On the Initialization of Clocks in Timed Formalisms

Marcello M. Bersani[a], Matteo Rossi[a], Pierluigi San Pietro[a,b]

[a] *Dipartimento di Elettronica Informazione e Bioingegneria, Politecnico di Milano, Piazza Leonardo da Vinci 32, Milano, Italy*
[b] *CNR IEIIT-MI, Milano, Italy*

Abstract

Constraint LTL over clocks (CLTLoc) is an extension of LTL allowing for atomic formulae of the form $x < c$ or $x = c$, which constrain the time delay measured by clock x with respect to constant value c. In a previous work, we showed that CLTLoc is equivalent to Timed Automata. The result was proven by considering a clock semantics that conforms to the original definition by Alur and Dill, i.e., when clocks are reset (i.e., equal to 0) in the origin, both CLTLoc and Timed Automata define the class of Timed ω-Regular languages. In this paper, we show that if we allow the clocks to have any value in the origin, the power of the formalism to express timed languages does not change, as long as non-Zeno languages are considered. If Zeno languages are allowed, then CLTLoc is strictly more powerful than TA. As a consequence of these results, we also show that non-Zeno Timed ω-Regular languages are closed with respect to the left quotient operation with timed regular languages over finite words.

Keywords: Metric temporal logic, Satisfiability, Formal languages, Language equivalence, Timed Automata, Timed regular languages, Zenoness.

1. Introduction

Timed Automata (TA) [1] are the standard operational formalism for real time modeling, with a large number of applications and theoretical results. Among the latter, logic characterizations are of great relevance, since they show whether techniques such as model or satisfiability checking can be applied to TA as well.

In previous work, we have bridged the gap between TA and temporal logic over the pointwise semantics, by considering the logic Constraint LTL over clocks (CLTLoc), an extension of LTL that still considers discrete positions, but it has also a finite set of variables over a dense time domain, behaving like clocks of TA, to measure time elapsing among events occurring at discrete positions. Unlike MTL, clocks are explicit resources in CLTLoc and, as in TA, they can be compared with constants over $\mathbb{N}_{\geq 0}$ (or $\mathbb{Q}$). In [2], we prove that satisfiability of CLTLoc is PSPACE-complete, by combining results on the decidability of CLTL [3],[4] over $\mathbb{R}$ with Region Graphs [1] capturing the time

Email addresses: marcellomaria.bersani@polimi.it (Marcello M. Bersani), matteo.rossi@polimi.it (Matteo Rossi), pierluigi.sanpietro@polimi.it (Pierluigi San Pietro)

behavior of variables. Moreover, the satisfiability of CLTLoc can be reduced to an instance of an SMT (Satisfiability Modulo Theories) problem. A decision procedure was then easily devised and implemented,[1] by adopting SMT solvers such as Z3 [5]. CLTLoc has been successfully employed to reduce MITL over continuous semantics [6], allowing us to implement the first effective tool solving the satisfiability of MITL.[2]

In [7], we proved the language equivalence of CLTLoc and TA over timed sequences. However, in TA all clocks are assumed to be initialized to 0, while CLTLoc does not impose any a priori constraint on clock values. Equivalence with TA was actually proved under the assumption that clocks in CLTLoc are well-initialized—i.e., all clocks in the first position are either equal to 0 or to a constant, the same for all clocks. This very naturally poses the question whether well-initialization in CLTLoc and the zero initialization of TA are essential constraints.

Indeed, under the assumption that clocks are initialized to zero, it is possible to show that a number of syntactic extensions of TA do not increase the expressive power of the formalism. A notable example is that of so-called diagonal constraints of the form $x \sim y + c$, where x and y are clocks [8].

In this paper, we wish to understand whether it would be possible to define a wider class of timed regular languages by not enforcing these restrictions on the initial value. Indeed, in Sec. 3, we show a Zeno language (i.e., in which timestamps accumulate) that is not timed regular, and that can be defined by means of a TA with one clock that is not initialized at 0. However, the main result of this paper is that in the common case of non-Zeno behaviors, the class of timed languages does not change.

We also wish to investigate whether the possibility that clocks initially have arbitrary values can make syntactic extensions such as the admissibility of diagonal constraints also semantically more powerful. Indeed, for non-Zeno behaviors diagonal constraints do not increase the expressive power, while they are more powerful in the Zeno case.

The proof is based on CLTLoc rather than TA, since a logic formalism allows for the addition of various properties as further logic constraints or through syntactic substitution in the original formula. Most of the proof is, however, completely independent of the formalism used (TA or CLTLoc), since it is based on the study of properties of regions and intervals of the real line.

As a consequence of the above equivalence result, we also show that Timed ω-Regular languages are closed under left quotient with Timed Regular Languages over finite words. A left quotient operation deletes from a timed ω-language the prefixes belonging to the language of a Timed Automaton that accepts finite timed words.

Whereas it is certainly possible to obtain these results directly for TA, to the best of our knowledge neither of them has been proved or stated in the existing

[1] github.com/fm-polimi/zot
[2] github.com/fm-polimi/qtlsolver

literature on TA. In addition, it is not clear that the syntactic extensions to TA that seem most likely to lead to the desired results (e.g., clock updates to non-zero values [9], silent transitions [8]) can be translated in this case into standard TA.

We also want to point out that, apart from its theoretical interest, the ability of a more general initialization may also be useful in practice. For example, when describing a timed system it may be the case that we want to focus only on "regime" behavior, i.e., one can abstract away from initialization and startup issues, but concentrate only the long-term behavior of the system. For instance, when modeling a real-time operating system we may not want to deal with system bootstrap, but only with a system which has already ended the startup phase and is ready to run. The bootstrap phase can thus be removed by the left-quotient operation, in order to focus only on the modeling and verification of regime behavior.

To summarize, after laying down in Sect. 2 some necessary background notions on CLTLoc and TA—including the fundamental definitions concerning the initialization of clocks in CLTLoc and TA—the paper studies various properties of CLTLoc and TA, and it introduces the following results:

- When arbitrary initialization of clocks is allowed, CLTLoc (with diagonal constraints) is strictly more expressive than TA over Zeno behaviors (Sect. 3, Corollary 2), which in turn are strictly more expressive than standard TA; to obtain these results, Sect. 3 also investigates the expressive power of different types of initializations of clocks for CLTLoc and TA.

- Over non-Zeno timed words, allowing diagonal constraints (i.e., of the form $x \sim y + c$) does not increase the expressive power of CLTLoc, even when clocks are not initialized at 0 (Sect. 5, Theorem 2). This result can be extended to TA (Sect. 7, Corollary 3). When Zeno timed words are considered, only certain kinds of diagonal constraints increase the expressive power of the logic (Sect. 5, Prop. 9).

- Over non-Zeno timed words, allowing for arbitrary initialization of clocks does not increase the expressive power of CLTLoc (Sect. 6, Theorem 3). The same holds for TA (Sect. 7, Corollary 3).

- TA are closed under left quotient (Sect. 8, Theorem 5).

2. Constraint LTL over clocks and Timed Automata

Constraint LTL over clocks [2] (CLTLoc) is a semantic fragment of CLTL [3] where formulae are defined with respect to a finite set AP of atomic propositions, a finite set V of clocks and the set of nonnegative reals $\mathbb{R}_{\geq 0}$. *CLTLoc formulae* are defined as follows:

$$\phi := \ p \mid x \sim c \mid x \sim y + c \mid \phi \wedge \phi \mid \neg\phi \mid \mathbf{X}\phi \mid \mathbf{Y}\phi \mid \phi\mathbf{U}\phi \mid \phi\mathbf{S}\phi$$

where x and y are variables in V, c is a constant in $\mathbb{N}_{\geq 0}$, $\sim$ is a relation of $\{<,=\}$ and $\mathbf{X}$, $\mathbf{Y}$, $\mathbf{U}$ and $\mathbf{S}$ are the usual "next", "previous", "until" and "since" operators of LTL. In the following, formulae of the form $x \sim c$ or $x \sim y + c$ are called "atomic formulae over clocks". The fragment of CLTLoc without atomic formulae over clocks is just LTL. The operators "eventually" $\mathbf{F}$ and "globally" $\mathbf{G}$ may be introduced as customary as abbreviations: $\mathbf{F}(\phi) = \top \mathbf{U} \phi$, and $\mathbf{G}(\phi) = \neg \mathbf{F}(\neg \phi)$. Similarly for the past operators $\mathbf{P}(\phi)$ and $\mathbf{H}(\phi)$, which are the dual of $\mathbf{F}$ and $\mathbf{G}$. Also, formulae such as $x > \alpha$, $x \neq \alpha$, etc., where α can be c or $y + c$, are abbreviations of $\neg(x = \alpha \vee x < \alpha)$, $\neg(x = \alpha)$. Notice that in this paper CLTLoc allows for atomic formulae of the clocks of the form $x \sim y + c$, with $c \geq 0$, whereas in previous works, for simplicity, we only considered the case $c = 0$. We study in Section 5 how this affects the expressive power of the logic.

The semantics of CLTLoc is defined with respect to a strict linear order $(\mathbb{N}_{\geq 0}, <)$ representing positions in time.

Clock values are defined by a mapping $\sigma : \mathbb{N}_{\geq 0} \times V \to \mathbb{R}_{\geq 0}$, assigning, for every position $i \in \mathbb{N}_{\geq 0}$, a value $\sigma(i, x)$ to each clock $x \in V$. Intuitively, a clock x measures the time elapsed since the last time when $x = 0$—i.e., the last "reset" of x. To ensure that time progresses at the same rate for every clock, σ is called a *clock assignment* when it satisfies the following condition: for every position $i \in \mathbb{N}_{\geq 0}$, there exists a "time delay" $\delta_i > 0$ such that for every clock $x \in V$:

$$\sigma(i+1, x) = \begin{cases} \sigma(i, x) + \delta_i, & \text{time progress} \\ 0 & \text{reset } x. \end{cases}$$

For each clock x, its initial value $\sigma(0, x)$ may be any non-negative real. By definition of the sequence of δ_i, it follows that time progress is strongly monotonic. Resets in a clock assignment are represented by value 0. In order to reset a clock x, it suffices to use the formula $x = 0$. For this reason, there is no distinction between the action of resetting a clock x and of testing whether $x = 0$ holds in the clock assignment.

An interpretation for a CLTLoc formula ϕ is a pair (π, σ), where σ is a clock assignment and $\pi : \mathbb{N}_{\geq 0} \to \wp(AP)$ maps every position to a set of atomic propositions. The semantics of ϕ at position $i \geq 0$ over (π, σ) is defined in Figure 1, where we assume that $\sigma(i, c) = c$ holds whenever c is a constant.

A CLTLoc formula ϕ is *satisfiable* if $(\pi, \sigma), 0 \models \phi$ holds, for some (π, σ); in this case, (π, σ) is called a *model* of ϕ, and we write $(\pi, \sigma) \models \phi$.

CLTLoc does not contain quantifiers, but it can express properties beyond counter-free languages. For instance, the language of all the (timed) words such that in every even position there is an occurrence of a (which is not first-order definable [10]) can be expressed by a CLTLoc formula using one clock variable. In the following formula, (the value of) clock z is used to describe the parity of the position of the timed word. So, the clock constraint $z = 0$ (resp. $z > 0$) indicates that the position is odd (resp. even):

$$z = 0 \wedge \mathbf{G}(z > 0 \Rightarrow a) \wedge \mathbf{G}\left(z = 0 \Leftrightarrow \mathbf{X}\left(z > 0\right)\right). \tag{1}$$

$$
\begin{aligned}
(\pi,\sigma),i \models p &\Leftrightarrow p \in \pi(i) \text{ for } p \in AP\\
(\pi,\sigma),i \models x \sim c &\Leftrightarrow \sigma(i,x) \sim \sigma(i,c)\\
(\pi,\sigma),i \models x \sim y + c &\Leftrightarrow \sigma(i,x) \sim \sigma(i,y) + \sigma(i,c)\\
(\pi,\sigma),i \models \neg\phi &\Leftrightarrow (\pi,\sigma),i \not\models \phi\\
(\pi,\sigma),i \models \phi \wedge \psi &\Leftrightarrow (\pi,\sigma),i \models \phi \text{ and } (\pi,\sigma),i \models \psi\\
(\pi,\sigma),i \models \mathbf{X}(\phi) &\Leftrightarrow (\pi,\sigma),i+1 \models \phi\\
(\pi,\sigma),i \models \mathbf{Y}(\phi) &\Leftrightarrow (\pi,\sigma),i-1 \models \phi \text{ and } i > 0\\
(\pi,\sigma),i \models \phi\mathbf{U}\psi &\Leftrightarrow \exists\, j \geq i : (\pi,\sigma),j \models \psi \text{ and}\\
&\quad\;\; \forall\, i \leq n < j, (\pi,\sigma),n \models \phi\\
(\pi,\sigma),i \models \phi\mathbf{S}\psi &\Leftrightarrow \exists\, 0 \leq j \leq i : (\pi,\sigma),j \models \psi \text{ and}\\
&\quad\;\; \forall\, j < n \leq i, (\pi,\sigma),n \models \phi
\end{aligned}
$$

Figure 1: Semantics of CLTLoc.

We now introduce the *timed language* of a CLTLoc formula ϕ.

A *timed ω-word* (sometimes called simply *timed word*) over $\wp(AP)$ is a pair (π_w, τ) where $\pi_w : \mathbb{N}_{>0} \to \wp(AP)$ and the *timed sequence* τ is a monotonic function $\tau : \mathbb{N}_{>0} \to \mathbb{R}_{\geq 0}$ such that, for all $i > 0$, $\tau(i) < \tau(i+1)$ holds (strong monotonicity). The value $\tau(i)$ is called the *timestamp* at position i, $i \in \mathbb{N}_{>0}$. To relate a timed ω-word (π_w, τ) and a CLTLoc model (π, σ) we need to introduce timestamps also in (π, σ). A very simple definition may assume that there is a clock in V which is never reset, except possibly at (the initial) position 0 (if no clock of this kind is in V, one can always just add it), whose values are conventionally assumed to correspond to time stamps. We call such a clock *Now*, verifying the axiom $\mathbf{XG}(\mathit{Now} > 0)$ (i.e., it is different from 0 in every position, except possibly at 0). A timed ω-word (π_w, τ) corresponds to a CLTLoc model (π, σ), denoted as $(\pi_w, \tau) = [(\pi, \sigma)]$, if $\pi_w(i+1) = \pi(i)$ and $\tau(i+1) = \sigma(i, \mathit{Now})$ for all $i \geq 0$. Since π_w and π are clearly the same sequence of elements of $\wp(AP)$, only differing in the set of indexes, in the rest of the work we will abuse the notation and indicate the propositions of the timed words deriving from CLTLoc models with the same symbol π.

Definition 1. The timed language of a CLTLoc formula ϕ is the set of timed ω-words (π, τ) such that there exists a CLTLoc model (π, σ) verifying $(\pi, \sigma) \models \phi$ and $(\pi, \tau) = [(\pi, \sigma)]$.

Definition 2. A timed word (π, τ) is *Zeno* if there is $t \in \mathbb{R}_{>0}$ such that, for all $i \in \mathbb{N}_{>0}$, $\tau(i) \leq t$. Given a Zeno timed word (π, τ), there exists $T \in \mathbb{R}_{>0}$ such that $\lim_{i\to\infty} \tau(i) = T$ (notice that $T > \tau(1)$, since we are assuming strong monotonicity).

We say that a timed language is *Zeno*, if it includes at least one Zeno timed word.

Remark 1. Given a CLTLoc formula ϕ that defines timed language L, one can build a formula ϕ_{NZ}, which defines the timed language L_{NZ} that includes exactly all timed words of L that are non-Zeno. To this end, it is enough for instance to define $\phi_{NZ} \stackrel{\text{def}}{=} \phi \wedge \mathbf{GF}(x_{NZ} > 1 \wedge \mathbf{X}(x_{NZ} = 0))$, where x_{NZ} is a clock that does not appear in ϕ. In the models of ϕ_{NZ}, there are infinitely many positions where clock x_{NZ} is strictly greater than 1 (i.e., at least one time unit has passed since its last reset) and it is restarted by means of a reset.

Definition 3. We say that two CLTLoc formulae are *language equivalent* when they define the same timed language. They are *model equivalent* (or simply equivalent) when they have the same CLTLoc models.

In the following, we generalize language equivalence to any timed formalism, by saying that two formalisms are language equivalent when they define the same family of timed languages.

Definition 4. A clock $x \in V$ is *well-initialized* (w.i.) in an assignment σ if it holds that $\sigma(0, x) = 0$ or $\sigma(0, x) = \sigma(0, Now)$ (recall that by definition $\sigma(0, Now) \geq 0$ holds). A timed ω-word (π, τ) belongs to the *initialized timed language* of ϕ if, and only if, there exists a clock assignment σ such that both $(\pi, \sigma) \models \phi$ and $(\pi, \tau) = [(\pi, \sigma)]$ hold, and each clock $x \in V$ is well-initialized in σ.

We now recall the basic definitions of Timed Automata, in a version allowing so-called *diagonal constraints* [8].

Let X be a finite set of clocks with values in $\mathbb{R}_{\geq 0}$. $\Gamma(X)$ is the set of clock constraints γ over X defined by the syntax $\gamma := x \sim c \mid x \sim y + c \mid \neg\gamma \mid \gamma \wedge \gamma$, where $\sim \in \{<, =\}$, $x, y \in X$ and $c \in \mathbb{N}_{\geq 0}$. A clock valuation is a function $v : X \to \mathbb{R}_{\geq 0}$. We write $v \models \gamma$ when the clock valuation satisfies γ. For $t \in \mathbb{R}_{\geq 0}$, $v + t$ denotes the clock valuation mapping each clock x to value $v(x) + t$—i.e., $(v + t)(x) = v(x) + t$ for all $x \in X$.

A *Timed Automaton* [1] is a tuple $\mathcal{A} = (\Sigma, Q, T, q_0, B)$ where Σ is a finite alphabet (sometimes, without loss of generality, we will consider $\Sigma = 2^{AP}$ for a set AP of atomic propositions), Q is a finite set of control states, $q_0 \in Q$ is the initial state, $B \subseteq Q$ is a subset of control states (corresponding to a Büchi condition) and $T \subseteq Q \times Q \times \Gamma(X) \times \Sigma \times 2^X$ is a set of transitions. Thus, a transition has the form $q \xrightarrow{\gamma,a,S} q'$ where $q, q' \in Q$, γ is a clock constraint of $\Gamma(X)$, $a \in \Sigma$, and S is a set of clocks to be reset. Two transitions $q \xrightarrow{\gamma,a,S} q'$ and $p \xrightarrow{\gamma',b,P} p'$ of T are *consecutive* when $q' = p$. A pair (q, v), where $q \in Q$ and $v : X \to \mathbb{R}_{\geq 0}$ is a clock valuation, is a *configuration* of $\mathcal{A}$. A *run* ρ of $\mathcal{A}$ over a timed ω-word $(\pi, \tau) \in (\Sigma \times \mathbb{R}_{\geq 0})^\omega$ is an infinite sequence of configurations $(q_{i_0}, v_0) \xrightarrow[\tau(1)]{\pi(1)} (q_{i_1}, v_1) \xrightarrow[\tau(2)]{\pi(2)} (q_{i_2}, v_2) \ldots$, satisfying the following three constraints:

- $q_{i_0} = q_0$;

- $q_{i_0} \xrightarrow{\gamma_1,\pi(1),S_1} q_{i_1} \xrightarrow{\gamma_2,\pi(2),S_2} q_{i_2} \dots$ is a sequence of consecutive transitions and, for all $i > 0$, $v_{i-1} + \tau(i) - \tau(i-1) \models \gamma_i$ (conventionally $\tau(0) = 0$);
- for all $x \in X$, $v_0(x) = 0$ and for all $i > 0$ either $v_i(x) = 0$, if $x \in S_i$, or $v_i(x) = v_{i-1}(x) + \tau(i) - \tau(i-1)$ otherwise.

Let $\mathit{inf}(\rho)$ be the set of control states $q \in Q$ such that $q = q_{i_j}$ for infinitely many positions $j \geq 0$ of ρ. A run is *accepting* when $\mathit{inf}(\rho) \cap B \neq \emptyset$—i.e., when a Büchi condition holds.

We can extend the notion of initialization in TA by allowing for some clocks to have a value different from zero in the initial state. More precisely, an *arbitrarily initialized Timed Automaton* $\mathcal{A}$ (a.i. TA for short) is a tuple $\mathcal{A} = (\Sigma, Q, T, q_0, B, N)$ where Σ, Q, T, q_0, B are as before, and N is a set of clocks ($N \subseteq X$) such that for each $x \in N$, for every run of $\mathcal{A}$, the initial state (q_0, v_0) of the run must satisfy $v_0(x) > 0$—i.e., the initial value of each $x \in N$ is greater than 0. We call a TA *non-initialized* if $N = X$, and *initialized* if $N = \emptyset$—obviously, an initialized TA is just a "classic" TA.

Remark. Traditionally, the usual definition of timed words allows the timestamp $\tau(1)$ at the first point to be 0. However, for technical reasons, in the rest of the paper we restrict the timestamp $\tau(1)$ of every timed word to be strictly greater than 0 (hence, $\sigma(0, \mathit{Now}) > 0$).

The following result states that initialized timed languages of CLTLoc are the same of timed regular languages (i.e., the timed languages recognized by Timed Automata).

Theorem 1. *The class of initialized timed languages associated with CLTLoc formulae coincides with the class of timed ω-regular languages.*

Proof sketch. The statement was proved in [12, Theorem 4] for a version of CLTLoc in which atomic formulae on the clocks of the form $x \sim y + c$ are admissible only if $c = 0$, but in the following we extend it to the general case.

In fact, given a (initialized) TA which includes diagonal constraints of the form $x \sim y + c$, there is an equivalent (initialized) diagonal-free TA [8], to which Theorem 4 of [12] can be applied to produce an equivalent CLTLoc formula. Conversely, given a CLTLoc formula ϕ that includes diagonal constraints, it is easy to build another CLTLoc formula ϕ' which does not include constraints of the form $x \sim y + c$, with $c > 0$, and such that the w.i. timed language of ϕ' is the same as the one of ϕ. To show this, consider that the negation of a constraint of the form $x < y + c$, with $c > 0$, is equivalent to the formula:

$$(x > 0)\mathbf{S}(y = 0 \wedge x \geq c)$$

stating that $x \geq y + c$ holds if x was never reset since a time instant when both y was reset and $x \geq c$ held (so x is still greater or equal to $y + c$ at the current instant). Notice that if y was never reset, then $y = \mathit{Now}$ holds at position 0 and the above formula is false: this is correct since $x = \mathit{Now}$ or $x = 0$ hold at 0, hence $x < y + c$ holds in both cases (since $c > 0$).

Therefore, $\neg((x > 0)\mathbf{S}(y = 0 \wedge x \geq c))$ can replace a constraint of the form $x < y + c$.

Similarly, every constraint of the form $x = y + c$, with $c > 0$, is instead replaced by the formula:

$$(x > 0 \wedge y > 0)\mathbf{S}(y = 0 \wedge x = c)$$

stating that both x and y were never reset since a time instant when y was reset and $x = c$ (so the value of x is just $y + c$ at the current instant).

Theorem 4 of [12] can then be applied to formula ϕ' resulting by the above replacements. Clearly, the atomic formulae over clocks occurring in ϕ' can only be of the forms $x \sim c$ or $x \sim y$. □

Remark 2. Temporal logic languages, such as CLTLoc, are customarily defined over infinite models, whereas the semantics of TA may also consider finite timed words (indeed, Section 8 considers the quotient of timed ω-languages with respect to timed languages over finite words). However, the results presented in the next sections for non-Zeno languages of timed ω-words are also valid when languages are restricted to finite timed words. In fact, it is always possible to interpret any finite timed word $(\bar{\pi}, \bar{\tau})$, with $\bar{\pi} : \{1, \ldots, n\} \to \wp(AP)$ and $\bar{\tau} : \{1, \ldots, n\} \to \mathbb{R}_{\geq 0}$, as the prefix of the non-Zeno timed ω-words (π, τ) where, for all $j > n$, $\pi(j) = \emptyset$ holds and $\tau(j)$ is arbitrary.

3. Expressiveness of Constraint LTL over clocks and arbitrarily initialized Timed Automata

Consider the (Zeno) language L_{ni}, taken from [8] (Example 22), made of timed words (π, τ) over the alphabet $\{a\}$ such that, for each $i \in \mathbb{N}_{>0}$ and some fixed $0 < \gamma < 1$, we have that $\pi(i) = \{a\}$ and $\tau(i) < \tau(1) + 1 - \gamma$ hold. In other words the sequence of timestamps accumulates to a value $\tau(1) + 1 - \gamma$ that depends on $\tau(1)$.

We have the following proposition.

Proposition 1 ([8]). *Language L_{ni} is not timed regular.*

However, it is easy to see that CLTLoc formula $\mathbf{G}(a \wedge 0 < x < 1)$ defines exactly L_{ni}. In fact, it is not possible for the timestamp to go beyond $\tau(1) + 1 - \sigma(0, x)$, because $0 < \sigma(i, x) < 1$, for all $i \geq 0$, and $\sigma(0, x) = \gamma$.

In addition, consider the slight variation of language L_{ni}, called L_{ni1}, such that for all $i \in \mathbb{N}_{>0}$ it holds that $\tau(i) < 1 - \gamma$. The timed words in L_{ni1} are such that the sequence of timestamps accumulates to a value $1 - \gamma$, which, unlike L_{ni}, does not depend on the first timestamp $\tau(1)$.

The following proposition holds.

Proposition 2. *Language L_{ni1} is not timed regular.*

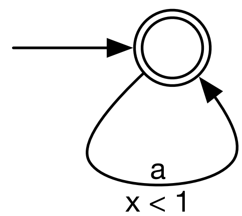

Figure 2: Non-initialized TA that recognizes language L_{ni1}.

Proof. If L_{ni1} were timed regular, then there would exist an initialized TA $\mathcal{A}_{ni1}$ that accepts L_{ni1}. From $\mathcal{A}_{ni1}$ we could then build an initialized TA recognizing L_{ni} in the following way. Let $q_{0,ni1}$ be the initial state of $\mathcal{A}_{ni1}$; we introduce a new initial state $q_{0,ni}$, from which a transition originates that reads symbol a and enters state $q_{0,ni1}$, resetting all clocks of $\mathcal{A}_{ni1}$ in doing so; after entering q_{ni1}, the automaton behaves exactly as $\mathcal{A}_{ni1}$. Such initialized TA would accept language L_{ni}, a contradiction. □

However, the non-initialized TA of Figure 2 accepts L_{ni1}; in addition, L_{ni1} is the language associated with CLTLoc formula $\mathbf{G}(a \wedge 0 < x < 1 \wedge \mathit{Now} < x)$, where clock x is not well-initialized.

Consider now the (non-Zeno) language L_1 of the timed words (π, τ) over the alphabet $\{a, b\}$ such that there exists an occurrence of a that occurs at timestamp 1—i.e., there exists an i such that $\pi(i) = \{a\}$ and $\tau(i) = 1$ hold. It is clear that L_1 is timed regular, by using, e.g., an initialized TA with one clock x by checking that if $x = 1$ then there is an a. However, a TA where all clocks are not initialized cannot recognize L_1, as stated in the next proposition.

Proposition 3. *There is no non-initialized TA that accepts the timed regular language L_1.*

Proof. We show that a non-initialized TA $\mathcal{A}_1$ that accepts a timed word (π_1, τ_1) of L_1 must also accept a timed word that does not belong to L_1. Consider the run ρ_1 of $\mathcal{A}_1$ corresponding to timed word (π_1, τ_1). Let $\alpha \in \mathbb{N}_{>0}$ be such that $\pi_1(\alpha) = a$ and $\tau_1(\alpha) = 1$; that is, $(q_{i_{\alpha-1}}, v_{\alpha-1}) \xrightarrow[\tau_1(\alpha)]{\pi_1(\alpha)} (q_{i_\alpha}, v_\alpha)$ is part of run ρ_1, the corresponding transition is $q_{i_{\alpha-1}} \xrightarrow{\gamma_\alpha, \pi_1(\alpha), S_\alpha} q_{i_\alpha}$ and the clock assignment $v_{\alpha-1} + \tau_1(\alpha) - \tau_1(\alpha - 1)$ satisfies γ_α. For each clock x of $\mathcal{A}_1$, let c_x be the value $\lfloor v_{\alpha-1}(x) + \tau_1(\alpha) - \tau_1(\alpha - 1) \rfloor$. Two cases hold: either $c_x < v_{\alpha-1}(x) + \tau_1(\alpha) - \tau_1(\alpha-1) < c_x + 1$, with $c_x \geq 0$, or $v_{\alpha-1}(x) + \tau_1(\alpha) - \tau_1(\alpha-1) = c_x$, with $c_x > 1$. In fact, if $v_{\alpha-1}(x) + \tau_1(\alpha) - \tau_1(\alpha - 1) = 1$ were true, then $v_{\alpha-1}(x) = \tau_1(\alpha - 1)$ would hold because $\tau_1(\alpha) = 1$. Since $v_j(x) = v_0(x) + \sum_{i=1}^{j}(\tau_1(i) - \tau_1(i-1))$, where $\tau(0) = 0$, then $v_{\alpha-1}(x) = \tau_1(\alpha - 1)$ if $v_0(x) = 0$, which is impossible since $\mathcal{A}_1$ is non-initialized. Second, $v_{\alpha-1}(x) + \tau_1(\alpha) - \tau_1(\alpha - 1) = 0$ cannot be true because $v_0(x) > 0$ holds and time is strictly monotonic ($\tau(i) > 0$ holds for $i > 0$); hence, $v_{\alpha-1}(x) + \tau_1(\alpha) - \tau_1(\alpha - 1) > 0$ holds.

Moreover, a clock x for which $v_\alpha(x) = c_x$ holds cannot satisfy $v_i(x) = d$, where $d \in \mathbb{N}_{\geq 0}$ and $0 < i < \alpha$. In fact, if $v_i(x) = d$ were true, then $0 < v_\alpha(x) - d = \tau_1(\alpha) - \tau_1(i) < 1$ would hold, since $0 < \tau_1(\alpha) - \tau_1(1) < 1$ is true. Hence, $v_\alpha(x) = d + \tau_1(\alpha) - \tau_1(i)$ cannot be an integer. So, for all $i < \alpha$, guard γ_i in ρ_1 cannot include a constraint of the form $x = d$, with $d \geq 0$.

We now define a timed word (π_1, τ_1') where τ_1' is the same as τ_1 except for $\tau_1'(\alpha)$ which is equal to $\tau(\alpha) + \epsilon$ for some $\epsilon < 1$ and $\epsilon \in \mathbb{R}_{>0}$. The timed word (π_1, τ_1'), where $\tau_1'(i) = \tau_1(i)$ if $i \neq \alpha$, and $\tau_1'(\alpha) = \tau_1(\alpha) + \epsilon$, has also an accepting run ρ_1', where every clock x whose value $v_{\alpha-1}(x) + \tau_1(\alpha) - \tau_1(\alpha - 1)$ is an integer at α is shifted negatively of ϵ. More precisely, ρ_1' is such that: (i) the sequence of control states is the same as in ρ_1; and (ii) the sequence of clock valuations is such that, for each clock x for which $v_{\alpha-1}(x) + \tau_1(\alpha) - \tau_1(\alpha - 1) = c_x$ (resp. $c_x - 1 < v_{\alpha-1}(x) + \tau_1(\alpha) - \tau_1(\alpha - 1) \leq c_x$) holds, for each $0 \leq i \leq \alpha$ it holds that $v_i'(x) = v_i(x) - \epsilon$ (resp. $v_i'(x) = v_i(x)$). In addition, ϵ can be chosen small enough that, for each clock such that $c_x < v_{\alpha-1}(x) + \tau_1(\alpha) - \tau_1(\alpha - 1) < c_x + 1$ holds, $v_{\alpha-1}(x) + \tau_1(\alpha) - \tau_1(\alpha - 1) + \epsilon < c_x + 1$ also holds. Finally, if clock constraint $x = y + c$ holds for some clock valuation $v_{i-1} + \tau_1(i) - \tau_1(i - 1)$, then $v_{i-1}(x) + \tau_1(i) - \tau_1(i-1)$ is a value in $\mathbb{N}_{>0}$ if, and only if, $v_{i-1}(y) + \tau_1(i) - \tau_1(i-1)$ also is; hence, in this case either both the values of x and y are offset by ϵ, or none is, so their relationship is preserved. Therefore, the timed word (π_1, τ_1') is accepted by $\mathcal{A}_1$ but it is not in L_1. □

From Proposition 2 and Proposition 3, since there is a non-initialized TA that accepts L_{ni1} and an initialized TA that accepts L_1, we have the following result.

Corollary 1. *The family of timed languages recognizable by non-initialized TA is incomparable with the family of timed regular languages.*

We have shown above that timed language L_{ni} can be defined through a CLTLoc formula. However, we have the following result.

Proposition 4. *There is no arbitrarily initialized TA that accepts timed language L_{ni}.*

Proof. The proof is by contradiction. Assume there is an a.i. TA $\mathcal{A}_{ni}$ that accepts language L_{ni}—i.e., all and only the timed words (π, τ), where $\pi(i) = \{a\}$, for all $i > 0$, and $0 < \tau(1) + 1 - \lim_{i\to\infty} \tau(i) < 1$, are accepted by $\mathcal{A}_{ni}$. We show that $\mathcal{A}_{ni}$ also accepts a timed word (π, τ') such that $\lim_{i\to\infty} \tau'(i) = \tau'(1) + 1$.

Let $\gamma = \lim_{i\to\infty}(\tau(1) + 1 - \tau(i))$; intuitively, $\lim_{i\to\infty}(\tau(i) - \tau(1))$ is the "duration" of the timed word, and γ is the gap from the latter to 1 (which is obviously less than 1). Let us call $C_{\mathcal{A}_{ni}}$ the maximum constant that appears in the guards of $\mathcal{A}_{ni}$, and consider a timed word (π, τ) that belongs to L_{ni} such that $\tau(1) = C_{\mathcal{A}_{ni}} + 1$ (which obviously exists). Let $\rho = (q_{i_0}, v_0) \xrightarrow[\tau(1)]{\pi(1)} (q_{i_1}, v_1) \ldots$ be the run corresponding to (π, τ). The guard γ_1 of the first transition taken $q_{i_0} \xrightarrow{\gamma_1, \pi(1), S_1} q_{i_1}$ clearly must allow $x > C_{\mathcal{A}_{ni}}$ for all clocks of $\mathcal{A}_{ni}$. Since

$\tau(1) = C_{\mathcal{A}_{ni}} + 1$, and the value of clock x used to evaluate γ_1 is $v_0(x) + \tau(1)$, then $v_0(x) + \tau(1) > C_{\mathcal{A}_{ni}}$ must be true. All subsequent transitions taken in the run must be such that, for each clock x, either constraint $x > C_{\mathcal{A}_{ni}}$ holds (if the clock is never reset), or constraint $x < 1$ does, if the clock has been previously reset. Indeed, if clock x is reset before (or at) a position i, then for every $j > i$, it holds that $v_j(x) = \tau(j) - \tau(i) < 1 - \gamma$ because $\lim_{i\to\infty}(\tau(i) - \tau(1)) = 1 - \gamma$. As a consequence, the timed word (π, τ') such that $\tau'(1) = \tau(1)$ and $\tau'(i) = \tau(i) + \gamma$ for all $i > 0$ is also accepted by $\mathcal{A}_{ni}$, as it allows for the same sequence of transitions as ρ. However, since $\lim_{i\to\infty} \tau(i) = \tau(1) + 1 - \gamma$ holds, then $\lim_{i\to\infty} \tau'(i) = \tau'(1) + 1$ also holds. However, (π, τ') does not belong to L_{ni}. □

Proposition 5. *Given an arbitrarily initialized TA $\mathcal{A} = (\Sigma, Q, T, q_0, B, N)$, there exists a language equivalent CLTLoc formula which includes diagonal constraints on Now.*

Proof. Let us consider the translation of automaton $\mathcal{A}$ into CLTLoc formula $\phi_{\mathcal{A}}$ defined in [12, Section 4], slightly modified to account for the fact that $\mathcal{A}$ can now include diagonal constraints. More precisely, in the modified translation each constraint $x \sim y + c$ in guards γ becomes the following CLTLoc formula (where x_1, x_2 is a pair of CLTLoc clocks that are used to capture the value of the corresponding clock x of TA $\mathcal{A}$, and x_{12} at every position acts as a "selector" of which of x_1, x_2 is to be used to express the guard on x; similarly for y_1, y_2, y_{12}). In particular, when $x_{12} = 0$ the value of clock x corresponds to the value of clock x_1, while it corresponds to the one of x_2 when $x_{12} > 0$. Two CLTLoc clocks are needed to represent one clock in a TA because CLTLoc formulae cannot distinguish between clock resets and the tests of clock values. For this reason, it is not possible to write a formula that expresses simultaneous test and reset operations on a single clock (as this would yield a contradiction such as $x > 1 \wedge x = 0$), but they can be represented by using two distinct clocks, working alternatively. For instance, the clock constraint $x > 1$ can be expressed by the formula $(x_{12} = 0 \wedge x_1 > 1) \vee (x_{12} > 0 \wedge x_2 > 1)$, together with a formula that forces resets of x_1 and x_2 to alternate and that constrains x_{12} to be 0, when the last reset was $x_1 = 0$ (i.e., clock x_1 is active), or not equal to 0, when the last reset was $x_2 = 0$ (i.e., clock x_2 is active). The CLTLoc formula translating clock constraints of the form $x \sim y + c$ is therefore:

$$
\begin{aligned}
&x_{12} = 0 \wedge y_{12} = 0 \wedge x_1 \sim y_1 + c \quad \vee \\
&x_{12} = 0 \wedge y_{12} > 0 \wedge x_1 \sim y_2 + c \quad \vee \\
&x_{12} > 0 \wedge y_{12} = 0 \wedge x_2 \sim y_1 + c \quad \vee \\
&x_{12} > 0 \wedge y_{12} > 0 \wedge x_2 \sim y_2 + c.
\end{aligned}
$$

Let ρ be a run of $\mathcal{A}$ and (π, σ) a model of $\phi_{\mathcal{A}}$. In the proof of [12, Section 4], the position 0 of (π, σ) is defined in order to represent the second configuration (q_1, v_1) in ρ where either $v_1(x) = 0$, if the first transition resets x, or $v_1(x) = \tau(1)$. Recall also that $\tau(1)$ is equal to $\sigma(0, \mathit{Now})$. Since $\mathcal{A}$ is arbitrarily initialized, to make clocks in $\mathcal{A}$ and $\phi_{\mathcal{A}}$ initialized in the same way it is enough to introduce in $\phi_{\mathcal{A}}$ the constraint

$$x_i = 0 \vee Now < x_i \tag{2}$$

(i.e., which holds at position 0) for each clock $x \in N$ and $i \in \{1, 2\}$, and the constraint

$$x_i = 0 \vee Now = x_i \tag{3}$$

for each $x \in X - N, i \in \{1, 2\}$. It is easy to see that, also in the case of arbitrarily initialized automata, if the clocks are initialized in the same way in $\mathcal{A}$ and $\phi_{\mathcal{A}}$, then the language associated with $\phi_{\mathcal{A}}$ is accepted by $\mathcal{A}$. The case $x \in X - N$ is dealt with as in [12, Section 4]: if $x_i = 0$ holds at position 0 then $v_1(x) = 0$ holds in ρ, whereas for $Now = x_i$ it holds that $v_1(x) = \tau(1)$, because $v_0(x) = 0$ holds, since x is well-initialized. In the other case, for $Now < x_i$, it holds that $\sigma(0, x_i) - \sigma(0, Now) = v_0(x)$, where $v_0(x) > 0$ because $x \in N$. □

Then, from propositions 4 and 5 we have the following result.

Corollary 2. *The class of languages that can be defined through CLTLoc formulae is strictly larger than the class of languages recognizable through a.i. TA.*

4. Preliminaries on regions

The proofs of the main results of this paper need the additional definitions and properties introduced in this section.

Let X be a set of clocks, and $C \in \mathbb{N}_{\geq 0}$ a constant. A *clock region* [1] REG is a set of clock valuations that obeys a maximal consistent set of constraints on clocks of the form $x \sim c$, $x \sim y + c$, and their negations, with $\sim \in \{<, =\}$, $c \in \mathbb{N}_{\geq 0}$, $c \leq C$, $x, y \in X$. The set $\mathfrak{R}(X, C)$ of clock regions defines a finite partition of the $|X|$-dimensional space $\mathbb{R}^{|X|}_{\geq 0}$.

We also introduce the notion of *one-dimensional region*, which will be mostly useful in Section 6. An *open interval* (α, β), for real numbers $0 \leq \alpha < \beta$, is the set of real numbers ν such that $\alpha < \nu < \beta$. With a convenient abuse of notation, the singular point α in the following is usually denoted as an open interval (α, α)—also called a *punctual interval*. If $I = (\alpha, \beta)$, for real numbers $0 \leq \alpha \leq \beta$, then define $\sup(I) = \alpha$ and $\inf(I) = \beta$.

A *one-dimensional region* (1D-region) is an open interval R of the form $(n - 1, n)$ for some $n \in \mathbb{N}_{>0}$, $1 \leq n \leq C$, or a punctual interval (n, n) for some $0 \leq n \leq C$, or the open interval $(C, +\infty)$.

Given a 1D-region R and a clock x, we write $R(x)$ to denote that clock x is in region R, i.e., either $\inf(R) < x < \sup(R)$ or $x = \inf(R) = \sup(R)$ hold. When no confusion can arise on clock x, we denote $R(x)$ as R.

We can define the *time-successor* relationship among regions as in [1]; also, we can build a nondeterministic Büchi automaton $TS(X, C)$ (hence, with no clocks), called *region automaton*, which captures the time-successor relationship between regions, in a similar vein as done in [2]. We may assume that every state in $TS(X,C)$ represents a region and it is both an initial and a final state. $TS(X,C)$ may be encoded in CLTLoc by means of a formula based on a LTL encoding of a Büchi automaton. To this end, before sketching a construction that is useful to provide a suitable upper bound on its size, we first present the notion of "region clocks", which are freshly introduced in this formula and do not appear in the automaton. Since a region is a set of constraints on clocks, the final CLTLoc formula represents the state space of the region automaton with a set Z_X of additional (well-initialized) clocks, called *region clocks*: each region clock represents the truth value of a clock constraint. The clock is 0 if the constraint holds, greater than 0 if it does not. Hence, there is a region clock for each constraint of the form $x \sim c$, $x \sim y+c$, with $x, y \in X$, constant $c \in \mathbb{N}_{\geq 0}$ (with $c \leq C$) and $\sim \in \{<, =\}$. Therefore, the region clocks in Z_X are denoted as $z_{[x \sim c]}$ or as $z_{[x \sim y+c]}$, with the obvious meaning. For the sake of readability, the clock constraint $z_{[x \sim c]} = 0$ is written as $[x \sim c]$ (obviously, $\neg[x \sim c]$ indicates $z_{[x \sim c]} > 0$).

Given a 1D-region R and a clock x, we indicate by $[\![R(x)]\!]$ the maximal consistent set of constraints on the region clocks of Z_X defining $R(x)$. For instance, if $R(x)$ is $3 < x < 4$, then $[\![3 < x < 4]\!]$ is $[x < 4] \wedge \neg[x < 3] \wedge \neg[x = 3]$—plus all constraints, such as $[x < 5]$, which are implied by these, and which are not shown for the sake of brevity. Similarly, one can define $[\![x = 1]\!]$ as $\neg[x = 0] \wedge \neg[x < 1] \wedge [x = 1] \wedge [x < 2] \wedge \ldots$. We can extend this notation also to diagonal constraints. For instance, a constraint of the form $x < y < x + 1$ can be encoded as the following formula $[\![x < y < x + 1]\!]$: $[y < x + 1] \wedge \neg[y < x] \wedge [y < x + 2] \wedge \neg[y = x + 1] \wedge \neg[y = x] \wedge \ldots$.

A CLTLoc formula, over the region clocks Z_X, describing the region automaton $TS(X,C)$ is denoted by $\Theta_{Z_X,C}$. Intuitively, a model of $\Theta_{Z_X,C}$ symbolically represents the regions that clocks X satisfy and their evolution determined by the elapsing of time. The representation is symbolical because $\Theta_{Z_X,C}$, and hence its models, does not constrain the actual value of clocks X. We show a fragment of $\Theta_{Z_X,C}$, since the whole formula is tedious to define completely. Let $\Xi^d_{x,y}$ be the following formula, for $0 \leq d \leq C - 2$:

$$
\begin{aligned}
&([\![d < x < d+1]\!] \wedge [\![d < y < d+2]\!]) \vee \\
&(([\![x = d+1]\!] \vee [\![d+1 < x < d+2]\!]) \wedge [\![d+1 < y < d+2]\!])\,.
\end{aligned}
$$

Intuitively, $\Xi^d_{x,y}$ describes the regions of x and y that are reachable in one step when both x and y start from $(0,1)$ and the time progress is equal to $d + \epsilon$, for some $\epsilon \in (0,1)$ and $0 \leq d \leq C - 2$.

The fragment we define (where formulae Λ^d cover the cases where $d = C - 1$ and $d = C$, which are not shown for brevity) describes the time successors of a state where every clock is in the interval $(0,1)$; the general case can be dealt with by multiple instances of this formula, by considering all possible regions of

x and y rather than only $(0, 1)$.

$$\mathbf{G}\left(\begin{array}{l}\displaystyle\bigvee_{0\le d\le C-2}\ \bigwedge_{x,y\in X}\left(\begin{array}{l}(\llbracket 0<x<1\rrbracket \wedge \llbracket 0<y<1\rrbracket \wedge \llbracket x<y\rrbracket) \Rightarrow \\ \mathbf{X}\left(\begin{array}{l}\left(\varXi^{d}_{x,y} \wedge \llbracket y-1<x<y\rrbracket\right) \ \vee \\ (\llbracket x=0\rrbracket \wedge \llbracket y=0\rrbracket \wedge \llbracket x=y\rrbracket) \ \vee \\ \left(\llbracket y=0\rrbracket \wedge \left(\begin{array}{ll}(\llbracket d<x<d+1\rrbracket \wedge \llbracket y+d<x<y+d+1\rrbracket) & \vee \\ (\llbracket x=d+1\rrbracket \wedge \llbracket x=y+d+1\rrbracket) & \vee \\ (\llbracket d+1<x<d+2\rrbracket \wedge \llbracket y+d+1<x<y+d+2\rrbracket) & \end{array}\right)\right) \ \vee \\ \llbracket x=0\rrbracket \wedge \ldots \end{array}\right)\end{array}\right) \\ \displaystyle\vee \bigvee_{C-1\le d\le C} \Lambda^{d}\end{array}\right).$$

Notice that the set AP of atomic propositions used in $\Theta_{Z_X,C}$ is empty. It is easy to see that, since formula $\Theta_{Z_X,C}$ does not explicitly enumerate all clock regions, but it only considers relationships between pairs of clocks, its size is polynomial in the number of clocks and in the maximum constant C when considering a unary encoding of C—more precisely, it is $O(|X|^2C^4)$. The size is instead exponential when considering a binary encoding of C.

1D-subregions

An interval I is a *one-dimensional subregion* (1D-subregion) if there exists a 1D-region R such that $I \subseteq R$, e.g., $(2.14, 2.71) \subseteq (2, 3)$. Interval I is then also called a subregion of R. Notice that I must be either open or punctual.

For every real number $\delta > 0$, we define the δ-successors of a 1D-subregion.

Definition 5. For all real number $\delta > 0$, for all 1D-subregions I, let $I \oplus \delta$ be the interval (not necessarily a subregion) $(\inf(I) + \delta, \sup(I) + \delta)$. If $I \oplus \delta$ is a 1D-subregion I', then we call I' the δ-successor of I and we write $I \rightsquigarrow_\delta I'$.

Example: $(0.2, 0.7) \rightsquigarrow_{0.2} (0.4, 0.9) \rightsquigarrow_{0.1} (0.5, 1)$; also, the δ-successor of $(0.4, 0.9)$ is not defined for $0.1 < \delta < 0.6$, for $1.1 < \delta < 1.6$, and so on.

Notice that if $I \rightsquigarrow_\delta I'$, then I, I' have the same size, i.e., $\sup(I) - \inf(I) = \sup(I') - \inf(I')$. The following propositions are immediate from the definition of $\rightsquigarrow$.

Proposition 6. *Let I, I'' be 1D-subregions and let δ', δ'' be positive real numbers.*

1. *If there is 1D-subregion I' such that $I \rightsquigarrow_{\delta'} I' \rightsquigarrow_{\delta''} I''$, then $I \rightsquigarrow_{\delta'+\delta''} I''$.*
2. *If $I \rightsquigarrow_{\delta'+\delta''} I''$ and there is a 1D-region R such that both $I \subseteq R, I'' \subseteq R$, then there exists $I' \subseteq R$ such that $I \rightsquigarrow_{\delta'} I' \rightsquigarrow_{\delta''} I''$.*
3. *If I is of the form (α, α), then $I'' = I \oplus \delta'$ is a 1D-subregion, and it holds that $I \rightsquigarrow_{\delta'} I''$.*

The following statement is an obvious consequence of the definition of $\rightsquigarrow$.

Statement 1. *Let $\mathcal{P}$ be the partition $\{(0, 1-\eta), (1-\eta, 1-\eta), (1-\eta, 1)\}$, where $0 < \eta < 1$ is a real number. For all $I_0 \in \mathcal{P}$, for all 1D-subregion I, for all $n \in \mathbb{N}_{\ge 0}$, if $I_0 \rightsquigarrow_{n+\eta} I$, then:*

1. *The following table shows all possible cases for I and I_0:*

I_0	I
$(0, 1-\eta)$	$(n+\eta, n+1) \subseteq (n, n+1)$
$(1-\eta, 1-\eta)$	$(n+1, n+1)$
$(1-\eta, 1)$	$(n+1, n+1+\eta) \subseteq (n+1, n+2)$

2. *For all nonempty $I'_0 \subseteq I_0$, there exists one, and only one, 1D-subregion I' such that $I'_0 \rightsquigarrow_{n+\eta} I'$; moreover, $I' \subseteq I$.*

The *strict dominance* relation among 1D-regions is the total order $\prec$ on 1D-regions defined by:

$$(0,0) \prec (0,1) \prec (1,1) \prec \cdots \prec (C,C) \prec (C,+\infty)$$

The reflexive closure of $\prec$ is denoted by $\preceq$ and it is called *non-strict dominance.*

It is also possible to extend the dominance relation to 1D-subregions. If I, I' are 1D-subregions, then $I \prec I'$ if $\sup(I) \leq \inf(I')$ and $\inf(I) < \sup(I')$. For instance, $(0.2, 0.5) \prec (1.3, 1.7)$ but also $(0.2, 0.5) \prec (0.5, 0.7)$ and $(0.5, 0.5) \prec (0.5, 0.7)$, whereas $(0.5, 0.5) \not\prec (0.5, 0.5)$.

Partitioning of 1D-regions

To prove the main results of this paper, it is fundamental to consider (finite or infinite) partitions of the 1D-region $R_0 = (0,1)$, e.g.,

$$\{(0, 0.3), (0.3, 0.3), (0.3, 0.8), (0.8, 0.8), (0.8, 1)\}.$$

We are actually interested in a special case of partition, defined next. Let $\overrightarrow{\Delta} = \Delta_1 \Delta_2 \ldots$, be a finite or infinite sequence of positive real numbers. The sequence is called a *temporal sequence* if it is monotonically increasing (i.e., $\Delta_1 < \Delta_2 < \ldots$); if $\overrightarrow{\Delta}$ is infinite and $\lim_{i \to +\infty} \Delta_i$ is a finite real value, then $\overrightarrow{\Delta}$ is called a *Zeno* sequence.

Denote with $\langle w \rangle$ the fractional part of a real value w and with $\lfloor w \rfloor$ its integer part.

Definition 6. Given an integer constant $C > 0$ and a temporal sequence $\overrightarrow{\Delta}$, *the maximal partition $\mathcal{P}_{\overrightarrow{\Delta}}$* of the interval $(0,1)$ is the partition in 1D-subregions including all, and only, the singular points $1 - \langle \Delta_j \rangle < 1$, for all $\Delta_j < C$, and $1 - \langle \lim_{i \to +\infty} \Delta_j \rangle < 1$ if $\lim_{i \to +\infty} \Delta_j < C$.

In the above definition, if $\lim_{i \to +\infty} \Delta_i$ is finite, with $\langle \lim_{i \to +\infty} \Delta_i \rangle = \eta > 0$, then the sequence is Zeno and $(1-\eta, 1-\eta)$ is called the *limit interval.* Notice that, if there is a punctual interval $(1-\eta, 1-\eta) \in \mathcal{P}_{\overrightarrow{\Delta}}$, for some $0 < \eta < 1$, then there exists $h \in \mathbb{N}_{\geq 0}$, with $h + \eta < C$, such that either $\Delta_j = h + \eta$ for some $j \geq 1$, or $\lim_{i \to +\infty} \Delta_i = h + \eta$. Also, by definition, $(0,0)$ and $(1,1)$ do not belong to the maximal partition.

Example 1. For instance, $\mathcal{P}_{\overrightarrow{\Delta}} = \{(0, 0.2), (0.2, 0.2), (0.2, 0.9)(0.9, 0.9)(0.9, 1)\}$ is the maximal partition of $(0, 1)$ for $\overrightarrow{\Delta} = \Delta_1\Delta_2$, with $\Delta_1 = 1.8, \Delta_2 = 5.1$ (with $C \geq 6$). In fact, the fractional parts $\langle\Delta_1\rangle$ and $\langle\Delta_2\rangle$ are, respectively, 0.8 and 0.1 and the singular points in $\mathcal{P}_{\overrightarrow{\Delta}}$ are $0.2 = 1 - 0.8$ and $0.9 = 1 - 0.1$. Notice that, e.g., $(0.2, 0.2) \rightsquigarrow_{\Delta_1} (2, 2)$ and $(0.9, 0.9) \rightsquigarrow_{\Delta_2} (6, 6)$—i.e., from every punctual 1D-subregion in the maximal partition it is possible to reach a punctual 1D-region by a delay Δ_1 or a delay Δ_2.

Consider the Zeno temporal sequence $\overrightarrow{\Delta} = 1.1, 1.11, 1.111, 1.1111, \ldots$; the corresponding maximal partition contains all the following punctual intervals:

$$(0.9, 0.9), (0.89, 0.89), (0.889, 0.889), (0.8889, 0.8889), \ldots$$

together with the limit interval $(8/9, 8/9)$ since the temporal sequence converges to $1 + 1/9$.

An immediate consequence of the definition is that for every temporal sequence the maximal partition is unique. Moreover, the number of punctual intervals in the maximal partition for a finite temporal sequence of length m is obviously at most the number m itself; it can be smaller than m when two fractional parts are equal, in the sense that $\Delta_j = n+\eta$ and $\Delta_i = k+\eta$, for some $i \neq j$, $n \neq k \in \mathbb{N}_{\geq 0}$, $0 < \eta < 1$. Therefore, the maximal partition for a temporal sequence of length m has at most $2m + 1$ elements, as it can immediately be verified.

Let $\mathcal{P}_{\overrightarrow{\Delta}}$ be a maximal partition for a temporal sequence $\overrightarrow{\Delta}$ and let I_0, I_1 be distinct 1D-subregions in $\mathcal{P}_{\overrightarrow{\Delta}}$. The following results are immediate. Propositions 7 and 8 state that the dominance relation between two intervals I_0 and I_1—with the possible exception of the limit interval—respects the dominance relations of the pairs of regions reached by a $\rightsquigarrow_\Delta$ shift. In particular, the relation is strict since they can reach—with the same $\rightsquigarrow_\Delta$ shift—a pair of 1D-regions whose dominance relation is strict.

Proposition 7. *Let $\mathcal{P}_{\overrightarrow{\Delta}}$ be a maximal partition for a temporal sequence $\overrightarrow{\Delta}$ and let I_0, I_1 be 1D-subregions in $\mathcal{P}_{\overrightarrow{\Delta}}$. If there is $\Delta > 0$ such that $I_1 \rightsquigarrow_\Delta I \subseteq R$, $I_0 \rightsquigarrow_\Delta I' \subseteq R'$, and $R' \prec R$ hold, then $I_0 \prec I_1$ holds.*

If a maximal partition $\mathcal{P}_{\overrightarrow{\Delta}}$ has a limit interval $I = (\eta, \eta)$ and for all Δ' in $\overrightarrow{\Delta}$ it holds that $\eta + \Delta' \notin \mathbb{N}_{>0}$ (that is, there are no Δ' in $\overrightarrow{\Delta}$ and $h \in \mathbb{N}_{>0}$ such that $I \rightsquigarrow_{\Delta'} (h, h)$ holds), then we say that the limit interval I is *essential*. Therefore, when I is not essential, then there is also a distance Δ in $\overrightarrow{\Delta}$ that takes I to an integer number. For instance, the limit interval $(8/9, 8/9)$ of the Zeno temporal sequence of Example 1 is essential. Instead, the limit interval $(8/9, 8/9)$ of the Zeno sequence $\overrightarrow{\Delta} = 1/9, 1.1, 1.11, 1.111, 1.1111, \ldots$ is not essential for the presence of $1/9$.

Proposition 8. *Let $\mathcal{P}_{\overrightarrow{\Delta}}$ be a maximal partition for a temporal sequence $\overrightarrow{\Delta}$, I_0, I_1 be 1D-subregions in $\mathcal{P}_{\overrightarrow{\Delta}}$ such that $I_0 \prec I_1$.*

1. *If I_1 is the essential limit interval and $\sup(I_0) = \inf(I_1)$, then for all Δ in $\overrightarrow{\Delta}$ and intervals I, I' such that $I_1 \rightsquigarrow_\Delta I$ and $I_0 \rightsquigarrow_\Delta I'$, there is a 1D-region R such that both $I \subset R$ and $I' \subset R$ hold.*

2. *Otherwise, there exists a value Δ in $\overrightarrow{\Delta}$ such that $I_0 \rightsquigarrow_\Delta I' \subseteq R'$ and $I_1 \rightsquigarrow_\Delta I \subseteq R$ (for 1D-subregions I, I' of 1D-regions R, R') and $R' \prec R$.*

Notice that Part 2 of Prop. 8, which enforces the dominance relation among regions, excludes the case where the limit interval I_1 is essential, which is covered by Part 1; indeed, when I_1 is not essential, then the dominance relation is determined as for any other pair of intervals of the maximal partition. For instance, 1D-subregions $I_0 = (0, 8/9)$ and $I_1 = (8/9, 8/9)$ of the maximal partition of the Zeno sequence of Example 1 verify the conditions of Part 1 of Prop. 8 and for every $\rightsquigarrow_\Delta$ shift they reach the same 1D-regions.

The definition of $\mathcal{P}_{\overrightarrow{\Delta}}$ considers a temporal sequence $\overrightarrow{\Delta}$ where every Δ_i represents the distance in time from the first instant. In some cases, it is useful to have a different notation for a maximal partition, based instead on time increments. We define it only for the case of finite sequences. Let $\overrightarrow{\delta} = \delta_1\delta_2 \ldots \delta_m$, where $m = |\overrightarrow{\delta}|$, be a sequence of $m > 0$ positive real numbers, called *region distances*. The region distances induce a corresponding temporal sequence $\overrightarrow{\Delta} = \Delta_1 \ldots \Delta_m$ as follows: for every i, $1 \leq i \leq m$, let $\Delta_i = \sum_{1 \leq j \leq i} \delta_j$. The maximal partition $\mathcal{P}_{\overrightarrow{\delta}}$ for $\overrightarrow{\delta}$ is just the maximal partition $\mathcal{P}_{\overrightarrow{\Delta}}$.

5. Elimination of Diagonal Constraints

Theorem 1 shows that, for CLTLoc formulae—as well as for TA—if one considers only initialized timed languages, then diagonal constraints can be eliminated without loss of generality. A similar result, with some restrictions, holds over CLTLoc models that are not necessarily well-initialized. A CLTLoc formula is called diagonal-free if it does not include diagonal constraints of the form $x \sim y + c$, with $c \in \mathbb{N}_{\geq 0}$. As shown in Section 3, the CLTLoc formula defining non-timed regular language L_{ni} of Proposition 1 does not include diagonal constraints. Then, eliminating diagonal constraints, as done in Theorem 2 below, does not guarantee the regularity of the defined language.

Theorem 2. *Let ϕ be a CLTLoc formula. Then:*

1. *If no diagonal constraint in ϕ is of the form $x \sim Now + c$ or of the form $Now \sim y + c$, then there exists a diagonal-free CLTLoc formula ϕ' language equivalent to ϕ.*

2. *There exists a CLTLoc formula ϕ' language equivalent to ϕ over non-Zeno timed words and without diagonal constraints.*

Let $C > 0$ be an integer constant.

Consider a set of clocks X. Define the following formula that relates the 1D-region of each $x \in X$ with the values of its corresponding region clocks:

$$bridge(X) \stackrel{\text{def}}{=} \bigwedge_{x \in X} \mathbf{G} \left(\begin{array}{c} (x = 0 \Leftrightarrow [x = 0]) \\ \wedge \\ \bigwedge_{1 \leq k \leq C} (x = k \Leftrightarrow [x = k]) \wedge (x < k \Leftrightarrow [x < k]) \end{array} \right)$$

It is clear that every model (π, σ) of the above bridge formula must be such that, for all $x \in X$, for all $0 \leq c \leq C$:

$$\sigma(i, z_{[x \sim c]}) = 0 \text{ if, and only if, } \sigma(i, x) \sim c \text{ holds.} \tag{4}$$

The goal of the bridge formula is to determine the actual 1D-region of each clock, to make the values of region clocks consistent with the values of the corresponding clocks: the actual 1D-region of clocks x and y may help the region automaton to determine whether a constraint $x \sim y + c$ holds or not. For example, if both $x = c'$ and $y = c' + 2$ hold, with $0 < c' \leq C - 2$, then the region automaton ensures that $[x = y + 2]$ holds, i.e., the diagonal constraint $x = y + 2$ is determined to be true, without checking the actual constraint.

However, the region automaton cannot always discriminate whether $[x \sim y + c]$ holds: for instance, if neither clocks x and y have ever been reset in the past, x is in the open interval $(c' - 1, c')$ and y is in the open interval $(c'' - 1, c'')$, then the region automaton can only ensure that the difference $x - y$ is in the open interval $(c' - c'' - 1, c' - c'' + 1)$, but it cannot decide whether $x - y$ is in the open interval $(c' - c'' - 1, c' - c'')$ or in $(c' - c'', c' - c'' + 1)$—i.e., it cannot decide which of x and y has the greatest fractional part.

Let $\widehat{X}$ be a set of clocks. To simplify the following proofs, we introduce a new set $\widetilde{X}$ of clocks as a marked copy of clocks in $\widehat{X}$, with the intended meaning that a clock x and its marked copy $\widetilde{x}$ have the same fractional part. Let $Z_{\widehat{X} \cup \widetilde{X}}$ be the set of region clocks for $\widehat{X} \cup \widetilde{X}$. Notice that the clock Now is still necessary, since Now determines the time stamps, hence it cannot be completely replaced by clock $\widetilde{Now}$ with the same fractional part.

The following Formula (5) (where $[\![c \leq x < c + 1]\!]$ is an abbreviation for $[\![x = c]\!] \vee [\![c < x < c + 1]\!]$) is used to restrict the sequences of regions, defined by $\Theta_{Z_{\widehat{X} \cup \widetilde{X}}, C}$, only to those conforming to the following property. At position 0, either both $x, \widetilde{x}$ have the same fractional part, but the integer part of $\widetilde{x}$ is equal to 0; or both are greater than C (in which case the relation among their fractional parts is irrelevant). In all other positions, the resets of $x, \widetilde{x}$ always occur at the same time (hence, their fractional parts are always the same).

$$fract(Z_{\widehat{X}\cup\widetilde{X}}) \stackrel{\text{def}}{=} \bigwedge_{x\in\widehat{X}} \left(\begin{array}{ll} \bigwedge_{0\le c<C} ([\![c \le x < c+1]\!] \Rightarrow [x = \widetilde{x} + c]) & \wedge \\ ([x = C] \Rightarrow [x = \widetilde{x} + C]) & \wedge \\ ([C < x] \Rightarrow ([C < \widetilde{x}] \wedge [x = \widetilde{x}])) & \wedge \\ \mathbf{XG}\,([x = 0] \Leftrightarrow [\widetilde{x} = 0]) & \end{array} \right) \tag{5}$$

Define:

$$constr_{\widetilde{X},C} \stackrel{\text{def}}{=} \Theta_{Z_{\widehat{X}\cup\widetilde{X}},C} \wedge bridge(\widetilde{X}) \wedge fract(Z_{\widehat{X}\cup\widetilde{X}}).$$

Formula $constr_{\widetilde{X},C}$ represents the runs of the region automaton $\Theta_{Z_{\widehat{X}\cup\widetilde{X}},C}$ and constrains the value of clocks in $\widetilde{X}$ according to $fract(Z_{\widehat{X}\cup\widetilde{X}})$ (the clocks x and $\tilde{x}$ have the same fractional part). Both $\Theta_{Z_{\widehat{X}\cup\widetilde{X}},C}$ and $constr_{\widetilde{X},C}$ are defined over $Z_{\widehat{X}\cup\widetilde{X}}$, whereas clocks in $\widetilde{X}$ appear explicitly in $bridge(\widetilde{X})$. The presence of $bridge(\widetilde{X})$ guarantees that the value of $[\tilde{x} \sim c]$ is "in agreement" with $\tilde{x} \sim c$ in all positions of the models. The following Lemma 1 allows us to extend this property to diagonal constraints over clocks in $\widetilde{X}$ also; i.e., $[\tilde{x} \sim \tilde{y} + c]$ is "in agreement" with $\tilde{x} \sim \tilde{y} + c$. The proof is based on the idea of changing the values of clocks $\tilde{x}$ without changing their 1D-regions, but so that the diagonal constraints $\tilde{x} \sim \tilde{y} + c$ actually agree with constraints $[\tilde{x} \sim \tilde{y} + c]$. This is crucial for proving the main theorem of this section about the elimination of diagonal constraints.

Lemma 1. *For every model (π, σ') of formula $constr_{\widetilde{X},C}$ there exists another model (π, σ) of the same formula such that $\sigma(i, z) = \sigma'(i, z)$ for all $z \in Z_{\widehat{X}\cup\widetilde{X}} \cup \{Now\}$ and $i \in \mathbb{N}_{\ge 0}$, and*

1. *for all $\widetilde{x}, \widetilde{y} \in \widetilde{X}, i \in \mathbb{N}_{\ge 0}, 0 \le c \le C$ if neither $\widetilde{x}$ nor $\widetilde{y}$ are Now, then the relation*

$$(*) \quad \sigma(i, z_{[\widetilde{x}\sim\widetilde{y}+c]}) = 0 \text{ if, and only if, } \sigma(i, \widetilde{x}) \sim \sigma(i, \widetilde{y}) + c$$

holds;

2. *if (π, σ') is non-Zeno, then relation $(*)$ holds also when $\widetilde{x}$ or $\widetilde{y}$ are Now.*

Proof. We first prove Part (1). In general, defining $\sigma(i, \widetilde{x}) = \sigma'(i, \widetilde{x})$ for $\widetilde{x} \in \widetilde{X}$ does not ensure that Relation $(*)$ holds: the value of region clocks at position 0 (which are linked to the values of constraints $\widetilde{x} \sim c$ by Formula $bridge(\widetilde{X})$ of $constr_{\widetilde{X},C}$) uniquely determines an initial symbolic region REG, which in general may be different from the initial region REG' defined by the actual clock assignment σ'. As already noticed, the definition of $constr_{\widetilde{X},C}$ (and in particular of subformula $bridge(\widetilde{X})$) ensures that the only case when $REG \neq REG'$ is if the fractional parts of two clocks $\widetilde{x}, \widetilde{y}$ are not in the same order determined by the region clocks $z_{[\widetilde{x} \sim \widetilde{y}+c]}$ (i.e., Relation (∗) does not hold).

We show how to change the value of the clocks to be consistent both with the region clocks and the non-diagonal clock constraints.

All clocks that start at position 0 with a value greater than C can easily be redefined to verify the constraints given by the region clocks. We separate two cases. If at position 0 both $\widetilde{x} > C$ and $\widetilde{y} > C$ hold, then any constraints of the form $z_{[\widetilde{x} \sim \widetilde{y}+c]} = 0$ can easily be made true or false by simply modifying the values of $\widetilde{x}$ and $\widetilde{y}$ as necessary. If, on the other hand, $\widetilde{x} > C$ and $\widetilde{y} \leq C$, the value of $\widetilde{x}$ can be redefined to satisfy not only the constraints defined by regions clocks, but also those that can be inferred from them (which are guaranteed to be satisfiable by the region automaton); for example, if $\sigma'(0, z_{[C<\widetilde{x}]}) = 0$, $\sigma'(0, z_{[\widetilde{y}<1]}) = 0$, $\sigma'(0, z_{[0<\widetilde{y}]}) = 0$, $\sigma'(0, z_{[2<y]}) = 0$, $\sigma'(0, z_{[y<3]}) = 0$, $\sigma'(0, z_{[y=\widetilde{y}+2]}) = 0$, $\sigma'(0, z_{[x=y+C-1]}) = 0$ (where $C > 3$ and $x, y \in \widehat{X}$) all hold, then it must also hold that $C+1 < \sigma(0, \widetilde{x}) < C+2$ (notice that region clocks $z_{[C+1<\widetilde{x}]}$ and $z_{[\widetilde{x}<C+2]}$ do not exist, but the truth of the corresponding constraint $C+1 < \widetilde{x} < C+2$ can be inferred from the other region clocks).

Clearly, this can be done for any number of clocks that are greater than C at 0. We can thus focus in the following on the subset $X^{\leq C} \subseteq \widetilde{X}$ of clocks of $\widetilde{X}$ that are not greater than C at position 0, hence their initial value is less than 1.

Define the equivalence relation $\approx \subseteq X^{\leq C} \times X^{\leq C}$ as the reflexive and symmetric closure of the following relation: For every $\widetilde{x}, \widetilde{y} \in X^{\leq C}$, $\widetilde{x} \approx \widetilde{y}$ if $\sigma'(0, z_{[\widetilde{x}=\widetilde{y}]}) = 0$. Hence, $\widetilde{x} \approx \widetilde{y}$ holds if, according to the symbolic region REG', the two clocks start with the same value at position 0.

It is obvious that $\widetilde{x} \approx \widetilde{y}$ is an equivalence relation over $X^{\leq C}$. Clocks in the same equivalence class must be assigned the same value (i.e., the same fractional part) by σ at position 0.

Let ι be the smallest value of all $i \in \mathbb{N}_{\geq 0}$ such that $\sigma'(i, \mathit{Now}) - \sigma'(0, \mathit{Now}) > C$ if any such i exists, otherwise (with an abuse of notation) let $\iota = +\infty$. Clearly, the latter case may only occur if the timed word is Zeno.

Similarly, for every $\widetilde{x} \in X^{\leq C}$ let $\iota_{\widetilde{x}}$ be the smallest value of all $i \in \mathbb{N}_{\geq 0}$ such that $i < \iota$ and $\sigma'(i, \widetilde{x}) = 0$ if any such i exists, otherwise (with an abuse of notation) let $\iota_{\widetilde{x}} = +\infty$. Hence, $\iota_{\widetilde{x}}$ is the position of the first reset of $\widetilde{x}$ (if any) before C.

If $\widetilde{x} \in X^{\leq C}$ and $\iota_{\widetilde{x}} < +\infty$ then for all $i \geq \iota_{\widetilde{x}}$ let $\sigma(i, \widetilde{x}) = \sigma'(i, \widetilde{x})$, since the value given by $\sigma'(i, \widetilde{x})$ after $\widetilde{x}$ is reset is in fact compatible with the region clocks by definition of $\mathit{constr}_{\widetilde{X},C}$.

For every $i \in \mathbb{N}_{\geq 0}, 0 < i < \iota$, let $\Delta_i = \sigma'(i, \mathit{Now}) - \sigma'(0, \mathit{Now})$. Let $\overrightarrow{\Delta}$ be the temporal sequence $\Delta_1, \Delta_2, \ldots$, and let $\mathcal{P}_{\overrightarrow{\Delta}}$ be the maximal partition for the temporal sequence $\overrightarrow{\Delta}$.

Consider first the subset $N \subseteq X^{\leq C}$ of clocks $\widetilde{y}$ such that: (i) for some $i < \iota_{\widetilde{y}}$ it holds that $\sigma'(i, \widetilde{y}) = c$, for some integer $0 < c \leq C$—that is, there is $\Delta_i \in \overrightarrow{\Delta}$ such that $\sigma'(0, \widetilde{y}) + \Delta_i = c$, which in turn entails that $(\langle \sigma'(0, \widetilde{y}) \rangle, \langle \sigma'(0, \widetilde{y}) \rangle)$ belongs to

$\mathcal{P}_{\overrightarrow{\Delta}}$; or (ii) $\sigma'(0, \widetilde{y}) = 0$. The fractional part of these clocks cannot be changed, otherwise the constraint $\widetilde{y} = c$ would be violated at position i. Therefore, for every $\widetilde{y} \in N$, let $\sigma(i, \widetilde{y}) = \sigma'(i, \widetilde{y})$ for every $i \in \mathbb{N}_{\geq 0}$. This assignment is consistent also with the auxiliary clocks. Hence, every clock $\widetilde{x} \in X^{\leq C} - N$ has a fractional part different from zero in every position $i < \iota_{\widetilde{x}}$.

We now show how to adjust the initial value of the fractional parts of every clock $\widetilde{x} \in X^{\leq C} - N$ in order to assign it a correct value also before $\iota_{\widetilde{x}}$. For every position i with $0 < i < \iota_{\widetilde{x}}$, the assignment σ will be just based on the initial value and the elapsed time since the origin, i.e., $\sigma(i, \widetilde{x}) = \sigma(0, \widetilde{x}) + \sigma'(i, \mathit{Now}) - \sigma'(0, \mathit{Now})$.

Finally, we can assume that, even in the case when the limit interval (η, η) is in the maximal partition, then $\sigma'(0, \widetilde{x}) \neq \eta$ for every $\widetilde{x} \in X^{\leq C} - N$. In fact, if $\widetilde{x} \in X^{\leq C} - N$, then (η, η) is the essential limit interval. This is the case of Prop. 8, Part 1, which means that the limit interval and the open interval I in the maximal partition $\mathcal{P}_{\overrightarrow{\Delta}}$ that immediately precedes it (i.e., such that $\sup(I) = \eta$) are indistinguishable from the point of view of the 1D-regions they traverse on sequence $\overrightarrow{\Delta}$. Then, if $\sigma'(0, \widetilde{x}) = \eta$ we can just modify $\sigma'(0, \widetilde{x})$ by assigning it any value in the open interval I preceding the limit interval: the region automaton does not differentiate the two cases (starting from I or starting from (η, η)).

The first case we consider is when $\widetilde{x}, \widetilde{y}$ are two clocks in $X^{\leq C}$ such that at some position i, with $i < \iota_{\widetilde{x}}, i < \iota_{\widetilde{y}}$, there exist two 1D-regions $R_{\widetilde{x}}$, $R_{\widetilde{y}}$ such that $\sigma'(i, \widetilde{x}) \in R_{\widetilde{x}}$, $\sigma'(i, \widetilde{y}) \in R_{\widetilde{y}}$ and $R_{\widetilde{x}} \prec R_{\widetilde{y}}$ hold. By Prop. 7, $\sigma'(0, \widetilde{x})$ and $\sigma'(0, \widetilde{y})$ must be in two distinct intervals $I_{\widetilde{x}}, I_{\widetilde{y}}$ of the maximal partition $\overrightarrow{\Delta}$, with, moreover, $I_{\widetilde{x}} \prec I_{\widetilde{y}}$. Therefore, the values of $\widetilde{x}, \widetilde{y}$ in σ' are already consistent with the region clocks and nothing needs to be changed. Constraint $[\widetilde{x} < \widetilde{y}]$ holds at position 0, because of the form of the region automaton. Moreover, $\sigma(0, \widetilde{x})$ (resp., $\sigma(0, \widetilde{y})$) may be assigned any value in $I_{\widetilde{x}}$ (resp., $I_{\widetilde{y}}$) without affecting both the truth of constraint $[\widetilde{x} < \widetilde{y}]$ and of the actual constraint $\widetilde{x} < \widetilde{y}$.

The second case we consider is when $\widetilde{x}, \widetilde{y}$ are two clocks in $X^{\leq C}$ such that in all positions i, with $i < \iota_{\widetilde{x}}, i < \iota_{\widetilde{y}}$, the clock assignment σ' is such that $\widetilde{x}$ and $\widetilde{y}$ belong to the same 1D-region. By Prop. 8, Part 2, $\sigma'(0, \widetilde{x})$ and $\sigma'(0, \widetilde{y})$ must be in the same interval I of the maximal partition $\overrightarrow{\Delta}$ (since we assumed that neither can start in the limit interval (η, η) of the maximal partition). In addition, either it holds that $\widetilde{x}, \widetilde{y} \in N$, or that $\widetilde{x}, \widetilde{y} \in X^{\leq C} - N$. As argued above, in the former case the fractional parts of the values of the clocks are exactly the same since there exist a position i, with $i < \iota_{\widetilde{x}}, i < \iota_{\widetilde{y}}$, where $\sigma'(i, \widetilde{x}) = \sigma'(i, \widetilde{y}) = c \leq C$. Hence, formula $\mathit{constr}_{\widetilde{X}, C}$ enforces that constraint $[\widetilde{x} = \widetilde{y}]$ holds at position 0. If both $\widetilde{x}$ and $\widetilde{y}$ are in $X^{\leq C} - N$, then by the form of the region automaton I is not punctual (i.e., $\inf(I) < \sup(I)$). Let $\sigma(0, \widetilde{y}) = \sigma'(0, \widetilde{y})$; for the value of $\sigma(0, \widetilde{x})$ consider the following three cases.

1. If the auxiliary clock $z_{[\widetilde{x} = \widetilde{y}]}$ is 0 at position 0, then let $\sigma(0, \widetilde{x})$ have the same fractional value of $\sigma'(0, \widetilde{y})$ in I (recall that, by the considerations above, we can safely assume that at position 0 all clocks are in intervals $(0, 1)$ or $(0, 0)$): $\sigma(0, \widetilde{x}) = \sigma'(0, \widetilde{y})$.

2. If the auxiliary clock $z_{[\widetilde{x}<\widetilde{y}]}$ is equal to 0, since I is not punctual let $\sigma(0,\widetilde{x}) = \alpha$, where $\alpha < \sigma'(0,\widetilde{y})$ is a value in I.

3. The case of the auxiliary clock $z_{[\widetilde{y}<\widetilde{x}]}$ equal to 0 is symmetrical to the previous one: let $\sigma(0,\widetilde{x}) = \beta$, where $\beta > \sigma'(0,\widetilde{y})$ is a value in I.

Since I is not punctual, this procedure can be applied to any number of clocks with the same property, defining their fractional part in I to verify the order of the fractional parts determined by the region clocks, which in turn guarantees that Relation $(*)$ holds when Part (1) is true.

We now prove also Part (2). The proof is the same as Part (1), with the difference that the auxiliary clocks of the form $z_{[\widetilde{x}\sim\widetilde{y}+c]}$ that occur in $constr_{\widetilde{X},C}$ may have *Now* instead of either $\widetilde{x}$ or $\widetilde{y}$.

As in the proof of Part (1), we focus only on the subset $X^{\leq C} - N$ of clocks. Since, crucially, the timed word is assumed to be non-Zeno, there is no limit interval. Hence, the case covered by Prop. 8, Part 1 never occurs. Then, any pair of clocks $\widetilde{x}, \widetilde{y}$ different from $\widetilde{Now}$ can be ordered as in the proof of Part (1) to be consistent with the region clocks. If $\widetilde{y}$ is $\widetilde{Now}$, the case where $\widetilde{y} \in N$ is as before (since $\widetilde{y}$ cannot be moved); if $\widetilde{y} \in X^{\leq C} - N$, then $I_{\widetilde{y}}$ is an open interval and the fractional parts of the other clocks can be distributed around it as necessary. □

Proof of Th. 2. We only show Part (1), since the proof of Part (2) is identical, using Condition (2) of Lemma 1 instead of Condition (1). We define:

$$\phi' = \phi_{AUX} \wedge constr_{\widetilde{X},C}$$

where ϕ_{AUX} is obtained by replacing all constraints in ϕ with the corresponding tests on auxiliary clocks, i.e., replacing every atomic formula of the form $x \sim c$ and $x \sim y + c$ by the formula $[x \sim c]$ and $[x \sim y + c]$, respectively.

We first show that if a timed word (π, τ) is in the timed language of ϕ then it is also in the timed language of ϕ'. Let (π, σ) be a CLTLoc model of ϕ such that $(\pi, \tau) = [(\pi, \sigma)]$, with $\sigma : \mathbb{N}_{\geq 0} \times \widehat{X} \to \mathbb{R}_+$.

Define a clock assignment σ' for $\widehat{X} \cup \widetilde{X} \cup Z_{\widehat{X}\cup\widetilde{X}}$ where for all $i \in \mathbb{N}_{\geq 0}$, $x \in \widehat{X}$ and $\widetilde{x} \in \widetilde{X}$:

1. if $\sigma(0,x) \leq C$ then $\sigma'(0,\widetilde{x}) = \langle\sigma(0,x)\rangle$, else $\sigma'(0,\widetilde{x}) = \sigma(0,x)$;

2. $\sigma'(i,\widetilde{x}) = 0$ if, and only if, $\sigma(i,x) = 0$.

3. if $\sigma'(i,\widetilde{x}) \neq 0$ and $i > 0$, then $\sigma'(i,\widetilde{x}) = \sigma'(i-1,\widetilde{x}) + \sigma(i, Now) - \sigma(i-1, Now)$.

4. $\sigma'(i,x) = \sigma(i,x)$

also, for all region clocks $z \in Z_{\widehat{X}\cup\widetilde{X}}$ of the form $z_{[x\sim c]}$ or of the form $z_{[x\sim y+c]}$ (with $x, y \in \widehat{X} \cup \widetilde{X}$) and for all $0 \leq c \leq C$:

1. If $\sigma'(i,x) \sim \sigma'(i,y) + c$ holds true, then let $\sigma'(i, z_{[x\sim y+c]}) = 0$, else let $\sigma'(i, z_{[x\sim y+c]})$ be defined as *Now* if $i = 0$, and as $\sigma'(i-1, z_{[x\sim y+c]}) + \sigma'(i, Now) - \sigma'(i-1, Now)$ if $i > 0$.

2. If $\sigma'(i,x) \sim c$ holds true, then let $\sigma'(i, z_{[x\sim c]}) = 0$, else let $\sigma'(i, z_{[x\sim c]})$ be defined as *Now* if $i = 0$, and as $\sigma'(i-1, z_{[x\sim c]}) + \sigma'(i, Now) - \sigma'(i-1, Now)$ if $i > 0$.

Mapping σ' is a clock assignment for ϕ' by definition. Clearly, (π, σ') is a CLTLoc model for ϕ_{AUX} (it includes assignments to clocks of set $\widehat{X}$, which do not appear in ϕ_{AUX}, hence whose value is irrelevant for the satisfiability of the formula): in ϕ_{AUX}, every clock constraint in ϕ corresponds to a constraint on a region clock that has the same truth value by construction in every position. Also by construction, (π, σ') is a model for $constr_{\widetilde{X},C}$. Thus, (π, σ') is a model for ϕ'; moreover, $(\pi, \tau) = [(\pi, \sigma')]$ (since $\sigma'(i, Now) = \sigma(i, Now)$ for every $i \in \mathbb{N}_{\geq 0}$). Therefore, (π, τ) is also in the timed language of ϕ'.

We now show the converse, i.e., that if a timed word (π, τ) is in the timed language of ϕ', then it is also in the timed language of ϕ. Let (π, σ') be a CLTLoc model of ϕ' such that $(\pi, \tau) = [(\pi, \sigma')]$. The thesis follows by Relation $(*)$ since Lemma 1, Condition (1) holds, only requiring to assign the correct integer parts to clocks in $\widehat{X}$. More precisely, define $\sigma(i, x)$ for $i \in \mathbb{N}_{\geq 0}$, $x \in \widehat{X}$ as follows:

- if $\sigma'(0, \widetilde{x}) < 1$ then let $\sigma(0, x) = \sigma'(0, \widetilde{x}) + h$ where h is the smallest integer such that $\sigma'(0, z_{[x<h]}) = 0$ or $\sigma'(0, z_{[x=h]}) = 0$ holds; if $\sigma'(0, \widetilde{x}) \geq 1$—which corresponds, by formula $constr_{\widetilde{X},C}$, to the condition $\sigma'(0, \widetilde{x}) > C$—let $\sigma(0, x) = \sigma'(0, \widetilde{x})$ matching constraint $[x = \widetilde{x}]$ of $constr_{\widetilde{X},C}$;

- $\sigma(i, x) = 0$ if, and only if, $\sigma'(i, \widetilde{x}) = 0$;

- if $\sigma(i, x) \neq 0$ and $i > 0$, then $\sigma(i, x) = \sigma(i-1, x) + \sigma'(i, Now) - \sigma'(i-1, Now)$.

□

We remark that the size of formula ϕ' defined in the proof of Theorem 2 is equal to the size of ϕ plus the size of $constr_{\widetilde{X},C}$, where the dominant term is the size of the region automaton $\Theta_{Z_{\widehat{X}\cup\widetilde{X}},C}$ (see Sect. 4).

Notice that Theorem 2 does not cover the case of Zeno timed words when formula ϕ contains diagonal constraints on *Now*. Indeed, consider the Zeno timed language L_{ni1} of Section 3, which is defined by CLTLoc formula $\mathbf{G}(a \wedge 0 < x < 1 \wedge Now < x)$. We have the following result that shows how, in the case of Zeno timed words, diagonal constraints of the form $Now \sim x$ increase the expressive power of the logic.

Proposition 9. *There is no CLTLoc formula that does not include constraints of the form $Now \sim x$ that defines language L_{ni1}.*

The proof of Proposition 9 is a straight consequence of the following lemma.

Lemma 2. *Let ϕ be a CLTLoc formula, whose set of clocks is $\widehat{X}$, that does not include constraints of the form $Now \sim x$. If ϕ admits a model (π, σ) such that $\lim_{i \to +\infty} \sigma(i, Now) = \gamma$ for some $\gamma < 1$, then it also admits a model (π, σ') such that $\lim_{i \to +\infty} \sigma'(i, Now) = 1$.*

Proof. Since clock *Now* is, by definition, always greater than 0, for all $i \in \mathbb{N}_{\geq 0}$ it holds that $0 < \sigma(i, Now) < 1$ (and, in general, that $\sigma(i, Now) < c$ for all $c \in \mathbb{N}_{\geq 0}$, with $c > 0$). Now, consider (π, σ') such that, for all $i \in \mathbb{N}_{\geq 0}$, $\sigma'(i, Now) = \sigma(i, Now) + 1 - \gamma$ and, for all $x \in \widehat{X} - \{Now\}$, $\sigma'(i, x) = \sigma(i, x)$. Clearly, it holds that $\lim_{i \to +\infty} \sigma'(i, Now) = 1$. In addition, the value of constraints $x \sim c$, $x \sim y + c$ is the same in both σ and σ', since the values of clocks x, y do not change between σ and σ'. Also, for all $i \in \mathbb{N}_{\geq 0}$ it holds that $0 < \sigma'(i, Now) < 1$ (and, in general, that $\sigma'(i, Now) < c$). Hence, the values of clock constraints are the same in σ and σ' (similarly for propositional letters, since π is the same), so if it holds that $(\pi, \sigma) \models \phi$, then it also holds that $(\pi, \sigma') \models \phi$. □

6. All non-Zeno CLTLoc timed languages are timed regular

The goal of this section is to show that, under a non-Zeno assumption, every timed language of CLTLoc is timed regular. In general, this does not hold for languages including Zeno timed words, as shown by language L_{ni} of Section 3.

The following proposition summarizes the results obtained in Section 5 for non-Zeno timed languages, since they will be useful in this section.

Proposition 10. *Let ϕ be a CLTLoc formula over the clocks of a set $\widehat{X}$, and let C be the greatest constant occurring in ϕ. There exists a diagonal-free CLTLoc formula ϕ' such that:*

- *ϕ' is defined over a set of clocks $X \cup \{Now\}$;*
- *ϕ' is language equivalent to ϕ over non-Zeno timed words;*
- *for every non-Zeno timed word (π, τ) in the language of ϕ, there is a model (π, σ) of ϕ' such that $(\pi, \tau) = [(\pi, \sigma)]$ with a clock assignment σ such that for all $x \in X$:*

$$0 \leq \sigma(0, x) < 1 \text{ or } \sigma(0, x) > C \text{ or } \sigma(0, x) = \sigma(0, Now).$$

The proof of the following statement is the focus of the remainder of this section.

Theorem 3. *For every non-Zeno CLTLoc timed language L, there exists a CLTLoc formula whose initialized timed language is L.*

We here outline the proof of the main theorem to facilitate the reading of the next sections. By Proposition 10, we can assume that a formula ϕ defining a timed language L is diagonal-free and it is defined on clocks in $X \cup \{Now\}$, which are not well-initialized in general.

Our goal is to define a formula ϕ' whose initialized timed language is L. To achieve this goal, formula ϕ' has a set of well-initalized fresh clocks, including region clocks that are used to evaluate constraints on a clock $x \in X$ in the positions preceding the first reset of x (if any) or before x becomes greater than C. Since x is only evaluated symbolically, its actual value in the first position is irrelevant: in every model (π, σ') of ϕ', $\sigma'(x, 0)$ can be assumed to be 0 or *Now*.

The core of the proof is showing that, given a model (π, σ') as above, there is actually a non-empty interval of values for the clock valuation of x in the initial position, which can be used to define a model (π, σ) for the original formula ϕ. In other words, we prove that there is indeed a non-emtpy interval for initial clock valuation $\sigma(0, x)$, with $0 \leq \sigma(0, x) < 1$ or $\sigma(0, x) > C$ and such that (π, σ) is a model of ϕ.

The existence of a non-empty interval for the initial assignment $\sigma(0, x)$ (Lemma 3) is based on the maximum constant occurring in ϕ and on the sequence of timestamps of a model of ϕ'. The sequence of timestamps can be determined by introducing a finite set of well-initialized clocks D, that are used to measure the distance among the positions where x reaches or leaves a 1D-region along the prefix (Lemma 4). The relation among the clocks in X and clocks D is captured by a formula encoding a region automaton over $X \cup D$, with the region for clocks D being "bridged" to their actual values (Lemma 5).

To prove the claim, we need some new definitions and various intermediate lemmata and propositions.

Initially Normalized Clock Regions

The goal of the following construction is to replace every clock $x \in X$ with fresh, well-initialized clocks. To this end, we consider the sequence of regions that each $x \in X$ traverses from the origin until its first reset: this may be represented symbolically by using a finite number of (well-initialized) region clocks. Along the prefix, the value of a clock constraint over x can be exactly determined by the region clocks, whereas, after the first reset of x, it can be determined by the actual value of the clock in the clock assignment. Hence, we focus on the sequences of regions traversed by clocks, until they are reset for the first time. Clearly, each clock may be reset independently of the others or it might not even be reset ever.

To avoid some complications in the proof, instead of a formula ϕ we consider its language equivalent formula ϕ' of Proposition 10. Since we are dealing with non-Zeno behavior, we consider finite sequences of clock regions where each clock starts from $[0, 1)$ and, assuming it is never reset, it eventually reaches region $(C, +\infty)$. Hence, it always stays in intervals of the form $(n, n + 1)$, or in single points n, with $n \in \mathbb{N}_{\geq 0}$, $n < C$, or in the open region $(C, +\infty)$. If, instead, the clock already starts from region $(C, +\infty)$, it always stays there.

As a consequence, we introduce next some definitions about finite sequences of regions.

Features of sequences of 1D-regions

Through the concept of monotonic sequence defined below, we capture the fact that in our models time is strictly increasing, therefore it is forbidden to stay in a region of the form (n, n) for more than one instant. For instance, the sequence

$$(0,1)(0,1)(0,1)(1,1)(1,2)(2,2)(3,4)$$

is monotonic, whereas $(1,1)(1,2)(2,2)(2,2)$ is not, since $(2,2)$ is consecutively repeated in the latter example. We also introduce the notion of complete sequences, in which time is always progressing at least until the clock hits C.

A *finite* sequence of $m \geq 1$ 1D-regions $\mathcal{R}_m = R_0 R_1 \dots R_m$ is called *monotonic* if, for every $1 \leq i \leq m$, $R_{i-1} \preceq R_i$ and each punctual interval of the form (h, h), with $1 \leq h \leq C$, appears at most once in the sequence; it is called *complete* if R_0 is $(0, 1)$, $R_{m-1} \prec R_m$ holds, and R_m is $(C, +\infty)$.

While in general a monotonic sequence may also define Zeno behaviors accumulating before C, in a complete, monotonic sequence time cannot stop progressing before clock x has reached C.

A monotonic sequence $R_0 R_1 \dots R_m(x)$ of $m+1$ 1D-regions is called *compact* (or *compactly monotonic*) if for all i, $1 \leq i \leq m-1$, we have $R_{i-1} \prec R_i$ or $R_i \prec R_{i+1}$. For example, the monotonic sequence $(0,1)(1,1)(1,2)(1,2)(2,2)(3,4)$ is also compact since it stays in the same region $(1,2)$ only for two positions, whereas $(0,1)(1,1)(1,2)(1,2)(1,2)(2,2)(3,4)$ is not, since there are three consecutive positions in region $(1,2)$. This definition is intended to abstract away long sequences of the same 1D-region, by considering only the entrance and the exit positions in the region and ignoring all intermediate positions.

It is immediate to prove that $m \leq 3C$ holds for every compactly monotonic sequence of $m+1$ 1D-regions. Moreover, it is always possible to "extract" a compactly monotonic sequence from a monotonic one, i.e., for every monotonic sequence of 1D-regions $R_0 R_1 \dots R_n$, there exist a value $m \leq n$, $1 \leq m \leq 3C$, and $m+1$ positions $0 = i_0 < i_1 < \dots < i_m \leq n$ such that $R_{i_0} R_{i_1} \dots R_{i_m}$ is a complete, compactly monotonic sequence; such a sequence with $R_{i_0} = R_0$ is also the only one.

Feasible sequences; extensions of compactly monotonic sequences

We first specify a notion of *time-successor* relation between clock regions, as introduced by [1], but applied to 1D-regions.

Definition 7 (time-successors of 1D-regions)**.** For all $\eta \in \mathbb{R}$, with $\eta \geq 0$, the relation $\hookrightarrow_\eta$ over 1D-regions is defined for all $h \in \mathbb{N}_{\geq 0}$ as:

- If the fractional part of η is 0, then $(h, h) \hookrightarrow_\eta (h + \eta, h + \eta)$ and $(h, h+1) \hookrightarrow_\eta (h + \eta, h + \eta + 1)$.
- If the fractional part of η is greater than 0, then let $n = \lfloor \eta \rfloor$ and:

1. $(h,h) \hookrightarrow_\eta (h+n, h+n+1)$;
2. $(h,h+1) \hookrightarrow_\eta (h+n, h+n+1)$;
3. $(h,h+1) \hookrightarrow_\eta (h+n+1, h+n+1)$, and
4. $(h,h+1) \hookrightarrow_\eta (h+n+1, h+n+2)$.

The special case $(0,1) \hookrightarrow_\eta R$ means that R is one of $(n,n+1),(n+1,n+1),(n+1,n+2)$, for $n = \lfloor \eta \rfloor$. For example, $(1,2) \hookrightarrow_{3.15} (4,5)$, but also $(1,2) \hookrightarrow_{3.15} (5,5)$ and $(1,2) \hookrightarrow_{3.15} (5,6)$ (notice that $1.85 + 3.15 = 5$, and $1.9 + 3.15 > 5$).

We have the following proposition.

Proposition 11. *Let R, R' be 1D-regions, and let g, g', γ, γ' be such that:*

- $g, g' \in \mathbb{N}_{\geq 0}$, $\gamma, \gamma' \in \mathbb{R}$, $0 \leq \gamma' \leq \gamma < 1$
- $(0,1) \hookrightarrow_{g+\gamma} R$ *and* $(0,1) \hookrightarrow_{g'+\gamma'} R'$ *hold*
- *if* $g+\gamma \leq g'+\gamma'$, *then* $R \hookrightarrow_{(g'+\gamma')-(g+\gamma)} R'$ *holds, otherwise* $R' \hookrightarrow_{(g+\gamma)-(g'+\gamma')} R$ *does.*

Then, the following tables list all possible cases for R, R', depending on the order relation of γ and γ':

$$\gamma' = \gamma \begin{cases} \begin{array}{cc} R & R' \\ (g,g+1) & (g',g'+1) \\ (g+1,g+1) & (g'+1,g'+1) \\ (g+1,g+2) & (g'+1,g'+2) \end{array} \end{cases} \qquad \gamma' < \gamma \begin{cases} \begin{array}{cc} R & R' \\ (g,g+1) & (g',g'+1) \\ (g+1,g+1) & (g',g'+1) \\ (g+1,g+2) & \begin{cases} (g',g'+1) \\ (g'+1,g'+1) \\ (g'+1,g'+2) \end{cases} \end{array} \end{cases}$$

Proof sketch. If $\gamma' = \gamma$, then $(g'+\gamma') - (g+\gamma)$ is an integer number, and the left table is obvious.

If $\gamma' < \gamma$, let us consider for simplicity the case where $g'+\gamma' > g+\gamma$, the other being similar. We have that $\lfloor (g'+\gamma') - (g+\gamma) \rfloor = g' - g - 1$ holds. Then, the fractional part of $(g'+\gamma')-(g+\gamma)$ is $(g'+\gamma')-(g+\gamma)-(g'-g-1) = 1+\gamma'-\gamma$, and it holds that $0 < 1+\gamma'-\gamma < 1$, from which the second line of the right-hand table easily follows from Definition 7. The first line of the right-hand table is obtained by noticing that, by Definition 7, $(g', g'+1)$ is the only 1D-region R' for which both $(g,g+1) \hookrightarrow_{(g'+\gamma')-(g+\gamma)} R'$ and $(0,1) \hookrightarrow_{g'+\gamma'} R'$ hold—in fact, $(g,g+1) \hookrightarrow_{(g'+\gamma')-(g+\gamma)} (g',g'+1)$ is obtained by applying case 4 in Definition 7, whereas $(0,1) \hookrightarrow_{g'+\gamma'} (g',g'+1)$ is obtained by applying case 2. The last line of the right-hand table essentially does not constrain R'. ☐

Definition 8. A monotonic sequence of 1D-regions $R_0 R_1 \ldots R_m$ is *feasible* for a sequence $\overrightarrow{\delta} = \delta_1 \ldots \delta_m$ of positive real numbers if, for every i, j, with $0 \leq i < j \leq m$, $R_i \hookrightarrow_{\delta_{i+1}+\delta_{i+2}+\ldots\delta_j} R_j$.

For instance, the region sequence $(0,1)(1,2)(2,3)$ is feasible for $\delta_1 = 1.8, \delta_2 = 0.5$, since $(0,1) \hookrightarrow_{1.8} (1,2)$, $(0,1) \hookrightarrow_{2.3} (2,3)$ and $(1,2) \hookrightarrow_{0.5} (2,3)$, whereas $(0,1)(1,2)(3,4)$ is not feasible for the same $\delta_1 = 1.8, \delta_2 = 0.5$, since $(0,1) \hookrightarrow_{2.3} (3,4)$ but $(1,2) \not\hookrightarrow_{0.5} (3,4)$. In fact, by case $\gamma' < \gamma$ (first line) of Proposition 11, if $(0,1) \hookrightarrow_{1.8} R$ and $(0,1) \hookrightarrow_{2.3} R'$, with $\gamma' = 0.3 < 0.8 = \gamma$ and $g = 1, g' = 2$, then when R is $(1,2) = (g, g+1)$ it follows that R' can only be $(g', g'+1) = (2,3)$.

The next definition relies on relation $\rightsquigarrow$ of Definition 5.

Definition 9. For all $m \geq 0$, for all sequences $\overrightarrow{\delta} = \delta_1 \ldots \delta_m$ of positive reals, for all region sequences $\mathcal{R}_m = R_0 R_1 \ldots R_m$ feasible for $\overrightarrow{\delta}$ let $I_0 \subseteq R_0$ be a 1D-subregion and for all $1 \leq i \leq m$ let $I_i = I_0 \oplus \sum_{j=1}^{i} \delta_i$. We say that I_0 is *compatible* with $\mathcal{R}_m$ if $I_1 \subseteq R_1, \ldots, I_m \subseteq R_m$ hold—i.e., $I_0 \rightsquigarrow_{\delta_1} I_1 \rightsquigarrow_{\delta_2} \cdots \rightsquigarrow_{\delta_m} I_m$ holds. Moreover, I_0 is called *maximally compatible* if every 1D-subregion I, disjoint from I_0, is not compatible with $\mathcal{R}_m$. The notion of (maximal) compatibility is naturally extended to infinite sequences of 1D-regions.

The next lemma is crucial in proving the main result. It is first exemplified on a few concrete cases, as follows.

Example 2. Consider again the maximal partition $\mathcal{P}_{\overrightarrow{\Delta}} = \{(0, 0.2), (0.2, 0.2), (0.2, 0.9)(0.9, 0.9)(0.9, 1)\}$ of Example 1. Clearly, the sequence $\mathcal{R}_2 = (0,1)(1,2)(5,6)$ is feasible for $\overrightarrow{\delta} = \delta_1\delta_2$ (where $\delta_1 = \Delta_1 = 1.8$ and $\delta_2 = 3.3 = \Delta_2 - \Delta_1$), since $(0, 0.2) \rightsquigarrow_{\delta_1} (1.8, 2) \subseteq (1,2)$, and $(0, 0.2) \rightsquigarrow_{\delta_1+\delta_2} (5.1, 5.3) \subseteq (5,6)$. Also, the 1D-subregion $(0, 0.2)$ is maximally compatible for $\overrightarrow{\delta}$, since any point outside $(0, 0.2)$ cannot traverse the sequence $\mathcal{R}_2$: as already noticed, $(0.2, 0.2) \rightsquigarrow_{\delta_1} (2,2)$, with region $(2,2)$ not being in the sequence. Therefore, $(0, 0.2)$ includes all, and only, points compatible with the sequence $\mathcal{R}_2$ and the given values δ_1, δ_2. This is a general fact, stated in Part 1 of the lemma. It should be clear that every 1D-subregion in the maximal partition corresponds to one, and only one, monotonic sequence compatible with the same values δ_1, δ_2. This is expressed by Part 2 of the lemma.

To give the intuition on the existence of an initial assignment for a clock x, proven in the next Lemma 3, consider Fig. 3, showing some positions of a prefix of a timed word and the corresponding temporal sequence. Assume that clock x has initial value in the interval $[0,1)$, according to Prop. 10. Based on the temporal sequence $\overrightarrow{\Delta}$, the clock assignment for x in every position is determined by the initial value of x plus the time elapsed since position 0. Curly brackets indicate the positions that satisfy a certain constraint. Assume that the valid constraints are those shown in the figure, whose value depends on the value of x and on the constraints enforced by formula ϕ. The positions that are indicated with a big circle are the ones that are relevant to determine the initial assignment for a clock, as the intermediate ones are implicit. From the first constraint $1 < x < 2$, given the delays defined by the temporal sequence, the value $\sigma(0,x)$ of clock x at position 0 such that $1 < x + 0.4$ and $x + 1.2 < 2$ both hold can only be in the interval $(0.6, 0.8)$. Similarly, the second constraint entails

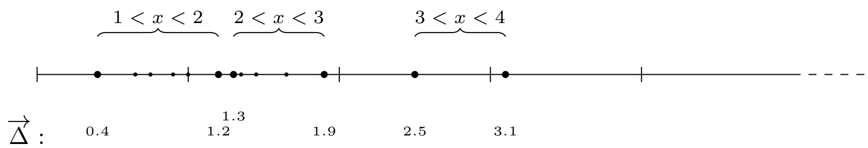

Figure 3: A timed sequence and the positions satisfying different clock constraints.

that $\sigma(0, x) \in (0.7, 1)$ and the third entails $\sigma(0, x) \in (0.5, 0.9)$. Therefore, all constraints can be satisfied together if $\sigma(0, x) \in (0.7, 0.8)$.

Before enunciating the lemma, recall that a sequence of region distances $\overrightarrow{\delta} = \delta_1 \ldots \delta_m$ induces a corresponding temporal sequence $\overrightarrow{\Delta} = \Delta_1 \ldots \Delta_m$ as follows: for every i, $1 \leq j \leq m$, let $\Delta_j = \sum_{1 \leq i \leq j} \delta_i$; also, the maximal partition $\mathcal{P}_{\overrightarrow{\delta}}$ for $\overrightarrow{\delta}$ is just the maximal partition $\mathcal{P}_{\overrightarrow{\Delta}}$ (see end of Sec. 4).

Lemma 3. *For all $m \geq 0$, for all sequences $\overrightarrow{\delta} = \delta_1 \ldots \delta_m$ of positive real numbers:*

1. *If a monotonic sequence $\mathcal{R}_m = (0, 1)R_1 \ldots R_m$ is feasible for $\overrightarrow{\delta}$, then there exists $I_0 \in \mathcal{P}_{\overrightarrow{\delta}}$ maximally compatible with $\mathcal{R}_m$ for $\overrightarrow{\delta}$.*
2. *For all $I_0 \in \mathcal{P}_{\overrightarrow{\delta}}$ there exists one, and only one, monotonic sequence $\mathcal{R}_m = (0, 1)R_1 \ldots R_m$ feasible for $\overrightarrow{\delta}$, such that I_0 is compatible with $\mathcal{R}_m$ for $\overrightarrow{\delta}$.*

Proof. We prove Parts 1 and 2 together.

The proof is by induction on $m \geq 0$. In the base case $m = 0$, the sequence δ is empty and thus $\mathcal{P}_{\overrightarrow{\delta}} = \{(0, 1)\}$. It is then vacuously true that the sequence of regions $\mathcal{R}_0 = (0, 1)$ is feasible for δ and $I_0 = (0, 1)$ is maximally compatible with $\mathcal{R}_0$; also Part (2) is obvious.

Assume the induction hypothesis holds for $m \geq 0$. Let $\overrightarrow{\delta}'$ be a sequence of $m + 1$ positive reals $\delta_1 \ldots \delta_m \delta_{m+1}$, corresponding to a maximal partition $\mathcal{P}_{\overrightarrow{\delta}'}$. We apply Part (1) of the induction hypothesis: if $\mathcal{R}_m = (0, 1)R_1 \ldots R_m$ is a feasible sequence for $\overrightarrow{\delta}$, then $I'_0 \in \mathcal{P}_{\overrightarrow{\delta}}$ is a 1D-subregion maximally compatible with $\mathcal{R}_m$. Let $I'_i \subseteq R_i$ be defined by $I'_0 \rightsquigarrow_{\Delta_i} I'_i$, for all $1 \leq i \leq m$.

We show, given the positive real δ_{m+1}, how the new partition $\mathcal{P}_{\overrightarrow{\delta}'}$ may differ from $\mathcal{P}_{\overrightarrow{\delta}}$. We first identify, depending on the value δ_{m+1}, a 1D-region R_{m+1} such that $\mathcal{R}_{m+1} = (0, 1)R_1 \ldots R_m R_{m+1}$ is feasible for the sequence $\overrightarrow{\delta}' = \delta_1 \ldots \delta_m \delta_{m+1}$. If $\Delta_{m+1} \geq C$, then $R_{m+1} = (C, +\infty)$ and by definition $\mathcal{P}_{\overrightarrow{\delta}'} = \mathcal{P}_{\overrightarrow{\delta}}$, i.e., $\mathcal{P}_{\overrightarrow{\delta}}$ is a maximal partition also for sequence $\overrightarrow{\delta}'$: moreover, it is obvious in this case that there is I'_{m+1} such that $I'_m \rightsquigarrow_{\delta_{m+1}} I'_{m+1}$, hence $I'_0 \rightsquigarrow_{\delta_1} I'_1 \cdots \rightsquigarrow_{\delta_m} I'_m \rightsquigarrow_{\delta_{m+1}} I'_{m+1}$. To prove Part 1, we can select $I_0 = I'_0$, which is compatible also with the sequence $\mathcal{R}_{m+1}$. I'_0 is also maximally compatible with $\mathcal{R}_{m+1}$ since it is maximally compatible with $\mathcal{R}_m$, thus verifying Part 1. Part 2 in this case follows immediately from the same part of the induction hypothesis, by defining $I_0 = I'_0$, and by the uniqueness of R_{m+1}.

Assume now $\Delta_{m+1} < C$. Let n and η be, respectively, the integer and the fractional part of Δ_{m+1}. By Statement 1, 1D-region R_{m+1} must be one of $(n, n+1)$, $(n+1, n+1)$, $(n+1, n+2)$. When $n = C-1$, 1D-region $(n+1, n+2)$ corresponds to $(C, +\infty)$. In this case, for simplicity, with a slight abuse of notation we will still indicate the region as $(n+1, n+2)$.

If $\eta = 0$ then let $I'_0 = (\alpha, \beta)$, for some $0 \leq \alpha < \beta \leq 1$. Then, $R_{m+1} = (n, n+1)$ is the only possible region reachable from $(0, 1)$ with $\Delta_{m+1} = n$. . Let $I_0 = I'_0, I_1 = I'_1, \ldots, I_m = I'_m$. Then, let I_{m+1} be the 1D-subregion such that $I_m \rightsquigarrow_{\delta_{m+1}} I_{m+1}$, with $I_{m+1} = (n+\alpha, n+\beta) \subseteq R_{m+1}$, since I_m must be $(\alpha+n-\delta_{m+1}, \beta+n-\delta_{m+1})$, hence $\mathcal{R}_{m+1} = \mathcal{R}_m R_{m+1}$ is feasible for the sequence δ'. I_0 is maximally compatible with $\mathcal{R}_{m+1}$ since it is maximally compatible with $\mathcal{R}_m$ and $I_m \rightsquigarrow_{\delta_{m+1}} I_{m+1}$.

If $\eta > 0$ there are two cases, whether the punctual interval $(1-\eta, 1-\eta)$ is in the partition $\mathcal{P}_{\overrightarrow{\delta}}$ or not.

Case $(1-\eta, 1-\eta) \notin \mathcal{P}_{\overrightarrow{\delta}}$. Then, there exist α, β, such that $0 \leq 1-\alpha < 1-\eta < 1-\beta \leq 1$ and $(1-\alpha, 1-\beta) \in \mathcal{P}_{\overrightarrow{\delta}}$. Then, $\mathcal{P}_{\overrightarrow{\delta}'} = \mathcal{P}_{\overrightarrow{\delta}} \cup \{(1-\alpha, 1-\eta), (1-\eta, 1-\eta), (1-\eta, 1-\beta)\} - \{(1-\alpha, 1-\beta)\}$. We separate three subcases: (1) $I'_0 = (1-\alpha, 1-\beta)$; (2) $I'_0 \prec (1-\alpha, 1-\beta)$; (3) $(1-\alpha, 1-\beta) \prec I'_0$.

1. Subcase $I'_0 = (1-\alpha, 1-\beta)$ (i.e., $1-\alpha < 1-\eta < 1-\beta$).

   We consider each possible form for R_{m+1}, using Statement 1 to define $I_0 \subseteq I'_0 = (1-\alpha, 1-\beta)$ leading to a subregion $I_{m+1} \subseteq R_{m+1}$.

   if $R_{m+1} = (n, n+1)$ then let $I_0 = (1-\alpha, 1-\eta) \subseteq (1-\alpha, 1-\beta)$, hence $I_0 \subseteq (0, 1-\eta)$, $I_{m+1} = (n+\eta+1-\alpha, n+1) \subseteq (n, n+1)$.

   if $R_{m+1} = (n+1, n+1)$ then let $I_0 = (1-\eta, 1-\eta) \subseteq (1-\alpha, 1-\beta)$: $I_{m+1} = (n+1, n+1)$;

   if $R_{m+1} = (n+1, n+2)$ then let $I_0 = (1-\eta, 1-\beta) \subseteq (1-\alpha, 1-\beta)$: $I_0 \subseteq (1-\eta, 1)$, $I_{m+1} = (n+1, n+\eta+1-\beta) \subseteq (n+1, n+2)$.

   Since for each form of R_{m+1} we selected $I_0 \subseteq I'_0$ such that $I_{m+1} \subseteq R_{m+1}$, I_0 is compatible with $\mathcal{R}_{m+1}$. To prove maximality of I_0, consider a real value $\gamma \notin I_0$, $0 < \gamma < 1$ and let $I = (\gamma, \gamma)$. If $\gamma \notin I'_0$, then by maximality of I'_0 it follows that the 1D-subregion I is not compatible with $\mathcal{R}_m$; therefore, I is also not compatible with $\mathcal{R}_{m+1} = \mathcal{R}_m R_{m+1}$. If $\gamma \in I'_0 = (1-\alpha, 1-\beta)$, then let $I \rightsquigarrow_{n+\eta} (n+\eta+\gamma, n+\eta+\gamma) = I'$. We consider the three above cases for I_0. If $I_0 = (1-\alpha, 1-\eta)$, then $1-\eta \leq \gamma < 1-\beta$: since $\gamma \geq 1-\eta$, it follows that $n+\eta+\gamma \geq n+1$, thus $I' \nsubseteq R_{m+1} = (n, n+1)$. The case $I_0 = (1-\eta, 1-\beta)$ is symmetrical. The case $I_0 = (1-\eta, 1-\eta)$ is obvious, since if $\gamma \neq (1-\eta)$ then $I' \neq (n+1, n+1)$. Hence, I_0 is maximally compatible with $\mathcal{R}_{m+1}$, i.e., Part 1 holds. Part 2 follows from the fact that for each I_0 in the three possible subcases there is exactly one corresponding 1D-region R_{m+1}.

2. Subcase $I'_0 = (1-\lambda, 1-\kappa)$, with $0 \leq 1-\lambda \leq 1-\kappa \leq 1-\alpha < 1-\eta$, i.e., $I'_0 \subseteq (0, 1-\eta)$. We claim that it is enough to let $I_0 = I'_0$. By

Statement 1, $I_{m+1} \subseteq (n, n+1)$. By maximality of the partition $\mathcal{P}_{\overrightarrow{\delta}}$, there exist $1 \leq h \leq m$ and $k \in \mathbb{N}_{\geq 0}$ such that $k+\kappa = \delta_1+\delta_2+\cdots+\delta_h$. By Statement 1, if $I_0 \subseteq (0, 1-\kappa)$ and $I_0 \rightsquigarrow_{k+\kappa} I_h$, we have $I_h \subseteq (k, k+1)$. Hence, $R_h = (k, k+1)$ or $R_h = (k+1, k+1)$ by induction hypothesis. Since $1-\kappa < 1-\eta$, we have $\kappa > \eta$. By Proposition 11 (with $R = R_h, \gamma = \kappa, g = k$ and $R' = R_{m+1}, g' = n, \gamma' = \eta$), the only possible value for R_{m+1} to make $\mathcal{R}_{m+1}$ feasible is $R_{m+1} = (n, n+1)$. Hence, I_0 is compatible with $\mathcal{R}_{m+1}$, while maximality of I_0 follows from the maximality of I'_0: Part 1 holds.

Part 2 is immediate by induction hypothesis and uniqueness of R_{m+1}.

3. Subcase $I'_0 = (1-\kappa, 1-\lambda)$, with $1-\eta < 1-\kappa \leq 1-\lambda$. This case is symmetrical to the previous one and just briefly sketched here: there are h, k such that $k+\kappa = \delta_1+\delta_2+\cdots+\delta_h$ and $R_h = (k+1, k+2)$ by Statement 1. By Proposition 11, third case (with $R = R_{m+1}, g = n, \gamma = \eta$ and $R' = R_h, \gamma' = \kappa, g' = k$), $R_{m+1} = (n+1, n+2)$. Let $I_0 = I'_0$: $I_{m+1} \subseteq (n+1, n+2)$ by Statement 1.

Case $(1-\eta, 1-\eta) \in \mathcal{P}_{\overrightarrow{\delta}}$. Then, $\mathcal{P}_{\overrightarrow{\delta}'} = \mathcal{P}_{\overrightarrow{\delta}}$. By maximality of the partition $\mathcal{P}_{\overrightarrow{\delta}}$, there exist h and k, with $1 \leq h \leq m$ and $k \in \mathbb{N}_{\geq 0}$ such that $k + \eta = \delta_1+\delta_2+\cdots+\delta_h$. Since $n+\eta = \delta_1+\cdots+\delta_{m+1}$, it follows that $\delta_{h+1}+\cdots+\delta_{m+1} = n-k$. Since $(1-\eta, 1-\eta) \in \mathcal{P}_{\overrightarrow{\delta}}$, there exist α, β, with $0 \leq 1-\alpha < 1-\eta < 1-\beta \leq 1$ such that both $(1-\alpha, 1-\eta), (1-\eta, 1-\beta)$ are in $\mathcal{P}_{\overrightarrow{\delta}}$. By Proposition 11, case $\gamma' = \gamma$, 1D-region R_h uniquely determines R_{m+1}.

Let $I_0 = I'_0$ and let $I_i = I'_i$, for all $1 \leq i \leq m$. Again, by induction hypothesis I_0 is maximally compatible with $\mathcal{R}_m$. We now show that in each of the possible values for $I'_0 = I_0$, 1D-region R_{m+1} makes $\mathcal{R}_{m+1}$ feasible; moreover, we can define I_{m+1} such that $I_0 \rightsquigarrow_{n+\eta} I_{m+1}$, showing that $I_{m+1} \subseteq R_{m+1}$ by applying Statement 1. Thus, I_0 is compatible with $\mathcal{R}_{m+1}$ and, by the maximal compatibility of I_0 with $\mathcal{R}_m$, it is also maximally compatible, thus proving Part 1.

1. Subcase $I'_0 = (1-\alpha, 1-\eta)$. We have $I_{m+1} = (n+\eta+1-\alpha, n+1) \subseteq (n, n+1)$ (since $\eta+1-\alpha < 1$, being $1-\alpha < 1-\eta$ by hypothesis). $I_h = (k+\eta+1-\alpha, k+1)$, hence $R_h = (k, k+1)$. Therefore, $R_{m+1} = (n, n+1) \supseteq I_{m+1}$ makes $\mathcal{R}_{m+1}$ feasible.
2. Subcase $I'_0 = (1-\eta, 1-\eta)$. We have $I_{m+1} = (n+1, n+1)$ and $I_h = (k+1, k+1)$, which is also R_h. Therefore, $R_{m+1} = (n+1, n+1) = I_{m+1}$ makes $\mathcal{R}_{m+1}$ feasible.
3. Subcase $I'_0 = (1-\eta, 1-\beta)$. We have $I_{m+1} = (n+1, n+\eta+1-\beta) \subseteq (n+1, n+2)$. Again, $I_h \subseteq (k+1, k+2) = R_h$. Therefore, $R_{m+1} = (n+1, n+2)$ makes $\mathcal{R}_{m+1}$ feasible.

The subcases where $I'_0 \prec (1-\alpha, 1-\beta)$ or $(1-\alpha, 1-\beta) \prec I'_0$ are identical to the subcases 2 and 3 of the previous Case $(1-\eta, 1-\eta) \notin \mathcal{P}_{\overrightarrow{\delta}}$ and may be skipped.

Part 2 derives from the fact that, in each of three above subcases for I_0, exactly one region R_{m+1} was shown to exist, such that $I_{m+1} \subseteq R_{m+1}$, with $I_0 \rightsquigarrow_{\delta_1+\dots+\delta_{m+1}} I_{m+1}$, and $(0,1)R_0 \dots R_{m+1}$ is feasible for $\delta_1, \dots, \delta_{m+1}$: by induction hypothesis, $(0,1)R_0 \dots R_m$ is the only feasible sequence for $\delta_1, \dots, \delta_m$, thus uniqueness is proved.

□

Elimination of non-initialized clocks

Let $\widetilde{X}$ be a set of clocks. Define a new set D of clocks $d_{i,x}$, for all $0 \leq i \leq 1+3C$ and for all $x \in \widetilde{X}$. For simplifying some of the following formulae, we add *Now* to set D. Let $Z_{\widetilde{X}}$ be the set of region clocks for $\widetilde{X}$, which is included in the set $Z_{D\cup\widetilde{X}}$ of the region clocks for $D \cup \widetilde{X}$.

We summarize here the remainder of the proof. In the proof of Theorem 3, we define a formula ϕ', language equivalent to ϕ, such that all its clocks are well-initialized. Formula ϕ' includes region clocks $Z_{\widetilde{X}}$ over $\widetilde{X}$, that are used to replace the clock constraints over the clocks X of ϕ until their first reset. Each clock $\tilde{x} \in \widetilde{X}$ is a copy of a clock $x \in X$, behaving in the same way until the first reset of x: $\tilde{x}$ is instead never reset. The region clocks in $Z_{\widetilde{X}}$ keep track of the regions visited by clocks in $\widetilde{X}$ so that clock constraints of the form $x \sim c$ can be replaced, before a reset, by $[\tilde{x} \sim c]$. The regions of $\tilde{x}$ are relevant in ϕ' only in the prefix ending at the first reset of the corresponding clock x, since after this reset formula ϕ' may use the actual value of x. Since formula ϕ' does not use the actual value of a clock $x \in X$ before the first reset of x itself, the initial value of x can be assumed to be 0—i.e., well-initialized. Formula ϕ' does not actually include clocks in $\widetilde{X}$, but only the region clocks $Z_{\widetilde{X}}$; the latter can always be assumed to be well-initialized, as the only relevant value of a region clock is whether it is greater than 0 or not. To define the correct evolution of clocks in X (and $\widetilde{X}$) by means of the region clocks in $Z_{\widetilde{X}}$, the well-initialized set D of clocks is used to measure the time distance among the positions of the model when x enters or leaves a region, with some suitable constraints over clocks D added to ϕ'. Given a model for ϕ, inducing a temporal sequence $\overrightarrow{\delta}$ and a region sequence for the clocks of X, there exists a model of ϕ' such that the clocks in D determine the same sequence $\overrightarrow{\delta}$, which is compatible with the region sequence for clocks $\widetilde{X}$.

Lemma 5 is fundamental to show that the evolution of the regions captured by clocks in $Z_{\widetilde{X}}$ can be determined only by using clocks in D, without actually "bridging" the region clocks $Z_{\widetilde{X}}$ and the clocks in $\widetilde{X}$. Thus, clocks in $\widetilde{X}$ can be eliminated and do not actually appear in the formula ϕ' of Theorem 3. Note that it would be possible to prove the same result for region clocks Z_X, instead of $Z_{\widetilde{X}}$; however, having a different set of clocks $\widetilde{X}$, which cannot be reset, simplifies both the statement and the proof of Lemma 5, and allows us to directly reuse the previous results on complete and monotonic sequences of regions.

Define the following shorthand, with the intended meaning that there is region change for x if the current region is different from the previous or the next one (which is captured by a change in one of the clocks $z_{[x\sim c]}$ associated with x):

$$\mathit{change}_x \stackrel{\text{def}}{=} \bigvee_{z_{[x\sim c]}\in Z} \left(\neg[x \sim c] \wedge \left(\mathbf{Y}\left([x \sim c]\right) \vee \mathbf{X}\left([x \sim c]\right)\right)\right)$$

As defined by the following formula, clock $d_{i,x}$ is reset at the i-th change of region as defined by change_x:

$$\mathit{reset}(d_{i,x}) \stackrel{\text{def}}{=} d_{i,x} = 0 \Leftrightarrow \mathit{change}_x \wedge \mathbf{Y}\left(\left(\bigwedge_{1\leq j\leq 3C} d_{j,x} > 0\right) \mathbf{S}(d_{i-1,x} = 0)\right)$$

Next, formula $\mathit{init}_{\widetilde{X}}$ captures sequences of regions in which each x is in $[0, 1)$ or greater than C in the initial instant, while $\mathit{noreset}(Z_{\widetilde{X}})$ that each x is never 0 after the first position:

$$\mathit{init}(Z_{\widetilde{X}}) = \bigwedge_{x\in\widetilde{X}} \left([x < 1] \vee [C < x]\right)$$

$$\mathit{noreset}(Z_{\widetilde{X}}) = \bigwedge_{x\in\widetilde{X}} \mathbf{XG}(\neg[x = 0])$$

while init_D captures the correct initialization of the new clocks in D and the fact that $d_{0,x}$ is never reset outside position 0:

$$\mathit{init}_D = \bigwedge_{\substack{x\in\widetilde{X}\\ 1\leq i\leq 3C}} d_{i,x} > 0 \wedge d_{0,x} = 0 \wedge \mathbf{XG}(d_{0,x} > 0)$$

Finally, upd_D is defined as follows:

$$\mathit{upd}_D = \ \Theta_{Z_{D\cup\widetilde{X}},C} \wedge \mathit{bridge}(D) \wedge \mathit{init}_D \wedge \bigwedge_{\substack{x\in\widetilde{X}\\ 1\leq i\leq 3C}} \mathbf{G}\left(\mathit{reset}(d_{i,x})\right)$$

Correspondence between clock assignments for $\widetilde{X}$ and for $D \cup Z_{\widetilde{X}}$.

The *natural sequence* for clock $x \in \widetilde{X}$ for a model (π, σ) of the region automaton $\Theta_{Z_{\widetilde{X}},C}$—with $\sigma : \mathbb{N}_{\geq 0} \times (\{\mathit{Now}\} \cup Z_{\widetilde{X}}) \to \mathbb{R}_{\geq 0}$—is the (unique) infinite sequence $R_0(x)R_1(x)\ldots$ of 1D-regions such that, for every $i \in \mathbb{N}_{\geq 0}$, $(\pi, \sigma), i \models [\![R_i(x)]\!]$ (recall that $[\![R_i(x)]\!]$ is the maximally consistent set of clock constraints on the region clocks $Z_{\widetilde{X}}$ that represents the region R_i).

The following property ensures that if a 1D-subregion $I_0 \in R_0(x)$ for a clock $x \in \widetilde{X}$ is compatible with a sequence of length limited by $3C$, then it is also compatible with the infinite natural sequence. This immediately entails that the number of clocks in D is bounded by $(1 + 3C)|\widetilde{X}|$, which will allow us to replace the set of clocks $\widetilde{X}$ with a bounded number of well-initialized clocks.

Lemma 4. *Let (π, σ) be a model of $init(Z_{\widetilde{X}}) \wedge noreset(Z_{\widetilde{X}}) \wedge upd_D$, with $\sigma : \mathbb{N}_{\geq 0} \times (\{Now\} \cup D \cup Z_{D\cup\widetilde{X}}) \to \mathbb{R}_{\geq 0}$; let $\Delta_i = \sigma(i, Now) - \sigma(0, Now)$ and $\delta_i = \Delta_i - \Delta_{i-1}$ for all $i \geq 1$, and let $R_0(x)R_1(x)\dots$ be the natural sequence of σ for x.*

For all $x \in \widetilde{X}$ there exist one, and only one, value $m \leq 1+3C$ and one, and only one, sequence of m positions $0 = i_0 < i_1 < \dots < i_m \in \mathbb{N}_{\geq 0}$ such that:

1. *for all $i \in \mathbb{N}_{\geq 0}$, if $R_i(x) \neq R_{i+1}(x)$, then there is $0 \leq j \leq m-1$ such that $i_j = i, i_{j+1} = i+1$;*
2. *the sequence of 1D-regions $\mathcal{R}_m = R_{i_0}(x)R_{i_1}(x)\dots R_{i_m}(x)$ is compactly monotonic and complete;*
3. *$\mathcal{R}_m$ is feasible for $\overrightarrow{\delta'} = \Delta_{i_1} - \Delta_{i_0}, \dots, \Delta_{i_j} - \Delta_{i_{j-1}}, \dots, \Delta_{i_m} - \Delta_{i_{m-1}}$.*
4. *If a 1D-subregion $I_0 \subseteq (0,1)$ is compatible with $\mathcal{R}_m$ for $\overrightarrow{\delta'}$, then I_0 is compatible with the natural sequence $R_0(x)R_1(x)\dots$ for $\delta_1\delta_2\dots$.*

Proof. Parts (1) and (2) follow from the definition of *reset* and $change_x$. In fact, the positions $i_1, \dots, i_m$ are exactly those where $change_x$ holds; moreover, $reset(d_{j,x})$ holds whenever at position i_j both $change_x$ holds and the last position where $change_x$ held was i_{j-1}.

Part (3) follows from the fact that at a position i_j the only clock to be reset is $d_{j,x}$. Therefore, at position i_j the value of clock $d_{h,x}$, for $h < j$, is equal to $\Delta_{i_j} - \Delta_{i_h}$. Moreover, the region automaton in upd_D ensures that the region clocks in $Z_{\widetilde{X}}$ are consistent with the region clocks in Z_D, and the latter are consistent with the actual values of clocks in D by virtue of the $bridge(D)$ formula in upd_D. Therefore, the sequence of regions is feasible for $\overrightarrow{\delta'}$—the definition of feasibility (for a clock x) considers all possible distances between pairs of regions $(R_{i_j}(x), R_{i_h}(x))$, which are tracked by the clocks $d_{0,x}, d_{1,x}, \dots$ in D.

We now prove Part (4). By definition of compatible subregion, there exist m non-empty subregions, here denoted $I_{i_1} \subseteq R_{i_1}, \dots, I_{i_m} \subseteq R_{i_m}$ such that $I_{i_{j-1}} \rightsquigarrow_{\delta'_j} I_{i_j}$ for every $1 \leq j \leq m$, where $\delta'_j = \sum_{i_{j-1} < i \leq i_j} \delta_i$. By definition of compactly monotonic sequence, for every $1 \leq j \leq m$, if $i \in \mathbb{N}_{\geq 0}$ is such that $i_{j-1} < i < i_j$, then we have $R_i = R_{i_{j-1}} = R_{i_j}$. By definition of δ'_j and by Proposition 6, Part 2, there exist 1D-subregions $I_{1+i_{j-1}}, \dots, I_{i_j - 1}$ such that $I_{i_{j-1}} \rightsquigarrow_{\delta_{1+i_{j-1}}} I_{1+i_{j-1}} \cdots \rightsquigarrow_{\delta_{i_j}} I_{i_j}$. Every finite prefix of the natural sequence is monotonic because $init(Z_{\widetilde{X}}) \wedge noreset(Z_{\widetilde{X}})$ guarantees that for each clock x, $z_{[x=0]}$ always captures that $x = 0$ is false—that is, 1D-region $(0,0)$ is never reached, apart possibly in position 0. Then, it holds that $I_{1+i_{j-1}} \subseteq R_{i_{j-1}} = R_{i_j}, \dots, I_{i_j - 1} \subseteq R_{i_j - 1} = R_{i_j}$. Therefore, we can find a 1D-subregion I_i for every position i from 0 to i_m. Since R_{i_m} is $(C, +\infty)$, also $R_i = (C, +\infty)$ for every $i \geq i_m$, hence just define I_i to be the 1D-subregion such that $I_{i_m} \rightsquigarrow_{\sigma(i,Now) - \sigma(i_m, Now)} I_i$ for every $i > i_m$. □

The following lemma is fundamental to prove the existence of an initial assignment for all the clocks in $\widetilde{X}$ if there is a clock assignment satisfying formula upd_D. The proof is based on Lemma 3, which guarantees the existence of a non-empty 1D-subregion for each clock, and on Lemma 4.

Notice that no clock of $\widetilde{X}$ appears in upd_D, since the latter does not include the bridge formulae over clocks in $\widetilde{X}$. The idea of the next lemma is to show that the bridge formula over $\widetilde{X}$ is indeed not necessary to evaluate correctly the region clocks of $Z_{\widetilde{X}}$.

Lemma 5. *Let ψ be a diagonal-free CLTLoc formula defined over a set of clocks Y such that $Y \cap (D \cup \widetilde{X}) = \{Now\}$; Y may also include region clocks of $Z_{\widetilde{X}}$, but only of the form $z_{[x\sim c]}$. Then, the following two formulae are language equivalent:*

$$\psi_D = \psi \wedge init(Z_{\widetilde{X}}) \wedge noreset(Z_{\widetilde{X}}) \wedge upd_D \tag{6}$$

$$\psi_{\widetilde{X}} = \psi \wedge init(Z_{\widetilde{X}}) \wedge noreset(Z_{\widetilde{X}}) \wedge bridge(\widetilde{X}) \wedge \Theta_{Z_{\widetilde{X}},C} \tag{7}$$

Moreover, every timed word in the language of ψ_D has a model where all clocks in $D \cup Z_{\widetilde{X}}$ are well-initialized.

Proof. We first prove that every timed word (π, τ) in the language of $\psi_{\widetilde{X}}$ is also in the language of ψ_D.

Let $\sigma_{\widetilde{X}} : \mathbb{N}_{\geq 0} \times (\{Now\} \cup Y \cup \widetilde{X} \cup Z_{\widetilde{X}}) \to \mathbb{R}_{\geq 0}$ be a clock assignment such that $[(\pi, \sigma_{\widetilde{X}})] = (\pi, \tau)$; thus, $(\pi, \sigma_{\widetilde{X}}) \models \psi \wedge init(Z_{\widetilde{X}}) \wedge noreset(Z_{\widetilde{X}}) \wedge bridge(\widetilde{X}) \wedge \Theta_{Z_{\widetilde{X}},C}$.

We define a clock assignment $\sigma_D : \mathbb{N}_{\geq 0} \times (D \cup Y \cup Z_{D \cup \widetilde{X}}) \to \mathbb{R}_{\geq 0}$ such that $\sigma_D(i, z) = \sigma_{\widetilde{X}}(i, z)$ for all $i \in \mathbb{N}_{\geq 0}$, $z \in \{Now\} \cup Y \cup Z_{\widetilde{X}}$. We have to complete σ_D by assigning values also to the other clocks in D, but obviously by the previous assignments we already ensured that $(\pi, \sigma_D) \models \psi \wedge init(Z_{\widetilde{X}}) \wedge noreset(Z_{\widetilde{X}}) \wedge \Theta_{Z_{\widetilde{X}},C}$. Since no clock in set $D \cup \widetilde{X}$ is in ψ, if $(\pi, \sigma_{\widetilde{X}})$ satisfies ψ, then also (π, σ_D) satisfies ψ.

Since no clock in $\widetilde{X}$ is reset after the first position according to $\sigma_{\widetilde{X}}$, for each $x \in \widetilde{X}$ there exists $n > 0$ such that the clock assignment $\sigma_{\widetilde{X}}$ defines a complete monotonic sequence of n 1D-regions, one region for each position, which can be enumerated as $R_0 R_1 \dots R_n$, with $\sigma_{\widetilde{X}}(i, x) \in R_i$ for all $0 \leq i \leq n$, with $R_n = (C, +\infty)$ and R_0 being either $(0, 0)$ or $(0, 1)$.

Let $R_{i_0} R_{i_1} \dots R_{i_m}$, with $m \leq n$ be the complete, compactly monotonic sequence extracted from $R_0 R_1 \dots R_n$, with $R_{i_0} = R_0$.

We complete the definition of σ_D as follows, by assigning the values of the clocks in D so that $(\pi, \sigma_D) \models upd_D$. For every $x \in \widetilde{X}$, $\sigma_D(0, d_{0,x}) = 0$.

For all $1 \leq j \leq m - 1$, for all $h \in \mathbb{N}_{\geq 0}$:

1. If $0 \leq h < i_j$, then let $\sigma_D(h, d_{j,x}) = \sigma_{\widetilde{X}}(h, Now)$.
2. If $i_j \leq h$, then let $\sigma_D(h, d_{j,x}) = \sigma_{\widetilde{X}}(h, Now) - \sigma_{\widetilde{X}}(i_j, Now)$.

We also assign the values of region clocks $Z_{D \cup \widetilde{X}}$ that are not in $Z_{\widetilde{X}}$ to match the values assigned to clocks D and $\widetilde{X}$ to satisfy $\mathit{bridge}(D)$ and the region automaton $\Theta_{Z_{D \cup \widetilde{X}}, C}$; for example, for all $i \in \mathbb{N}_{\geq 0}$ we assign $\sigma_D(i, z_{[d_{i,x} \sim c]}) = 0$ if, and only if, $\sigma_D(i, d_{i,x}) \sim c$ holds and $\sigma_D(i, z_{[d_{i,x} \sim x+c]}) = 0$ if, and only if, $\sigma_D(i, d_{i,x}) \sim \sigma_{\widetilde{X}}(i, x) + c$.

By construction, σ_D is well-initialized (since region clocks can always be modified to be well-initialized) and satisfies upd_D. It is immediate to verify that mapping σ_D is a clock assignment, since at each position $h \geq 1$ all clocks, which are not reset, are incremented of the same amount $\sigma_{\widetilde{X}}(h, \mathit{Now}) - \sigma_{\widetilde{X}}(h-1, \mathit{Now})$.

We now prove that every timed word (π, τ) in the language of ψ_D is also in the language of $\psi_{\widetilde{X}}$.

First, we notice that in every model σ_D satisfying upd_D, all clocks in D and all region clocks in $Z_{\widetilde{X}}$ can be assumed to be well-initialized. Hence, let $\sigma_D : \mathbb{N}_{\geq 0} \times (Y \cup D \cup Z_{D \cup \widetilde{X}}) \to \mathbb{R}_{\geq 0}$ be a well-initialized clock assignment such that $[(\pi, \sigma_D)] = (\pi, \tau)$, thus satisfying $\psi \wedge \mathit{upd}_D \wedge \mathit{init}(Z_{\widetilde{X}}) \wedge \mathit{noreset}(Z_{\widetilde{X}})$. Notice that by definition of upd_D, formula $\Theta_{Z_{D \cup \widetilde{X}}, C}$ holds, which implies that also $\Theta_{Z_{\widetilde{X}}, C}$ is satisfied. Thus, the values of clocks in $Z_{\widetilde{X}}$ define a natural sequence of regions $R_0(x) R_1(x) \ldots$ for every clock x, one region for each position. Notice that $\mathit{init}(Z_{\widetilde{X}}) \wedge \mathit{noreset}(Z_{\widetilde{X}})$ symbolically imposes that 1D-region $(0, 0)$ is never reached for all clocks in $\widetilde{X}$ (no clock reset), apart possibly in position 0, hence the sequence is monotonic. By Lemma 4, Parts (1), (2) and (3), there exist a value m, with $1 \leq m \leq 3C$, and $m+1$ positions $0 = i_0 < i_1 < \cdots < i_m$ such that $\mathcal{R}_m = R_{i_0}(x) R_{i_1}(x) \ldots R_{i_m}(x)$ is a complete and compactly monotonic sequence which is feasible for region distances $\delta_{i_1}, \ldots, \delta_{i_m}$, with $R_0 = R_{i_0}$.

We now define a clock assignment $\sigma_{\widetilde{X}} : \mathbb{N}_{\geq 0} \times (Y \cup \{\mathit{Now}\} \cup \widetilde{X} \cup Z_{\widetilde{X}}) \to \mathbb{R}_{\geq 0}$ satisfying also $\mathit{bridge}(\widetilde{X})$. For every $i \in \mathbb{N}_{\geq 0}$, let $\sigma_{\widetilde{X}}(i, z) = \sigma_D(i, z)$ for all $z \in Y \cup \{\mathit{Now}\} \cup Z_{\widetilde{X}}$. If $R_0(x) = (0, 0)$, then for every $h \geq 0$, let $\sigma_{\widetilde{X}}(h, x) = \sigma_D(h, \mathit{Now}) - \sigma_D(0, \mathit{Now})$. It is obvious that $\sigma_{\widetilde{X}}$ is a clock assignment for x and it is in agreement with the regions of x assigned by σ_D.

If $R_0(x) = (0, 1)$, by Lemma 3, Part 1, applied to sequence $\mathcal{R}_m$, there exists a non-empty 1D-subregion $I_0(x) \subseteq R_{i_0}(x) = R_0(x)$, compatible with $\mathcal{R}_m$ for $\delta_{i_1}, \ldots, \delta_{i_m}$. Let $\sigma_{\widetilde{X}}(0, x) = \alpha$, where α is any value in I_0. For every $h > 0$, let $\sigma_{\widetilde{X}}(h, x) = \alpha + \sigma_D(h, \mathit{Now}) - \sigma_D(0, \mathit{Now})$. It is obvious that $\sigma_{\widetilde{X}}$ is a clock assignment (although in general it is not well-initialized). We show that it is in agreement with σ_D. By Lemma 4, Part (4), $I_0(x)$ is also compatible with the sequence $R_0(x) R_1(x) \ldots$ for the region distances $\delta_h = \sigma_D(h, \mathit{Now}) - \sigma_D(h - 1, \mathit{Now}) = \sigma_{\widetilde{X}}(h, \mathit{Now}) - \sigma_{\widetilde{X}}(h - 1, \mathit{Now})$, $1 \leq h$. Therefore, $\sigma_{\widetilde{X}}(h, x) \in R_h(x)$, which means that $\sigma_{\widetilde{X}}(h, x) \sim c$ holds if, and only if, $\sigma_{\widetilde{X}}(h, z_{[x \sim c]}) = 0$ holds. Thus, the clock assignment $\sigma_{\widetilde{X}}$ satisfies $\mathit{bridge}(\widetilde{X})$. This entails that, since Y may only include region clocks of the form $z_{[x \sim c]}$, but not of the form $z_{[x \sim y+c]}$, correctly satisfying rectangular constraints $x \sim c$ through formula $\mathit{bridge}(\widetilde{X})$ is enough to make the value of formula ψ accurate, even if the ordering of the fractional parts of the clocks of $\widetilde{X}$ does not match the value of region clocks $z_{[x \sim y+c]}$. □

Proof of Theorem 3

Given a formula ϕ defining L, we can assume by Proposition 10 that it is a diagonal-free formula defined over a set of clocks $X \cup \{Now\}$. Moreover, by the same proposition, we may assume that every non-Zeno timed word in the language of ϕ has a clock assignment σ such that for all $x \in X$ we have $0 \leq \sigma(0, x) < 1$, or $\sigma(0, x) > C$ or $\sigma(0, x) = \sigma(0, Now)$.

Let $\widetilde{X}$ be a copy of set X and let $Z_{\widetilde{X}}$ be the set of region clocks of $\widetilde{X}$.

We define a new formula $r(\phi)$, defined over the set of clocks $X \cup \{Now\} \cup Z_{\widetilde{X}}$, by replacing every clock constraint over a clock $x \in X$ with the corresponding constraint on a region clock in $Z_{\widetilde{X}}$ as long as clock x has not been reset. For every integer c, $0 < c \leq C$:

$$r(x \sim c) := (\mathbf{P}(x = 0) \Rightarrow x \sim c) \wedge (\neg \mathbf{P}(x = 0) \Rightarrow [\widetilde{x} \sim c])$$
$$r(x = 0) := x = 0$$

Recall that $\mathbf{P}(\psi)$ is an abbreviation for $\top \mathbf{S} \psi$ (i.e., ψ holds now or in the past). For instance, $r(x < c)$ replaces the constraint $x < c$ with a formula stating that, if x was reset, then $x < c$ holds, otherwise $[\widetilde{x} < c]$ holds—i.e., the symbolic region of x is such that $x < c$ holds. Therefore, $r(\phi)$ does not include clocks in $\widetilde{X}$; in addition, it includes region clocks of the form $z_{[\widetilde{x} \sim c]}$, but none of the form $z_{[\widetilde{x} \sim \widetilde{y} + c]}$—therefore it satisfies the hypotheses for formula ψ of Lemma 5. Define the following formula

$$\phi' = r(\phi) \wedge upd_D \wedge init(Z_{\widetilde{X}}) \wedge noreset(Z_{\widetilde{X}})$$

over the set of clocks $D \cup X \cup Z_{\widetilde{X}}$. Since the actual value of the clocks that are not reset at position 0 is irrelevant for the evaluation of $r(\phi)$, we can safely assume that all those clocks are equal to Now in the initial position—i.e., all clocks in a model of ϕ' can be assumed to be well-initialized.

We notice that, by Lemma 5, ϕ' is language equivalent to

$$\phi'' = r(\phi) \wedge bridge(\widetilde{X}) \wedge noreset(Z_{\widetilde{X}}) \wedge init(Z_{\widetilde{X}}) \wedge \Theta_{Z_{\widetilde{X}}, C}$$

defined over the set of clocks $X \cup Z_{\widetilde{X}} \cup \widetilde{X} \cup \{Now\}$ (i.e., it also includes the copy clocks $\widetilde{X}$, but no clock in D). Notice that, in every model of ϕ'', every constraint of the form $\widetilde{x} \sim c$ has a value compatible with that of $z_{[\widetilde{x} \sim c]}$ because of the bridge formula.

To show that ϕ is language equivalent to ϕ', it is enough to show that ϕ is language equivalent to ϕ''.

Given a timed word (π, τ) in the language of ϕ, we prove that (π, τ) is also in the language of ϕ''. Let $\sigma : \mathbb{N}_{\geq 0} \times (X \cup \{Now\}) \to \mathbb{R}_{\geq 0}$ be a clock assignment such that $[(\pi, \sigma)] = (\pi, \tau)$. Define a clock assignment $\sigma'' : \mathbb{N}_{\geq 0} \times (X \cup \{Now\} \cup Z_{\widetilde{X}} \cup \widetilde{X}) \to \mathbb{R}_{\geq 0}$ for ϕ'' as follows.

As in the proof of Lemma 1, let ι be the smallest value of all $i \in \mathbb{N}_{\geq 0}$ such that $\sigma(i, Now) - \sigma(0, Now) > C$ (which must always exist since the timed word is non-Zeno); for every $x \in X$ let ι_x be the smallest value of all $i \in \mathbb{N}_{\geq 0}$ such that $i < \iota$ and $\sigma(i, x) = 0$ if any such i exists, otherwise (with an abuse of notation) let $\iota_x = +\infty$. Let $\sigma''(i, x) = \sigma(i, x)$ for all $i \geq 0$ and $x \in X \cup \{Now\}$. For every clock $\widetilde{x} \in \widetilde{X}$, let $\sigma''(0, \widetilde{x}) = \sigma(0, x)$ and $\sigma''(i, \widetilde{x}) = \sigma(i, Now) - \sigma(0, Now) + \sigma(0, x)$. Finally, define $\sigma''(i, z_{[\widetilde{x} \sim c]}) = 0$ if, and only if, $\sigma''(i, \widetilde{x}) \sim c$ and $\sigma''(i, z_{[\widetilde{x} \sim \widetilde{y} + c]}) = 0$ if, and only if, $\sigma''(i, \widetilde{x}) \sim \sigma''(i, \widetilde{y}) + c$. Thus, (π, σ'') is a model of $bridge(\widetilde{X}) \wedge noreset(Z_{\widetilde{X}}) \wedge init(Z_{\widetilde{X}}) \wedge \Theta_{Z_{\widetilde{X}}, C}$. To show that (π, σ'') is a model of ϕ'', we are left to prove that it is a model of $r(\phi)$.

Consider in fact a subformula of the form $r(x \sim c)$: we have that for $i \geq \iota_x$, $(\pi, \sigma''), i \models r(x \sim c)$ holds if, and only if $\sigma(i, x) \sim c$ also holds; for $0 \leq i < \iota_x$, $(\pi, \sigma''), i \models r(x \sim c)$ if, and only if, $\sigma''(i, z_{[\widetilde{x} \sim c]}) = 0$, which (because of the bridge formula) holds if, and only if, $\sigma''(i, \widetilde{x}) \sim c$ holds; by definition of σ'', the latter formula, before ι_x, holds if, and only if, $\sigma(i, x) \sim c$ holds. Hence, in every position the evaluation of $r(\phi)$ according to (π, σ'') is the same of evaluation of ϕ according to (π, σ).

Given a timed word (π, τ) in the language of ϕ'', we now prove that (π, τ) is also in the language of ϕ. Let (π, σ'') be such that $[(\pi, \sigma'')] = (\pi, \tau)$, with $\sigma'' : \mathbb{N}_{\geq 0} \times (X \cup \{Now\} \cup Z_{\widetilde{X}} \cup \widetilde{X}) \to \mathbb{R}_{\geq 0}$. For every $x \in X$, let ι_x be defined as above, considering clock assignment σ''.

Let $\sigma : \mathbb{N}_{\geq 0} \times (X \cup \{Now\}) \to \mathbb{R}_{\geq 0}$ be defined as follows. For every $i \geq 0$, let $\sigma(i, Now) = \sigma''(i, Now)$; for every $x \in X$, if $i \geq \iota_x$, then let $\sigma(i, x) = \sigma''(i, x)$ and if $0 < i < \iota_x$ then let $\sigma(i, x) = \sigma(0, x) + \sigma(i, Now) - \sigma(0, Now)$.

We need to assign the initial value $\sigma(0, x)$. By Lemma 4, Part (1) to (3), the natural sequence of regions determined by region clocks in $Z_{\widetilde{X}}$ corresponds to a compactly monotonic and complete sequence $\mathcal{R}_m = R_{i_0}(x) R_{i_1}(x) \dots R_{i_m}(x)$, feasible for $\overrightarrow{\delta} = \Delta_{i_1} - \Delta_{i_0}, \dots, \Delta_{i_j} - \Delta_{i_{j-1}}, \dots, \Delta_{i_m} - \Delta_{i_{m-1}}$ (where $\Delta_i = \sigma(i, Now) - \sigma(0, Now)$).

By Lemma 3, Part 1, there exists I_0 maximally compatible with $\mathcal{R}_m$ for δ. We can select thus any value in I_0 as the initial value $\sigma(0, x)$. As in the first part of the proof, we can show that, also before ι_x, $\sigma(i, x) \sim c$ holds if, and only if $(\pi, \sigma), i \models r(x \sim c)$. ☐

We remark that in the size of formula ϕ' defined in the proof of Theorem 3, as for the case of formula ϕ' in the proof of Theorem 2, the dominant terms are the size of formula ϕ and the size of the region automaton $\Theta_{Z_{D \cup \widetilde{X}}, C}$.

7. On arbitrarily initialized Timed Automata

In this section we extend the results of Section 5 and of Section 6 to arbitrarily initialized TA, focusing on non-Zeno timed languages. To this end, we exploit Proposition 5, and in particular the translation defined in its proof which shows how, given an a.i. TA $\mathcal{A}$, we can build a CLTLoc formula $\phi_{\mathcal{A}}$ that is language equivalent with respect to $\mathcal{A}$.

Theorem 4. *Consider an arbitrarily initialized TA $\mathcal{A}$, which can include clock constraints of the form $x \sim y + c$, and which accepts the non-Zeno timed language $L_{\mathcal{A}}$. There exists an initialized TA $\mathcal{A}'$, which does not contain diagonal constraints $x \sim y + c$, which accepts $L_{\mathcal{A}}$.*

Proof. From Proposition 5, given an a.i. TA $\mathcal{A}$ we can define a CLTLoc formula $\phi_{\mathcal{A}}$ that is language equivalent with respect to $\mathcal{A}$. Thanks to the non-Zenoness of language $L_{\mathcal{A}}$ and Theorem 3 we can build a CLTLoc formula $\phi'_{\mathcal{A}}$, whose initialized timed language is the same one defined by $\phi_{\mathcal{A}}$, that is, $L_{\mathcal{A}}$. In addition, the translation defined in the proof of Theorem 3 is such that formula $\phi'_{\mathcal{A}}$ does not include diagonal constraints.

Finally, from Theorem 1, we can build from $\phi'_{\mathcal{A}}$ an initialized TA $\mathcal{A}'$ that accepts language $L_{\mathcal{A}}$. A close inspection of the translation from $\phi'_{\mathcal{A}}$ to $\mathcal{A}'$ (which ultimately is the one of the proof of Theorem 4 of [12]) shows that it does not introduce any diagonal constraints. □

Corollary 3. *A non-Zeno timed language L is recognized by an a.i. TA $\mathcal{A}$ if, and only if, there is an initialized TA $\mathcal{A}'$ that recognizes L.*

Corollary 4. *Given an a.i. TA $\mathcal{A}$ that recognizes non-Zeno timed language L, there is a diagonal-free TA $\mathcal{A}'$ that recognizes L.*

8. Closure of non-Zeno timed regular languages with respect to left quotient

Let us consider timed languages made of finite words, i.e., timed words with a finite number of positions. To denote that a timed word is finite, or that a language only includes timed words that are finite, we add a subscript F, as in (π_F, τ_F) and $\mathcal{L}_F$. A language $\mathcal{L}_F$ is hereto called *timed $*$-language* if it only includes finite timed words. A timed word (π_F, τ_F) with n positions is recognized by a TA $\mathcal{A}$ if, and only if, $(q_{i_0}, v_0) \xrightarrow[\tau_F(1)]{\pi_F(1)} (q_{i_1}, v_1) \xrightarrow[\tau_F(2)]{\pi_F(2)} (q_{i_2}, v_2), \dots, (q_{i_{n-1}}, v_{n-1}) \xrightarrow[\tau_F(n)]{\pi_F(n)} (q_{i_n}, v_n)$ is a finite run of $\mathcal{A}$ that ends in an accepting state (i.e., such that $q_{i_n} \in B$). We say that a timed $*$-language $\mathcal{L}_F$ is *timed $*$-regular* if there is a Timed Automaton recognizing all, and only, the words of $\mathcal{L}_F$.

We define the *finite left quotient* of a timed ω-language with respect to a timed $*$-language as follows.

Definition 10. Let $\mathcal{L}$ be a timed ω-language and let $\mathcal{Q}_F$ be a timed $*$-language. The *finite left quotient*, written $\mathcal{L}/_{\mathcal{Q}_F}$, of $\mathcal{L}$ with respect to $\mathcal{Q}_F$ is the timed ω-language such that a timed word (π', τ') is in $\mathcal{L}/_{\mathcal{Q}_F}$ if, and only if, there is a finite timed word (π_F, τ_F) of $\mathcal{Q}_F$ such that $(\pi_F(1), \tau_F(1)), \dots, (\pi_F(n), \tau_F(n)), (\pi'(1), \tau'(1)), (\pi'(2), \tau'(2)), \dots$ is a timed ω-word of $\mathcal{L}$.

Consider, for example, the timed $*$-language $\mathcal{Q}_F$ such that a timed word (π_F, τ_F), with $\pi_F : \{1, \dots, n\} \to \wp(\{a, b\})$ and $\tau_F : \{1, \dots, n\} \to \mathbb{R}_{\geq 0}$, is in

$\mathcal{Q}_F$ if, and only if, the last symbol of the timed word is b, and it occurs at a timestamp greater than 1, while all others are a's—that is, for all $1 \leq i < n$, it holds that $\pi_F(i) = \{a\}$, and also $\pi_F(n) = \{b\}$ and $\tau_F(n) > 1$. Let $\mathcal{L}$ be the timed ω-language over alphabet $\{a, b\}$ such that $(\pi, \tau) \in \mathcal{L}$ if, and only if, there is $j \in \mathbb{N}_{>0}$ such that $\pi(j) = \{b\}$ holds, and for all $i \in \mathbb{N}_{>0}$, $i \neq j$, it holds that $\pi(i) = \{a\}$—that is, $\mathcal{L}$ is made of all and only timed words in which exactly one b appears, and all other symbols are a's. Then, $\mathcal{L}/_{\mathcal{Q}_F}$ is the timed ω-language of all timed words in which only a's appear, and they have timestamps greater than 1. For example $(\{a\}, 0.7), (\{b\}, 1.2)$ is in $\mathcal{Q}_F$, $(\{a\}, 0.7), (\{b\}, 1.2), (\{a\}, 1.5), (\{a\}, 2.5), (\{a\}, 3.5) \dots$ is in $\mathcal{L}$, and $(\{a\}, 1.5), (\{a\}, 2.5), (\{a\}, 3.5) \dots$ is in $\mathcal{L}/_{\mathcal{Q}_F}$. Notice that the timed word that belongs to the quotient starts from the first timestamp after the removed prefix, so the timestamps of the suffix are unchanged.

We can prove the following result.

Theorem 5. *Let $\mathcal{L}$ be a non-Zeno timed ω-regular language and let $\mathcal{Q}_F$ be a timed $*$-regular language. Then, the finite left quotient $\mathcal{L}/_{\mathcal{Q}_F}$ is timed ω-regular.*

Proof. To prove the claim, we show that we can build a CLTLoc formula that defines $\mathcal{L}/_{\mathcal{Q}_F}$, and we use it to obtain an initialized TA that accepts $\mathcal{L}/_{\mathcal{Q}_F}$.

Let $\mathcal{A}_\mathcal{L}$ and $\mathcal{A}_{\mathcal{Q}_F}$ be two TA accepting the two languages. The idea is to keep track of the regions in which the clocks of $\mathcal{A}_\mathcal{L}$ and $\mathcal{A}_{\mathcal{Q}_F}$ are during their execution, compose the two automata to recognize timed words whose prefix is in $\mathcal{Q}_F$, and recognize the regions that are reached by the clocks when the prefix ends. Those are the regions from which the TA recognizing the finite left quotient starts its execution.

More precisely, consider automaton $\mathcal{A}'_{\mathcal{Q}_F}$, which is the same as $\mathcal{A}_{\mathcal{Q}_F}$, except that all control states are accepting. In addition, we introduce a fresh clock, *ts*, which is never reset, hence it tracks the value of the timestamp. If we build the intersection of $\mathcal{A}_\mathcal{L}$ and $\mathcal{A}'_{\mathcal{Q}_F}$ through the usual TA (over ω-words) intersection, we obtain a new TA, which accepts all timed ω-words of $\mathcal{L}$ that have a prefix with a (not necessarily accepting) run in $\mathcal{A}_{\mathcal{Q}_F}$. Let $\mathcal{A}_{\mathcal{L}\cdot\mathcal{Q}_F}$ be this automaton. The set of control states of $\mathcal{A}_{\mathcal{L}\cdot\mathcal{Q}_F}$ is the product of the control states of $\mathcal{A}_\mathcal{L}$ and of $\mathcal{A}_{\mathcal{Q}_F}$, as the latter are the same as those of $\mathcal{A}'_{\mathcal{Q}_F}$. Then, consider a timed ω-word (π, τ) that has an accepting run in $\mathcal{A}_{\mathcal{L}\cdot\mathcal{Q}_F}$ that goes through an accepting control state $q_{\mathcal{Q}_F}$ of $\mathcal{A}_{\mathcal{Q}_F}$. The timed ω-word that corresponds to the suffix of (π, τ) starting from $q_{\mathcal{Q}_F}$ belongs to the finite left quotient $\mathcal{L}/_{\mathcal{Q}_F}$. In addition, all timed ω-words of $\mathcal{L}/_{\mathcal{Q}_F}$ have an accepting run in $\mathcal{A}_{\mathcal{L}\cdot\mathcal{Q}_F}$ that visits at least once an accepting control state $q_{\mathcal{Q}_F}$ of $\mathcal{A}_{\mathcal{Q}_F}$.

Let us now consider the region automaton [1] $\mathcal{R}(\mathcal{A}_{\mathcal{L}\cdot\mathcal{Q}_F})$ corresponding to TA $\mathcal{A}_{\mathcal{L}\cdot\mathcal{Q}_F}$. The transitions of the region automaton are labeled with the symbols of the transitions of the original TA, but the automaton embeds in its states the evolution of the clocks (including the constraints that hold on the clocks when the transitions are taken). In addition, we explicitly add the clock constraints of the target region of each transition to the guard of that transition. More precisely, each transition $\langle q, R\rangle \xrightarrow{R'_S, a} \langle q', R'\rangle$ of $\mathcal{R}(\mathcal{A}_{\mathcal{L}\cdot\mathcal{Q}_F})$ corresponds to a transition $q \xrightarrow{\gamma,a,S} q'$ of $\mathcal{A}_{\mathcal{L}\cdot\mathcal{Q}_F}$ (where R'_S is the set of clock constraints of region R' minus all constraints where a clock $x \in S$ appears), and the clock constraints of region R'_S satisfy guard γ. We build the composition of $\mathcal{A}_{\mathcal{L}\cdot\mathcal{Q}_F}$ and $\mathcal{R}(\mathcal{A}_{\mathcal{L}\cdot\mathcal{Q}_F})$, whose set of control states is the product of the control states and of the regions of $\mathcal{A}_{\mathcal{L}\cdot\mathcal{Q}_F}$, and which includes, for each pair of transitions as above, the transition $\langle q, R\rangle \xrightarrow{\gamma\wedge R'_S,a,S} \langle q', R'\rangle$. A state $\langle q, R\rangle$ of the composed automaton is accepting if, and only if, q in $\mathcal{A}_{\mathcal{L}\cdot\mathcal{Q}_F}$ is accepting. We call $\mathcal{A}_{\mathcal{L}\cdot\mathcal{Q}_F\times\mathcal{R}}$ the composed automaton. $\mathcal{A}_{\mathcal{L}\cdot\mathcal{Q}_F\times\mathcal{R}}$ recognizes the same timed ω-language as $\mathcal{A}_{\mathcal{L}\cdot\mathcal{Q}_F}$, but it also keeps track of the clock regions reached by the automaton in the accepting states.

By Proposition 5, we translate automaton $\mathcal{A}_{\mathcal{L}\cdot\mathcal{Q}_F\times\mathcal{R}}$ into a language equivalent CLTLoc formula $\phi_{\mathcal{A}_{\mathcal{L}\cdot\mathcal{Q}_F\times\mathcal{R}}}$. The quotient language $\mathcal{L}/_{\mathcal{Q}_F}$ corresponds to the suffixes of timed words (π, σ) of $\mathcal{A}_{\mathcal{L}\cdot\mathcal{Q}_F\times\mathcal{R}}$ that start from states in which the component $q_{\mathcal{Q}_F}$ from the state space of $\mathcal{A}_{\mathcal{Q}_F}$ is accepting in the latter automaton. To define these suffixes, we define a new formula $\phi_{\mathcal{L}/_{\mathcal{Q}_F}}$, by modifying $\phi_{\mathcal{A}_{\mathcal{L}\cdot\mathcal{Q}_F\times\mathcal{R}}}$ as follows: we replace the subformulae that capture the first transition of automaton $\mathcal{A}_{\mathcal{L}\cdot\mathcal{Q}_F\times\mathcal{R}}$ [12, Formula (3) of Section 4.1] with new ones, each representing a transition originating from a state that is reachable from the initial one and in which the $q_{\mathcal{Q}_F}$ component is accepting. Also, we remove the constraints (2) and (3) that link clocks x_i with clock Now, and we add a constraint $\mathit{Now} = \mathit{ts}_1$ establishing the equality of clock Now of formula $\phi_{\mathcal{A}_{\mathcal{L}\cdot\mathcal{Q}_F\times\mathcal{R}}}$ with clock ts_1 representing the value of clock ts introduced above—that is, we establish that the clock tracking the timestamp in formula $\phi_{\mathcal{L}/_{\mathcal{Q}_F}}$ has the same value of the one tracking the timestamp in the original automaton.

It is standard to show that the timed language of formula $\phi_{\mathcal{L}/_{\mathcal{Q}_F}}$ is indeed the quotient $\mathcal{L}/_{\mathcal{Q}_F}$. In fact, each a timed word (π', τ') of $\mathcal{L}/_{\mathcal{Q}_F}$ is such that there is an accepting run $(q_{i_0}, v_0) \xrightarrow[\tau(1)]{\pi(1)} (q_{i_1}, v_1), \dots, (q_{i_{n-1}}, v_{n-1}) \xrightarrow[\tau(n)]{\pi(n)} (q_{i_n}, v_n) \xrightarrow[\tau'(1)]{\pi'(1)} (q_{i_{n+1}}, v_{n+1}) \xrightarrow[\tau'(2)]{\pi'(2)} (q_{i_{n+2}}, v_{n+2}), \dots$ of automaton $\mathcal{A}_{\mathcal{L}\cdot\mathcal{Q}_F\times\mathcal{R}}$ such that the $\mathcal{Q}_F$ component of state q_{i_n} is accepting. It can be shown as in the proof of Theorem 2 of [12] that from the suffix $(q_{i_n}, v_n) \xrightarrow[\tau'(1)]{\pi'(1)} (q_{i_{n+1}}, v_{n+1}) \xrightarrow[\tau'(2)]{\pi'(2)} (q_{i_{n+2}}, v_{n+2}), \dots$ of the run one can build a CLTLoc model (π', σ') that satisfies formula $\phi_{\mathcal{L}/_{\mathcal{Q}_F}}$ and such that $(\pi', \tau') = [(\pi', \sigma')]$ holds.

Dually, consider a CLTLoc model (π', σ') of formula $\phi_{\mathcal{L}/_{\mathcal{Q}_F}}$ and the timed word (π', τ') such that $(\pi', \tau') = [(\pi', \sigma')]$ holds. As in the proof of Theorem 2 of [12], one can build an accepting run $(q_{i_n}, v_n) \xrightarrow[\tau'(1)]{\pi'(1)} (q_{i_{n+1}}, v_{n+1}) \xrightarrow[\tau'(2)]{\pi'(2)} (q_{i_{n+2}}, v_{n+2}), \dots$ of automaton $\mathcal{A}_{\mathcal{L}\cdot\mathcal{Q}_F\times\mathcal{R}}$ starting from a configuration (q_{i_n}, v_n) in which the $\mathcal{Q}_F$ component of q_{i_n} is accepting. By construction, (q_{i_n}, v_n) is reachable from the initial configuration (q_{i_0}, v_0) of $\mathcal{A}_{\mathcal{L}\cdot\mathcal{Q}_F\times\mathcal{R}}$. Then, thanks to the fact that q_{i_n} includes the information about the region of clock ts, one can choose a delay δ_n, a configuration $(q_{i_{n-1}}, v_{n-1})$, and a $\pi(n) \in \wp(AP)$ such that $(q_{i_{n-1}}, v_{n-1}) \xrightarrow[\tau'(1)-\delta_n]{\pi(n)} (q_{i_n}, v_n) \xrightarrow[\tau'(1)]{\pi'(1)} (q_{i_{n-1}}, v_{n-1}), \dots$ is an accepting run starting from $(q_{i_{n-1}}, v_{n-1})$, and so on, until an accepting run $(q_{i_0}, v_0) \xrightarrow[\tau(1)]{\pi(1)} (q_{i_1}, v_1), \dots, (q_{i_{n-1}}, v_{n-1}) \xrightarrow[\tau(n)]{\pi(n)} (q_{i_n}, v_n) \xrightarrow[\tau'(1)]{\pi'(1)} (q_{i_{n+1}}, v_{n+1}), \dots$ of $\mathcal{A}_{\mathcal{L}\cdot\mathcal{Q}_F \times \mathcal{R}}$ is obtained.

By Theorem 3, there exists a CLTLoc formula $\phi'_{\mathcal{L}/_{\mathcal{Q}_F}}$ whose initialized timed language is $\mathcal{L}/_{\mathcal{Q}_F}$ and by Theorem 1 we can build an initialized TA $\mathcal{A}'_{\mathcal{L}/_{\mathcal{Q}_F}}$ that accepts language $\mathcal{L}/_{\mathcal{Q}_F}$, hence $\mathcal{L}/_{\mathcal{Q}_F}$ is timed ω-regular. □

Acknowledgments. We thank the anonymous reviewers for the useful suggestions which helped us improve the quality of the paper.

References

[1] R. Alur, D. L. Dill, A theory of timed automata, Theoretical Computer Science 126 (2) (1994) 183–235.

[2] M. M. Bersani, M. Rossi, P. San Pietro, A tool for deciding the satisfiability of continuous-time metric temporal logic, Acta Informatica 53 (2) (2015) 171–206.

[3] S. Demri, D. D'Souza, An automata-theoretic approach to constraint LTL, Information and Computation 205 (3) (2007) 380–415.

[4] M. M. Bersani, A. Frigeri, A. Morzenti, M. Pradella, M. Rossi, P. San Pietro, Constraint LTL satisfiability checking without automata, Journal of Applied Logic 12 (4) (2014) 522 – 557.

[5] Microsoft Research, Z3: An efficient SMT solver, github.com/Z3Prover/z3 (2015).

[6] M. M. Bersani, M. Rossi, P. San Pietro, An SMT-based approach to satisfiability checking of MITL, Information and Computation 245 (2015) 72–97.

[7] M. M. Bersani, M. Rossi, P. San Pietro, A logical characterization of timed (non-)regular languages, in: Mathematical Foundations of Computer Science, Vol. 8634 of Lecture Notes in Computer Science, 2014, pp. 75–86.

[8] B. Bérard, V. Diekert, P. Gastin, A. Petit, Characterization of the expressive power of silent transitions in timed automata, Fundam. Inf. 36 (2,3) (1998) 145–182.

[9] P. Bouyer, C. Dufourd, E. Fleury, A. Petit, Updatable timed automata, Theoretical Computer Science 321 (2) (2004) 291–345.

[10] P. Wolper, Temporal logic can be more expressive, in: Proceedings of the 22nd Annual Symposium on Foundations of Computer Science, 1981, pp. 340–348.

[11] P. V. Suman, P. K. Pandya, An introduction to timed automata, World Scientific, 2012, Ch. 4, pp. 111–146.

[12] M. M. Bersani, M. Rossi, P. San Pietro, A logical characterization of timed regular languages, Theoretical Computer Science 658 (Part A) (2017) 46 – 59. doi:https://doi.org/10.1016/j.tcs.2016.07.020.