On MITL and alternating timed automata over infinite words

Thomas Brihaye[1], Morgane Estiévenart[1⋆], and Gilles Geeraerts[2]

[1] UMons, Belgium, [2] U.L.B., Belgium

Abstract. *One clock alternating timed automata* (OCATA) have been introduced as natural extension of (one clock) timed automata to express the semantics of MTL [15]. In this paper, we consider the application of OCATA to the problems of model-checking and satisfiability for MITL (a syntactic fragment of MTL), interpreted over infinite words. Our approach is based on the *interval semantics* (recently introduced in [5] in the case of finite words) extended to infinite words. We propose region-based and zone-based algorithms, based on this semantics, for MITL model-checking and satisfiability. We report on the performance of a prototype tool implementing those algorithms.

1 Introduction

Model-checking [7] is today one of the most prominent and successful techniques to establish *automatically* the correctness of a computer system. The system designer provides a *model-checker* with a *model* of the system and a *formal property* that the system must respect. The model-checker either proofs that the system respects the property, or outputs an error trace that can be used for debugging. For their implementation, many model-checkers rely on the so-called *automata-based approach*, where the behaviours of the system and the set of *bad behaviours* are represented by the languages $L(B)$ and $L(A_{\neg\varphi})$ of Büchi automata B and $A_{\neg\varphi}$ respectively. Then, the model-checker performs automata-based manipulations to check whether $L(B) \cap L(A_{\neg\varphi}) = \emptyset$. While Büchi automata are adequate for modeling systems, properties are more easily expressed by means of logical sentences. The linear temporal logic (LTL for short) is arguably one of the most studied logic to express such requirements. Algorithms to turn an LTL formula φ into a Büchi automaton A_φ recognising the same language are well-known, thereby enabling its use in model-checkers.

Yet, this classical theory is not adequate for reasoning about *real-time* properties of systems, because Büchi automata and LTL can only express *sequence of events*, but have no notion of (time) *distance* between those events. Introduced by Alur and Dill in 1994 [1], *timed automata* (an extension of Büchi automata with clocks, i.e. real variables that evolve synchronously) are today the best accepted model for those real-time systems. Symmetrically, several logics have been introduced to specify real-time properties of systems. Among them, MITL (a syntactic fragment of MTL [11]) is particularly appealing, because it combines expressiveness [3] and tractability (MTL is mostly undecidable [3], while

⋆ This author has been supported by a FRIA scholarship.

model-checking and satisfiability are ExpSpace-c in MITL). A comprehensive and efficient automata-based framework to support MITL model-checking (and other problems such as satisfiability) is thus highly desirable. In a recent work [5] we made a first step towards this goal in the restricted case of *finite words semantics*. We rely on *one-clock alternating timed automata* (OCATA for short), in order to avoid the direct, yet involved, translation from MITL to timed automata first introduced in [3]. The translation from MITL to OCATA – which has been introduced by Ouaknine and Worrell in the general case of MTL [15] – is straightforward. However, the main difficulty with alternating timed automata is that they cannot, in general, be converted into an equivalent timed automaton, even in the one-clock case. Indeed, a run of an alternating automaton can be understood as several copies of the same automaton running in parallel on the same word. Unfortunately, the clock values of all the copies are not always synchronised, and one cannot bound, a priori, the number of different clock values that one must track along the run. Hence, contrary to the untimed word case, subset construction techniques cannot be directly applied to turn an OCATA into a timed automaton (with finitely many clocks).

Our solution [5] amounts to considering an alternative semantics for OCATA, that we call the *interval semantics*, where clock valuations are not punctual values but intervals with real endpoints. One of the features of this semantic is that several clock values can be grouped into intervals, thanks to a so-called *approximation function*. For instance, consider a configuration of an OCATA with three copies of the automata currently in the same location ℓ, and with clock values 0.42, 1.2 and 5.7 respectively. It can be approximated by a single interval $[0.42, 5.7]$, meaning: 'there are two copies witch clock values 0.42 and 5.7, and there are *potentially* several copies with clock values in the interval $[0.42, 5.7]$'. This technique allows to reduce the number of variables needed to track the clock values of the OCATA. While this grouping yields an under-approximation of the accepted language, we have showed [5] that, in the case of finite words, and for an OCATA A_φ obtained from an MITL formula φ, one can always define an approximation function s.t. the language of A_φ is preserved and the number of intervals along all runs is bounded by a constant $M(\varphi)$ depending on the formula. Using classical subset construction and tracking the endpoints of each interval by means of a pair of clocks, we can then translate the OCATA into a Büchi TA accepting the same language.

In the present work, we continue this line of research and demonstrate that our techniques carry on to the infinite words case. Achieving this result is not straightforward because OCATA on infinite words have not been studied as deeply as in the finite words case, probably because infinite words language emptiness of OCATA is decidable only on restricted subclasses [15,17]. Hence, to reach our goal, we make several technical contributions regarding infinite words OCATA, that might be of interest outside this work. First, in Section 3 we adapt the interval semantics of [5] to the infinite words case. Then, in Section 4, we introduce *tree-like* OCATA (TOCATA for short), a subclass of OCATA that exhibit some structure akin to a tree (in the same spirit as the Weak and

Very Weak Alternating Automata [12,?]). For every MTL formula φ, we show that the OCATA A_φ obtained by the Ouaknine and Worrell construction [15] recognises the language of φ (a property that had never been established in the case of infinite words, as far as we know[1]), is in fact a TOCATA. This shows in particular that TOCATA are semantically different from the 'weak OCATA' introduced in [17] (where 'weak' refers to weak accepting conditions), and whose emptiness problem is decidable. We prove specific properties of TOCATA that are important in our constructions (for instance, TOCATA on infinite words can be easily complemented), and we adapt the classical Miyano and Hayashi construction [14] to obtain a procedure to translate any TOCATA A_φ obtained from an MITL formula into an equivalent timed Büchi automaton $\mathcal{B}_\varphi$. Equipped with these theoretical results, we propose in Section 6 algorithms to solve the *satisfiability* and *model-checking* problems of MITL. We define region-based and zone-based [2] versions of our algorithms. Our algorithms work *on-the-fly* in the sense that they work directly on the structure of the OCATA $\mathcal{A}_\varphi$ (whose size is linear in the size of φ), and avoid building $\mathcal{B}_\varphi$ beforehand (which is, in the worst case, exponential in the size of φ). Finally in Section 6, we present prototype tools implementing those algorithms. To the best of our knowledge, these are the first tools solving those problems for the full MITL. We report on and compare their performance against a benchmark of MITL formulas whose sizes are parametrised. While still preliminary, the results are encouraging.

2 Preliminaries

Basic notions. Let $\mathbb{R}$, $\mathbb{R}^+$ and $\mathbb{N}$ denote the sets of real, non-negative real and natural numbers respectively. We call ***interval*** a convex subset of $\mathbb{R}$. We rely on the classical notation $\langle a, b\rangle$ for intervals, where $\langle$ is (or [, $\rangle$ is) or], $a \in \mathbb{R}$ and $b \in \mathbb{R} \cup \{+\infty\}$. For an interval $I = \langle a, b\rangle$, we let $\inf(I) = a$ be the *infimum* of I, $\sup(I) = b$ be its *supremum* (a and b are called the *endpoints* of I) and $|I| = \sup(I) - \inf(I)$ be its *length*. We note $\mathcal{I}(\mathbb{R})$ the set of all intervals. We note $\mathcal{I}(\mathbb{R}^+)$ (resp. $\mathcal{I}(\mathbb{N}^{+\infty})$) the set of all intervals whose endpoints are in $\mathbb{R}^+$ (resp. in $\mathbb{N} \cup \{+\infty\}$). Let $I \in \mathcal{I}(\mathbb{R})$ and $t \in \mathbb{R}$, we note $I + t$ for $\{i + t \in \mathbb{R} \mid i \in I\}$. Let I and J be two intervals, we let $I < J$ iff $\forall i \in I, \forall j \in J : i < j$.

Let Σ be a finite alphabet. An *infinite word* on a set S is an infinite sequence $s = s_1 s_2 s_3 \ldots$ of elements in S. An infinite time sequence $\bar{\tau} = \tau_1 \tau_2 \tau_3 \ldots$ is an infinite word on $\mathbb{R}^+$ s.t. $\forall i \in \mathbb{N}, \tau_i \leq \tau_{i+1}$. An *infinite timed word* over Σ is a pair $\theta = (\bar{\sigma}, \bar{\tau})$ where $\bar{\sigma}$ is an infinite word over Σ, $\bar{\tau}$ an infinite time sequence. We also note θ as $(\sigma_1, \tau_1)(\sigma_2, \tau_2)(\sigma_3, \tau_3)\ldots$. We denote by $T\Sigma^\omega$ the set of all infinite timed words. A *timed language* is a (possibly infinite) set of infinite timed words.

Metric Interval Time Logic. Given a finite alphabet Σ, the formulas of MITL are defined by the following grammar, where $\sigma \in \Sigma$, $I \in \mathcal{I}(\mathbb{N}^{+\infty})$ is non-singular:

[1] Even in [15] where the authors consider a fragment of MTL over infinite words, but consider only safety properties that are reduced to questions on finite words.

$$\varphi := \top \mid \sigma \mid \varphi_1 \wedge \varphi_2 \mid \neg\varphi \mid \varphi_1 U_I \varphi_2.$$

We rely on the following usual shortcuts $\Diamond_I \varphi$ stands for $\top U_I \varphi$, $\Box_I \varphi$ for $\neg\Diamond_I \neg\varphi$, $\varphi_1 \tilde{U}_I \varphi_2$ for $\neg(\neg\varphi_1 U_I \neg\varphi_2)$, $\Box\varphi$ for $\Box_{[0,\infty)}\varphi$ and $\Diamond\varphi$ for $\Diamond_{[0,\infty)}\varphi$.

Given an MITL formula φ, we note $Sub(\varphi)$ the set of all subformulas of φ, i.e.: $Sub(\varphi) = \{\varphi\}$ when $\varphi \in \{\top\} \cup \Sigma$, $Sub(\neg\varphi) = \{\neg\varphi\} \cup Sub(\varphi)$ and $Sub(\varphi) = \{\varphi\} \cup Sub(\varphi_1) \cup Sub(\varphi_2)$ when $\varphi = \varphi_1 U_I \varphi_2$ or $\varphi = \varphi_1 \wedge \varphi_2$. We let $|\varphi|$ denote the *size of* φ, defined as the *number* of U or $\tilde{U}$ modalities it contains.

Definition 1 (Semantics of MITL). *Given an infinite timed word $\theta = (\bar{\sigma}, \bar{\tau})$ over Σ, a position $i \in \mathbb{N}_0$ and an MITL formula φ, we say that θ satisfies φ from position i, written $(\theta, i) \models \varphi$ iff the following holds:*
$(\theta, i) \models \top$ $\quad$ $(\theta, i) \models \varphi_1 \wedge \varphi_2$ *iff* $(\theta, i) \models \varphi_1$ *and* $(\theta, i) \models \varphi_2$
$(\theta, i) \models \sigma$ *iff* $\sigma_i = \sigma$ $\quad$ $(\theta, i) \models \neg\varphi$ *iff* $(\theta, i) \not\models \varphi$
$(\theta, i) \models \varphi_1 U_I \varphi_2$ *iff* $\exists j \geq i$: $(\theta, i) \models \varphi_2$, $\tau_j - \tau_i \in I \wedge \forall i \leq k < j$: $(\theta, k) \models \varphi_1$
*We say that θ **satisfies** φ, written $\theta \models \varphi$, iff $(\theta, 1) \models \varphi$. We note $[\![\varphi]\!]$ the timed language $\{\theta \mid \theta \models \varphi\}$.*

Observe that, for every MITL formula φ, $[\![\varphi]\!]$ is a timed language and that we can transform any MITL formula in an equivalent MITL formula in *negative normal form* (in which negation can only be present on letters $\sigma \in \Sigma$) using the operators: $\wedge, \vee, \neg, U_I$ and $\tilde{U}_I$.

Example 2. We can express the fact that '*every occurrence of p is followed by an occurrence of q between 2 and 3 time units later*' by: $\Box(p \Rightarrow \Diamond_{[2,3]} q)$. Its negation, $\neg\big(\Box(p \Rightarrow \Diamond_{[2,3]} q)\big)$, is equivalent to the following negative normal form formula: $\top U_{[0,+\infty)}(p \wedge \bot \tilde{U}_{[2,3]} \neg q)$.

Alternating timed automata. One-clock alternating timed automata (OCATA for short) have been introduced by Ouaknine and Worrell to define the language of MTL formulas [16]. We will rely on OCATA to build our automata based framework for MITL. Let $\Gamma(L)$ be a set of formulas of the form $\top$, or $\bot$, or $\gamma_1 \vee \gamma_2$ or $\gamma_1 \wedge \gamma_2$ or ℓ or $x \bowtie c$ or $x.\gamma$, with $c \in \mathbb{N}$, $\bowtie \in \{<, \leq, >, \geq\}$, $\ell \in L$. We call $x \bowtie c$ a *clock constraint*. Then, a *one-clock alternating timed automaton* (OCATA) [16] is a tuple $\mathcal{A} = (\Sigma, L, \ell_0, F, \delta)$ where Σ is a finite alphabet, L is a finite set of locations, ℓ_0 is the initial location, $F \subseteq L$ is a set of accepting locations, $\delta : L \times \Sigma \to \Gamma(L)$ is the transition function. Intuitively, disjunctions in $\delta(\ell)$ model non-determinism, conjunctions model the creation of several copies of the automata running in parallel (that must all accept for the word to be accepted) and $x.\gamma$ means that x (the clock of the OCATA) is reset when taking the transition.

We assume that, for all γ_1, γ_2 in $\Gamma(L)$: $x.(\gamma_1 \vee \gamma_2) = x.\gamma_1 \vee x.\gamma_2$, $x.(\gamma_1 \wedge \gamma_2) = x.\gamma_1 \wedge x.\gamma_2$, $x.x.\gamma = x.\gamma$, $x.(x \bowtie c) = 0 \bowtie c$, $x.\top = \top$ and $x.\ \bot = \bot$. Thus, we can write any formula of $\Gamma(L)$ in disjunctive normal form, and, from now on, we assume that $\delta(\ell, \sigma)$ is written in disjunctive normal form. That is, for all ℓ, σ, we have $\delta(\ell, \sigma) = \bigvee_j \bigwedge_k A_{j,k}$, where each term $A_{j,k}$ is of the form ℓ, $x.\ell$,

$x \bowtie c$ or $0 \bowtie c$, with $\ell \in L$ and $c \in \mathbb{N}$. We call *arc* of the OCATA $\mathcal{A}$ a triple $(\ell, \sigma, \bigwedge_k A_{j,k})$ s.t. $\bigwedge_k A_{j,k}$ is a disjunct in $\delta(\ell, \sigma)$. For an arc $a = (\ell, \sigma, \bigwedge_k A_{j,k})$, we note $Start(a) = \ell$, $Dest(a) = \{\ell' \mid \ell' \text{ is a location present in } \bigwedge_k A_{j,k}\}$ and we say that a is labeled by σ.

Example 3. Fig. 1 (left) shows an OCATA $\mathcal{A}_\varphi = \big\{\Sigma, \{\ell_\Box, \ell_\Diamond\}, \ell_\Box, \{\ell_\Box\}, \delta\big\}$, over the alphabet $\Sigma = \{a, b\}$, and with transition function: $\delta(\ell_\Box, a) = \ell_\Box \wedge x.\ell_\Diamond$, $\delta(\ell_\Box, b) = \ell_\Box$, $\delta(\ell_\Diamond, a) = \ell_\Diamond$ and $\delta(\ell_\Diamond, b) = \ell_\Diamond \vee (x \geq 1 \wedge x \leq 2)$. We depict a conjunctive transition such as $\delta(\ell_\Box, a) = \ell_\Box \wedge x.\ell_\Diamond$ by an arrow splitting in two branches connected to $\ell_\Box$ and $\ell_\Diamond$ (they might have different resets: the reset of clock x is depicted by $x := 0$). Intuitively, when reading an a from $\ell_\Box$ with clock value v, the automaton starts *two copies of itself*, the former in location $\ell_\Box$, with clock value v, the latter in location $\ell_\Diamond$ with clock value 0. Both copies should accept the suffix for the word to be accepted. The edge labeled by $b, x \in [1, 2]$ from $\ell_\Diamond$ has no target location: it depicts the fact that, when the automaton has a copy in location $\ell_\Diamond$ with a clock valuation in $[1, 2]$, the copy accepts all further suffixes and can thus be *removed* from the automaton.

3 The intervals semantics for OCATA on infinite words

In this section, we adapt to infinite timed words the intervals semantics introduced in [5]. In this semantics, *configurations* are sets of *states* (ℓ, I), where ℓ is a location of the OCATA and I is an *interval* (while in the standard semantics states are pairs (ℓ, v), where v is the valuation of the clock). Intuitively, a state (ℓ, I) is an abstraction of a set of states of the form (ℓ, v) with $v \in I$.

Formally, a *state* of an OCATA $\mathcal{A} = (\Sigma, L, \ell_0, F, \delta)$ is a pair (ℓ, I) where $\ell \in L$ and $I \in \mathcal{I}(\mathbb{R}^+)$. We note $S = L \times \mathcal{I}(\mathbb{R}^+)$ the state space of $\mathcal{A}$. When $I = [v, v]$ (sometimes denoted $I = \{v\}$), we shorten (ℓ, I) by (ℓ, v). A *configuration* of an OCATA $\mathcal{A}$ is a (possibly empty) finite set of states of $\mathcal{A}$ in which all intervals associated with a same location are disjoint. In the rest of the paper, we sometimes see a configuration C as a function from L to $2^{\mathcal{I}(\mathbb{R}^+)}$ s.t. for all $\ell \in L$: $C(\ell) = \{I \mid (\ell, I) \in C\}$. We note $\text{Config}(\mathcal{A})$ the set of all configurations of $\mathcal{A}$. The *initial configuration* of $\mathcal{A}$ is $\{(\ell_0, 0)\}$. For a configuration C and a delay $t \in \mathbb{R}^+$, we note $C + t$ the configuration $\{(\ell, I + t) \mid (\ell, I) \in C\}$. From now on, we assume that, for all configurations C and all locations ℓ: when writing $C(\ell)$ as $\{I_1, \ldots, I_m\}$ we have $I_i < I_{i+1}$ for all $1 \leq i < m$. Let E be a finite set of intervals from $\mathcal{I}(\mathbb{R}^+)$. We let $\|E\| = |\{[a, a] \in E\}| + 2 \times |\{I \in E \mid \inf(I) \neq \sup(I)\}|$ denote the number of *individual clocks* we need to encode all the information present in E, using one clock to track singular intervals, and two clocks to retain $\inf(I)$ and $\sup(I)$ respectively for non-singular intervals I. For a configuration C, we let $\|C\| = \sum_{\ell \in L} \|C(\ell)\|$.

Interval semantics [5]. Let $M \in \text{Config}(\mathcal{A})$ be a configuration of an OCATA $\mathcal{A}$, and $I \in \mathcal{I}(\mathbb{R}^+)$. We define the satisfaction relation "$\models_I$" on $\Gamma(L)$ as:

$$
\begin{array}{ll}
M \models_I \top & M \models_I \ell \quad \text{iff } (\ell, I) \in M \\
M \models_I \gamma_1 \wedge \gamma_2 \ \text{ iff } M \models_I \gamma_1 \text{ and } M \models_I \gamma_2 & M \models_I x \bowtie c \ \text{ iff } \forall x \in I, x \bowtie c \\
M \models_I \gamma_1 \vee \gamma_2 \ \text{ iff } M \models_I \gamma_1 \text{ or } M \models_I \gamma_2 & M \models_I x.\gamma \quad \text{iff } M \models_{[0,0]} \gamma
\end{array}
$$

We say that a configuration M is a *minimal model* of the formula $\gamma \in \Gamma(L)$ wrt the interval $I \in \mathcal{I}(\mathbb{R}^+)$ iff $M \models_I \gamma$ and there is no $M' \subsetneq M$ such that $M' \models_I \gamma$. Intuitively, for $\ell \in L, \sigma \in \Sigma$ and $I \in \mathcal{I}(\mathbb{R}^+)$, a minimal model of $\delta(\ell, \sigma)$ wrt I represents a configuration the automaton can reach from state (ℓ, I) by reading σ. Observe that the definition of $M \models_I x \bowtie c$ only allows to take a transition $\delta(\ell, \sigma)$ from state (ℓ, I) if all the values in I satisfy the clock constraint $x \bowtie c$ of $\delta(\ell, \sigma)$. We denote $\mathsf{Succ}((\ell, I), \sigma) = \{M \mid M \text{ is a minimal model of } \delta(\ell, \sigma) \text{ wrt } I\}$. We lift the definition of Succ to configurations C as follows: $\mathsf{Succ}(C, \sigma)$ is the set of all configurations C' of the form $\cup_{s \in C} M_s$, where, for all $s \in C$: $M_s \in \mathsf{Succ}(s, \sigma)$. That is, each $C' \in \mathsf{Succ}(C, \sigma)$ is obtained by chosing one minimal model M_s in $\mathsf{Succ}(s, \sigma)$ for each $s \in C$, and taking the union of all those M_s.

Example 4. Let us consider again the OCATA of Fig. 1 (left), and let us compute the minimal models of $\delta(\ell_\Diamond, b) = \ell_\Diamond \vee (x \geq 1 \wedge x \leq 2)$ wrt to $[1.5, 2]$. A minimal model of $\ell_\Diamond$ wrt $[1.5, 2]$ is $M_1 = \{(\ell_\Diamond, [1.5, 2])\}$. A minimal model of $(x \geq 1 \wedge x \leq 2)$ is $M_2 = \emptyset$ since all values in $[1.5, 2]$ satisfy $(x \geq 1 \wedge x \leq 2)$. As $M_2 \subseteq M_1$, M_2 is the unique minimal model of $\delta(\ell_\Diamond, b)$ wrt $[1.5, 2]$: $\mathsf{Succ}((\ell_\Diamond, [1.5, 2]), b) = \{M_2\}$.

Approximation functions Let us now recall the notion of *approximation functions* that associate with each configuration C, a set of configurations that *approximates C and contains less states* than C. Formally, for an OCATA $\mathcal{A}$, an *approximation function* is a function $f : \text{Config}(\mathcal{A}) \mapsto 2^{\text{Config}(\mathcal{A})}$ s.t. for all configurations C, for all $C' \in f(C)$, for all locations $\ell \in L$: (*i*) (i) $\|C'(\ell)\| \leq \|C(\ell)\|$; (ii) for all $I \in C(\ell)$, there exists $J \in C'(\ell)$ s.t. $I \subseteq J$; and (iii) for all $J \in C'(\ell)$, there are $I_1, I_2 \in C(\ell)$ s.t. $\inf(J) = \inf(I_1)$ and $\sup(J) = \sup(I_2)$. We note $APP_\mathcal{A}$ the set of approximation functions for $\mathcal{A}$. We lift all approximation functions f to sets $\mathcal{C}$ of configurations in the usual way: $f(\mathcal{C}) = \cup_{C \in \mathcal{C}} f(C)$. In the rest of the paper we will rely mainly on approximation functions that enable to *bound* the number of clock copies in all configurations along all runs of an OCATA $\mathcal{A}$. Let $k \in \mathbb{N}$, we say that $f \in APP_\mathcal{A}$ is a *k-bounded approximation function* iff for all $C \in \text{Config}(\mathcal{A})$, for all $C' \in f(C)$: $\|C'\| \leq k$.

f-Runs of OCATA We can now define formally the notion of *run* of an OCATA in the interval semantics. This notion will be parametrised by an approximation function f, that will be used to reduce the number of states present in each configuration along the run. Each new configuration in the run is thus obtained in three steps: letting time elapse, performing a discrete step, and applying the approximation function. Formally, let $\mathcal{A}$ be an OCATA of state space S, $f \in APP_\mathcal{A}$ be an approximation function and $\theta = (\sigma_1, \tau_1)(\sigma_2, \tau_2) \ldots (\sigma_i, \tau_i) \ldots$ be an infinite timed word. Let us note $t_i = \tau_i - \tau_{i-1}$ for all $i \geq 1$, assuming $\tau_0 = 0$. An

f-run of $\mathcal{A}$ on θ is an infinite sequence $C_0, C_1, \ldots, C_i, \ldots$ of configurations s.t.: $C_0 = \{(\ell_0, 0)\}$ and for all $i \geq 1$: $C_i \in f(\mathsf{Succ}(C_{i-1} + t_i, \sigma_i))$. Observe that for all pairs of configurations C, C' s.t. $C' \in f(\mathsf{Succ}(C + t, \sigma))$ for some f, t and σ, each $s \in C$ can be associated with a unique set $\mathsf{dest}(C, C', s) \subseteq C'$ containing all the 'successors' of s in C' and obtained as follows. Let $\overline{C} \in \mathsf{Succ}(C + t, \sigma)$ be s.t. $C' \in f(\overline{C})$. Thus, by definition, $\overline{C} = \cup_{\overline{s} \in C} M_{\overline{s}}$, where each $M_{\overline{s}} \in \mathsf{Succ}(\overline{s}, \sigma)$ is the minimal model that has been chosen for $\overline{s}$ when computing $\mathsf{Succ}(C, \sigma)$. Then, $\mathsf{dest}(C, C', s) = \{(\ell', J) \in C' \mid (\ell', I) \in M_s \text{ and } I \subseteq J\}$. Remark that $\mathsf{dest}(C, C', s)$ is well-defined because intervals are assumed to be disjoint in configurations. The function dest allows to define a DAG representation of runs, as is usual with alternating automata. We regard a run $\pi = C_0, C_1, \ldots, C_i, \ldots$ as a rooted DAG $G_\pi = (V, \rightarrow)$, whose vertices V correspond to the states of the OCATA (vertices at depth i correspond to C_i), and whose set of edges $\rightarrow$ expresses the OCATA transitions. Formally, $V = \cup_{i \geq 0} V_i$, where for all $i \geq 0$: $V_i = \{(s, i) \mid s \in C_i\}$ is the set of all vertices of depth i. The root of G_π is $((\ell_0, 0), 0)$. Finally, $(s_1, i_1) \rightarrow (s_2, i_2)$ iff $i_2 = i_1 + 1$ and $s_2 \in \mathsf{dest}(C_{i-1}, C_i, s_1)$. From now on, we will mainly rely on the DAG characterisation of f-runs.

Example 5. Fig. 2 displays three DAG representation of run prefixes of $\mathcal{A}_\varphi$ (Fig. 1), on the word $(a, 0.1)(a, 0.2)(a, 1.9)(b, 2)(b, 3)\ldots$ (grey boxes highlight the successive configurations). π only is an Id-run and shows why the number of clock copies cannot be bounded in general: if $\mathcal{A}_\varphi$ reads n a's between instants 0 and 1, n copies of the clock are created in location $\ell_\Diamond$.

Please find below an equivalent definition of an f-run in which the time elapsing and discrete transition (i.e. the acting of reading a letter) are distinguished. The definition below is inspired from the classical definition of run of an OCATA [16] and is more convenient to manipulate in the proves of our results.

Definition 6. *Let $\mathcal{A}$ be an OCATA and let $f \in APP_\mathcal{A}$ be an approximation function. The f-*semantics of $\mathcal{A}$ *is the transition system $\mathcal{T}_{\mathcal{A},f} = (\mathrm{Config}(\mathcal{A}), \rightsquigarrow, \longrightarrow_f)$ on configurations of $\mathcal{A}$ defined as follows:*

- *the transition relation $\rightsquigarrow$ takes care of the elapsing of time: $\forall t \in \mathbb{R}^+, C \overset{t}{\rightsquigarrow} C'$ iff $C' = C + t$. We let $\rightsquigarrow = \bigcup_{t \in \mathbb{R}^+} \overset{t}{\rightsquigarrow}$.*
- *the transition relation $\longrightarrow$ takes care of discrete transitions between locations and of the approximation: $C \xrightarrow{\sigma} C'$ iff $C' \in f(Succ(C, \sigma))$. We let $\longrightarrow_f = \bigcup_{\sigma \in \Sigma} \xrightarrow{\sigma}_f$.*

Let $\theta = (\bar{\sigma}, \bar{\tau})$ be an infinite timed word and let $f \in APP_\mathcal{A}$ be an approximation function. Let us note $t_i = \tau_i - \tau_{i-1}$ for all $i \geq 1$, assuming $\tau_0 = 0$. An *f-run* of $\mathcal{A}$ (of state space S) on θ is also an infinite sequence of discrete and continuous transitions in $\mathcal{T}_{\mathcal{A},f}$ that is labelled by θ, i.e. a sequence of the form: $C_0 \overset{t_1}{\rightsquigarrow} C_1 \xrightarrow{\sigma_1}_f C_2 \overset{t_2}{\rightsquigarrow} C_3 \xrightarrow{\sigma_2}_f \ldots \overset{t_i}{\rightsquigarrow} C_{2i-1} \xrightarrow{\sigma_i}_f C_{2i} \ldots$.

In the rest of the paper, we (sometimes) use the abbreviation $C_i \xrightarrow{t,\sigma}_f C_{i+2}$ for

$C_i \overset{t}{\rightsquigarrow} C_{i+1} = C_i + t \xrightarrow{\sigma}_f C_{i+2}$: the sequence of configurations $C_0, C_2, C_4, \ldots, C_{2i}, \ldots$ corresponds to the sequence of configurations of depth $0, 1, 2, \ldots, i, \ldots$ (respectively) in the DAG defined previously, i.e. $V_0, V_1, V_2, \ldots, V_i, \ldots$. On Fig. 2, the Id-run π of the automaton $\mathcal{A}_\varphi$ of Fig. 1 is represented as a DAG and as a sequence of $\xrightarrow{\sigma}_{Id}$ transitions (above) and as a DAG (below). This example should convince the reader of the equivalence of these definitions.

f-language of OCATA We can now define the accepted language of an OCATA, parameterised by an approximation function f. A *branch* of an f-run G is a (finite or) infinite path in G_π. We note $Bran^\omega(G)$ the set of all *infinite* branches of G_π and, for a branch β, we note $Infty(\beta)$ the set of locations occurring infinitely often along β. An f-run is *accepting* iff $\forall \beta \in Bran^\omega(G)$, $Infty(\beta) \cap F \neq \emptyset$ (i.e. we consider Büchi acceptance condition). We say that an infinite timed word θ is f-accepted by $\mathcal{A}$ iff there exists an accepting f-run of $\mathcal{A}$ on θ. We note $L^\omega_f(\mathcal{A})$ the language of all infinite timed words f-accepted by $\mathcal{A}$. We close the section by observing that a standard semantics for OCATA (where clock valuations are punctual values instead of intervals) is a particular case of the interval semantics, obtained by using the approximation function Id s.t. $Id(C) = \{C\}$ for all C. We denote by $L^\omega(\mathcal{A})$ the language $L^\omega_{Id}(\mathcal{A})$. Then, the following proposition shows the impact of approximation functions on the accepted language of the OCATA: they can only lead to *under-approximations* of $L^\omega(\mathcal{A})$.

Proposition 7. *For all OCATA $\mathcal{A}$, for all $f \in APP_\mathcal{A}$: $L^\omega_f(\mathcal{A}) \subseteq L^\omega(\mathcal{A})$.*

Proof (Idea). In Id-runs, all clock values are punctual, while in f-runs, clock values can be non-punctual intervals. Consider a set $(\ell, v_1), \ldots, (\ell, v_n)$ of states in location ℓ and with punctual values $v_1 \leq \ldots \leq v_n$, and consider its approximation $s = (\ell[v_1, v_n])$. Then, if a σ-labeled transition is firable from s, it is also firable from all (ℓ, v_i). The converse is not true: there might be a set of σ-labeled transitions that are firable from each (ℓ, v_i), but no σ-labeled transition firable from s, because *all clock values* in I must satisfy the transition guard. □

4 TOCATA: a class of OCATA for MITL

In this section, we introduce the class of *tree-like* OCATA (TOCATA for short), and show that, when applying, to an MITL formula φ, the construction defined by Ouaknine and Worrell [16] in the setting of MTL interpreted on *finite words*, one obtains a TOCATA that accepts the *infinite words* language of φ. To prove this result, we rely on the specific properties of TOCATA (in particular, we show that their acceptance condition can be made simpler than in the general case). Then, we show that there is a family of *bounded approximation functions* $f^\star_\varphi$, s.t., for every MITL formula φ, $L^\omega_{f^\star_\varphi}(\mathcal{A}_\varphi) = L^\omega(\mathcal{A}_\varphi)$. This result will be crucial to the definition of our on-the-fly model-checking algorithm in Section 5. We also exploit it to define a natural procedure that builds, for all MITL formula φ, a *Büchi timed automaton* $\mathcal{B}_\varphi$ accepting $[\![\varphi]\!]$.

From MITL to OCATA. We begin by recalling the syntactic translation from MTL (a superset of MITL) to OCATA, as defined by Ouaknine and Worrell [16]. Observe that it has been defined in the setting of *finite words*, hence we will need to prove that it is still correct in the infinite words setting. Let φ be an MITL formula (in negative normal form). We let $\mathcal{A}_\varphi = (\Sigma, L, \ell_0, F, \delta)$ where: L is the set containing the initial copy of φ, noted 'φ_{init}', and all the formulas of $Sub(\varphi)$ whose outermost connective is 'U' or '$\tilde{U}$'; $\ell_0 = \varphi_{init}$; F is the set of the elements of L of the form $\varphi_1 \tilde{U}_I \varphi_2$. Finally δ is defined by induction on the structure of φ:

- $\delta(\varphi_{init}, \sigma) = x.\delta(\varphi, \sigma)$
- $\delta(\varphi_1 \vee \varphi_2, \sigma) = \delta(\varphi_1, \sigma) \vee \delta(\varphi_2, \sigma)$; $\delta(\varphi_1 \wedge \varphi_2, \sigma) = \delta(\varphi_1, \sigma) \wedge \delta(\varphi_2, \sigma)$
- $\delta(\varphi_1 U_I \varphi_2, \sigma) = (x.\delta(\varphi_2, \sigma) \wedge x \in I) \vee (x.\delta(\varphi_1, \sigma) \wedge \varphi_1 U_I \varphi_2 \wedge x \leq sup(I))$
- $\delta(\varphi_1 \tilde{U}_I \varphi_2, \sigma) = (x.\delta(\varphi_2, \sigma) \vee x \notin I) \wedge (x.\delta(\varphi_1, \sigma) \vee \varphi_1 \tilde{U}_I \varphi_2 \vee x > sup(I))$
- $\forall \sigma_1, \sigma_2 \in \Sigma$: $\delta(\sigma_1, \sigma_2) = \begin{cases} \text{true if } \sigma_1 = \sigma_2 \\ \text{false if } \sigma_1 \neq \sigma_2 \end{cases}$ and $\delta(\neg\sigma_1, \sigma_2) = \begin{cases} \text{false if } \sigma_1 = \sigma_2 \\ \text{true if } \sigma_1 \neq \sigma_2 \end{cases}$
- $\forall \sigma \in \Sigma$: $\delta(\top, \sigma) = \top$ and $\delta(\bot, \sigma) = \bot$.

Example 8. As an example, consider again the OCATA $\mathcal{A}_\varphi$ in Fig. 1. It accepts exactly $[\![\Box(a \Rightarrow \Diamond_{[1,2]} b)]\!]$. It has been obtained by means of the above construction (trivial guards such as $x \leq +\infty$, and the state φ_{init} have been omitted).

Tree-like OCATA Let us now define a strict subclass of OCATA that captures all the infinite words language of MTL formulas, but whose acceptance condition can be made simpler. An OCATA $\mathcal{A} = (\Sigma, L, \ell_0, F, \delta)$ is a TOCATA iff there exists a partition $L_1, L_2, \ldots, L_m$ of L and a partial order $\preccurlyeq$ on the sets $L_1, L_2, \ldots, L_m$ s.t.: (i) each L_i contains either only accepting states or no accepting states and (ii) the partial order $\preccurlyeq$ is compatible with the transition relation and yields the 'tree-like' structure of the automaton. Here is the formal definition of a TOCATA.

Definition 9. *An OCATA $\mathcal{A} = (\Sigma, L, \ell_0, F, \delta)$ is a TOCATA iff there exists a partition $L_1, L_2, \ldots, L_m$ of L satisfying:*

- *each L_i contains either only accepting states or no accepting states: $\forall 1 \leq i \leq m$ either $L_i \subseteq F$ or $L_i \cap F = \emptyset$, and*
- *there is a partial order $\preccurlyeq$ on the sets $L_1, L_2, \ldots, L_m$ compatible with the transition relation and that yields the 'tree-like' structure of the automaton in the following sense: $\preccurlyeq$ is s.t. $L_j \preccurlyeq L_i$ iff $\exists \sigma \in \Sigma$, $\ell \in L_i$ and $\ell' \in L_j$ such that ℓ' is present in $\delta(\ell, \sigma)$.*

In particular, OCATA built from *MTL* formulas, such as $\mathcal{A}_\varphi$ in Fig. 1, are TOCATA. Since MTL is a superset of MITL, this proposition is true in particular for MITL formulas:

Proposition 10. *For every* MTL *formula φ, $\mathcal{A}_\varphi$ is a TOCATA.*

Proof. Let $L = \{\ell_1, \ell_2, \ldots, \ell_m\}$ be the locations of $\mathcal{A}_\varphi$. We consider the partition $\{\ell_1\}, \{\ell_2\}, \ldots, \{\ell_m\}$ of L and the order $\preccurlyeq$ s.t. $\{\ell_j\} \preccurlyeq \{\ell_i\}$ iff ℓ_j is a subformula of ℓ_i. It is easy to check that they satisfy the definition of TOCATA. □

Properties of TOCATA Let us now discuss two peculiar properties of TOCATA that are not enjoyed by OCATA. The first one is concerned with the acceptance condition. In the general case, a run of an OCATA is accepting iff all its branches visit accepting states infinitely often. Thanks to the partition characterising a TOCATA, this condition can be made simpler: a run is now accepting iff each branch eventually visits accepting states *only*, because it reaches a partition of the locations that are all accepting.

Proposition 11. *An Id-run G_π of a TOCATA $\mathcal{A} = (\Sigma, L, \ell_0, F, \delta)$ is* accepting *iff $\forall \beta = \beta_0\beta_1 \dots \beta_i \dots \in Bran^\omega(G_\pi)$, $\exists n_\beta \in \mathbb{N}$ s.t. $\forall i > n_\beta$: $\beta_i = ((\ell, v), i)$ implies $\ell \in F$.*

Proof. First, remark that the "if" case is trivial. In the following, we prove the "only if" case. As $\mathcal{A}$ is a tree-like OCATA, there exists a partition of L into disjoint subsets $L_1, L_2, \dots, L_m$ satisfying:

1. $\forall 1 \leq i \leq m$ either $L_i \subseteq F$ or $L_i \cap F = \emptyset$, and
2. there is a partial order $\preccurlyeq$ on the sets $L_1, L_2, \dots, L_m$ such that $L_j \preccurlyeq L_i$ iff $\exists \sigma \in \Sigma$, $\ell \in L_i$ and $\ell' \in L_j$ such that ℓ' is present in $\delta(\ell, \sigma)$.

Let π be an accepting Id-run of $\mathcal{A}$ and G_π its associated DAG. Let $\beta = \beta_0\beta_1 \dots \beta_i \dots$ be a (finite or infinite) branch of G_π, we must prove that either β is finite, either $\exists n_\beta \in \mathbb{N}$ s.t. $\forall i > n_\beta$: $\beta_i = ((\ell, v), i)$ implies $\ell \in F$. As π is accepting, either β is finite, either there exists a smallest n_0 such that β_{n_0} has an accepting location. If β is finite, we are done. So, let us suppose that β is infinite. In this case, there exists a smallest n_0 such that β_{n_0} has an accepting location, say $\ell \in L_i$ for a certain $1 \leq i \leq m$. 1. implies that $L_i \subseteq F$. Thanks to 2., β_{n_0+1} is found:

a. taking an arc looping on L_i, or
b. taking an arc going to L_j, for a certain $1 \leq j \leq m$ with $L_j \preccurlyeq L_i$ and $L_i \neq L_j$.

Remark that case a. can be repeated infinitely many times, while case b. can happen at most $m - 1$ times. So, there exists $n^\star \geq n_0$ and $1 \leq i^\star \leq m$ such that $\forall i > n^\star$: $\beta_i = ((\ell, v), i)$ implies $\ell \in L_{i^\star}$. As π is an accepting Id-run, 1. implies that $L_{i^\star} \subseteq F$ and $n^\star$ is the research n_β: $\forall i > n^\star$, $\beta_i = ((\ell, v), i)$ implies $\ell \in F$. □

The second property of interest for us is that TOCATA can be easily complemented. One can simply swap accepting and non-accepting locations, and 'dualise' the transition relation, *without changing the acceptance condition*[2] (as in the case of OCATA on finite words [16]). Formally, the dual of a formula $\varphi \in \Gamma(L)$ is the formula $\overline{\varphi}$ defined inductively as follows. $\forall \ell \in L$, $\overline{\ell} = \ell$; $\overline{false} = true$ and $\overline{true} = false$; $\overline{\varphi_1 \vee \varphi_2} = \overline{\varphi_1} \wedge \overline{\varphi_2}$; $\overline{\varphi_1 \wedge \varphi_2} = \overline{\varphi_1} \vee \overline{\varphi_2}$; $\overline{x.\varphi} = x.\overline{\varphi}$; the dual of a clock constraint is its negation (for example: $\overline{x \leq c} = x > c$). Then,

[2] In general, applying this construction yields an OCATA *with co-Büchi acceptance condition* for the complement of the language.

for all TOCATA $\mathcal{A} = (\Sigma, L, \ell_0, F, \delta)$, we let $\mathcal{A}^C = (\Sigma, L, \ell_0, L \setminus F, \overline{\delta})$ where $\overline{\delta}(\ell, \sigma) = \overline{\delta(\ell, \sigma)}$. Thanks to Proposition 11, we will prove that $\mathcal{A}^C$ accepts the complement of $\mathcal{A}$'s language.

Before proving this, we make several useful observations about the transition relation of an OCATA. Let δ be the transition function of some OCATA, let ℓ be a location, let σ be a letter, and assume $\delta(\ell, \sigma) = \bigvee_k a_k$, where each a_k is an *arc*, i.e. a conjunction of atoms of the form: ℓ', $x.\ell'$, $x \bowtie c$, $0 \bowtie c$, $\top$ or $\bot$. Then, we observe that *each minimal model* of $\delta(\ell, \sigma)$ wrt some interval I corresponds to *firing one of the arcs a_k from* (ℓ, I). That is, each minimal model can be obtained by choosing an arc a_k from $\delta(\ell, \sigma)$, and applying the following procedure. Assume $a_k = \ell_1 \wedge \cdots \wedge \ell_n \wedge x.(\ell_{n+1} \wedge \cdots \wedge \ell_m) \wedge \varphi$, where φ is a conjunction of clock constraints. Then, a_k is firable from a minimal model (ℓ, σ) iff $I \models \varphi$ (otherwise, no minimal model can be obtained from a_k). In this case, the minimal model is $\{(\ell_i, I) | 1 \leq i \leq n\} \cup \{(\ell_i, [0,0]) | n+1 \leq i \leq m\}$.
From now on, we consider that $\delta(\ell, \sigma)$ is always written in disjunctive normal form, i.e. $\delta(\ell, \sigma) = \bigvee_{k \in K} \bigwedge A_k$, where the terms A_k might be $\ell, x.\ell, x \bowtie c, 0 \bowtie c, \top$ or $\bot$, for $\ell \in L$ and $c \in \mathbb{N}$. Then, each minimal model M of $\delta(\ell, \sigma)$ wrt I has the form $A_k[I]$, for some $k \in K$, where $A_k[I] = \{(\ell, J) \mid \ell \in A_k\} \cup \{(\ell, \{0\}) \mid x.\ell \in A_k\}$ and I satisfies all the clocks constraints of A_k.

Proposition 12. *For all TOCATA $\mathcal{A}$, $L^\omega(\mathcal{A}^C) = T\Sigma^\omega \setminus L^\omega(\mathcal{A})$.*

Proof.
Claim. Let θ be an infinite timed word. Let π be an Id-run of $\mathcal{A}$ on θ and π' be a Id-run of $\mathcal{A}^C$ on θ. Then, π and π' have a common branch.
Proof of the claim. Let π be a Id-run of $\mathcal{A}$, denoted $C_0 \overset{t_1}{\rightsquigarrow} C_1 \xrightarrow{\sigma_1} C_2 \overset{t_2}{\rightsquigarrow} C_3 \xrightarrow{\sigma_2} \ldots \overset{t_n}{\rightsquigarrow} C_{2n-1} \xrightarrow{\sigma_n} C_{2n} \ldots$ and π' be a Id-run of $\mathcal{A}^C$, denoted $C'_0 \overset{t_1}{\rightsquigarrow} C'_1 \xrightarrow{\sigma_1} C'_2 \overset{t_2}{\rightsquigarrow} C'_3 \xrightarrow{\sigma_2} \ldots \overset{t_n}{\rightsquigarrow} C'_{2n-1} \xrightarrow{\sigma_n} C'_{2n} \ldots$. We will recursively construct a common branch of π and π'. We will denote its elements by $e_0 e_1 e_2 ...$ (so, each $e_i \in C_i \cap C'_i$ and is a certain state (ℓ, v)).
Basis: $C_0 = C'_0 = \{(\ell_0, 0)\}$, so, we take $e_0 = (\ell_0, 0)$.
Induction: Suppose we constructed a common beginning of branch of π and π': $e_0 e_1 e_2 ... e_{2n-3} e_{2n-2}$. Let us show that we can extend this common beginning of branch, constructing e_{2n-1} and e_{2n}.
We know that $e_{2n-2} \in C_{2n-2} \cap C'_{2n-2}$. Suppose $e_{2n-2} = (\ell, v)$. Then, transitions $C_{2n-2} \overset{t_n}{\rightsquigarrow} C_{2n-1}$ and $C'_{2n-2} \overset{t_n}{\rightsquigarrow} C'_{2n-1}$ both send $e_{2n-2} = (\ell, v)$ on $(\ell, v + t_n)$. So we can construct $e_{2n-1} = (\ell, v + t_n) \in C_{2n-1} \cap C'_{2n-1}$.
Now, suppose that $C_{2n-1} = \{(\ell_i, v_i)_{i \in I}\}$ and let $i^\star \in I$ such that $e_{2n-1} = (\ell_{i^\star}, v_{i^\star})$. C_{2n} is obtained thanks to $\delta(\ell_i, \sigma_n)$ (for all $i \in I$), which can be written as (in disjunctive normal form) $\bigvee_{j \in J} \bigwedge_{k \in K_j} A_{jk}$ where each A_{jk} is a term of type ℓ, $x.\ell$ or $x \bowtie c$. For each $j \in J$, we can then define $A_j[v] = \{(\ell, v) \mid \exists k \in K_j : \ell \in A_{jk}\} \cup \{(\ell, 0) \mid \exists k \in K_j : x.\ell \in A_{jk}\}$: each $A_j[v_i]$, for a v_i satisfying the clock constraints present in $\bigcup_{k \in K_j} A_{jk}$, is a minimal model of $\delta(\ell_i, \sigma_n)$ wrt v_i

(and they are the only ones !). On one hand, there exists a $j^\star \in J$ such that $A_{j^\star}[v_{i^\star}] \subseteq C_{2n-1}$ and v_i satisfies the clock constraints present in $\bigcup_{k \in K_{j^\star}} A_{j^\star k}$: for each element (ℓ, v) of $A_{j^\star}[v_{i^\star}]$, there is a transition in π linking $e_{2n-1} = (\ell_{i^\star}, v_{i^\star})$ to (ℓ, v). On the other hand, $\overline{\delta}(\ell_i, \sigma_n) = \bigwedge_{j \in J} \bigvee_{k \in K_j} \overline{A_{jk}}$: for an A_{jk} of type ℓ or $x.\ell$, $\overline{A_{jk}} = A_{jk}$, and for $A_{jk} = x \bowtie c$, $\overline{A_{jk}}$ is the negation of $x \bowtie c$. As $v_{i^\star}$ satisfies the clock constraints present in $\bigcup_{k \in K_{j^\star}} A_{j^\star k}$, it does not satisfy the clock constraints present in $\bigcup_{k \in K_{j^\star}} \overline{A_{j^\star k}}$. Letting $C'_{2n-1} = \{(\ell_{i'}, v_{i'})_{i' \in I'}\}$, each minimal model of $\overline{\delta}(\ell_i, \sigma_n)$ wrt $v_{i'}$ must be such that, for each $j \in J$:

- either $v_{i'}$ satisfies a certain clock constraint present in $\bigcup_{k \in K_j} A_{jk}$,
- or this minimal model contains an element of $A_j[v_{i'}]$.

In particular, the minimal model of $\overline{\delta}(\ell_{i^\star}, \sigma_n)$ wrt $v_{i^\star}$ that is used in π' to construct C_{2n} contains an element of $A_{j^\star}[v_{i^\star}]$ (because it does not satisfy any clock constraint in $\bigcup_{k \in K_{j^\star}} A_{j^\star k}$) and there is a transition in π' linking $e_{2n-1} = (\ell_{i^\star}, v_{i^\star})$ to this element. It is this element we choose to be e_{2n}. □

We will prove the proposition thanks to this claim. We first prove that $L^\omega(\mathcal{A}) \cap L^\omega(\mathcal{A}^C) = \emptyset$, and then that $L^\omega(\mathcal{A}) \cup L^\omega(\mathcal{A}^C) = T\Sigma^\omega$.
$\mathbf{L^\omega(\mathcal{A}) \cap L^\omega(\mathcal{A}^C) = \emptyset}$. Suppose by contradiction that there exists a timed word θ such that $\theta \in L^\omega(\mathcal{A})$ and $\theta \in L^\omega(\mathcal{A}^C)$. Then, there exists accepting Id-runs π and π' (resp.) of $\mathcal{A}$ and $\mathcal{A}^C$ on θ. By the previous claim, π and π' have a common branch $e_0 e_1 e_2 ....$ As π is an accepting Id-run of $\mathcal{A}$, $\exists n \in \mathbb{N}$, $\forall m > n$, the location of e_m is in F. But as π' is an accepting Id-run of $\mathcal{A}^C$, $\exists n' \in \mathbb{N}$, $\forall m > n$, the location of e_m is in $L \setminus F$: this is impossible and so $\theta \notin L^\omega(\mathcal{A})$ or $\theta \notin L^\omega(\mathcal{A}^C)$.
$\mathbf{L^\omega(\mathcal{A}) \cup L^\omega(\mathcal{A}^C) = T\Sigma^\omega}$. Let θ be a timed word such that $\theta \notin L^\omega(\mathcal{A})$, we must prove that $\theta \in L^\omega(\mathcal{A}^C)$. We will inductively construct an accepting Id-run of $\mathcal{A}^C$ on θ. To actually construct an *accepting* Id-run, we will maintain an additional property of the beginning of Id-run we extend at each inductive step, say π_{2n} denoted $C_0 \overset{t_1}{\rightsquigarrow} C_1 \xrightarrow{\sigma_1} C_2 \overset{t_2}{\rightsquigarrow} C_3 \xrightarrow{\sigma_2} \ldots \overset{t_n}{\rightsquigarrow} C_{2n-1} \xrightarrow{\sigma_n} C_{2n}$: "for each branch β of π_{2n}, there exists a beginning of Id-run of $\mathcal{A}$ on θ such that:

- either β is a branch of π' and π' can not be prolonged into a complete Id-run of $\mathcal{A}$ (it is "blocking"),
- or π' can be prolonged into a complete Id-run π_c of $\mathcal{A}$ such that β is the beginning of a branch of π_c on which each location of F only occurs a finite number of times."

We note ($^\star\mathbf{2n}$) this property.
Basis. We construct $C_0 = \{(\ell_0, 0)\}$. As there is no accepting Id-run of $\mathcal{A}$ on θ, property ($^\star\mathbf{0}$) is trivially verified.

Induction. Suppose we constructed a beginning of Id-run π_{2n} of $\mathcal{A}^C$ on θ: $C_0 \overset{t_1}{\rightsquigarrow} C_1 \xrightarrow{\sigma_1} C_2 \overset{t_2}{\rightsquigarrow} C_3 \xrightarrow{\sigma_2} \ldots \overset{t_n}{\rightsquigarrow} C_{2n-1} \xrightarrow{\sigma_n} C_{2n}$ such that property ($\star\mathbf{2n}$) is verified. We will extend π_{2n} to obtain a beginning of Id-run $C_0 \overset{t_1}{\rightsquigarrow} C_1 \xrightarrow{\sigma_1} C_2 \overset{t_2}{\rightsquigarrow} C_3 \xrightarrow{\sigma_2} \ldots \overset{t_n}{\rightsquigarrow} C_{2n-1} \xrightarrow{\sigma_n} C_{2n} \overset{t_{n+1}}{\rightsquigarrow} C_{2n+1} \xrightarrow{\sigma_{n+1}} C_{2n+2}$ such that property ($\star\mathbf{2n+2}$) is verified.
First, we construct $C_{2n+1} = C_{2n} + t_{n+1}$, so that π_{2n+1} is $C_0 \overset{t_1}{\rightsquigarrow} C_1 \xrightarrow{\sigma_1} C_2 \overset{t_2}{\rightsquigarrow} C_3 \xrightarrow{\sigma_2} \ldots \overset{t_n}{\rightsquigarrow} C_{2n-1} \xrightarrow{\sigma_n} C_{2n} \overset{t_{n+1}}{\rightsquigarrow} C_{2n+1}$. ($\star\mathbf{2n+1}$) is trivially verified thanks to ($\star\mathbf{2n}$).
Now, suppose that $C_{2n+1} = \{(\ell_i, v_i)_{i\in I}\}$. We know that (($\star\mathbf{2n+1}$)) for each branch β of π_{2n}, there exists a beginning of Id-run π' of $\mathcal{A}$ on θ such that:

- either β is a branch of π' and π' can not be prolonged into a complete Id-run of $\mathcal{A}$,
- or π' can be prolonged into a complete Id-run π_c of $\mathcal{A}$ such that β is the beginning of a branch of π_c on which each location of F only occurs a finite number of times.

In particular, for each $(\ell_i, v_i) \in C_{2n+1}$, there exists a beginning of Id-run π_i of $\mathcal{A}$ on θ such that (ℓ_i, v_i) is the (2n+1)th element of a branch of π_i. π_i must then evolve towards a minimal model of $\delta(\ell_i, \sigma_{n+1})$ wrt v_i. Let us note $\delta(\ell_i, \sigma_{n+1})$ as $\bigvee_{j\in J} \bigwedge_{k\in K_j} A_{jk}$, as previously. As there is no accepting Id-run of $\mathcal{A}$ on θ, whatever is the minimal model $A_j[v_i]$ we decide to take as successor of (ℓ_i, v_i) in a beginning of Id-run of $\mathcal{A}$ on θ, and no matter how it is then prolonged, it will contain a branch on which each location of F only occurs a finite number of times or will be "blocking". Let us note $(\ell_i^j, v_i^j)_{i\in I, j\in J}$ the successors of (ℓ_i, v_i) on the branches on which each location of F only occurs a finite number of timed or are "blocking"[3], considering a certain Id-run of $\mathcal{A}$ on θ going from (ℓ_i, v_i) to the minimal model $A_j[v_i]$. We construct $C_{2n+2} = \{(\ell_i^j, v_i^j)_{i\in I, j\in J}\}$ (it is actually the union, for $i \in I$, of minimal models of $\overline{\delta}(\ell_i, \sigma_{n+1}) = \bigwedge_{j\in J} \bigvee_{k\in K_j} \overline{A_{jk}}$ wrt v_i). By construction, ($\star\mathbf{2n+2}$) is verified.
Thanks to this induction, we can construct an infinite Id-run π, $C_0 \overset{t_1}{\rightsquigarrow} C_1 \xrightarrow{\sigma_1} C_2 \overset{t_2}{\rightsquigarrow} C_3 \xrightarrow{\sigma_2} \ldots \overset{t_n}{\rightsquigarrow} C_{2n-1} \xrightarrow{\sigma_n} C_{2n} \ldots$, such that for each branch β of π, there exists a Id-run or a beginning of Id-run π' of $\mathcal{A}$ on θ such that:

- either β is a branch of π' and π' can not be prolonged into a complete Id-run of $\mathcal{A}$ (it is "blocking"): it means that β is a finite and non-blocking branch of π,

[3] Remark that a branch of a Id-run of $\mathcal{A}$ is blocking reading σ_{n+1} iff $\bigcup_{k\in K_j} A_{jk}$ contains a clock constraint $x \bowtie c$ not satisfied in v_i or there is no transition from ℓ_i reading σ_{n+1}. In the first case $\overline{A_{jk}}$ contains the negation of $x \bowtie c$, which is verified by v_i, and in the second case $\delta(\ell_i, \sigma_{n+1}) = \text{false}$ and so $\overline{\delta}(\ell_i, \sigma_{n+1}) = \text{true}$. In both cases (in particular) $\bigvee_{k\in K_j} \overline{A_{jk}}$ will be satisfied in $\overline{\delta}(\ell_i, \sigma_{n+1})$ even if no successor is attributed to (ℓ_i, v_i) (the branch ending in (ℓ_i, v_i) in pi_{2n} is finite but not blocking).

– or π' is a Id-run of $\mathcal{A}$ such that β is a branch of this Id-run on which each location of F only occurs a finite number of times.

So, all the infinite branches of π visit $L \setminus F$ infinitely often and π is an accepting Id-run of $\mathcal{A}^C$ on θ. □

TOCATA and MITL Equipped with those results we can now expose the two main results of this section. First, the translation from MTL to OCATA introduced in [16] carries on to infinite words (to the best of our knowledge this had not been proved before and does not seem completely trivial since our proof requires the machinery of TOCATA developed in this paper). Second, for every MITL formula φ, we can devise an $M(\varphi)$-bounded[4] approximation function $f^\star_\varphi$ to bound the number of intervals needed along all runs of the intervals semantics of the TOCATA $\mathcal{A}_\varphi$, while retaining the semantics of φ. Notice that this second property fails when applied to formulae φ of MTL.

Theorem 13. *For every MTL formula φ: $L^\omega(\mathcal{A}_\varphi) = [\![\varphi]\!]$.*

Proof. This has been proved in the finite words case in [16, Prop. 6.4]. This proof relies crucially on the fact that OCATA can be complemented in this case. Thanks to Proposition 12, we can adapt the proof of [16]. □

We will now show there exists a family of bounded approximation functions $f^\star_\varphi$ s.t. for every MITL formula φ, $L^\omega_{f^\star_\varphi}(\mathcal{A}_\varphi) = L^\omega(\mathcal{A}_\varphi)$. We first recall from [5] the exact value of the upper bound $M(\varphi)$ on the number of clock copies (intervals) we need to consider in the configurations to recognise an MITL formula φ. Then, we recall the definition, for an MITL formula φ, of the approximation function $f^\star_\varphi$.
Let φ be an MITL formula in negative normal form. We define $M(\varphi)$, thanks to $M^\infty(\varphi)$ and $M^1(\varphi)$ defined as follows

– if $\varphi = \sigma$ or $\varphi = \neg\sigma$ (with $\sigma \in \Sigma$), then $M(\varphi) = 2$ and $M^\infty(\varphi) = M^1(\varphi) = 0$.
– if $\varphi = \varphi_1 \wedge \varphi_2$, then $M(\varphi) = \max\{2, M^1(\varphi_1) + M^1(\varphi_2)\}$, $M^\infty(\varphi) = M^\infty(\varphi_1) + M^\infty(\varphi_2)$ and $M^1(\varphi) = M^1(\varphi_1) + M^1(\varphi_2)$.
– if $\varphi = \varphi_1 \vee \varphi_2$, then $M(\varphi) = \max\{2, M^1(\varphi_1), M^1(\varphi_2)\}$, $M^\infty(\varphi) = \max\{M^\infty(\varphi_1), M^\infty(\varphi_2)\}$ and $M^1(\varphi) = \max\{M^1(\varphi_1), M^1(\varphi_2)\}$.
– if $\varphi = \varphi_1 U_I \varphi_2$, then $M(\varphi) = \max\{2, M^\infty(\varphi_1) + M^1(\varphi_2) + 1\}$, $M^\infty(\varphi) = \left(4 \times \left\lceil \frac{inf(I)}{|I|} \right\rceil + 2\right) + M^\infty(\varphi_1) + M^\infty(\varphi_2)$ and $M^1(\varphi) = M^\infty(\varphi_1) + M^1(\varphi_2) + 1$.
– if $\varphi = \varphi_1 \tilde{U}_I \varphi_2$, then $M(\varphi) = \max\{2, M_1(\varphi_1) + M_\infty(\varphi_2) + 1\}$, $M_\infty(\varphi) = \left(2 \times \left\lceil \frac{sup(I)}{|I|} \right\rceil + 2\right) + M_\infty(\varphi_1) + M_\infty(\varphi_2)$ and $M_1(\varphi) = M_1(\varphi_1) + M_\infty(\varphi_2) + 1$.

[4] $M(\varphi) \leq |\varphi| \times \max_{I \in \mathcal{I}_\varphi} \left(4 \times \left\lceil \frac{\inf(I)}{|I|} \right\rceil + 2, 2 \times \left\lceil \frac{\sup(I)}{|I|} \right\rceil + 2\right)$, where $\mathcal{I}_\varphi$ is the set of all the intervals that occur in φ.

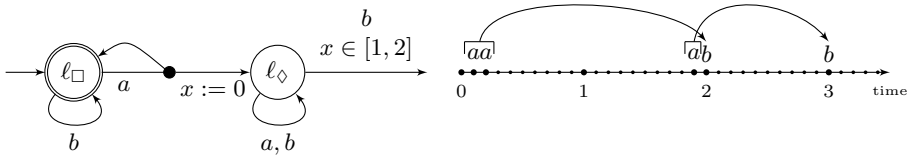

Fig. 1. (left) OCATA $\mathcal{A}_\varphi$ with $\varphi = \Box(a \Rightarrow \Diamond_{[1,2]} b)$. (right) The grouping of clocks.

Throughout this description, we assume an OCATA $\mathcal{A}$ with set of locations L. Let $S = \{(\ell, I_0), (\ell, I_1), \ldots, (\ell, I_m)\}$ be a set of states of $\mathcal{A}$, all in the same location ℓ, with $I_0 < I_1 < \cdots < I_m$. Then, we let $\mathsf{Merge}(S) = \{(\ell, [0, sup(I_1)]), (\ell, I_2), \ldots, (\ell, I_m)\}$ if $I_0 = [0,0]$ and $\mathsf{Merge}(S) = S$ otherwise, i.e., $\mathsf{Merge}(S)$ is obtained from S by *grouping* I_0 and I_1 iff $I_0 = [0,0]$, otherwise $\mathsf{Merge}(S)$ does not modify S. Observe that, in the former case, if I_1 is not a singleton, then $\|\mathsf{Merge}(S)\| = \|S\| - 1$. Now, we can lift the definition of Merge to configurations. Let C be a configuration of $\mathcal{A}$ and let $k \in \mathbb{N}$. We let:

$$\mathsf{Merge}(C,k) = \left\{C' \mid \|C'\| \leq k \text{ and } \forall \ell \in L : C'(\ell) \in \left\{\mathsf{Merge}\left(C(\ell)\right), C(\ell)\right\}\right\}$$

Observe that $\mathsf{Merge}(C,k)$ is a (possibly empty) *set of configurations*, where each configuration (i) has at most k clock copies, and (ii) can be obtained by applying (if possible) or not the Merge function to each $C(\ell)$. Let us now define a family of k-bounded approximation functions, based on Merge. Let $k \geq 2 \times |L|$ be a bound and let C be a configuration, assuming that $C(\ell) = \{I^\ell_1, \ldots, I^\ell_{m_\ell}\}$ for all $\ell \in L$. Then:

$$F^k(C) = \begin{cases} \mathsf{Merge}(C,k) & \text{If } \mathsf{Merge}(C,k) \neq \emptyset \\ \left\{(\ell, [inf(I^\ell_1), sup(I^\ell_{m_\ell})]) \mid \ell \in L\right\} & \text{otherwise} \end{cases}$$

Roughly speaking, the $F^k(C)$ function tries to obtain configurations C' that approximate C and s.t. $\|C'\| \leq k$, using the Merge function. If it fails to, i.e., when $\mathsf{Merge}(C,k) = \emptyset$, $F^k(C)$ returns a single configuration, obtained from C be grouping all the intervals in each location. The latter case occurs in the definition of F^k for the sake of completeness. When the OCATA $\mathcal{A}$ has been obtained from an MITL formula φ, and for k big enough (see hereunder) each $\theta \in [\![\varphi]\!]$ will be recognised by at least one F^k-run of $\mathcal{A}$ that traverses only configurations obtained thanks to Merge. We can now finally define $f^\star_\varphi$ for every MITL formula φ, by letting $f^\star_\varphi = F^K$, where $K = \max\{2 \times |L|, M(\varphi)\}$.
We insist on the importance of Proposition 11 and Theorem 13 that enable to adapt with a large facility the proof of Theorem 13 from [5] to infinite words, providing the following theorem:

Theorem 14. *For every MITL formula φ, there exists an $M(\varphi)$-bounded approximation function $f^\star_\varphi$ such that $L^\omega_{f^\star_\varphi}(\mathcal{A}_\varphi) = L^\omega(\mathcal{A}_\varphi) = [\![\varphi]\!]$.*

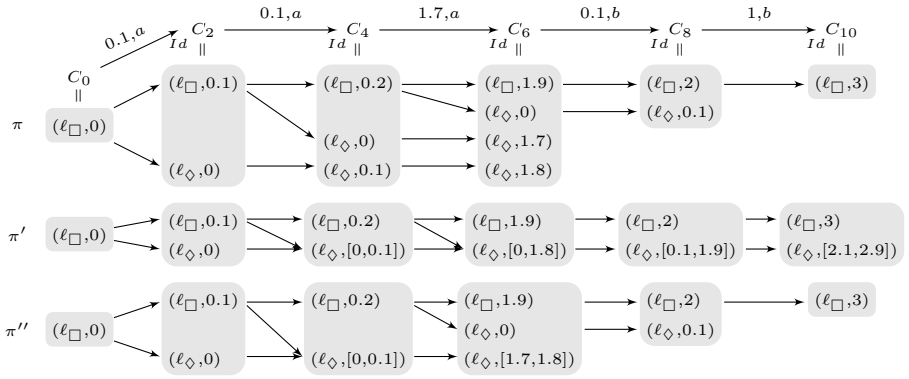

Fig. 2. Several OCATA runs.

Example 15. Let us illustrate the idea behind the approximation function $f^\star_\varphi$ by considering again the run prefixes on $\theta = (a, 0.1)(a, 0.2)(a, 1.9)(b, 2)(b, 3)(b, 4)\ldots$ in Fig. 2. The two first positions (with $\sigma_1 = \sigma_2 = a$) of θ satisfy $\Diamond_{[1,2]}b$, *thanks to the b in position* 4 (with $\tau_4 = 2$), while position 3 (with $\sigma_3 = a$) satisfies $\Diamond_{[1,2]}b$ *thanks to the b in position* 5 (with $\tau_5 = 3$), see Fig. 1 (right). Hence, $f^\star_\varphi$ groups the two clock copies created in $\ell_\Diamond$ when reading the two first a's, but keeps the third one apart. This yields the $f^\star_\varphi$-run π'' in Fig. 2. On the other hand, the strategy of grouping all the clock copies present in each location, which yields π', is not a good solution. This prefix cannot be extended to an accepting run because of the copy in state $(\ell_\Diamond, [2.1, 2.9])$ in the rightmost configuration, that will never be able to visit an accepting location.

A Büchi transition system for each OCATA. Thanks to the bound $M(\varphi)$ on the number of necessary clock copies and the approximation function $f^\star_\varphi$, given by Theorem 14, we can use the method of Miyano Hayashi [14] to construct a Büchi timed transition system from A_φ: $\mathsf{MHTS}\left(\mathcal{A}_\varphi, f^\star_\varphi\right)$. Each state of a configuration of $\mathcal{A}_\varphi$ will be marked by $\top$ or $\bot$. Intuitively, a state is marked by $\top$ iff all the branches it belongs to have visited a final location of $\mathcal{A}_\varphi$ since the last accepting state of $\mathsf{MHTS}\left(\mathcal{A}_\varphi, f^\star_\varphi\right)$ (i.e., a state where all markers are $\top$). Formally, for an *OCATA* $\mathcal{A}$ and an approximation function f, we define the transition system $\mathsf{MHTS}\,(\mathcal{A}, f) = (\Sigma, S^{\mathsf{MH}}, s_0^{\mathsf{MH}}, \rightsquigarrow^{\mathsf{MH}}, \rightarrow^{\mathsf{MH}}, \alpha)$ where: $S^{\mathsf{MH}} = \{(\ell_k, I_k, m_k)_{k\in K} \mid \{(\ell_k, I_k)_{k\in K}\}$ is a configuration of $\mathcal{A}$ and $\forall k \in K, m_k \in \{\top, \bot\}\}$, $s_0^{\mathsf{MH}} = \{(\ell_0, [0, 0], \bot)\}$ (because $\ell_0 \notin F$), $\alpha = \{(\ell_k, I_k, m_k)_{k\in K} \in S^{\mathsf{MH}} \mid \forall k \in K, m_k = \top\}$. For $t \in \mathbb{R}$ and $s, s' \in S$, supposing $s = \{(\ell_k, I_k, m_k)_{k\in K}\}$, we have $s \overset{t}{\rightsquigarrow}^{\mathsf{MH}} s'$ iff $s' = \{(\ell_k, I_k + t, m_k)_{k\in K}\}$. $\rightsquigarrow^{\mathsf{MH}} = \bigcup_{t\in\mathbb{R}} \overset{t}{\rightsquigarrow}^{\mathsf{MH}}$.

"$\rightarrow^{\mathsf{MH}}$", representing the discrete transitions, is defined in way that (1) a transition with $\rightarrow^{\mathsf{MH}}$ between two states of S^{MH} corresponds to a discrete transition between the configurations of $\mathcal{A}$ they represent and (2) the markers of the third component of states of S^{MH} are kept updated. Actually, if $s \overset{\sigma}{\rightarrow}^{\mathsf{MH}} s'$, we want

the last component of a trio of s' to be $\bot$ iff its first component is not in F and, either it comes from the grouping of trios emanating from trios such that at least one of them had $\bot$ as last component. Moreover, if $s \in \alpha$, we want to start again the marking: we put all the third components of the trios of s to $\bot$ before proceeding as previously. Formally:

(i) For $s \in S^{\mathsf{MH}} \setminus \alpha$ and $s' \in S^{\mathsf{MH}}$, supposing $s = \{(\ell_k, I_k, m_k)_{k\in K}\}$ and $s' = \{(\ell_{k'}, I_{k'}, m_{k'})_{k'\in K'}\}$, $s \overset{\sigma}{\to}^{\mathsf{MH}} s'$ iff
   (a) $\{(\ell_k, I_k)_{k\in K}\} \xrightarrow{\sigma}_f \{(\ell_{k'}, I_{k'})_{k'\in K'}\}$, i.e $\{(\ell_{k'}, I_{k'})_{k'\in K'}\} \in f(\mathsf{Succ}(\{(\ell_k, I_k)_{k\in K}\}, \sigma))$;
   (b) $\forall k' \in K'$: $(\ell_{k'} \in F \;\Rightarrow\; m_{k'} = \top)$;
   (c) $\forall k' \in K'$ with $\ell_{k'} \notin F$: if $\exists k \in K$ s.t. $(\ell_k, I_k, \bot) \in s$ and $(\ell_{k'}, I_{k'}) \in \mathsf{dest}(\mathsf{conf}(s), \mathsf{conf}(s'), (\ell_k, I_k, \bot))$, we have $m_{k'} = \bot$; otherwise, $m_{k'} = \top$.

(ii) For $s \in \alpha$ and $s' \in S^{\mathsf{MH}}$, supposing $s = \{(\ell_k, I_k, \top)_{k\in K}\}$, $s \overset{\sigma}{\to}^{\mathsf{MH}} s'$ iff $\{(\ell_k, I_k, \bot)_{k\in K}\} \overset{\sigma}{\to}^{\mathsf{MH}} s'$ according to the rules in (i).

Proposition 16. *Let $\mathcal{A}$ be an OCATA and f be an approximation function:* $L^\omega(\mathsf{MHTS}(\mathcal{A}, f)) = L^\omega_f(\mathcal{A})$.

Proof. ($\supseteq$) Let $\theta = (\bar\sigma, \bar\tau) \in L^\omega_f(\mathcal{A})$, with $\bar\sigma = \sigma_1\sigma_2\cdots\sigma_n\ldots$ and $\bar\tau = \tau_1\tau_2\cdots\tau_n\ldots$. We will prove that $\theta \in L^\omega(\mathsf{MHTS}(\mathcal{A}, f))$. Let us note $t_i = \tau_i - \tau_{i-1}$ for all $1 \le i \le |\theta|$, assuming $\tau_0 = 0$. We have an accepting f-run of $\mathcal{A}$ on θ, say $\pi : C_0 \overset{t_1}{\rightsquigarrow} C_1 \xrightarrow{\sigma_1}_f C_2 \overset{t_2}{\rightsquigarrow} C_3 \xrightarrow{\sigma_2}_f \ldots \overset{t_i}{\rightsquigarrow} C_{2i-1} \xrightarrow{\sigma_i}_f C_{2i}\ldots$. We must prove that there is an accepting run of $\mathsf{MHTS}(\mathcal{A}, f)$ on θ, say π': $s_0 \overset{t_1}{\rightsquigarrow}^{\mathsf{MH}} s_1 \overset{\sigma_1}{\to}^{\mathsf{MH}} s_2 \overset{t_2}{\rightsquigarrow}^{\mathsf{MH}} s_3 \overset{\sigma_2}{\to}^{\mathsf{MH}} \ldots \overset{t_i}{\rightsquigarrow}^{\mathsf{MH}} s_{2i-1} \overset{\sigma_i}{\to}^{\mathsf{MH}} s_i \ldots$. We construct π' by induction, proving additionally that the two following properties hold for $j \ge 0$:

($\star 2j$) $C_{2j} = \{(\ell_k, I_k)_{k\in K}\}$ iff $s_{2j} = \{(\ell_k, I_k, m_k)_{k\in K} \mid m_k \in \{\top, \bot\}\}$;
($*2j$) if a location of F occurs on all the branches of π between the last configuration $C_{2j'}$ such that $s_{2j'} \in \alpha$ (or, failing that, between C_0) and C_{2j}, then, $s_{2j} \in \alpha$.

Basis: We know that $C_0 = (\ell_0, [0,0])$ and $s_0 = \{(\ell_0, [0,0], \bot)\}$. It is clear that ($\star 0$) and ($*0$) are verified because only $\ell_0 \notin F$ occurs on the (unique) branch of π.
Induction: Suppose that we constructed π' until s_{2i} and that ($\star 2j$) and ($*2j$) are verified $\forall 0 \le j \le i$. We will construct π' until $s_{2(i+1)}$ in way ($\star 2(i+1)$) and ($*2(i+1)$) will still be verified.

First, we must construct s_{2i+1} such that $s_{2i} \overset{t_{i+1}}{\rightsquigarrow}^{\mathsf{MH}} s_{2i+1}$. Suppose $s_{2i} = \{(\ell_k, I_k, m_k)_{k\in K}\}$, as ($\star 2i$) is verified by hypothesis, it means that C_{2i} can also be written as $\{(\ell_k, I_k)_{k\in K}\}$. We must choose s_{2i+1} to be $\{(\ell_k, I_k + t_{i+1}, m_k)_{k\in K}\}$. As $C_{2i} \overset{t_{i+1}}{\rightsquigarrow} C_{2i+1}$, $C_{2i+1} = C_{2i} + t_{i+1} = \{(\ell_k, I_k + t_{i+1})_{k\in K}\}$. Remark that the following property holds:

($\star 2i+1$) $C_{2i+1} = \{(\ell_k, I_k)_{k\in K}\}$ iff $s_{2i+1} = \{(\ell_k, I_k, m_k)_{k\in K} \mid m_k \in \{\top, \bot\}\}$.

Secondly, we must construct $s_{2(i+1)}$ such that $s_{2i+1} \overset{\sigma_{i+1}}{\to} s^{\mathsf{MH}}_{2(i+1)}$. We know that $C_{2i+1} \overset{\sigma_{i+1}}{\longrightarrow}_f C_{2(i+1)}$. Suppose $s_{2i+1} = \{(\ell_k, I_k, m_k)_{k\in K}\}$, as $(\star 2i+1)$ is verified, C_{2i+1} can be written as $\{(\ell_k, I_k)_{k\in K}\}$. Let us further suppose that $C_{2(i+1)} = \{(\ell_{k'}, I_{k'})_{k'\in K'}\}$. We construct $s_{2(i+1)} = \{(\ell_{k'}, I_{k'}, m_{k'})_{k'\in K'}\}$ to be the unique state[5] of S^{MH} such that $\{(\ell_k, I_k)_{k\in K}\} \overset{\sigma_{i+1}}{\longrightarrow}_f \{(\ell_{k'}, I_{k'})_{k'\in K'}\}$. $(\star 2(i+1))$ is trivially verified. It stays to prove that $(*2(i+1))$ is satisfied. Suppose that a location of F occurs on all the branches of π between the last configuration $C_{2j'}$ such that $s_{2j'} \in \alpha$ (or, failing that, between C_0) and $C_{2(i+1)}$. We must prove that $s_{2(i+1)} \in \alpha$, i.e. all the trios of $s_{2(i+1)}$ has $\top$ as last component. Let $\beta = \beta_0\beta_1\beta_2 \dots \beta_{2j'} \dots \beta_{2j} \dots \beta_{2(i+1)} \dots$ be a branch of π. The hypothesis implies there exists a transition $\overset{\sigma_j}{\longrightarrow}_f$, for $j' \leq j \leq i+1$, such that $\beta_{2j} = (\ell, I)$ for a certain $\ell \in F$. By $(*2j)$, $(\ell, I, m_k) \in s_{2j}$ for a certain m_k in $\{\top, \bot\}$, but point (i) (b) of the definition of $\to$ obliges m_k to be $\top$. So, the third components associated in π' to the different states of the branches of π will gradually (between $s_{2j'}$ and $s_{2(i+1)}$) become $\top$. We must still ensure they will eventually never become $\bot$ again. In fact, a pair (ℓ, I) of a certain state s_j (for $2j' \leq j \leq 2(i+1)$) of π, corresponding to $(\ell, I, \top)$ in π', can have a successor (ℓ', I') in π such that this corresponds to $(\ell', I', \bot)$ in π' iff s_j is accepting and $\ell' \notin F$ (which is not possible under the present hypothesis) or $(\ell', I', \bot)$ comes from the grouping of trios (thanks to f) emanating from trios such that at least one of them had $\bot$ as last component (case (i) (b) of the definition of $\to$). It means that no location of F occurs on one of the branches of π leading to (ℓ', I'), say $\beta' = \beta'_0\beta'_1\beta'_2 \dots \beta_{2j'} \dots \beta'_{2k'} \dots \beta'_{2(i+1)} \dots$, with $\beta'_{2k} = (\ell', I')$, since $\beta'_{2j'}$ (else, we contradict case (i) (c)). But, by hypothesis, a location of F occurs on all the branches of π between steps $2j'$ and $2(i+1)$, so there exists a transition $\overset{\sigma_{2\tilde{j}}}{\longrightarrow}_f$, for $k' < \tilde{j} \leq i+1$, such that $\beta'_{2\tilde{j}}$ has its location in F: once again, point (i) (b) of the definition of $\to$ obliges $m_{2\tilde{j}}$ to be $\top$.
As a location of F occurs on all branches of π between steps $2j'$ and $2(i+1)$ and there is only a finite number of branches leading to a state of $C_{2(i+1)}$, we conclude that we can only encounter this last case a finite number of time and so $s_{2(i+1)} \in \alpha$: $(*2(i+1))$ is satisfied.
To end this part of the proof, we must show that π' is accepting. The previous induction proves that $(*2j)$ is verified for all $j \geq 0$. As π is accepting, a location of F occurs on all the branches of π infinitely often, and so there is an infinite number of j and j' such that the antecedent of $(*2j)$ is true. So, in this same infinite number of times, we know that $s_{2j} \in \alpha$, what proves that π' is accepting.
$(\subseteq)$ Let $\theta = (\bar{\sigma}, \bar{\tau}) \in L^\omega(\mathsf{MHTS}\,(\mathcal{A}, f))$, with $\bar{\sigma} = \sigma_1\sigma_2 \cdots \sigma_n \dots$ and $\bar{\tau} = \tau_1\tau_2 \cdots \tau_n \dots$. We will prove that $\theta \in L^\omega_f(\mathcal{A})$. Let us note $t_i = \tau_i - \tau_{i-1}$ for all $1 \leq i \leq |\theta|$, assuming $\tau_0 = 0$. We have an accepting run of $\mathsf{MHTS}\,(\mathcal{A}, f)$ on θ, say π: $s_0 \overset{t_1}{\rightsquigarrow}^{\mathsf{MH}} s_1 \overset{\sigma_1}{\to}^{\mathsf{MH}} s_2 \overset{t_2}{\rightsquigarrow}^{\mathsf{MH}} s_3 \overset{\sigma_2}{\to}^{\mathsf{MH}} \dots \overset{t_i}{\rightsquigarrow}^{\mathsf{MH}} s_{2i-1} \overset{\sigma_i}{\to}^{\mathsf{MH}} s_i \dots$. We must prove that there is an accepting f-run of $\mathcal{A}$ on θ, say π': $C_0 \overset{t_1}{\rightsquigarrow} C_1 \overset{\sigma_1}{\longrightarrow}_f$

[5] once the minimal models of the definition of $\overset{\sigma_{i+1}}{\to}$ are chosen, it is easy to see there exists a unique possible choice for the values of the m_k of $s_{2(i+1)}$.

$C_2 \overset{t_2}{\rightsquigarrow} C_3 \xrightarrow{\sigma_2}_f \dots \overset{t_i}{\rightsquigarrow} C_{2i-1} \xrightarrow{\sigma_i}_f C_{2i} \dots$. We construct π' by induction, proving additionally that the two following properties hold for $j \geq 0$:

($\star 2j$) $C_{2j} = \{(\ell_k, I_k)_{k\in K}\}$ iff $s_{2j} = \{(\ell_k, I_k, m_k)_{k\in K} \mid m_k \in \{\top, \bot\}\}$;
($*2j$) if $s_{2j} \in \alpha$, then, a location of F occurs on all the branches of π' between the last configuration $C_{2j'}$ such that $s_{2j'} \in \alpha$ (or, failing that, between C_0) and C_{2j}.

Basis: We know that $s_0 = \{(\ell_0, [0,0], \bot)\}$ and $C_0 = (\ell_0, [0,0])$. ($\star 0$) and ($*0$) are trivially verified.
Induction: Suppose that we constructed π' until C_{2i} and that ($\star 2j$) and ($*2j$) are verified $\forall 0 \leq j \leq i$. We will construct π' until $C_{2(i+1)}$ in way ($\star 2(i+1)$) and ($*2(i+1)$) will still hold.

First, we must construct C_{2i+1} such that $C_{2i} \overset{t_{i+1}}{\rightsquigarrow} C_{2i+1}$. We know that $s_{2i} \overset{t_{i+1}}{\rightsquigarrow}^{\mathsf{MH}} s_{2i+1}$. Suppose $s_{2i} = \{(\ell_k, I_k, m_k)_{k\in K}\}$, as ($\star 2i$) is verified by hypothesis, $C_{2i} = \{(\ell_k, I_k)_{k\in K}\}$. We must choose C_{2i+1} to be $C_{2i+1} = \{(\ell_k, I_k + t_{i+1})_{k\in K}\}$. As $s_{2i} \overset{t_{i+1}}{\rightsquigarrow}^{\mathsf{MH}} s_{2i+1}$, $s_{2i+1} = \{(\ell_k, I_k + t_{i+1}, m_k)_{k\in K}\}$. Remark that the following property holds:

($\star 2i+1$) $C_{2i+1} = \{(\ell_k, I_k)_{k\in K}\}$ iff $s_{2i+1} = \{(\ell_k, I_k, m_k)_{k\in K} \mid m_k \in \{\top, \bot\}\}$.

Secondly, we must construct $C_{2(i+1)}$ such that $C_{2i+1} \xrightarrow{\sigma_{i+1}}_f C_{2(i+1)}$. We know that $s_{2i+1} \xrightarrow{\sigma_{i+1}}^{\mathsf{MH}} s_{2(i+1)}$. Suppose $s_{2i+1} = \{(\ell_k, I_k, m_k)_{k\in K}\}$, as ($\star 2i+1$) is verified, $C_i = \{(\ell_k, I_k)_{k\in K}\}$. Let us suppose further suppose that $s_{i+1} = \{(\ell_{k'}, I_{k'}, m_{k'})_{k'\in K'}\}$. We construct $C_{2(i+1)} = \{(\ell_{k'}, I_{k'})_{k'\in K'}\}$ so that ($\star 2(i+1)$) holds. Remark that, by definition of $\rightarrow^{\mathsf{MH}}$, as $s_{2i+1} \xrightarrow{\sigma_{i+1}}^{\mathsf{MH}} s_{2(i+1)}$, $C_{2i+1} \xrightarrow{\sigma_{i+1}}_f C_{2(i+1)}$, what we wanted. It stays to prove that ($*2(i+1)$) is satisfied. Suppose that $s_{2(i+1)} \in \alpha$ (i.e. $\forall k' \in K'$, $m_{k'} = 1$). We must prove that, between the last configuration $C_{2j'}$ such that $s_{2j'} \in \alpha$ (or, failing that, between C_0) and $C_{2(i+1)}$, a location of F occurs on all the branches of π'. Let $\beta = \beta_0\beta_1\beta_2 \dots \beta_{2j'} \dots \beta_{2j} \dots \beta_{2(i+1)} \dots$ be a branch of π' and suppose by contradiction that $\forall j' \leq j \leq i+1$, β_{2j} has not its location in F. As $s_{j2'} \in \alpha$, all the third components of $s_{2j'}$ are replaced by $\bot$ (likewise, by definition of s_0, its unique trio has $\bot$ as last component) before evolving reading $\sigma_{j'+1}, \sigma_{j'+2}, \dots, \sigma_{i+1}$ thanks to the rules in (i) in the definition of $\rightarrow^{\mathsf{MH}}$. But, observing rules in (i), when a trio has $\bot$ as last component, it can only evolve to a trio with $\top$ as last component if case (2) it satisfied, what is impossible along β. This contradicts the fact that $s_{2(i+1)} \in \alpha$.
To end the proof, we must show that π' is accepting. The previous induction proves that ($*2j$) holds for all $j \geq 0$. As π is accepting, we know that $s_{2j} \in \alpha$ for infinitely many j and so, between any two successive such j's all the branches of π' saw F. We conclude that all the branches of π' saw F infinitely often, what proves that π' is accepting. □

Büchi timed automata for MITL formulas. Building on this formalisation, one can define a timed automaton with Büchi acceptance condition $\mathcal{B}_\varphi$

$= (\Sigma, B, b_0, X, F^{\mathcal{B}}, \delta^{\mathcal{B}})$ that simulates $\mathsf{MHTS}\left(\mathcal{A}_\varphi, f^\star_\varphi\right)$, and thus accepts $[\![\varphi]\!]$, for every MITL formula φ.

A Büchi timed automaton (TA) is a tuple $\mathcal{B} = (\Sigma, B, b_0, X, F^{\mathcal{B}}, \delta^{\mathcal{B}})$, where Σ is a finite *alphabet*, B is a finite set of *locations*, $b_0 \in B$ is the *initial location*, X is a finite set of *clocks*, $F^{\mathcal{B}} \subseteq B$ is the set of *accepting locations*, and $\delta^{\mathcal{B}} \subseteq B \times \Sigma \times \mathcal{G}(X) \times 2^X \times B$ is a finite set of *transitions*, where $\mathcal{G}(X)$ denotes the set of *guards on* X, i.e. the set of all finite conjunctions of *clock constraints* on clocks from X. For a transition (b, σ, g, r, b'), we say that g is its *guard*, and r its *reset*. A *configuration* of a TA is a pair (b, v), where $v : X \mapsto \mathbb{R}^+$ is a *valuation* of the clocks in X. We denote by $\mathrm{Config}\,(\mathcal{B})$ the set of all configurations of $\mathcal{B}$. For all $t \in \mathbb{R}^+$, we have (time successor) $(b, v) \overset{t}{\rightsquigarrow} (b', v')$ iff $b = b'$ and $v' = v + t$ where $v + t$ is the valuation s.t. for all $x \in X$: $(v + t)(x) = v(x) + t$. For all $\sigma \in \Sigma$, we have (discrete successor) $(b, v) \xrightarrow{\sigma} (b', v')$ iff there is $(b, \sigma, g, r, b') \in \delta$ s.t. $v \models g$, for all $x \in r$: $v'(x) = 0$ and for all $x \in X \setminus r$: $v'(x) = v(x)$. We write $(b, v) \xrightarrow{t,\sigma} (b', v')$ iff there is $(b'', v'') \in \mathrm{Config}\,(\mathcal{B})$ s.t. $(b, v) \overset{t}{\rightsquigarrow} (b'', v'') \xrightarrow{\sigma} (b', v')$. Let $\theta = (\bar{\sigma}, \bar{\tau})$ be a timed word with $\bar{\sigma} = \sigma_1\sigma_2\cdots\sigma_n\ldots$ and $\bar{\tau} = \tau_1\tau_2\cdots\tau_n\ldots$. Let us note $t_i = \tau_i - \tau_{i-1}$ for all $i \geq 1$, assuming $\tau_0 = 0$. A *run* of $\mathcal{B}$ on θ is an infinite sequence of successor steps that is labelled by θ, i.e. a sequence of the form: $(b_0, v_0) \xrightarrow{t_1,\sigma_1} (b_1, v_1) \xrightarrow{t_2,\sigma_2} (b_2, v_2) \xrightarrow{t_3,\sigma_3} (b_3, v_3)\ldots$, where v_0 assigns 0 to all clocks. Such a run of $\mathcal{B}$ is accepting iff there is infinitely many (b_i, v_i) with an accepting b_i. We say that a timed word θ is accepted by $\mathcal{B}$ iff there exists an accepting run of $\mathcal{B}$ on θ. We denote by $L^\omega(\mathcal{B})$ the *language* of $\mathcal{B}$, i.e. the set of infinite timed words accepted by $\mathcal{B}$.

For an MITL formula φ, we construct $\mathcal{B}_\varphi = (\Sigma, B, b_0, X, F^{\mathcal{B}}, \delta^{\mathcal{B}})$ as follows. Locations of $\mathcal{B}_\varphi$ associate with each location ℓ of $\mathcal{A}_\varphi$ a sequence of triples (x, y, m), where x and y are clocks that store the infimum and supremum of an interval respectively, and m is a Miyano-Hayashi marker. Formally, for a set of clocks X, we let $\mathsf{loc}(X)$ be the set of functions S that associate with each $\ell \in L$ a finite sequence $(x_1, y_1, m_1), \ldots, (x_n, y_n, m_n)$ where, for $1 \leq i \leq n$, $m_i \in \{\top, \bot\}$ and (x_i, y_i) is a pair of clocks from X s.t. each clock only occurs once in $S(L)$. Then:

- X is the set of clocks of $\mathcal{B}_\varphi$ s.t. $|X| = M(\varphi)$;
- $B = \{S \in \mathsf{loc}(X)\}$ is the set of locations of $\mathcal{B}_\varphi$. Thus, a configuration (S, v) of $\mathcal{B}_\varphi$ (where S is the location and v the valuation of the clocks X) encodes the labeled configuration $C = \{(\ell, [v(x), v(y)], m) \mid (x, y, m) \in S(\ell)\}$;
- $b_0 = S_0$ is the initial location of $\mathcal{B}_\varphi$ and is s.t. $\forall \ell \in L \setminus \{\ell_0\}$, $S_0(\ell) = \emptyset$, and $S_0(\ell_0) = (x, y, \bot)$, where x and y are two clocks arbitrarily chosen from X;
- $F^{\mathcal{B}} = \{S \in B \,|\, (x, y, m) \in S(L) \Rightarrow m = \top\}$ is the set of final locations of $\mathcal{B}_\varphi$.

Finally, we must define the set of transitions $\delta^{\mathcal{B}}$ to let $\mathcal{B}_\varphi$ simulate the executions of $\mathcal{A}_\varphi$. First, we observe that, for each location $\ell \in L$, for each $\sigma \in \Sigma$, all *arcs* in δ are either of the form $(\ell, \sigma, \mathit{true})$ or $(\ell, \sigma, \mathit{false})$ or of the form $\big(\ell, \sigma, \ell \wedge x.(\ell_1 \wedge \cdots \wedge \ell_k) \wedge g\big)$ or of the form $\big(\ell, \sigma, x.(\ell_1 \wedge \cdots \wedge \ell_k) \wedge g\big)$, where g is *guard* on x, i.e. a finite conjunction of clock constraints on x. Let $S \in B$ be a location of $\mathcal{B}_\varphi$, $\ell \in L$, $\sigma \in \Sigma$ be a letter and (x, y, m) be a 3-tuple occurring in $S(\ell)$. Let us

associate to this 3-tuple an arc a of δ of the form (ℓ, σ, γ). Then, we associate to a a *guard* $\mathsf{guard}\,(a)$, and two sets $\mathsf{reset}\,(a)$ and $\mathsf{loop}\,(a)$, defined as follows:

- if $\gamma \in \{true, false\}$, then, $\mathsf{guard}\,(a) = \gamma$ and $\mathsf{reset}\,(a) = \mathsf{loop}\,(a) = \emptyset$.
- if γ is of the form $x.(\ell_1 \wedge \cdots \wedge \ell_k) \wedge g$, then $\mathsf{guard}\,(a) = g$, $\mathsf{reset}\,(a) = \{\ell_1, \ldots, \ell_k\}$ and $\mathsf{loop}\,(a) = \emptyset$.
- if γ is of the form $\ell \wedge x.(\ell_1 \wedge \cdots \wedge \ell_k) \wedge g$, then $\mathsf{guard}\,(a) = g$, $\mathsf{reset}\,(a) = \{\ell_1, \ldots, \ell_k\}$ and $\mathsf{loop}\,(a) = \{(x, y)\}$.

Thanks to those definitions, we can now define $\delta^{\mathcal{B}}$. Let S^{Δ} be a location in B. We want that to encounter an accepting location of $\mathcal{B}_\varphi$ in a run corresponds to the occurrence of a location of F on all branches of the corresponding run of $\mathcal{A}_\varphi$. When such an accepting location of $\mathcal{B}_\varphi$ is encountered, we need to start again the marking: the following definition of S reflects this need. If $S^{\Delta} \in F^{\mathcal{B}}$, we let S to be such that: $\forall \ell \in L$, $S(\ell) = \{(x, y, \bot) \mid (x, y, \top) \in S^{\Delta}(\ell)\}$. If $S^{\Delta} \notin F^{\mathcal{B}}$, we let $S = S^{\Delta}$. We assume:

$$\{(\ell_1, x_1, y_1, m_1), \ldots, (\ell_k, x_k, y_k, m_k)\} = \{(\ell, x, y, m) \mid (x, y, m) \in S(\ell) \cup \bar{S}(\ell)\}.$$

Then, $(S^{\Delta}, \sigma, g, r, S') \in \delta^{\mathcal{B}}$ iff there is a set $A = \{(a_i)_{i=1}^k\}$:

- For all $i \in \{1, \ldots, k\}$: a_i is an arc of δ of the form (ℓ, σ, γ_i) associated to $(x_i, y_i, m_i) \in S(\ell_i)$.
- For each $\ell \in L \setminus F$, we let $S^\star(\ell) = (x_1^\star, y_1^\star, m_1^\star)(x_2^\star, y_2^\star, m_2^\star) \cdots (x_n^\star, y_n^\star, m_n^\star)$ be obtained from $S(\ell)$ by deleting all the trios (x, y, m) such that $(x, y) \notin \bigcup_{i=1}^k \mathsf{loop}\,(a_i)$. For each $\ell \in L \cap F$, we let $S^\star(\ell) = (x_1^\star, y_1^\star, \top)(x_2^\star, y_2^\star, \top) \cdots (x_n^\star, y_n^\star, \top)$ be obtained from $S(\ell)$ by deleting all the trios (x, y, m) such that $(x, y) \notin \bigcup_{i=1}^k \mathsf{loop}\,(a_i)$. Then, for all $\ell \in L$:

  1. *if* $\ell \notin \bigcup_{i=1}^k \mathsf{reset}\,(a_i)$*:*

     $S'(\ell) = S^\star(\ell)$

  2. *else, if* $\ell \in F$ *(in particular, for* $1 \leq i \leq n$, $m_i^\star = \top$*):*

     $S'(\ell) \in \big\{(x, y, \top) \cdot S^\star(\ell)\ ,\ (x, y_1^\star, m_1^\star)(x_2^\star, y_2^\star, m_2^\star) \cdots (x_n^\star, y_n^\star, m_n^\star)\big\}$

  3. *else, if* $\ell \in \bigcup\limits_{\substack{i \in \{1,\ldots,k\} \\ m_i = 0}} \mathsf{reset}\,(a_i)$*:*

     $S'(\ell) \in \big\{(x, y, \bot) \cdot S^\star(\ell)\ ,\ (x, y_1^\star, \bot)(x_2^\star, y_2^\star, m_2^\star) \cdots (x_n^\star, y_n^\star, m_n^\star)\big\}$

  4. *else* (i.e. $\ell \notin \bigcup\limits_{\substack{i \in \{1,\ldots,k\} \\ m_i = 0}} \mathsf{reset}\,(a_i)$ and $\ell \in \bigcup\limits_{\substack{i \in \{1,\ldots,k\} \\ m_i = 1}} \mathsf{reset}\,(a_i)$)

     $S'(\ell) \in \big\{(x, y, \top) \cdot S^\star(\ell)\ ,\ (x, y_1^\star, m_1^\star)(x_2^\star, y_2^\star, m_2^\star) \cdots (x_n^\star, y_n^\star, m_n^\star)\big\}$

  When $S'(\ell) = (x, y, \bot) \cdot S^\star(\ell)$ or $(x, y, \top) \cdot \bar{S}^\star(\ell)$, we let $R_\ell = \{x, y\}$; when $S'(\ell) = (x, y_1^\star, \bot)(x_2^\star, y_2^\star, m_2^\star) \cdots (x_n^\star, y_n^\star, m_n^\star)$ or $(x, y_1^\star, m_1^\star)(x_2^\star, y_2^\star, m_2^\star) \cdots (x_n^\star, y_n^\star, m_n^\star)$, we let $R_\ell = \{x\}$; and we let $R_\ell = \emptyset$ otherwise.

– $g = \bigwedge_{1 \leq i \leq k} (\mathsf{guard}\,(a_i)\,[x/x_i] \wedge \mathsf{guard}\,(a_i)\,[x/y_i])$.
– $r = \cup_{\ell \in L} R_\ell$.

Theorem 17. $L^\omega(\mathcal{B}_\varphi) = L^\omega_{f^\star_\varphi}(\mathcal{A}_\varphi)$.

Proof. To prove this, we will show that the transition system $S_{\mathcal{B}_\varphi} = (\mathrm{Config}\,(\mathcal{B}_\varphi), \rightsquigarrow, \longrightarrow)$ induced by $\mathcal{B}_\varphi$ in the classical semantics is $\mathsf{MHTS}\left(\mathcal{A}_\varphi, f^\star_\varphi\right)$ in which $(S, v) \in \mathrm{Config}\,(\mathcal{B}_\varphi)$ corresponds to $\{(\ell, [v(x), v(y)], m) \mid (x, y, m) \in S(\ell)\}$. It is easy to see that the initial configuration of $\mathcal{B}_\varphi$, (S_0, v_0), where for all $x \in X$, $v_0(x) = 0$ corresponds to the initial state s_0 of $\mathsf{MHTS}\left(\mathcal{A}_\varphi, f^\star_\varphi\right)$. Now, suppose that we reached a configuration $(S, v) \in \mathrm{Config}\,(\mathcal{B}_\varphi)$ corresponding to the state T of $\mathsf{MHTS}\left(\mathcal{A}_\varphi, f^\star_\varphi\right)$, i.e. $T = \{(\ell, [v(x), v(y)], m) \mid (x, y, m) \in S(\ell)\}$.

Timed transition: let $t \in \mathbb{R}$, $(S, v) \overset{t}{\rightsquigarrow} (S, v+t)$ while $T = \bigcup_{\ell \in L} \{(\ell, [v(x), v(y)], m) | (x, y, m) \in S(\ell)\} \overset{t}{\rightsquigarrow}^{\mathsf{MH}} \bigcup_{\ell \in L} \{(\ell, [v(x), v(y)] + t, m) \mid (x, y, m) \in S(\ell)\}$: these two images correspond.

Discrete transition:

From $S_{\mathcal{B}_\varphi}$ to $\mathcal{S}_{\mathcal{A}, f^\star_\varphi}$: Suppose that $(S^\Delta, v) \xrightarrow{\sigma} (S', v')$ and that T^Δ corresponds to (S^Δ, v), i.e. $T^\Delta = \bigcup_{\ell \in L} \{(\ell, [v(x), v(y)], m) \mid (x, y, m) \in S^\Delta(\ell)\}$. We will show that there exists T' such that $T^\Delta \xrightarrow{\sigma} T'$ in $\mathcal{S}_{\mathcal{A}, f^\star_\varphi}$ and (S', v') and T' correspond. As, $(S^\Delta, v) \xrightarrow{\sigma} (S', v')$, if $S^\Delta \in F^{\mathcal{B}}$, S^Δ is turned to $S = \{(x, y, \bot) \mid (x, y, \top) \in S(L)\}$. In this case, $T^\Delta \in \alpha$ and it is turned to $T = \bigcup_{\ell \in L} \{(\ell, [v(x), v(y)], \bot) \mid (x, y, \top) \in S(\ell)\}$ which corresponds to (S, v). If $S^\Delta \notin F^{\mathcal{B}}$, $T^\Delta \notin \alpha$ and we let $S = S^\Delta$ and $T = T^\Delta$. Then, for all $\ell \in L$, an arc $a_{(x,y)}$ must have been associated to each $(x, y, m) \in S(\ell)$. $S^\star$ is then defined in way each (x, y, m) associated to an arc that loop is still associated to the same location while the others disappear. In the last 4 cases (1., 2., 3. and 4.) all the locations to which an arc goes without looping (characterized by the fact that there is a reset through this location in our automata for MITL formula $\mathcal{A}_\varphi$) are considered: either two new reset (!) clocks are associated to these locations, or a new reset (!) clock replace the clock $x^\star_1$ of the first pair of clocks associated to this location (and so, to the smallest represented interval). Whatever is the case thanks to which (S', v') has been formed, taking the corresponding arc of $\mathcal{B}_\varphi$, each of the intervals (x, y) of S must have satisfied the clock constraints $\mathsf{guard}\left(a_{(x,y)}\right)[x/x_i] \wedge \mathsf{guard}\left(a_{(x,y)}\right)[x/y_i]$, what corresponds to the fact that the whole interval $[v(x), v(y)]$ satisfies $\mathsf{guard}\left(a_{(x,y)}\right)[x/x_i]$ (because this guard is an interval and is so convex). Moreover, v' is obtained from v by associating $\bot$ to each clock reset in the case 1., 2., 3. or 4. used to create (S', v'), and by associating $v(x)$ to all other clock x. This treatment of (S, v) to obtain (S', v') exactly correspond to the fact that $\bigcup_{\ell \in L} \{(\ell, [v(x), v(y)]) | (x, y, m) \in S(\ell)\} \xrightarrow{\sigma}_{f^\star_\varphi} \bigcup_{\ell \in L} \{(\ell, [v'(x), v'(y)]) | (x, y, m) \in S'(\ell)\}$ thanks to the minimal models obtained following arc $a_{(x,y)}$ from each $(\ell, [v(x), v(y)])$ ($f^\star_\varphi$ is applied to the result in way a merging is effectuated iff clock $x^\star_1$ is replaced by a new reset clock from S to S'). We so take $T' = \bigcup_{\ell \in L} \{(\ell, [v'(x), v'(y)], m) \mid (x, y, m) \in S'(\ell)\}$. It is not difficult to see that, whatever is the case thanks to which (S', v') has been formed, the marking of pairs of clock of S' enables T' to satisfy conditions (b) and (c) of the

definition of the transition $\longrightarrow$ of $\mathcal{S}_{\mathcal{A},f^\star_\varphi}$. We so have $T^\Delta \xrightarrow{\sigma} T'$ in $\mathcal{S}_{\mathcal{A},f^\star_\varphi}$ and (S',v') and T' correspond.

From $\mathcal{S}_{\mathcal{A},f^\star_\varphi}$ to $S_{\mathcal{B}_\varphi}$: Suppose that $T^\Delta \xrightarrow{\sigma} T'$ in $\mathcal{S}_{\mathcal{A},f^\star_\varphi}$ and that (S^Δ, v) and T^Δ correspond, i.e. $T^\Delta = \bigcup_{\ell\in L}\{(\ell,[v(x),v(y)],m) \mid (x,y,m) \in S^\Delta(\ell)\}$. We will show that there exists (S',v') such that $(S^\Delta,v) \xrightarrow{\sigma} (S',v')$. As $T^\Delta \xrightarrow{\sigma} T'$, if $T^\Delta \in \alpha$, T^Δ is turned to $T = \bigcup_{\ell\in L}\{(\ell,[v(x),v(y)],\bot) \mid (x,y,\top) \in S(\ell)\}$. In this case, $S^\Delta \in F^{\mathcal{B}}$ and is turned to $S = \{(x,y,\bot) \mid (x,y,\top) \in S(L)\}$, so that (S,v) corresponds to T. If $T^\Delta \notin \alpha$, $S^\Delta \notin F^{\mathcal{B}}$ and we let $T = T^\Delta$ and $S = S^\Delta$. Then, $T \xrightarrow{\sigma} T'$ and the fact that T and (S,v) correspond imply that $\bigcup_{\ell\in L}\{(\ell,[v(x),v(y)]) \mid (x,y,m) \in S(\ell)\} \xrightarrow{\sigma}_{f^\star_\varphi} \{(\ell,I) \mid (\ell,I,m) \in T'\}$. It means an arc $a_{(x,y)}$ has been chosen for each (x,y) to create a minimal model of $\delta(\ell,\sigma)$ wrt $[v(x),v(y)]$. We take (S',v') to be the **unique** successor of (S,v) in $S_{\mathcal{B}_\varphi}$ obtained associating to each (x,y,m) the arc $a_{(x,y)}$ and such that R_ℓ is a singleton iff the application of $f^\star_\varphi$ merged the new interval created in location ℓ with the previous smallest one (it is not difficult to see that we well obtain a **unique** successor this way observing the definition of $\delta^{\mathcal{B}}$). As detailed in the section "From $S_{\mathcal{B}_\varphi}$ to $\mathcal{S}_{\mathcal{A},f^\star_\varphi}$" of this proof, (S',v') will be such that $\bigcup_{\ell\in L}\{(\ell,[v(x),v(y)]) \mid (x,y,m) \in S(\ell)\} \xrightarrow{\sigma}_{f^\star_\varphi} \bigcup_{\ell\in L}\{(\ell,[v'(x),v'(y)]) \mid (x,y,m) \in S'(\ell)\}$ thanks to the minimal models obtained following arc $a_{(x,y)}$ from each $(\ell,[v(x),v(y)])$ ($f^\star_\varphi$ is applied to the result in way a merging is effectuated iff clock $x^\star_1$ is replaced by a new reset clock from S to S'). In other words, T' and (S',v') correspond, thanks to the fact that the unique possible marking present on the third components of elements of T' will be the same than the obtained marking of elements of (S',v') (it is easy to see, observing all the cases thanks to which T' has been constructed). □

5 MITL model-checking and satisfiability with TOCATA

In this section, we fix an MITL formula φ and a TA $\mathcal{B} = (\Sigma, B, b_0, X, \delta^{\mathcal{B}}, F^{\mathcal{B}})$, and we consider the two following problems: (i) the *model-checking problem* asks whether $L(\mathcal{B}) \subseteq [\![\varphi]\!]$; (ii) the *satisfiability problem* asks whether $[\![\varphi]\!] \neq \emptyset$. The construction of the TA $\mathcal{B}_{\neg\varphi}$ from φ of the previous section allows to solve those problems using classical algorithms [1]. Unfortunately, building $\mathcal{B}_{\neg\varphi}$ can be prohibitive in practice. To mitigate this difficulty, we present an efficient *on-the-fly* algorithm to perform MITL model-checking, which has as input the TA $\mathcal{B}$ and the TOCATA $\mathcal{A}_{\neg\varphi}$ (whose size is linear in the size of φ). It consists in exploring symbolically the state space of the timed transition system $\mathcal{S}_{\mathcal{B},\neg\varphi}$ which is obtained by first taking the synchronous product of $\mathsf{TTS}\left(\mathcal{A}_{\neg\varphi}, f^\star_{\neg\varphi}\right)$ and the transition system[6] $\mathsf{TTS}\left(\mathcal{B}\right)$ of $\mathcal{B}$ [1], and then associating Miyano-Hayashi markers with its states, by adapting the construction of $\mathsf{MHTS}\left(\mathcal{A}, f\right)$ given above to cope with the configurations of $\mathcal{B}$. Namely, we associate a Miyano-Hayashi marker with the configurations of $\mathcal{B}$ too, and a state of $\mathcal{S}_{\mathcal{B},\neg\varphi}$ is accepting iff all markers (including the one on the $\mathcal{B}$ configuration) are $\top$. Obviously, $L(\mathcal{B}) \subseteq [\![\varphi]\!]$ iff $\mathcal{S}_{\mathcal{B},\neg\varphi}$

[6] See appendix **??** for a formal definition

has no accepting run (i.e., no run visiting accepting states infinitely often). Symmetrically, we can solve the *satisfiability problem* by looking for accepting run in $\mathsf{MHTS}\left(\mathcal{A}_\varphi, f^\star_\varphi\right)$ (since the techniques are similar for model-checking and satisfiability, we will only detail the former in this section).

Formally, we define the transition system $\mathcal{S}_{\mathcal{B},\neg\varphi} = (\Sigma, S, s_0, \rightsquigarrow, \rightarrow, \alpha)$ where: S is the set of elements of the form $\{(\ell_k, I_k, m_k)_{k\in K}\} \cup \{(b, v, m_\mathcal{B})\}$ where $\{(\ell_k, I_k)_{k\in K}\}$ is a configuration of $\mathcal{A}_{\neg\varphi}$, (b, v) is a state of $\mathcal{B}$, $m_\mathcal{B} \in \{\top, \bot\}$ and $\forall k \in K, m_k \in \{\top, \bot\}$, $s_0 = \{(\ell_0, [0,0], \bot), (b_0, v_0, m)\}$, where $m = \top$ iff $b_0 \in F^\mathcal{B}$; α contains all the elements of S of the form $\{(\ell_k, I_k, \top)_{k\in K}\} \cup \{(b, v, \top)\}$.

For $t \in \mathbb{R}$ and $s, s' \in S$, supposing $s = \{(\ell_k, I_k, m_k)_{k\in K}\} \cup \{(b, v, m_\mathcal{B})\}$, we have $s \overset{t}{\rightsquigarrow} s'$ iff $s' = \{(\ell_k, I_k + t, m_k)_{k\in K}\} \cup \{(b, v + t, m_\mathcal{B})\}$. $\rightsquigarrow = \bigcup_{t\in\mathbb{R}} \overset{t}{\rightsquigarrow}$.

"$\rightarrow$" is defined in way that (1) a transition between two states of $\mathcal{S}_{\mathcal{B},\neg\varphi}$ corresponds to a transition between the configurations of $\mathcal{A}_{\neg\varphi}$ they contain, (2) a transition between the states of $\mathcal{B}$ they contain and (3) the markers of the third component of states of S are kept updated. Formally, $\rightarrow = \bigcup_{\sigma\in\Sigma} \overset{\sigma}{\rightarrow}$, where:

(i) For $s \in S \setminus \alpha$ and $s' \in S$, supposing $s = \{(\ell_k, I_k, m_k)_{k\in K}\} \cup \{(b, v, m_\mathcal{B})\}$ and $s' = \{(\ell_{k'}, I_{k'}, m_{k'})_{k'\in K'}\} \cup \{(b', v', m'_\mathcal{B})\}$, $s \overset{\sigma}{\rightarrow} s'$ iff
   (a) $\{(\ell_k, I_k)_{k\in K}\} \xrightarrow{\sigma}_{f^\star_{\neg\varphi}} \{(\ell_{k'}, I_{k'})_{k'\in K'}\}$ in $\mathcal{A}_{\neg\varphi}$, i.e $\{(\ell_{k'}, I_{k'})_{k'\in K'}\} \in f^\star_{\neg\varphi}(\mathsf{Succ}(\{(\ell_k, I_k)_{k\in K}\}, \sigma))$, and $(b, v) \xrightarrow{\sigma} (b', v')$ in $\mathcal{B}$;
   (b) $\forall k' \in K'$: $(\ell_{k'} \in F \Rightarrow m_{k'} = \top)$;
   (c) $\forall k' \in K'$ with $\ell_{k'} \notin F$: if $\exists k \in K$ s.t. $(\ell_k, I_k, \bot) \in s$ and $(\ell_{k'}, I_{k'}) \in \mathsf{dest}(\mathsf{conf}(s), \mathsf{conf}(s'), (\ell_k, I_k, \bot))$, we have $m_{k'} = \bot$; otherwise, $m_{k'} = \top$;
   (d) $m'_\mathcal{B} = \top$ iff $m_\mathcal{B} = \top$ or $b' \in F^\mathcal{B}$.

(ii) For $s \in \alpha$ and $s' \in S$, supposing $s = \{(\ell_k, I_k, \top)_{k\in K}\} \cup \{(b, v, \top)\}$, $s \overset{\sigma}{\rightarrow} s'$ iff $\{(\ell_k, I_k, \bot)_{k\in K}\} \cup \{(b, v, \bot)\} \overset{\sigma}{\rightarrow} s'$ according to the rules in (i).

Defining $f^\star_{\neg\varphi}$ on configurations of $\mathcal{B}\times\mathcal{A}_{\neg\varphi}$ in way $f^\star_{\neg\varphi}(\{(\ell_k, I_k)_{k\in K}\}\cup\{(b, v)\}) = F^M(\{(\ell_k, I_k)_{k\in K}\})\cup\{(b, v)\}$, with $M = \max\{2\times|L|, M(\varphi)\}$, the following proposition can be proven similarly as Proposition 16.

Proposition 18. *For every MITL formula φ, the associated $\mathcal{A}_{\neg\varphi}$ and $f^\star_{\neg\varphi}$, and for every Büchi timed automaton $\mathcal{B}$: $L^\omega(\mathcal{S}_{\mathcal{B},\neg\varphi}) = L^\omega_{f^\star_{\neg\varphi}}(\mathcal{B} \times \mathcal{A}_{\neg\varphi})$.*

Region-based algorithms Since $\mathcal{S}_{\mathcal{B},\varphi}$, we adapt the region abstraction of [16] in order to cope with: 1. the valuations of the clocks that are now *intervals*; and 2. the Miyano-Hayashi markers. Following the approach of [16] we represent each region by a unique word. In the sequel, we note $c_{\max}$ the maximal constant of automata $\mathcal{B}$ and $\mathcal{A}_{\neg\varphi}$.

Definition 19. *We define the equivalence relation $\sim$ on $\mathbb{R}^+$ by: $u \sim v$ iff either u and $v > c_{\max}$, or u and $v \leq c_{\max}$, $\lceil u\rceil = \lceil v\rceil$ and $\lfloor u\rfloor = \lfloor v\rfloor$. The set of classes of $\sim$, called regions, is $REG = \{\{0\}, (0,1), \{1\}, (1,2), \ldots, (c_{\max} - 1, c_{\max}), \{c_{\max}\}, (c_{\max}, +\infty)\}$. We will note $Reg(v)$ the class of $v \in \mathbb{R}^+$.*

In the sequel, $\forall v \in \mathbb{R}^+$, we note $frac(v)$ the fractionnal part of v. Moreover, we suppose that $\forall v > c_{\max}, frac(v) = 0$.
We can now define an equivalence relation $\equiv$ on S.

Definition 20. *Let s and s' be two states of S such that the configurations of $\mathcal{A}_{\neg\varphi}$ they contain have the same cardinality. We suppose that $s = \{(\ell_k, I_k, m_k)_{k\in K}\} \cup \{(b, v, m_{\mathcal{B}})\}$ and (without loss of generality) $s' = \{(\ell'_k, I'_k, m'_k)_{k\in K}\} \cup \{(b', v', m'_{\mathcal{B}})\}$. Then, we define $s \equiv s'$ iff:*

1. $b = b'$ *and* $\forall k \in K : \ell_k = \ell'_k$; $m_{\mathcal{B}} = m'_{\mathcal{B}}$ *and* $\forall k \in K : m_k = m'_k$,
2. $\forall 1 \leq p \leq n : v(x_p) \sim v'(x_p)$ *and* $\forall k \in K : (\inf(I_k) \sim \inf(I'_k) \wedge \sup(I_k) \sim \sup(I'_k))$,
3. $\forall 1 \leq p, q \leq n : frac(v(x_p)) \bowtie frac(v(x_q))$ *iff* $frac(v'(x_p)) \bowtie frac(v'(x_q))$,
4. $\forall k, k' \in K : frac(\inf(I_k)) \bowtie frac(\inf(I_{k'}))$ *iff* $frac(\inf(I'_{k'})) \bowtie frac(\inf(I'_{k'}))$,
5. $\forall k, k' \in K : frac(\sup(I_k)) \bowtie frac(\sup(I_{k'}))$ *iff* $frac(\sup(I'_k)) \bowtie frac(\sup(I'_{k'}))$,
6. $\forall k, k' \in K : frac(\inf(I_k)) \bowtie frac(\sup(I_{k'}))$ *iff* $frac(\inf(I'_k)) \bowtie frac(\sup(I'_{k'}))$,
7. $\forall k \in K, \forall 1 \leq p \leq n : frac(\inf(I_k)) \bowtie frac(v(x_p))$ *iff* $frac(\inf(I'_k)) \bowtie frac(v'(x_p))$,
8. $\forall k \in K, \forall 1 \leq p \leq n : frac(\sup(I_k)) \bowtie frac(v(x_p))$ *iff* $frac(\sup(I'_k)) \bowtie frac(v'(x_p))$,

where $\bowtie \in \{<, =, >\}$.

In this definition, the same indices have been chosen for the two configurations of $\mathcal{A}_{\neg\varphi}$ in aim to make "correspond" (ℓ_k, I_k) with (ℓ'_k, I'_k). Condition 1. forces corresponding (ℓ_k, I_k, m_k) and (ℓ'_k, I'_k, m'_k), as well as the two states of $\mathcal{B}$, to have (resp.) the same locations and markers. Condition 2. forces intervals of corresponding (ℓ_k, I_k) and (ℓ'_k, I'_k) to have their infima and suprema in the same regions, and values of the same clocks of the two states of $\mathcal{B}$ to be in the same regions. The other conditions forces all the clock values present in s (clock values, values of infimum and supremum of intervals) to present the same fractional part order than the corresponding clock values present in s'.

Proposition 21 (Time-abstract bisimulation). *Let $s_1, s_2 \in S$ such that $s_1 \equiv s_2$. Then, for each transition $s_1 \overset{t}{\rightsquigarrow} z_1$ with $t \in \mathbb{R}^+$ (resp. $s_1 \xrightarrow{\sigma}_{f^\star_{\neg\varphi}} z_1$, with $\sigma \in \Sigma$) and $z_1 \in S$, there exists $t' \in \mathbb{R}^+$ and $z_2 \in S$ such that $s_2 \overset{t'}{\rightsquigarrow} z_2$ (resp. there exists $z_2 \in S$ such that $z_1 \xrightarrow{\sigma}_{f^\star_{\neg\varphi}} z_2$) and $z_1 \equiv z_2$.*

Proof. Let us note $s_1 = \{(\ell_k, I_k, m_k)_{k\in K}\} \cup \{(b, v, m_b)\}$, as $s_1 \equiv s_2$, we can suppose that $s_2 = \{(\ell_k, I'_k, m_k)_{k\in K}\} \cup \{(b, v', m_b)\}$. In the following, we will use the following notation $C = \{(\ell_k, I_k)_{k\in K}\}$, $C' = \{(\ell_k, I'_k)_{k\in K}\}$, $s = \{(b, v)\}$ and $s' = \{(b, v')\}$.
First, suppose that $s_1 \xrightarrow{\sigma}_{f^\star_{\neg\varphi}} z_1$, with $\sigma \in \Sigma$ and $z_1 = \{(\ell_q, J_q, m_q)_{q\in Q}\} \cup \{(r, v^\star, m_r)\}$. Let us note $D = \{(\ell_q, J_q)_{q\in Q}\}$ and $z = (r, v^\star)$. On the first hand, we have that $C \xrightarrow{\sigma}_{f^\star_{\neg\varphi}} D$ and so $D \in f^\star_{\neg\varphi}(E)$, where $E = \bigcup_{k\in K} E_k$ and each E_k is a minimal model of $\delta(\ell_k, \sigma)$ wrt I_k. We recall that $E = \bigcup_{k\in K} A_k[I_k]$ where A_k is

the set of terms of the disjunctive normal form of $\delta(\ell_k, \sigma)$. Let $E' = \bigcup_{k \in K} A_k[I'_k]$, i.e., we use the same arcs from C to E than from C' to E', for each $D' \in f^\star_{\neg\varphi}(E')$, we have that $C' \xrightarrow{\sigma}_{f^\star_{\neg\varphi}} D'$ (because as $s_1 \equiv s_2$, thanks to condition 2. of the definition of "$\equiv$", I_k and I'_k satisfy the same clock constraints). On the other hand, $s_1 \xrightarrow{\sigma}_{f^\star_{\neg\varphi}} z_1$ means that $s \xrightarrow{\sigma} z$: there exists an arc $(b, \sigma, r, c, R) \in \Delta$ such that $v \models c$ and $\forall x_p \in R : v^\star(x_p) = 0$, while $\forall x_p \notin R : v^\star(x_p) = v(x_p)$. Let us note $z' = (r, v'^\star)$ where $v'^\star$ is such that $\forall x_p \in R : v'^\star(x_p) = 0$, while $\forall x_p \notin R : v'^\star(x_p) = v'(x_p)$. We have that $s' \xrightarrow{\sigma} z'$ following the arc (b, σ, r, c, R): as $s_1 \equiv s_2$, the condition 2. of the definition of "$\equiv$" enables $v(x_1), \ldots, v(x_n)$ and $v'(x_1), \ldots, v'(x_n)$ to satisfy the same clock constraints (so that $v' \models c$). Let us note $z_3 = \{(\ell, I, m) | (\ell, I) \in E\} \cup \{(r, v^\star, m_r)\}$ the [7] element of S such that $s \xrightarrow{\sigma} z_3$, and $z_4 = \{(\ell, I, m) | (\ell, I) \in E'\} \cup \{(r, v'^\star, m'_r)\}$ the unique element of S such that $s \xrightarrow{\sigma} z_3$. We will prove that $z_3 \equiv z_4$. Condition 1. of the definition of $\equiv$ is satisfied because C and C' owned the same markers, as well as s and s' (because $s_1 \equiv s_2$) and we chose the same minimal models to go from C to E than from C' to E' and the same arc of $\mathcal{B}$ to go from s to z than from s' to z' (these are the unique parameters for the choice of the markers of z_3 and z_4). We still must prove that conditions 2. to 8. are satisfied. The clock values observed for verifying conditions 2. to 8. are included in $\{v(x_p) | 1 \leq p \leq n\} \cup \{v'(x_p) | 1 \leq p \leq n\} \cup \{inf(I_k) | k \in K\} \cup \{sup(I_k) | k \in K\} \cup \{inf(I'_k) | k \in K\} \cup \{sup(I'_k) | k \in K\} \cup \{0\}$ (as the discrete transitions either let the clocks values unchanged or replace them by 0). However, conditions 2. to 8. were verified on $\{v(x_p) | 1 \leq p \leq n\} \cup \{v'(x_p) | 1 \leq p \leq n\} \cup \{inf(I_k) | k \in K\} \cup \{sup(I_k) | k \in K\} \cup \{inf(I'_k) | k \in K\} \cup \{sup(I'_k) | k \in K\}$ thanks to the fact that $s_1 \equiv s_2$. Moreover if a clock value is replaced by 0 in z or E, the corresponding clock value in z' or E' is also replaced by 0 (which is the smallest possible clock value, so that its comparisons with all other clocks values will enable to verify 3. to 8.). So, $z_3 \equiv z_4$.

Now, as $D \in f^\star_{\neg\varphi}(E)$, we choose $D' \in f^\star_{\neg\varphi}(E')$ such that the intervals of E' grouped to obtain D' correspond to those grouped in E to obtain D. Let us recall that $D = \{(\ell_q, J_q)_{q \in Q}\}$ and let us note $D' = \{(\ell_q, J'_q)_{q \in Q}\}$ Formally, we want D' to be such that: $\forall (\ell_k, I_k), (\ell_{\tilde{k}}, I_{\tilde{k}}) \in C : \big((\ell_q, J_q) \in \mathsf{dest}(C, D, (\ell_k, I_k)) \wedge (\ell_q, J_q) \in \mathsf{dest}(C, D, (\ell_{\tilde{k}}, I_{\tilde{k}}))\big) \Rightarrow \big((\ell_q, J'_q) \in \mathsf{dest}(C', D', (\ell_k, I'_k)) \wedge (\ell_q, J'_q) \in \mathsf{dest}(C', D', (\ell_{\tilde{k}}, I'_{\tilde{k}}))\big)$. We conclude that $z_1 \equiv z_2$ (the argument is the same as that why $z_3 \equiv z_4$).

Secondly, suppose that $s_1 \overset{t}{\rightsquigarrow} z_1$ for a certain $t \in \mathbb{R}^+$. We must prove there exists $t' \in \mathbb{R}^+$ and a configuration z_2 such that $s_2 \overset{t'}{\rightsquigarrow} z_2$ **(*)**. To do that, we first define the "time successor" of an element s_1 of S, noted $next(s_1)$: it is an element of the first equivalence class of $\equiv$ reachable from the class of s_1 (and different from it) letting time elapsing. Then, we prove that $next(s_1) \equiv next(s_2)$. We finally deduce **(*)** from this last result.

Let $\mathrm{V} = \{v(x_i) | 1 \leq p \leq n\} \cup \{v | (\ell_k, I_k) \in C \wedge (v = \inf(I_k) \vee v = sup(I_k))\}$ be

[7] once the minimal models and the arc of $\mathcal{B}$ of the definition of $\xrightarrow{\sigma}$ are chosen, it is easy to see there exists a unique possible choice for the values of the markers of z_3.

the set of clock values present in s and C. We note $\mu = \max\{frac(v) | v \in V\}$. We define d as $\frac{1-\mu}{2}$ if V contains an integer smaller or equal to c_{max}, and $1-\mu$ otherwise. We define the time successor of s_1, noted $next(s_1)$, to be $\{(\ell_k, I_k + d, m_k)_{k\in K}\} \cup \{(b, v+d, m_b)\}$.
We will prove that: as $s_1 \equiv s_2$, we have that $next(s_1) \equiv next(s_2)$ **(**)**. We have that $next(s_1) = \{(\ell_k, I_k + d, m_k)_{k\in K}\} \cup \{(b, v+d, m_b)\}$ for a certain $d > 0$, and $next(s_2) = \{(\ell_k, I'_k + d', m_k)_{k\in K}\} \cup \{(b, v'+d', m_b)\}$ for a certain $d' > 0$. The effect on s_1 is either, if an integer is present among the clocks values, to keep the order of their fractional parts unchanged, either, otherwise, to permute the order of the fractional parts of the clocks values with the largest fractional parts such that they now have a zero fractional part (and so are the clock values with the smallest fractional parts). The effect on s_2 being the same, in the same cases, and the conditions of $s_1 \equiv s_2$ certifying that an integer is present among the clocks values of s_1 iff there is an integer between the clocks values of s_2, conditions 1. to 8. are still verified on $next(s_1)$ and $next(s_2)$: $next(s_1) \equiv next(s_2)$.
Now, as $s_1 \xrightarrow{t} z_1$: it means there exists an $n \geq 0$ such that $z_1 \equiv next^n(s_1)$. As $s_1 \equiv s_2$, we deduce from **(**)** that: $next^n(s_1) \equiv next^n(s_2)$ and so, taking $z_2 = next^n(s_2)$, we verify **(*)** (d' is the sum of the n d's used to recursively compute $next(s_2), next^2(s_2), \ldots, next^n(s_2)$). □

Remark that the size of the configurations of $\mathcal{A}_{\neg\varphi}$ we can encounter in $\mathcal{S}_{\mathcal{B},\neg\varphi}$ is bounded by $M(\neg\varphi)$, thanks to the use of $f^{\star}_{\neg\varphi}$: there is only a finite number of such configurations, and so the number of configurations of $\mathcal{A}_{\neg\varphi}$ we can encounter in $\mathcal{S}_{\mathcal{B},\neg\varphi}$ is finite. Therefore, as the number of regions is also finite, the quotient of $\mathcal{S}_{\mathcal{B},\neg\varphi}$ by $\equiv$ is finite and we can elaborate a model-checking algorithm using it.
Following the approach of [16] we will represent each of these regions by a unique word. Once again, the definition of these words must be adapted. On one hand, they must maintain an additional component corresponding to markers due to the Miyano Hayashi construction. On the other hand, each interval must be split in two parts: one representing the infimum of the interval (its region and the relative value of its fractional part) and the other its supremum. Yet, we must use natural numbers to associate each infimum of interval to the suitable supremum. We encode regions of $\mathcal{S}_{\mathcal{B},\neg\varphi}$ by finite words whose letters are finite sets of tuples of the form (ℓ, r, m, k), where $\ell \in L \cup L^{\mathcal{B}}$, $r \in REG$, $m \in \{\top, \bot\}$ and $0 \leq k \leq M(\varphi)/2$. Here is the definition of the function H that associate to each $s \in S$ the region it is in.
For $s = \{(\ell_k, I_k, m_k)_{k\in K}\} \cup \{(b, v, m)\}$, $H(s) = H_1 H_2 \cdots H_m$ is defined as follows:

1. For each location ℓ, let $C(\ell) = \{(\ell', I, m) \in C \mid \ell' = \ell\}$. Assume $C(\ell) = \{(\ell_1, I_1, m_1), \ldots, (\ell_k, I_k, m_k)\}$, with $I_1 \leq \cdots \leq I_k$. Then, we first build $E_\ell = \{(\ell_i, \inf(I_i), m_i, i), (\ell_i, \sup(I_i), m_i, i) \mid 1 \leq i \leq k\}$.
2. We treat $(\ell^{\mathcal{B}}, v, m)$ symmetrically, and let $E^{\mathcal{B}} = \{(\ell^{\mathcal{B}}, v(x_1), m, 1), \ldots, (\ell^{\mathcal{B}}, v(x_n), m, n)\}$. We let $\mathcal{E} = E^{\mathcal{B}} \cup_{\ell\in L} E_\ell$. That is, all elements in $\mathcal{E}$ are tuples (ℓ, v, m, i), where ℓ is a location (of $\mathcal{A}_\varphi$ or $\mathcal{B}$), v is a real value (interval endpoint or clock value), m is a Miyano-Hayashi marker and i is

bookkeeping information that links v to an interval (if ℓ is a location of $\mathcal{A}_\varphi$), or to a clock (ℓ is a location of $\mathcal{B}$).

3. We partition $\mathcal{E}$ into $\mathcal{E}_1, \ldots, \mathcal{E}_m$ s.t. each $\mathcal{E}_i$ contains all elements from $\mathcal{E}$ with the same fractional part to their second component (assuming $frac(u) = 0$ for all $u > \mathrm{c}_{\max}$). We assume the ordering $\mathcal{E}_1$, $\mathcal{E}_2$,..., $\mathcal{E}_m$ reflects the increasing ordering of the fractional parts.
4. For all $1 \leq i \leq m$, we obtain H_i from $\mathcal{E}_i$ by replacing the second component of all elements in $\mathcal{E}_i$ by the region from REG they belong to.

Thus, noting $\max_\ell$ the maximal number of interval that can be present in location $\ell \in L$ of $\mathcal{A}_{\neg\varphi}$ (given by Theorem 14), $H(s)$ will be a finite word over the alphabet $\Lambda = (B \cup L) \times REG \times \{\top, \bot\} \times \{1, 2, \ldots, \max(\max_{\ell \in L}(\max_\ell), n)\}$. We will also view H as a function $H : S \to \Lambda^\star$.

Example 22. Consider a TA $\mathcal{B}$ with 1 clock, let $\mathrm{c}_{\max} = 2$, and let $s = \big\{\{(\ell_1, [0, 1.3], \bot), (\ell_1, [1.8, 2.7], \top)\}, (\ell^{\mathcal{B}}, 1.3, \bot)\big\}$. The first step of the construction yields the set $\mathcal{E} = \{(\ell_1, 0, \bot, 1), (\ell_1, 1.3, \bot, 1), (\ell_1, 1.8, \top, 2), (\ell_1, 2.7, \top, 2), (\ell^{\mathcal{B}}, 1.3, \bot, 1)\}$. Then, we have $H(s) = \big\{(\ell_1, \{0\}, \bot, 1), (\ell_1, (2, +\infty), \top, 2)\big\}$ $\big\{(\ell_1, (1, 2), \bot, 1), (\ell^{\mathcal{B}}, (0, 1), \bot, 1)\big\}\big\{(\ell_1, (1, 2), \top, 2)\big\}$.

Proposition 23. *Let $s, s' \in S$. We have: $s \equiv s'$ iff $H(s) = H(s')$.*

Proof. ($\Rightarrow$) Suppose that $s \equiv s'$, then the order of the fractional parts of all the clock values that s contains is the same than those of all the corresponding clock values that s' contains (see $\equiv$ conditions 3. to 8.). Moreover, their corresponding states have their infima (resp. their suprema, resp. clocks values) in the same region (see $\equiv$ condition 2.) and their locations are the same. The way $H(s)$ and $H(s')$ are constructed, their will be no difference between these two words.
($\Leftarrow$) Suppose $H(s) = H(s')$, then we associate each interval of the configuration of $\mathcal{A}_{\neg\varphi}$ s contains, clearly defined by two 4-tuples in $H(s)$, to the interval of s' that is represented by the two corresponding 4-tuples of $H(s')$. Moreover we associate each value v_i of the clock x_i of $\mathcal{B}$, clearly defined by a certain 4-tuple in $H(s)$, with the value of v_i' of the clock x_i of $\mathcal{B}$ represented in the corresponding 4-tuple of $H(s')$. As $H(s) = H(s')$, conditions 1. and 2. of $\equiv$ are of course verified. The other conditions are also respected thanks to the groupings executed on the elements of $H(s)$ and $H(s')$ to have an increasing order of the fractional parts of the second components (i.e. clock) values. □

Definition 24. *We define:*

$$\mathcal{H} = \mathcal{S}_{\mathcal{B},\neg\varphi}/\equiv = \{H(s) \mid s \in S\}.$$

For all $W^1, W^2 \in \mathcal{H}$ and $\sigma \in \Sigma$ we define $W^1 \xrightarrow{\sigma} W^2$ iff $\forall s^1 \in H^{-1}(W^1)$, $\exists s^2 \in H^{-1}(W^2) : s^1 \xrightarrow{\sigma} s^2$.
For all $W^1, W^2 \in \mathcal{H}$, we define $W^1 \longrightarrow_T W^2$ iff $\forall s^1 \in H^{-1}(W^1)$, $\exists t \in \mathbb{R}$ and $\exists s^2 \in H^{-1}(W^2) : s^1 \overset{t}{\rightsquigarrow} s^2$.

Proposition 25. *Let $W^1, W^2 \in \mathcal{H}$, $\sigma \in \Sigma$ and $t \in \mathbb{R}^+$.*
$W^1 \xrightarrow{\sigma} W^2$ iff $\exists s^1 \in H^{-1}(W^1)$ and $s^2 \in H^{-1}(W^2) : s^1 \xrightarrow{\sigma} s^2$.

Proof. ($\Rightarrow$) Follows directly from Definition 24.
($\Leftarrow$) Suppose that $s^1 \in H^{-1}(W^1), s^2 \in H^{-1}(W^2)$, and that $s^1 \xrightarrow{\sigma} s^2$. Let $s^3 \in H^{-1}(W^1)$, we must prove that $\exists s^4 \in H^{-1}(W^2) : s^3 \xrightarrow{\sigma} s^4$. As $s^3 \in H^{-1}(W^1)$ and $s^1 \in H^{-1}(W^1)$, by Proposition 23, $s^3 \equiv s^1$. As $s^1 \xrightarrow{\sigma} s^2$, Proposition 21 ensures that $\exists s^4 \in H^{-1}(W^2) : s^3 \xrightarrow{\sigma} s^4$.

Definition 26. *Let $\mathcal{W} \subseteq \mathcal{H}$, we define:*

$$Post(\mathcal{W}) := \{W' \in \mathcal{H} \mid \exists \sigma \in \Sigma,\ W \in \mathcal{W} \text{ and } W'' \in \mathcal{H} : W \longrightarrow_T W'' \xrightarrow{\sigma} W'\}.$$

We claim that, for any word $W \in \mathcal{H}$, $Post(W)$ is finite and effectively computable. Let $W \in \mathcal{H}$. The set of all W'' such that $W \longrightarrow_T W''$ is a finite set of words with the same number of 4-tuples than W. We form this set accumulating the words computed recursively as follows, using at each time the last W_{next} obtained (and starting from W):

- if the first letter of W_{next} contains 4-tuples whose second component is $\{0\}$ or $\{1\}$ or ... or $\{c_{\max}\}$,
  1. the 4-tupples of this first letter whose second component is $\{c_{\max}\}$ are replaced by the same 4-tuples in which $\{c_{\max}\}$ is replaced by $(c_{\max}, +\infty)$,
  2. the other 4-tuples of this first letter are deleted from it (if it then becomes empty, it is omitted). A new set of 4-tuples is created as a new second letter: it will contain these same 4-tuples in which the second component is replaced by the immediately following region ($(0,1)$ instead of $\{0\}$, $(1,2)$ instead of $\{1\}$, ..., $(c_{\max}-1, c_{\max})$ instead of $\{c_{\max}-1\}$). The end of the word does not change.

  The following W_{next} is the word created like this ;
- else, the last letter of W_{next} is deleted. Its 4-tuples are modified to create a new set that will contain these same 4-tuples in which the second component is replaced by the immediately following region ($\{1\}$ instead of $(0,1)$, $\{2\}$ instead of $(1,2), \ldots, \{c_{\max}\}$ instead of $(c_{\max}-1, c_{\max})$). This new set is either joined with the first letter of the modified W_{next}, if it contains 4-tuples having $(c_{\max}, +\infty)$ as second components, either added as a new first letter of the modified W_{next} otherwise. The rest of the word does not change. The following W_{next} is the word created like this ;

We stop when we encounter a W_{next} that has a unique letter whose 4-tuples have $(c_{\max}, +\infty)$ as second components.
Then, for each possible W'' we easily find a $s \in S$ such that $H(s) = W''$ (note that the choice of s does not matter thanks to Proposition 25). For all $\sigma \in \Sigma$, it is easy to compute the set of elements s' of S such that $s \xrightarrow{\sigma} s'$. This set is finite and, from each of its elements s', we can get back $H(s')$.
Once we have examined each letter $\sigma \in \Sigma$, for each possible W'', the (finite !) set of all the $H(s')$ found form $Post(W)$.

Definition 27. *Let $\mathcal{W} \subseteq \mathcal{H}$ and $n \in \mathbb{N}_0$, we define: $Post^n(\mathcal{W}) = Post^{n-1}(\mathcal{W})$, with $Post^0(\mathcal{W}) = \mathcal{W}$ and $Post^1(\mathcal{W}) = Post(\mathcal{W})$.*
We define $Post^*(\mathcal{W}) = \bigcup_{n\in\mathbb{N}} Post^n(\mathcal{W})$ and $Post^+(\mathcal{W}) = \bigcup_{n\in\mathbb{N}_0} Post^n(\mathcal{W})$.

Remark that, as $\mathcal{H}$ is finite, $\exists m \in \mathbb{N} : Post^*(\mathcal{W}) = Post^m(\mathcal{W})$.

Let us note $H_0 := H(s_0)$ the word of $\mathcal{H}$ corresponding to the initial state of $\mathcal{S}_{\mathcal{B},\neg\varphi}$. We say that a word W of $\mathcal{H}$ is accepting iff the third components of all the 4-tuples it contains are "1" (such words correspond to accepting states of $\mathcal{S}_{\mathcal{B},\neg\varphi}$). We note $\mathcal{F} \subseteq \mathcal{H}$ the set of accepting words of $\mathcal{H}$.
Here is an (classical) algorithm for the model-checking of MITL in which the reachable words of $\mathcal{H} = \mathcal{S}_{\mathcal{B},\neg\varphi}/\equiv$ are computed "on the fly".

Algorithm 1 ModelCheckingMITL
Input: A TA $\mathcal{B}$ and the ATA $\mathcal{A}_{\neg\varphi}$, for $\varphi \in MITL$.
Output: "true" iff $\mathcal{B} \models \varphi$.

```
C ← ∅
D ← Post*(H_0) ∩ F
while C ≠ D do
    C ← D
    D ← Post+(D) ∩ F
end while
if D = ∅ then
    return true
else
    return false
end if
```

We use the following theorem to prove that this algorithm is correct.

Theorem 28 ([13] - Theorem 2.3.20). *Let $\mathcal{A} =< Q, \Sigma, I, \rightarrow, F >$ be a non-deterministic Büchi automaton.*

$$L(\mathcal{A}) = \emptyset \quad \textit{iff} \quad GFP(\lambda X.Post^+(X) \cap F \cap Post^\star(I)) \ = \ \emptyset.$$

Remark that, noting $E_0 = Post^\star(H_0) \cap \mathcal{F}$ and $E_i = Post^+(E_{i-1}) \cap \mathcal{F}$, at the end of the i-st passage in the while loop of Algorithm 1, $C = E_i$. The following lemma proves the correctness of Algorithm 1.

Lemma 29. *Let $i \geq 1$. $E_i = E_{i-1}$ iff $E_{i-1} = GFP(\lambda X.Post^+(X) \cap \mathcal{F} \cap Post^\star(H_0))$.*

Proof. ($\Leftarrow$) Suppose that $E_{i-1} = GFP(\lambda X.Post^+(X) \cap \mathcal{F} \cap Post^\star(H_0))$. In particular, $E_{i-1} = Post^+(E_{i-1}) \cap \mathcal{F} \cap Post^\star(H_0)$, i.e.:

$$E_{i-1} = E_i \cap Post^\star(H_0) \quad (*).$$

We will show that, $\forall j \geq 1$, $E_j \subseteq E_{j-1}$: in particular it means that $E_i \subseteq E_0 = Post^\star(H_0) \cap \mathcal{F} \subseteq Post^\star(H_0)$. It enables to conclude from $(*)$ that $E_{i-1} = E_i$.
Basis: We prove that $E_1 \subseteq E_0$. We know that $Post^\star(H_0) \cap \mathcal{F} \subseteq Post^\star(H_0)$. As function $Post^+$ is monotonic: $Post^+(Post^\star(H_0) \cap \mathcal{F}) \subseteq Post^+(Post^\star(H_0)) \subseteq Post^\star(H_0)$. So, $E_1 = Post^+(Post^\star(H_0) \cap \mathcal{F}) \cap \mathcal{F} \subseteq Post^\star(H_0) \cap \mathcal{F} = E_0$.
Induction: Suppose that $\forall 0 < k < i$, $E_k \subseteq E_{k-1}$. We must prove that $E_i \subseteq E_{i-1}$. By induction hypothesis, $E_{i-1} \subseteq E_{i-2}$, and as function $Post^+$ is monotonic: $Post^+(E_{i-1}) \subseteq Post^+(E_{i-2})$. Hence, $E_i = Post^+(E_{i-1}) \cap \mathcal{F} \subseteq Post^+(E_{i-2}) \cap \mathcal{F} = E_{i-1}$.
$(\Rightarrow)$ Suppose that $E_i = E_{i-1}$. Let us first prove that E_{i-1} is a fixed point of $\lambda(X) = Post^+(X) \cap \mathcal{F} \cap Post^\star(H_0)$. $\lambda(E_{i-1}) = Post^+(E_{i-1}) \cap \mathcal{F} \cap Post^\star(H_0) = E_i \cap Post^\star(H_0) = E_i = E_{i-1}$.
Now, let us prove that E_{i-1} is a greatest fixed point of λ. Let $Y \supseteq E_{i-1}$ such that $Y = \lambda(Y)$. We must prove that $Y = E_{i-1}$. As we already know that $E_{i-1} \subseteq Y$, it stays to prove that $Y \subseteq E_{i-1}$. As $Y = \lambda(Y)$, we have $Y = Post^+(Y) \cap \mathcal{F} \cap Post^\star(H_0) \subseteq \mathcal{F} \cap Post^\star(H_0) = E_0$. As $Y \subseteq E_0$ and function $Post^+$ is monotonic, $Post^+(Y) \subseteq Post^+(E_0)$. So, $Y = Post^+(Y) \cap \mathcal{F} \cap Post^\star(H_0) \subseteq Post^+(E_0) \cap \mathcal{F} \cap Post^\star(H_0) = E_1 \cap Post^\star(H_0) = E_1$. Applying inductively the same reasoning, we obtain that, $\forall k \geq 0$, $Y \subseteq E_k$. Hence, $Y \subseteq E_{i-1}$. □

Here is a theorem giving a rough size of the number of words that Algorithm 1 must explore in the worst case.

Theorem 30. *For every MITL formula φ, the associated OCATA $\mathcal{A}_{\neg\varphi} = (\Sigma, L, \ell_0, F, \delta)$, and all timed automaton $\mathcal{B} = (\Sigma, B, b_0, X, F^{\mathcal{B}}, \delta^{\mathcal{B}})$ with n clocks, $\mathcal{H}$ (constructed thanks to $\mathcal{S}_{\mathcal{B},\neg\varphi}$) contains $O(2^m)$ elements, where $m = (|B| + |L|) \cdot (2.c_{\max} + 2) \cdot 2 \cdot \max\left(\max_{i=1}^{n}(\frac{\max_i}{2}), n\right) \cdot (M(\neg\varphi) + n)$.*

Proof. Let us note $m' = (|B| + |L|) \cdot (2.c_{\max} + 2) \cdot 2 \cdot \max\left(\max_{i=1}^{n}(\frac{\max_i}{2}), n\right)$. There is $2^{m'}$ elements in $\Lambda = (B \cup L) \times REG \times \{\top, \bot\} \times \{1, 2, \ldots, \max(\max_{i=1}^{n}(\frac{\max_i}{2}), n)\}$. $\mathcal{H}$ is a set of words of $\Lambda^\star$ having at most $M(\neg\varphi) + n$ letters. There are:

- 1 word of 0 letters on $\Lambda^\star$;
- $2^{m'}$ word of 1 letters on $\Lambda^\star$;
- $2^{2.m'}$ word of 2 letters on $\Lambda^\star$;
- $2^{3.m'}$ word of 3 letters on $\Lambda^\star$;
- ... ;
- $2^{(M(\neg\varphi)+n).m'}$ word of $M(\neg\varphi) + n$ letters on $\Lambda^\star$;

Globally, there are $\Sigma_{j=0}^{M(\neg\varphi)+n} 2^{j.m'} = O\left(2^{(M(\neg\varphi)+n).m'}\right) = O(2^m)$ words in $\mathcal{H}$.

Zone-based algorithms In the case of TAs, zones have been advocated as a data structure which is more efficient in practice than regions [1]. Let us close this section by showing how zones for OCATA [2] can be adapted to represent set of states of $\mathcal{S}_{\mathcal{B},\neg\varphi}$. Intuitively, a zone is a constraint on the values of the clock

copies, with additional information ($loc_{\mathcal{A}}$ and $loc_{\mathcal{B}}$) to associate clock copies of $\mathcal{A}_{\neg\varphi}$ and clocks of $\mathcal{B}$ respectively to locations and Miyano-Hayashi markers.

Let us note x the unique clock of $\mathcal{A}_{\neg\varphi}$ and $x_1^{\mathcal{B}}, x_2^{\mathcal{B}}, \ldots, x_n^{\mathcal{B}}$ the clocks of $\mathcal{B}$. We will note $Copies(x)$ the set of $M(\neg\varphi)$ copies of x denoted $x_1, x_2, \ldots, x_{M(\neg\varphi)/2}, y_1, y_2, \ldots, y_{M(\neg\varphi)/2}$. Intuitively, each pair of clock copies (x_i, y_i) will represent an interval. We also note $Copies_{begin}(x) = \{x_1, x_2, \ldots, x_{M(\neg\varphi)/2}\}$, $Copies_{end}(x) = \{y_1, y_2, \ldots, y_{M(\neg\varphi)/2}\}$, $Copies^m(x) = \{x_1, x_2, \ldots, x_m, y_1, y_2, \ldots, y_m\}$, $Copies^m_{begin}(x) = \{x_1, x_2, \ldots, x_m\}$ for $1 \leq m \leq M(\neg\varphi)/2$, and $Copies^0(x) = Copies^0_{begin}(x) = \emptyset$.
Our definition of zone uses a supplementary clock x_0 whose value is always 0. Zones are also defined thanks to extended clock constraints on a set of clocks C, which are of the form $c \bowtie k$ and $c_1 - c_2 \bowtie k$ for $c, c_1, c_2 \in C$, $k \in \mathbb{N}$ and $\bowtie \in \{<, \leq, >, \geq\}$. When $k = 0$, we will shorten $c_1 - c_2 \bowtie k$ by $c_1 \bowtie c_2$, and we will also use $c_1 = c_2$ as shorthand for $c_1 - c_2 \geq 0 \wedge c_1 - c_2 \leq 0$.

Definition 31. *A zone $\mathcal{Z}_m$ is a 3-tuple $(loc_{\mathcal{A}}, loc_{\mathcal{B}}, Z)$ where (1) $loc_{\mathcal{A}} : Copies^m_{begin}(x) \to L \times \{\top, \bot\}$, (2) $loc_{\mathcal{B}}$ is a pair composed of a location of B and a marker in $\{\top, \bot\}$ and (3) Z is a set of extended clock constraints on $Copies^m(x) \cup \{x_0, x_1^{\mathcal{B}}, x_2^{\mathcal{B}}, \ldots, x_n^{\mathcal{B}}\}$, interpreted as a conjunction (it is a 'classical zone' on this set of clocks [1]).*

A zone $\mathcal{Z}_m = (loc_{\mathcal{A}}, loc_{\mathcal{B}}, Z)$, with $loc_{\mathcal{A}}(t) = (loc^t, mark^t)$ and $loc_{\mathcal{B}} = (loc, mark)$, represents the set of states of $\mathcal{S}_{\mathcal{B},\neg\varphi}$ of type $\{(loc, vx_1^{\mathcal{B}}, vx_2^{\mathcal{B}}, \ldots, vx_n^{\mathcal{B}}, mark), (loc^{x_1}, [vx_1, vy_1], mark^{x_1}), \ldots, (loc^{x_m}, [vx_m, vy_m], mark^{x_m})\}$, where $vx_1^{\mathcal{B}}, vx_2^{\mathcal{B}}, \ldots, vx_n^{\mathcal{B}}, vx_1, vy_1, \ldots, vx_m, vy_m$ are values respectively satisfying the extended clock constraints on $x_1^{\mathcal{B}}, x_2^{\mathcal{B}}, \ldots, x_n^{\mathcal{B}}, x_1, y_1, \ldots, x_m, y_m$ present in Z. By abuse of terminology, we will sometimes note "$s \in \mathcal{Z}_m$" to refer to a state of $\mathcal{S}_{\mathcal{B},\neg\varphi}$ represented by $\mathcal{Z}_m$.

Definition 32. *The initial zone is $\mathcal{Z}_1^{init} = (loc_{\mathcal{A}}, loc_{\mathcal{B}}, Z)$ with $Z = \{x_1^{\mathcal{B}} = 0, \ldots x_n^{\mathcal{B}} = 0, x_1 = 0, y_1 = 0\}$, $loc_{\mathcal{A}}(x_1) = (\ell_0, \bot)$ and $loc_{\mathcal{B}} = (b_0, mark)$ where $mark = \top$ iff b_0 is accepting.*
A zone Z is accepting iff $\forall 1 \leq i \leq m$, $\exists \ell_i \in L$ s.t. $loc_{\mathcal{A}}(x_i) = (\ell_i, \top)$ and $\exists b \in B$ s.t. $loc_{\mathcal{B}} = (b, \top)$.

Definition 33. *Let $\mathcal{Z}_m$ be a zone. $Post_T(\mathcal{Z}_m)$ denotes the set of configurations that are timed successors of a configuration in $\mathcal{Z}_m$.*

We can easily compute $Post_T(\mathcal{Z}_m)$ from $\mathcal{Z}_m$ deleting all clock constraints of the form $z - x_0 < k$ or $z - x_0 \leq k$, for $z \in Copies^m(x) \cup \{x_1^{\mathcal{B}}, x_2^{\mathcal{B}}, \ldots, x_n^{\mathcal{B}}\}$, of $\mathcal{Z}_m$.

Definition 34. *Let $\mathcal{Z}_m$ be a zone. $Post_D(\mathcal{Z}_m)$ denotes the set of configurations that are discrete successors of a configuration in $\mathcal{Z}_m$.*

We compute $Post_D(\mathcal{Z}_m)$, where $\mathcal{Z}_m = (loc_{\mathcal{A}}, loc_{\mathcal{B}}, Z)$, as follows:

- for each label $\sigma \in \Sigma$,
- for each possible transition $t_{\mathcal{B}}$ labelled by σ of $\mathcal{B}$, starting from location $loc_{\mathcal{B}}$,

– for each possible combination of transitions of $\mathcal{A}_{\neg\varphi}$ $(t_1, \ldots, t_m)$, all labelled by σ, such that $\forall 1 \leq i \leq m$, t_i starts from $loc_{\mathcal{A}}(x_i)$[8],

we are looking for possible successors as follow.
The possible successor zones are found following the arc $t_{\mathcal{B}}$ from the location of $loc_{\mathcal{B}}$, t_1 from the location of $loc_{\mathcal{A}}(x_1)$, ..., and t_m from the location of $loc_{\mathcal{A}}(x_m)$. First, to take $t_{\mathcal{B}}$ implies to satisfy the clock constraint it carries: these clock constraints must be added to those of Z on $\{x_1^{\mathcal{B}}, x_2^{\mathcal{B}}, \ldots, x_n^{\mathcal{B}}\}$ (it can be easily done on Z using a well-known algorithm on classical zones [6]). Location $loc_{\mathcal{B}}$ must also be modified in the location $t_{\mathcal{B}}$ goes to and, potentially, certain clocks among $\{x_1^{\mathcal{B}}, x_2^{\mathcal{B}}, \ldots, x_n^{\mathcal{B}}\}$ need to be reset (once again, it can be easily done using a well-known algorithm on classical zones [6]). Second, to take transitions t_i (for $1 \leq i \leq m$) implies that (x_i, y_i) must satisfy the clock constraints it carries: these clock constraints must be added to those of $\mathcal{Z}_m$ on $Copies(x)$. Moreover, to take transitions $t_1, \ldots, t_m$ can create new clock copies of value 0 in certain locations or/and letting clock copies with de same zone constraints in the same locations. The new copies with value 0 that have just been created in certain locations can either:

– be grouped with the previous smallest interval associated to this location
  ⇝ let us call ℓ this location: it corresponds to reset the clock x_i such that, in $\mathcal{Z}_m$, $loc_{\mathcal{A}}(x_i) = (\ell, mark)$ for a certain $mark \in \{0, 1\}$ and t_i loops on ℓ for a clock copy x_i with a minimum value.
  (We can only do this if there were such an interval associated to this location !)
– create a new interval [0,0] associated to this location in the zone.
  ⇝ let us call ℓ this location: it corresponds to use two new unused clock copies of x, x_{m+1} and y_{m+1}, and extend function loc in way $loc_{\mathcal{A}}(x_{m+1}) = \ell$. Clocks x_{m+1} and y_{m+1} must be reset.
  (We can only do this if the new number of intervals associated to the considered location does not exceed the maximal number of intervals we are allowed to associate to this location.)

Moreover, the markers of $loc_{\mathcal{A}}$ and $loc_{\mathcal{B}}$ must be kept updated.

Formally, let us note:

– $\forall \sigma \in \Sigma : E(c, \sigma) = \{$arcs t labeled by σ and s.t. $Start(t)$ is the location of $loc_{\mathcal{A}}(c)\}$,
– $\forall \sigma \in \Sigma : E(\mathcal{B}, \sigma) = \{$arcs starting from the location of $loc_{\mathcal{B}}$ and labeled by $\sigma\}$,
– $\forall \sigma \in \Sigma : Z \odot \sigma = \{(t_{\mathcal{B}}, t_1, \ldots, t_m) \mid t_{\mathcal{B}} \in E(\mathcal{B}, \sigma)$ and $\forall 1 \leq i \leq m : t_i \in E(x_i, \sigma)\}$,
– for every arc t: $Constr(t) = \{c \mid c$ constraint present on the arc $t\}$,

[8] they associate to each interval (x_i, y_i) a transition t_i to take

– for $t_{\mathcal{B}} = (b_{start}, \sigma, g, r, b_{arrival})$, $g_{t_{\mathcal{B}},t_1,\ldots,t_m} = g \quad \wedge \bigwedge_{\substack{1\leq i\leq m \\ c_i\in Constr(t_i)}} \Big(c_i|_{x=x_i} \wedge c_i|_{x=y_i} \Big)$,
– for $t_{\mathcal{B}} = (b_{start}, \sigma, g, r, b_{arrival})$, $LM(t_{\mathcal{B}}, t_1, \ldots, t_m) = \{(loc, mark) \mid (\exists 1 \leq i \leq m : loc \in Dest(t_i), loc$ is not equal to the location of $loc_{\mathcal{A}}(x_i)$ and $mark$ is the marker of $loc_{\mathcal{A}}(x_i))$ or $(loc = b_{arrival}$ and $mark$ is the marker of $loc_{\mathcal{B}})\}$,
– $Loop(t_1, \ldots, t_m) = \{x_i \mid$ the location of $loc_{\mathcal{A}}(x_i)$ is in $Dest(t_i)\} \cup \{y_i \mid$ the location of $loc_{\mathcal{A}}(x_i)$ is in $Dest(t_i)\}$,
– $minLoop(t_1, \ldots, t_m) = \{x_i \in Loop(t_1, \ldots, t_m) \cap Copies_{begin}(x) \mid \forall x'_i \in Loop(t_1, \ldots, t_m), x'_i \geq x_i$ results from the constraints of Z$\}$,
– $Reset^{\ell}(t_1, \ldots, t_m) = \{x^{\ell}, y^{\ell}\}$ for a certain $x^{\ell} \in Copies_{begin}(x) \backslash Loop(t_1, \ldots, t_m)$ and $y^{\ell} \in Copies_{end}(x) \setminus Loop(t_1, \ldots, t_m)\}$ if these sets are not empty and $\exists mark \in \{0, 1\}$ s.t. $(\ell, mark) \in LM(t_{\mathcal{B}}, t_1, \ldots, t_m)$; otherwise, $Reset^{\ell}(t_1, \ldots, t_m) = \emptyset$,
– $R^{\ell} = \{Reset^{\ell}(t_1, \ldots, t_m)\} \cup \{\{x_i\} \mid Reset^{\ell}(t_1, \ldots, t_m) \neq \emptyset$, $x_i \in minLoop(t_1, \ldots, t_m)$ and $loc_{\mathcal{A}}(x_i) = \ell\}$.

$\mathcal{Z}'_{m'} = (loc'_{\mathcal{A}}, loc'_{\mathcal{B}}, Z') \in Post_D(\mathcal{Z}_m)$ iff (1) $\mathcal{Z}_m$ is not final, $g_{t_{\mathcal{B}},t_1,\ldots,t_m} \cap Z$ is satisfiable and $\exists \sigma \in \Sigma, (t_{\mathcal{B}}, t_1, \ldots, t_m) \in Z \odot \sigma$ and $(r_{\ell_1}, \ldots, r_{\ell_p})$, with $\forall 1 \leq i \leq p, r_{\ell_i} \in R_{\ell_i}$, such that :

– $Z' = [\Big(r \cup \bigcup_{j=1}^{p} r_{\ell_j}\Big) := 0] \, (g_{t_{\mathcal{B}},t_1,\ldots,t_m} \cap Z)$,
– $\forall x_i \in Loop(t_1, \ldots, t_m)$, $loc'_{\mathcal{A}}(x_i)$ is composed of the location of $loc_{\mathcal{A}}(x_i)$ and of:
  • marker "1" if the location of $loc_{\mathcal{A}}(x_i)$ is in F,
  • the marker of $loc_{\mathcal{A}}(x_i)$ otherwise ;

  $\forall x_{\ell} \in Reset^{\ell}(t_1, \ldots, t_m) \cap Copies_{begin}(x)$, $loc'_{\mathcal{A}}(x_{\ell})$ is composed of the location ℓ and of marker:
  • "1" if $\ell \in F$ or $(\ell, 0) \notin LM(t_{\mathcal{B}}, t_1, \ldots, t_m)$,
  • "0" otherwise ;
– $loc'_{\mathcal{B}}$ is composed of $Dest(t_{\mathcal{B}})$ and of marker:
  • "1" if $Dest(t_{\mathcal{B}}) \in F^{\mathcal{B}}$ or $(Dest(t_{\mathcal{B}}), 0) \notin LM(t_{\mathcal{B}}, t_1, \ldots, t_m)$,
  • "0" otherwise ;

or (2) $\mathcal{Z}_m$ is final and, defining $loc^{\star}_{\mathcal{A}}$ such that $\forall 1 \leq i \leq m$, if $loc_{\mathcal{A}}(x_i) = (\ell, 1)$, then $loc^{\star}_{\mathcal{A}}(x_i) = (\ell, 0)$, $\mathcal{Z}'_{m'}$ satisfies the conditions of case (1) in which $loc^{\star}_{\mathcal{A}}$ plays the role of $loc_{\mathcal{A}}$.

Proposition 35. *Let $\mathcal{Z}_m$ be a zone. $Post_D(\mathcal{Z}_m) = \{s' \mid \exists s \in \mathcal{Z}_m$ s.t. $s \to s'$ in $\mathcal{S}_{\mathcal{B},\neg\varphi}\}$.*

Proof. ($\subseteq$) Suppose that $\mathcal{Z}_m = (loc_{\mathcal{A}}, loc_{\mathcal{B}}, Z)$, where $loc_{\mathcal{B}} = (b, mark_{\mathcal{B}})$ and $\forall 1 \leq k \leq m$, $loc_{\mathcal{A}}(x_k) = (\ell_k, mark_k)$. Let $s' \in Post_D(\mathcal{Z})$. There exists a certain $s' \in \mathcal{Z}'_{m'}$ for a certain $\mathcal{Z}'_{m'} \in Post_D$ constructed thanks to $\sigma, t_{\mathcal{B}}, t_1, \ldots, t_m, r_{\ell_1}, \ldots, r_{\ell_p}$, we suppose that $t_{\mathcal{B}} = (b_{start}, \sigma, g, r, b_{arrival})$. Let us suppose that $s' = \{(\ell'_{k'}, I'_{k'}, mark'_{k'})_{k'=1}^{m'}\} \cup \{(b', v', mark'_{\mathcal{B}})\}$, with $(\forall 1 \leq k' \leq m')$ $I'_{k'} = [v'x_{k'}, v'y_{k'}]$. We construct $s = \{(\ell_k, I_k, mark_k)_{k=1}^{m}\} \cup \{(b, v, mark_{\mathcal{B}})\}$, with $(\forall 1 \leq k \leq m)$ $I_k = [vx_k, vy_k]$, where:

1. Arc of $\mathcal{B}$ without reset: $\forall 1 \leq i \leq n$: if $x_i^{\mathcal{B}} \notin r$, we define $v(x_i^{\mathcal{B}}) = v'(x_i^{\mathcal{B}})$,
2. New complete interval: $\forall 1 \leq k \leq m$: if there exists $1 \leq j \leq p$ such that $x_k \in r_{\ell_j} \cap Reset^{\ell_j}(t_1, \ldots, t_m)$, then x^{ℓ_k} and y^{ℓ_k} was not used in $\mathcal{Z}$ and their values must not be defined,
3. Loop without merge: $\forall 1 \leq k \leq m$: if $\forall 1 \leq j \leq p$, $x_k \notin r_{\ell_j}$ but that $loc'_{\mathcal{A}}$ is defined on x_k, we define $vx_k = v'x_k$ and $vy_k = v'y_k$,
4. Loop with merge: $\forall 1 \leq k \leq m'$: if there exists $1 \leq j \leq p$ such that $x_k \in r_{\ell_j} \setminus Reset^{\ell_j}(t_1, \ldots, t_m)$, then we define $vy_k = v'y_k$,
5. Arc going out or arc of $\mathcal{B}$ with reset: the values vx_k, vy_k and $v(x_i^{\mathcal{B}})$ that we still must define are arbitrarily chosen in way they satisfy the extended clock constraints of $g_{t_{\mathcal{B}},t_1,\ldots,t_m} \cap Z$ (which is possible because $g_{t_{\mathcal{B}},t_1,\ldots,t_m} \cap Z$ is satisfiable and we only chose values of clock/clock copies that have the same in $\mathcal{Z}'_{m'}$, so that they can not prevent $g_{t_{\mathcal{B}},t_1,\ldots,t_m} \cap Z$ from being satisfiable).

We suppose that $\mathcal{Z}_m$ is not accepting, so that s is not accepting, the proof is very similar in the other case. We must prove that $s \xrightarrow{\sigma} s'$ in $\mathcal{S}_{\mathcal{B},\neg\varphi}$ (see conditions (i)-(a),(b),(c) and (d) of the definition of $\rightarrow$ in $\mathcal{S}_{\mathcal{B},\neg\varphi}$). Condition (d) (resp. (b) and (c)) is (are) trivially verified by definition of the function $loc'_{\mathcal{B}}$ (resp. $loc'_{\mathcal{A}}$) of $\mathcal{Z}'_{m'}$. Its stays to prove (a), i.e.: $(b, v) \xrightarrow{\sigma} (b', v')$ in $\mathcal{B}$ and $\{(\ell_k, I_k)_{k=1}^m\} \xrightarrow{\sigma}_{f^\star_{\neg\varphi}} \{(\ell'_{k'}, I'_{k'})_{k'=1}^{m'}\}$ in $\mathcal{A}_{\neg\varphi,f^\star_{\neg\varphi}}$.
We first show that $(b, v) \xrightarrow{\sigma} (b', v')$ in $\mathcal{B}$ thanks to $t_{\mathcal{B}}$. Indeed, $v \models g$ because g is contained in $g_{t_{\mathcal{B}},t_1,\ldots,t_m} \cap Z$ (satisfied) ; $\forall x \in r$, $v'(x) = 0$ because it is reset by construction of $\mathcal{Z}'_{m'}$ and $\forall x \in \{x_1^{\mathcal{B}}, \ldots, x_2^{\mathcal{B}}\} \setminus r$, $v'(x) = v(x)$ by 1..
Now, let us show that $\{(\ell_k, I_k)_{k=1}^m\} \xrightarrow{\sigma}_{f^\star_{\neg\varphi}} \{(\ell'_{k'}, I'_{k'})_{k'=1}^{m'}\}$ in $\mathcal{A}_{\neg\varphi,f^\star_{\neg\varphi}}$. We must prove there exists minimal models M_k, for $1 \leq k \leq m$ of $\delta(\ell_k, \sigma)$ wrt I_k such that $\{(\ell'_{k'}, I'_{k'})_{k'=1}^{m'}\} \in f^\star_{\neg\varphi}(\bigcup_{k=1}^m M_k)$. For each $1 \leq k \leq m$, we take M_k to be the minimal model of $\delta(\ell_k, \sigma)$ wrt I_k obtained following the arc t_k: it can be taken because its constraints are verified on vx_k and vy_k (present in $g_{t_{\mathcal{B}},t_1,\ldots,t_m} \cap Z$) and as this constraint can only be an interval (convex), it is also verified on each $i \in I_k$. We then take the element E of $f^\star_{\neg\varphi}(\bigcup_{k=1}^m M_k)$ that merges the two more little intervals present in ℓ_j (for $1 \leq j \leq p$) iff r_{ℓ_j} is a singleton. It stays to prove that the obtained configuration is exactly $\{(\ell'_{k'}, I'_{k'})_{k'=1}^{m'}\}$. It is based on the following facts:

a. each (ℓ_k, I_k) that does not loop disappear from $\{(\ell_k, I_k)_{k=1}^m\}$ to E and is not present in $\{(\ell'_{k'}, I'_{k'})_{k'=1}^{m'}\}$: the clock copies representing such intervals disappear from $\mathcal{Z}_m$ to $\mathcal{Z}'_{m'}$ (i.e. $loc_{\mathcal{A}}$ is not defined on them anymore),
b. for each location ℓ_j $(1 \leq j \leq p)$ destination of at least one t_k $(1 \leq k \leq m)$, $(\ell_j, [0, 0])$ is present in $\bigcup_{k=1}^m M_k$ and
   - if $r_{\ell_j} = Reset^{\ell_j}(t_1, \ldots, t_m)$: two new clock copies, say x_{ℓ_j} and y_{ℓ_j} are used by $\mathcal{Z}'_{m'}$. They are defined and reset in way $(\ell_j, [0, 0])$ is also present in $\{(\ell'_{k'}, I'_{k'})_{k'=1}^{m'}\}$;
   - else: by definition of r_{ℓ_j}, R^{ℓ_j} and $minLoop(t_1, \ldots, t_m)$, the clock copy x_j representing the beginning of the more little interval in $\{(\ell_k, I_k)_{k=1}^m\}$ that loops on ℓ_j, say $I_j = [vx_j, vy_j]$, is reset in $\mathcal{Z}'_{m'}$: $\{(\ell'_{k'}, I'_{k'})_{k'=1}^{m'}\}$ contains $(\ell_j, [0, vy_j])$;

c. each (ℓ_k, I_k) that loops is still present in $\bigcup_{k=1}^{m} M_k$: the clock copies representing I_k in $\mathcal{Z}_m$ are still present in $\mathcal{Z}'_{m'}$ but the clock copy representing its beginning could have been reset, so that $\{(\ell'_{k'}, I'_{k'})_{k'=1}^{m'}\}$ contains either $(\ell_k, [vx_k, vy_k])$ or $(\ell_k, [0, vy_k])$,
d. when computing E from $\bigcup_{k=1}^{m} M_k$, we know the two more little intervals present in ℓ_j (for $1 \leq j \leq p$) are merged iff r_{ℓ_j} is non empty and equal to $Reset^{\ell_j}(t_1, \ldots, t_m)$. So, $(\ell_j, [0,0])$ and (ℓ_j, I_j) are merged in $(\ell_j, [0, vy_j])$ iff I_j is the more little interval that loops on ℓ_j, r_{ℓ_j} is a singleton and so must contain the clock copy representing its beginning : x_j. In this case and only in this case, x_j is then reset constructing $\mathcal{Z}'_{m'}$ so that E contains $(\ell_j, [0, vy_j])$ iff $\{(\ell'_{k'}, I'_{k'})_{k'=1}^{m'}\}$ also contains $(\ell_j, [0, vy_j])$.

($\supseteq$) Let $\mathcal{Z}_m = (loc_{\mathcal{A}}, loc_{\mathcal{B}}, Z)$ be a zone and s' be such that $\exists s \in \mathcal{Z}_m$ s.t. $s \rightarrow s'$ in $\mathcal{S}_{\mathcal{B},\neg\varphi}$. Let us show that $s' \in Post_D(\mathcal{Z}_m)$. Let us suppose that $s = \{(\ell_k, I_k, mark_k)_{k=1}^{m}\} \cup \{(b, v, mark_{\mathcal{B}})\}$, with $(\forall 1 \leq k \leq m)$ $I_k = [vx_k, vy_k]$ and $s' = \{(\ell'_{k'}, I'_{k'}, mark'_{k'})_{k'=1}^{m'}\} \cup \{(b', v', mark'_{\mathcal{B}})\}$, with $(\forall 1 \leq k' \leq m')$ $I'_{k'} = [v'x_{k'}, v'y_{k'}]$. As $s \rightarrow s'$, there exists $\sigma \in \Sigma$ such that :

- $(b, v) \xrightarrow{\sigma} (b', v')$ in $\mathcal{B}$, i.e.: there exists an arc $t_{\mathcal{B}} = (b, \sigma, g, r, b')$ s.t. $v \models g$, $\forall x \in r$, $v'(x) = 0$ and $\forall x \in \{x_1^{\mathcal{B}}, \ldots, x_n^{\mathcal{B}}\} \setminus r$, $v'(x) = v(x)$;
- $\{(\ell_k, I_k)_{k=1}^{m}\} \xrightarrow{\sigma}_{f^{\star}_{\neg\varphi}} \{(\ell'_{k'}, I'_{k'})_{k'=1}^{m'}\}$ in $\mathcal{A}_{\neg\varphi, f^{\star}_{\neg\varphi}}$, i.e.: $\{(\ell'_{k'}, I'_{k'})_{k'=1}^{m'}\} = E \in f^{\star}_{\neg<Phi}(\bigcup_{k=1}^{m} M_k)$ for certain minimal models M_k of $\delta(\ell_k, \sigma)$ wrt I_k, which are themselves obtained by taking certain arcs t_k (for $1 \leq k \leq m$) from ℓ_k.

Let us define r_{ℓ_j}, for $1 \leq j \leq p$ by:

- $r_{\ell_j} = Reset^{\ell_j}(t_1, \ldots, t_m)$ contains two new clock copies *iff* $\exists 1 \leq k \leq m$ s.t. t_k goes to ℓ_j with a reset and no merge is applied by $f^{\star}_{\neg\varphi}$ on ℓ_j,
- r_{ℓ_j} contains the clock representing the beginning of the more little interval present in ℓ_j that loops on ℓ_j taking one of the t_k (for $1 \leq k \leq m$) *iff* $\exists 1 \leq k \leq m$ s.t. t_k goes to ℓ_j with a reset and a merge is applied by $f^{\star}_{\neg\varphi}$ on ℓ_j,
- $r_{\ell_j} = \emptyset$ in the other case, i.e. when none of the t_k (for $1 \leq k \leq m$) goes to ℓ_j with a reset.

It is easy to prove that the $\mathcal{Z}'_{m'} \in Post_D(\mathcal{Z}_m)$ induced by $\sigma, t_{\mathcal{B}}, t_1, \ldots, t_m, r_{\ell_1}, \ldots, r_{\ell_p}$ contains s' thanks to the following facts:

- as $s \in \mathcal{Z}_m$, the constraints of Z are satisfied by its clock values and having take arcs $t_{\mathcal{B}}, t_1, \ldots, t_m$ ensure, the bounds of the intervals and the clock values of s satisfy $g_{t_{\mathcal{B}}, t_1, \ldots, t_m} \cap Z$ (in particular $g_{t_{\mathcal{B}}, t_1, \ldots, t_m} \cap Z$ is satisfiable and $\mathcal{Z}'_{m'}$ really exists),
- a clock $x_i^{\mathcal{B}}$ (for $1 \leq i \leq n$) is reset in the construction of $\mathcal{Z}'_{m'}$ iff $v'(x_i^{\mathcal{B}}) = 0$,
- the facts b., c. and d. of the proof of inclusion $\subseteq$ are still true here.
- the markers are treated in a similar manner from $\mathcal{Z}_m$ to $\mathcal{Z}'_{m'}$ as from s to s', observing the definition of the $\mathcal{Z}'_{m'} \in Post_D(\mathcal{Z}_m)$ induced by $\sigma, t_{\mathcal{B}}, t_1, \ldots, t_m$, $r_{\ell_1}, \ldots, r_{\ell_p}$ and conditions (i)-(a),(b),(c),(d) and (ii) of the definition of $\rightarrow$ in $\mathcal{S}_{\mathcal{B},\neg\varphi}$.

□

Algorithm 1 is still correct replacing H_0 by $\mathcal{Z}_1^{init}$, the function $Post(\mathcal{W})$ by $Approx_\beta(Post(\mathcal{Z}))$[9] and if $\mathcal{F}$ designates the set of accepting zones. The proof relies on a very simple adaptation of Theorem 2 of [4] to our definition of zone.

Eliminating useless clock copies In many practical examples, MITL formulas contain modalities of the form $U_{[0,+\infty)}$ or $\tilde{U}_{[0,+\infty)}$ that do not impose any real-time constraints (in some sense, they are LTL modalities). For instance, consider the $\Box$ modality in $\varphi = \Box(a \Rightarrow \Diamond_{[1,2]} b)$. When this occurs in a formula φ, we can simplify the representation of configurations of $\mathcal{A}_\varphi$, by dropping the values of the clocks associated to those modalities (these clocks can be regarded as *inactive* in the sense of [8]). We call those configurations *reduced configurations*. In the example of Fig. 1, this amounts to skipping the clocks associated to $\ell_\Box$, and the configuration $\{(\ell_\Box, 0.1)(\ell_\Diamond, 0)\}$ of $\mathcal{A}_\varphi$ in Fig. 1 can be represented by a pair $(\{\ell_\Box\}, \{(\ell_\Diamond, 0)\})$. As we will see in the next section, maintaining reduced configurations, when possible, usually improves the performance of the algorithms.
One can also verify that the values of the clock copies present in the initial location $(\neg\varphi)_{init}$ of $\mathcal{A}_{\neg\varphi}$ are not relevant. Let us note $Sub_{[0,+\infty)}(\neg\varphi) = \{\varphi \in Sub(\neg\varphi) \mid \varphi = \varphi_{init}$ or the outermost operator of φ is $U_{[0,+\infty)}$ or $\tilde{U}_{[0,+\infty)}\}$. Then, the states of $\mathcal{A}$ can be a couple (ℓ, I) where $\ell \in L \setminus Sub_{[0,+\infty)}(\neg\varphi)$ and $I \in \mathcal{I}(\mathbb{R}^+)$ as well as a singleton $\ell \in Sub_{[0,+\infty)}(\neg\varphi)$. A reduced configuration is then a pair (S, C) where $S \subseteq Sub_{[0,+\infty)}(\neg\varphi)$ and C is a configuration of $\mathcal{A}_{\neg\varphi}$ such that $\forall(\ell, v) \in C$, $\ell \notin Sub_{[0,+\infty)}(\neg\varphi)$. It is clear that all the previous results still hold on reduced configurations and we also implemented Algorithm 1 with them : the region or zone abstraction is only used on $L \setminus Sub_{[0,+\infty)}(\neg\varphi\varphi)$.

6 Experimental results

To evaluate the practical feasibility of our approach, we have implemented the region and zone-based algorithms for model-checking and satisfiability in a prototype tool. To the best of our knowledge, this is the first implementation to perform MITL model-checking and satisfiability using an automata-based approach. We first consider a benchmark for the *satisfiability problem*, adapted from the literature on LTL [9] and consisting of six parametric formulas (with $k \in \mathbb{N}$ and $I \in \mathcal{I}(\mathbb{N}^{+\infty})$):

$$E(k,I) = \bigwedge_{i=1,\dots,k} \Diamond_I\, p_i \qquad U(k,I) = (\dots(p_1 U_I p_2) U_I \dots) U_I p_k$$
$$A(k,I) = \bigwedge_{i=1,\dots,k} \Box_I\, p_i \qquad T(k,I) = p_1 \tilde{U}_I (p_2 \tilde{U}_I (p_3 \dots p_{k-1} \tilde{U}_I p_k) \dots)$$
$$Q(k,I) = \bigwedge_{i=1,\dots,k} (\Diamond_I p_i \vee \Box_I p_{i+1}) \qquad R(k,I) = \bigwedge_{i=1,\dots,k} (\Box_I\, (\Diamond_I p_i) \vee\ \Diamond_I (\Box_I p_{i+1}))$$

Table 1 (top) reports on the running time, number of visited regions and returned answer (column 'Sat ?') of the prototype on several instances of those formulas, for the different data structures. A time out was set after 5 minutes, and OOM

[9] see [4] for details on the definition of $Approx_\beta$

Table 1. Benchmark for satisfiability (top) and model-checking (bottom). Reported values are execution time in ms / number of visited states.

Sat ?	Formula	Size	Regions	Reduced regions	Zones	Reduced zones
Sat	$E(5,[0,+\infty))$	5	74 / 61	16 / 31	58 / 36	39 / 31
Sat	$E(10,[0,+\infty))$	10	3296 / 2045	369 / 1023	1374 / 1033	2515 / 1023
Sat	$E(5,[5,8))$	5	382 / 228	394 / 228	83 / 33	86 / 33
Sat	$E(10,[5,8))$	10	70129 / 7172	79889 / 7172	1982 / 1025	2490 / 1025
Sat	$A(10,[0,+\infty))$	10	1 / 1	1 / 1	4 / 1	5 / 1
Sat	$A(10,[5,8))$	10	1926 / 7	2036 / 5	3036 / 2	3153 / 2
Sat	$U(10,[0,+\infty))$	9	231 / 7	5 / 4	16 / 1	6 / 1
Unsat	$U(2,[5,8])$	2	13 / 6	15 / 8	4 / 2	4 / 2
Unsat	$U(3,[5,8])$	3	OOM	OOM	OOM	OOM
Sat	$T(10,[0,+\infty[)$	9	> 5min	3 / 2	33 / 3	7 / 2
Sat	$T(10,[5,8))$	9	52 / 2	40 / 2	11 / 2	11 / 2
Sat	$R(5,[0,+\infty))$	20	> 5min	301 / 270	4307 / 1321	145 / 81
Sat	$R(10,[0,+\infty))$	40	> 5min	OOM	OOM	> 5min
Sat	$R(5,[5,8))$	20	OOM	6996 / 117	1299 / 36	1518 / 36
Sat	$R(10,[5,8))$	40	> 5min	> 5min	> 5min	> 5min
Sat	$Q(5,[0,+\infty))$	10	44 / 39	11 / 20	43 / 29	22 / 20
Sat	$Q(10,[0,+\infty))$	20	1209 / 1041	286 / 521	841 / 540	933 / 521
Sat	$Q(5,[5,8))$	10	497 / 98	378 / 57	167 / 32	181 / 32
Sat	$Q(10,[5,8))$	20	35776 / 2646	20324 / 2912	81774 / 782	86228 / 782

Floors	Formula	Form./ TA size	OK ?	Regions	Zones	Red. zones
2	$\Box \bigwedge_{i=1,2} (o_i \Rightarrow \Diamond_{]1,2]} c_i)$	3 / 10	×	166 / 89	57 / 35	56 / 32
2	$\Box \bigwedge_{i=1,2} (b_i \Rightarrow \Diamond_{[0,4]} o_i)$	3 / 10	✓	302 / 225	31 / 31	26 / 25
2	$\Box \bigwedge_{i=1,2} (l_i \Rightarrow \Diamond_{[0,6]} o_i)$	3 / 10	✓	820 / 690	68 / 51	60 / 40
3	$\Box \bigwedge_{i=1,\dots,3} (o_i \Rightarrow \Diamond_{]1,2]} c_i)$	4 / 37	×	681 / 480	463 / 154	337 / 140
3	$\Box \bigwedge_{i=1,\dots,3} (b_i \Rightarrow \Diamond_{[0,12]} o_i)$	4 / 37	✓	>5min	1148 / 541	1008 / 376
3	$\Box \bigwedge_{i=1,\dots,3} (l_i \Rightarrow \Diamond_{[0,14]} o_i)$	4 / 37	✓	>5min	1321 / 774	1387 / 540
4	$\Box \bigwedge_{i=1,\dots,4} (o_i \Rightarrow \Diamond_{]1,2]} c_i)$	5 / 114	×	5570 / 1638	1381 / 498	1565 / 461
4	$\Box \bigwedge_{i=1,\dots,4} (b_i \Rightarrow \Diamond_{[0,20]} o_i)$	5 / 114	✓	>5min	26146 / 5757	22776 / 3156
4	$\Box \bigwedge_{i=1,\dots,4} (l_i \Rightarrow \Diamond_{[0,22]} o_i)$	5 / 114	✓	>5min	52167 / 7577	48754 / 4337
5	$\Box \bigwedge_{i=1,\dots,5} (o_i \Rightarrow \Diamond_{]1,2]} c_i)$	6 / 311	×	61937 / 4692	3216 / 1402	3838 / 1310
5	$\Box \bigwedge_{i=1,\dots,5} (b_i \Rightarrow \Diamond_{[0,28]} o_i)$	6 / 311	✓	>5min	>5min	OOM
5	$\Box \bigwedge_{i=1,\dots,5} (l_i \Rightarrow \Diamond_{[0,30]} o_i)$	6 / 311	✓	OOM	>5min	>5min

stands for 'out of memory'. Our second benchmark evaluates the performance of our *model-checking tool*. We consider a family of timed automata $\mathcal{B}_k^{lift}$ that model a *lift*, parametrised by the number k of floors. A button can be pushed at each floor to call it. The alphabet contains a letter l_i for each floor i saying that lift has been called at this floor. A button to send the lift at each floor is present in it: the alphabet contains a letter b_i for each floor i saying that button i has just been pushed. The lift takes 1 second to go from a floor to the next one and to open/close its doors : it stays 1 second open between these opening/closure. Letter o_i (resp. c_i) signifies the lift open (resp. close) its doors at floor i ; letter p_i means the lift pass floor i without stopping. The lift goes up (resp. down) as long as it is called upper (resp. lower). When it is not called anywhere, it goes to the medium floor and stays opened there.

Fig. 3 gives a representation of $\mathcal{B}_2^{lift}$ as example. We represent two similar edges (same starting and ending locations, same reset) carrying two different letters by a unique edge carrying these two letters. A location of this automaton is a 4-tuple $(n, direction, go, open?)$ where n is the number of the floor the cabin is present in ($0 \leq n < 2$, 0 representing the ground floor), $direction \in \{u, d, h\}$ is

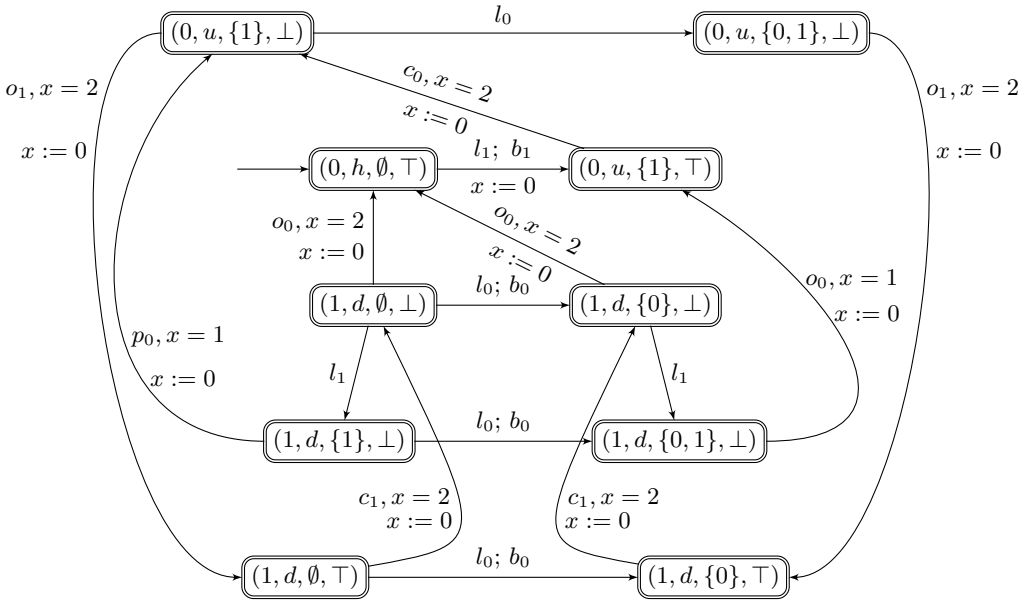

Fig. 3. $\mathcal{B}_2^{lift}$

u if the lift is going up, d if it is going down and h if it stays at the floor it is present in, *go* is the set of floors to which the cabin must go (because the lift has been called at this floor or because the button present in the cabin has been pushed to send the lift at this floor), and *open*? $\in \{\top, \bot\}$ is $\top$ iff the doors of the cabin are opened.

References

1. R. Alur and D. L. Dill. A theory of timed automata. *Theor. Comput. Sci.*, 126(2):183–235, 1994.
2. P. A. Abdulla, J. Deneux, J. Ouaknine, K. Quaas and J. Worrell. Universality Analysis for One-Clock Timed Automata. *Fundam. Inform.*, 89(4):419–450, 2008.
3. R. Alur, T. Feder and T. A. Henzinger. The benefits of relaxing punctuality. *J. ACM*, 43(1):116–146, 1996.
4. P. Bouyer. Timed Automata may cause some troubles. Research Report LSV-02-9, Lab. Spécification et Vérification, CNRS & ENS de Cachan, France, 2002.
5. T. Brihaye, M. Estiévenart and G. Geeraerts. On MITL and Alternating Timed Automata. In *FORMATS*, volume 8053 of *LNCS*. Springer, 2013.
6. J. Bengtsson and W. Yi. Timed Automata: Semantics, Algorithms and Tools. *Lectures on Concurrency and Petri Nets* 87-124, 2003.
7. E. M. Clarke, O. Grumberg, and D. Peled. *Model checking*. MIT Press, 2001.
8. C. Daws and S. Yovine. Reducing the number of clock variables of timed automata. *Real-Time Systems*, 73-81, 1996.

9. G. Geeraerts, G Kalyon, T. Le Gall, N. Maquet and J.-F. Raskin. Lattice-Valued Binary Decision Diagrams. In *ATVA*, volume 6252 of *LNCS*. Springer, 2010.
10. P. Gastin and D. Oddoux. Fast LTL to Büchi Automata Translation. In *CAV*, volume 2102 of *LNCS*. Springer, 2001.
11. R. Koymans. Specifying real-time properties with metric temporal logic. *Real-Time Systems*, 2(4):255–299, 1990.
12. O. Kupferman and M. Y. Vardi. Weak alternating automata are not that weak. *ACM Trans. Comput. Log.*, 2(3):408–429, 2001.
13. N. Maquet New Algorithms and Data Structures for the Emptiness Problem of Alternating Automata. PhD thesis, Université Libre de Bruxelles, 2011.
14. S. Miyano and T. Hayashi. Alternating Finite Automata on omega-Words. *Theor. Comput. Sci.*, 32:321–330, 1984.
15. J. Ouaknine and J. Worrell. On the decidability of metric temporal logic. In *LICS'05*. IEEE, 188-197, 2005.
16. J. Ouaknine and J. Worrell. On the decidability and complexity of metric temporal logic over finite words. *Logical Methods in Computer Science*, 3(1), 2007.
17. P. Parys and I. Walukiewicz. Weak Alternating Timed Automata. *Logical Methods in Computer Science*, 8(3), 2012.