Distributed Event Clock Automata

Ortiz Vega, James; Legay, Axel; Schobbens, Pierre-Yves

Published in:
CIAA 2011

DOI:
10.1007/978-3-642-22256-6_23

Publication date:
2011

PReCISE – FUNDP
University of Namur
Rue Grandgagnage, 21
B-5000 Namur
Belgium

PAPER March, 2011

AUTHORS	J. Ortiz, A. Legay, P-Y. Schobbens
EMAIL	jor@info.fundp.ac.be, alegay@irisia.fr, pierre-yves.schobbens@fundp.ac.be
VENUE	CIAA'11
STATUS	Extended version of the paper Submitted to CIAA'11
PROJECT	
FUNDING	

Distributed Event Clock Automata

Distributed Event Clock Automata
Extended Abstract

James Ortiz[1], Axel Legay[2 3], and Pierre-Yves Schobbens[1]

[1] Computer Science Faculty, University of Namur
[2] INRIA/IRISA, Rennes
[3] Institut Montefiore, University of Liège
james.ortizvega@fundp.ac.be
alegay@irisa.fr
pierre-yves.schobbens@fundp.ac.be

Abstract. In distributed real-time systems, we cannot assume that clocks are perfectly synchronized. To model them, we use independent clocks and define their multi-timed semantics. The universal timed language, and the timed language inclusion of icTA are shown undecidable. Thus, we propose Recursive Distributed Event Clock Automata (DECA). DECA are closed under all boolean operations and their timed language inclusion problem is decidable (more precisely PSPACE-complete), allowing stepwise refinement. We also propose Recursive Distributed Event Clock Temporal Logic (DECTL), a real-time logic with independent time evolutions. This logic can be model-checked by translating a DECTL formula into a DECA automaton.

1 Introduction

Real-Time Distributed Systems (RTDS) take an increasingly important role in our society, including in aircrafts and spacecrafts, satellite telecommunication networks or positioning systems. Distributed Systems consist of computer systems at different locations, that communicate through a network to achieve their function. Real-Time Systems have to obey strict requirements about the time of their actions. To ensure these, they rely on clocks. When systems are widely distributed, we cannot assume that their clocks are perfectly synchronized.

One of the most successful techniques for modeling real-time systems are Timed Automata (TA) [2]. A timed automaton is a finite automaton augmented with real-valued clocks. Constraints on these clocks are used to restrict the behaviors of the automaton. The model of TA assumes perfect clocks: all clocks have infinite precision and are perfectly synchronized.

This causes TA to have an undecidable language inclusion problem [2]. The situation contrasts strongly with the one of automata without real time, where the problems of complementation, language inclusion, emptiness, union and intersection are decidable, as well as the satisfiability and validity of propositional linear temporal logic (LTL). These properties are the basis of the success of model-checking. When all these problems are decidable, we call the formalism (automata or logic) fully decidable. These negative results spurred a quest for expressive but still fully decidable formalisms.

To restore decidability, [4] proposed to restrict the behavior of clocks. The key idea is that the problematic clocks of TA are reset by non-deterministic transitions. In contrast, an event clock (EC) x_p is reset when a given atomic proposition p occurs. The event clock values are deterministic and thus Event Clock Automata ECA are determinizable, making language inclusion decidable and thus enabling refinement based development. Event clocks can also be introduced in temporal logic [15]. An event clock constraint is naturally translated into a proposition $\lhd_I p$, that means "the last time that a p occurred was d time units ago, where d lies in I". However, the expressiveness of ECA is rather weak. Furthermore, this logic violates the substitution principle: Any proposition should be replaceable by a formula.

Therefore [9] introduced the notion of "recursive" event. In a recursive event model, the reset of a clock is decided by a lower-level automaton (or formula). This automaton cannot read the clock that it is resetting. Clock resets are thus still deterministic, but the concept of "event" is now much more expressive. $\rhd_I$ and $\lhd_I$ are modalities that can contain any subformulas, and can be nested. The temporal logic of recursive event clocks (variously called SCL [15] or EventClockTL [9]) has the same expressiveness as Metric Interval Temporal Logic MITL [3] (a decidable fragment of MTL where punctual constraints are forbidden) in the interval semantics. First-and second-order monadic logics with matching expressiveness have been provided [9], yielding a natural, robust, fully decidable level of real-time expressiveness. In this paper, we remove the assumption of perfect clock synchronization. Here, inspired by [6, 10, 1], we study the worst case: the clocks can advance totally independently if they are in different processes. While [1] only studied untimed languages, namely the universal and existential languages, here we define and study the corresponding timed languages. [13, 8] studied the opposite case, where the difference between clocks (drift) is infinitesimally small.

Our first contribution is to extend the Recursive Event Clock Automata (RECA) with such distributed (a.k.a independent) clocks, yielding the Distributed Recursive Event Clock Automata (DECA). We will show that DECA are determinizable, thus closed under complementation, and thus that their language inclusion problem is decidable (more exactly, PSPACE-complete). We also show the decidability and regularity of their universal languages.

Our second contribution is to extend EventClockTL with distributed clocks. This gives us the (Recursive) Distributed Event Clock Temporal Logic (DECTL), which we show to be PSPACE-complete.

Structure of the paper. The rest of the paper is organized as follows. Sections 2 and 3 recall preliminary notions. Section 4 extends the semantics to multiple timed languages. Section 5 defines DECA and studies their properties. Section 6 examines real-time temporal logics: it recalls EventClockTL [15], then introduces and studies DECTL.

2 Preliminaries

We briefly recall the various models of time that are used in the literature [5]. We present our results in the interval semantics, that is the richest and most natural (but also most difficult) model. We also recall clocks and their constraints.

2.1 Models of time

Models of time can be linear, considering a single future, or branching, considering several alternative futures. We only consider linear time in this paper. Our goal here is to model real-time systems, and thus we use the real numbers as our model of time. This avoids a premature commitment to a specific discretization of time. In this paper, we use the *interval semantics*, where the state of the model is known at any point in time, as opposed to *point semantics*, where it is known only at events.

Let $\mathbb{P}$ be a finite set of *propositional symbols*. A *letter* is an element of a finite set Σ. In this paper, we choose to define a letter as propositional valuation over $\mathbb{P}$, so we pose $\Sigma = 2^{\mathbb{P}}$. Let $\mathbb{N}$ be the set of natural numbers, $\mathbb{R}$ denote the set of real numbers, $\mathbb{R}_{\geq 0}$ the set of non-negative real numbers. We use the interval semantics. We denote by $\mathcal{I}_{\mathbb{R}_{\geq 0}}$ the set of real intervals whose bounds are in $\mathbb{R}_{\geq 0}$. An interval $I \in \mathcal{I}_{\mathbb{R}_{\geq 0}}$ is a convex subset of $\mathbb{R}_{\geq 0}$. Two intervals I and I' are said to be *adjacent* when they are disjoint: $I \cap I' = \emptyset$ and $I \cup I'$ is an interval. An (alternating) interval sequence is a sequence $I = I_0 I_1 \cdots$ of non-empty intervals of $\mathcal{I}_{\mathbb{R}_{\geq 0}}$ where : (i) $I_0 = \{0\}$; (ii) singular and open intervals alternate; (iii) successive intervals I_j and I_{j+1} are *adjacent* for all $j \geq 0$, (iv) if infinite, the sequence of intervals is *progressive*, i .e., for every $t \in \mathbb{R}_{\geq 0}$, there exists $j \in \mathbb{N}$ such that $t \in I_j$. A interval state sequence ρ can equivalently be seen as an sequence of elements in $2^{\mathbb{P}} \times \mathcal{I}_{\mathbb{R}+}$. It can also be seen as a *signal*, i.e. a function from $\mathbb{R}^+$ to states: Let $\rho = (\sigma, I)$ be a interval state sequence and given $t \in \mathbb{R}^+$, let $i \in \mathbb{N}$ be the interval such that $t \in I_i$. We define $\rho(t)$ as the state σ_i. A signal derived from an ISS will always have finite variability. Below, our automata will consider two ISS that define the same signal as equivalent, even if the intervals might be split differently. Our automata assume finite variability. In Section 4, we will extend this model to several time scales.

2.2 Clocks

A clock is a variable that increases with time. Thus, the value of a clock is the time elapsed since its last reset. When we use continuous time, there is not always a “last” reset, e.g. when the reset holds in an open interval. For this case, we will use non-standard clock values of the form υ^+, intuitively meaning that the clock was reset just υ units before. The set of non-standard real numbers, noted $\mathbb{R}^+_{\geq 0}$, is the set of $\{\upsilon, \upsilon^+ \mid \upsilon \in \mathbb{R}_{\geq 0}\}$, ordered by $<_{ns}$ as following: $\upsilon_1 <_{ns} \upsilon_2^+$ iff $\upsilon_1 \leq \upsilon_2$. The addition is commutative, and $\upsilon_1^+ + \upsilon_2 = (\upsilon_1 + \upsilon_2)^+$. $\mathbb{R}^+_{\perp}$ is $\mathbb{R}^+_{\geq 0}$ plus a special value $\perp$ for uninitialized clocks. $\perp$ is not comparable to other values, and is absorbing for addition.

Let *X* be a finite set of clock names. A clock valuation over *X* is a mapping $\nu : X \to \mathbb{R}^+_{\perp}$. For a valuation ν and a time value $t \in \mathbb{R}_{\geq 0}$, let $\nu + t$ denote the valuation such that $(\nu + t)(x) = \nu(x) + t$, for each clock $x \in X$.

The set of constraints over *X*, denoted $\Phi(X)$, is defined by the following grammar, where ϕ ranges over $\Phi(X)$, $x \in X$, $c \in \mathbb{N}$, and $\sim \in \{<, \leq, =, >, \geq\}$:

$$\phi ::= true \mid x \sim c \mid \phi_1 \wedge \phi_2$$

We write $\nu \models \phi$ when the valuation ν satisfies the constraint ϕ. When *x* has the value $\perp$, we evaluate $x \sim c$ to false.

3 Automata Background

Based on time and clocks, several variants of timed automata have been proposed after the seminal Timed Automata TA [2]. Below, we review briefly icTA [1] and RECA [9], that are the basis of our DECA.

3.1 Timed Automata

A Timed Automaton (TA) [2] is a finite state automaton augmented with clocks: real variables that can be reset to 0, and otherwise increase at a uniform rate. Time is thus global, and clocks are perfectly precise and synchronized. Our definition of TA has the following minor peculiarities:

1. We use a continuous signal semantics throughout the paper, i.e. the state is a function of time.
2. In particular, we do not allow to be in two locations, or to make two transitions, at the same time. Time strictly increases along an ISS, as in [2].
3. We observe states, rather than actions, to link with temporal logics.
4. We implicitly allow ϵ-transitions [7], that were absent from [2].

Definition 1. *A Timed Automaton is a tuple $\mathcal{A} = (\Sigma, X, S, s_0, \rightarrow_{ta}, Inv, \gamma, F)$, such that:*

(i) *Σ, a finite alphabet.*
(ii) *X, a finite set of positive real variables called clocks.*
(iii) *S, a finite set of locations.*
(iv) *$s_0 \in S$, the initial location.*
(v) *$\rightarrow_{ta} \subseteq S \times \Phi(X) \times 2^X \times S$, a finite set of transitions.*
(vi) *$Inv : S \rightarrow \Phi(X)$ gives the invariant.*
(vii) *$\gamma : (S \cup \rightarrow_{ta}) \rightarrow \Sigma$, a labelling of locations and transitions.*
(viii) *F, an acceptance condition. For instance, for finite acceptance, we have $F \subseteq S$, a set of final locations. We also use Büchi (where $F \subseteq S$) or parity conditions (where $F : S \rightarrow \mathbb{N}$).*

TA are neither determinizable nor complementable. Their emptiness problem can be solved using the region construction, but their inclusion problem is undecidable [2].

3.2 Timed Automata with Independent Clocks

Distributed Timed Automata (DTA) [10, 1] consist of a number of local timed automata. Each automaton owns clocks. The clocks of a process evolve synchronously, but independently of the clocks of the other processes. The clocks belonging to one process can be read by another process, but a clock can only be reset by its owner process.

Definition 2. *A DTA is a tuple $\mathcal{D} = (Proc, \mathcal{A}, \pi)$, such that :*

(i) *Proc is a nonempty, finite set of process labels.*
(ii) *$\mathcal{A}$ is an indexed set of Timed Automata, $\mathcal{A} = (\mathcal{A}_q)_{q \in Proc}$.*
(iii) *$\pi : \bigcup_{q \in Proc} X_q \rightarrow Proc$ maps each clock to its owner process.*

where $\mathcal{A}_q$ can only reset owner clocks and $\mathcal{A}_q = (\Sigma_q, X_q, S_q, S_q^0, \to_q, I_q, \gamma_q, F_q)$ are Timed Automata.

Note that a process can read a clock of another process, since the x^q need not be disjoint. In [1], DTA are not much studied. Instead, their product is first computed, giving rise to the class of Timed Automata with independent clocks (icTA). icTA assume a *signature*. A *signature* is a pair $(\mathit{Proc}, \mathbb{P})$, where *Proc* is a nonempty finite set of process labels, and $\mathbb{P}$ is a finite set of propositional symbols, from which we define $\Sigma = 2^{\mathbb{P}}$. For a tuple t that is indexed by *Proc*, t_p refers to the projection of t onto $p \in \mathit{Proc}$.

Definition 3. *An* icTA *is a pair $(\mathcal{A}, \pi)$, where $\mathcal{A}$ is a TA and $\pi : X \to \mathit{Proc}$ maps each clock to a process.*

The semantics of DTA, icTA, and our DECA, depends on the local evolutions of time. This is modelled by a tuple $\tau = (\tau_q)_{q \in \mathit{Proc}}$ of local time functions. Each local time function τ_q maps the reference time to the local time of process q, i.e, $\tau_q : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$. The functions τ_q must be continuous, strictly increasing, and divergent, and satisfy $\tau_q(0) = 0$. The set of all these tuples τ is denoted by $Rates$. We can also consider τ as a mapping to a tuple of local times: $\tau : \mathbb{R}_{\geq 0} \to (\mathbb{R}_{\geq 0})^{\mathit{Proc}}$. Note that the reference time is arbitrary.

Definition 4. *Given a clock valuation $\nu : X \to \mathbb{R}_{\geq 0}$ and a delay tuple $t \in \mathbb{R}^{Proc}$, the valuation $\nu + t$ is defined by $(\nu + t)(x) = \nu(x) + t_{\pi(x)}$ for all $x \in X$.*

A run of an icTA $\mathcal{A}$ for τ is an alternating sequence of states and transitions $q_0 \xrightarrow{\zeta_0} q_1 \xrightarrow{d_1} q_1 + d_1 \xrightarrow{\zeta_1} \ldots$, where $i \geq 0$, d_i is a non-decreasing sequence of values from $\mathbb{R}_{\geq 0}$, states are triples of a location, a clock valuation, and lastly the reference time: $q \in Q = \{(s, \nu, t) \in S \times \mathbb{R}_{\geq 0}^X \times \mathbb{R}_{\geq 0} \mid \nu \models \mathit{Inv}(s)\}$. A run should furthermore satisfy:

1. the starting state is $q_0 = (s_0, \nu_0, 0)$, where ν_0 assigns 0 to all the clocks,
2. the transitions must alternate between two types:
   - Delay transition, i.e. spending time in a location: $q_i \xrightarrow{d} q_i + d$, where $q_i = (s_i, \nu_i, t_i)$, and $q_i + d = (s_i, \nu_i + (\tau(t_i + d) - \tau(t_i)), t_i + d)$, if the invariant is continuously true in local time: $\forall t \in]t_i, t_i + d[\colon \nu_i + (\tau(t) - \tau(t_i)) \models \mathit{Inv}(s_i)$.
   - Discrete transition: following a transition $\zeta_i = (s_{i-1}, \phi, Y, s_i) \in \to_{\mathsf{icTA}}$ when the clock constraint ϕ is satisfied. The clocks in Y are then reset. This transition is instantaneous. $(s_{i-1}, \nu_{i-1}, t_{i-1}) \xrightarrow{\zeta_i} (s_i, \nu_i, t_i)$, such that $\nu_{i-1} \models \phi$, $\nu_i = \nu_{i-1}[Y \to 0], t_{i-1} = t_i$.
3. The acceptance condition is verified, e.g. for a finite automaton, $s_n \in F$.

Given a run $\rho = (s_0, \nu_0, t_0) \xrightarrow{\zeta_0} (s_1, \nu_1, t_1) \ldots$ of $\mathcal{A}$ for τ, we define its ISS as $(\gamma(\zeta_0), \{t_0\}), (\gamma(s_1),]t_1, t_2[), \ldots$. The language $\mathcal{L}(\mathcal{A}, \tau)$ is defined as the set of ISS of accepting runs of $\mathcal{B}$ for τ, closed under $\equiv$, the equivalence generated by merging adjacent intervals with the same labelling. The existential language of $\mathcal{A}$ is denoted by $\mathcal{L}_\exists(\mathcal{A}) = \bigcup_{\tau \in \mathit{Rates}} \mathcal{L}(\mathcal{A}, \tau)$ and the universal semantics of $\mathcal{A}$ is denoted by $\mathcal{L}_\forall(\mathcal{A}) = \bigcap_{\tau \in \mathit{Rates}} \mathcal{L}(\mathcal{A}, \tau)$.

If $|Proc|$ = 1, then an icTA $\mathcal{A}$ actually reduces to an ordinary timed automaton and we have $\mathcal{L}_\forall(\mathcal{A}) = \mathcal{L}(\mathcal{A}, \tau)$ for any $\tau \in Rates$. Moreover, if $|Proc| > 1$ and $\tau \in Rates$ exhibits, for all $p \in Proc$, the same local time evolution, then $\mathcal{L}(\mathcal{A}, \tau)$ is the language of $\mathcal{A}$ considered as an ordinary timed automaton.

3.3 Recursive Event Clocks Automata

Recursive Event Clock Automata (RECA) [14, 9] extend ECA [5]. "Recursive" refers to the fact that the resets of an event clock $x_\mathcal{B}$ are controlled by a lower-level automaton $\mathcal{B}$: When $\mathcal{B}$ passes in a monitored location, it resets $x_\mathcal{B}$. We present here a version of RECA for continuous time, where transitions have all properties of locations.

Definition 5. *A RECA $\mathcal{A}$ of level $l \in \mathbb{N}$ is a tuple $\mathcal{A} = (\Sigma, C, S, s_0, \rightarrow_{reca}, M, \gamma, \delta, F)$, such that:*

(i) Σ is a finite alphabet.
(ii) C is a finite set of clocks, of the form $x_\mathcal{B}$ or $y_\mathcal{B}$, with $\mathcal{B}$ a lower-level RECA.
(iii) S is a finite set of locations.
(iv) $s_0 \in S$ is the initial location.
(v) $\rightarrow_{reca} \subseteq S \times S$ are the transitions.
(vi) $M \subseteq (S \cup \rightarrow_{reca})$ is the set of monitored locations or transitions: when the automaton visit such a location, it resets the associated clock.
(vii) $\gamma : (S \cup \rightarrow_{reca}) \rightarrow \Sigma$ is a labelling function which labels each location or transition with a symbol.
(viii) $\delta : (S \cup \rightarrow_{reca}) \rightarrow \Phi(C)$ gives the guard or invariant clock constraints.
(ix) F is an acceptance condition, e.g. a set of final locations, or of Büchi accepting locations.

Throughout the paper, we assume this uniform naming convention. RECA can be determinized and thus complemented: They are *fully decidable* [9].

4 Multi-Timed Languages

Surprisingly, Akshay et al. [1] only consider untimed languages for their timed automata. We are interested in timed languages, but we have several times here. We thus define a *multi-ISS* as a sequence of letters and (local time) interval sequences, one for each local time in $P \subseteq Proc$: $\mu = (\sigma, (I_q)_{q\in P})$. Let τ_P be a rate defined on P. Given an interval I, we can obtain the corresponding multi-interval $\tau_P(I)$ by applying τ_P to its bounds, for instance $\tau_P(]t_i, t_{i+1}[) =]\tau_P(t_i), \tau_P(t_{i+1})[$. This extends naturally to interval sequences. Given an ISS $\rho = (\sigma, I)$ (expressed in the reference time), its *multi-ISS* $\tau_P(\rho)$ is $(\sigma, \tau_P(I))$. Let $\mathcal{B}$ be an icTA, the language $\mathcal{L}(\mathcal{B}, \tau)$, is defined as the set of ISS of accepting runs of $\mathcal{B}$ for τ, closed under the equivalence generated by merging adjacent intervals with the same propositional labelling. However, ISS is not significant here, since it is expressed in the (arbitrary) reference time. The multi-timed language $\mathcal{L}(\mathcal{B}, \tau, P)$ is the set of all accepted *multi-ISS*: $\mathcal{L}(\mathcal{B}, \tau, P) = \tau_P(\mathcal{L}(\mathcal{B}, \tau))$. If we select a subset Q of P, we can project a *multi-ISS*

$\rho = (\sigma, (I_q)_{q \in P})$ to this subset Q, noted $\rho|_Q = (\sigma, (I_q)_{q \in Q})$. This projection extends naturally to languages. In particular, the UNTIME operation [2] is the case with $P = \emptyset$. Note that $\mathcal{L}(\mathcal{B}, \tau, P)|_Q = \mathcal{L}(\mathcal{B}, \tau, Q)$. When there is only one process *Proc* $= \{q\}$, the timed language observed by q, $\mathcal{L}((\mathcal{A}, \pi), \tau, \{q\})$, does not depend on τ, and for an icTA, it is the usual timed language $\mathcal{L}(\mathcal{A})$ of its TA. When τ is the identity, we also obtain the usual timed language.

For expressing real-time requirements, we have to choose the process(es) that will measure the time. When we want to avoid some forbidden timed behaviours, we naturally consult the existential timed semantics: we consider local times as non-deterministic. If we want a given timed behaviour to be possible whatever the evolution of local times, we check that it belongs to the universal semantics. Thus we define, for an automaton $\mathcal{B}$ and a subset of its processes P:

- the existential timed language observed by P : $\mathcal{L}_\exists(\mathcal{B}, P) = \bigcup_{\tau \in Rates} \mathcal{L}(\mathcal{B}, \tau, P)$
- the universal timed language observed by P : $\mathcal{L}_\forall(\mathcal{B}, P) = \bigcap_{\tau \in Rates} \mathcal{L}(\mathcal{B}, \tau, P)$

The untimed languages defined in [1] are the special cases with an empty set of observers. More generally:

Theorem 1. $\mathcal{L}_\exists(\mathcal{B}, P)|_Q = \mathcal{L}_\exists(\mathcal{B}, Q)$ *where* $Q \subseteq P$.

Proof.

$$\begin{aligned}
\mathcal{L}_\exists(\mathcal{B}, P)|_Q &= (\bigcup_{\tau \in Rates} \mathcal{L}(\mathcal{B}, \tau, P))|_Q \\
&= (\bigcup_{\tau \in Rates} \tau_P(\mathcal{L}(\mathcal{B}, \tau)))|_Q \\
&= (\bigcup_{\tau \in Rates} \{\tau_P(\sigma, I) \mid (\sigma, I) \in \mathcal{L}(\mathcal{B}, \tau))\})|_Q \\
&= (\{(\sigma, \tau_P(I)) \mid (\sigma, I) \in \mathcal{L}(\mathcal{B}, \tau)) \wedge \tau \in Rates\})|_Q \\
&= \{(\sigma, \tau_Q(I)) \mid (\sigma, I) \in \mathcal{L}(\mathcal{B}, \tau)) \wedge \tau \in Rates\} \\
&= \mathcal{L}_\exists(\mathcal{B}, Q)
\end{aligned}$$

but for universal languages, we need to prove the following theorems :

Theorem 2. *For any icTA* $\mathcal{B}$*, if* $|P| \geq 2$ *then* $\mathcal{L}_\forall(\mathcal{B}, P) = \emptyset$ *where* $P \subseteq$ *Proc.*

Proof. We prove this by contradiction. Consider the set of processes $P = \{p_1, p_2, \ldots, p_n\}$ with $n \geq 2$, the universal timed language $\mathcal{L}_\forall(\mathcal{B}, P)$ where $P \subseteq$ *Proc* and an ISS ρ, then we want to prove that for all $\tau \in$ *Rates*, $\tau_P(\rho) \notin \mathcal{L}(\mathcal{B}, \tau, P)$. Assume to the contrary that for some $P = \{p_1, p_2, \ldots, p_n\}$ with $n \geq 2$, the universal timed language $\mathcal{L}_\forall(\mathcal{B}, P)$ = $\bigcap_{\tau \in Rates} \mathcal{L}(\mathcal{B}, \tau, P)$ = $\bigcap_{\tau \in Rates} \tau_P(\mathcal{L}(\mathcal{B}, \tau)) \neq \emptyset$, then there is ρ, for all $\tau \in$ *Rates*, $\tau_P(\rho) \in \mathcal{L}(\mathcal{B}, \tau, P)$. Consider the ISS ρ = $(\gamma(\zeta_0), \{0\})$, $(\gamma(s_1),]0, t_1[)$, ..., the tuple of local functions τ = $(\tau_{p_1}, \tau_{p_2}, \ldots, \tau_{p_n})$ and since $\neg\forall\tau$, $\tau_P(\rho) \in \mathcal{L}(\mathcal{B}, \tau, P)$, then we have that $\neg\forall\tau$, $\exists\theta$ that is $(s_0, \nu_0, t_0) \xrightarrow{\zeta_0} (s_1, \nu_1, t_1) \ldots$ a run of $\mathcal{B}$, where $\tau_P(\rho)$

= $\tau_P(\mathsf{ISS}(\theta))$. Given $\mathcal{B}$, $\tau \in$ *Rates* and ρ = $(\gamma(\zeta_0), \{0\})$, $(\gamma(s_1),]0, t_1[)$, ..., we have that $\tau_{p_1}(t_1)$ = t_{p_1} with $\kappa_1 \leq t_{p_1} \leq \kappa_2$, $\tau_{p_2}(t_1)$ = t_{p_2} with $\kappa_1 \leq t_{p_2} \leq \kappa_2$, ..., $\tau_{p_n}(t_1)$ = t_{p_n} with $\kappa_1 \leq t_{p_n} \leq \kappa_2$, where we assume due to the construction that κ_1, κ_2 are both integer and for each $1 \leq i \leq n$, $t_{p_i} \in [\kappa_1, \kappa_2]$. Let $\tau_{p_1}(t)$ = $t_{p_1} \cdot t$, $\tau_{p_2}(t)$ = $2 \cdot t_{p_2} \cdot t$, ..., $\tau_{p_n}(t)$ = $n \cdot t_{p_n} \cdot t$ implies $t_{p_1} \cdot t_1 = t_{p_1}$ then $t_1 = 1$, $2 \cdot t_{p_2} \cdot t_1 = t_{p_2}$ then $t_1 = 1/2$, ..., $n \cdot t_{p_n} \cdot t_1 = t_{p_n}$ then $t_1 = 1/n$ which is impossible and contradicts that $\mathcal{L}_\forall(\mathcal{B}, P) \neq \emptyset$.

Theorem 3. *For some* icTA $\mathcal{B}$ *with final states* $F \neq \emptyset$*, if* $|P| < 2$ *then* $\mathcal{L}_\forall(\mathcal{B}, P) \neq \emptyset$ *where* $P \subseteq$ *Proc.*

Proof. We prove this by contradiction. Consider the set of a single process P = $\{p\}$ and the universal timed language $\mathcal{L}_\forall(\mathcal{B}, P)$ where $P \subseteq$ *Proc* and an ISS ρ, then we want to prove that for all $\tau \in$ *Rates*, $\tau_P(\rho) \in \mathcal{L}(\mathcal{B}, \tau, P)$. Assume to the contrary that for some P = $\{p\}$, the universal timed language $\mathcal{L}_\forall(\mathcal{B}, P)$ = $\bigcap_{\tau \in Rates} \mathcal{L}(\mathcal{B}, \tau, P)$ = $\bigcap_{\tau \in Rates} \tau_P(\mathcal{L}(\mathcal{B}, \tau)) = \emptyset$, then there is ρ, for all $\tau \in$ *Rates*, $\tau_P(\rho) \notin \mathcal{L}(\mathcal{B}, \tau, P)$. Consider the ISS ρ = $(\gamma(\zeta_0), \{0\})$, $(\gamma(s_1),]0, t_1[)$, ..., the tuple of local functions τ = (τ_p) and since $\neg\forall\tau$, $\tau_P(\rho) \in \mathcal{L}(\mathcal{B}, \tau, P)$, then we have that $\neg\forall\tau$, $\exists\theta$ that is $(s_0, \nu_0, t_0) \xrightarrow{\zeta_0} (s_1, \nu_1, t_1) \dots$ a run of $\mathcal{B}$, where $\tau_P(\rho)$ = $\tau_P(\mathsf{ISS}(\theta))$. Given $\mathcal{B}$, $\tau \in$ *Rates* and ρ = $(\gamma(\zeta_0), \{0\})$, $(\gamma(s_1),]0, t_1[)$, ..., we have that $\tau_{p_1}(t_1)$ = t_p with $\kappa_1 \leq t_{p_1} \leq \kappa_2$, where we assume due to the construction that κ_1, κ_2 are both integer and $t_p \in [\kappa_1, \kappa_2]$. Let $\tau_p(t)$ = $t_p \cdot t$, implies $t_p \cdot t_1 = t_p$ then $t_1 = 1$, which contradicts that $\mathcal{L}_\forall(\mathcal{B}, P) = \emptyset$.

Theorem 4. $\mathcal{L}_\forall(\mathcal{B}, P)|_Q \subseteq \mathcal{L}_\forall(\mathcal{B}, Q)$ *where* $Q \subseteq P$.

Proof. Consider the set of processes P = $\{p_1, p_2, \dots, p_n\}$ with $n \geq 2$, the universal timed language $\mathcal{L}_\forall(\mathcal{B}, P)|_Q$ where $Q \subseteq P \subseteq$ *Proc* and $2 \leq |Q| \leq n$, then we want to prove that $\mathcal{L}_\forall(\mathcal{B}, P)|_Q \subseteq \mathcal{L}_\forall(\mathcal{B}, Q)$.

$$\begin{aligned} \mathcal{L}_\forall(\mathcal{B}, P)|_Q &= \Big(\bigcap_{\tau \in Rates} \mathcal{L}(\mathcal{B}, \tau, P)\Big)|_Q \\ &= \emptyset \subseteq \mathcal{L}_\forall(\mathcal{B}, Q) \end{aligned}$$

Theorem 5. *For any* icTA $\mathcal{B}$*,* $\mathcal{L}_\exists(\mathcal{B}, Q)$ *is the language of an* icTA *on* Q*.*

Construction 1 The existential timed languages can be computed by a variant of the region construction, of which the construction of [1] is a special case. Let $q \in$ *Proc* be a process whose clocks we want to eliminate, i.e. we have an icTA $\mathcal{B}$ on *Proc* and we would like to construct an icTA on *Proc* $\setminus \{q\}$ whose existential language is $\mathcal{L}_\exists(\mathcal{B}, Proc \setminus \{q\})$. We construct the region equivalence, but on the clocks of q only. This gives a region icTA without the clocks of q, and where the locations are now a pair of an original location and a region constraint on clocks of q, which has the required language. If we want to eliminate several processes, we eliminate them one by one: eliminating several processes together would give a result that does not reflect the independence of their clocks.

Theorem 6. *For any* icTA $\mathcal{B}$*,* $\mathcal{L}_\exists(\mathcal{B}, \{q\})$ *is the language of a* TA*.*

Construction 2 The existential timed language can be computed by the region construction [1] of a icTA for the independent clocks of the single process $q \in \mathit{Proc}$, whose clocks evolve at of same speed, i.e., they follow the same clock rate, then our model corresponds to a standard timed automata [2]. We construct the region equivalence with the clocks of q. A region constraint is of the form $\bigwedge_{q \in \mathit{Proc}} \phi_q$. This gives a region icTA with the clocks of q, and where the locations are now a pair of an original location and a region constraint on clocks of q, which has the required language. We label the region state by $\bigwedge_{q \in \mathit{Proc}} \phi_q$. Then we mark as final the locations where all members are final (which, in turn, means that one of their members is an original final state), to represent that the ISS must be accepted under evolution of time τ_q. The resulting automaton is a TA.

This variety of languages leads to three generalisations of the classical problems of emptiness, inclusion, intersection and union. First, the τ-wise definitions:

Definition 6. *Given icTA or DECA $\mathcal{A}, \mathcal{B}, \mathcal{C}$,*

1. *$\mathcal{C}$ is an τ-intersection of $\mathcal{A}, \mathcal{B}$ iff $\forall \tau \in Rates, \mathcal{L}(\mathcal{C}, \tau) = \mathcal{L}(\mathcal{A}, \tau) \cap \mathcal{L}(\mathcal{B}, \tau)$*
2. *$\mathcal{C}$ is an τ-union of $\mathcal{A}, \mathcal{B}$ iff $\forall \tau \in Rates, \mathcal{L}(\mathcal{C}, \tau) = \mathcal{L}(\mathcal{A}, \tau) \cup \mathcal{L}(\mathcal{B}, \tau)$*
3. *$\mathcal{C}$ is a τ-complement automaton of $\mathcal{A}$ iff $\forall \tau \in Rates, \mathcal{L}(\mathcal{C}, \tau) = \mathcal{L}(\mathcal{A}, \tau)^c$, where c is the complement operator.*
4. *$\mathcal{A}$ is a τ-language-included in $\mathcal{B}$ iff $\forall \tau \in Rates, \mathcal{L}(\mathcal{A}, \tau) \subseteq \mathcal{L}(\mathcal{B}, \tau)$*
5. *The τ-emptiness problem for $\mathcal{A}$ is $\forall \tau \in Rates, \mathcal{L}(\mathcal{A}, \tau) = \emptyset$*

The existential variant use respectively the existential timed language observed by $P \subseteq \mathit{Proc}$.

Definition 7. *Given icTA or DECA $\mathcal{A}, \mathcal{B}, \mathcal{C}$,*

1. *$\mathcal{C}$ is an $\exists$-intersection observed by P of $\mathcal{A}, \mathcal{B}$ iff $\mathcal{L}_\exists(\mathcal{C}, P) = \mathcal{L}_\exists(\mathcal{A}, P) \cap \mathcal{L}_\exists(\mathcal{B}, P)$*
2. *$\mathcal{C}$ is an $\exists$-union observed by P of $\mathcal{A}, \mathcal{B}$ iff $\mathcal{L}_\exists(\mathcal{C}, P) = \mathcal{L}_\exists(\mathcal{A}, P) \cup \mathcal{L}_\exists(\mathcal{B}, P)$*
3. *$\mathcal{C}$ is a $\exists$-complement automaton observed by P of $\mathcal{A}$ iff $\mathcal{L}_\exists(\mathcal{C}, P) = \mathcal{L}_\exists(\mathcal{A}, P)^c$, where c is the complement operator.*
4. *$\mathcal{A}$ is $\exists$-language-included observed by P in $\mathcal{B}$ iff $\mathcal{L}_\exists(\mathcal{A}, P) \subseteq \mathcal{L}_\exists(\mathcal{B}, P)$*
5. *The $\exists$-emptiness problem observed by P for $\mathcal{A}$ is $\mathcal{L}_\exists(\mathcal{A}, P) = \emptyset$*

The universal variant use respectively the universal timed language observed by $P \subseteq \mathit{Proc}$.

Definition 8. *Given icTA or DECA $\mathcal{A}, \mathcal{B}, \mathcal{C}$,*

1. *$\mathcal{C}$ is an $\forall$-intersection observed by P of $\mathcal{A}, \mathcal{B}$ iff $\mathcal{L}_\forall(\mathcal{C}, P) = \mathcal{L}_\forall(\mathcal{A}, P) \cap \mathcal{L}_\forall(\mathcal{B}, P)$*
2. *$\mathcal{C}$ is an $\forall$-union observed by P of $\mathcal{A}, \mathcal{B}$ iff $\mathcal{L}_\forall(\mathcal{C}, P) = \mathcal{L}_\forall(\mathcal{A}, P) \cup \mathcal{L}_\forall(\mathcal{B}, P)$*
3. *$\mathcal{C}$ is a $\forall$-complement automaton observed by P of $\mathcal{A}$ iff $\mathcal{L}_\forall(\mathcal{C}, P) = \mathcal{L}_\forall(\mathcal{A}, P)^c$, where c is the complement operator.*
4. *$\mathcal{A}$ is $\forall$-language-included observed by P in $\mathcal{B}$ iff $\mathcal{L}_\forall(\mathcal{A}, P) \subseteq \mathcal{L}_\forall(\mathcal{B}, P)$*
5. *The $\forall$-emptiness problem observed by P for $\mathcal{A}$ is $\mathcal{L}_\forall(\mathcal{A}, P) = \emptyset$*

The τ-wise definitions are indeed the strongest:

- A construction for union is the following :

Construction 3 Let $\mathcal{A} = (\Sigma^{\mathcal{A}}, X^{\mathcal{A}}, S^{\mathcal{A}}, s_0^{\mathcal{A}}, \rightarrow_{icta}^{\mathcal{A}}, \gamma^{\mathcal{A}}, Inv^{\mathcal{A}}, F^{\mathcal{A}}, \pi^{\mathcal{A}})$ and $\mathcal{B} = (\Sigma^{\mathcal{B}}, X^{\mathcal{B}}, S^{\mathcal{B}}, s_0^{\mathcal{B}}, \rightarrow_{icta}^{\mathcal{B}}, \gamma^{\mathcal{B}}, Inv^{\mathcal{B}}, F^{\mathcal{B}}, \pi^{\mathcal{B}})$ be two icTA. Without loss of generality we assume that the sets of clocks $X^{\mathcal{A}}$ and $X^{\mathcal{B}}$ (and respectively the sets of locations $S^{\mathcal{A}}$ and $S^{\mathcal{B}}$) are all pairwise disjoint.

Union Let $\mathcal{C} = (\Sigma^{\mathcal{C}}, X^{\mathcal{C}}, S^{\mathcal{C}}, s_0^{\mathcal{C}}, \rightarrow_{icta}^{\mathcal{C}}, \gamma^{\mathcal{C}}, Inv^{\mathcal{C}}, F^{\mathcal{C}}, \pi^{\mathcal{C}})$ be the icTA defined as follows:

(i) $\Sigma^{\mathcal{C}} = \Sigma^{\mathcal{A}} \cup \Sigma^{\mathcal{B}}$,
(ii) $X^{\mathcal{C}} = X^{\mathcal{A}} \cup X^{\mathcal{B}}$,
(iii) $S^{\mathcal{C}} = S^{\mathcal{A}} \cup S^{\mathcal{B}}$,
(iv) $s_0^{\mathcal{C}} = (s_0^{\mathcal{A}}, s_0^{\mathcal{B}})$,
(v) $\rightarrow_{icta}^{\mathcal{C}} = \rightarrow_{icta}^{\mathcal{A}} \cup \rightarrow_{icta}^{\mathcal{B}}$,
(vi) $\gamma^{\mathcal{C}} = \gamma^{\mathcal{A}} \cup \gamma^{\mathcal{B}}$,
(vii) $Inv^{\mathcal{C}} = Inv^{\mathcal{A}} \cup Inv^{\mathcal{B}}$,
(viii) $F^{\mathcal{C}} = F^{\mathcal{A}} \cup F^{\mathcal{B}}$,
(ix) $\pi^{\mathcal{C}} = \pi^{\mathcal{A}} \cup \pi^{\mathcal{B}}$,

The proof of union correctness is the following:

Correctness We need to show that for all $\tau \in Rates$, $\rho \in \mathcal{L}(\mathcal{C}, \tau)$ iff $\rho \in \mathcal{L}(\mathcal{A}, \tau) \cup \mathcal{L}(\mathcal{B}, \tau)$. Assume that $\rho = (\gamma(\zeta_0), \{t_0\}), (\gamma(s_1),]t_1, t_2[), \ldots, (\gamma(s_n),]t_{n-1}, t_n[) \in \mathcal{L}(\mathcal{C}, \tau)$ is an ISS and an icTA $\mathcal{C}$. Then ρ originates from some runs $\theta = (s_0^{\mathcal{C}}, \nu_0, t_0) \xrightarrow{\zeta_0} (s_1^{\mathcal{C}}, \nu_1, t_1) \ldots \xrightarrow{\zeta_{n-1}} (s_n^{\mathcal{C}}, \nu_n, t_n)$ of $\mathcal{C}$ with regard to τ, where a run is an alternating sequence of states and transitions. The states are triples of a location, a clock valuation, and lastly the reference time: $\{(s, \nu, t) \in S^{\mathcal{C}} \times \mathbb{R}_{\geq 0}^{X} \times \mathbb{R}_{\geq 0} \mid \nu \models Inv(s)\}$. The transitions must alternate between two types: (*i*) Delay transition, i.e. spending time in a location: $q_i \xrightarrow{d} q_i + d$, where $q_i = (s_i, \nu_i, t_i)$, and $q_i + d = (s_i, \nu_i + (\tau(t_i + d) - \tau(t_i)), t_i + d)$, if the invariant is continuously true in local time: $\forall t \in]t_i, t_i + d[$: $\nu_i + (\tau(t) - \tau(t_i)) \models Inv(s_i)$. (*ii*) Discrete transition: following a transition $\zeta_i = (s_{i-1}, \phi, Y, s_i) \in \rightarrow_{\mathsf{icTA}}$ when the clock constraint ϕ is satisfied. The clocks in Y are then reset. This transition is instantaneous. $(s_{i-1}, \nu_{i-1}, t_{i-1}) \xrightarrow{\zeta_i} (s_i, \nu_i, t_i)$, such that $\nu_{i-1} \models \phi$, $\nu_i = \nu_{i-1}[Y \rightarrow 0], t_{i-1} = t_i$. For any clock $x^{\mathcal{C}} \in X^{\mathcal{C}}$, $\nu(x^{\mathcal{C}}) = \tau(t) - \tau(t_i)$ where $i \geq 0$ is the index of the last transition which reset clock $x^{\mathcal{C}}$ or is $\tau(t)$ if $x^{\mathcal{C}}$ was never reset. We can say the $\mathcal{C}$ accepts ρ at time t with regard to τ, if there is a run θ for an equivalent of ρ that visits an accepting location $s_n^{\mathcal{C}} \in F^{\mathcal{C}}$ at t and on the *rate* τ, where we can safely assume due to the construction that, for each $1 \leq i \leq n$, $s_0^{\mathcal{C}} \xrightarrow{\rho}_{\mathcal{C}} s_n^{\mathcal{C}}$. By induction hypothesis we know that $\rho = (\gamma(\zeta_0), \{t_0\}), (\gamma(s_1),]t_1, t_2[), \ldots, (\gamma(s_n),]t_{n-1}, t_n[) \in (\mathcal{L}(\mathcal{A}, \tau) \cup \mathcal{L}(\mathcal{B}, \tau))$, then there is an accepting run $\theta' = (s_0^{\mathcal{A}\cup\mathcal{B}}, \nu_0', t_0) \xrightarrow{\zeta_0} (s_1^{\mathcal{A}\cup\mathcal{B}}, \nu_1', t_1) \ldots \xrightarrow{\zeta_{n-1}} (s_n^{\mathcal{A}\cup\mathcal{B}}, \nu_n', t_n)$ be the run for ρ that visits an accepting location $s_n^{\mathcal{A}\cup\mathcal{B}} \in (F^{\mathcal{A}} \cup F^{\mathcal{B}})$ at t and on the *rate* τ, where we can safely assume due to the construction that, for each $1 \leq i \leq n$, $s_0^{\mathcal{A}\cup\mathcal{B}} \xrightarrow{\rho}_{\mathcal{A}\cup\mathcal{B}} s_n^{\mathcal{A}\cup\mathcal{B}}$. Firstly, we say that $\mathcal{C}$ accepts an ISS ρ at t with regard to *rate* τ, if there is a run θ for ρ that visits a accepting location $s_n^{\mathcal{C}} \in F^{\mathcal{C}}$ at t. Secondly the clock valuation depends on the ISS ρ, on the reference time of evaluation t, and on the *rate* τ. It is easy to see that for each $1 \leq i \leq n$ the clock valuation assigns a (non-standard) positive real to each clock variable $\nu_i = \nu_i^{'}$. Thirdly, since $s_{n-1}^{\mathcal{C}} \in (S^{\mathcal{A}} \cup S^{\mathcal{B}})$ and there exists a transition $\zeta_{n-1}^{\mathcal{C}} = (s_{n-2}^{\mathcal{C}}, \phi, Y, s_{n-1}^{\mathcal{C}}) \in \rightarrow_{icTA}^{\mathcal{C}}$, there exists $s_n^{\mathcal{A}\cup\mathcal{B}} \in (F^{\mathcal{A}} \cup F^{\mathcal{B}})$ such that $s_n^{\mathcal{C}} \in (F^{\mathcal{A}} \cup F^{\mathcal{B}})$ and there exists a transition $\zeta_n^{\mathcal{A}\cup\mathcal{B}} = ((s_{n-1}^{\mathcal{A}}, \phi, Y, s_n^{\mathcal{A}}) \in \rightarrow_{icTA}^{\mathcal{A}} \cup (s_{n-1}^{\mathcal{B}}, \phi, Y, s_n^{\mathcal{B}}) \in \rightarrow_{icTA}^{\mathcal{B}})$ where the clock constraints are satisfied by the valuation $\nu_n^{'}$. Hence we can adjunct $s_0^{\mathcal{A}\cup\mathcal{B}} \xrightarrow{\rho} s_n^{\mathcal{A}\cup\mathcal{B}}$ to the run over $\mathcal{A}$ and $\mathcal{B}$. It follows immediately that $\mathcal{C}$ accepts the set of ISS ρ at time t and on the *rate* τ, then the language $\mathcal{L}(\mathcal{C}, \tau)$ is the set of ISS ρ of accepting runs of $\mathcal{C}$ with regard to τ, and the language $\mathcal{L}(\mathcal{A}, \tau) \cup \mathcal{L}(\mathcal{B}, \tau)$ is defined as the set of ISS ρ of accepting runs of the union of $\mathcal{A}$ and $\mathcal{B}$ with regard to τ. We have that $\mathcal{L}(\mathcal{C}, \tau) = \mathcal{L}(\mathcal{A}, \tau) \cup \mathcal{L}(\mathcal{B}, \tau)$.

- A construction for intersection is the following :

Construction 4 Let $\mathcal{A} = (\Sigma^{\mathcal{A}}, X^{\mathcal{A}}, S^{\mathcal{A}}, s_0^{\mathcal{A}}, \rightarrow_{icta}^{\mathcal{A}}, \gamma^{\mathcal{A}}, \mathit{Inv}^{\mathcal{A}}, F^{\mathcal{A}}, \pi^{\mathcal{A}})$ and $\mathcal{B} = (\Sigma^{\mathcal{B}}, X^{\mathcal{B}}, S^{\mathcal{B}}, s_0^{\mathcal{B}}, \rightarrow_{icta}^{\mathcal{B}}, \gamma^{\mathcal{B}}, \mathit{Inv}^{\mathcal{B}}, F^{\mathcal{B}}, \pi^{\mathcal{B}})$ be two icTA. Without loss of generality we assume that the sets of clocks $X^{\mathcal{A}}$ and $X^{\mathcal{B}}$ (and respectively the sets of locations $S^{\mathcal{A}}$ and $S^{\mathcal{B}}$) are all pairwise disjoint.

Intersection Let $\mathcal{C} = (\Sigma^{\mathcal{C}}, X^{\mathcal{C}}, S^{\mathcal{C}}, s_0^{\mathcal{C}}, \rightarrow_{icta}^{\mathcal{C}}, \gamma^{\mathcal{C}}, \mathit{Inv}^{\mathcal{C}}, F^{\mathcal{C}}, \pi^{\mathcal{C}})$ be the icTA defined as follows:

(i) $\Sigma^{\mathcal{C}} = \Sigma^{\mathcal{A}} \cap \Sigma^{\mathcal{B}}$,
(ii) $X^{\mathcal{C}} = X^{\mathcal{A}} \cup X^{\mathcal{B}}$,
(iii) $S^{\mathcal{C}} = S^{\mathcal{A}} \times S^{\mathcal{B}}$,
(iv) $s_0^{\mathcal{C}} = (s_0^{\mathcal{A}}, s_0^{\mathcal{B}})$,
(v) For all $s_1^{\mathcal{A}}$, $s_2^{\mathcal{A}} \in S^{\mathcal{A}}$, $s_1^{\mathcal{B}}$, $s_2^{\mathcal{B}} \in S^{\mathcal{B}}$, $\phi^{\mathcal{C}} = \phi^{\mathcal{A}} \wedge \phi^{\mathcal{B}}$, and $Y^{\mathcal{C}} = Y^{\mathcal{A}} \cup Y^{\mathcal{B}}$: $((s_1^{\mathcal{A}}, s_2^{\mathcal{A}}), \phi^{\mathcal{C}}, Y^{\mathcal{C}}, (s_1^{\mathcal{B}}, s_2^{\mathcal{B}})) \in \rightarrow_{icta}^{\mathcal{C}}$ iff there exist transitions $(s_1^{\mathcal{A}}, \phi^{\mathcal{A}}, Y^{\mathcal{A}}, s_2^{\mathcal{A}}) \in \rightarrow_{icta}^{\mathcal{A}}$ and $(s_1^{\mathcal{B}}, \phi^{\mathcal{B}}, Y^{\mathcal{B}}, s_2^{\mathcal{B}}) \in \rightarrow_{icta}^{\mathcal{B}}$,
(vi) For all $s_1^{\mathcal{A}} \in S^{\mathcal{A}}$, $s_2^{\mathcal{B}} \in S^{\mathcal{B}}$ and $s \in S^{\mathcal{C}}$, $\gamma^{\mathcal{C}}(s) = \gamma^{\mathcal{A}}(s_1) \wedge \gamma^{\mathcal{B}}(s_2)$,
(vii) $\mathit{Inv}^{\mathcal{C}} = \mathit{Inv}^{\mathcal{A}} \wedge \mathit{Inv}^{\mathcal{B}}$,
(viii) $F^{\mathcal{C}} = F^{\mathcal{A}} \times F^{\mathcal{B}}$,
(ix) $\pi^{\mathcal{C}} = \pi^{\mathcal{A}} \cup \pi^{\mathcal{B}}$,

The proof of intersection correctness is the following:

Correctness We need to show that for any $\tau \in \mathit{Rates}$, $\rho \in \mathcal{L}(\mathcal{C}, \tau)$ iff $\rho \in \mathcal{L}(\mathcal{A}, \tau) \cap \mathcal{L}(\mathcal{B}, \tau)$. Assume that $\rho = (\gamma(\zeta_0), \{t_0\}), (\gamma(s_1),]t_1, t_2[), \ldots, (\gamma(s_n),]t_{n-1}, t_n[) \in \mathcal{L}(\mathcal{C}, \tau)$ is an ISS and an icTA $\mathcal{C}$. Then ρ originates from some runs $\theta = (s_0^{\mathcal{C}}, \nu_0, t_0) \xrightarrow{\zeta_0} (s_1^{\mathcal{C}}, \nu_1, t_1) \ldots \xrightarrow{\zeta_{n-1}} (s_n^{\mathcal{C}}, \nu_n, t_n)$ of $\mathcal{C}$ with regard to τ, where a run is an alternating sequence of states and transitions. The states are triples of a location, a clock valuation, and lastly the reference time: $\{(s, \nu, t) \in S^{\mathcal{C}} \times \mathbb{R}_{\geq 0}^{X} \times \mathbb{R}_{\geq 0} \mid \nu \models \mathit{Inv}(s)\}$. The transitions must alternate between two types: (*i*) Delay transition, i.e. spending time in a location: $q_i \xrightarrow{d} q_i + d$, where $q_i = (s_i, \nu_i, t_i)$, and $q_i + d = (s_i, \nu_i + (\tau(t_i + d) - \tau(t_i)), t_i + d)$, if the invariant is continuously true in local time: $\forall t \in]t_i, t_i + d[: \nu_i + (\tau(t) - \tau(t_i)) \models \mathit{Inv}(s_i)$. (*ii*) Discrete transition: following a transition $\zeta_i = (s_{i-1}, \phi, Y, s_i) \in \rightarrow_{\mathsf{icTA}}$ when the clock constraint ϕ is satisfied. The clocks in Y are then reset. This transition is instantaneous. $(s_{i-1}, \nu_{i-1}, t_{i-1}) \xrightarrow{\zeta_i} (s_i, \nu_i, t_i)$, such that $\nu_{i-1} \models \phi$, $\nu_i = \nu_{i-1}[Y \rightarrow 0], t_{i-1} = t_i$. For any clock $x^{\mathcal{C}} \in X^{\mathcal{C}}$, $\nu(x^{\mathcal{C}}) = \tau(t) - \tau(t_i)$ where $i \geq 0$ is the index of the last transition which reset clock $x^{\mathcal{C}}$ or is $\tau(t)$ if $x^{\mathcal{C}}$ was never reset. We can say the $\mathcal{C}$ accepts ρ at time t with regard to τ, if there is a run θ for an equivalent of ρ that visits an accepting location $s_n^{\mathcal{C}} \in F^{\mathcal{C}}$ at t and on the *rate* τ, where we can safely assume due to the construction that, for each $1 \leq i \leq n$, $s_0^{\mathcal{C}} \xrightarrow{\rho}_{\mathcal{C}} s_n^{\mathcal{C}}$. By induction hypothesis we know that $\rho = (\gamma(\zeta_0), \{t_0\}), (\gamma(s_1),]t_1, t_2[), \ldots, (\gamma(s_n),]t_{n-1}, t_n[) \in (\mathcal{L}(\mathcal{A}, \tau) \cap \mathcal{L}(\mathcal{B}, \tau))$, then there is an accepting run $\theta' = (s_0^{\mathcal{A}\cap\mathcal{B}}, \nu_0', t_0) \xrightarrow{\zeta_0} (s_1^{\mathcal{A}\cap\mathcal{B}}, \nu_1', t_1) \ldots \xrightarrow{\zeta_{n-1}} (s_n^{\mathcal{A}\cap\mathcal{B}}, \nu_n', t_n)$ be the run for ρ that visits an accepting location $s_n^{\mathcal{A}\cap\mathcal{B}} \in (F^{\mathcal{A}} \times F^{\mathcal{B}})$ at time t and on the rate τ, where we can safely assume due to the construction that, for each $1 \leq i \leq n$, $s_0^{\mathcal{A}\cap\mathcal{B}} \xrightarrow{\rho}_{\mathcal{A}\cap\mathcal{B}} s_n^{\mathcal{A}\cap\mathcal{B}}$. Firstly, we say that $\mathcal{C}$ accepts an ISS ρ at t with regard to *rate* τ, if there is a run θ for ρ that visits a accepting location $s_n \in F^{\mathcal{C}}$ at t. Secondly the clock valuation depends on the ISS ρ, on the reference time of evaluation t, and on the *rate* τ. It is easy to see that for each $1 \leq i \leq n$ the clock valuation assigns a (non-standard) positive real to each clock variable $\nu_i = \nu_i'$. Thirdly, since $s_{n-1}^{\mathcal{C}} \in (S^{\mathcal{A}} \times S^{\mathcal{B}})$ and there exists a transition $\zeta_{n-1}^{\mathcal{C}} = (s_{n-2}, \phi, Y, s_{n-1}) \in \rightarrow_{icTA}^{\mathcal{C}}$, there exists $s_n^{\mathcal{A}\cap\mathcal{B}} \in (F^{\mathcal{A}} \times F^{\mathcal{B}})$ such that $s_n^{\mathcal{C}} \in (F^{\mathcal{A}} \times F^{\mathcal{B}})$ and there exists a transition $\zeta_n^{\mathcal{A}\cap\mathcal{B}} = ((s_{n-1}^{\mathcal{A}}, s_n^{\mathcal{A}}), \phi^{\mathcal{C}}, Y^{\mathcal{C}}, (s_{n-1}^{\mathcal{B}}, s_n^{\mathcal{B}})) \in \rightarrow_{icta}^{\mathcal{C}}$ iff there exist transitions $(s_{n-1}^{\mathcal{A}}, \phi^{\mathcal{A}}, Y^{\mathcal{A}}, s_n^{\mathcal{A}}) \in \rightarrow_{icta}^{\mathcal{A}}$ and $(s_{n-1}^{\mathcal{B}}, \phi^{\mathcal{B}}, Y^{\mathcal{B}}, s_n^{\mathcal{B}}) \in \rightarrow_{icta}^{\mathcal{B}}$, for all $s_{n-1}^{\mathcal{A}}, s_n^{\mathcal{A}} \in S^{\mathcal{A}}$, $s_{n-1}^{\mathcal{B}}, s_n^{\mathcal{B}} \in S^{\mathcal{B}}$, $\phi^{\mathcal{C}} = \phi^{\mathcal{A}} \wedge \phi^{\mathcal{B}}$, and $Y^{\mathcal{C}} = Y^{\mathcal{A}} \cup Y^{\mathcal{B}}$, where the clock constraints are satisfied by the valuation ν_n'. Hence we can adjunct $s_0^{\mathcal{A}\cap\mathcal{B}} \xrightarrow{\rho} s_n^{\mathcal{A}\cap\mathcal{B}}$ to the run over $\mathcal{A}$ and $\mathcal{B}$. It follows immediately that $\mathcal{C}$ accepts the set of ISS ρ at time t and on the *rate* τ, then the language $\mathcal{L}(\mathcal{C}, \tau)$ is the set of ISS ρ of accepting runs of $\mathcal{C}$ with regard to τ, and the language $\mathcal{L}(\mathcal{A}, \tau) \cap \mathcal{L}(\mathcal{B}, \tau)$ is defined as the set of ISS ρ of accepting runs of the intersection of $\mathcal{A}$ and $\mathcal{B}$ with regard to τ. We have that $\mathcal{L}(\mathcal{C}, \tau) = \mathcal{L}(\mathcal{A}, \tau) \cap \mathcal{L}(\mathcal{B}, \tau)$.

Theorem 7. *If $\mathcal{C}$ is an τ-union of $\mathcal{A}, \mathcal{B}$, then for any $P \subseteq Proc$, $\mathcal{L}_{\exists}(\mathcal{C}, P) = \mathcal{L}_{\exists}(\mathcal{A}, P) \cup \mathcal{L}_{\exists}(\mathcal{B}, P)$.*

Proof.

$$\begin{aligned}
\mathcal{L}_{\exists}(\mathcal{A}, P) \cup \mathcal{L}_{\exists}(\mathcal{B}, P) &= (\bigcup_{\tau \in \mathit{Rates}} \mathcal{L}(\mathcal{A}, \tau, P)) \cup (\bigcup_{\tau \in \mathit{Rates}} \mathcal{L}(\mathcal{B}, \tau, P)) \\
&= (\bigcup_{\tau \in \mathit{Rates}} \tau_P(\mathcal{L}(\mathcal{A}, \tau))) \cup (\bigcup_{\tau \in \mathit{Rates}} \tau_P(\mathcal{L}(\mathcal{B}, \tau))) \\
&= (\bigcup_{\tau \in \mathit{Rates}} \{\tau_P(\rho) \mid \rho \in \mathcal{L}(\mathcal{A}, \tau)\}) \cup \\
&\quad (\bigcup_{\tau \in \mathit{Rates}} \{\tau_P(\rho) \mid \rho \in \mathcal{L}(\mathcal{B}, \tau)\}) \\
&= \{(\sigma, \tau_P(I)) \mid (\rho \in \mathcal{L}(\mathcal{A}, \tau)) \vee (\rho \in \mathcal{L}(\mathcal{B}, \tau))
\end{aligned}$$

$$
\begin{aligned}
&\wedge\ \tau \in \mathit{Rates}\} \\
&= \{(\sigma, \tau_P(I)) \mid \rho \in (\mathcal{L}(\mathcal{A}, \tau) \cup \mathcal{L}(\mathcal{B}, \tau)) \\
&\quad \wedge\ \tau \in \mathit{Rates}\} \\
&= \{(\sigma, \tau_P(I)) \mid \rho \in \mathcal{L}(\mathcal{C}, \tau) \wedge\ \tau \in \mathit{Rates}\} \\
&= \mathcal{L}_{\exists}(\mathcal{C}, P)
\end{aligned}
$$

Theorem 8. *If $\mathcal{C}$ is an τ-intersection of $\mathcal{A}, \mathcal{B}$, then for any $P \subseteq \mathit{Proc}$, $\mathcal{L}_{\forall}(\mathcal{C}, P) = \mathcal{L}_{\forall}(\mathcal{A}, \tau) \cap \mathcal{L}_{\forall}(\mathcal{B}, P)$.*

Proof.

$$
\begin{aligned}
\mathcal{L}_{\forall}(\mathcal{A}, P) \cap \mathcal{L}_{\forall}(\mathcal{B}, P) &= (\bigcap_{\tau \in \mathit{Rates}} \mathcal{L}(\mathcal{A}, \tau, P)) \cap (\bigcap_{\tau \in \mathit{Rates}} \mathcal{L}(\mathcal{B}, \tau, P)) \\
&= (\bigcap_{\tau \in \mathit{Rates}} \tau_P(\mathcal{L}(\mathcal{A}, \tau))) \cap (\bigcap_{\tau \in \mathit{Rates}} \tau_P(\mathcal{L}(\mathcal{B}, \tau))) \\
&= (\bigcap_{\tau \in \mathit{Rates}} \{\tau_P(\rho) \mid \rho \in \mathcal{L}(\mathcal{A}, \tau)\}) \cap \\
&\quad (\bigcap_{\tau \in \mathit{Rates}} \{\tau_P(\rho) \mid \rho \in \mathcal{L}(\mathcal{B}, \tau)\}) \\
&= (\bigcap_{\tau \in \mathit{Rates}} \{\tau_P(\rho) \mid (\rho \in \mathcal{L}(\mathcal{A}, \tau)) \wedge \\
&\quad (\rho \in \mathcal{L}(\mathcal{B}, \tau))\} \wedge\ \tau \in \mathit{Rates}\}) \\
&= (\bigcap_{\tau \in \mathit{Rates}} \{\tau_P(\rho) \mid \rho \in (\mathcal{L}(\mathcal{A}, \tau) \cap \\
&\quad \mathcal{L}(\mathcal{B}, \tau))\} \wedge\ \tau \in \mathit{Rates}\}) \\
&= (\bigcap_{\tau \in \mathit{Rates}} \{\tau_P(\rho) \mid \rho \in \mathcal{L}(\mathcal{C}, \tau) \wedge\ \tau \in \mathit{Rates}\}) \\
&= \mathcal{L}_{\forall}(\mathcal{C}, P)
\end{aligned}
$$

Theorem 9. *If $\mathcal{C}$ is an τ-intersection of $\mathcal{A}, \mathcal{B}$, then for any $P \subseteq \mathit{Proc}$, $\mathcal{L}_{\exists}(\mathcal{C}, P) = \mathcal{L}_{\exists}(\mathcal{A}, P) \cap \mathcal{L}_{\exists}(\mathcal{B}, P)$.*

Proof. The τ-intersection of $\mathcal{C}$, follow from the constructions similar to the Theorem 8.

Theorem 10. *If $\mathcal{C}$ is an τ-union of $\mathcal{A}, \mathcal{B}$, then for any $P \subseteq \mathit{Proc}$, $\mathcal{L}_{\forall}(\mathcal{C}, P) = \mathcal{L}_{\forall}(\mathcal{A}, \tau) \cup \mathcal{L}_{\forall}(\mathcal{B}, P)$.*

Proof. The τ-union of $\mathcal{C}$, follow from the constructions similar to the Theorem 7.

We note that the above theorems are valid for the finite version, but also for the infinite ones, e.g. for Büchi automata, which are determinized to a parity automaton

[12]. The different classes of union for the infinite version of an icTA can be constructed similar to the Construction 3. The construction of intersection for the infinite version of an icTA can be constructed using the same construction of intersection for the Büchi automata.

- A construction for intersection for infinite version is the following :

Construction 5 Let $\mathcal{A} = (\Sigma^{\mathcal{A}}, X^{\mathcal{A}}, S^{\mathcal{A}}, s_0^{\mathcal{A}}, \rightarrow_{icta}^{\mathcal{A}}, \gamma^{\mathcal{A}}, Inv^{\mathcal{A}}, F^{\mathcal{A}}, \pi^{\mathcal{A}})$ and $\mathcal{B} = (\Sigma^{\mathcal{B}}, X^{\mathcal{B}}, S^{\mathcal{B}}, s_0^{\mathcal{B}}, \rightarrow_{icta}^{\mathcal{B}}, \gamma^{\mathcal{B}}, Inv^{\mathcal{B}}, F^{\mathcal{B}}, \pi^{\mathcal{B}})$ be Büchi icTA. Without loss of generality we assume that the sets of clocks $X^{\mathcal{A}}$ and $X^{\mathcal{B}}$ (and respectively the sets of locations $S^{\mathcal{A}}$ and $S^{\mathcal{B}}$) are all pairwise disjoint.

Intersection Let $\mathcal{C} = (\Sigma^{\mathcal{C}}, X^{\mathcal{C}}, S^{\mathcal{C}}, s_0^{\mathcal{C}}, \rightarrow_{icta}^{\mathcal{C}}, \gamma^{\mathcal{C}}, Inv^{\mathcal{C}}, F^{\mathcal{C}}, \pi^{\mathcal{C}})$ be the icTA defined as follows:

(i) $\Sigma^{\mathcal{C}} = \Sigma^{\mathcal{A}} \cap \Sigma^{\mathcal{B}}$,
(ii) $X^{\mathcal{C}} = X^{\mathcal{A}} \cup X^{\mathcal{B}}$,
(iii) $S^{\mathcal{C}} = S^{\mathcal{A}} \times S^{\mathcal{B}} \times \{1, 2\}$,
(iv) $s_0^{\mathcal{C}} = (s_0^{\mathcal{A}}, s_0^{\mathcal{B}}, 1)$,
(v) For all $s_1^{\mathcal{A}}$, $s_2^{\mathcal{A}} \in S^{\mathcal{A}}$, $s_1^{\mathcal{B}}$, $s_2^{\mathcal{B}} \in S^{\mathcal{B}}$, $\phi^{\mathcal{C}} = \phi^{\mathcal{A}} \wedge \phi^{\mathcal{B}}$, and $Y^{\mathcal{C}} = Y^{\mathcal{A}} \cup Y^{\mathcal{B}}$ and $i, j \in \{1, 2\}$: $((s_1^{\mathcal{A}}, s_1^{\mathcal{B}}, i), \phi^{\mathcal{C}}, Y^{\mathcal{C}}, (s_2^{\mathcal{A}}, s_2^{\mathcal{B}}, j)) \in \rightarrow_{icta}^{\mathcal{C}}$ iff there exist transitions $(s_1^{\mathcal{A}}, \phi^{\mathcal{A}}, Y^{\mathcal{A}}, s_2^{\mathcal{A}}) \in \rightarrow_{icta}^{\mathcal{A}}$ and $(s_1^{\mathcal{B}}, \phi^{\mathcal{B}}, Y^{\mathcal{B}}, s_2^{\mathcal{B}}) \in \rightarrow_{icta}^{\mathcal{B}}$ and:
 (a) if $i = 1$ and $s_1^{\mathcal{A}} \in F^{\mathcal{A}}$, then $j = 2$, or
 (b) if $i = 2$ and $s_1^{\mathcal{B}} \in F^{\mathcal{B}}$, then $j = 1$, or
 (c) neither a) or b) above applies and $j = i$.
(vi) For all $s_1^{\mathcal{A}} \in S^{\mathcal{A}}$, $s_2^{\mathcal{B}} \in S^{\mathcal{B}}$ and $s \in S^{\mathcal{C}}$, $\gamma^{\mathcal{C}}(s) = \gamma^{\mathcal{A}}(s_1) \wedge \gamma^{\mathcal{B}}(s_2)$,
(vii) $Inv^{\mathcal{C}} = Inv^{\mathcal{A}} \wedge Inv^{\mathcal{B}}$,
(viii) $F^{\mathcal{C}} = F^{\mathcal{A}} \times S^{\mathcal{B}} \times \{1\}$,
(ix) $\pi^{\mathcal{C}} = \pi^{\mathcal{A}} \cup \pi^{\mathcal{B}}$,

The proof of intersection correctness for infinite version is the following:

Correctness We need to show that for any $\tau \in Rates$, $\rho \in \mathcal{L}(\mathcal{C}, \tau)$ iff $\rho \in \mathcal{L}(\mathcal{A}, \tau) \cap \mathcal{L}(\mathcal{B}, \tau)$. Assume that $\rho = (\gamma(\zeta_0), \{t_0\}), (\gamma(s_1),]t_1, t_2[), \ldots \in \mathcal{L}(\mathcal{C}, \tau)$ is an ISS and an icTA $\mathcal{C}$. Then ρ originates from some runs $\theta = (s_0^{\mathcal{C}}, \nu_0, t_0) \xrightarrow{\zeta_0} (s_1^{\mathcal{C}}, \nu_1, t_1)$ $\ldots$ of $\mathcal{C}$ with regard to τ, where a run is an alternating sequence of states and transitions. The states are triples of a location, a clock valuation, and lastly the reference time: $\{(s, \nu, t) \in S^{\mathcal{C}} \times \mathbb{R}_{\geq 0}^{X} \times \mathbb{R}_{\geq 0} \mid \nu \models Inv(s)\}$. The transitions must alternate between two types: (*i*) Delay transition, i.e. spending time in a location: $q_i \xrightarrow{d} q_i + d$, where $q_i = (s_i, \nu_i, t_i)$, and $q_i + d = (s_i, \nu_i + (\tau(t_i + d) - \tau(t_i)), t_i + d)$, if the invariant is continuously true in local time: $\forall t \in]t_i, t_i + d[$: $\nu_i + (\tau(t) - \tau(t_i)) \models Inv(s_i)$. (*ii*) Discrete transition: following a transition $\zeta_i = (s_{i-1}, \phi, Y, s_i) \in \rightarrow_{\mathsf{icTA}}$ when the clock constraint ϕ is satisfied. The clocks in Y are then reset. This transition is instantaneous. $(s_{i-1}, \nu_{i-1}, t_{i-1}) \xrightarrow{\zeta_i} (s_i, \nu_i, t_i)$, such that $\nu_{i-1} \models \phi$, $\nu_i = \nu_{i-1}[Y \rightarrow 0], t_{i-1} = t_i$. For any clock $x^{\mathcal{C}} \in X^{\mathcal{C}}$, $\nu(x^{\mathcal{C}}) = \tau(t) - \tau(t_i)$ where $i \geq 0$ is the index of the last transition which reset clock $x^{\mathcal{C}}$ or is $\tau(t)$ if $x^{\mathcal{C}}$ was never reset. To determine whether a run of ρ is accepting, we consider the set $inf(\rho) \subseteq F^{\mathcal{C}}$ which is the set of all locations that occur in ρ infinitely often. We can say the $\mathcal{C}$ accepts

ρ at time t with regard to τ, if there is a run θ for an equivalent of ρ that visits infinitely often accepting locations in $F^{\mathcal{C}}$ at t and on the *rate* τ, where we can safely assume due to the construction that, *inf*$(\rho) \cap F^{\mathcal{C}} \neq \emptyset$. By induction hypothesis we know that $\rho = (\gamma(\zeta_0), \{t_0\}), (\gamma(s_1),]t_1, t_2[), \ldots \in (\mathcal{L}(\mathcal{A}, \tau) \cap \mathcal{L}(\mathcal{B}, \tau))$, then there is an accepting run $\theta' = (s_0^{\mathcal{A}\cap\mathcal{B}}, \nu_0', t_0) \xrightarrow{\zeta_0} (s_1^{\mathcal{A}\cap\mathcal{B}}, \nu_1', t_1) \ldots$ be the run for ρ at time t and on the rate τ, where we can safely assume due to the construction that, *inf*$(\rho) \cap F^{\mathcal{A}\cap\mathcal{B}} \neq \emptyset$. Firstly, we say that $\mathcal{C}$ accepts an ISS ρ at t with regard to *rate* τ, if there is a run θ for ρ that visits visits infinitely often accepting locations in $F^{\mathcal{A}\cap\mathcal{B}}$ at t. Secondly the clock valuation depends on the ISS ρ, on the reference time of evaluation t, and on the *rate* τ. It is easy to see that for each for $i \geq 0$ the clock valuation assigns a (non-standard) positive real to each clock variable $\nu_i = \nu_i'$. Thirdly, since $s^{\mathcal{C}} \in (S^{\mathcal{A}} \times S^{\mathcal{B}} \times \{1, 2\})$ and there exists a transition $\zeta^{\mathcal{C}} = (s, \phi, Y, s')$ $\in \rightarrow_{icTA}^{\mathcal{C}}$, there exists a $s^{\mathcal{A}\cap\mathcal{B}}$ that visit infinitely often to $(F^{\mathcal{A}} \times S^{\mathcal{B}} \times \{1\})$ such that $s^{\mathcal{C}} \in (S^{\mathcal{A}} \times S^{\mathcal{B}} \times \{1, 2\})$ and there exists a transition $\zeta^{\mathcal{A}\cap\mathcal{B}} = ((s_1^{\mathcal{A}}, s_1^{\mathcal{B}}, i), \phi^{\mathcal{C}}$, $Y^{\mathcal{C}}$, $(s_2^{\mathcal{A}}, s_2^{\mathcal{B}}, j)) \in \rightarrow_{icta}^{\mathcal{C}}$ iff there exist transitions $(s_1^{\mathcal{A}}, \phi^{\mathcal{A}}, Y^{\mathcal{A}}, s_2^{\mathcal{A}}) \in \rightarrow_{icta}^{\mathcal{A}}$ and $(s_1^{\mathcal{B}}, \phi^{\mathcal{B}}, Y^{\mathcal{B}}, s_2^{\mathcal{B}}) \in \rightarrow_{icta}^{\mathcal{B}}$ for all $s_1^{\mathcal{A}}$, $s_2^{\mathcal{A}} \in S^{\mathcal{A}}$, $s_1^{\mathcal{B}}$, $s_2^{\mathcal{B}} \in S^{\mathcal{B}}$, $\phi^{\mathcal{C}} = \phi^{\mathcal{A}} \wedge \phi^{\mathcal{B}}$, and $Y^{\mathcal{C}} = Y^{\mathcal{A}} \cup Y^{\mathcal{B}}$ and $i, j \in \{1, 2\}$, and:

(a) if $i = 1$ and $s_1^{\mathcal{A}} \in F^{\mathcal{A}}$, then $j = 2$, or

(b) if $i = 2$ and $s_1^{\mathcal{B}} \in F^{\mathcal{B}}$, then $j = 1$, or

(c) neither a) or b) above applies and $j = i$.

Hence we can adjunct *inf*$(\rho) \cap (F^{\mathcal{A}} \times S^{\mathcal{B}} \times \{1\})$ to the run over $\mathcal{A}$ and $\mathcal{B}$. It follows immediately that $\mathcal{C}$ accepts the set of ISS ρ at time t and on the *rate* τ, then the language $\mathcal{L}(\mathcal{C}, \tau)$ is the set of ISS ρ of accepting runs of $\mathcal{C}$ with regard to τ, and the language $\mathcal{L}(\mathcal{A}, \tau) \cap \mathcal{L}(\mathcal{B}, \tau)$ is defined as the set of ISS ρ of accepting runs of the intersection of $\mathcal{A}$ and $\mathcal{B}$ with regard to τ. We have that $\mathcal{L}(\mathcal{C}, \tau) = \mathcal{L}(\mathcal{A}, \tau) \cap \mathcal{L}(\mathcal{B}, \tau)$.

However, icTA are not determinizable, not closed under complement, and their inclusion problem is undecidable (whether τ-wise, existential with at least one observer, or universal with at most one observer), because TA [2] are a special case of icTA.

5 Distributed Event Clock Automata

To restore full decidability, we use event clocks [5]. For expressiveness, we use RECA [9] with independent clocks [1]. The event clock $x_{\mathcal{A}}^q$ (or $y_{\mathcal{A}}^q$) denotes records the time since the last (resp. next) time that the automaton $\mathcal{A}$ could visit a monitored state, measured in the local time of process q.

Definition 9. *A distributed recursive event clock automaton (DECA) is a pair* $(\mathcal{A}, \pi)$ *where* $\mathcal{A}$ *is a RECA and* $\pi : C \to Proc$ *maps each clock to a process.*

Definition 10. *A* run θ *of a DECA* $\mathcal{A}$ *for a rate* τ *is a pair of sequences* (s, I)*:* s *gives an alternation of transitions and locations* $\delta_1, s_1, \delta_2, s_2, \ldots$*, and* I *is an interval sequence, such that:*

(i) The run starts from the initial state: $\delta_1 \in \{s_0\} \times S$.

(ii) For all $i > 1$*, the run follows a discrete transition:* $\delta_i = (s_{i-1}, s_i) \in \rightarrow_{reca}$

(iii) The clock constraints are satisfied by the valuation of the clocks defined below: $\forall t \in \mathbb{R}_{\geq 0}, \nu(\rho, t, \tau) \models \delta(\theta(t))$.

(iv) It satisfies the acceptance condition, e.g. it visits infinitely often a Büchi accepting location.

The ISS of a run $\theta = (s, I)$ of a DECA $\mathcal{A}$ is the pair $(\gamma(s), I)$. We say that $\mathcal{A}$ accepts an ISS ρ at t with τ, if there is a run θ for an equivalent of ρ that visits a monitored location at t. This is noted by $(t, \rho) \in \mathcal{L}^+(\mathcal{A}, \tau)$. This time t will be used to reset the clock, below.

The clock valuation depends on the ISS ρ, on the reference time of evaluation t, and on the rate τ. It assigns a (non-standard) positive real, or undefined, to each clock variable.

$$\nu(\rho, t, \tau, x^q_{\mathcal{B}}) = \begin{cases} \tau_q(t) - \tau_q(r) & \text{if } r = \max\{s < t | (s, \rho) \in \mathcal{L}^+(\mathcal{B}, \tau)\} \text{ exists} \\ (\tau_q(t) - \tau_q(r))^+ & \text{else, if } r = \sup\{s < t | (s, \rho) \in \mathcal{L}^+(\mathcal{B}, \tau)\} \text{ exists} \\ \bot & \text{else} \end{cases}$$

$$\nu(\rho, t, \tau, y^q_{\mathcal{B}}) = \begin{cases} \tau_q(l) - \tau_q(t) & \text{if } l = \min\{s > t | (s, \rho) \in \mathcal{L}^+(\mathcal{B}, \tau)\} \text{ exists} \\ (\tau_q(l) - \tau_q(t))^+ & \text{else, if } l = \inf\{s > t \,| (s, \rho) \in \mathcal{L}^+(\mathcal{B}, \tau)\} \text{ exists} \\ \bot & \text{else} \end{cases}$$

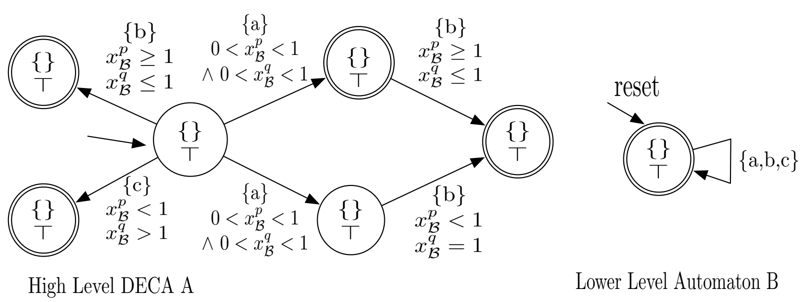

Fig. 1. Example of DECA from [1]

Example 1. The example of Fig.1 from [1] is in fact both a DECA and an icTA $\mathcal{A}$ over *Proc* $= \{p, q\}$, and the set of propositions $\mathbb{P} = \{a, b, c\}$. States have an empty labelling. Both clocks are reset by the initial monitored transition of $\mathcal{B}$. After this, they may diverge. The existential timed language, here, is read from the automaton:

$$\mathcal{L}_\exists(\mathcal{A}, Proc) = \mathsf{ITL}^1(\{(a, t^p_1, t^q_1) \mid 0 < t^p_1 < 1 \wedge 0 < t^q_1 < 1\}$$
$$\cup\{(b, t^p_1, t^q_1) \mid t^p_1 \geq 1 \wedge 0 < t^q_1 \leq 1\} \cup \{(c, t^p_1, t^q_1) \mid 0 < t^p_1 < 1 \wedge t^q_1 > 1\}$$
$$\cup\{(a, t^p_1, t^q_1), (b, t^p_2, t^q_2) \mid 0 < t^p_1 < 1 \wedge 0 < t^q_1 < 1 \wedge t^q_1 < t^q_2 \leq 1 \wedge t^p_1 < t^p_2\})$$

Here, all universal timed languages are empty: $\mathcal{L}_\forall(\mathcal{A}, P) = \emptyset$ for $P \geq 2$. For instance, we cannot have $(a, t_a) \in \mathcal{L}_\forall(\mathcal{A}, \{p\})$, because there are some τ where the time of q increases steeply, and gets over 1 before the time of p could reach t_a. However, the universal untimed language $\mathcal{L}_\forall(\mathcal{A}, \emptyset)$ is $\{a, ab\}$.

5.1 Multi-Timed Languages of DECA

DECA inherit the main property of RECA: they are determinizable. The determinization preserves the τ-wise, existential and universal languages.

Definition 11. *Two states or locations s_1, s_2 have* disjoint labellings *iff $\gamma(s_1) \neq \gamma(s_2)$ or there is no clock valuation ν such that $\nu \models \delta(s_1)$ and $\nu \models \delta(s_2)$.*

Definition 12. *A DECA or RECA $\mathcal{A}$ is deterministic iff all the following conditions hold:*

(i) $\mathcal{A}$ has exactly one initial location $s_0 \in S$ and,
(ii) Any two distinct successor locations $s_2 \neq s_3, s_1 \to s_2, s_1 \to s_3$, and with same labellings have mutually exclusive clock constraints, i.e. for all clock valuation ν, $\nu \not\models \delta(s_3) \wedge \delta(s_2)$.

Definition 13. *A* DECA *$\mathcal{A}$ is complete iff: for any symbol $Q \in \Sigma$, any clock valuation ν, and for any location $s \in S$ there is a successor location $s_1 \in S$ with $\gamma(s_1) = Q$ and $\nu \models \delta(s_1)$.*

Therefore, if $\mathcal{A}$ is a complete DECA, then for each ISS ρ, there is at least one accepting run of $\mathcal{A}$. The determinism ensures that, at each time t during a run, the choice of the next state is uniquely determined by the current location of the automaton and (ρ, τ).

We give the procedure for the construction of a deterministic Det($\mathcal{A}$) from the given a DECA $\mathcal{A}$. This construction is identical to the construction for RECA.

Construction 6 Given $\mathcal{A}$, we construct the deterministic *DECA* $\mathcal{B}$ = Det($\mathcal{A}$) as follows:

(i) The set of propositions used in $\mathcal{B}$ is the same as the set of propositions used in $\mathcal{A}$: $\Sigma^{\mathcal{B}} = \Sigma^{\mathcal{A}}$,
(ii) The set of clocks used in $\mathcal{B}$ is the same as the set clocks used in $\mathcal{A}$, $C^{\mathcal{B}} = C^{\mathcal{A}}$.
(iii) The set of locations of $\mathcal{B}$ is the set of non-empty subsets of locations of $\mathcal{A}$ with the same labelling, that is $\{s_1, s_2, \cdots, s_n\} \in S^{\mathcal{B}}$ iff :
  (a) $n \geq 1$,
  (b) for all i, $1 \leq i \leq n$: $s_i \in S^{\mathcal{A}}$,
  (c) for all i, j such that $1 \leq i < j \leq n$, we have that $\gamma^{\mathcal{A}}(s_i) = \gamma^{\mathcal{A}}(s_j)$.
(iv) The starting location of $\mathcal{B}$ is the only initial location of $\mathcal{A}$, that is, $q = s_0^{\mathcal{B}}$ where $q \in S^{\mathcal{B}}$, iff:
  (a) $s = s_0^{\mathcal{A}}$, where $s \in S^{\mathcal{A}}$,
  (b) there does not exists a location q' with

[1] ITL will add the missing intervals between time points.

i. $\gamma^{\mathcal{B}}(q') = \gamma^{\mathcal{B}}(q)$,
ii. $s = q'$, then $s = s_0^{\mathcal{A}}$,
iii. $q = q'$.

(v) A transition relation, we define $(q_1, q_2) \in \rightarrow_{reca}^{\mathcal{B}} \subseteq S^{\mathcal{B}} \times S^{\mathcal{B}}$ iff:
  (a) for all $s_2 \in q_2$, there exists $s_1 \in q_1$ such that, $(s_1, s_2) \in \rightarrow_{deca}^{\mathcal{A}}$; the locations in q_2 are $\rightarrow_{deca}^{\mathcal{A}}$-successors of locations in q_1,
  (b) for all $s_2 \in S^{\mathcal{A}}$ such that $\gamma^{\mathcal{A}}(s_2) = \gamma^{\mathcal{B}}(q_2)$ and there exists $s_1 \in q_1$ with $(s_1, s_2) \in \rightarrow_{deca}^{\mathcal{A}}$, we have $s_2 \in q_2$, i.e. q_2 is the maximal set of locations that share the label of q_2 and are $\rightarrow_{deca}^{\mathcal{A}}$-successors of a location of q_1.

(vi) A location $q \in S^{\mathcal{B}}$ belongs to the set $M^{\mathcal{B}}$ of monitored locations iff there exists a location of $\mathcal{A}$ in q that is monitored, i.e. $q \in M^{\mathcal{B}}$ iff there exists $s \in q$ such that $s \in M^{\mathcal{A}}$.

(vii) A transition $(q_1, q_2) \in \rightarrow_{reca}^{\mathcal{B}} \subseteq S^{\mathcal{B}} \times S^{\mathcal{B}}$ belongs to the set $M^{\mathcal{B}}$ of monitored locations iff there exists a transition of $\mathcal{A}$ in (q_1, q_2) that is monitored, i.e. $(q_1, q_2) \in M^{\mathcal{B}}$ iff there exists $(s_1, s_2) \in (q_1, q_2)$ such that $(s_1, s_2) \in M^{\mathcal{A}}$.

(viii) The labeling function $\gamma^{\mathcal{B}}(q) = \gamma^{\mathcal{A}}(s)$ with $s \in q$, for all $q \in S^{\mathcal{B}}$. The locations $q \in S^{\mathcal{B}}$ are labelled with the same label as in $\mathcal{A}$.

(ix) The labeling function for a transition relation $(q_1, q_2) \in \rightarrow_{reca}^{\mathcal{B}}$ and $(s_1, s_2) \in \rightarrow_{reca}^{\mathcal{A}}$, $\gamma^{\mathcal{B}}((q_1, q_2)) = \gamma^{\mathcal{A}}((s_1, s_2))$ with $(s_1, s_2) \in (q_1, q_2)$, for all $(q_1, q_2) \in \rightarrow_{deca}^{\mathcal{B}} \subseteq S^{\mathcal{B}} \times S^{\mathcal{B}}$. The transitions $(q_1, q_2) \in \rightarrow_{deca}^{\mathcal{B}} \subseteq S^{\mathcal{B}} \times S^{\mathcal{B}}$ are labelled with the same label as in $\mathcal{A}$.

(x) The invariant clock constraint $\delta^{\mathcal{B}}(q) = \bigwedge \delta^{\mathcal{A}}(s)$ with $s \in q$, for all $q \in S^{\mathcal{B}}$. The locations $q \in S^{\mathcal{B}}$ are assigned with the conjunction of all the clock constraints of $s \in q$.

(xi) The invariant clock constraint for a transition relation $(q_1, q_2) \in \rightarrow_{reca}^{\mathcal{B}}$ and $(s_1, s_2) \in \rightarrow_{reca}^{\mathcal{A}}$, $\delta^{\mathcal{B}}((q_1, q_2)) = \bigwedge \delta^{\mathcal{A}}((s_1, s_2))$ with $(s_1, s_2) \in (q_1, q_2)$, for all $(q_1, q_2) \in \rightarrow_{reca}^{\mathcal{B}} \subseteq S^{\mathcal{B}} \times S^{\mathcal{B}}$. The transitions $(q_1, q_2) \in \rightarrow_{reca}^{\mathcal{B}} \subseteq S^{\mathcal{B}} \times S^{\mathcal{B}}$ are assigned with the conjunction of all the clock constraints of $(s_1, s_2) \in (q_1, q_2)$.

(xii) The accepting locations $F^{\mathcal{B}} \subseteq S^{\mathcal{B}}$.

Now we recall the definition of an equivalence relation between clock valuations based on the event-recording clocks and event-prediction clocks. An equivalence class of a clock valuation is called a *clock region* (of $\mathcal{A}$). For a clock valuation ν, $[\nu]$ denotes the clock region that contains ν. The set of clock regions of $\mathcal{A}$ is denoted by $\mathsf{REG}(\mathcal{A})$. Let $q \in Proc$. Then, we obtain a notion of of equivalence $\sim$ between two such valuations.

Definition 14. *Two clock valuations ν, ν' are in the same region, denoted $\nu \sim \nu'$, for a DECA $\mathcal{A} = (\Sigma, C, S, s_0, \rightarrow_{reca}, M, \gamma, \delta, F, \phi)$ iff the following conditions are considered:*

(i) ν and ν' agree on which clocks have the undefined value $\perp$. Those clocks are called undefined. The set of clocks undefined in the clock valuation ν is denoted $\mathsf{UND}(\nu)$. The other clocks are called active. The set of clocks active in the clock valuation ν is denoted $\mathsf{ACT}(\nu)$.

(ii) ν *and* $\nu^{'}$ *agree on the integer part of all active clocks that are at most* K*, where* K *is the biggest constant appearing in the clock constraints of the transitions of* $\mathcal{A}$*:*

(a) For each $x \in$ ***ACT***(ν)*, if* $(\nu)(x) \leq K$ *or* $(\nu^{'})(x) \leq K$ *implies* $\lfloor \nu(x) \rfloor$ = $\lfloor \nu^{'}(x) \rfloor$.

(iii) ν *and* $\nu^{'}$ *agree on the fractional part of all active clocks that are at most* K*:*

(a) For an event-prediction clock y*,* $fract(\nu(y)) = \nu(y) - \lfloor \nu(y) \rfloor$ *and for an event-recording clock* x*,* $fract(\nu(x)) = \nu(x) - \lfloor \nu(x) \rfloor$*. For all* z_1*,* $z_2 \in$ ***ACT***(ν) *with* $\nu(z_1) \leq K$ *and* $\nu(z_2) \leq K$ *:*

(1) $fract(\nu(z_1))$ *= 0 iff* $fract(\nu^{'}(z_1))$ *= 0,*

(2) $fract(\nu(z_1)) \leq fract(\nu(z_2))$ *iff* $fract(\nu^{'}(z_1)) \leq fract(\nu^{'}(z_2))$

A *clock region* is an equivalence class of clock valuations induced by $\sim$. Each region can be characterized by the finite set of constraints it satisfies. Now, we say that two clock valuations ν and $\nu^{'}$ over *C* are equivalent, denoted $\nu \sim \nu^{'}$ if they are equivalent when restricted to each process, $\nu_p \sim_p \nu^{'}_p$ for all $p \in$ *Proc*. The number of clock regions is finite: $2^{O(|C| \cdot \log(|C| \cdot K))}$, where $|C|$ the number of clocks, $|S|$ is the number of locations and K be the largest integer constant that appears in the clock constraints in $\mathcal{A}$.

The equivalence relation $\sim_p$ for $p \in$ *Proc* over the clock valuations can be extended over the set of possible configurations of the DECA. Thus, two configurations are equivalent, i.e. $(s_1, \nu_1) \sim_p (s_2, \nu_2)$ iff $s_1 = s_2$ and $\nu_1 \sim_p \nu_2$. The resulting equivalent classes of configurations of an DECA $\mathcal{A}$, are captured by the so-called *region automaton*. The *region automaton* corresponding to a given DECA $\mathcal{A}$ is defined as a finite state automaton with the state space $S \times \{\mathsf{Reg}(\mathcal{A})\}$, where $\mathsf{Reg}(\mathcal{A})$ is the clock region. The number of locations in the region automaton is $|S| \cdot 2^{O(|C| \cdot \log(|C| \cdot K))}$ where $|C|$ the number of clocks, $|S|$ is the number of locations and K be the largest integer constant that appears in the clock constraints in $\mathcal{A}$.

Proposition 1. *For every DECA* $\mathcal{A}$*,* ***Det***$(\mathcal{A})$ *accepts the same language:* $\mathcal{L}^{+}(\mathcal{A}, \tau)$ = $\mathcal{L}^{+}(\mathbf{Det}(\mathcal{A}), \tau)$ *then for all* $\mathcal{A}$*,* ***Det***$(\mathcal{A})$ *is deterministic.*

Proof. Let ρ be an ISS. We show that there exists an accepting run θ for ρ at time t over $\mathcal{A}$ for a rate τ iff there exists an accepting run $\theta^{'}$ for the same τ and ρ over $\mathsf{Det}(\mathcal{A})$. We are using induction over the length of the runs. Let $\theta = (s_0, I_0), (s_1, I_1), (s_2, I_2), \ldots (s_n, I_n)$ be the run for ρ at time t and on the *rate* τ over $\mathcal{A}$, where *s* are an alternation of transitions and locations $\delta_1, s_1, \delta_2, s_2, \cdots$, (where $\delta_i = (s_{i-1}, s_i) \in \rightarrow_{deca}$), and I_i is an alternating sequence of interval. By induction hypothesis this is possible only if there exists a run $\theta' = (s^{'}_0, I_0), (s^{'}_1, I_1), (s^{'}_2, I_2), \ldots (s^{'}_n, I_n)$ for ρ at time t and on the *rate* τ over $\mathsf{Det}(\mathcal{A})$, where we can safely assume due to the construction that, for each $1 \leq i \leq n$, $s_i \in s^{'}_i$. Firstly, we say that $\mathcal{A}$ accepts an ISS ρ at t with *rate* τ, if there is a run θ for ρ that visits a monitored location at t. This time t will used to reset the clock. Secondly the clock valuation depends on the ISS ρ, on the reference time of evaluation t, and on the *rate* τ. It is easy to see that for each $1 \leq i \leq n$ the clock valuation assigns a (non-standard) positive real, or undefined, to each clock variable $\nu(\rho, t, \tau)_{\mathcal{A}}$ = $\nu^{'}(\rho, t, \tau)_{\mathsf{Det}(\mathcal{A})}$. Thirdly, since $s_{n-1} \in s^{'}_{n-1}$ and there exists a transition δ_i from

(s_{n-1}, s_n) in $\mathcal{A}$, there exists $s_n^{'}$ such that $s_n \in s_n^{'}$ and there exists a transition $\delta_n^{'}$ from $(s_{n-1}^{'}, s_n^{'})$ where the clock constraints are satisfied by the valuation $\nu^{'}(\rho, t, \tau)_{\mathsf{Det}(\mathcal{A})}$. Hence we can adjunct $s_{n-1}^{'}$, $\delta_n^{'}$, $s_n^{'}$ to the run over $\mathsf{Det}(\mathcal{A})$. It follows immediately that $\mathcal{A}$ accepts the set of ISS ρ at time t and on the *rate* τ, then the language $\mathcal{L}^+(\mathcal{A}, \tau)$ is defined as the set of ISS ρ of accepting runs of $\mathcal{A}$ with regard to τ, and the language $\mathcal{L}^+(\mathsf{Det}(\mathcal{A}), \tau)$ is defined as the set of ISS ρ of accepting runs of $\mathsf{Det}(\mathcal{A})$ with regard to τ. We have that $\mathcal{L}^+(\mathcal{A}, \tau) = \mathcal{L}^+(\mathsf{Det}(\mathcal{A}), \tau)$. The other direction of the implication is proved using a similar argument.

Proposition 2. *For every DECA $\mathcal{A}$, $\mathsf{Det}(\mathcal{A})$ accepts the same existential language $\mathcal{L}_\exists(\mathsf{Det}(\mathcal{A}), P)$ = $\mathcal{L}_\exists(\mathcal{A}, P)$ for any $P \subseteq$ Proc then for all $\mathcal{A}$, $\mathsf{Det}(\mathcal{A})$ is deterministic.*

Proof. We can derive this easily using the Proposition 1.

Proposition 3. *For every DECA $\mathcal{A}$, $\mathsf{Det}(\mathcal{A})$ accepts the same universal language $\mathcal{L}_\forall(\mathsf{Det}(\mathcal{A})$, $P)$ = $\mathcal{L}_\forall(\mathcal{A}, P)$ for any $P \subseteq$ Proc then for all $\mathcal{A}$, $\mathsf{Det}(\mathcal{A})$ is deterministic.*

Proof. We can derive this easily using the Proposition 1.

The theorems are valid for the finite version, but also for the infinite ones, e.g. for Büchi automata, which are determinized to a parity automaton [12]. The construction of a $\mathsf{Det}(\mathcal{A})$ for the infinite version of a DECA $\mathcal{A}$ can be constructed using the same construction of deterministic for finite version.

Proposition 4. *For every DECA $\mathcal{A}$, $\mathsf{Det}(\mathcal{A})$ accepts the same language: $\mathcal{L}^+(\mathcal{A}, \tau)$ = $\mathcal{L}^+(\mathsf{Det}(\mathcal{A}), \tau)$ then for all $\mathcal{A}$, $\mathsf{Det}(\mathcal{A})$ is deterministic.*

Proof. Let ρ be an ISS. We show that there exists an accepting run θ for ρ at time t over $\mathcal{A}$ for a rate τ iff there exists an accepting run $\theta^{'}$ for the same τ and ρ over $\mathsf{Det}(\mathcal{A})$. We are using induction over the length of the runs. Let θ = $(s_0, I_0), (s_1, I_1), (s_2, I_2), \ldots$ be the run for ρ at time t and on the *rate* τ over $\mathcal{A}$, where *s* are an alternation of transitions and locations $\delta_1, s_1, \delta_2, s_2, \cdots$, (where $\delta_i = (s_{i-1}, s_i) \in \rightarrow_{deca}$), and I_i is an alternating sequence of interval. By induction hypothesis this is possible only if there exists a run θ' = $(s_0^{'}, I_0), (s_1^{'}, I_1), (s_2^{'}, I_2), \ldots$ for ρ at time t and on the *rate* τ over $\mathsf{Det}(\mathcal{A})$, where we can safely assume due to the construction that, for $i \geq 1$, $s_i \in s_i^{'}$. Firstly, we say that $\mathcal{A}$ accepts an ISS ρ at t with *rate* τ, if there is a run θ for ρ that visits infinitely often a monitored location at t. This time t will used to reset the clock. Secondly the clock valuation depends on the ISS ρ, on the reference time of evaluation t, and on the *rate* τ. It is easy to see that for $i \geq 1$ the clock valuation assigns a (non-standard) positive real, or undefined, to each clock variable $\nu(\rho, t, \tau)_{\mathcal{A}}$ = $\nu^{'}(\rho, t, \tau)_{\mathsf{Det}(\mathcal{A})}$. Thirdly, since $s_{i-1} \in s_{i-1}^{'}$ and there exists a transition δ_i from (s_{i-1}, s_i) in $\mathcal{A}$, there exists $s_i^{'}$ such that $s_i \in s_i^{'}$ and there exists a transition $\delta_i^{'}$ from $(s_{i-1}^{'}, s_i^{'})$ where the clock constraints are satisfied by the valuation $\nu^{'}(\rho, t, \tau)_{\mathsf{Det}(\mathcal{A})}$. Hence we can adjunct $s_{i-1}^{'}$, $\delta_i^{'}$, $s_i^{'}$ to the run over $\mathsf{Det}(\mathcal{A})$. It follows immediately that $\mathcal{A}$ accepts the set of ISS ρ at time t and on the *rate* τ, then the language $\mathcal{L}^+(\mathcal{A}, \tau)$ is defined as the set of ISS

ρ of accepting runs of $\mathcal{A}$ with regard to τ, and the language $\mathcal{L}^+(\mathsf{Det}(\mathcal{A}), \tau)$ is defined as the set of ISS ρ of accepting runs of $\mathsf{Det}(\mathcal{A})$ with regard to τ. We have that $\mathcal{L}^+(\mathcal{A}, \tau) = \mathcal{L}^+(\mathsf{Det}(\mathcal{A}), \tau)$. The other direction of the implication is proved using a similar argument.

Theorem 11. *DECA is closed under union operation.*

(Construction 7) Let $\mathcal{A} = (\Sigma^{\mathcal{A}}, S^{\mathcal{A}}, s_0^{\mathcal{A}}, \rightarrow_{deca}^{\mathcal{A}}, C^{\mathcal{A}}, \gamma^{\mathcal{A}}, \delta^{\mathcal{A}}, M^{\mathcal{A}}, F^{\mathcal{A}}, \pi^{\mathcal{A}})$ and $\mathcal{B} = (\Sigma^{\mathcal{B}}, S^{\mathcal{B}}, s_0^{\mathcal{B}}, \rightarrow_{deca}^{\mathcal{B}}, C^{\mathcal{B}}, \gamma^{\mathcal{B}}, \delta^{\mathcal{B}}, M^{\mathcal{B}}, F^{\mathcal{B}}, \pi^{\mathcal{B}})$ be two DECA. Without loss of generality we assume that the sets of clocks $C^{\mathcal{A}}$ and $C^{\mathcal{B}}$ (and respectively the sets of locations $S^{\mathcal{A}}$ and $S^{\mathcal{B}}$) are all pairwise disjoint.

Union Let $\mathcal{C} = (\Sigma^{\mathcal{C}}, S^{\mathcal{C}}, s_0^{\mathcal{C}}, \rightarrow_{deca}^{\mathcal{C}}, C^{\mathcal{C}}, \gamma^{\mathcal{C}}, \delta^{\mathcal{C}}, M^{\mathcal{C}}, F^{\mathcal{C}}, \pi^{\mathcal{C}})$ be the DECA defined as follows:

(i) The alphabet in $\Sigma^{\mathcal{C}}$ are as in $\mathcal{A}$, that is $\Sigma^{\mathcal{C}} = \Sigma^{\mathcal{A}} = \Sigma^{\mathcal{B}}$,

(ii) The clocks of $\mathcal{C}$ are the disjoint union of $\mathcal{A}$ and $\mathcal{B}$, that is $C^{\mathcal{C}} = C^{\mathcal{A}} \cup C^{\mathcal{B}}$,

(iii) The locations of $\mathcal{C}$ are tuples (s, μ) such that either :

  (a) $s \in S^{\mathcal{A}}$, $\mu \in (\Sigma^{\mathcal{C}} \cup \delta^{\mathcal{C}})$ and for all $\varsigma \in (\Sigma^{\mathcal{A}} \cup \delta^{\mathcal{A}})$, $\varsigma \in \mu$ iff $\varsigma \in (\gamma^{\mathcal{A}}(s) \cup \delta^{\mathcal{A}}(s))$, which will ensure the coherence of the labelling of (s, μ) with the labelling of s in $\mathcal{A}$,

  (b) or $s \in S^{\mathcal{B}}$, $\mu \in (\Sigma^{\mathcal{C}} \cup \delta^{\mathcal{C}})$ and for all $\varsigma \in (\Sigma^{\mathcal{B}} \cup \delta^{\mathcal{B}})$, $\varsigma \in \mu$ iff $\varsigma \in (\gamma^{\mathcal{A}}(s) \cup \delta^{\mathcal{A}}(s))$, which will ensure the coherence of the labelling of (s, μ) with the labelling s in $\mathcal{B}$,

(iv) The starting location of $\mathcal{C}$ is the following $s_0^{\mathcal{C}} = \{(s, \mu) \in S^{\mathcal{C}} \mid s = s_0^{\mathcal{A}} \textit{ or } s = s_0^{\mathcal{B}})\}$,

(v) The subset of monitored locations of $\mathcal{C}$ is the following set: $M^{\mathcal{C}} = \{(s, \mu) \in S^{\mathcal{C}} \mid s \in M^{\mathcal{A}} \textit{ or } s \in M^{\mathcal{B}}\}$,

(vi) The label of the location (s, μ) is simply the set of symbols $\xi : \gamma^{\mathcal{C}}((s, \mu)) = \xi$, where $\xi \in \Sigma^{\mathcal{C}}$, for every $(s, \mu) \in S^{\mathcal{C}}$,

(vii) The clock constraints of the locations (s, μ) in $\mathcal{C}$ is the intersection of the clock constraints of the location (s_1, μ) in $\mathcal{A}$ and the location (s_2, μ) in $\mathcal{B}$, that is $\delta^{\mathcal{C}}(s, \mu) = \delta^{\mathcal{A}}(s_1, \mu) \cup \delta^{\mathcal{B}}(s_2, \mu)$,

(viii) The transition relation of $\mathcal{C}$ is the following subset of $S^{\mathcal{C}} \times S^{\mathcal{C}} : \rightarrow_{deca}^{\mathcal{C}} = \{[(s_1, \mu_1), (s_2, \mu_2)] \mid (s_1, s_2) \in \rightarrow_{deca}^{\mathcal{A}} \textit{ or } (s_1, s_2) \in \rightarrow_{deca}^{\mathcal{B}}\}$,

(ix) The accepting conditions for $\mathcal{C}$ is the union of the accepting condition for $\mathcal{A}$ and $\mathcal{B}$, that is $F^{\mathcal{C}} = \{(s, \mu) \mid s \in F^{\mathcal{A}} \textit{ or } s \in F^{\mathcal{B}}\}$.

(x) The function that maps each clock to a process in $\mathcal{C}$ are the union of $\mathcal{A}$ and $\mathcal{B}$, that is $\pi^{\mathcal{C}} = \pi^{\mathcal{A}} \cup \pi^{\mathcal{B}}$,

Theorem 12. *DECA is closed under intersection operation.*

(Construction 8) Let $\mathcal{A} = (\Sigma^{\mathcal{A}}, S^{\mathcal{A}}, s_0^{\mathcal{A}}, \rightarrow_{deca}^{\mathcal{A}}, C^{\mathcal{A}}, \gamma^{\mathcal{A}}, \delta^{\mathcal{A}}, M^{\mathcal{A}}, F^{\mathcal{A}}, \pi^{\mathcal{A}})$ and $\mathcal{B} = (\Sigma^{\mathcal{B}}, S^{\mathcal{B}}, s_0^{\mathcal{B}}, \rightarrow_{deca}^{\mathcal{B}}, C^{\mathcal{B}}, \gamma^{\mathcal{B}}, \delta^{\mathcal{B}}, M^{\mathcal{B}}, F^{\mathcal{B}}, \pi^{\mathcal{B}})$ be two DECA. Without loss of generality we assume that the sets of clocks $C^{\mathcal{A}}$ and $C^{\mathcal{B}}$ (and respectively the sets of locations $S^{\mathcal{A}}$ and $S^{\mathcal{B}}$) are all pairwise disjoint.

Intersection Let $\mathcal{C} = (\Sigma^{\mathcal{C}}, S^{\mathcal{C}}, s_0^{\mathcal{C}}, \rightarrow_{deca}^{\mathcal{C}}, C^{\mathcal{C}}, \gamma^{\mathcal{C}}, \delta^{\mathcal{C}}, M^{\mathcal{C}}, F^{\mathcal{C}}, \pi^{\mathcal{C}})$ be the DECA defined as follows:

(i) The alphabet in $\Sigma^{\mathcal{C}}$ are as in $\mathcal{A}$, that is $\Sigma^{\mathcal{C}} = \Sigma^{\mathcal{A}} = \Sigma^{\mathcal{B}}$,

(ii) The clocks of $\mathcal{C}$ are the union of $\mathcal{A}$ and $\mathcal{B}$, that is $C^{\mathcal{C}} = C^{\mathcal{A}} \cup C^{\mathcal{B}}$,

(iii) The set of locations of $\mathcal{C}$ are the tuples (s^a, s^b) such that $s^a \in S^{\mathcal{A}}$, $s^b \in S^{\mathcal{B}}$ and for all $\varsigma \in (\Sigma^{\mathcal{A}} \cup \delta^{\mathcal{A}}) \cap (\Sigma^{\mathcal{B}} \cup \delta^{\mathcal{B}})$, $\varsigma \in (\gamma^{\mathcal{A}}(s^a) \cup \delta^{\mathcal{A}}(s^a))$ iff $\varsigma \in (\gamma^{\mathcal{B}}(s^b) \cup \delta^{\mathcal{B}}(s^b))$,

(iv) The starting location of $\mathcal{C}$ is the following $s_0^{\mathcal{C}} = \{(s^a, s^b) \in S^{\mathcal{C}} \mid s^a = s_0^{\mathcal{A}}\ and\ s^b = s_0^{\mathcal{B}}\}$,

(v) The subset of monitored locations of $\mathcal{C}$ is the following set: $M^{\mathcal{C}} = \{(s^a, s^b) \in S^{\mathcal{C}} \mid s^a \in M^{\mathcal{A}}\ and\ s^b \in M^{\mathcal{B}}\}$,

(vi) The label locations (s^a, s^b) of $\mathcal{C}$ is the intersection of the label of s^a in $\mathcal{A}$ and the label of s^b in $\mathcal{B}$, that is $\gamma^{\mathcal{C}}((s^a, s^b)) = \gamma^{\mathcal{A}}(s^a) \wedge \gamma^{\mathcal{B}}(s^b)$, for every $(s^a, s^b) \in S^{\mathcal{C}}$,

(vii) The transition relation of $\mathcal{C}$ is the following set $S^{\mathcal{C}} \times S^{\mathcal{C}} : \rightarrow_{deca}^{\mathcal{C}} = \{[(s_1^a, s_1^b), (s_2^a, s_2^b)] \mid (s_1^a, s_2^a) \in \rightarrow_{deca}^{\mathcal{A}} \vee (s_1^a = s_2^a)\ and\ (s_1^b, s_2^b) \in \rightarrow_{deca}^{\mathcal{B}} \vee (s_1^b = s_2^b)\}$,

(viii) The clock constraints of the locations (s^a, s^b) of $\mathcal{C}$ is the intersection of the clock constraints of the location s^a in $\mathcal{A}$ and the location s^b in $\mathcal{B}$, that is $\delta^{\mathcal{C}}((s^a, s^b)) = \delta^{\mathcal{A}}(s^a) \wedge \delta^{\mathcal{B}}(s^b)$,

(ix) The accepting conditions for $\mathcal{C}$ is defined using a generalized Büchi condition : $F^{\mathcal{C}} = \{G_{\mathcal{A}}, G_{\mathcal{B}}\}$ with $G_{\mathcal{A}} = \{(s^a, s^b) \mid s^a \in F^{\mathcal{A}}\}$ and $G_{\mathcal{B}} = \{(s^a, s^b) \mid s^b \in F^{\mathcal{B}}\}$ and the reduction from a generalized Büchi automata to Büchi automata is : $F^{\mathcal{C}} = F^{\mathcal{A}} \times \{1\}$.

(x) The function that maps each clock to a process in $\mathcal{C}$ are the union of $\mathcal{A}$ and $\mathcal{B}$, that is $\pi^{\mathcal{C}} = \pi^{\mathcal{A}} \cup \pi^{\mathcal{B}}$,

Theorem 13. *DECA is closed under complementation operation.*

Complementation Given a *DECA* $\mathcal{A} = (\Sigma^{\mathcal{A}}, S^{\mathcal{A}}, s_0^{\mathcal{A}}, \rightarrow_{deca}^{\mathcal{A}}, C^{\mathcal{A}}, \gamma^{\mathcal{A}}, \delta^{\mathcal{A}}, M^{\mathcal{A}}, F^{\mathcal{A}}, \pi^{\mathcal{A}})$, such that: DECA are closed under complementation since they are determinizable. In the Construction 6, we show that for a automata $\mathcal{A}$, we can have a deterministic automata Det($\mathcal{A}$). In that case we can construct as following : (*i*) We construct a automata Det($\mathcal{A}$) as in Construction 6. (*ii*) A deterministic automaton $\mathcal{A}$ has a unique run corresponding to a given ISS ρ (see Definition 14). (*iii*) Hence replacing F by $S \backslash F$ as the final locations for the acceptance of $\mathcal{A}$ would result in the automaton accepting the complement of $\mathcal{L}(\mathcal{A}, \tau)^c$.

Theorem 14. *The τ-wise emptiness problem for DECA is PSPACE-complete.*

Proof. Using region technique [2] and [14] showed that the emptiness problem for RECA is decidable. As we shown in this paper, a RECA is a DECA. The emptiness problem of DECA is decidable in $n \cdot 2^{n \cdot \log n}$, where n is the number of locations, m is the number of clocks, and c is the largest constant that appears in clock constraints, it follows that emptiness of a DECA$\mathcal{A}$ can be checked in PSPACE.

Theorem 15. *The τ-wise language inclusion problem for DECA is PSPACE-complete.*

Proof. Consider two DECA $\mathcal{A}$ and $\mathcal{B}$ with regard to $\tau \in$ *rates*, such that each automaton has at most n locations, let m be the number of clocks. Let c be the largest constant that appears in the clock constraints. To check whether $\mathcal{L}(\mathcal{A}, \tau) \subseteq \mathcal{L}(\mathcal{B}, \tau)$, we first determinize $\mathcal{B}$ to $\mathcal{B}^1$, after we complement $\mathcal{B}^1$ to $\mathcal{B}^2$. The automata $\mathcal{B}^2$ has $2^{n \cdot \log n}$ locations since it is a Büchi automaton and the integer constants that appear in the clock constraints of $\mathcal{B}^2$ are bounded by c. Let $\mathcal{D}$ be the intersection of $\mathcal{A}$ and $\mathcal{B}^2$. The *DECA* $\mathcal{D}$ has $n \cdot 2^{n \cdot \log n}$ locations, where the integer constants that appear in the clock constraints of $\mathcal{D}$ are also bounded by c.

From Theorems 14 and 15 and the Construction 7, 8 and 9, we have the next Theorem.

Theorem 16. *The τ-wise universality problem for DECA is PSPACE-complete.*

Proof. We can derive this easily using the Theorem 14 and 15

Theorem 17. *The existential emptiness and language inclusion problem for DECA are PSPACE-complete.*

Proof. We can derive this easily using the Theorem 14, 15 and 16.

Theorem 18. *The universal emptiness and language inclusion problem for DECA are PSPACE-complete.*

Proof. We can derive this easily using the Theorem 14, 15 and 16.

Corollary 1. *For all DECA $\mathcal{B}, Q \subseteq Proc, \mathcal{L}_\exists(\mathcal{B}, Q)$ is the language of a DECA.*

Construction 9 For the existential languages of DECA, we can eliminate a process q from a DECA while preserving the existential language of the remaining processes. We first complete and determinize automata appearing in the clocks of this process q. We then make their product with the main automaton. We then perform the region construction [3] on the clocks of q. Remember that the clocks are constrained to be 0 in the respective monitored location, i.e. when at least one original monitored location appears in this construction, and that prediction clocks run backwards so that it is the complement of their fractional part that participates in the region construction [3]. The region construction for prediction clocks is non-deterministic and is not a bisimulation quotient, unlike the one of TA, but preserves the language [14]. Note that the elimination of the clocks of one process only, allows independent evolution of the other clocks. The resulting automaton is still a DECA.

Corollary 2. *For all DECA $\mathcal{B}, q \in Proc, \mathcal{L}_\forall(\mathcal{B}, \{q\})$ is the language of a RECA.*

Construction 10 For the universal languages $\mathcal{L}_\forall(\mathcal{B}, Q)$ of a DECA $\mathcal{B}$, we complete and determinize the main automaton $\mathcal{B}$. Then we apply the region construction for independent clocks [1]. The automaton becomes non-deterministic, because each region has several successors, depending on τ. Note that the region construction for ECA was already non-deterministic. A region constraint is expressed as a conjunction $\bigwedge_{p \in Proc} \phi_p$. We label each region state by $\bigwedge_{p \in Q} \phi_p$. Then we determinize it again but we mark as final the locations where all members are final (which, in turn, means that one of their members is an original final state), to represent that the ISS must be accepted under *all* evolutions of time τ. The resulting automaton is a RECA.

Corollary 3. *For all* DECA $\mathcal{B}, q \in Proc, \mathcal{L}_\forall(\mathcal{B}, \{q\})$ *is the language of a* DECA*.*

Construction 11 For the universal languages of DECA, we can eliminate a process q from a DECA while preserving the universal language of the remaining processes. We first complete and determinize automata appearing in the clocks of this process q. We then make their product with the main automaton. We then perform the region construction [3] on the clocks of q. Remember that the clocks are constrained to be 0 in the respective monitored location, i.e. when at least one original monitored location appears in this construction, and that prediction clocks run backwards so that it is the complement of their fractional part that participates in the region construction [3]. The region construction for prediction clocks is non-deterministic and is not a bisimulation quotient, unlike the one of TA, but preserves the language [14]. Note that the elimination of the clocks of one process only, allows independent evolution of the other clocks. The resulting automaton is still a DECA.

Corollary 4. *For all* DECA $\mathcal{B}, Q \subseteq Proc, \mathcal{L}_\exists(\mathcal{B}, Q)$ *is the language of a* RECA*.*

Construction 12 For the existential languages $\mathcal{L}_\exists(\mathcal{B}, Q)$ of a DECA $\mathcal{B}$, we complete and determinize the main automaton $\mathcal{B}$. Then we apply the region construction for independent clocks [1]. The automaton becomes non-deterministic, because each region has several successors, depending on τ. Note that the region construction for ECA was already non-deterministic. A region constraint is expressed as a conjunction $\bigwedge_{p \in Proc} \phi_p$. We label each region state by $\bigwedge_{p \in Q} \phi_p$. Then we determinize it again but we mark as final the locations where all members are final (which, in turn, means that one of their members is an original final state), to represent that the ISS must be accepted under *all* evolutions of time τ. The resulting automaton is a RECA.

In contrast, the universal language of DTA and icTA is undecidable [1].

6 Recursive Distributed Event Clocks Temporal Logic

The aim of this section is to construct a fully decidable distributed real-time logic to specify requirements on RTDS. (Recursive) Distributed Event Clock Temporal Logic (DECTL) extend the (Recursive) Event Clock Temporal Logic (EventClockTL) [15, 9] with distributed (a.k.a. independent) clocks. As in Section 3.2, we assume a set of processes *Proc*. The clocks of each process will evolve according to its local time given by a Rate τ. DECTL is based on LTL, and adds two local real-time modalities. The recording modality $\lhd_I^q \phi$ means that ϕ was true last time in the interval I according to the local time of q. Symmetrically, the predicting modality $\rhd_I^q \phi$ says the ϕ will occur within I according to the local time of q. If we have only one process, we find back EventClockTL [15].

We could construct similarly a more expressive logic that allows to observe not only the last ϕ, but also the last but one, and more generally the last but n ϕ [11]. This logic is still translatable in DECA. We could also extend the known expressive equivalence of EventClockTL and MITL+Past [9] to construct a version of MITL with independent clocks. Lastly, independent clocks can also be introduced in a linear μ-calculus, to increase expressiveness by counting. We do not present these logics here by lack of space.

Definition 15. *The* formulas *of* DECTL *are defined by the grammar:*

$$\phi ::= \ true \mid p \in \mathbb{P} \mid \ \rhd_I^q \ \phi \mid \ \lhd_I^q \ \phi \mid \phi_1 \ \wedge \ \phi_2 \mid \neg\phi \mid \phi_1 \, \mathcal{U} \, \phi_2 \mid \phi_1 \, \mathcal{S} \, \phi_2$$

where p is a propositional symbol, $I \in \mathcal{I}_{\mathbb{N}}$ is an interval and $q \in Proc$. We can now define how to evaluate the truth value of a DECTL *formula along an* ISS *ρ and a Rate τ, noted $(\rho, t) \models_\tau \phi_1$. We omit τ below.*

$(\rho, t) \models p$ *iff* $p \in \rho(t)$
$(\rho, t) \models \neg\, \phi$ *iff* $(\rho, t) \not\models \phi$
$(\rho, t) \models \phi_1 \wedge \phi_2$ *iff* $(\rho, t) \models \phi_1$ *and* $(\rho, t) \models \phi_2$
$(\rho, t) \models \phi_1 \, \mathcal{U} \, \phi_2$ *iff* $\exists\, t' > t.\, (\rho, t') \models \phi_2$ *and* $\forall\, t'' \in (t, t')$, $(\rho, t'') \models \phi_1$
$(\rho, t) \models \phi_1 \, \mathcal{S} \, \phi_2$ *iff* $\exists\, t' < t.\, (\rho, t') \models \phi_2$ *and* $\forall\, t'' \in (t', t)$, $(\rho, t'') \models \phi_1$
$(\rho, t) \models \lhd_I^q \phi$ *iff* $\exists\, t' < t.\, \tau_q(t) - \tau_q(t') \in I \wedge (\rho, t') \models \phi$
and $\forall\, t'' < t.\, \tau_q(t) - \tau_q(t'') < I$, $(\rho, t'') \not\models \phi$
$(\rho, t) \models \rhd_I^q \phi$ *iff* $\exists t' > t. \tau_q(t') - \tau_q(t) \in I \wedge (\rho, t') \models \phi$
and $\forall\, t'' > t.\, \tau_q(t'') - \tau_q(t) < I$, $(\rho, t'') \not\models \phi$

Example 2. The formula $\neg(\mathcal{F}b \wedge \neg \rhd_{\leq 1}^q b)$, where $\mathcal{F}b = \mathit{true}\, \mathcal{U} b$ says that the first b, if any, must occur within 1 second, as measured by q. It holds on the automation of Fig.1. However, the formula measured by p, $\neg(\mathcal{F}b \wedge \neg \rhd_{\leq 1}^p b)$, does not hold.

6.1 Expressiveness between DECTL and DECA

In this section we show that DECA are sufficiently expressive to define all DECTL properties. We can translate any DECTL formula ϕ into a DECA automaton $\mathcal{A}_\phi$ that accepts the pairs (ρ, t), such that $(\rho, t) \models_\tau \phi$, for all τ by a tableau construction. The translation is done level by level, where the level of a formula is the nesting depth of real-time modalities. A formula $\rhd_I^q \phi$ is translated as constraint $x_{\mathcal{A}_\phi}^q \in I$. The formula ϕ is recursively translated in a tableau automaton $\mathcal{A}_\phi$ where the monitored states are the states containing ϕ.

Definition 16. *The level of a* DECTL *formulas ϕ_1, ϕ_2, ϕ_3 denoted by $\mathcal{LF}$, is a recursive function that satisfying the following :*

$\mathcal{LF}(p) = 0$
$\mathcal{LF}(\phi_1 \vee \phi_2) = Max(\mathcal{LF}(\phi_1) \cup \mathcal{LF}(\phi_2))$
$\mathcal{LF}(\neg\, \phi_1) = \mathcal{LF}(\phi_1)$
$\mathcal{LF}(\phi_1 \, \mathcal{U} \, \phi_2) = Max(\mathcal{LF}(\phi_1) \cup \mathcal{LF}(\phi_2))$
$\mathcal{LF}(\phi_1 \, \mathcal{S} \, \phi_2) = Max(\mathcal{LF}(\phi_1) \cup \mathcal{LF}(\phi_2))$
$\mathcal{LF}(\rhd_I^q \, \phi_3) = 1 + \mathcal{LF}(\, \phi_3)$
$\mathcal{LF}(\lhd_I^q \, \phi_3) = 1 + \mathcal{LF}(\phi_3)$

A formula ϕ is of level i, if $\mathcal{LF}(\phi) = i$. Recursively, we can define that the formula ϕ_3 is a level k where $0 \leq k < i$ and ϕ_1, ϕ_2 is a level j where $0 \leq j \leq i$.

Definition 17. *The closure set of a* DECTL *formula ϕ_1, ϕ_2, ϕ_3 denoted by $\mathcal{CF}$, is a recursive function that satisfying the following:*

$\mathcal{CF}(p) = \{p\}$
$\mathcal{CF}(\phi_1 \vee \phi_2) = \mathcal{CF}(\phi_1) \cup \mathcal{CF}(\phi_2) \cup \{ \phi_1 \vee \phi_2 \}$
$\mathcal{CF}(\neg \phi_1) = \mathcal{CF}(\phi_1)$
$\mathcal{CF}(\phi_1 \,\mathcal{U}\, \phi_2) = \mathcal{CF}(\phi_1) \cup \mathcal{CF}(\phi_2) \cup \{ \phi_1 \,\mathcal{U}\, \phi_2 \}$
$\mathcal{CF}(\phi_1 \,\mathcal{S}\, \phi_2) = \mathcal{CF}(\phi_1) \cup \mathcal{CF}(\phi_2) \cup \{ \phi_1 \,\mathcal{S}\, \phi_2 \}$
$\mathcal{CF}(\rhd_I^q \phi_3) = \bigcup_{q \in Proc} \rhd_I^q \phi_3$
$\mathcal{CF}(\lhd_I^q \phi_3) = \bigcup_{q \in Proc} \lhd_I^q \phi_3$
$\mathcal{CF}^c(\phi_1)$ is the set $\mathcal{CF}(\phi_1)$ closed by negation.

Theorem 19. *For every* DECTL *formula ϕ, we can construct a* DECA *$\mathcal{A}_\phi$, that accepts the pairs (ρ, t), where ρ is defined on the set of propositions appearing in ϕ and a time $t \in \mathbb{R}^+$, such that $(\rho, t) \models_\tau \phi$.*

Proof. Let $\mathcal{LF}(\phi) = 0$. We define a transition structure $\mathcal{B} = (Q, q_0, \rightarrow_{ts}, Q_F)$ that checks the semantics of the operators and propositions of level 0 formula. For the other levels i, such that $i > 0$, we will transform the transition structure into a DECA. Let DECTL formula ϕ, ϕ_1, ϕ_2, we define $\mathcal{B}$ as follows:

(i) The state S is the set of pairs (s, φ), where $s \in 2^{\mathcal{CF}^c(\phi)}$ with $\top \in s$ and $\varphi \in \{open, sing\}$(indicating if the control can stay in the state for an open interval of time or just a singular interval of time) and the following properties are verified:
  (a) For all $\phi_1 \in \mathcal{CF}^c(\phi)$: $\phi_1 \in s$ iff $\neg \phi_1 \notin s$.
  (b) For all $(\phi_1 \vee \phi_2) \in \mathcal{CF}^c(\phi)$: $\phi_1 \vee \phi_2 \in s$ iff $\phi_1 \in s$ or $\phi_2 \in s$.
  (c) For all $(\phi_1 \,\mathcal{U}\, \phi_2) \in \mathcal{CF}^c(\phi)$:
    (1) If $\phi_2 \in s$ and $\varphi = open$ then $\phi_1 \,\mathcal{U}\, \phi_2 \in s$.
    (2) If $\phi_1 \,\mathcal{U}\, \phi_2 \in s$ and $\varphi = open$ then $\phi_1 \in s$ or $\phi_2 \in s$.
  (d) For all $(\phi_1 \,\mathcal{S}\, \phi_2) \in \mathcal{CF}^c(\phi)$:
    (1) If $\phi_2 \in s$ and $\varphi = open$ then $\phi_1 \,\mathcal{S}\, \phi_2 \in s$.
    (2) If $\phi_1 \,\mathcal{S}\, \phi_2 \in s$ and $\varphi = open$ then $\phi_1 \in s$ or $\phi_2 \in s$.

(ii) The initial state is the subset of pairs $(s, \varphi) \in Q$, such that $\varphi = sing$ and does not exists $\phi_1 \,\mathcal{S}\, \phi_2 \in \mathcal{CF}^c(\phi)$ and $\phi_1 \,\mathcal{S}\, \phi_2 \in s$. That initial state is singular and it does not contains a since formula in positive form.

(iii) The transition relation $\rightarrow_{ts}$ is subset $[(s_1, \varphi_1), (s_2, \varphi_2)]$ of $Q \times Q$ that respects the following restrictions:
  (a) $\varphi_1 = open$ and $\varphi_2 = sing$ or $\varphi_1 = sing$ and $\varphi_2 = open$.
  (b) The following rules express how until formulas are transferred form one state to the next of the transition structure:
    (1) $\phi_1 \,\mathcal{U}\, \phi_2 \in s_1 \wedge \varphi_1 = sing$ iff $\phi_1 \,\mathcal{U}\, \phi_2 \in s_2$.
    (2) $\phi_1 \,\mathcal{U}\, \phi_2 \in s_1 \wedge \varphi_1 = open \wedge \phi_2 \notin s_1$, implies $(\phi_1 \,\mathcal{U}\, \phi_2 \in s_2 \wedge \phi_1 \in s_2) \vee \phi_2 \in s_2$.
    (3) $\phi_1 \in s_1 \wedge \varphi_1 = open \wedge (\phi_1 \in s_2 \vee (\phi_2 \in s_2 \wedge \phi_1 \,\mathcal{U}\, \phi_2 \in s_2))$ implies $\phi_1 \,\mathcal{U}\, \phi_2 \in s_1$.
  (c) The following are for the since formulas:
    (1) $\phi_1 \,\mathcal{S}\, \phi_2 \in s_2 \wedge \varphi_2 = sing$ iff $\phi_1 \,\mathcal{S}\, \phi_2 \in s_1$.

(2) $\phi_1 \mathcal{S} \phi_2 \in s_2 \wedge \varphi_2$ = *open* $\wedge\ \phi_2 \notin s_2$ implies $\phi_2 \in s_1 \vee (\phi_1 \in s_1 \wedge (\phi_1 \mathcal{S} \phi_2) \in s_1)$.
(3) $\phi_1 \in s_2 \wedge \varphi_2$ = *open* $\wedge\ (\phi_2 \in s_1 \vee (\phi_1 \mathcal{S} \phi_2) \in s_1)$ implies $\phi_1 \mathcal{S} \phi_2 \in s_2$.

(iv) We use a generalized Büchi acceptance condition. For each formula $\phi_1 \mathcal{U} \phi_2 \in \mathcal{CF}^c(\phi)$, there is a set $Q_{F_{\phi_1 \mathcal{U} \phi_2}} = \{(s, \varphi) \mid \phi_1 \mathcal{U} \phi_2 \notin s \vee \phi_2 \in s\}$.

Now, we will transforms the transition structure $\mathcal{B}$ into a DECA $\mathcal{A}_\phi$. We construct $\mathcal{A}_\phi$ = $(\Sigma^{\mathcal{A}_\phi}, C^{\mathcal{A}_\phi}, S^{\mathcal{A}_\phi}, s_0^{\mathcal{A}_\phi}, M^{\mathcal{A}_\phi}, \rightarrow_{deca}^{\mathcal{A}_\phi}, \gamma^{\mathcal{A}_\phi}, \delta^{\mathcal{A}_\phi}, F^{\mathcal{A}_\phi}, \pi^{\mathcal{A}_\phi})$ as follows :

(i) The set of symbols used by $\mathcal{A}_\phi$ is the set of propositional symbols that appear in the formula ϕ, $\Sigma^{\mathcal{A}_\phi} = \{p \mid p \in \mathcal{CF}^c(\phi)\}$,
(ii) The set of clocks used by $\mathcal{A}_\phi$ is the set of clocks that appear in the formula ϕ, $C^{\mathcal{A}_\phi} = \{\rhd_I^q \cup \lhd_I^q \mid q \in Proc\ and\ I\ is\ an\ interval\}$,
(iii) The set of locations $S^{\mathcal{A}_\phi}$ is the set of pairs $((s, \varphi), \varsigma)$ such that:
(a) $(s, \varphi) \in S^{\mathcal{A}_\phi}$.
(b) ς is a label that is open iff φ = *open*.
(c) The labeling is propositionally consistent with the formula in s: for all proposition $p \in \mathbb{P}$: $p \in \varsigma$ iff $p \in s$,
(iv) The initial location $s_0^{\mathcal{A}_\phi}$ is the subset of locations $((s, \varphi), \varsigma) \in S^{\mathcal{A}_\phi}$ such that $(s, \varphi) = s_0^{\mathcal{A}_\phi}$,
(v) The set $M^{\mathcal{A}_\phi}$ of monitored locations is the subset of locations $((s, \varphi), \varsigma) \in S^{\mathcal{A}_\phi}$ such that $\phi \in s$, that is the subset of locations where the formula ϕ is true,
(vi) The transition relation is the set of pairs $[((s_1, \varphi_1), \varsigma_1), ((s_2, \varphi_2), \varsigma_2)]$ with $((s_i, \varphi_i), \varsigma_i) \in S^{\mathcal{A}_\phi}$ for $i \in \{1, 2\}$, such that: $((s_1, \varphi_1), (s_2, \varphi_2)) \in \rightarrow_{deca}$,
(vii) The labeling function $\gamma^{\mathcal{A}_\phi}$ is defined as follows: $\gamma^{\mathcal{A}_\phi}(((s, \varphi), \varsigma)) = \varsigma$,
(viii) The clock constraints $\delta^{\mathcal{A}_\phi}$ is defined as follows: $\delta^{\mathcal{A}_\phi}(((s, \varphi), \varsigma)) = \varphi$,
(ix) We transfer in $\mathcal{A}_\phi$ the generalized Büchi acceptance condition of the transition structure $\mathcal{B}$: $F^{\mathcal{A}_\phi}$ is the set of sets of accepting locations $\{F_1, F_2, \cdots F_n\}$ where each F_i corresponds to a set of accepting states in *S* as follows: $F_i = \{((s, \varphi), \varsigma) \mid (s, \varphi) \in Q_F\}$,
(x)

The logic *ReDECTL* and *ReDECA* in fact use the same clocks:

Theorem 20. *$\nu(\rho, t, \tau, x_{\mathcal{A}}^q) \in I$ iff $(\rho, t) \models_\tau \lhd_I^q p$, where p is a proposition of process q such that $(\rho, t) \models_\tau p$ iff $\mathcal{A}$ accepts ρ at time t with τ.*

The construction is exponential in the size of the non-real time part of the formula, but linear in the real-time part. The test of emptiness is done by the region construction presented in Section 5, that is exponential in the real-time part but linear for the rest.

Corollary 5. *The Satisfiability and validity problems for DECTL are decidable.*

Proof. The satisfiability of a DECTL formula ϕ can be decided by constructing $\mathcal{A}_\phi$, the automata for ϕ and testing if $\mathcal{L}(\mathcal{A}_\phi) \neq \emptyset$. Similarly the validity of a DECTL formula ϕ can be decided by constructing $\mathcal{A}_{\neg\phi}$, the automaton for the negation of ϕ and testing if $\mathcal{L}(\mathcal{A}_{\neg\phi}) = \emptyset$.

Corollary 6. *Satisfiability and validity of* DECTL *are* PSPACE*-complete.*

Corollary 7. *The automaton $\mathcal{A}_\phi$ has $n \cdot 2^{O(n \cdot log\ c \cdot n)}$ locations, where n is the length of the formula ϕ (the number of propositions, modal operators and logical connectives) the number of clocks is n and c is the largest constant appearing in the constraints in $\mathcal{A}_\phi$.*

7 Conclusions

We have proposed the basis of a framework for analyzing distributed real-time systems through of the introduction of independent (or distributed) event clocks, inspired by DTA [10] and icTA [1]. In contrast to [1], we have given a real-time semantics, and thus we can specify real-time properties. We have defined DECA and proved they are fully decidable, and thus that their language inclusion problem is PSPACE-complete, as for classical automata. This give us an algorithm to verify real-time properties. Actually, since we can use zones without constraints between independent clocks, and since the number of regions is reduced wrt. ECA, we can even expect verification to be faster in practice. We also plan further work to gain more speed through partial-order techniques. In contrast, DTA [10] and icTA [1] have undecidable inclusion problems. We have also shown that the universal (timed) languages of DECA are decidable and regular, unlike the universal languages of icTA [1]. We propose the logic DECTL to specify real-time properties with distributed observers in linear temporal logic. We have shown that the problems of satisfiability, validity and model-checking are decidable for DECTL, more precisely PSPACE-complete, as for LTL - we cannot hope better.

References

1. S. Akshay, B. Bollig, P. Gastin, M. Mukund, and K. N. Kumar. Distributed timed automata with independently evolving clocks. In F. van Breugel and M. Chechik, editors, *CONCUR*, volume 5201 of *LNCS*, pages 82–97. Springer, 2008.
2. R. Alur and D. L. Dill. A theory of timed automata. *Theor. Comput. Sci.*, 126(2):183–235, 1994.
3. R. Alur, T. Feder, and T. A. Henzinger. The benefits of relaxing punctuality. *ACM*, 43(1):116–146, 1996.
4. R. Alur, L. Fix, and T. A. Henzinger. A determinizable class of timed automata. In *CAV*, volume 818 of *LNCS*, pages 1–13. Springer, 1994.
5. R. Alur and T. A. Henzinger. Logics and models of real time: A survey. In *REX Workshop*, volume 600 of *LNCS*, pages 74–106. Springer, 1991.
6. J. Bengtsson, B. Jonsson, J. Lilius, and W. Yi. Partial order reductions for timed systems. In D. Sangiorgi and R. de Simone, editors, *CONCUR*, volume 1466 of *LNCS*, pages 485–500. Springer, 1998.
7. B. Bérard, A. Petit, V. Diekert, and P. Gastin. Characterization of the expressive power of silent transitions in timed automata. *Fundam. Inform.*, 36(2-3):145–182, 1998.
8. M. De Wulf, L. Doyen, N. Markey, and J.-F. Raskin. Robustness and implementability of timed automata. In Y. Lakhnech and S. Yovine, editors, *FORMATS/FTRTFT*, volume 3253 of *LNCS*, pages 118–133. Springer, 2004.

9. T. A. Henzinger, J.-F. Raskin, and P.-Y. Schobbens. The regular real-time languages. In *ICALP*, volume 1443 of *LNCS*, pages 580–591. Springer, 1998.
10. P. Krishnan. Distributed timed automata. *Electr. Notes Theor. Comput. Sci.*, 28, 1999.
11. J. Ortiz, A. Legay, and P.-Y. Schobbens. Memory event clocks. In K. Chatterjee and T. A. Henzinger, editors, *FORMATS*, volume 6246 of *LNCS*, pages 198–212. Springer, 2010.
12. N. Piterman. From nondeterministic buchi and streett automata to deterministic parity automata. In *Proceedings of the 21st Annual IEEE Symposium on Logic in Computer Science*, pages 255–264, Washington, DC, USA, 2006. IEEE Computer Society.
13. A. Puri. Dynamical properties of timed automata. In A. P. Ravn and H. Rischel, editors, *FTRTFT*, volume 1486 of *LNCS*, pages 210–227. Springer, 1998.
14. J.-F. Raskin. *Logics, Automata and Classical Theories for Deciding Real Time*. Phd thesis, FUNDP University, Belgium, 1999.
15. J.-F. Raskin and P.-Y. Schobbens. State clock logic: A decidable real-time logic. In *HART*, volume 1201 of *LNCS*, pages 33–47. Springer, 1997.