Fundamenta Informaticae XX (2007) 1–28
IOS Press

Bounded Model Checking for the Existential Fragment of TCTL_{-G} and Diagonal Timed Automata[*]

Bożena Woźna[†][‡] and Andrzej Zbrzezny
Institute of Mathematics and Computer Science,
Jan Długosz University
Al. Armii Krajowej 13/15, 42-200 Częstochowa, Poland
{*b.wozna,a.zbrzezny*}*@ajd.czest.pl*

Abstract. Bounded Model Checking (BMC) is one of the well known SAT based symbolic model checking techniques. It consists in searching for a counterexample of a particular length, and generating a propositional formula that is satisfiable iff such a counterexample exists. The BMC method is feasible for the various classes of temporal logic; in particular it is feasible for TECTL (the existential fragment of Time Computation Tree Logic) and Diagonal-free Timed Automata. The main contribution of the paper is to show that the concept of Bounded Model Checking can be extended to deal with TECTL_{-G} properties of Diagonal Timed Automata. We have implemented our new BMC algorithm, and we present preliminary experimental results, which demonstrate the efficiency of the method.

1. Introduction

Model checking is a verification technique that was originally developed for (untimed) temporal logics [8]. Its main idea is to represent a finite state system, often deriving from a hardware or software design, as a labelled transition system (model), to represent a specification (property) by a modal formula, and to check automatically whether the formula holds in the model.

Over the past few years, the interest in automated verification has been moved towards concurrent real-time systems, and now verification of such systems is an active area of research. Various classes of models for real time systems have been proposed in the literature, but Timed Automata [2] and Timed

[*] The authors acknowledge support from the research grant No. 3T11C01128

[†] The research presented here was mainly conducted while B. Woźna was supported by EPSRC (grant GR/S49353).

[‡] Address for correspondence: Institute of Mathematics and Computer Science, Jan Długosz University, Al. Armii Krajowej 13/15, 42-200 Częstochowa, Poland

Petri Nets [20] are the best known ones. The properties to be verified are usually expressed either in standard temporal logics like LTL [8] and CTL [13], or in their timed versions like MITL [3] and TCTL [1]. In the paper we have decided to consider a restricted version of TCTL, since the model checking problem for branching time logics is decidable and PSPACE-complete in contrary to the model checking problem for MITL that is decidable but EXPSPACE-complete [28].

The practical applicability of the model checking method is strongly restricted by so-called the *state explosion problem*. Therefore, different reduction techniques were proposed to minimise models. The major methods include application of partial order reductions [9, 17, 22, 23, 32], symmetry reductions [14], abstraction techniques [10, 11], BDD-based symbolic storage methods [7, 18], and SAT-related algorithms [5, 19, 21, 30, 25, 26, 27, 35, 34].

Bounded model checking (BMC) is one of the SAT-based (satisfiability checking) methods, and it was introduced as a technique complementary to the BDD-based symbolic model checking for LTL [5]. The main idea of BMC is to search for an execution (or a set of executions) of the system of some length k that constitutes a counterexample for a tested property. If no counterexample of length k can be found, then k is increased by one. The efficiency of this method is based upon the observation that if a system is faulty, then often examining only a (small) fragment of its state space is sufficient for finding an error. Obviously, when testing large models and complex formulae the efficiency of the BMC method is dependent on the speed of the chosen SAT solver, with which the test is carried out. As SAT checkers have progressively become more effective, the efficiency of BMC has improved, an observation experimentally demonstrated in, among others, [5, 26, 27].

In order to represent and verify real time systems it is convenient to have expressive and easy-to-use models. Timed automata with diagonal constraints (or Diagonal Timed Automata [31]) are sufficiently expressive to describe the essential aspects of time-dependent real-life problems in a variety of application domains; in particular, they are very useful for modelling scheduling problems [15]. It is known that diagonal constraints can be eliminated from diagonal timed automata, and that they do not add expressive power to diagonal-free timed automata [4]. However, a construction of diagonal-free timed automaton for a given diagonal timed automaton leads to an exponential (in the number of diagonal constraints) blowup of the number of states of the automaton, and this blowup is unavoidable. Therefore, in the present paper we deal with diagonal timed automata.

The main contribution of the paper consists in showing that the concept of Bounded Model Checking can be extended to deal with TECTL_{-G} (the existential fragment of TCTL without the modality standing for "*always in the future within an interval*") properties of Diagonal Timed Automata. In particular, we show that the discretisation method [37] can be applied to model check an arbitrary TECTL_{-G} formula over Diagonal Timed Automata. Moreover, we have implemented the new BMC algorithm, and we provide some preliminary experimental results that seem to be promising.

The rest of the paper is organised as follows. In Section 2 we briefly introduce diagonal timed automata and a model for them. In Section 3 we give a syntax and semantics for TCTL_{-G} and its subsets. Section 4 contains the basic definitions and notations that are used in Section 5 to define a translation of the model checking problem for TCTL_{-G} to the model checking problem of CTL^{y}_{-G}. In Section 6 we present a discretisation technique that is a base for our implementation of the BMC method for TECTL_{-G}, defined in Section 7. In Section 8 we provide some preliminary experimental results for a modified Fischer mutual exclusion protocol, and in Section 9 we show how to tailor the BMC technique from Section 7 to a specially chosen subclass of TECTL_{-G} formulae. The last section contains a discussion of related work and final remarks.

2. Diagonal Timed Automata

To define diagonal time automata formally we need to say what type of clock constraints are allowed as guards and invariants. For a finite set of real variables X, called *clocks*, and a set of natural numbers $\mathbb{N} = \{0, 1, \ldots\}$, we define the set $C(X)$ of *clock constraints* over X by the following grammar:

$$\mathfrak{cc} ::= \; true \mid x \sim c \mid x - y \sim c \mid \mathfrak{cc} \wedge \mathfrak{cc}$$

where $x,\ y\ \in\ X, c\ \in\ \mathbb{N}$ and $\sim \in\ \{<, \leq, =, >, \geq\}$.

A *clock valuation* v is a total function from X into the set of nonnegative real numbers $\mathbb{R}$. $\mathbb{R}^X$ denotes the set of all clock valuations. For a clock constraint $\mathfrak{cc} \in C(X)$, $[\![\mathfrak{cc}]\!]$ denotes the set of all clock valuations that satisfy $\mathfrak{cc}$. The clock valuation assigning the value 0 to all clocks in X is denoted by v^0. For $v \in \mathbb{R}^X$ and $\delta \in \mathbb{R}$, the clock valuation $v + \delta$ assigns the value $v(x) + \delta$ to each clock $x \in X$. For $v \in \mathbb{R}^X$ and $Y \subseteq X$, $v[Y]$ denotes the clock valuation that assigns the value 0 to each clock in Y and leaves the values of the other clocks unchanged.

Definition 2.1. (Diagonal Timed Automaton)
Let $\mathcal{PV}$ be a set of propositional variables. A *diagonal timed automaton* (for short a *timed automaton*) is a tuple $\mathcal{A} = (\Sigma, L, l^0, X, \mathcal{I}, R, \mathcal{V})$, where Σ is a nonempty finite set of actions, L is a nonempty finite set of locations, $l^0 \in L$ is an initial location, $\mathcal{V} : L \mapsto 2^{\mathcal{PV}}$ is a valuation function assigning to each location a set of atomic propositions true in that location, X is a finite set of clocks, $\mathcal{I} : L \mapsto C(X)$ is a state invariant function, and $R \subseteq L \times \Sigma \times C(X) \times 2^X \times L$ is a transition relation.

An element $(l,\ \sigma,\ \mathfrak{cc},\ Y,\ l') \in R$ represents a transition from the location l to the location l' labelled with the action σ. The invariant condition allows the automaton to stay at the location l as long only as the constraint $\mathcal{I}(l)$ is satisfied. The guard $\mathfrak{cc}$ has to be satisfied to enable the transition. The transition resets all clocks in the set Y to the value 0.

The semantics of a timed automaton $\mathcal{A}$ is defined by associating to it a *dense model* as defined below.

Definition 2.2. (Dense Model)
Let $\mathcal{A} = (\Sigma, L, l^0, X, \mathcal{I}, R, \mathcal{V})$ be a timed automaton. A *dense model* for $\mathcal{A}$ is a tuple $\mathcal{G}(\mathcal{A}) = (\Sigma \cup \mathbb{R}, Q, q^0, \widetilde{\mathcal{V}}, \rightarrow)$, where $\Sigma \cup \mathbb{R}$ is a set of labels, $Q = L \times \mathbb{R}^X$ is a set of states, $q^0 = (l^0, v^0)$ is an initial state, $\widetilde{\mathcal{V}} : Q \mapsto 2^{\mathcal{PV}}$ is a valuation function such that $\widetilde{\mathcal{V}}((l, v)) = \mathcal{V}(l)$, and $\rightarrow \subseteq Q \times (\Sigma \cup \mathbb{R}) \times Q$ is a transition relation of $\mathcal{G}(\mathcal{A})$ defined by:

- Time transitions: $(l, v) \xrightarrow{\delta} (l, v + \delta)$ iff $(\forall 0 \leq \delta' \leq \delta)\ v + \delta' \in [\![\mathcal{I}(l)]\!]$
- Action transitions: $(l, v) \xrightarrow{\sigma} (l', v')$ iff $(\exists \mathfrak{cc} \in C(X))(\exists Y \subseteq X)$ such that $v' = v[Y]$, $(l,\ \sigma,\ \mathfrak{cc},\ Y,\ l') \in R,\ \ v \in [\![\mathfrak{cc}]\!]$, and $v' \in [\![\mathcal{I}(l')]\!]$.

Intuitively, a time transition does not change a location, but describes an equal increase in the value of all the clocks, provided that the new clock valuations still satisfy all the location invariants. An action transition corresponds to an action performed by the automaton under consideration. Following this, its location changes accordingly, and all the clocks that are associated with the action are set to zero (i.e., the ones which belong to the set $Y \subseteq X$). Obviously, the action can be performed only if the underling enabling condition is satisfied.

The below lemma shows that for the considered set of clock constraints $C(X)$, in the semantics of timed automata the condition of a time transition $(l, v) \xrightarrow{\delta} (l, v + \delta)$ can be replaced by the following:

$$v \in [\![\mathcal{I}(l)]\!] \text{ and } v + \delta \in [\![\mathcal{I}(l)]\!].$$

Lemma 2.1. Let $\mathfrak{cc} \in C(X)$, $v \in \mathbb{R}^X$, and $\delta \in \mathbb{R}$. If $v \in [\![\mathfrak{cc}]\!]$ and $v + \delta \in [\![\mathfrak{cc}]\!]$, then for each $(0 \leq \delta' \leq \delta)$ $v + \delta' \in [\![\mathfrak{cc}]\!]$.

Proof: Straightforward by induction on clock constraints. □

For $(l, v) \in Q$, let $(l, v) + \delta$ denote $(l, v + \delta)$. A q_0-*run* ρ of $\mathcal{A}$ is a sequence of states:

$$q_0 \xrightarrow{\delta_0} q_0 + \delta_0 \xrightarrow{\sigma_0} q_1 \xrightarrow{\delta_1} q_1 + \delta_1 \xrightarrow{\sigma_1} q_2 \xrightarrow{\delta_2} \ldots,$$

where $q_i \in Q$, $\sigma_i \in \Sigma$ and $\delta_i \in \mathbb{R}_+$ for each $i \in \mathbb{N}$. In other words, a run is an infinite sequence of states such that action transitions are taken infinitely often and time transitions are aggregated. Notice that the semantics does not permit to perform two consecutive action transitions, that is, between each two action transitions some time must pass. This is a convenient way of representing a series of events to be taken in a continuous time.

Given a run $\rho = q_0 \xrightarrow{\delta_0} q_0 + \delta_0 \xrightarrow{\sigma_0} q_1 \xrightarrow{\delta_1} q_1 + \delta_1 \xrightarrow{\sigma_1} q_2 \xrightarrow{\delta_2} \ldots$, we say that a run $\rho' = q'_0 \xrightarrow{\delta'_0} q'_0 + \delta'_0 \xrightarrow{\sigma'_0} q'_1 \xrightarrow{\delta'_1} q'_1 + \delta'_1 \xrightarrow{\sigma'_1} q'_2 \xrightarrow{\delta'_2} \ldots$ is a *suffix* of ρ if there exists $i \in \mathbb{N}$ such that $q'_0 = q_i$ and $\sigma'_j = \sigma_{i+j}$ and $\delta'_j = \delta_{i+j}$ for each $j \in \mathbb{N}$.

In line with much of the literature of the real life systems we make the assumption that a system under consideration runs continuously without termination. This requirement is normally expressed by distinguishing between *discrete progress* and *time progress*. Under discrete progress we allow for action transitions to happen infinitely often, that is, no state occurs without action successors. Under time progress one assumes that time may pass without an upper bound; this is usually formalised by the notion of *non-zeno* runs.

Formally, an infinite run ρ is said to be *non-zeno* iff $\Sigma_{i \in \mathbb{N}} \delta_i$ is unbounded. An infinite run ρ is said to be *zeno* iff $\Sigma_{i \in \mathbb{N}} \delta_i$ is bounded by some real value. As an example, consider the automaton shown in Figure 1. Its q_0-run $(q_0, 0) \xrightarrow{1} (q_0, 1) \xrightarrow{a} (q_0, 1) \xrightarrow{0.5} (q_0, 1.5) \xrightarrow{a} (q_0, 1.5) \xrightarrow{0.25} (q_0, 1.75) \xrightarrow{a} (q_0, 1.75) \xrightarrow{0.125} (q_0, 1.875) \xrightarrow{a} (q_0, 1.875) \xrightarrow{0.0625} \ldots$ is zeno. On the other hand the following q_0-run $(q_0, 0) \xrightarrow{1} (q_0, 1) \xrightarrow{b} (q_1, 1) \xrightarrow{1} (q_1, 2) \xrightarrow{c} (q_1, 0) \xrightarrow{1} (q_1, 1) \xrightarrow{c} (q_1, 0) \xrightarrow{1} (q_1, 1) \xrightarrow{c} \ldots$ is non-zeno.

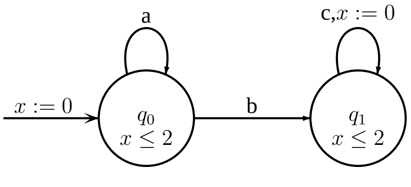

Figure 1. An example of non-time progressive timed automaton.

We say that $\mathcal{A}$ is *time-progressive* iff all its q^0-runs are non-zeno. For easiness of presentation, we consider only time-progressive timed automata; note that time-progressiveness can be checked as in[31].

3. Time Computation Tree Logic without G

In this section, we formally present a syntax and semantics of TCTL_{-G}, a fragment of Time Computation Tree Logic (TCTL) [1] that does not include the modality for "always in the future within some interval".

Syntax. Let $\mathcal{PV}$ be a set of propositional variables containing the symbol $\top$ (denoting the constant "true"), and I an interval in $\mathbb{R}$ with integer bounds of the form $[n, n']$, $[n, n')$, $(n, n']$, (n, n'), (n, ∞), and $[n, \infty)$, for $n, n' \in \mathbb{N}$. For $p \in \mathcal{PV}$, the set of TCTL_{-G} formulae is defined by the following grammar:

$$\varphi := p \mid \neg p \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \mathrm{E}(\varphi \mathrm{U}_I \varphi) \mid \mathrm{A}(\varphi \mathrm{U}_I \varphi)$$

The other basic operators are defined as usual: $\bot \overset{def}{=} \neg\top$, $\alpha \Rightarrow \beta \overset{def}{=} \neg\alpha \vee \beta$, $\alpha \Leftrightarrow \beta \overset{def}{=} (\alpha \Rightarrow \beta) \wedge (\beta \Rightarrow \alpha)$, $\mathrm{EF}_I\alpha \overset{def}{=} \mathrm{E}(\top \mathrm{U}_I \alpha)$, $\mathrm{AF}_I\alpha \overset{def}{=} \mathrm{A}(\top \mathrm{U}_I \alpha)$.

TECTL_{-G} is a fragment of TCTL_{-G} such that the temporal formulae are restricted to the Boolean combination of $\mathrm{E}(\alpha \mathrm{U}_I \beta)$ only. Similarly, TACTL_{-G} is a fragment of TCTL_{-G} such that the temporal formulae are restricted to the Boolean combination of $\mathrm{A}(\alpha \mathrm{U}_I \beta)$ only.

Semantics. Let $\mathcal{A} = (\Sigma, L, l^0, X, \mathcal{I}, R, \mathcal{V})$ be a timed automaton, $\mathcal{G}(\mathcal{A}) = (\Sigma \cup \mathbb{R}, Q, q^0, \widetilde{\mathcal{V}}, \rightarrow)$ a dense model for $\mathcal{A}$, $\rho = q_0 \xrightarrow{\delta_0} q_0 + \delta_0 \xrightarrow{\sigma_0} q_1 \xrightarrow{\delta_1} q_1 + \delta_1 \xrightarrow{\sigma_1} q_2 \xrightarrow{\delta_2} \ldots$ a run of $\mathcal{A}$ such that $q_i \in Q$ and $\delta_i \in \mathbb{R}_+$ for all $i \in \mathbb{N}$, and $f_{\mathcal{A}}(q_0)$ denote the set of all such q_0-runs that are suffixes of the q^0-runs. In order to define a satisfaction relation for TCTL_{-G}, we define the notion of a *dense path π_ρ corresponding to run ρ*. This can be done in an unique way because of the assumption that $\delta_i \in \mathbb{R}_+$. First, let $idx(\rho, r)$ be the greatest $i \in \mathbb{N}$ such that $\Sigma_{j=0}^{i-1}\delta_j \leq r$. Notice that for $i = 0$ we let $\Sigma_{j=0}^{i-1}\delta_j = 0$. So, for $r \leq \delta_0$, $idx(\rho, r) = 0$. Now, *a dense path π_ρ corresponding to ρ* is a mapping from $\mathbb{R}$ to the set of states Q such that $\pi_\rho(r) = q_i + r - \Sigma_{j=0}^{i-1}\delta_j$ where $i = idx(\rho, r)$.

Definition 3.1. (Satisfaction)
Let $\mathcal{G}(\mathcal{A}) = (\Sigma \cup \mathbb{R}, Q, q^0, \widetilde{\mathcal{V}}, \rightarrow)$ be a *dense model*, q a state, and α, β TCTL_{-G} formulae. The satisfaction relation $\models$, which indicates truth of a formula in the dense model $\mathcal{G}(\mathcal{A})$ at state q, is defined inductively as follows:

$$\mathcal{G}(\mathcal{A}), q \models p \ \text{ iff } \ p \in \widetilde{\mathcal{V}}(q), \quad \mathcal{G}(\mathcal{A}), q \models \alpha \vee \beta \ \text{ iff } \ \mathcal{G}(\mathcal{A}), q \models \alpha \text{ or } \mathcal{G}(\mathcal{A}), q \models \beta,$$

$$\mathcal{G}(\mathcal{A}), q \models \neg p \ \text{ iff } \ p \notin \widetilde{\mathcal{V}}(q), \quad \mathcal{G}(\mathcal{A}), q \models \alpha \wedge \beta \ \text{ iff } \ \mathcal{G}(\mathcal{A}), q \models \alpha \text{ and } \mathcal{G}(\mathcal{A}), q \models \beta,$$

$$\mathcal{G}(\mathcal{A}), q \models \mathrm{E}(\alpha \mathrm{U}_I \beta) \text{ iff } (\exists\, \rho \in f_{\mathcal{A}}(q))(\exists r \in I)[\mathcal{G}(\mathcal{A}), \pi_\rho(r) \models \beta \text{ and } (\forall r' < r)\ \mathcal{G}(\mathcal{A}), \pi_\rho(r') \models \alpha],$$

$$\mathcal{G}(\mathcal{A}), q \models \mathrm{A}(\alpha \mathrm{U}_I \beta) \text{ iff } (\forall\, \rho \in f_{\mathcal{A}}(q))(\exists r \in I)[\mathcal{G}(\mathcal{A}), \pi_\rho(r) \models \beta \text{ and } (\forall r' < r)\ \mathcal{G}(\mathcal{A}), \pi_\rho(r') \models \alpha].$$

We end the section by defining the notion of validity in a dense model and defining formally the model checking problem.

A TCTL_{-G} formula φ is *valid in a dense model* $\mathcal{G}(\mathcal{A})$ (denoted by $\mathcal{G}(\mathcal{A}) \models \varphi$) iff $\mathcal{G}(\mathcal{A}), q^0 \models \varphi$, i.e. φ is true at the initial state of the dense model $\mathcal{G}(\mathcal{A})$; checking validity for given $\mathcal{G}(\mathcal{A})$ and φ is called the *model checking problem*.

4. Equivalence of Clock Assignments and Region Graph

Consider a system that is described by a time automaton $\mathcal{A}$. By Definition 2.2, the dense model $\mathcal{G}(\mathcal{A})$ has infinitely many states, however, not all of these states are distinguishable by TCTL$_{-G}$, which is interpreted over $\mathcal{G}(\mathcal{A})$. Namely, if two states agree on the integral parts of all the clock values, and also on the ordering of the fractional parts of all the clocks, then the computation trees rooted at these two states cannot be distinguished by TCTL$_{-G}$formulae. The integral parts of the clocks in a state are needed to determine whether or not a particular enabling condition is met, whereas the ordering of the fractional parts is needed to decide which clock will change its integral part first. For example, if two clocks x and y are between 0 and 1 in a state, then whether or not a transition with enabling condition $x \geq 1$ can be followed by a transition with enabling condition $y \geq 1$, depends on whether or not the state satisfies $x < y$. Now we formalise the above.

Definition 4.1. (Weak Region Equivalence)
Let $\mathcal{A} = (\Sigma, L, l^0, X, \mathcal{I}, R, \mathcal{V})$ be a timed automaton, and for any $t \in \mathbb{R}$, let $\langle t \rangle$ denote the fractional part of t and $\lfloor t \rfloor$ its integral part. A relation $\cong \subseteq \mathbb{R}^X \times \mathbb{R}^X$, called *weak region equivalence* [37], is defined as follows. For two clock valuations u and v in $\mathbb{R}^X$, we say that $u \cong v$ iff all the following conditions hold:

E1. $\lfloor u(x) \rfloor = \lfloor v(x) \rfloor$, for all $x \in X$,

E2. $\langle u(x) \rangle = 0$ iff $\langle v(x) \rangle = 0$, for all $x \in X$,

E3. $\langle u(x) \rangle < \langle u(y) \rangle$ iff $\langle v(x) \rangle < \langle v(y) \rangle$, for all $x,\ y \in X$.

Note that the condition E3 implies the following condition:

E4. $\langle u(x) \rangle = \langle u(y) \rangle$ iff $\langle v(x) \rangle = \langle v(y) \rangle$, for all $x,\ y \in X$

The relation $\cong$ partitions $\mathbb{R}^X$ into *zones* denoted by Z, Z', and so on; the set of all the zones is denoted by $Z(|X|)$. Observe that the relation $\cong$ has an infinite index, so the set $Z(|X|)$ is infinite.

To perform a TCTL$_{-G}$ model checking algorithm efficiently, we need to consider a more coarser model than the dense one. A *region graph (RG)*, as defined below, can be viewed as such a model.

Definition 4.2. (Region Graph)
Let $\mathcal{A} = (\Sigma, L, l^0, X, \mathcal{I}, R, \mathcal{V})$ be a timed automaton. *A Region graph* for $\mathcal{A}$ is a tuple $M_{rg} = (\Sigma \cup \{\tau\}, Q_{rg}, q^0_{rg}, \widetilde{\mathcal{V}}_{rg}, \to_{rg})$, where $Q_{rg} = L \times Z(|X|)$ is a set of states called *regions*, $q^0_{rg} = (l^0, \{v^0\})$ is an initial region, $\widetilde{\mathcal{V}}_{rg} : Q_{rg} \mapsto 2^{\mathcal{PV}}$ is a valuation function defined by: $\widetilde{\mathcal{V}}_{rg}((l, Z)) = \mathcal{V}(l)$, and $\to_{rg} \subseteq Q_{rg} \times (\Sigma \cup \{\tau\}) \times Q_{rg}$ is defined by:

- Time transition: $(l, Z) \xrightarrow{\tau}_{rg} (l, Z')$ iff there exist $v \in Z$ and $v' \in Z'$ such that
  a.) $(l, v) \xrightarrow{\delta} (l, v')$ for some $\delta \in \mathbb{R}_+$, and
  b.) if $(l, v) \xrightarrow{\delta'} (l, v'') \xrightarrow{\delta''} (l, v')$ with δ', $\delta'' \in \mathbb{R}_+$, and $(l, Z'') \in Q_{rg}$ for some Z'' such that $v'' \in Z''$, then $v \cong v''$ or $v'' \cong v'$.

- Action transition: For any $\sigma \in \Sigma$, $(l, Z) \xrightarrow{\sigma}_{rg} (l', Z')$ iff there exists Z'' such that $(l, Z) \xrightarrow{\tau}_{rg} (l, Z'')$ and there exist $v \in Z''$ and $v' \in Z'$ such that $(l, v) \xrightarrow{\sigma} (l', v')$.

Observe that since the set $Z(|X|)$ is infinite, so the set Q_{rg}. Note also that the case (b) of the definition of the time transition is necessary to preserve validity of the TCTL_{-G} formulae of the form $\mathrm{E}(\alpha \mathrm{U}_I \beta)$ and $\mathrm{A}(\alpha \mathrm{U}_I \beta)$ over the region graph. This will be shown below.

Let us start by extending the equivalence relation $\cong$ to an equivalence relation over the set of states of the dense model, and proving some auxiliary lemmas 4.1- 4.4.

Definition 4.3. For any (l, u) and (l', u') in Q, we say that $(l, u) \cong (l', u')$ iff $l = l'$ and $u \cong u'$.

Lemma 4.1. Let X be a set of clocks, and $u, v \in \mathbb{R}^X$ be clock valuations such that $u \cong v$. For any clock constraint $\mathfrak{cc} \in C(X)$, $u \in [\![\mathfrak{cc}]\!]$ iff $v \in [\![\mathfrak{cc}]\!]$.

Proof: Straightforward induction on clock constraints. □

Lemma 4.2. Let u, v be clock valuations such that $u \cong v$. For every $\delta \in \mathbb{R}$ there exists $\delta' \in \mathbb{R}$ such that $u + \delta \cong v + \delta'$.

Proof: We omit the proof as it is analogous to the proof of the Lemma 4.3 of [36] □

Lemma 4.3. Let $\sigma \in \Sigma$, and q_1, q_2 be states such that $q_1 \cong q_2$. For each state q_3 such that $q_1 \xrightarrow{\sigma} q_3$ there exists a state q_4 such that $q_2 \xrightarrow{\sigma} q_4$ and $q_3 \cong q_4$.

Proof: Straightforward, by using Lemma 4.1. □

Lemma 4.4. Let q_0, q_0' be two states such that $q_0 \cong q_0'$, and ρ be a run $q_0 \xrightarrow{\delta_0} q_0 + \delta_0 \xrightarrow{\sigma_0} q_1 \xrightarrow{\delta_1} q_1 + \delta_1 \xrightarrow{\sigma_1} q_2 \xrightarrow{\delta_2} \ldots$ of $\mathcal{G}(\mathcal{A})$. There exists a run ρ' of $\mathcal{G}(\mathcal{A})$ such that $\pi_{\rho'}(0) = q_0'$, and for every $r \in \mathbb{R}$, $\pi_\rho(r) \cong \pi_{\rho'}(r))$.

Proof: Straightforward, by using Lemmas 4.2 and 4.3. □

We can now show that all the states that belong to the same region satisfy the same set of TCTL_{-G} formulae. In fact, we could show that Lemma 4.5 holds for the whole TCTL language, however since in the paper we are interested in the TCTL_{-G} fragment of TCTL, we will prove the lemma for this fragment only.

Lemma 4.5. Let $\mathcal{A}$ be a timed automaton, φ a TCTL_{-G} formula, $\mathcal{G}(\mathcal{A})$ a dense model for $\mathcal{A}$, l a location, and u, v two clock valuations such that $u \cong v$. Then, $\mathcal{G}(\mathcal{A}), (l, u) \models \varphi$ iff $\mathcal{G}(\mathcal{A}), (l, v) \models \varphi$.

Proof: [Induction on the length of TCTL_{-G} formulae].

It is easy to see that the thesis holds for all the propositional variables and for all the negations of propositional variables. If φ is of the form $\alpha \vee \beta$ or $\alpha \wedge \beta$, then the proof of the thesis is straightforward. It remains to prove that the thesis holds for formulae of the form $\mathrm{E}(\alpha \mathrm{U}_I \beta)$ and $\mathrm{A}(\alpha \mathrm{U}_I \beta)$.

Consider a formula φ to be of the form $\mathrm{E}(\alpha \mathrm{U}_I \beta)$ and suppose that $\mathcal{G}(\mathcal{A}), q \models \mathrm{E}(\alpha \mathrm{U}_I \beta)$. By Definition 3.1 we have that for some run $\rho \in f_{\mathcal{A}}(q)$ there exists $r \in I$ such that

$$\mathcal{G}(\mathcal{A}), \pi_\rho(r) \models \beta \text{ and } (\forall r' < r)\ \mathcal{G}(\mathcal{A}), \pi_\rho(r') \models \alpha$$

By Lemma 4.4 there exists a run $\rho' \in f_{\mathcal{A}}(q')$ such that for every $r \in \mathbb{R}$, $\pi_\rho(r) \cong \pi_{\rho'}(r)$. Thus, by the induction hypotheses $\mathcal{G}(\mathcal{A}), \pi_{\rho'}(r) \models \beta$ and $(\forall r' < r)\ \mathcal{G}(\mathcal{A}), \pi_{\rho'}(r') \models \alpha$. Hence it follows that $\mathcal{G}(\mathcal{A}), q' \models \mathrm{E}(\alpha \mathrm{U}_I \beta)$.

The proof of the case $\mathrm{A}(\alpha \mathrm{U}_I \beta)$ is analogous. □

5. Translation from $\mathbf{TCTL}_{-G}$ to $\mathbf{CTL}^y_{-G}$

Here, we show how to translate the model checking problem for TCTL_{-G} to the model checking problem for CTL^y_{-G}, a logic defined later on. The idea of the translation is taken from [1], and consists in encoding all the time intervals appearing in the TCTL_{-G} formula under consideration by propositional variables.

Formally, given a timed automaton $\mathcal{A} = (\Sigma, L, l^0, X, \mathcal{I}, R, \mathcal{V})$, and a TCTL_{-G} formula φ, we first construct a new timed automaton $\mathcal{A}_\varphi = (\Sigma', L', l'^0, X', \mathcal{I}', R', \mathcal{V}')$ in the following way:

- $\Sigma' = \Sigma \cup \{\sigma_y\}$, $L' = L$, $l'^0 = l^0$, $\mathcal{I}' = \mathcal{I}$, and $\mathcal{V}' = \mathcal{V}$,
- $X' = X \cup \{y\}$. A new clock y corresponds to all the time intervals appearing in φ; one clock is sufficient to perform the bounded model checking algorithm that is proposed in the next section, however, other model checking methods can require one clock per one time interval appearing in the TCTL_{-G} formula under consideration.
- $R' = R \cup \{(l, \sigma_y, true, \{y\}, l) \mid l \in L\}$. The new transitions are used to reset the new clock y, and so to start the runs over which sub-formulae of φ are checked.

Next, we extend the set of propositional variables $\mathcal{PV}$ into the set $\mathcal{PV}' = \mathcal{PV} \cup \mathcal{PV}_\varphi \cup \{p_b\}$, where $\mathcal{PV}_\varphi = \{p_{y \in I} \mid I \text{ is a time interval in } \varphi\}$ and p_b is a proposition variable representing so called *boundary regions*; a region (l, Z) is called *boundary* if for each $\delta \in \mathbb{R}_+$ and each $v \in Z$, $(v, v + \delta) \notin \cong$. Then, we construct the region graph $M_{rg} = (\Sigma \cup \{\tau\}, Q_{rg}, q^0_{rg}, \widetilde{\mathcal{V}}_{rg}, \rightarrow_{rg})$ for $\mathcal{A}_\varphi$ such that $\widetilde{\mathcal{V}}_{rg} : Q_{rg} \mapsto 2^{\mathcal{PV}'}$, $M_{rg}, (l, Z) \models p_{y \in I}$ iff there exists $v \in Z$ such that $v(y) \in I$, and $M_{rg}, (l, Z) \models p_b$ iff there exists $x \in \mathrm{X}$ such that for all $v \in Z$, $\langle v(x) \rangle = 0$. Finally, we translate the TCTL_{-G} formula φ into a CTL^y_{-G} formula $\psi = \mathrm{cr}(\varphi)$ in such a way that the model checking of φ over the dense model for $\mathcal{A}$ can be reduced to the model checking of ψ over the region graph for $\mathcal{A}_\varphi$. In order to describe this translation formally, first we have to define the CTL^y_{-G} language.

For $p \in \mathcal{PV}'$, the set of CTL^y_{-G} formulae $\mathcal{WF}$ is defined by the following grammar:

$$\alpha := p \mid \neg p \mid \alpha \wedge \alpha \mid \alpha \vee \alpha \mid \mathrm{E}_y(\alpha \mathrm{U} \alpha) \mid \mathrm{A}_y(\alpha \mathrm{U} \alpha)$$

The CTL^y_{-G} language is interpreted over the region graph for $\mathcal{A}_\varphi$ as defined below.

Definition 5.1. (Satisfaction for $\mathbf{CTL}^y_{-G}$)

Let $M_{rg} = (\Sigma \cup \{\tau\}, Q_{rg}, q^0_{rg}, \widetilde{\mathcal{V}}_{rg}, \rightarrow_{rg})$ be the region graph for $\mathcal{A}_\varphi$, $q \in Q_{rg}$, α, β formulae of CTL^y_{-G}, $\rightarrow_{\mathcal{A}}$ denote the part of $\rightarrow_{rg}$, where transitions are labelled with elements of $\Sigma \cup \{\tau\}$, and $\rightarrow_y$ denote the transitions that reset the clock y. A *path* π in M_{rg} is a sequence $(q_0, q_1, \ldots)$ of states such that

$q_i \rightarrow_{\mathcal{A}} q_{i+1}$ for each $i \in \mathbb{N}$, and $\Pi(s)$ denotes the set of all the paths starting at q in M_{rg}. The satisfaction relation $\models$ is defined inductively as follows:

$$\begin{array}{ll}
M_{rg}, q \models p & \text{iff}\, p \in \widetilde{\mathcal{V}}_{rg}(q), \\
M_{rg}, q \models \neg p & \text{iff}\, p \notin \widetilde{\mathcal{V}}_{rg}(q), \\
M_{rg}, q \models \alpha \vee \beta & \text{iff}\, M_{rg}, q \models \alpha \ \text{ or } \ M_{rg}, q \models \beta, \\
M_{rg}, q \models \alpha \wedge \beta & \text{iff}\, M_{rg}, q \models \alpha \ \text{ and } \ M_{rg}, q \models \beta, \\
M_{rg}, q \models \mathrm{E}_y(\alpha \mathrm{U} \beta) & \text{iff}\, (\exists q' \in Q_{rg})(q \rightarrow_y q' \text{ and } (\exists \pi \in \Pi(q'))(\exists m \geq 0)\ [M_{rg}, \pi(m) \models \beta \\
& \quad \text{and } (\forall j < m)\ M_{rg}, \pi(j) \models \alpha]), \\
M_{rg}, q \models \mathrm{A}_y(\alpha \mathrm{U} \beta) & \text{iff}\, (\exists q' \in Q_{rg})(q \rightarrow_y q' \text{ and } (\forall \pi \in \Pi(q'))(\exists m \geq 0)\ [M_{rg}, \pi(m) \models \beta \\
& \quad \text{and } (\forall j < m)\ M_{rg}, \pi(j) \models \alpha]).
\end{array}$$

A CTL^{y}_{-G} formula φ is *valid in* M_{rg} (denoted $M_{rg} \models \varphi$) iff $M_{rg}, q^0_{rg} \models \varphi$, i.e., φ is true at the initial state of M_{rg}.

Having defined the CTL^{y}_{-G} logic, we can now introduce the translation mentioned above.

Definition 5.2. (Translation)
Let φ be a TCTL_{-G} formula. φ is translated inductively into the CTL^{y}_{-G} formula $\mathrm{cr}(\varphi)$ as follows:

- $\mathrm{cr}(p) = p$ for $p \in \mathcal{PV}'$
- $\mathrm{cr}(\neg p) = \neg\mathrm{cr}(p)$ for $p \in \mathcal{PV}'$
- $\mathrm{cr}(\alpha \vee \beta) = \mathrm{cr}(\alpha) \vee \mathrm{cr}(\beta)$
- $\mathrm{cr}(\alpha \wedge \beta) = \mathrm{cr}(\alpha) \wedge \mathrm{cr}(\beta)$
- $\mathrm{cr}(\mathrm{E}(\alpha \mathrm{U}_I \beta)) = \mathrm{E}_y(\mathrm{cr}(\alpha)\mathrm{U}(\mathrm{cr}(\beta) \wedge p_{y \in I} \wedge (p_b \vee \mathrm{cr}(\alpha))))$
- $\mathrm{cr}(\mathrm{A}(\alpha \mathrm{U}_I \beta)) = \mathrm{A}_y(\mathrm{cr}(\alpha)\mathrm{U}(\mathrm{cr}(\beta) \wedge p_{y \in I} \wedge (p_b \vee \mathrm{cr}(\alpha))))$

It is clear how to translate the TCTL_{-G} formulae that are either disjunction or conjunction of the TCTL_{-G} formulae. The same concerns the proposition variables and their negations. To translate the TCTL_{-G} formulae of the form $\mathrm{E}(\alpha \mathrm{U}_I \beta)$ we have to ensure that:

a.) there exists a path $\pi = (s_0, \ldots, s_i, \ldots)$ in the region graph for $\mathcal{A}_\varphi$ that starts at a state with the value of the clock y equal to zero; this is expressed by the quantifier E_y.

b.) there is a state $s_i = (l, v)$ on the path π such that $v(y) \in I$ and the translation of β holds in this state; this is expressed by the requirement $\mathrm{cr}(\beta) \wedge p_{y \in I}$.

c.) the translation of α holds in all the states s_j on the path π, for $j < i$; this is expressed by employing the standard until operator, i.e., $\mathrm{cr}(\alpha)\mathrm{U}(\mathrm{cr}(\beta) \wedge p_{y \in I})$.

To understand the conjunct $p_b \vee \mathrm{cr}(\alpha)$ notice that we have to take into consideration the shape of a region in which $\mathrm{cr}(\beta)$ holds. Namely, if this region is not boundary, then its borders are open, and therefore each state belonging to the region has some time predecessors that also belong to the same region. Thus, if we require that $\mathrm{E}(\alpha \mathrm{U}_I \beta)$ holds, then $\mathrm{cr}(\alpha)$ holds continuously until $\mathrm{cr}(\beta)$, and $\mathrm{cr}(\alpha)$ must hold at all the states of the region where $\mathrm{cr}(\beta)$ holds; this is expressed by the condition $p_b \vee \mathrm{cr}(\alpha)$ put in conjunction with $\mathrm{cr}(\beta) \wedge p_{y \in I}$.

To translate the TCTL_{-G} formulae of the form $\mathrm{A}(\alpha \mathrm{U}_I \beta)$ we proceed in a similar way as in the case of $\mathrm{E}(\alpha \mathrm{U}_I \beta)$. The only difference is in point (a), i.e., we have to ensure now that for all the paths $\pi = (s_0, \ldots, s_i, \ldots)$ in the region graph for $\mathcal{A}_\varphi$ that starts at a state with the value of the clock y equal to zero.

The following lemma shows that validity of the TCTL_{-G} formula φ over the model for $\mathcal{A}$ is equivalent to the validity of the corresponding CTL^{y}_{-G} formula $\mathrm{cr}(\varphi)$ over the region graph for $\mathcal{A}_\varphi$ with the extended valuation function.

Lemma 5.1. Let $\mathcal{A}$ be a timed automaton, $\mathcal{G}(\mathcal{A})$ a dense model for $\mathcal{A}$, φ TCTL_{-G} formula, and M_{rg} the region graph for $\mathcal{A}_\varphi$. Further, let $v \lfloor X$ denote the restriction of the valuation v to the set X, $(l, v) \lfloor X \overset{def}{=} (l, v \lfloor X)$, and for any state $(l, v) \lfloor X \in Q$, $[(l, v)]$ denote the equivalence class of (l, v) induced by the relation $\cong$. Then, $\mathcal{G}(\mathcal{A}), (l, v) \lfloor X \models \varphi$ iff $M_{rg}, [(l, v)] \models \mathrm{cr}(\varphi)$.

Proof: [Induction on the length of formulae]
("Left-to-right") It is obvious that the thesis holds for all the propositional variables and their negations. Next, assume that the hypothesis holds for all the proper sub-formulae of φ. If φ is equal to either $\alpha \wedge \beta$ or $\alpha \vee \beta$, then it is easy to check that the lemma holds. Consider φ to be of the following forms:

- $\varphi = \mathrm{E}(\alpha \mathrm{U}_I \beta)$. By Definition 3.1, we have that $\mathcal{G}(\mathcal{A}), (l, v) \lfloor X \models \mathrm{E}(\alpha \mathrm{U}_I \beta)$ if

$$(\exists\, \rho \in f_{\mathcal{A}}((l, v) \lfloor X))(\exists r \in I)[\mathcal{G}(\mathcal{A}), \pi_\rho(r) \models \beta \text{ and } (\forall r' < r)\ \mathcal{G}(\mathcal{A}), \pi_\rho(r') \models \alpha] \qquad (1)$$

By the definition of run we have that ρ must be of the following form:

$$(l_0, v_0) \overset{\delta_0}{\rightarrow} (l_0, v_0) + \delta_0 \overset{\sigma_0}{\rightarrow} (l_1, v_1) \overset{\delta_1}{\rightarrow} (l_1, v_1) + \delta_1 \overset{\sigma_1}{\rightarrow} (l_2, v_2) \overset{\delta_2}{\rightarrow} \ldots \qquad (2)$$

where $(l_0, v_0) = (l, v) \lfloor X$, and $\delta_i \in \mathbb{R}_+$ for all $i \geq 0$. Since ρ is progressive, we have that $r = \sum_{j=0}^{i-1} \delta_j + \delta$ for some $0 \leq \delta < \delta_i$ and $i \geq 0$. Consider the following "augmented" run ρ^*:

$$(l_0, v_0^*) \overset{\delta_0}{\rightarrow} (l_0, v_0^*) + \delta_0 \overset{\sigma_0}{\rightarrow} (l_1, v_1^*) \overset{\delta_1}{\rightarrow} (l_1, v_1^*) + \delta_1 \overset{\sigma_1}{\rightarrow} (l_2, v_2^*) \overset{\delta_2}{\rightarrow} \ldots \qquad (3)$$

where $(\forall i \geq 0)(\forall x \in X' \setminus \{y\})\ v_i^*(x) = v_i(x)$, and $v_0^*(y) = 0$ and $(\forall i > 0)$, $v_i^*(y) = \sum_{j=0}^{i-1} \delta_j$. It is easy to notice that ρ^* is a run of $\mathcal{A}_\varphi$. Now, since the clock y cannot be reset along ρ^*, it we have that $r = v_i^*(y) + \delta$, which implies that $p_{y \in I} \in \widetilde{\mathcal{V}}(\pi_{\rho^*}(r))$. Consider the following path:

$$\begin{aligned} \pi = {} & [(l_0, v_0^*)], [(l_0, v_0^*) + \delta_0^1], \ldots, [(l_0, v_0^*) + \delta_0^{n_0}], [(l_1, v_1^*)], \\ & [(l_1, v_1^*) + \delta_1^1], \ldots, [(l_1, v_1^*) + \delta_1^{n_1}], [(l_2, v_2^*)] \ldots, \\ & \ldots \\ & [(l_{i-1}, v_{i-1}^*)], [(l_{i-1}, v_{i-1}^*) + \delta_{i-1}^1], \ldots, [(l_{i-1}, v_{i-1}^*) + \delta_{i-1}^{n_{i-1}}], [(l_i, v_i^*)], \\ & [(l_i, v_i^*) + \delta_i^1], \ldots, [(l_i, v_i^*) + \delta_i^{n_\delta}], \ldots, [(l_i, v_i^*) + \delta_i^{n_i}][(l_{i+1}, v_{i+1}^*)] \ldots, \end{aligned} \qquad (4)$$

with $\delta_i = \sum_{j=0}^{n_i} \delta_i^j$, $\delta_i^j \in (0, 1)$, and for all $j \in \{0, \ldots, n_i - 1\}$ either $(l_i, v_i^* + \delta_i^j) \cong (l_i, v_i^* + \delta_i^{j+1})$ or $[(l_i, v_i^* + \delta_i^j)] \overset{\tau}{\rightarrow}_{rg} [(l_i, v_i^* + \delta_i^{j+1})]$, and $\pi(k) = [\pi_{\rho^*}(r)]$ for

$$k = \begin{cases} \sum_{j=0}^{i-1} n_j, & \delta = 0 \\ \sum_{j=0}^{i-1} n_j + n_\delta, & \delta > 0 \end{cases}$$

By the construction of the path π, we have that π is a valid path of M_{rg} and $p_{y \in I} \in \widetilde{\mathcal{V}}_{rg}(\pi(k))$ (note that $p_{y \in I} \in \widetilde{\mathcal{V}}(\pi_{\rho^*}(r))$).

Now, consider the following two cases:

A. Let $[\pi_{\rho^*}(r)]$ be a boundary region. Since $[\pi_{\rho^*}(r)]$ is boundary, by the definition of the valuation function $\widetilde{\mathcal{V}}_{rg}$ we have that $p_b \in \widetilde{\mathcal{V}}_{rg}(\pi(k))$. Further, since (1) holds, by the induction assumption, the construction of π and Lemma 4.5, we have that $M_{rg}, \pi(k) \models \mathrm{cr}(\beta)$ and $M_{rg}, \pi(j) \models \mathrm{cr}(\alpha)$ for all $j < k$.

B. Let $[\pi_{\rho^*}(r)]$ be a non-boundary state. Then, since the interval I is of the form $[a, b]$, $[a, b)$, $(a, b]$, (a, b), $[a, \infty)$, or (a, ∞) for $a,\ b \in \mathbb{N}$, we have that there exists $r' < r$ such that $\pi_{\rho^*}(r) \cong \pi_{\rho^*}(r')$. Thus, since (1) holds, by Lemma 4.5 we have that $\mathcal{G}(\mathcal{A}), \pi_{\rho^*}(r) \models \alpha$. Further, by the induction assumption and the construction of π we have that $M_{rg}, \pi(k) \models \mathrm{cr}(\beta)$ and $M_{rg}, \pi(j) \models \mathrm{cr}(\alpha)$ for all $j \leq k$.

Further, it is easy to see that $[(l, v)] \rightarrow_{rg} \pi(0)$ [1]. Therefore, by the definition of the satisfaction relation for CTL^{y}_{-G} formulae, we conclude that $M_{rg}, [(l, v)] \models \mathrm{cr}(\varphi)$.

- $\varphi = \mathrm{A}(\alpha \mathrm{U}_I \beta)$. The proof is similar to the EU_I case.

("Right-to-left") It is obvious that the thesis holds for all the propositional variables and their negations. Next, assume that the hypothesis holds for all the proper sub-formulae of φ. If φ is equal to either $\alpha \wedge \beta$ or $\alpha \vee \beta$, then it is easy to check that the lemma holds. Consider φ to be of the following forms:

- $\varphi = \mathrm{cr}(\mathrm{E}(\alpha \mathrm{U}_I \beta))$. By definitions of cr and the satisfaction relation for CTL^{y}_{-G} formulae, we have that $M_{rg}, [(l, v)] \models \mathrm{E}_y(\mathrm{cr}(\alpha)\mathrm{U}(\mathrm{cr}(\beta) \wedge p_{y \in I_i} \wedge (p_b \vee \mathrm{cr}(\alpha))))$ if

$$(\exists [(l, v')] \in Q_{rg})([(l, v)] \rightarrow_y [(l, v')] \text{ and } (\exists \pi \in \Pi([(l, v')]))(\exists m \geq 0) \tag{3}$$
$$[M_{rg}, \pi(m) \models (\mathrm{cr}(\beta) \wedge p_{y \in I} \wedge (p_b \vee \mathrm{cr}(\alpha))) \text{ and } (\forall j < m)\ M_{rg}, \pi(j) \models \mathrm{cr}(\alpha)])$$

Observe that π is of the following form:

$$\begin{gathered}\pi = [(l_0, v_0^*)], [(l_0, v_0^*) + \delta_0^1], \ldots, [(l_0, v_0^*) + \delta_0^{n_0}], [(l_1, v_1^*)], \\ [(l_1, v_1^*) + \delta_1^1], \ldots, [(l_1, v_1^*) + \delta_1^{n_1}], [(l_2, v_2^*)] \ldots, \\ \ldots \\ [(l_{i-1}, v_{i-1}^*)], [(l_{i-1}, v_{i-1}^*) + \delta_{i-1}^1], \ldots, [(l_{i-1}, v_{i-1}^*) + \delta_{i-1}^{n_{i-1}}], [(l_i, v_i^*)], \\ [(l_i, v_i^*) + \delta_i^1], \ldots, [(l_i, v_i^*) + \delta_i^{n_\delta}], \ldots, [(l_i, v_i^*) + \delta_i^{n_i}][(l_{i+1}, v_{i+1}^*)] \ldots,\end{gathered} \tag{5}$$

such that $(l_0, v_0^*) = (l, v')$, $\delta_i^j \in (0, 1)$, for all $j \in \{0, \ldots, n_i - 1\}$ either $(l_i, v_i^* + \delta_i^j) \cong (l_i, v_i^* + \delta_i^{j+1})$ or $[(l_i, v_i^* + \delta_i^j)] \xrightarrow{\tau}_{rg} [(l_i, v_i^* + \delta_i^{j+1})]$, and $v_0^*(y) = 0$ and $(\forall i > 0)$, $v_i^*(y) = \sum_{j=0}^{i-1} \sum_{t=1}^{n_i} \delta_j^t$. Consider the following "augmented" run ρ^*:

$$\begin{gathered}(l_0, v_0^*) \overset{\delta_0}{\rightarrow} (l_0, v_0^*) + \delta_0 \overset{\sigma_0}{\rightarrow} (l_1, v_1^*) \overset{\delta_1}{\rightarrow} (l_1, v_1^*) + \delta_1 \overset{\sigma_1}{\rightarrow} (l_2, v_2^*) \overset{\delta_2}{\rightarrow} \ldots \\ \ldots \\ (l_{i-1}, v_{i-1}^*) \overset{\delta_{i-1}}{\rightarrow} (l_{i-1}, v_{i-1}^*) + \delta_{i-1} \overset{\sigma_{i-1}}{\rightarrow} (l_i, v_i^*) \overset{\delta_{n_\delta}}{\rightarrow} (l_i, v_i^*) + \delta_i^{n_\delta} \ldots\end{gathered} \tag{7}$$

[1] Note that $\pi(0) = [(l, v_0^*)]$, $v_0^* \lfloor X = v \lfloor X$, and $v_0^*(y) = 0$.

where $\delta_i = \sum_{j=0}^{n_i} \delta_i^j$ for $i \geq 0$. Next, take the following run ρ:

$$\begin{gathered}(l_0, v_0) \overset{\delta_0}{\to} (l_0, v_0) + \delta_0 \overset{\sigma_0}{\to} (l_1, v_1) \overset{\delta_1}{\to} (l_1, v_1) + \delta_1 \overset{\sigma_1}{\to} (l_2, v_2) \overset{\delta_2}{\to} \dots \\ \dots \\ (l_{i-1}, v_{i-1}) \overset{\delta_{i-1}}{\to} (l_{i-1}, v_{i-1}) + \delta_{i-1} \overset{\sigma_{i-1}}{\to} (l_i, v_i) \overset{\delta_{n_\delta}}{\to} (l_i, v_i) + \delta_i^{n_\delta} \dots\end{gathered} \tag{8}$$

where for all $i \geq 0$, $v_i = v_i^* \downharpoonright X$. Then, associate with ρ a dense path $\pi_\rho : \mathbb{R} \mapsto Q$ such that $\pi_\rho(r) = (l_i, v_i) + \delta$ and $0 \leq \delta \leq \delta_i$ $r = \sum_{j=0}^{i-1} \delta_j + \delta_i$. Moreover, assume that $\pi(m) = [\pi_\rho(r_m)]$ for $r_m = \sum_{j=0}^{i-1} \delta_j + \delta_i^{n_\delta}$. Since (3) holds, by the construction of the run ρ, the inductive assumption, and Lemma 4.5 we have that

$$\mathcal{G}(\mathcal{A}), \pi_\rho(r_m) \models \beta \wedge p_{y \in I} \wedge (p_b \vee \alpha) \tag{9}$$

and for all $r' \leq r_{m-1}$ with $[\pi_\rho(r_{m-1})] = \pi(m-1)$

$$\mathcal{G}(\mathcal{A}), \pi_\rho(r') \models \alpha \tag{10}$$

Since (9) holds, we have that $p_{y \in I} \in \widetilde{\mathcal{V}}(\pi_\rho(r_m))$. This implies that $r \in I$. So, to conclude that $\mathcal{G}(\mathcal{A}), q \models \mathrm{E}(\alpha \mathrm{U}_I \beta)$, it remains to show that for all $r_{m-1} < r'' < r_m$, $\mathcal{G}(\mathcal{A}), \pi_\rho(r'') \models \alpha$ holds.

Consider the following two cases:

- Let $\mathcal{G}(\mathcal{A}), \pi_\rho(r_m) \models \alpha$. Then, by the construction of the run ρ, we have that for all $r_{m-1} < r'' < r_m$ either $\pi_\rho(r_m) \cong \pi_\rho(r'')$ or $\pi_\rho(r_{m-1}) \cong \pi_\rho(r'')$. Since $\mathcal{G}(\mathcal{A}), \pi_\rho(r_m) \models \alpha$ and Condition (10) and Lemma 4.5 holds, we have that for all $r_{m-1} < r'' < r_m$, $\mathcal{G}(\mathcal{A}), \pi_\rho(r'') \models \alpha$.
- Let $\mathcal{G}(\mathcal{A}), \pi_\rho(r_m) \models p_b$. Then, by the construction of the run ρ, we have that for all $r_{m-1} < r'' < r_m$, $\pi_\rho(r_{m-1}) \cong \pi_\rho(r'')$. Since Condition (10) and Lemma 4.5 hold, we have that for all $r_{m-1} < r'' < r_m$, $\mathcal{G}(\mathcal{A}), \pi_\rho(r'') \models \alpha$.

Therefore, we conclude that $\mathcal{G}(\mathcal{A}), \pi_\rho(0) \models \mathrm{E}(\alpha \mathrm{U}_I \beta)$. Since $\pi_\rho(0) = (l_0, v_0) = (l, v \downharpoonright X)$, we have that $\mathcal{G}(\mathcal{A}), (l, v \downharpoonright X) \models \mathrm{E}(\alpha \mathrm{U}_I \beta)$.

- $\varphi = cr(\mathrm{A}(\alpha \mathrm{U}_I \beta))$. The proof is similar to the EU_I case.

□

6. Discretisation

Here we show how to represent the dense model by a specially chosen representatives that enjoys the following two properties. Firstly, the discretised model (see Definition 6.1) that is defined over these special representatives preserves validity of CTL_{-G}^{y} formulae. Secondly, both the region graph and the discretised model (build for the same automaton) preserve the same set of CTL_{-G}^{y} formulae; the reason why we introduce this new model, is that the discretised model constitutes the base for the implementation of the BMC method, presented later on.

We start off with some auxiliary notions. Let $\mathbb{Q}$ be a set of rational numbers. For every $m \in \mathbb{N}$, we define D_m by $\{\frac{n}{2^m} \mid n \in \mathbb{N}\}$, E_m by $D_m \setminus \{0\}$, the set D of the discretised clock values as $\bigcup_{m=0}^{\infty} D_m$, and the set E of labels as $\bigcup_{m=1}^{\infty} E_m$. The following two lemmas constitute the base for the definition of the discretised model.

Lemma 6.1. ([37])
For every $v \in \mathbb{R}^X$ there exist $u \in D^X$ such that $u \cong v$.

Lemma 6.2. ([37])
Let $v \in \mathbb{R}^X$ be a clock valuation, $\delta \in \mathbb{R}$, and $m \in \mathbb{N}$. Then for each $u \in D_m^X$ such that $u \cong v$ there exists $\delta' \in E_{m+1}$ such that $v + \delta \cong u + \delta'$. Moreover, $u + \delta' \in D_{m+1}^X$.

We can now define a discretised model for $\mathcal{A}$.

Definition 6.1. (Discretised model)
Let $\mathcal{A} = (\Sigma, L, l^0, X, \mathcal{I}, R, \mathcal{V})$ be a timed automaton. A *discretised model* for $\mathcal{A}$ is the tuple $M = (\Sigma \cup E, S, s^0, \widetilde{\mathcal{V}}_d, \rightarrow_d)$, where $S = L \times D^X$ is a set of (discretised) states, $s^0 = (l^0, v^0)$ is the initial state, $\widetilde{\mathcal{V}}_d : S \mapsto 2^{\mathcal{PV}}$ is a valuation function such that $\widetilde{\mathcal{V}}_d((l,v)) = \mathcal{V}(l)$, and the transition relation $\rightarrow_d$ is defined as follows:

- Time transitions: for any $\delta \in E$, $(l,v) \xrightarrow{\delta}_d (l, v+\delta)$ iff $(l,v) \xrightarrow{\delta} (l, v+\delta)$ in $\mathcal{G}(\mathcal{A})$ and $(\forall \delta' \leq \delta)$ $v + \delta' \cong v$ or $v + \delta' \cong v + \delta$
- Action transitions: for any $\sigma \in \Sigma$, $(l,v) \xrightarrow{\sigma}_d (l', v')$ iff $(\exists v'')(\exists \delta)$ such that $(l,v) \xrightarrow{\delta}_d (l, v'')$ and $(l, v'') \xrightarrow{\sigma} (l', v')$ in $\mathcal{G}(\mathcal{A})$

Definition 6.2. (Satisfaction for CTL_{-G}^{y} over the discretised model)
Let $M = (\Sigma \cup E, S, s^0, \widetilde{\mathcal{V}}_d, \rightarrow_d)$ be the discretised model for $\mathcal{A}_\varphi$, $s \in S$, α, β formulae of CTL_{-G}^{y}, $\rightarrow_{\mathcal{A}}$ denote the part of $\rightarrow_d$, where transitions are labelled with elements of $\Sigma \cup E$, and $\rightarrow_y$ denote the transitions that reset the clock y. A *path* π in M is a sequence $(s_0, s_1, \ldots)$ of states such that $s_i \rightarrow_{\mathcal{A}} s_{i+1}$ for each $i \in \mathbb{N}$, and $\Pi(s)$ denotes the set of all the paths starting at s in M. The satisfaction relation $\models$ is defined inductively as follows:

$$\begin{array}{ll}
M, s \models p & \text{iff}\, p \in \widetilde{\mathcal{V}}_d(s), \\
M, s \models \neg p & \text{iff}\, p \notin \widetilde{\mathcal{V}}_d(s), \\
M, s \models \alpha \vee \beta & \text{iff}\, M, s \models \alpha \ \text{ or } \ M, s \models \beta, \\
M, s \models \alpha \wedge \beta & \text{iff}\, M, s \models \alpha \ \text{ and } \ M, s \models \beta, \\
M, s \models \mathrm{E}_y(\alpha \mathrm{U} \beta) & \text{iff}\, (\exists s' \in S)(s \rightarrow_y s' \text{ and} \\
& \quad (\exists \pi \in \Pi(s'))(\exists m \geq 0)\ [M, \pi(m) \models \beta \text{ and } (\forall j < m)\ M, \pi(j) \models \alpha]), \\
M, s \models \mathrm{A}_y(\alpha \mathrm{U} \beta) & \text{iff}\, (\exists s' \in S)(s \rightarrow_y s' \text{ and} \\
& \quad (\forall \pi \in \Pi(s'))(\exists m \geq 0)\ [M, \pi(m) \models \beta \text{ and } (\forall j < m)\ M, \pi(j) \models \alpha]).
\end{array}$$

A CTL_{-G}^{y} formula φ is *valid in* M (denoted $M \models \varphi$) iff $M, s^0 \models \varphi$, i.e., φ is true at the initial state of M.

We will now prove that the discretised model preserves validity of CTL_{-G}^{y} formulae. We start off with some auxiliary lemmas 6.3- 6.6.

Lemma 6.3. Let X be a set of clocks, and $u, v \in D^X$ be clock valuations such that $u \cong v$. Then, for any clock constraint $\mathfrak{cc} \in C(X)$, $u \in [\![\mathfrak{cc}]\!]$ iff $v \in [\![\mathfrak{cc}]\!]$.

Proof: Straightforward induction on clock constraints. □

Lemma 6.4. Let u, v be clock valuations such that $u \cong v$. For every $\delta \in E$ there exists $\delta' \in E$ such that $u + \delta \cong v + \delta'$.

Proof: We omit the proof as it is analogous to the proof of the Lemma 4.3 of [36] □

Lemma 6.5. Let $\sigma \in \Sigma$, and let s_1, s_2 be states of a discretised model M such that $s_1 \cong s_2$. Then, for each state $s_3 \in S$ such that $s_1 \xrightarrow{\sigma}_d s_3$, there exists a state $s_4 \in S$ such that $s_2 \xrightarrow{\sigma}_d s_4$ and $s_3 \cong s_4$.

Proof: Straightforward, by using Lemma 6.3. □

Lemma 6.6. Let s_0, s'_0 be two states of a discretised model M such that $s_0 \cong s'_0$, and π a path in M such that $\pi(0) = s_0$. Then, there exists a path π' of M such that $\pi'(0) = s'_0$ and for each $i > 0$, $\pi(i) \cong \pi'(i)$.

Proof: Straightforward, by Lemmas 6.4 and 6.5. □

We can now show that states belonging to the same region (in fact, to the region restricted to the discretised states only) satisfies the same set of CTL^y_{-G} formulae.

Lemma 6.7. Let $\mathcal{A}$ be a timed automaton, φ a TCTL_{-G} formula, M a discretised model for $\mathcal{A}_\varphi$, l a location in $\mathcal{A}$, and u, v two clock valuations such that $v \cong u$. Then, $M, (l, v) \models \mathrm{cr}(\varphi)$ iff $M, (l, u) \models \mathrm{cr}(\varphi)$.

Proof: [Induction on the length of TCTL_{-G} formulae]. It is easy to see that the thesis holds for all the propositional variables and for all the negations of propositional variables. If φ is of the form $\alpha \vee \beta$ or $\alpha \wedge \beta$, then the proof of the thesis is straightforward. So, it remains to prove that the thesis holds for formulae of the form $\mathrm{E}(\alpha \mathrm{U}_I \beta)$ and $\mathrm{A}(\alpha \mathrm{U}_I \beta)$.

Consider a formula of the form $\mathrm{E}(\alpha \mathrm{U}_I \beta)$ and suppose that $M, (l, v) \models \mathrm{cr}(\mathrm{E}(\alpha \mathrm{U}_I \beta))$. By the definition of cr and the satisfaction relation over the discretised model we have

$$\begin{gathered}(\exists (l', v') \in S)((l, v) \to_y (l', v') \text{ and } (\exists \pi \in \Pi((l', v')))(\exists m \geq 0) \\ [M, \pi(m) \models (\mathrm{cr}(\beta) \wedge p_{y \in I_i} \wedge (p_b \vee \mathrm{cr}(\alpha))) \text{ and } (\forall j < m)\ M, \pi(j) \models \mathrm{cr}(\alpha)])\end{gathered} \tag{11}$$

Since $(l, v) \cong (l, u)$ and $(l, v) \to_y (l', v')$, by Lemma 6.5 we have that there exists a state (l'', v'') such that $(l, u) \to_y (l'', v'')$ and $(l'v') \cong (l'', v'')$. By Lemma 6.6 there exists a path π' in M such that $\pi'(0) = (l'', v'')$, and for each $i \geq 0$, $\pi(i) \cong \pi'(i)$. Thus, by the induction hypotheses and by the cases for conjunction and disjunction we have that

$$M, \pi'(m) \models (\mathrm{cr}(\beta) \wedge p_{y \in I_i} \wedge (p_b \vee \mathrm{cr}(\alpha))) \text{ and } (\forall j < m)\ M, \pi'(j) \models \mathrm{cr}(\alpha)$$

Thus, it follows that $M, (l, u) \models \mathrm{cr}(\mathrm{E}(\alpha \mathrm{U}_I \beta))$.

The proof of the case $\mathrm{A}(\alpha \mathrm{U}_I \beta)$ is analogous. □

We conclude the section with a lemma that stays that both the region graph and the discretised model for a given automaton preserve validity of the same set of CTL^{y}_{-G} formulae.

Lemma 6.8. Let $\mathcal{A}$ be a timed automaton, φ a TCTL_{-G} formula, M_{rg} a region graph for $\mathcal{A}_\varphi$, and M a discretised model for $\mathcal{A}_\varphi$. Moreover, for any state $q \in S$, let $[q]$ denote the equivalence class of q induced by the relation $\cong$. Then, for any $q \in S$, $M, q \models \mathrm{cr}(\varphi)$ if, and only if $M_{rg}, [q] \models \mathrm{cr}(\varphi)$.

Proof: It is easy to see that the thesis holds for all the propositional variables and for all the negations of propositional variables. If φ is of the form $\alpha \vee \beta$ or $\alpha \wedge \beta$, then the proof of the thesis is straightforward. So, it remains to prove that the thesis holds for formulae of the form $\mathrm{E}(\alpha \mathrm{U}_I \beta)$ and $\mathrm{A}(\alpha \mathrm{U}_I \beta)$.

Consider a formula φ to be of the form $\mathrm{E}(\alpha \mathrm{U}_I \beta)$, and suppose that $M, q \models \mathrm{cr}(\mathrm{E}(\alpha \mathrm{U}_I \beta))$. By the definitions of cr and the satisfaction relation for the discretised model we have

$$\begin{gathered}(\exists q' \in S)(q \rightarrow_y q' \text{ and } (\exists \pi \in \Pi(q'))(\exists m \geq 0) \\ [M, \pi(m) \models (\mathrm{cr}(\beta) \wedge p_{y \in I_i} \wedge (p_b \vee \mathrm{cr}(\alpha))) \text{ and } (\forall j < m)\ M, \pi(j) \models \mathrm{cr}(\alpha)])\end{gathered} \tag{12}$$

Consider now the following path π_{rg}:

$$\pi_{rg} = [\pi(0)], [\pi(1)], [\pi(2)], \ldots \tag{13}$$

It is obvious that π_{rg} is a valid path in M_{rg}. Moreover, by Lemma 6.7 we have that

$$M_{rg}, [\pi(m)] \models (\mathrm{cr}(\beta) \wedge p_{y \in I_i} \wedge (p_b \vee \mathrm{cr}(\alpha))) \text{ and } (\forall j < m)\ M_{rg}, [\pi(j)] \models \mathrm{cr}(\alpha) \tag{14}$$

Since $q \rightarrow_y q'$, by definitions of $\rightarrow_d$ and $\rightarrow_{rg}$ we have that $[q] \rightarrow_y [q']$. Therefore, since (14) holds, it follows that $M_{rg}, [q] \models \mathrm{cr}(\mathrm{E}(\alpha \mathrm{U}_I \beta))$.

Conversely, suppose that $M_{rg}, [q] \models \mathrm{cr}(\mathrm{E}(\alpha \mathrm{U}_I \beta))$. By definitions of cr and the satisfaction relation over the region graph we have

$$\begin{gathered}(\exists [q'] \in Q_{rg})([q] \rightarrow_y [q'] \text{ and } (\exists \pi \in \Pi([q']))(\exists m \geq 0) \\ [M_{rg}, \pi(m) \models (\mathrm{cr}(\beta) \wedge p_{y \in I_i} \wedge (p_b \vee \mathrm{cr}(\alpha))) \text{ and } (\forall j < m)\ M_{rg}, \pi(j) \models \mathrm{cr}(\alpha)])\end{gathered} \tag{15}$$

Observe that π is of the following form:

$$\begin{gathered}\pi = [(l_0, v_0)], [(l_0, v_0) + \delta_0^1], \ldots, [(l_0, v_0) + \delta_0^{n_0}], [(l_1, v_1)], \\ [(l_1, v_1) + \delta_1^1], \ldots, [(l_1, v_1) + \delta_1^{n_1}], [(l_2, v_2)] \ldots,\end{gathered} \tag{16}$$

with $(l_0, v_0) = q'$, $\delta_i^j \in E$, and all $j \in \{0, \ldots, n_i - 1\}$ such that either $(l_i, v_i + \delta_i^j) \cong (l_i, v_i + \delta_i^{j+1})$ or $[(l_i, v_i + \delta_i^j)] \xrightarrow{\tau}_{rg} [(l_i, v_i + \delta_i^{j+1})]$. Next, take the following path π_d:

$$\pi = (l_0, v_0), (l_0, v_0) + \delta_0^1, \ldots, (l_0, v_0) + \delta_0^{n_0}, (l_1, v_1), (l_1, v_1) + \delta_1^1, \ldots, (l_1, v_1) + \delta_1^{n_1}, (l_2, v_2) \ldots, \tag{17}$$

Since (15) holds, by the construction of π_d and Lemma 6.7 we have that

$$M, \pi_d(m) \models (\mathrm{cr}(\beta) \wedge p_{y \in I_i} \wedge (p_b \vee \mathrm{cr}(\alpha))) \text{ and } (\forall j < m)\ M, \pi_d(j) \models \mathrm{cr}(\alpha) \tag{18}$$

Since $[q] \rightarrow_y [q']$ holds, by the definition of $\rightarrow_y$, we have that $q \rightarrow_y q'$. Therefore, since (18) holds, it follows that $M, q \models \mathrm{cr}(\mathrm{E}(\alpha \mathrm{U}_I \beta))$.

The proof of the case $\mathrm{A}(\alpha \mathrm{U}_I \beta)$ is analogous. □

7. Bounded Model Checking for TECTL$_{-G}$

Let $\mathcal{A}$ be a time automaton, and φ a TECTL_{-G} formula, To perform bounded model checking for TECTL_{-G} we proceed by extending the technique used for TECTLK and diagonal-free automata [33]. Namely, we first translate the model checking problem from TECTL_{-G} into that problem for ECTL^{y}_{-G} as in Section 5, and then we define BMC for ECTL^{y}_{-G}.

7.1. BMC for ECTL$^{y}_{-G}$.

All the known BMC techniques are based on so called $k-$bounded semantics. In particular, BMC for ECTL^{y}_{-G} is based on the $k-$bounded semantics for ECTL^{y}_{-G}, the definition of which we present below.

7.1.1. Bounded Semantics

We start off with some auxiliary definitions. Let $M = (\Sigma \cup E, S, s^0, \widetilde{\mathcal{V}}_d, \rightarrow_d)$ be a discretised model, and $k \in \mathbb{N}_+$ a bound [2]. As before, we denote by $\rightarrow_{\mathcal{A}}$ the part of $\rightarrow_d$, where transitions are labelled with elements of $\Sigma \cup E$, and by $\rightarrow_y$ the transitions that reset the clock y. A $k-$*path* π in M is a finite sequence of states $(s_0, \ldots, s_k)$ such that $s_i \rightarrow_{\mathcal{A}} s_{i+1}$ for each $0 \leq i < k$, and the set of all the k-paths starting at s in M is denoted by $\Pi_k(s)$. A *k-model* for M is a structure $M_k = (\Sigma \cup E, S, s^0, \widetilde{\mathcal{V}}_d, P_k, P_y)$, where $P_k = \bigcup_{s \in S} \Pi_k(s)$ and $P_y = \{(s, s') \mid s \rightarrow_y s' \text{ and } s, s' \in S\}$; note that the set $P_k \cup P_y$ completely reflect the relation $\rightarrow_d$ of M.

We can now define a bounded semantics for ECTL^{y}_{-G} formulae. Let α, β be ECTL^{y}_{-G} formulae, $k \in \mathbb{N}_+$ a bound, and $M_k, s \models \alpha$ denotes that α is true at the state s of M_k. The bounded satisfaction relation $\models$ is defined inductively as follows:

$$M_k, s \models p \quad \text{iff } p \in \widetilde{\mathcal{V}}_d(s), \quad M_k, s \models \alpha \vee \beta \text{ iff } M_k, s \models \alpha \text{ or } M_k, s \models \beta,$$
$$M_k, s \models \neg p \text{ iff } p \notin \widetilde{\mathcal{V}}_d(s), \quad M_k, s \models \alpha \wedge \beta \text{ iff } M_k, s \models \alpha \text{ and } M_k, s \models \beta,$$
$$M_k, s \models \mathrm{E}_y(\alpha \mathrm{U} \beta) \text{ iff } (\exists s' \in S)((s, s') \in P_y \text{ and } (\exists \pi \in \Pi_k(s'))(\exists 0 \leq j \leq k)(M_k, \pi(j) \models \beta$$
$$\text{and } (\forall 0 \leq i < j) M_k, \pi(i) \models \alpha),$$

An ECTL^{y}_{-G} formula φ is *valid in k-model* M_k (denoted $M \models_k \varphi$) iff $M_k, s^0 \models \varphi$, i.e., φ is true at the initial state of the k-model M_k.

We can now describe how the model checking problem ($M \models \varphi$) can be reduced to the bounded model checking problem ($M \models_k \varphi$).

Lemma 7.1. Let $k \in \mathbb{N}_+$, M be a discretised model, M_k its k-model, and φ an ECTL^{y}_{-G} formula. Then, for each state s of M, $M_k, s \models \varphi$ implies $M, s \models \varphi$.

Proof: By straightforward induction on the length of φ. □

Lemma 7.2. Let M be a discretised model and φ an ECTL^{y}_{-G} formula. Then, for each state s of M such that $M, s \models \varphi$, there exists $k \in \mathbb{N}_+$ such that $M_k, s \models \varphi$.

[2] By $\mathbb{N}_+$ we denote the set of positive natural numbers, i.e., the set $\{1, 2, 3, \ldots\}$.

Proof: [Induction on the length of φ] The lemma follows directly for the propositional variables and their negations. Next, assume that the hypothesis holds for all the proper sub-formulae of φ. If φ is equal to either $\alpha \wedge \beta$ or $\alpha \vee \beta$, then it is easy to check that the lemma holds. Consider φ to be of the following forms:

- $\varphi = \mathrm{E}_y(\alpha \mathrm{U} \beta)$. By the definition of the *unbounded* semantics we have that there exists a state s' in M such that $s \rightarrow_y s'$ and there exists a path $\pi \in \Pi(s')$ such that there exists $m \geq 0$ with $M, \pi(m) \models \beta$ and $(\forall 0 \leq i < m)(M, \pi(i) \models \alpha)$. Thus, by the inductive assumption we have that there exists $k_m \geq m$ such that $M_{k_m}, \pi(m) \models \beta$, and there exists $k_i \geq i$ such that $M_{k_i}, \pi(i) \models \alpha$ for all $0 \leq i < m$. Now, take $k = max\{k_0, \ldots, k_m\}$ and consider the prefix π_k of length k of the path π. It is obvious that $\pi_k \in \Pi_k(s')$. By the definition of the k-model, $(s, s') \in P_y$. Therefore, by the definition of the *bounded* semantics we have that $M_k, s \models \mathrm{E}_y(\alpha \mathrm{U} \beta)$.

□

Theorem 7.1. Let M be a discretised model and φ an ECTL^y_{-G} formula. Then, $M \models \varphi$ iff there exists $k \in \mathbb{N}_+$ such that $M \models_k \varphi$.

Proof: Follows from Lemmas 7.1 and 7.2. □

7.1.2. Submodels of k-models

The previous section ends with the following conclusion: to prove that an ECTL^y_{-G} formula ψ holds in a discretised model M, it is enough to prove that ψ holds in its k-model M_k, for some $k \in \mathbb{N}_+$. In this subsection we will show a stronger property. Namely, we will prove that ψ holds in M if and only if ψ holds in a s^0-*submodel* of M_k. We start by defining the notion of this submodel.

Definition 7.1. (Submodel)
A s-*submodel* of k-model $M_k = (\Sigma \cup E, S, s^0, \widetilde{\mathcal{V}}_d, P_k, P_y)$ is a tuple $M'(s) = (\Sigma \cup E, S', s, \widetilde{\mathcal{V}}'_d, P'_k, P'_y)$, such that $P'_k \subseteq P_k$, $S' = \{r \in S \mid (\exists \pi \in P'_k)(\exists i \leq k)\pi(i) = r\} \cup \{s\}$, $P'_y \subseteq P_y \cap S' \times S'$, and $\widetilde{\mathcal{V}}'_d = \widetilde{\mathcal{V}}_d \restriction S'$.

The bounded semantics for ECTL^y_{-G} over a s-submodel $M'(s)$ is defined as for M_k.

We will now introduce a definition of a function f_k which will give us a bound on the number of k-paths in the s-submodel $M'(s)$ and a function $f_{k,y}$ that gives a bound on the number of elements of the set P'_y in the s-submodel $M'(s)$. We will show later that these two bounds guarantee that the validity of ψ in $M'(s)$ is equivalent to the validity of ψ in M_k. The function $f_k : \mathcal{WF} \mapsto \mathbb{N}$ is defined by:

- $f_k(p) = f_k(\neg p) = 0$, where $p \in \mathcal{PV}'$,
- $f_k(\alpha \vee \beta) = max\{f_k(\alpha), f_k(\beta)\}$,
- $f_k(\alpha \wedge \beta) = f_k(\alpha) + f_k(\beta)$,
- $f_k(\mathrm{E}_y(\alpha \mathrm{U} \beta)) = k \cdot f_k(\alpha) + f_k(\beta) + 1$,

The function $f_{k,y} : \mathcal{WF} \mapsto \mathbb{N}$ is defined by:

- $f_{k,y}(p) = f_{k,y}(\neg p) = 0$, where $p \in \mathcal{PV}'$,
- $f_{k,y}(\alpha \vee \beta) = max\{f_{k,y}(\alpha), f_{k,y}(\beta)\}$,

- $f_{k,y}(\alpha \wedge \beta) = f_{k,y}(\alpha) + f_{k,y}(\beta)$,
- $f_{k,y}(\mathrm{E}_y(\alpha \mathrm{U} \beta)) = k \cdot f_{k,y}(\alpha) + f_{k,y}(\beta) + 1$.

Lemma 7.3. Let $M'(s)$ and $M''(s)$ be two s-submodels of M_k with $P'_k \subseteq P''_k$, $P'_y \subseteq P''_y$, and ψ an ECTL^{y}_{-G} formula. If $M'(s) \models_k \psi$, then $M''(s) \models_k \psi$.

Proof: By straightforward induction on the length of ψ. □

Lemma 7.4. $M_k, s \models \psi$ iff there is a s-submodel $M'(s)$ of M_k with $|P'_k| \leq f_k(\psi)$ and $|P'_y| \leq f_{k,y}(\psi)$ such that $M'(s), s \models \psi$.

Proof: The 'right-to-left' implication is straightforward. To prove 'left-to-right' implication, we will use induction on the length of ψ.

The 'left-to-right' implication follows directly for the propositional variables and their negations. Assume that the hypothesis holds for all the proper sub-formulae of ψ, and consider the following cases:

- Let $\psi = \alpha \vee \beta$ and $M_k, s \models \alpha \vee \beta$. By the definition of the bounded semantics we have that $M_k, s \models \alpha$ or $M_k, s \models \beta$. Hence, by induction we have that there is a s-submodel $M'(s)$ of M_k such that $M'(s), s \models \alpha$ and $|P'_k| \leq f_k(\alpha)$ and $|P'_y| \leq f_{k,y}(\alpha)$, or there is a s-submodel $M''(s)$ of M_k such that $M''(s), s \models \beta$ and $|P''_k| \leq f_k(\beta)$ and $|P''_y| \leq f_{k,y}(\beta)$. Now, consider a s-submodel $M'''(s)$ of M_k such that $P'''_k = P'_k$ and $P'''_y = P'_y$ if $M'(s), s \models \alpha$, $P'''_k = P''_k$ and $P'''_y = P''_y$ otherwise. Thus, $|P'''_k| \leq max\{f_k(\alpha), f_k(\beta)\}$ and $|P'''_y| \leq max\{f_{k,y}(\alpha), f_{k,y}(\beta)\}$. It is obvious that $M'''(s), s \models \alpha$ or $M'''(s), s \models \beta$. Therefore, by the definition of the bounded semantics we have that $M'''(s), s \models \alpha \vee \beta$.

- Let $\psi = \alpha \wedge \beta$ and $M_k, s \models \alpha \wedge \beta$. By the definition of the bounded semantics we have that $M_k, s \models \alpha$ and $M_k, s \models \beta$. Hence, by induction we have that there is a s-submodel $M'(s)$ of M_k such that $M'(s), s \models \alpha$ and $|P'_k| \leq f_k(\alpha)$ and $|P'_y| \leq f_{k,y}(\alpha)$, and there is a s-submodel $M''(s)$ of M_k such that $M''(s), s \models \beta$ and $|P''_k| \leq f_k(\beta)$ and $|P''_y| \leq f_{k,y}(\beta)$. Now, consider the s-submodel $M'''(s)$ of M_k such that $P'''_k = P'_k \cup P''_k$ and $P'''_y = P'_y \cup P''_y$. It is easy to observe that $|P'''_k| \leq f_k(\alpha) + f_k(\beta)$ and $|P'''_y| \leq f_{k,y}(\alpha) + f_{k,y}(\beta)$. So, by Lemma 7.3, we have that $M'''(s), s \models \alpha$ and $M'''(s), s \models \beta$. Therefore, by the definition of the bounded semantics we have that $M'''(s), s \models \alpha \wedge \beta$.

- Let $\psi = \mathrm{E}_y(\alpha \mathrm{U} \beta)$ and $M_k, s \models \mathrm{E}_y(\alpha \mathrm{U} \beta)$. By the definition of the bounded semantics, there is a state $s' \in S$ such that $(s, s') \in P_y$ and there is a $k-$path $\pi \in \Pi_k(s')$ such that

$$(\exists 0 \leq m \leq k)(M_k, \pi(m) \models \beta \text{ and } (\forall 0 \leq i < m) M_k, \pi(i) \models \alpha) \tag{19}$$

Hence, by the inductive assumption, for all i such that $0 \leq i < m$ there are $\pi(i)$-submodels $M^i(\pi(i))$ of M_k with $|P^i_k| \leq f_k(\alpha)$ and $|P^i_y| \leq f_{k,y}(\alpha)$ and

$$M^i(\pi(i)), \pi(i) \models \alpha \tag{20}$$

and there is a $\pi(m)$-submodel $M^m(\pi(m))$ of M_k with $|P^m_k| \leq f_k(\beta)$ and $|P^m_y| \leq f_{k,y}(\beta)$ and

$$M^m(\pi(m)), \pi(m) \models \beta \tag{21}$$

Consider a s-submodel $M'(s)$ of M_k such that $P'_k = \bigcup_{i=0}^{m} P^i_k \cup \{\pi\}$ and $P'_y = \bigcup_{i=0}^{m} P^i_y \cup \{(s,s')\}$. Thus, by the construction of $M'(s)$, we have that $(s,s') \in P'_y$ and $\pi \in P'_k$. Therefore, since conditions (19), (20), and (21) hold, by the definition of the bounded semantics, we have that $M', s \models \mathrm{E}_y(\alpha \mathrm{U} \beta)$ and $|P'_k| \leq k \cdot f_k(\alpha) + f_k(\beta) + 1$ and $|P'_y| \leq k \cdot f_{k,y}(\alpha) + f_{k,y}(\beta) + 1$.

□

From Lemma 7.4 we can now derive the following.

Corollary 7.1. $M_k, s^0 \models \psi$ iff there is a s^0-submodel $M'(s^0)$ of M_k with $|P'_k| \leq f_k(\psi)$ and $|P'_y| \leq f_{k,y}(\psi)$ such that $M'(s^0), s^0 \models \psi$.

Proof: It follows from the definition of bounded semantics and Lemma 7.4, by using $s = s^0$. □

Theorem 7.2. Let M be a discretised model, M_k its k-model, and ψ an ECTL^{y}_{-G} formula. $M \models \psi$ iff there exists $k \in \mathbb{N}_+$ and there exists s^0-submodel $M'(s^0)$ of M_k with $P'_k \leq f_k(\psi)$ and $|P'_y| \leq f_{k,y}(\psi)$ such that $M'(s^0) \models_k \psi$.

Proof: Follows from Theorem 7.1 and Corollary 7.1. □

7.1.3. Translation to Boolean formulae

As it was mentioned before, the main idea of BMC for ECTL^{y}_{-G} consists in translating the model checking problem for ECTL^{y}_{-G} into the problem of satisfiability of a propositional formula. Given an ECTL^{y}_{-G} formula ψ and a discretised model M, this proposition formula is of the following form:

$$[M,\psi]_k = [M^{\psi,s^0}]_k \wedge [\psi]_k^{[0,0]}$$

The first conjunct of $[M,\psi]_k$ represents all the possible s^0-submodels of M that consist of $f_k(\psi)$ $k-$paths of M. The second conjunct of $[M,\psi]_k$ encodes a number of constraints that must be satisfied on the s^0-submodels of M, which consists of all the k-paths of M, for ψ to be satisfied. Once this translation is defined, checking whether $[M,\psi]_k$ is satisfiable is done by using a SAT-checker.

Assume that each state s of the discretised model M is encoded by a bit-vector whose length, say b, depends on the number of locations, the number of clocks, and on the bound $k \in \mathbb{N}_+$. Then, each state s of M can be represented by a vector $w = (w[1], \ldots, w[b])$ (called a *global state variable*), where each $w[i]$ is a propositional variable for $i = 1, \ldots, b$. A finite sequence $(w_0, \ldots, w_k)$ of global state variables is called a *symbolic k-path*[3].

A finite sequence $(w_0, \ldots, w_k)$ of global state variables is called a *symbolic k-path*. In general, we need to consider not just one but a number of symbolic k-paths. This number depends on the formula ψ under investigation, and it is returned as the value $f_k(\psi)$ of the function f_k. The j-th symbolic k-path is denoted by $w_{0,j}, \ldots, w_{k,j}$, where $w_{i,j}$ are global state variables for $1 \leq j \leq f_k(\psi)$, $0 \leq i \leq k$. For two global state variables w, w', we define the following propositional formulae:

- $I_s(w)$ is a formula over w, which is true for a valuation s_w of w iff $s_w = s$.
- $p(w)$ is a formula over w, which is true for a valuation s_w of w iff $p \in \widetilde{\mathcal{V}}_d(s_w)$, where $p \in \mathcal{PV}'$,

[3] In general, we consider not just one but a number of symbolic k-paths. This number depends on the formula ψ under investigation, and it is returned as the value $f_k(\psi)$ of the function f_k.

- $\mathcal{R}(w, w')$ is a formula over w, w', which is true for two valuations s_w of w and $s_{w'}$ of w' iff $s_w \rightarrow_{\mathcal{A}} s_{w'}$ (encodes the non-resetting transition relation of M),
- $R_y(w, w')$ is a formula over w, w', which is true for two valuations s_w of w and $s_{w'}$ of w' iff $s_w \rightarrow_y s_{w'}$ (encodes the transitions resetting the clock y).

The propositional formula $[M, \psi]_k$ is defined over a global state variable $w_{n,m}$, for $0 \leq m \leq k$ and $1 \leq n \leq f_k(\psi)$; the index n denotes the number of a symbolic path, whereas the index m the position at that path. The formal definition of the first conjunct of $[M, \psi]_k$ is the following:

$$[M^{\psi,s^0}]_k := I_{s^0}(w_{0,0}) \wedge \bigwedge_{n=1}^{f_k(\psi)} \bigwedge_{m=0}^{k-1} \mathcal{R}(w_{m,n}, w_{m+1,n})$$

The second conjunct of $[M, \psi]_k$, i.e. the formula $[\psi]_k^{[0,0]}$ is defined inductively as follows:

$[p]_k^{[m,n]} := p(w_{m,n})$, $\quad [\alpha \wedge \beta]_k^{[m,n]} := [\alpha]_k^{[m,n]} \wedge [\beta]_k^{[m,n]}$,
$[\neg p]_k^{[m,n]} := \neg p(w_{m,n})$, $\quad [\alpha \vee \beta]_k^{[m,n]} := [\alpha]_k^{[m,n]} \vee [\beta]_k^{[m,n]}$,
$[\mathrm{E}_y(\alpha \mathrm{U} \beta)]_k^{[m,n]} := \bigvee_{i=1}^{f_k(\psi)} (R_y(w_{m,n}, w_{0,i}) \wedge \bigvee_{j=0}^{k} ([\beta]_k^{[j,i]} \wedge \bigwedge_{l=0}^{j-1} [\alpha]_k^{[l,i]}))$.

Lemma 7.5. Let M be discretised model, M_k its k-model, and ψ an ECTL$^y_{-G}$ formula. For each state s of M, the following holds: $[M^{\psi,s}]_k \wedge [\psi]_k^{[0,0]}$ is satisfiable iff there is a submodel $M'(s)$ of M_k with $|P'_k| \leq f_k(\psi)$ and $|P'_y| \leq f_{k,y}(\psi)$ such that $M'(s), s \models \psi$.

Proof: $(=>)$ Let $[M^{\psi,s}]_k \wedge [\psi]_k^{[0,0]}$ be satisfiable. By the definition of the translation, the propositional formula $[\psi]_k^{[0,0]}$ encodes all the sets of $k-$paths of size $f_k(\psi)$ which satisfy the formula ψ and all the sets of transitions resetting the clock y of size $f_{k,y}(\psi)$. By the definition of the unfolding of the transition relation, the propositional formula $[M^{\psi,s}]_k$ encodes $f_k(\psi)$ symbolic k-paths to be valid $k-$paths of M_k. Hence, there is a set of $k-$paths in M_k, which satisfies the formula ψ, of size smaller or equal to $f_k(\psi)$, and there is a set of transitions resetting the clock y of size smaller or equal to $f_{k,y}(\psi)$. Thus, we conclude that there is a submodel $M'(s)$ of M_k with $|P'_k| \leq f_k(\psi)$ and $|P'_y| \leq f_{k,y}(\psi)$ and $M'(s), s \models \psi$.

$(<=)$ The proof is by induction on the length of ψ and can be done in the same way as in [26]. □

Theorem 7.3. Let M be a discretised model, and ψ an ECTL$^y_{-G}$ formula. Then, $M \models \psi$ iff there exists $k \in \mathbb{N}_+$ such that $[M^{\psi,s^0}]_k \wedge [\psi]_k^{[0,0]}$ is satisfiable.

Proof: Follows from Theorem 7.2 and Lemma 7.5. □

8. Experimental results

The BMC algorithm presented above has been implemented in the programming language C++, and preliminary experiments have been performed. We have done this on the computer equipped with the processor AMD Athlon XP 1800 (1544 MHz), 768 MB main memory, and the operating system Linux.

As a real time system to be model checked we have taken a modified *Fischer mutual-exclusion protocol* (MUTEX) [37]. It is modelled by a network of n diagonal timed automata, each one modelling

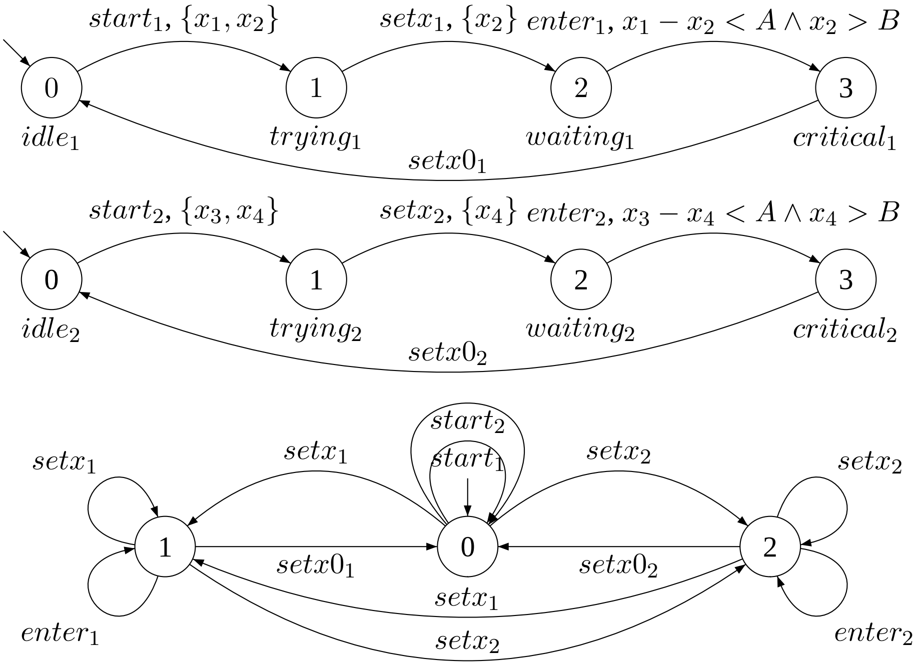

Figure 2. A network of timed automata for a modified Fischer mutual exclusion protocol for $n = 2$

a process, together with one diagonal timed automaton modelling the global variable that is used to coordinate the processes' access to the protocol; the network is shown in Figure 2. The global system is obtained as the parallel composition of the components; note that this protocol behaves in the same way as the Fischer mutual-exclusion protocol described in [31].

We have checked the above protocol for fulfilment the following property: "*it is possible that after A time units Process 1 will get its critical section as the first one*". It can be expressed by the following TECTL_{-G} formula φ:

$$\mathrm{E}((\bigwedge_{1<i\leq n} \neg c_i) U_{[A,\infty)} c_1)$$

where c_i and c_j, for $i, j \in \{1, \ldots, n\}$, are propositional variables encoding the fact that the process i (respectively j) is in its "critical section".

Table 2 presents experimental results for φ on one symbolic $11-$path, and Table 3 shows $11-$path that is a witness for φ and it was automatically generated by our tool.

9. The BMC technique for a subclass of TECTL_{-G} formulae

In Section 4 we have defined the region graph as a model that is much coarser than the dense one. Then, in Section 5, we have proved that the validity problem of a TCTL_{-G} formula φ over the model for $\mathcal{A}$ is equivalent to the validity of the corresponding CTL^{y}_{-G} formula $\mathrm{cr}(\varphi)$ over the region graph for $\mathcal{A}_\varphi$. Next, in Section 7 we have defined the BMC technique for the existential fragment of TCTL_{-G} and

NoP	BMC				**MiniSat**	
	variables	**clauses**	**sec**	**MB**	**sec**	**MB**
2	27454	82080	1.4	5.8	4.3	9.2
10	117142	353032	6.9	25.1	3.6	21.0
20	240614	725760	16.2	53.7	36.5	40.8
50	670168	2021414	56.3	95.7	41.6	96.6
100	1584196	4775170	171.0	207.9	80.8	238.4
150	2746492	8273730	334.4	182.6	123.1	410.7
200	4154717	12510105	579.0	684.1	213.9	614.5

Table 2. The results for property φ when $A = 2$ and $B = 1$.

depth	**locations**			**clock valuations**				
	$P1$	$P2$	Var	x_1	x_2	x_3	x_4	y
0	0	0	0	$0\frac{0}{8192}$	$0\frac{0}{8192}$	$0\frac{0}{8192}$	$0\frac{0}{8192}$	$0\frac{0}{8192}$
1	0	0	0	$0\frac{4472}{8192}$	$0\frac{4472}{8192}$	$0\frac{4472}{8192}$	$0\frac{4472}{8192}$	$0\frac{4472}{8192}$
2	0	0	0	$1\frac{0}{8192}$	$1\frac{0}{8192}$	$1\frac{0}{8192}$	$1\frac{0}{8192}$	$1\frac{0}{8192}$
3	1	0	0	$0\frac{0}{8192}$	$0\frac{0}{8192}$	$1\frac{0}{8192}$	$1\frac{0}{8192}$	$1\frac{0}{8192}$
4	1	0	0	$0\frac{8160}{8192}$	$0\frac{8160}{8192}$	$1\frac{8160}{8192}$	$1\frac{8160}{8192}$	$1\frac{8160}{8192}$
5	1	0	0	$1\frac{0}{8192}$	$1\frac{0}{8192}$	$2\frac{0}{8192}$	$2\frac{0}{8192}$	$2\frac{0}{8192}$
6	2	0	1	$1\frac{0}{8192}$	$0\frac{0}{8192}$	$2\frac{0}{8192}$	$2\frac{0}{8192}$	$2\frac{0}{8192}$
7	2	0	1	$1\frac{3829}{8192}$	$0\frac{3829}{8192}$	$2\frac{3829}{8192}$	$2\frac{3829}{8192}$	$2\frac{3829}{8192}$
8	2	0	1	$2\frac{0}{8192}$	$1\frac{0}{8192}$	$3\frac{0}{8192}$	$3\frac{0}{8192}$	$3\frac{0}{8192}$
9	2	0	1	$2\frac{8191}{8192}$	$1\frac{8191}{8192}$	$3\frac{8191}{8192}$	$3\frac{8191}{8192}$	$3\frac{8191}{8192}$
10	2	0	1	$3\frac{0}{8192}$	$2\frac{0}{8192}$	$4\frac{0}{8192}$	$4\frac{0}{8192}$	$4\frac{0}{8192}$
11	3	0	1	$3\frac{0}{8192}$	$2\frac{0}{8192}$	$4\frac{0}{8192}$	$4\frac{0}{8192}$	$4\frac{0}{8192}$

Table 3. A witness for the formula φ.

diagonal time automata. Thereby, we have provided a general formalism for automated verification of TECTL_{-G} properties of real-time systems that are modelled via diagonal timed automata. However, as each general technique, also this one can be improved by tailoring it into interesting subclasses of TECTL_{-G} properties. Here, we consider a subclass that is defined by the following grammar:

$$\psi := p \mid \neg\psi \mid \psi \vee \psi \mid \psi \wedge \psi \mid \mathrm{E}(\varphi \mathrm{U}_I \varphi), \text{ where}$$
$$\varphi := p \mid \neg p \mid \varphi \wedge \varphi \mid \varphi \vee \varphi$$

We will now show that the above subclass can be evaluated over *forward projection graph (FPG)* that differs from the region graph in the definition of time transitions only, which we now call the *forward projection transitions*. The main idea behind the forward projection transitions consists in agglomerating several timed transitions of region graph into one. Due to such an agglomeration, the length of searched counterexamples becomes considerably shorter. Moreover, a single step of the counterexample is either a time transition or an action transition only. In consequence, as the next section shows, the BMC method for the chosen subclass, which is defined over FPG, is much more efficient than the BMC method for this subclass defined over the region graph.

Let us move now to the formal background of the BMC method for the chosen subclass of TECTL_{-G}. We begin with a definition of a forward projection graph; the satisfaction relation for ECTL^{y}_{-G} formulae over the forward projection graph is defined in the same way as over the region graph.

Definition 9.1. (Forward Projection Graph)
A forward projection graph for the timed automaton $\mathcal{A} = (\Sigma, L, l^0, X, \mathcal{I}, R, \mathcal{V})$ is a tuple $M_{fp} = (\Sigma \cup \{\tau^*\}, Q_{fp}, q^0_{fp}, \widetilde{\mathcal{V}}_{fp}, \rightarrow_{fp})$, where $Q_{fp} = L \times Z(|X|)$ is a set of regions, $q^0_{fp} = (l^0, \{v^0\})$ is an initial region, $\widetilde{\mathcal{V}}_{fp} : S \rightarrow 2^{\mathcal{PV}}$ is a valuation function defined by $\widetilde{\mathcal{V}}_{fp}((l,Z)) = \mathcal{V}(l)$, and $\rightarrow_{fp} \subseteq S \times (\Sigma \cup \{\tau^*\}) \times S$ is defined by:

- Time transition: $(l,Z) \xrightarrow{\tau^*}_{fp} (l,Z')$ iff there exist $v \in Z$ and $v' \in Z'$ such that $(l,v) \xrightarrow{\delta} (l,v')$ for some $\delta \in \mathbb{R}_+$, and
- Action transition: For any $\sigma \in \Sigma$, $(l,Z) \xrightarrow{\sigma}_{fp} (l',Z')$ iff there exists Z'' such that $(l,Z) \xrightarrow{\tau^*}_{fp} (l,Z'')$ and there exist $v \in Z''$ and $v' \in Z'$ such that $(l,v) \xrightarrow{\sigma} (l',v')$.

A consequence of the above definition is the following:

Lemma 9.1. Let $\mathcal{A}$ be a timed automaton, $\mathcal{G}(\mathcal{A})$ a dense model for $\mathcal{A}$, φ belongs to the considered subclass of TECTL_{-G} formulae, and M_{fp} the forward projection graph for $\mathcal{A}_\varphi$. Further, let $(l,v)\lfloor X \overset{def}{=} (l, v\lfloor X)$, and for any state $(l,v)\lfloor X \in Q$, $[(l,v)]$ denote the equivalence class of (l,v) induced by the relation $\cong$. Then, $\mathcal{G}(\mathcal{A}), (l,v)\lfloor X \models \varphi$ iff $M_{fp}, [(l,v)] \models \mathrm{cr}(\varphi)$.

Proof: The proof follows directly from Lemma 5.1 and the definition of the valuation function $\widetilde{\mathcal{V}}_{fp}$. □

Now we will show that both the forward projection graph and the discretised model with the time transitions replaced by the forward projection transitions preserve validity of the same set of ECTL^{y}_{-G} formulae. We start by giving a definition of a discretised model with forward projection; the satisfaction relation for ECTL^{y}_{-G} formulae over the discretised model with forward projection is defined in the same way as over the discretised model.

Definition 9.2. (Discretised model with forward projection)
Let $\mathcal{A} = (\Sigma, L, l^0, X, \mathcal{I}, R, \mathcal{V})$ be a timed automaton. A *discretised model with forward projection* for $\mathcal{A}$ is the tuple $M = (\Sigma \cup E, S, s^0, \widetilde{\mathcal{V}}_d, \rightarrow_d)$, where $S = L \times D^X$ is a set of states, $s^0 = (l^0, v^0)$ is the initial state, $\widetilde{\mathcal{V}}_d : S \mapsto 2^{\mathcal{PV}}$ is a valuation function such that $\widetilde{\mathcal{V}}_d((l,v)) = \mathcal{V}(l)$, and the transition relation $\rightarrow_d$ is defined as follows:

- Forward projection transitions: for any $\delta \in E$, $(l,v) \xrightarrow{\delta}_d (l, v{+}\delta)$ iff $v \in [\![\mathcal{I}(l)]\!]$ and $v{+}\delta \in [\![\mathcal{I}(l)]\!]$.

- Action transitions: for any $\sigma \in \Sigma$, $(l, v) \xrightarrow{\sigma}_d (l', v')$ iff $(\exists v'')(\exists \delta \in E)$ such that $(l, v) \xrightarrow{\delta}_d (l, v'')$ and $(l, v'') \xrightarrow{\sigma} (l', v')$ in $\mathcal{G}(\mathcal{A})$

As a result of the above definition we get the following lemma.

Lemma 9.2. Let $\mathcal{A}$ be a timed automaton, φ belongs to the considered subclass of TECTL_{-G} formulae, M_{fp} a forward projection graph for $\mathcal{A}_\varphi$, and M a discretised model with forward projection for $\mathcal{A}_\varphi$. Moreover, for any state $q \in S$, let $[q]$ denote the equivalence class of q induced by the relation $\cong$. Then, for any $q \in S$, $M, q \models \mathrm{cr}(\varphi)$ if, and only if $M_{fp}, [q] \models \mathrm{cr}(\varphi)$.

Proof: It is easy to see that the thesis holds for all the propositional variables and for all the negations of propositional variables. If φ is of the form $\alpha \vee \beta$ or $\alpha \wedge \beta$, then the proof of the thesis is straightforward. So, it remains to prove that the thesis holds for formulae of the form $\mathrm{E}(\alpha \mathrm{U}_I \beta)$ and $\mathrm{A}(\alpha \mathrm{U}_I \beta)$.

Consider a formula of the form $\mathrm{E}(\alpha \mathrm{U}_I \beta)$ and suppose that $M, q \models \mathrm{cr}(\mathrm{E}(\alpha \mathrm{U}_I \beta))$. By definitions of cr and the satisfaction relation for the discretised model we have

$$(\exists q' \in S)(q \rightarrow_y q' \text{ and } (\exists \pi \in \Pi(q'))(\exists m \geq 0) \tag{22}$$
$$[M, \pi(m) \models (\mathrm{cr}(\beta) \wedge p_{y \in I_i} \wedge (p_b \vee \mathrm{cr}(\alpha))) \text{ and } (\forall j < m)\ M, \pi(j) \models \mathrm{cr}(\alpha)])$$

Consider now the following path π_{rg}:

$$\pi_{rg} = [\pi(0)], [\pi(1)], [\pi(2)], \ldots \tag{23}$$

It is obvious that π_{rg} is a valid path in M_{fp}. Moreover, by Lemma 6.7 we have that

$$M_{fp}, [\pi(m)] \models (\mathrm{cr}(\beta) \wedge p_{y \in I_i} \wedge (p_b \vee \mathrm{cr}(\alpha))) \text{ and } (\forall j < m)\ M_{fp}, [\pi(j)] \models \mathrm{cr}(\alpha) \tag{24}$$

Since $q \rightarrow_y q'$, by definitions of $\rightarrow_d$ and $\rightarrow_{fp}$ we have that $[q] \rightarrow_y [q']$. Therefore, since (24) holds, it follows that $M_{fp}, [q] \models \mathrm{cr}(\mathrm{E}(\alpha \mathrm{U}_I \beta))$.

Conversely, suppose that $M_{fp}, [q] \models \mathrm{cr}(\mathrm{E}(\alpha \mathrm{U}_I \beta))$. By definitions of cr and the satisfaction relation over the region graph we have

$$(\exists [q'] \in Q_{rg})([q] \rightarrow_y [q'] \text{ and } (\exists \pi \in \Pi([q']))(\exists m \geq 0) \tag{25}$$
$$[M_{fp}, \pi(m) \models (\mathrm{cr}(\beta) \wedge p_{y \in I_i} \wedge (p_b \vee \mathrm{cr}(\alpha))) \text{ and } (\forall j < m)\ M_{fp}, \pi(j) \models \mathrm{cr}(\alpha)])$$

Observe that π is of the following form:

$$\pi = [(l_0, v_0)], [(l_0, v_0) + \delta_0], [(l_1, v_1)], [(l_1, v_1) + \delta_1], [(l_2, v_2)] \ldots$$

with $(l_0, v_0) = q'$, $\delta_i \in E$.

Next, take the following path $\pi_d = (l_0, v_0), (l_0, v^0) + \delta_0, (l_1, v_1), (l_1, v_1) + \delta_1, (l_2, v_2) \ldots$. Since (25) holds, by the construction of π_d and Lemma 6.7 we have that

$$M, \pi_d(m) \models (\mathrm{cr}(\beta) \wedge p_{y \in I_i} \wedge (p_b \vee \mathrm{cr}(\alpha))) \text{ and } (\forall j < m)\ M, \pi_d(j) \models \mathrm{cr}(\alpha) \tag{26}$$

Since $[q] \rightarrow_y [q']$ holds, by the definition of $\rightarrow_y$, we have that $q \rightarrow_y q'$. Therefore, since (26) holds, it follows that $M, q \models \mathrm{cr}(\mathrm{E}(\alpha \mathrm{U}_I \beta))$.

The proof of the case $\mathrm{A}(\alpha \mathrm{U}_I \beta)$ is analogous. □

Given the above we can now apply the BMC technique, described in Section 7, to the considered subclass of TECTL_{-G} formulae. As a model to be encoded we take the discretised model with forward projection.

9.1. Experimental results

Consider the MUTEX protocol from Section 8 and the property $\varphi = \mathrm{E}((\bigwedge_{1<i\leq n} \neg c_i) U_{[A,\infty)} c_1)$ again. Table 4 shows experimental results for φ that has been generated via BMC technique defined over the forward projection graph. In this settings we only need one symbolic $6-$path to prove that φ is true in the model of MUTEX; while the region graph is the base for the BMC technique, we need $12-$path to prove the property.

Now consider the following property: "*if* $\mathrm{A} \leq B$*, then the mutual exclusion property is violated*". This property can be expressed by the following TECTL_{-G} formula:

$$\psi = \mathrm{EF}_{[A,\infty]}(\bigvee_{1\leq i\neq j\leq n} (c_i \wedge c_j))$$

where the propositions c_i and c_j state that the process i (respectively j) is in its "critical section"; here, we would like to emphasise that we could not check this property via the BMC technique when the region graph was considered as a basic model.

It is easy to see that for each $k > 0$ the value of the function f_k for the formula ψ is equal to 1. It follows that the counterexample, if exists, can be found on one symbolic $k-$path. The corresponding experimental results are shown in Table 4 and Table 5. In particular, Table 5 contains results for the "violated" mutual exclusion property on one symbolic $12-$path, and Table 6 shows a $12-$path that is a witness for the property ψ, and it was automatically generated by our tool.

	BMC				**MiniSat**	
NoP	**variables**	**clauses**	**sec**	**MB**	**sec**	**MB**
2	4981	13834	0.2	1.0	0.0	4.3
10	20386	57580	1.1	4.2	0.2	6.5
50	125724	361312	9.6	26.5	1.7	21.1
100	318148	923263	29.2	69.9	2.1	49.6
200	905481	2654641	112.7	152.3	17.0	130.0
400	2890132	8547373	430.6	257.2	25.8	408.5
500	4286682	12706423	695.0	842.7	42.3	608.7

Table 4. The results for property φ when $A = 2$ and $B = 1$.

	BMC				**MiniSat**	
NoP	**variables**	**clauses**	**sec**	**MB**	**sec**	**MB**
2	25430	69288	0.8	5.0	0.3	7.1
10	110860	308224	3.9	22.5	9.2	20.4
20	229607	642912	8.9	47.8	71.2	49.5
50	665698	1886712	33.6	141.4	2070.9	357.0
100	1660717	4764376	101.7	362.7	1820.7	511.9

Table 5. Mutual exclusion violated. $k = 12$, $A = 3$ and $B = 2$.

depth	locations			clock valuations				
	$P1$	$P2$	Var	x_1	x_2	x_3	x_4	y
0	0	0	0	$0\frac{0}{256}$	$0\frac{0}{256}$	$0\frac{0}{256}$	$0\frac{0}{256}$	$0\frac{0}{256}$
1	0	0	0	$4\frac{0}{256}$	$4\frac{0}{256}$	$4\frac{0}{256}$	$4\frac{0}{256}$	$4\frac{0}{256}$
2	1	0	0	$0\frac{0}{256}$	$0\frac{0}{256}$	$4\frac{0}{256}$	$4\frac{0}{256}$	$4\frac{0}{256}$
3	1	0	0	$2\frac{1}{256}$	$2\frac{1}{256}$	$6\frac{1}{256}$	$6\frac{1}{256}$	$6\frac{1}{256}$
4	1	1	0	$2\frac{2}{256}$	$2\frac{2}{256}$	$0\frac{0}{256}$	$0\frac{0}{256}$	$6\frac{2}{256}$
5	1	1	0	$2\frac{7}{256}$	$2\frac{7}{256}$	$0\frac{5}{256}$	$0\frac{5}{256}$	$6\frac{7}{256}$
6	2	1	1	$2\frac{14}{256}$	$0\frac{0}{256}$	$0\frac{10}{256}$	$0\frac{10}{256}$	$6\frac{14}{256}$
7	2	1	1	$4\frac{15}{256}$	$2\frac{1}{256}$	$2\frac{11}{256}$	$2\frac{11}{256}$	$8\frac{15}{256}$
8	3	1	1	$4\frac{30}{256}$	$2\frac{2}{256}$	$2\frac{22}{256}$	$2\frac{22}{256}$	$8\frac{30}{256}$
9	3	1	1	$4\frac{63}{256}$	$2\frac{35}{256}$	$2\frac{55}{256}$	$2\frac{55}{256}$	$8\frac{63}{256}$
10	3	2	2	$4\frac{126}{256}$	$2\frac{70}{256}$	$2\frac{110}{256}$	$0\frac{0}{256}$	$8\frac{126}{256}$
11	3	2	2	$8\frac{126}{256}$	$6\frac{70}{256}$	$6\frac{110}{256}$	$4\frac{0}{256}$	$12\frac{126}{256}$
12	3	3	2	$8\frac{252}{256}$	$6\frac{140}{256}$	$6\frac{220}{256}$	$4\frac{0}{256}$	$12\frac{252}{256}$

Table 6. Mutual exclusion violated - a witness.

10. Final Remarks

Our paper extends and improves the results of [27] and [24], where a general BMC approach for the existential fragment of TCTL and diagonal-free automata was described. The idea of BMC is taken from the paper [5]. Timed Automata have been defined and investigated in many papers [1, 31], but we adapt the definition given in [31]. Model checking for TCTL_{-G} was considered by several authors using different approaches: over clock region models [1], on-the-fly [6], space-efficient [16], over minimal models [31, 12], and using SAT-methods [27, 29]. Our approach is closely related to [2] and [31], from which we draw the idea of translating of the model checking problem for TCTL_{-G} to the model checking problem for CTL^{y}_{-G}.

The paper presents preliminary experimental results only, but they show that the proposed verification method is quite efficient and worth exploring. Since the literature for the formal verification of diagonal timed automata does not provide any other TCTL_{-G} model checking method that works on the fragments of models under consideration, we cannot compare our results with others.

References

[1] Alur, R., Courcoubetis, C., Dill, D.: Model Checking in Dense Real-Time, *Information and Computation*, **104**(1), 1993, 2–34.

[2] Alur, R., Dill, D.: A Theory of Timed Automata, *Theoretical Computer Science*, **126**(2), 1994, 183–235.

[3] Alur, R., Feder, T., Henzinger, T.: The Benefits of Relaxing Punctuality, *Journal of the ACM*, **43**(1), 1996, 116–146.

[4] Alur, R., Madhusudan, P.: Decision Problems for Timed Automata: A Survey, *Formal Methods for the Design of Real-Time Systems*, 3185, Springer Berlin / Heidelberg, 2004, ISBN 978-3-540-23068-7.

[5] Biere, A., Cimatti, A., Clarke, E., Fujita, M., Zhu, Y.: Symbolic Model Checking Using SAT Procedures Instead of BDDs, *Proceedings of the ACM/IEEE Design Automation Conference (DAC'99)*, 1999.

[6] Bouajjani, A., Tripakis, S., Yovine, S.: On-the-Fly Symbolic Model Checking for Real-Time Systems, *Proceedings of the 18th IEEE Real-Time Systems Symposium (RTSS'97)*, IEEE Computer Society, 1997.

[7] Bryant, R.: Graph-Based Algorithms for Boolean Function Manipulation, *IEEE Transaction on Computers*, **35**(8), 1986, 677–691.

[8] Clarke, E. M., Grumberg, O., Peled, D. A.: *Model Checking*, The MIT Press, Cambridge, Massachusetts, 1999, ISBN 0-262-03270-8.

[9] Dams, D., Gerth, R., Knaack, B., Kuiper, R.: Partial-Order Reduction Techniques for Real-Time Model Checking, *Proceedings of the 3rd International Workshop on Formal Methods for Industrial Critical Systems*, 1998.

[10] Dams, D., Grumberg, O., Gerth, R.: Abstract Interpretation of Reactive Systems: Abstractions Preserving ACTL*, ECTL* and CTL*, *Proceedings of the IFIP Working Conference on Programming Concepts, Methods and Calculi (PROCOMET'94)*, Elsevier Science Publishers, 1994.

[11] Daws, C., Tripakis, S.: Model Checking of Real-Time Reachability Properties Using Abstractions, *Proceedings of the 4th International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS'98)*, 1384, Springer-Verlag, 1998.

[12] Dembiński, P., Penczek, W., Pólrola, A.: Automated Verification of Infinite State Concurrent Systems: an Improvement in Model Generation, *Proceedings of the 4th International Conference on Parallel Processing and Applied Mathematics (PPAM'01)*, 2328, Springer-Verlag, 2002.

[13] Emerson, E. A.: Temporal and Modal Logic, in: *Handbook of Theoretical Computer Science* (J. van Leeuwen, Ed.), Elsevier Science Publishers, 1990, 996–1071.

[14] Emerson, E. A., Sistla, A. P.: Symmetry and Model Checking, *Formal Methods in System Design*, **9**, 1995, 105–131.

[15] Fersman, E., Pettersson, P., Yi, W.: Timed automata with asynchronous processes: Schedulability and decidability, *Proceedings of the 8th International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS'02)*, 2280, Springer-Verlag, 2002.

[16] Kupferman, O., Henzinger, T. A., Vardi, M. Y.: A Space-Efficient On-the-fly Algorithm for Real-Time Model Checking, *Proceedings of the 7th International Conference on Concurrency Theory (CONCUR'96)*, 1119, Springer-Verlag, 1996.

[17] Lilius, J.: Efficient State Space Search for Time Petri Nets, *Proceedings of MFCS Workshop on Concurrency, Brno'98*, 18, Elsevier Science Publishers, 1999.

[18] McMillan, K. L.: *Symbolic Model Checking*, Kluwer Academic Publishers, 1993, ISBN 0-7923-9380-5.

[19] McMillan, K. L.: Applying SAT Methods in Unbounded Symbolic Model Checking, *Proc. of the 14th Int. Conf. on Computer Aided Verification (CAV'02)*, 2404, Springer-Verlag, 2002.

[20] Merlin, P., Farber, D. J.: Recoverability of Communication Protocols – Implication of a Theoretical Study, *IEEE Trans. on Communications*, **24**(9), 1976, 1036–1043.

[21] de Moura, L., Rueß, H., Sorea, M.: Lazy Theorem Proving for Bounded Model Checking over Infinite Domains, *Proceedings of the 18th International Conference on Automated Deduction (CADE-18)*, 2392, Springer-Verlag, 2002.

[22] Pagani, F.: Partial Orders and Verification of Real-Time Systems, *Proceedings of the 4th International Symposium on Formal Techniques in Real Time and Fault Tolerant Systems (FTRTFT'96)*, 1135, Springer-Verlag, 1996.

[23] Peled, D.: Partial Order Reduction: Linear and Branching Temporal Logics and Process Algebras, *Proceedings of Partial Order Methods in Verification (POMIV'96)*, 29, Amer. Math. Soc., 1996.

[24] Penczek, W., Pólrola, A.: Specification and Model Checking of Temporal Properties in Time Petri Nets and Timed Automata, *Proceedings of the 25th International Conference on Applications and Theory of Petri Nets (ATPN'04)*, 3099, Springer-Verlag, 2004.

[25] Penczek, W., Pólrola, A., Woźna, B., Zbrzezny, A.: Bounded Model Checking for Reachability Testing in Time Petri Nets, *Proceedings of the International Workshop on Concurrency, Specification and Programming (CS&P'04)*, 170, Humboldt University, 2004.

[26] Penczek, W., Woźna, B., Zbrzezny, A.: Bounded Model Checking for the Universal Fragment of CTL, *Fundamenta Informaticae*, **51**(1-2), 2002, 135–156.

[27] Penczek, W., Woźna, B., Zbrzezny, A.: Towards Bounded Model Checking for the Universal Fragment of TCTL, *Proceedings of the 7th International Symposium on Formal Techniques in Real-Time and Fault Tolerant Systems (FTRTFT'02)*, 2469, Springer-Verlag, 2002.

[28] R. Alur, T.A. Henzinger: Logics and Models of Real-Time: A Survey, *Real Time: Theory in Practice*, 600, Springer-Verlag, 1991.

[29] Seshia, S., Bryant, R.: Unbounded, Fully Symbolic Model Checking of Timed Automata Using Boolean Methods, *Proceedings of the 15th International Conference on Computer Aided Verification (CAV'03)*, 2725, Springer-Verlag, 2003.

[30] Sorea, M.: Bounded Model Checking for Timed Automata, *Proceedings of the 3rd Workshop on Models for Time-Critical Systems (MTCS'02)*, 68, Elsevier Science Publishers, 2002.

[31] Tripakis, S., Yovine, S.: Analysis of Timed Systems Using Time-Abstracting Bisimulations, *Formal Methods in System Design*, **18**(1), 2001, 25–68.

[32] Wolper, P., Godefroid, P.: Partial Order Methods for Temporal Verification, *Proceedings of the 4th International Conference on Concurrency Theory (CONCUR'93)*, 715, Springer-Verlag, 1993.

[33] Woźna, B., Lomuscio, A., Penczek, W.: Bounded Model Checking for knowledge over real time, *Proceedings of the 4st International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS'05)*, I, ACM Press, July 2005.

[34] Woźna, B., Zbrzezny, A.: Checking ACTL* Properties of Discrete Timed Automata via Bounded Model Checking, *Proceedings of the 1st International Workshop on Formal Analysis and Modeling of Timed Systems (FORMATS'03)*, 2791, Springer-Verlag, 2004.

[35] Woźna, B., Zbrzezny, A., Penczek, W.: Checking Reachability Properties for Timed Automata via SAT, *Fundamenta Informaticae*, **55**(2), 2003, 223–241.

[36] Zbrzezny, A.: Improvements in SAT-based Reachability Analysis for Timed Automata, *Fundamenta Informaticae*, **60**(1-4), 2004, 417–434.

[37] Zbrzezny, A.: SAT-based Reachability Checking for Timed Automata with Diagonal Constraints, *Fundamenta Informaticae*, **67**(1-3), 2005, 303–322.