International Journal of Control and Automation
Vol. 10, No. 12 (2017), pp.37-50
http//dx.doi.org/10.14257/ijca.2017.10.12.04

Symbolic Model-Checking for Abstracting Inevitability Modalities over Transient States

Mohammed Achkari Begdouri*, Houda Bel Mokadem and Mohamed El Haddad

Department of Computer Science and Communication/LABTIC
ENSA of TANGIER, AbdelMalek Essaadi University,
P.O. Box 1818 Principal Tangier, Tangier, Morocco
**achkari.med@gmail.com*

Abstract

The context of this study is the model-checking of timed systems. The timed logic $TCTL^{\Delta}$ has been introduced as a powerful extension of TCTL, in order to specify transient states that last for less than k time units. The decidability of the model-checking algorithm has been proved for all modalities of this extension, using a suitable adaptation of Alur and Dill's region graph. Unfortunately, this theoretical result cannot be normally implemented because of its state-space explosion problem. But this is not surprising since, even for the classical timed logic TCTL, the region graph algorithm is not used in model-checkers like UPPAAL or KRONOS. Indeed, these tools use instead a so-called zone algorithm and data structures like DBMs.

In previous work, we presented a zone-based model-checking algorithm for the $TCTL^{\Delta}$ reachability modality $EU^{k}_{\sim c}$. We propose here an extension of this study, in order to specify the other modalities, namely the inevitability formulas $AU^{k}_{\sim c}$. We present symbolic model-checking algorithms computing characteristic sets of all $AU^{k}_{\sim c}$ modalities and check their truth values. We also present a complete correctness proof of these algorithms, and their implementations using the DBM data structure.

***Keywords**: Timed automata, symbolic model checking, inevitability, backward analysis algorithms, correctness, data structures*

1. Introduction

Recently computerized systems have developed rapidly and have become more and more complex. Unfortunately, this development leads to an increased vulnerability for errors. Many of those errors could have been avoided if implemented softwares had been formally verified prior to their use. The need for formal verification of such systems is therefore becoming an increasingly important priority.

In this approach, automatic verification, more specifically model-checking, has been widely growing over the last thirty years. In fact, it has been extended to real-time systems, where quantitative conditions about time have to be handled explicitly. To describe quantitative requirements of systems, we can use timed logics to express timed specifications.

Timed Models. Real-time model-checking has been mostly proposed and developed in the framework of Alur and Dill's Timed Automata [24], *i.e.*, automata extended with a set of real-valued variables, called clocks, that evolve synchronously with time, and allowing to express constraints over delays between different actions of the modeled system [13, 2]. This formalism is now regularly applied to analyse real-time control programs [12, 11] and timing analysis of algorithms and industrial systems [4]. Also, response-time and

Received (January 27, 2017), Review Result (October 31, 2017), Accepted (November 10, 2017)
* Corresponding Author

ISSN: 2005-4297 IJCA

robustness analysis based on timed automata modeling multitask applications running under real-time operating systems have received significant research effort [14, 6]. Furthermore, verification algorithms have been extended to these models, and several analysis tools have been developed [7] and successfully applied to numerous case studies [10, 4].

Timed temporal logics and duration properties. Following the study of timed automata, timed temporal logics have been proposed to extend the classical untimed temporal logics with quantitative modalities. There was several ways of expressing such constraints, for example, the timed logic TCTL has been proposed as a natural extension of CTL [25], where modalities are augmented with time comparisons of the form $\sim$ c, where $\sim$ is a comparison operator. We also cite the parametrized TCTL [21] where TCTL and the timed automata are in turn extended with parameters.

In another direction, numerous works have been devoted to the algorithmic computation of duration properties for timed systems. Since clocks are sometimes not expressive enough, hybrid variables have been considered. The resulting model of hybrid automata has been extensively studied in the last few years [15, 9, 3].

Further research has thus been dedicated to weaker models where hybrid variables are only used as observers, *i.e.* are not checked in the automaton and thus play no role during a computation. These variables, sometimes called costs or prices, can be used in an optimization criterium [19] or as constraints in temporal logic formulas. For instance, the logic WCTL [20], interpreted over timed automata extended with costs, adds cost constraints on modalities: it is possible to express that a given state is reachable within a fixed cost bound [8, 5].

Abstracting transient states. There exists several systems that handle variables whose values are subject to instantaneous changes. Such cases occur often when practical examples in the area of industrial automation are considered. Thus the need for abstracting transient states becomes an important requirement, especially with critical systems where all changes have to be controlled and taken into account. This motivated the work in [17, 16], where events that do not last continuously for at least k time units could be abstracted by introducing an extension of TCTL called TCTL$^{\Delta}$. The decidability result of TCTL$^{\Delta}$ model-checking problem is based an extension of the region graph proposed in [16]. However, the region graph is not used for implementation, but tools like UPPAAL or KRONOS use a so-called “zone algorithm”. This algorithm computes on-the-fly the set of reachable symbolic states, that is pairs (q,Z) where q is a control state and Z a zone. Zones have a practical advantage is that they can be easily implemented using DBMs data structures [18].

Contribution. The aim of this paper is to provide implementable model-checking algorithms for TCTL$^{\Delta}$ inevitability modalities. The algorithms we propose are an extension of the zone algorithm used for TCTL timed logics, and present the continuation of the work started in [1], regarding the reachability modality $EU^{k}{}_{\sim c}$. The main result of this paper is the proof of correctness of our algorithms.

Outline. This paper is organized as follows: we first present basic notions on timed automata model and give the main features of TCTL$^{\Delta}$ timed logics (Section 2). After we shortly recall the classical zone algorithm for the timed logic TCTL (Section 3); we explain thereafter our algorithms, we give a complete proof of its correctness (Section 4); the following section is devoted to present a sample model-checking Pseudo-Code for an inevitability modality (Section 5); we finally give some concluding remarks (Section 6).

2. Basic Notions

We first recall the definition of timed automata proposed by Alur and Dill in [24] and then we remind timed temporal logic TCTL$^{\Delta}$ [16].

2.1. Notations

Let N and R denote the sets of natural and non-negative real numbers, respectively. Let X be a set of real valued clocks. We write C(X) for the set of boolean expressions over atomic formulae of the form $x \sim k$ with $x \in X$, $k \in N$, and $\sim \in \{<,\leq,=,\geq,>\}$. Constraints of C(X) are interpreted over valuations for clocks, *i.e.* mappings from X to R. R^X denotes the set of valuations. For every $v \in R^X$ and $d \in R$, we use $v + d$ to denote the time assignment which maps each clock $x \in X$ to the value $v(x)+d$. For every $r \subseteq X$, we write $v[r \leftarrow 0]$ for the valuation that maps each clock in r to the value 0 and agrees with v over $X \setminus r$. Let AP be a set of atomic propositions.

2.2. Timed Automata

Definition 1. A timed automaton (TA) [24] is a tuple $A = (X,Q_A, q_{init},\rightarrow_A, Inv_A, l_A)$ where X is a finite set of clocks, Q_A is a finite set of locations or control states and $q_{init} \in Q_A$ is the initial location. The set $\rightarrow_A \subseteq Q_A \times C(X) \times 2^X \times Q_A$ is a finite set of action transitions: for $(q, g, r, q') \in \rightarrow_A$, g is the enabling condition and r is a set of clocks to be reset with the transition (we write $q \rightarrow_{g,r} q'$). InvA : $Q_A \rightarrow C(X)$ assigns an invariant to each control state. Finally lA : $Q_A \rightarrow 2^{AP}$ labels every location with a subset of AP.

A state (or configuration) of a TA A is a pair (q, v), where $q \in Q_A$ is the current location and $v \in R^X$ is the current clock valuation. The initial state of A is (q_{init} , v_0) with $v_0(x) = 0$ for any x in X. There are two kinds of transition. From (q, v), it is possible to perform the action transition $q \rightarrow_{g,r} q'$ if $v \models g$ and $v[r \leftarrow 0] \models InvA(q')$ and then the new configuration is $(q', v[r \leftarrow 0])$. It is also possible to let time elapse, and reach (q, v + d) for some $d \in R$ whenever the invariant is satisfied along the delay. Formally the semantics of a TA A is given by a Timed Transition System (TTS) $TA = (S, s_{init},\rightarrow_{TA}, l)$ where:

- $S = \{(q, v) \mid q \in Q_A$ and $v \in R^X$ s.t. $v \models Inv_A(q)\}$ and $s_{init} = (q_{init}, v_0)$.
- $\rightarrow_{TA} \subseteq S \times S$ and we have $(q, v)\rightarrow_{TA}(q', v')$ iff
  - either $q' = q$, $v' = v + d$ and $v + d' \models Inv_A(q)$ for any $d' \leq d$. This is a delay transition, we write $(q, v) \rightarrow_d (q, v + d)$,
  - or $\exists q \rightarrow_{g,r} q'$ and $v \models g$, $v' = v[r \leftarrow 0]$ and $v' \models Inv_A(q')$. This is an action transition, we write $(q, v) \rightarrow a\ (q', v')$.
- $l : S \rightarrow 2^{AP}$ labels every state (q, v) with the subset $l_A(q)$ of AP .

An execution (or run) of A is an infinite path $s_0 \rightarrow_{TA} s_1 \rightarrow_{TA} s_2 \ldots$ in TA such that (1) time diverges and (2) there are infinitely many action transitions. Let Exec(s) be the set of all executions from s. With a run ρ: $(q_0, v_0) \rightarrow_{d1}\rightarrow_a (q_1, v_1) \rightarrow_{d2}\rightarrow_a \ldots$ of A, we associate the sequence of absolute dates defined by $t_0 = 0$ and $t_i = \Sigma_{j\leq i}\, d_j$ for $i \geq 1$, and in the sequel, we often write ρ as the sequence $((q_i, v_i, t_i))_{i\geq 0}$.

A state (q, v) can occur several times along a run ρ, the notion of position allows us to distinguish them: every occurrence of a state is associated with a unique position. Given a position p, the corresponding state is denoted by s_p. The standard notions of prefix, suffix and subrun apply to paths in TTS: given a position $p \in \rho$, $\rho^{\leq p}$ is the prefix leading to p, $\rho^{\geq p}$ is the suffix issued from p. Finally, a subrun σ from p to p' is denoted by $p \rightarrow_\sigma p'$.

Given a position $p \in \rho$, the prefix $\rho^{\leq p}$ has a duration, $Time(\rho^{\leq p})$, defined as the sum of all delays along $\rho^{\leq p}$. For a subset $P \subseteq \rho$ of positions in ρ, we define a natural measure $\mu (P) = \mu\{Time(\rho^{\leq p}) \mid p \in P\}$, where μ is Lebesgue measure on the set of real numbers. In the sequel, we only use this measure when P is a subrun of ρ: in this case, for a subrun σ such that $p \rightarrow_\sigma p'$, we simply have $\mu(\sigma) = Time(\rho^{\leq p'}) - Time(\rho^{\leq p})$.

2.3. The Timed Temporal Logic TCTL$^\Delta$

The syntax of TCTL was extended in [16] to express that a formula holds everywhere except on subruns with duration a parameter $k \in N$: TCTL$^\Delta$ is obtained by adding to TCTL the modalities $E\,U^k_{\sim c}$ and $AU^k_{\sim c}$, where $c, k \in N$.

We include the following abbreviations:

$EF^k{}_{\sim c}\ \phi =_{def} E(T\ U^k{}_{\sim c}\ \phi)$ $AF^k{}_{\sim c}\ \phi =_{def} A(T\ U^k{}_{\sim c}\ \phi)$

$EG^k{}_{\sim c}\ \phi =_{def} \neg AF^k{}_{\sim c}\ \neg\phi$ $AG^k{}_{\sim c}\ \phi =_{def} \neg EF^k{}_{\sim c}\ \neg\phi$

Definition 2 (Semantics of TCTL$^{\Delta}$). The following clauses define when a state s of some TTS $T = (S, s_{init}, \rightarrow, l)$ satisfies a TCTL$^{\Delta}$ formula ϕ, written $s \models \phi$, by induction over the structure of ϕ.

$s \models \neg\phi$	iff	$s \models \phi$
$s \models \phi \wedge \psi$	iff	$s \models \phi$ and $s \models \psi$
$s \models E\phi U_{\sim c}\psi$	iff	$\exists\ \rho \in Exec(s)$ s.t. $\rho \models \phi U_{\sim c}\psi$
$s \models A\phi U_{\sim c}\psi$	iff	$\forall\ \rho \in Exec(s)$ we have $\rho \models \phi U_{\sim c}\psi$
$s \models E\phi U^k{}_{\sim c}\psi$	iff	$\exists\ \rho \in Exec(s)$ s.t. $\rho \models \phi U^k{}_{\sim c}\psi$
$s \models A\phi U^k{}_{\sim c}\psi$	iff	$\forall\ \rho \in Exec(s)$ we have $\rho \models \phi U^k{}_{\sim c}\psi$
$\rho \models \phi U_{\sim c}\psi$	iff	$\exists p \in \rho$ s.t. $Time(\rho^{\leq p}) \sim c \wedge s_p \models \psi \wedge \forall p' <_{\rho} p, s_{p'} \models \phi$
$\rho \models \phi Uk{\sim}c\psi$	iff	there exists a subrun σ along ρ, a position $p \in \sigma$ s.t. $Time(\rho{\leq}p){\sim}c \wedge \mu(\sigma) > k \wedge \forall p' {\in} \sigma, sp' \models \psi$ and for all subrun σ' s.t. $\sigma' <_{\rho} p \wedge \forall p' \in \sigma', sp' \models \neg\phi$ we have $\mu(\sigma') \leq k$

Modality $E\phi U^k{}_{\sim c}\psi$ means that it is possible to reach a sufficiently long interval (>k) where ψ is true, around a position at a distance $\sim$ c and, before this position, ϕ is everywhere true except along negligible duration subpaths ($\leq$ k). Whereas modality $A\phi U^k{}_{\sim c}\psi$ means that along any path, ψ lasts long enough (> k) around a position at a distance $\sim$ c and, before this position, ϕ is everywhere true except along negligible duration subpaths ($\leq$ k).

2.4. Decidability Result for TCTL$^{\Delta}$

The decidability result of TCTL$^{\Delta}$ model-checking is based on a generalization of the Alur and Dill's region graph as presented in [16]. However, this theoretical result cannot be normally implemented, because in dense time models, the construction of the region graph leads to the state-space explosion problem [23]. Instead of it, and for reasons of efficiency, model-checkers like UPPAAL use a symbolic analysis algorithm, called the "zone algorithm", in order to explore finitely the reachable symbolic states [22]. The implementation of this algorithm is based on a data structure called the Difference Bounded Matrices [18], DBMs for short.

The aim of this paper is to provide such implementable algorithms for model-checking TCTL$^{\Delta}$ inevitability modalities. So, we first shortly recall the zone algorithm for TCTL timed logics. Then we will present our symbolic model-checking algorithms with the complete correctness proof.

3. Classical Zone Algorithm, State of the Art

In this section, we describe briefly the on-the-fly algorithm implemented in some model-checkers for verifying TCTL timed logics. Before presenting this algorithm, we first present the symbolic representation called Zone.

3.1. Zones

The set of configurations of a timed automaton is infinite. To check this model, it is therefore necessary to be able to manipulate large sets of configurations, and thus to provide an efficient symbolic representation, called zone. A zone is a set of valuations defined by a conjunction of atomic constraints $x \sim c$ or $x - y \sim c$ where x and y are clocks, $\sim$ is a comparison sign, and c is a integer constant. In forward and backward analysis, the objects that will be handled are symbolic states (q,Z) where q is a control state of the automaton and Z a zone. On zones, multiple operations can be performed (Future, Past, Clock reset. . .). A detailed presentation of zones can be found in [13].

3.2. The Algorithm

The algorithm presented in [22] aims to calculate for each TCTL formula, its characteristic set defined as set of pairs (q,Z) where q is a control state of the automaton and Z a zone. We describe here only the the algorithm of $E\phi 1U{\sim}c\phi 2$, the other modalities can be found for example in [25]. For the formula $E\phi 1U{\sim}c\phi 2$, the characteristic set is given by the following recurrent sequence:

$$[[E\phi_1 U_{\sim c}\phi_2]] = EU([z \leftarrow 0][[\phi_1]], [[\phi_2]] \cap [[z \sim c]])$$

Where z is the clock corresponding to the operator U and $EU(R_1,R_2) = U_{i\geq 0} E_i$ with:

$$\begin{cases} E_0 = R_2 \\ E_{i+1} = Pre[R_1](E_i) \cup Pre(E_i) \end{cases}$$

$Pre[R_1](E_i)$ represents the set of configurations that allow to reach E_i by letting time pass while staying in R_1, while $Pre(E_i)$ represents the configurations that allow to reach E_i by taking an action transition. A clock is attached to each U operator in the formula. It is used to handle subscripts $\sim c$ in until modalities. We note that the above analysis is in fact a backward analysis.

4. TCTL$^{\Delta}$ Inevitability Modalities: Symbolic Model-Checking Algorithms

In this section, we show our symbolic model-checking algorithms for TCTL$^{\Delta}$ inevitability modalities using a backward analysis. We present in parallel the complete correctness proof for each algorithm.

4.1. Modality $E\phi_1 U^k_{\sim c}\phi_2$

For this modality, we recall briefly the approach opted in [1], based on splitting the semantics of $E\phi_1 U^k_{\sim c}\phi_2$ in two parts (as depicted in Figure 1). The left part represents the subrun where ϕ_1 is true everywhere except along negligible duration subpaths ($\leq$ k). While the right part represents the subrun where ϕ_2 lasts long enough around a position ($z \sim c$), and before this position ϕ_1 is true except along negligible duration subpaths.

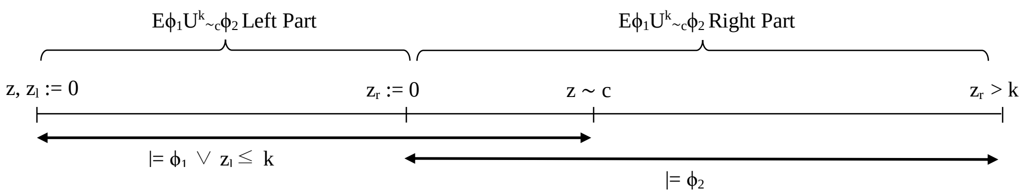

Figure 1. Illustration of $E\phi_1 U^k_{\sim c}\phi_2$ Modality

We proved in [1] that the characteristic set of $E\phi_1 U^k_{\sim c}\phi_2$ is given as the least upper bound of the following stationary and increasing (by inclusion) sequence:

$$\begin{cases} X_0 = [[RP(E\phi_1 U^k{}_{\sim c}\phi_2)]] \\ X_{n+1} = X_n \vee (([[\phi_1]] \triangleright [z_l \leftarrow 0]X_n) \vee ([[(\neg\phi_1 \wedge z_l \leq k)]] \triangleright X_n)) \end{cases}$$

Where $RP(E\phi_1 U^k_{\sim c}\phi_2)$ denotes the right part modality of $E\phi_1 U^k_{\sim c}\phi_2$, given as follows:

$$[[RP(E\phi_1 U^k_{\sim c}\phi_2)]] = [z_r \leftarrow 0]\mathrm{Sup}\ Y_n$$

Such that Sup Y_n denotes the least upper bound of the sequence Y_n. We define the also stationary and increasing sequence Y_n as:

$$\begin{cases} Y_0 = [[(z \sim c) \wedge (E\ \phi_2\ U\ (\phi_2 \wedge z_r > k))]] \\ \end{cases}$$

$$Y_{n+1} = Y_n \vee (([[\phi_2 \wedge \phi_1]] \rhd [z_l \leftarrow 0]Y_n) \vee ([[\phi_2 \wedge (\neg\phi_1 \wedge z_l \leq k)]] \rhd Y_n$$

Note that z, z_l are reset when the stationary value of the sequence X_n is reached, *i.e.* after that the set of symbolic states satisfying $E\phi_1U^k{}_{\sim c}\phi_2$ is computed.

Predecessor operator $\rhd$:

The predecessor operator $\rhd$is defined as follows [1]:

Given a TA A, a TTS T = (S, s_{init},→, l), an alphabet Σ which denotes a finite set of actions and two characteristic sets Q_1 and Q_2. Calculate $Q_1 \rhd Q_2$ is to determine:

–All the instantaneous predecessors of Q_2 states that verify Q_1, *i.e.* the states satisfying Q_1 and can reach Q_2 by an action transition denoted $Q_1 \rhd_a Q_2$.

– Union, all temporal predecessors of Q_2 that verify Q_1, *i.e.* all states that can reach a state of Q_2 by a delay transition, such that all intermediates states are in Q_1:

$$q \in Q_1 \rhd_t Q_2 \Leftrightarrow q \in Q_1 \wedge \exists t > 0 \text{ s.t. }$$
$$q + t \in Q_2 \text{ and } \forall t' < t \;\; q + t' \in Q_1$$

4.2. Modality $A\phi_1U^k{}_{\sim c}\phi_2$

For this modality, we distinguish between cases according to the signs of “z ∼ c”.

4.2.1. $A\phi_1U^k\phi_2$

This modality indicates that along any path, ϕ_2 lasts long enough (> k) and before, ϕ_1 is true everywhere except along paths having negligible durations (≤ k). In other words, $A\phi_1U^k\phi_2$ ensures that (1) along any path, eventually ϕ_2 holds for at least k t.u., and (2) it is not possible to have $\neg\phi_1$ for k t.u. unless either ϕ_2 has been verified for k t.u. before, or ϕ_2 is true and will hold for k t.u. The negation of (2) is depicted in the following figure (Figure 2):

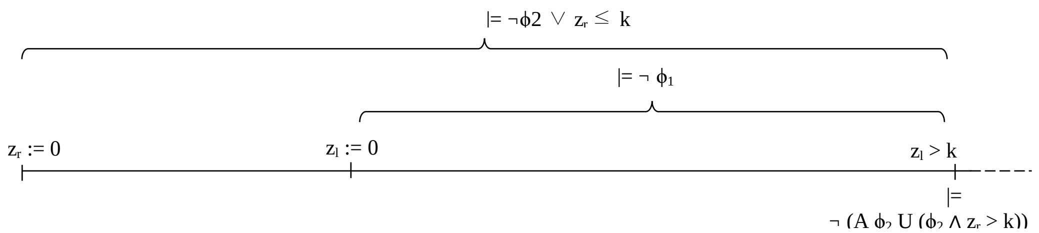

Figure 2. Illustration of $A\phi_1U^k\phi_2$ Property (2) Negation

We prove that the characteristic set of $A\phi_1U^k\phi_2$ is given as follows:

$$[[A\phi_1U^k\phi_2]] = [[AF([z_r \leftarrow 0](E\ \phi_2\ U\ (\phi_2 \wedge z_r > k)))]] \wedge [z_r \leftarrow 0]\neg\text{Sup } X_n$$

Where ¬Sup X_n denotes the negation of the least upper bound of the sequence X_n. The stationary and increasing (by inclusion) sequence X_n represents the negation of property (2) (as depicted in the figure above), and is defined by:

$$\begin{cases} X_0 = [z_l \leftarrow 0]\text{Sup } Y_n \\ X_{n+1} = X_n \vee (([[\neg\phi_2]] \rhd [z_r \leftarrow 0]X_n) \vee ([[(\phi \wedge z_r \leq k)]] \rhd X_n)) \end{cases}$$

And Y_n is also a stationary and increasing (by inclusion) sequence, that represents the first term of X_n (as depicted in the figure above). The sequence Y_n is defined as:

$$\begin{cases} Y_0 = [[(\neg\phi_1 \wedge z_l > k) \wedge \neg(A\ \phi_2\ U\ (\phi_2 \wedge z_r > k))]] \\ Y_{j+1} = Y_j \vee (([[\neg\phi_1 \wedge \neg\phi_2]] \rhd [z_r \leftarrow 0]Y_j) \vee ([[\neg\phi_1 \wedge (\phi_2 \wedge z_r \leq k)]] \rhd Y_j)) \end{cases}$$

Proof [sketch.] We have to show that:

$$[[A\phi_1U^k\phi_2]] = [[AF([z_r \leftarrow 0](E\ \phi_2\ U\ (\phi_2 \wedge z_r > k)))]] \wedge [z_r \leftarrow 0]\neg\text{Sup } X_n$$

$\subseteq$ / Let $q \in [[A\phi_1 U^k \phi_2]]$:

• We have obviously: $q \in [[AF([z_r \leftarrow 0](E\ \phi_2\ U\ (\phi_2 \wedge z_r > k)))]]$

We show now that $q \in \neg Sup\ X_n$ (Note that we can prove that the sequences X_n and Y_n are stationary and increasing by inclusion in the same way as shown in [1]).

Suppose that $q \in Sup\ X_n$. As X_n is stationary, so $\exists k \in N$, s.t. $Sup\ X_n = X_k$. Then $q \in X_k$:

<u>if k = 0 :</u> then $q \in X_0 = [z_r \leftarrow 0][z_l \leftarrow 0]Sup\ Y_n$ s.t :

$$\begin{cases} Y_0 = [[(\neg\phi_1 \wedge z_l > k) \wedge \neg(A\ \phi_2\ U\ (\phi_2 \wedge z_r > k))]] \\ Y_{j+1} = Y_j \vee (([[\neg\phi_1 \wedge \neg\phi_2]] \triangleright [z_r \leftarrow 0]Y_j) \vee ([[\neg\phi_1 \wedge (\phi_2 \wedge z_r \leq k)]] \triangleright Y_j)) \end{cases}$$

Therefore there is a path from q that satisfies all the time $\neg\phi_1 \wedge (\neg\phi_2 \vee z_r \leq k)$ until it reaches a position that satisfies $\neg\phi_1 \wedge z_l > k$, and there is at least one path from that position $\models \neg(\phi_2\ U\ (\phi_2 \wedge z_r > k))$. Then $q \notin [[A\phi_1 U^k \phi_2]]$, contradiction. So $q \in \neg Sup\ X_n$.

<u>if k ≠ 0 :</u> then $q \in X_k$ s.t $k \neq 0$,

$$\begin{cases} X_0 = [z_l \leftarrow 0]Sup\ Y_n \\ X_{n+1} = X_n \vee (([[\neg\phi_2]] \triangleright [z_r \leftarrow 0]X_n) \vee ([[(\phi_2 \wedge z_r \leq k)]] \triangleright X_n)) \end{cases}$$

i.e. there exists a path from q that satisfies all the time $(\neg\phi_2 \vee z_r \leq k)$ until reaching a state $q' \in X_0$. Then $q \notin [[A\phi_1 U^k \phi_2]]$, contradiction. So $q \in \neg Sup\ X_n$. Therefore, we have:

$$[[A\phi_1 U^k \phi_2]] \subseteq [[AF([z_r \leftarrow 0](E\ \phi_2\ U\ (\phi_2 \wedge z_r > k)))]] \wedge [z_r \leftarrow 0]\neg Sup\ X_n$$

$\supseteq$ / Let $q \in [[AF([z_r \leftarrow 0](E\ \phi_2\ U\ (\phi_2 \wedge z_r > k)))]] \wedge [z_r \leftarrow 0]\neg Sup\ X_n$:

• As $q \in [[AF([z_r \leftarrow 0](E\ \phi_2\ U\ (\phi_2 \wedge z_r > k)))]]$, then for every path from q there exists a sub path σ where ϕ_2 lasts at least k t.u.

• On another side, $q \in \neg Sup\ X_n$, this certifies that it is not possible that $\neg\phi_1$ lasts long (> k) before the sub path σ, unless either $\phi 2$ has been verified for k t.u. before, or $\phi 2$ is true and will hold for k t.u. (*i.e.* $q \in [[A\phi_1 U^k \phi_2]]$, then we have:

$$[[A\phi_1 U^k \phi_2]] \supseteq [[AF([z_r \leftarrow 0](E\ \phi_2\ U\ (\phi_2 \wedge z_r > k)))]] \wedge [z_r \leftarrow 0]\neg Sup\ X_n$$

Finally, we have:

$$[[A\phi_1 U^k \phi_2]] = [[AF([z_r \leftarrow 0](E\ \phi_2\ U\ (\phi_2 \wedge z_r > k)))]] \wedge [z_r \leftarrow 0]\neg Sup\ X_n$$

4.2.2. $A\phi_1 U^k_{<c}\phi_2$

This modality indicates that along any path ϕ_2 lasts long enough (> k) around a position at a distance < c, and before this position ϕ_1 is true everywhere except along paths having negligible durations ($\leq$ k). In other words, $A\phi_1 U^k{<}c\phi_2$ ensures that (1) along any path, eventually ϕ_2 holds for at least k t.u. around a position at a distance < c ($AF^k{<}c\phi_2$), and (2) it is not possible to have $\neg\phi_1$ for k t.u. unless either ϕ_2 has been verified for k t.u. before, or ϕ_2 is true and will hold for k t.u.

Note that in the case of $\sim \in \{<, \leq\}$, it is necessary and sufficient that $z \sim c$ be verified at the beginning of the subrun where ϕ_2 lasts long enough (> k), that is why the condition (2) described above did not change from that described in the formula $A\phi_1 U^k \phi_2$.

For dealing with this case, we first consider the formula $AF^k{<}c\phi_2$ and more precisely we consider the dual modality $EG^k{<}c$.

We have: $EG^k_{<c}\ \neg\phi_2 = \neg AF^k_{<c}\phi_2$, then: $s \models EG^k_{<c}\ \neg\phi_2 \Leftrightarrow \exists\rho \in Exec(s)\ |\ \forall\sigma \in subrun(\rho) : \mu(\sigma) > k \Rightarrow (\forall p \in \sigma, Time(\rho^{\leq p}) \sim c) \vee (\exists p \in \sigma \text{ s.t. } sp \models \neg\phi_2)$

$EG^k_{<c}\ \neg\phi_2$ expresses that there exists an execution (from the current state s) where any subrun σ s.t. (a) $\mu(\sigma) > k$ and (b) σ contains states located before c t.u. from s, contains a state satisfying $\neg\phi_2$. Thus states satisfying $\neg\phi_2$ have to occur "often" (at least every k t.u.) during c + k t.u. The formula $EG^k_{<c}\ \neg\phi_2$ is depicted in the following figure (Figure 3):

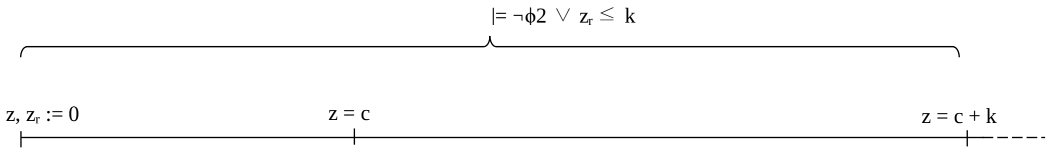

Figure 3. Illustration of $EG^{k}_{<c}\neg\phi_2$ Modality

Therefore, we prove that the characteristic set of $AF^{k}_{<c}\phi_2$ is given as follows:

$$[[AF^{k}_{<c}\phi_2]] = [z \leftarrow 0][z_r \leftarrow 0]\neg Sup\ V_n$$

Where V_n is a stationary and increasing (by inclusion) sequence, that represents $EG^{k}_{<c}\ \neg\phi_2$ (as depicted in the figure above). The sequence V_n is defined as:

$$\begin{cases} V_0 & = [[z = c + k]] \\ V_{n+1} & = V_n \vee (([[\neg\phi_2]] \rhd [z_r \leftarrow 0]V_n) \vee ([[(\phi_2 \wedge z_r \leq k)]] \rhd V_n)) \end{cases}$$

Finally, the characteristic set of $A\phi_1 Uk_{<c}\phi_2$ is given as follows:

$$[[A\phi_1 U^{k}_{<c}\phi_2]] = [z \leftarrow 0][z_r \leftarrow 0]\neg Sup\ V_n \wedge [z_r \leftarrow 0]\neg Sup\ X_n$$

Proof [sketch.] For this modality, we have to show that:

$$[[A\phi_1 U^{k}_{<c}\phi_2]] = [z \leftarrow 0][z_r \leftarrow 0]\neg Sup\ V_n \wedge [z_r \leftarrow 0]\neg Sup\ X_n$$

$\subseteq$ / Let $q \in [[A\phi_1 U^{k}_{<c}\phi_2]]$

• We show now that $q \in [z \leftarrow 0][z_r \leftarrow 0]\neg Sup\ V_n$ (Note that we can prove that the sequence V_n is stationary and increasing by inclusion in the same way as shown in [1]).

Suppose that $q \in [z \leftarrow 0][z_r \leftarrow 0]Sup\ V_n$ s.t:

$$\begin{cases} V_0 & = [[z = c + k]] \\ V_{n+1} & = V_n \vee (([[\neg\phi_2]] \rhd [z_r \leftarrow 0]V_n) \vee ([[(\phi_2 \wedge z_r \leq k)]] \rhd V_n)) \end{cases}$$

Then there is a path from q that reaches the position $z = c+k$, without verifying before $\phi 2$ long enough around a position $z < c$. contradiction. So $q \in \neg Sup\ V_n$

• We show in the same way as $A\phi_1 Uk\phi_2$ that $q \in [z_r \leftarrow 0]\neg Sup\ X_n$.

$\supseteq$ / Let $q \in [z \leftarrow 0][z_r \leftarrow 0]\neg Sup\ V_n \wedge [z_r \leftarrow 0]\neg Sup\ X_n$

• As $q \in [z \leftarrow 0][z_r \leftarrow 0]\neg Sup\ V_n$, then along all path from q there exists a sub-path satisfying ϕ_2 long enough, before reaching the position $c + k$. Therefore, for every path from q there is a sub path σ of length ($> k$) where ϕ_2 is true, and this path inevitably contains a position located strictly before c u.t.

• On another side, $q \in [z_r \leftarrow 0]\neg Sup\ X_n$, this certifies that it is not possible that $\neg\phi_1$ lasts long ($> k$) before the sub path σ, unless either ϕ_2 has been verified for k t.u. before, or ϕ_2 is true and will hold for k t.u. (*i.e.* $q \in [[A\phi_1 Uk{<}c\phi_2]]$, then we have:

$$[[A\phi_1 U^{k}_{<c}\phi_2]] = [z \leftarrow 0][z_r \leftarrow 0]\neg Sup\ V_n \wedge [z_r \leftarrow 0]\neg Sup\ X_n$$

The pseudo-code version of the Model-Checking algorithm for this modality is shown in Algorithm 1 (Section 5).

4.2.3. $A\phi_1 U^{k}_{\leq c}\phi_2$

This case is very similar to the previous one. The dual formula $EG^{k}_{\leq c}\ \neg\phi_2$ is depicted in the following figure (Figure 4):

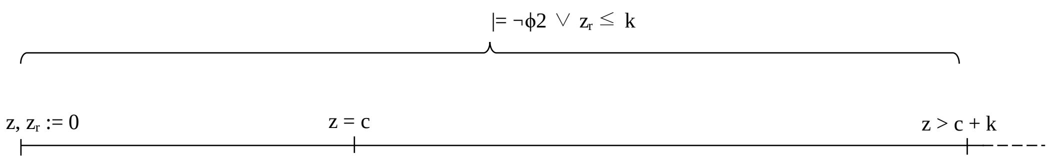

Figure 4. Illustration of $EG^{k}_{\leq c}\ \neg\phi 2$ Modality

Therefore, we prove that the characteristic set of $A\phi_1 U^k_{\leq c}\phi_2$ is given as follows:

$$[[A\phi_1 U^k_{\leq c}\phi_2]] = [z \leftarrow 0][z_r \leftarrow 0]\neg \mathrm{Sup}\ V'_n \wedge [z_r \leftarrow 0]\neg \mathrm{Sup}\ X_n$$

Where V'_n is a stationary and increasing (by inclusion) sequence, that represents $EG^k{}_{\leq c}$ $\neg\phi 2$ (as depicted in the figure above). The sequence V'_n is defined as:

$$\begin{cases} V'_0 & = [[z > c + k]] \\ V'_{n+1} & = V'_n \vee (([[\neg\phi 2]] \rhd [z_r \leftarrow 0]V'_n) \vee ([[(\phi_2 \wedge z_r \leq k)]] \rhd V'_n)) \end{cases}$$

Proof [sketch.] The same previous proof can be adapted to show that:

$$[[A\phi_1 U^k \leq c\phi_2]] = [z \leftarrow 0][z_r \leftarrow 0]\neg \mathrm{Sup}\ V'_n \wedge [z_r \leftarrow 0]\neg \mathrm{Sup}\ X_n$$

4.2.4. $A\phi_1 U^k_{\sim c}\phi_2$: $\sim \in \{>, \geq\}$

This modality indicates that along any path ϕ_2 lasts long enough ($> k$) around a position at a distance $\sim$ c, and before this position ϕ_1 is true everywhere except along paths having negligible durations ($\leq$ k). In other words, $A\phi_1 U^k{}_{\sim c}\phi_2$ ensures that:

(1) along any path, eventually ϕ_2 holds for at least k t.u. around a position at a distance $\sim$ c, note that in this case it is necessary and sufficient that $z \sim c$ be verified at the end of the sufficiently long subrun ($> k$) where ϕ_2 is true,

$$U_1 = [[[z \leftarrow 0]AF([z_r \leftarrow 0]E\phi_2 U(\phi_2 \wedge (z_r > k) \wedge (z \sim c))]]),$$

(2) it is not possible to have $\neg\phi_1$ for k t.u. in a position satisfying $\neg(z \sim c)$

$$U_2 = [[[z \leftarrow 0]\neg E(true)U([z_l \leftarrow 0]E\neg\phi_1 U(\neg\phi_1 \wedge (z_l > k) \wedge \neg(z \sim c)))]], \text{ and}$$

(3) it is not possible to have $\neg\phi_1$ for k t.u. in a position at a distance $\sim$ c unless either ϕ_2 has been verified for k t.u. around a position at a distance $\sim$ c before, or ϕ_2 is true and will hold for k t.u. The negation of (3) is depicted in the following figure (Figure 5):

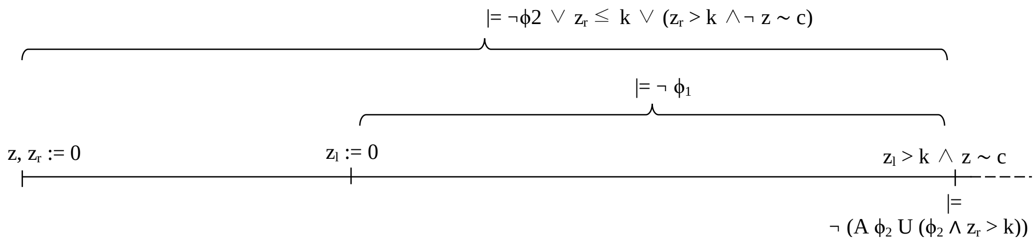

Figure 5. Illustration of $A\phi_1 U^k{}_{>,\geq c}\phi_2$ Property (3) Negation

We prove that the characteristic set of $A\phi_1 U^k_{\sim c}\phi_2$ is given as follows:

$$[[A\phi_1 U^k_{\sim c}\phi_2]] = U_1 \wedge U_2 \wedge [z \leftarrow 0][z_r \leftarrow 0]\neg \mathrm{Sup}\ X_n$$

Where X_n is a stationary and increasing (by inclusion) sequence, that represents the negation of (3) (as depicted in the figure above). The sequence X_n is defined as :

$$\begin{cases} X_0 & = [z_l \leftarrow 0]\mathrm{Sup}\ Y_n \\ X_{n+1} & = X_n \vee (([[\neg\phi_2]] \rhd [z_r \leftarrow 0]X_n) \vee ([[(\phi_2 \wedge z_r \leq k)]] \rhd X_n) \vee \\ & \quad ([[(\phi_2 \wedge z_r > k) \wedge \neg(z \sim c)]] \rhd X_n)) \end{cases}$$

And Y_n is also a stationary and increasing (by inclusion) sequence, that represents the first term of X_n (as depicted in the figure above). The sequence Yn is defined as:

$$\begin{cases} Y_0 & = [[\neg\phi_1 \wedge (z_l > k) \wedge (z \sim c) \wedge \neg(A\ \phi_2\ U\ (\phi_2 \wedge z_r > k))]] \\ Y_{j+1} & = Y_j \vee (([[\neg\phi_1 \wedge \neg\phi_2]] \rhd [z_r \leftarrow 0]Y_j) \vee ([[\neg\phi_1 \wedge (\phi_2 \wedge z_r \leq k)]] \rhd Y_j) \vee \\ & \quad ([[\neg\phi_1 \wedge (\phi_2 \wedge z_r > k) \wedge \neg(z \sim c)]] \rhd Y_j)) \end{cases}$$

Proof [sketch.] For this modality, we have to show that:

$$[[A\phi_1 U^k_{\sim c}\phi_2]] = U_1 \wedge U_2 \wedge [z \leftarrow 0][z_r \leftarrow 0]\neg \mathrm{Sup}\ X_n$$

$\subseteq$ / Let $q \in [[A\phi_1 U^k_{\sim c}\phi_2]]$

• It is obvious to see that for every path from q ϕ_2 lasts long enough (> k) around a position satisfying $z \sim c$, and so at the end of this sub path we have always $z \sim c$.

Then $q \in U_1$, such that:

$$U_1 = [[[z \leftarrow 0]AF([z_r \leftarrow 0]E\phi_2 U(\phi_2 \wedge (z_r > k) \wedge (z \sim c))]])$$

• Suppose that $q \in \neg U_2$ with

$$U_2 = [[[z \leftarrow 0]\neg E(true)U([z_l \leftarrow 0]E\neg\phi_1 U(\neg\phi_1 \wedge (z_l > k) \wedge \neg(z \sim c)))]]$$

So there exists a path from q containing a position located at $\neg \sim c$, where $\neg\phi_1$ lasted long enough, contradiction because $q \in [[A\phi_1 U^k_{\sim c}\phi_2]]$.

Then we have $q \in U_2 = [[[z \leftarrow 0]\neg E(true)U([z_l \leftarrow 0]E\neg\phi_1 U(\neg\phi_1 \wedge (z_l > k) \wedge \neg(z \sim c)))]]$,

We show now that $q \in [z \leftarrow 0][z_r \leftarrow 0]\neg \mathrm{Sup}\ X_n$ s.t. X_n is a recurrent sequence defined as:

$$\begin{cases} X_0 = [z_l \leftarrow 0]\ \mathrm{Sup}\ Y_n \\ X_{n+1} = X_n \vee (([[\neg\phi_2]] \rhd [z_r \leftarrow 0]X_n) \vee ([[(\phi_2 \wedge z_r \leq k)]] \rhd X_n) \vee \\ \qquad ([[\phi_2 \wedge (z_r > k) \wedge \neg(z \sim c)]] \rhd X_n)) \end{cases}$$

And Y_n is also a recurrent sequence defined as:

$$\begin{cases} Y_0 = [[\neg\phi_1 \wedge (z_l > k) \wedge (z \sim c) \wedge \neg(A\ \phi_2\ U\ (\phi_2 \wedge z_r > k))]] \\ Y_{j+1} = Y_j \vee (([[\neg\phi_1 \wedge \neg\phi_2]] \rhd [z_r \leftarrow 0]Y_j) \vee ([[\neg\phi_1 \wedge (\phi_2 \wedge z_r \leq k)]] \rhd Y_j) \\ \qquad \vee ([[\neg\phi_1 \wedge (\phi_2 \wedge z_r > k) \wedge \neg(z \sim c)]] \rhd Y_j)) \end{cases}$$

First of all, we can prove that the sequences X_n and Y_n are stationary and increasing by inclusion in the same way as shown in [1].

Suppose that $q \in \mathrm{Sup}\ X_n$. As X_n is stationary, so $\exists k \in N$, s.t. $\mathrm{Sup}\ X_n = X_k$. Then $q \in X_k$:

if k = 0 : then $q \in X_0 = [z \leftarrow 0][z_r \leftarrow 0][z_l \leftarrow 0]\mathrm{Sup}\ Y_n$, *i.e.* there exists a path from q, that satisfies all the time $\neg\phi_1 \wedge \neg\phi_2 \vee z_r \leq k \vee ((z_r > k) \wedge \neg(z \sim c))$ until reaching a position that satisfies $(\neg\phi_1 \wedge (z_l > k) \wedge (z \sim c))$. And there is at least one path from that position $\models \neg(\phi_2\ U\ (\phi_2 \wedge z_r > k))$. Contradiction. So $q \in \neg \mathrm{Sup}\ X_n$.

if $k \neq 0$: then $q \in X_k$ s.t k = 0, *i.e.* there exists a path from q that satisfies all the time $(\neg\phi_2 \vee z_r \leq k \vee (z_r > k \wedge \neg z \sim c))$ until reaching a state $q' \in X_0$. Then $q \notin [[A\phi_1 U^k_{\sim c}\ \phi_2]]$, contradiction. So $q \in \neg \mathrm{Sup}\ X_n$. Therefore, we have:

$$[[A\phi_1 U^k_{\sim c}\phi_2]] \subseteq U_1 \wedge U_2 \wedge [z \leftarrow 0][z_r \leftarrow 0]\neg \mathrm{Sup}\ X_n$$

$\supseteq$ / Let $q \in U_1 \wedge U_2 \wedge [z \leftarrow 0][z_r \leftarrow 0]\neg \mathrm{Sup}\ X_n$

• As $q \in U_1 = [[[z \leftarrow 0]AF([z_r \leftarrow 0]E\phi_2 U(\phi_2 \wedge (z_r > k) \wedge (z \sim c))]])$, then for every path from q there exists a sub path σ where ϕ_2 lasts long enough around a position satisfying $z \sim c$.

• On another side $q \in U_2 = [[[z \leftarrow 0]\neg E(true)U([z_l \leftarrow 0]E\neg\phi_1 U(\neg\phi_1 \wedge (z_l > k) \wedge \neg(z \sim c)))]]$, *i.e.* it is not possible to have a sub path from all path starting from q where $\neg\phi_1$ lasts long enough before the position $z \sim c$.

• We also have $q \in [z \leftarrow 0][z_r \leftarrow 0]\neg \mathrm{Sup}\ X_n$, this certifies that it is not possible that $\neg\phi_1$ lasts long (> k) after the position $\sim c$, unless either ϕ_2 has been verified for k t.u. around a position satisfying $z \sim c$ before, or ϕ_2 is true and will hold for k t.u. (*i.e.*, $q \in [[A\phi_1 U^k_{\sim c}\phi_2]]$), then we have:

$$U_1 \wedge U_2 \wedge [z \leftarrow 0][z_r \leftarrow 0]\neg \mathrm{Sup}\ X_n \subseteq [[A\phi_1 U^k_{\sim c}\phi_2]]$$

Finally, we have:

$$[[A\phi_1 U^k_{\sim c}\phi_2]] = U_1 \wedge U_2 \wedge [z \leftarrow 0][z_r \leftarrow 0]\neg \mathrm{Sup}\ X_n$$

4.2.5. $A\phi_1 U^k_{=c}\phi_2$

In this case, we use the following equivalences [16] in order to reduce the model-checking algorithm to previous modalities:

$$A\phi_1 U^k_{=c}\phi_2 \equiv AF^k_{=c}\phi_2 \wedge AG^k_{\leq c-k}(\phi_1) \quad \text{if } c \geq k$$
$$A\phi_1 U^k_{=c}\phi_2 \equiv AF^k_{=c}\phi_2 \quad \text{if } c < k$$

For, $AG^k_{\leq c-k}(\phi_1)$ we have : $AG^k_{\leq c}(\phi_1) = \neg EF^k_{\leq c}\ \neg\phi_1$ (algorithm already given for this modality). So it remains to give the zone algorithm for $AF^k_{=c}\phi_2$.

For dealing with this case, we first consider the dual modality $EG^k_{=c}$.

We have: $EG^k_{=c}\neg\phi_2 = \neg AF^k_{=c}\phi_2$, then $s \models EG^k_{=c}\neg\phi_2 \Leftrightarrow \exists\rho \in Exec(s) \mid \forall\sigma \in subrun(\rho)$: $\mu(\sigma) > k \Rightarrow (\forall p \in \sigma, Time(\rho^{\leq p} \neq c)) \vee (\exists p \in \sigma \text{ s.t. } sp \models \neg\phi_2)$.

$EG^k_{=c}\neg\phi_2$ expresses that there exists an execution from the current state s where any subrun σ s.t. (a) $\mu(\sigma) > k$ and (b) σ contains a state located at duration c from s, contains a state satisfying $\neg\phi_2$. Thus we have to verify that there exists an execution where $\neg\phi2$ holds or has held "recently" (*i.e.* in less than k t.u. ago) for any state located at a duration in [c; c + k].

The formula $EG^k_{=c}\ \neg\phi_2$ is depicted in the following figure (Figure 6):

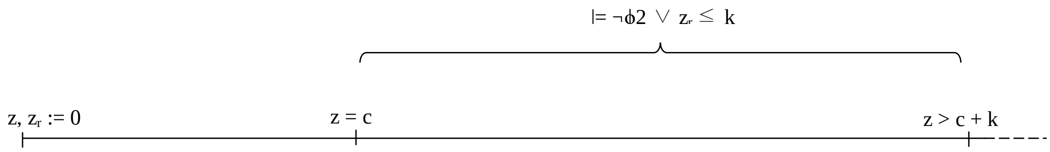

Figure 6. Illustration of $EG^k_{=c}\neg\phi_2$ Modality

We prove that the characteristic set of AFk $_{=c}\phi_2$ is given as follows:

$$[[AF^k_{=c}\phi_2]] = [z \leftarrow 0][z_r \leftarrow 0]\neg \text{Sup } X_n$$

Where X_n is a stationary and increasing (by inclusion) sequence, that represents $EG^k_{=c}\neg\phi_2$ (as depicted in the figure above). The sequence X_n is defined as :

$$\begin{cases} X_0 = [[z_{=c}]] \wedge \text{Sup } Y_n \\ X_{n+1} = X_n \vee ((\ [[(z < c) \wedge \neg\phi_2]] \triangleright [z_r \leftarrow 0]X_n\ ) \vee ([[(z < c) \wedge \phi_2]] \triangleright X_n\ )) \end{cases}$$

And Y_n is also a stationary and increasing (by inclusion) sequence, that represents the first term of X_n (as depicted in the figure above). The sequence Y_n is defined as:

$$\begin{cases} Y_0 = [[z > c + k]] \\ Y_{j+1} = Yj \vee (([[\neg\phi_2 \wedge z \geq c]] \triangleright [z_r \leftarrow 0]Y_j) \vee ([[(\phi_2 \wedge z_r \leq k) \wedge z \geq c]] \triangleright Y_j\ )) \end{cases}$$

Proof [sketch.] For this modality, we have to show that:

$$[[AF^k_{=c}\phi_2]] = [z \leftarrow 0][z_r \leftarrow 0]\neg \text{Sup } X_n$$

s.t. X_n is a recurrent sequence defined as :

$$\begin{cases} X_0 = [[z = c]] \wedge \text{Sup } Y_n \\ X_{n+1} = X_n \vee ((\ [[(z < c) \wedge \neg\phi_2]] \triangleright [z_r \leftarrow 0]X_n\ ) \vee ([[(z < c) \wedge \phi_2]] \triangleright X_n)) \end{cases}$$

And Y_n is also a recurrent sequence defined as:

$$\begin{cases} Y_0 = [[z > c + k]] \\ Y_{j+1} = Y_j \vee ((\ [[\neg\phi_2 \wedge z \geq c]] \triangleright [z_r \leftarrow 0]Y_j) \vee ([[(\phi_2 \wedge z_r \leq k) \wedge z \geq c]] \triangleright Y_j)) \end{cases}$$

First of all, we can prove that the sequences X_n and Y_n are stationary and increasing by inclusion in the same way as shown in [1].

$\subseteq$ / Let $q \in [[AF^k_{=c}\phi_2]]$, suppose that $q \in [z \leftarrow 0][z_r \leftarrow 0]\text{Sup } X_n$, therefore there exists a path from q such that any position between c and c + k satisfying $\neg\phi_2 \vee z_r \leq k$. This clearly contradicts the fact that $q \in [[AF^k_{=c}\phi_2]]$, *i.e.* $q \in [z \leftarrow 0][z_r \leftarrow 0]\neg\text{Sup } X_n$.

$\supseteq$ / Let $q \in [z \leftarrow 0][z_r \leftarrow 0]\neg\mathrm{Sup}\ X_n$. Suppose that $q \notin [[AF^k_{=c}\phi 2]]$, then $q \in EG^k_{=c}\ \neg\phi_2$. Then, there exists a path ρ from q, such that all sub path σ having length > k and containing a configuration located at c time units from q, must contain a position where $\neg\phi_2$ is true. Now consider the sub path ρ' of ρ from the position z = c, clearly this sub path verifies $(\neg\phi_2 \vee z_r \leq k) \vee (z > c + k)$ this is a contradiction with the fact that $q \in [z \leftarrow 0][z_r \leftarrow 0]\neg\mathrm{Sup}\ X_n$.

Finally, we have:

$$[[AF^k_{=c}\phi_2]] = [z \leftarrow 0][z_r \leftarrow 0]\neg\mathrm{Sup}\ X_n$$

4.3. Implementation of Algorithms using DBMs

The DBM acronym means difference bounded matrice. It is a classical data structure widely used for representing systems of difference constraints, which has a significant interest for the verification of timed systems because they can be used to represent zones. DBMs are now intensively used to analyze timed automata [18]. Moreover, the DBMs are appropriate to implement algorithms proposed in the previous subsection. Indeed, we have shown in [1] how to compute, using the DBMs, all operations on zones appearing in the model-checking algorithms of $TCTL^{\Delta}$ inevitability modalities. We also gave in [1] an effective method for computing the operation $Q_1 \triangleright Q_2$.

5. Pseudo-Codes for $TCTL^{\Delta}$ Model-Checking Algorithms

We give here the pseudo-code version of the Model-Checking algorithm for $A\phi_1 Uk_{<c}\phi_2$. The algorithms' pseudo-codes of the other inevitability modalities can be developed exactly with the same approach, based on the results of subsection 4.2.

Algorithm 1 computes step-by-step the characteristic of the modality $A\phi_1 U^k_{<c}\phi_2$, using a backward analysis approach we have seen in subsection 4.2.2. We start by computing the least upper bound of the sequence Y_n. The first term of Y_n is given as the characteristic set of a classical TCTL formula. Then we compute iteratively the terms of Y_n until reaching a stationary value which is obviously the least upper bound of Y_n. Similarly, we compute the least upper bound of the sequence X_n. The stop condition of X_n's iterations is also given by convergence to its stationary value. Then we compute the least upper bound of the sequence V_n. The first term of V_n is given as the characteristic set of a simple clock constraint. After, we compute iteratively the terms of V_n until reaching its stationary value, which is evidently its least upper bound. Finally, the characteristic set of formula $A\phi_1 U^k{}_{<c}\phi_2$ is given by the intersection of X_n's least upper bound negation and V_n's least upper bound negation.

We note that all operations used in this algorithms (intersection of sets of symbolic states, predecessor operators and clocks reset, …) are reduced to known operations on zones. These operations are easily implemented through DBM data structure as we have shown in [1].

Algorithm 1 Model-Checking of $A\phi_1 Uk_{<c}\phi_2$ Modality

```
function Characteristic Set(Aϕ1Uk<cϕ2 : TCTL^Δ)

// (* TCTL formula *)
TargetSetYn := [[ (¬ϕ1 ∧ zl > k) ∧¬(A ϕ2 U (ϕ2 ∧ zr > k))]];

repeat
    CurrentSet    := TargetSetYn;
    TargetSetYn := TargetSetYn ∪ CurrentSet;
    TargetSetYn := TargetSetYn ∪ ([[¬ϕ1 ∧¬ϕ2]] ▷[zr← 0] CurrentSet);
    TargetSetYn := TargetSetYn ∪ ([[¬ϕ1 ∧ (ϕ2 ∧ zr ≤ k)]] ▷CurrentSet);
until TargetSetYn = CurrentSet
```



```
TargetSetXn := [zl← 0] TargetSetYn;
repeat
    CurrentSet    := TargetSetXn;
    TargetSetXn := TargetSetXn ∪ CurrentSet;
    TargetSetXn := TargetSetXn ∪ ([[¬ϕ2]] ▷[zr ← 0] CurrentSet);
    TargetSetXn := TargetSetXn ∪ ([[(ϕ2 ∧ zr ≤ k)]] ▷CurrentSet);
until TargetSetXn = CurrentSet
TargetSetVn := [[z = c + k]];
repeat
    CurrentSet    := TargetSetVn;
    TargetSetVn := TargetSetVn ∪ CurrentSet;
    TargetSetVn := TargetSetVn ∪ ([[¬ϕ2]] ▷[zr ← 0] CurrentSet);
    TargetSetVn := TargetSetVn ∪ ([[(ϕ2 ∧ zr ≤ k)]] ▷CurrentSet);
until TargetSetVn = CurrentSet
TargetSet := [z ← 0][zr ← 0]¬ TargetSetVn ∩[zr ← 0]¬ TargetSetXn;
return TargetSet;
end function
```

6. Conclusion

In this paper, we proposed implementable model-checking algorithms for TCTL$^{\Delta}$ inevitability modalities. We presented a complete correctness proof for each proposed procedure. The main result of this paper is the overcome of the state-space explosion problem caused by the theoretical TCTL$^{\Delta}$ model-checking algorithm based on regions.

Moreover, we have described the implementation of our algorithms using zones and DBMs, which is the same approach as the one used in tools like UPPAAL or KRONOS. Furthermore, this paper completes the study started in [1], regarding the reachability modality $EU^{k}_{\sim c}$. Indeed, no much work is now necessary to get a model-checker that deals with all TCTL$^{\Delta}$ modalities.

References

[1] M. A. Begdouri, H. B. Mokadem and M. E. Haddad, "An algorithmic approach for abstracting transient states in timed systems", International Journal of Advanced Computer Science and Applications, vol. 7, no. 5, **(2016)**, pp. 500–509.

[2] P. Bouyer, U. Fahrenberg, K. G. Larsen, N. Markey, J. Ouaknine and J. Worrell, "Model checking real-time systems", Handbook of Model Checking, Springer, **(2016)**.

[3] L. Guozheng, C. Zining and G. Zheng, "Approximated model checking for multirate hybrid zia", International Journal of Control and Automation, vol. 9, no. 2, **(2016)**, pp. 271–286.

[4] M. A. Begdouri and H.B. Mokadem, "Verification of a timed concurrent system with Uppaal", In 11th International Pluridisciplinary Congress on Quality, Dependability and Sustainability (QUALITA' 2015), Nancy, France, **(2015)**.

[5] P. Bouyer, K. G. Larsen and N. Markey, "Lower-bound constrained runs in weighted timed automata", Performance Evaluation, vol. 73, **(2014)**, pp. 91–109.

[6] P. Bouyer, N. Markey and O. Sankur, "Robustness in timed automata", In Proceedings of the 7th Workshop on Reachability Problems in Computational Models (RP'13), volume 8169, Springer, **(2013)**.

[7] G. Behrmann, A. David, K. G. Larsen, P. Pettersson and W. Yi, "Developing Uppaal over 15 years", Software: Practice and Experience, vol. 41, no. 2, **(2011)**, pp. 133–142.

[8] P. Bouyer, U. Fahrenberg, K. G. Larsen and N. Markey, "Quantitative analysis of realtime systems using priced timed automata", Communications of the ACM, vol. 54, no. 9, **(2011)**, pp. 78–87.

[9] L. Bu, Y. Li, L. Wang and X. Li, "Bach: A toolset for bounded reachability analysis of linear hybrid systems", Journal of Software, vol. 22, no. 4, **(2011)**, pp. 640–658.

[10] H.B. Mokadem, B. Berard, V. Gourcuff, O.D. Smet and J.M. Roussel, "Verification of a timed multitask system with Uppaal", IEEE Transactions on Automation Science and Engineering, vol. 7, no. 4, (**2010**), pp. 921–932.
[11] B. Berard, M. Bidoit, A. Finkel, F. Laroussinie, A. Petit, L. Petrucci and P. Schnoebelen, "Systems and software verification: model-checking techniques and tools", Springer Publishing Company, Incorporated, **(2010)**.
[12] P. Bouyer, "Model-checking timed temporal logics", In Proceedings of the 4th Workshop on Methods for Modalities (M4M-5), volume 231 of Electronic Notes in Theoretical Computer Science, Elsevier Science Publishers, **(2009)**.
[13] P. Bouyer and F. Laroussinie, "Model checking timed automata", In Stephan Merz and Nicolas Navet, editors, Modeling and Verification of Real-Time Systems, ISTE Ltd. – John Wiley & Sons, Ltd., **(2008)**.
[14] A. Brekling, M. R. Hansen and J. Madsen, "Models and formal verification of multiprocessor system-on-chips", The Journal of Logic and Algebraic Programming, vol. 77, no. 1–2, **(2008)**, pp. 1–19.
[15] Z. Haibin and D. Zhenhua, "Symbolic reachability analysis of multirate hybrid systems", Journal of Xi'an Jiaotong University, vol. 41, no. 4, **(2007)**, pp. 412–415.
[16] H.B Mokadem, B. Berard, P. Bouyer and F. Laroussinie, "Timed temporal logics for abstracting transient states", In Proceedings of the 4th International Symposium on Automated Technology for Verification and Analysis, Springer, **(2006)**.
[17] H.B. Mokadem, B. Berard, P. Bouyer and F. Laroussinie, "A new modality for almost everywhere propeties in timed automata", In Proceedings of the 16th International Conference on Concurrency Theory (CONCUR05), volume LNCS 3653. Springer, **(2005)**.
[18] P. Pettersson and W. Yi, "Formal modeling and analysis of timed systems", In Third International Conference, FORMATS 2005, Uppsala, Sweden, September 26-28, 2005, Proceedings, Lecture Notes in Computer Science. Springer, vol.3829, **(2005)**.
[19] P. Bouyer, E. Brinksma and K. G. Larsen, "Staying alive as cheaply as possible", In Proceedings of the 7th International Workshop on Hybrid Systems: Computation and Control (HSCC'04), volume 2993 of Lecture Notes in Computer Science, Springer, **(2004)**.
[20] T. Brihaye, V. Bruyere and J. F. Raskin, "Model-checking for weighted timed automata", In Proceedings of the Joint Conference on Formal Modelling and Analysis of Timed Systems and Formal Techniques in Real-Time and Fault Tolerant System, Springer, vol. 3253, **(2004)**.
[21] V. Bruyere, E. Dall'Olio and J. F. Raskin, "Durations, parametric model-checking in timed automata with presburger arithmetic", In Proceedings of the 20th Annual Symposium on Theoretical Aspects of Computer Science (STACS'03), volume 2607 of Lecture Notes in Computer Science, Springer, **(2003)**.
[22] S. Yovine, "Model checking timed automata", In School on Embedded Systems, vol. 1494 of Lecture Notes in Computer Science, Springer-Verlag, **(1998)**.
[23] S. Yamane, "The symbolic model-checking for real-time systems", In Proceedings of the Eighth Euromicro Workshop on Real-Time Systems, **(1996)**.
[24] R. Alur and D. Dill, "A theory of timed automata", Theoretical Computer Science (TCS), vol. 126, no. 2, **(1994)**, pp. 183–235.
[25] T. A. Henzinger, X. Nicollin, J. Sifakis and S. Yovine, "Symbolic model-checking for real-time systems", Information and Computation, vol. 111, no. 2, **(1994)**, pp. 193–244.

Authors

Mohammed Achkari Begdouri, he obtained his state engineer diploma in Computer Science from AbdelMalek Essaadi University, Morocco, in 2011. Actually, his is PhD student at National School of Applied Sciences of Tangier, Morocco (ENSA de Tanger). His research focuses on Theoretical Computer Science and Software Applications.

Houda Bel Mokadem, she received the Ph.D. degree in 2006 from École Normale Supérieure de Cachan, Cachan, France. She is an Associate Professor at ENSA de Tanger (Morocco). Her research area is the verification of temporal properties. She has published several research papers at international journals and conference proceedings.

Mohamed El Haddad, he received the Ph.D. degree in 1994 from Université Paris-Sud, France. He is a Full Professor at ENSA de Tanger (Morocco). His research focuses on Theoretical Computer Science and Software Applications. He has published several research papers at international journals and conference proceedings.

