}$, $\langle\!\langle \exists\Box_{[0,d\rangle}\phi \rangle\!\rangle_A$ is time-convex.
Proof : We assume that there is a state (q, ν) and a $t \in \mathbb{R}^{\geq 0}$ such that (q, ν) and $(q, \nu + t)$ both satisfies $\langle\!\langle \exists\Box_{[0,d\rangle}\phi \rangle\!\rangle_A$. This implies that $A, (q, \nu) \models \phi$ and $A, (q, \nu + t) \models \phi$. Since ϕ is time-convex, we know that for every $t' \in [0, t]$, $(q, \nu+t') \in \langle\!\langle \phi \rangle\!\rangle$. Moreover, there is a non-Zeno run ρ from $(q, \nu + t)$ that satisfies $\Box_{[0,d\rangle}\phi$. This implies that along ρ, $\Box_{[0,d-(t-t')\rangle}\phi$ is also true. Thus the concatenation of the time progression from $(q, \nu + t')$ to $(q, \nu + t)$ and ρ is a proof that $(q, \nu + t') \in \langle\!\langle \exists\Box[0, d)\phi \rangle\!\rangle_A$. ■

G Proof for lemma 25

Lemma 25 Given a TA A and two zone predicates η_ϕ, η_ψ and a TCTL formula ϕ such that $\langle\!\langle \phi \rangle\!\rangle_A = \langle\!\langle \eta_\phi \rangle\!\rangle_A$, $\langle\!\langle \texttt{Tbck}(\eta_\phi, \eta_\psi) \rangle\!\rangle_A = \langle\!\langle \texttt{Adaptive\_Tbck}(\eta_\phi, \phi, \eta_\psi) \rangle\!\rangle_A$.
Proof : The lemma is true based on lemmas 16 to 23, the fact that the intersection of two time-convex state space is still time-convex, the correct construction of predicate $\texttt{time\_concave}(\phi)$, and lemma 13. ■

H Proof for lemma 26

Lemma 26 Given a TA A and a TCTL formula ϕ, $\langle\!\langle \phi \rangle\!\rangle_A = \langle\!\langle [\![\phi]\!]_A^{new} \rangle\!\rangle_A$.
Proof : The evaluation of $[\![\phi]\!]_A^{new}$ differs from that of $[\![\phi]\!]_A$ only in the formulations used for $\exists\Box$-formulas. Given such a formula, say $\exists\Box_{\langle c,d\rangle}\phi$, according to the semantics of TCTL, the formula is satisfied, if there is a run as the concatenation of a head run segment ρ_1, a middle run segment ρ_2, and a non-Zeno tail run segment ρ_3 as described in section 6. Then by a straightforward structural inductive analysis, we find the followings are true.

- The inner-most modal formula $\exists\Box\Diamond_{[k,\infty)}\phi$ is satisfied if and only if there is such a non-Zeno tail run segment ρ_3.
- The inner $\exists\bigcup$-formula is satisfied if and only if there is such a middle run segment ρ_2 followed by a non-Zeno tail run segment ρ_3.
- The outer existentially quantified $\exists\bigcup$-formula is satisfied if and only if there is such a head run segment ρ_1 followed by a middle run segment ρ_2 and then by a non-Zeno tail run segment ρ_3.

With this structural inductive analysis, it is clear that the lemma is true. ■

I Proof for lemma 27

Lemma 27 Given a TA A, a zone predicate η, and an interval $\langle c, d\rangle \subseteq \mathbb{R}^{\geq 0}$. $\langle\!\langle [\![\forall\Diamond_{\langle c,d\rangle}\eta]\!]_A^{app} \rangle\!\rangle_A \subsetneq \langle\!\langle [\![\forall\Diamond_{\langle c,d\rangle}\eta]\!]_A^{new-app} \rangle\!\rangle_A \subsetneq \langle\!\langle \forall\Diamond_{\langle c,d\rangle}\eta \rangle\!\rangle_A$.
Proof : The proof is similar to the one for lemma 8. For the first part of the lemma, note that for the timed inevitability of $\forall\Diamond_{\langle c,d\rangle}\eta_2$, the set runs considered in formulation $(\exists\Box)$ is a superset of those considered in formulation $(\exists\hat{\Box})$. The extra runs considered in formulation $(\exists\Box)$ are those Zeno runs with time converges to a value in $[0, d\rangle$. Thus it is clear that this part is true with the semantics of $\forall\Diamond$-formulas.

For the second part of the lemma, note that for the timed inevitability of $\forall\Diamond_{\langle c,d\rangle}\eta_2$, the set runs considered in formulation $(\exists\hat{\Box})$ is a superset of those considered in formulation $(\exists\Box^{NZ})$. The extra runs considered in formulation $(\exists\hat{\Box})$ are those Zeno runs that starts with

- a head segment with states of time in $[0, d\rangle - \langle c, d\rangle$ from the start of the run; and
- a middle segment with states of time in $\langle c, d\rangle$ from the start of the run; and
- a tail Zeno run segment.

Thus it is clear that this part is true with the semantics of $\forall\Diamond$-formulas.

With the two parts proven, we know the lemma is proven. ■