Centre Fédéré en Vérification

Technical Report number 2003.14

Real-Time Model-Checking: Parameters Everywhere

Véronique Bruyère and Jean-François Raskin

This work was partially supported by a FRFC grant: 2.4530.02

http://www.ulb.ac.be/di/ssd/cfv

Real-Time Model-Checking: Parameters Everywhere

VÉRONIQUE BRUYÈRE
Institut d'Informatique
Université de Mons-Hainaut,
Le Pentagone, Avenue du Champ de Mars 6,
B-7000 Mons, Belgium,
Email: Veronique.Bruyere@umh.ac.be

JEAN-FRANÇOIS RASKIN
Département d'Informatique
Université Libre de Bruxelles,
Boulevard du Triomphe CP 212,
B-1050-Bruxelles, Belgium
Email: Jean-Francois.Raskin@ulb.ac.be

Abstract

In this paper[1], we study the model-checking and parameter synthesis problems of the logic TCTL over timed automata where parameters are allowed both in the model (timed automaton) and in the property (temporal formula). Our results are as follows. On the negative side, we show that the model-checking problem of TCTL extended with parameters is undecidable over timed automata with only one parametric clock. The undecidability result needs equality in the logic. On the positive side, we show that when equality is not allowed in the logic, the model-checking and the parameter synthesis problems become decidable. Our method is based on automata theoretic principles and an extension of our method to express duration of runs in timed automata using Presburger arithmetic.

1 Introduction

In this paper, we further investigate the model-checking problem of real-time formalisms with parameters. In recent works, parametric real-time model-checking problems have been studied by several authors. Alur et al study in [AHV93] the analysis of timed automata where clocks are compared to parameters. They showed that when only one clock is compared to parameters, the emptiness problem is decidable. But this problem becomes undecidable when three clocks are compared to parameters. Wang in [Wan95, WH97], Emerson et al in [ET99], Alur et al in [AETP99] and the authors of this paper in [BDR02] study the introduction of parameters in temporal logics. The model-checking problem for TCTL extended with parameters over timed automata (without parameters) is decidable. On the other hand, only a fragment of LTL extended with parameters is decidable.

Unfortunately, in all those previous works, the parameters are *only* in the model (expressed as a timed automaton) or *only* in the property (expressed as a temporal logic formula). Nevertheless, when expressing a temporal property of a parametric system, it is *natural* to refer in the temporal formula to the parameters used in the system. In this paper, we study the model-checking problem of the logic TCTL *extended with parameters* over the runs of a timed automaton with *one parametric clock*. To the best of our knowledge, this is the first work that study the model-checking and parameter synthesis problems with parameters both in the model and in the property.

Let us illustrate the kind of properties that we can express with a parametric temporal logic over a parametric timed automata. The automaton $\mathcal{A}$ of Figure 1 is a discrete timed automaton with two clocks x, y and two parameters θ_1 and θ_2. Here we explicitely model the elapse of time by self loops labelled by 1. Other transitions are instantaneous. State q_0 is labelled with atomic proposition σ and in all other states this proposition is false. The possible runs of this automaton starting at q_0 are as follows. The control instantaneously leaves q_0 and goes through q_1, q_2, q_3 to come back in q_0, the time spent in this cycle is constrained by the parameters θ_1 and θ_2. In fact, the control has to leave q_1 at most θ_1 time units after entering it and the control has stay exactly θ_2 time units in state q_2. To express properties of those behaviors, we use TCTL logic augmented with parameters. Let us consider the next three formulae for configuration $(q_0, 0, 0)$, i.e. the control is in state q_0 and clocks x and y have value 0:

(i) $\forall\Box(\sigma \rightarrow \forall\Diamond_{\leq\theta_3}\sigma)$

(ii) $\forall\theta_1\forall\theta_2\exists\theta_3 \cdot (\theta_3 = 2\theta_1 + 2) \wedge$

[1]Supported by the FRFC project "Centre Fédéré en Vérification" funded by the Belgian National Science Fundation (FNRS) under grant nr 2.4530.02

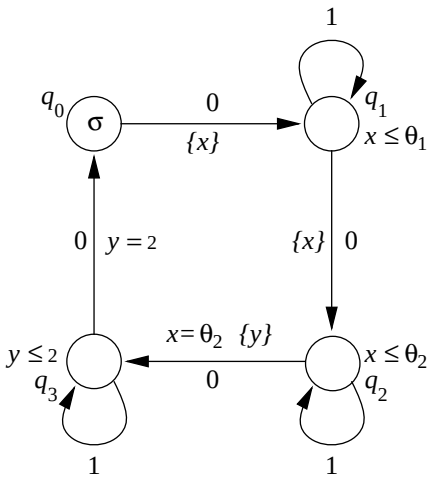

Figure 1. A parametric timed automaton

$$(\theta_2 \le \theta_1 \rightarrow \forall\Box(\sigma \rightarrow \forall\Diamond_{\le \theta_3}\sigma))$$

$$(iii)\ \ \forall\theta_1 \exists\theta_3 \cdot (\theta_3 = 2\theta_1 + 2) \wedge (\theta_1 \ge 5 \rightarrow \forall\Box(\sigma \rightarrow \forall\Diamond_{< \theta_3}\sigma))$$

The *parameter synthesis problem* associated to formula (i), asks for which values of θ_1, θ_2 and θ_3, the formula is true at configuration $(q_0, 0, 0)$. By observing the model and the formula, we can deduce the following constraint on the parameters: $\theta_3 \ge \theta_1 + \theta_2 + 2$. This means that any cycle throught the four states has duration bounded by $\theta_1 + \theta_2 + 2$. Formula (ii) formalizes the next question "In all the cases where the value assigned to parameter θ_1 is greater than the value assigned to parameter θ_2, is it true that any cycle has a duration bounded by $2\theta_1 + 2$?". As there is no free parameter in the question, the question has a YES-NO answer. This is a *model-checking problem*. For formula (ii), the answer is YES in configuration $(q_0, 0, 0)$. Finally, formula (iii) lets parameter θ_2 free and formalizes the question "What are the possible values that can be given to θ_2 such that for any value of $\theta_1 \ge 5$, a cycle throught the four states lasts at most $2\theta_1 + 1$ time units ?". This is again a parameter synthesis problem and the answer is $\theta_2 \le 4$.

In this paper, we study the algorithmic treatment of such problems. Our results are as follows. On the negative side, we show that the model-checking problem of TCTL extended with parameters is *undecidable* over timed automata with *only one* parametric clock. The undecidability result needs *equality in the logic*. On the positive side, we show that when equality is not allowed in the logic, the model-checking problem becomes *decidable* and the parameter synthesis problem is solvable. Our algorithm is based on automata theoretic principles and an extension of our method (see [BDR02]) to express durations of paths of a timed automata using Presburger arithmetic. In particular, all the formulae given in the example above are in the decidable fragment.

The paper is organized as follows. In Section 2, we introduce the model of one parametric clock timed automaton and the parametric extension of TCTL that we consider. In Section 3, we establish the undecidability of the model-checking problem if equality can be used in the logic and we show how to solve the problem algorithmically if the use of equality is not allowed in the logic. Proofs of two important propositions introduced in Section 3 are postponed in Section 4. We finish the paper in Section 5 by drawing some conclusions.

2 Parameters Everywhere

In this section, we introduce *parameters* in the automaton used to model the system *as well as* in the logic used to specify properties of the system. The automata are parametric timed automata as defined in [AHV93] with a *discrete* time domain and *one* parametric clock. The logic is Parametric Timed CTL Logic as defined in [BDR02].

Notation 1 Let Θ be a fixed finite set of *parameters* θ that appear *both* in the automata and in the logical formulae. A *parameter valuation* for Θ is a function $v : \Theta \rightarrow \mathbb{N}$ which assigns a natural number to each parameter $\theta \in \Theta$. In the sequel, $\alpha, \beta, \ldots$ mean any linear term $\Sigma_{i \in I} c_i \theta_i + c$, with $c_i, c \in \mathbb{N}$ and $I \subseteq \{1, \ldots, m\}$. A parameter valuation v is naturally extended to linear terms by defining $v(c) = c$ for any $c \in \mathbb{N}$.

We denote by x the *unique* parametric clock. The same notation x is used for both the clock and a value of the clock. A *guard* g is any conjunction of $x \sim \alpha$ with $\sim \in \{=, <, \le, >, \ge\}$. We denote by $\mathcal{G}$ the set of guards. Notation $x \models_v g$ means that x satisfies g under valuation v. We use notation Σ for the set of *atomic propositions*.

2.1 Parametric Timed Automata

We recall the definition of one parametric clock timed automata as introduced in [AHV93]. We make the hypothesis that non-parametric clocks have all been eliminated by a technique related to the region construction, see [AHV93] for details.

Definition 2 A *parametric timed automaton* $\mathcal{A}$ is a tuple $(Q, E, \mathcal{L}, \mathcal{I})$, where Q is a finite set of *states*, $E \subseteq Q \times \{0, 1\} \times \mathcal{G} \times 2^{\{x\}} \times Q$ is a finite set of *edges*, $\mathcal{L} : Q \rightarrow 2^{\Sigma}$ is a *labeling* function and $\mathcal{I} : Q \rightarrow \mathcal{G}$ assigns an *invariant* $\mathcal{I}(q)$ to each state q.

A *configuration* of $\mathcal{A}$ is a pair (q, x), where q is a state and x is a clock value.

Whenever a parameter valuation v is given, $\mathcal{A}$ becomes a usual one-clock timed automaton denoted by $\mathcal{A}^v$. We recall the next definitions of transition and run in $\mathcal{A}^v$.

Definition 3 Let v be a parameter valuation. A *transition* $(q,x) \xrightarrow{\tau} (q',x')$ between two configurations (q,x) and (q',x') is allowed in $\mathcal{A}^v$ if (1) $x \models_v \mathcal{I}(q)$ and $x' \models_v \mathcal{I}(q')$, (2) there exists an edge $(q,\tau,g,r,q') \in E$ such that $x+\tau \models_v g$ and $x' = 0$ if $r = \{x\}$, $x' = x+\tau$ if $r = \varnothing$.

A *run* $\rho = (q_i,x_i)_{i\geq 0}$ of $\mathcal{A}^v$ is an infinite sequence of transitions $(q_i,x_i) \xrightarrow{\tau_i} (q_{i+1},x_{i+1})$ such that $\Sigma_{i\geq 0}\tau_i = \infty$ (*Non Zenoness* property). The *duration* $t = \mathrm{D}_\rho(q_i,x_i)$ at configuration (q_i,x_i) of ρ is equal to $t = \Sigma_{0\leq j<i}\tau_j$. A *finite run* ρ is a finite sequence of transitions. It is shortly denoted by $(q,x) \rightsquigarrow (q',x')$ such that (q,x) (resp. (q',x')) is its first (resp. last) configuration. Its *duration* D_ρ is equal to $\mathrm{D}_\rho(q',x')$.

Note that in the previous definition of transition, time increment τ is first added to x, guard g is then tested, and finally x is reset according to r.

2.2 Parametric Timed CTL Logic

Formulae of *Parametric Timed CTL logic*, PTCTL for short, are formed by a bloc of quantifiers over some parameters followed by a quantifier-free temporal formula. They are defined as follows. Notation σ means any atomic proposition $\sigma \in \Sigma$ and α,β are linear terms as before.

Definition 4 A PTCTL formula f is of the form

$$f = Q_1\theta_1 \ \cdots \ Q_k\theta_k \ \varphi$$

such that $k \geq 0$, $\{\theta_1,\ldots,\theta_k\} \subseteq \Theta$, $Q_j \in \{\exists,\forall\}$ for each j, $1 \leq j \leq k$, and φ is given by the following grammar

$$\varphi ::= \sigma \mid \alpha \sim \beta \mid \neg\varphi \mid \varphi \vee \varphi \mid \exists\bigcirc\varphi \mid \varphi \exists \mathrm{U}_{\sim\alpha} \varphi \mid \varphi \forall \mathrm{U}_{\sim\alpha}\varphi$$

Note that usual operators $\exists\mathrm{U}$ and $\forall\mathrm{U}$ are obtained as $\exists\mathrm{U}_{\geq 0}$ and $\forall\mathrm{U}_{\geq 0}$. We also use the following abbreviations: $\exists\Diamond_{\sim\alpha}\varphi$ for $\top\exists\mathrm{U}_{\sim\alpha}\varphi$, $\forall\Diamond_{\sim\alpha}\varphi$ for $\top\forall\mathrm{U}_{\sim\alpha}\varphi$, $\exists\Box_{\sim\alpha}\varphi$ for $\neg\forall\Diamond_{\sim\alpha}\neg\varphi$, and $\forall\Box_{\sim\alpha}\varphi$ for $\neg\exists\Diamond_{\sim\alpha}\neg\varphi$. We use notation QF-PTCTL for the set of *quantifier-free* formulae φ of PTCTL. The set of parameters that are *free* in f, that is, not under the scope of a quantifier, is denoted by Θ_f. Thus, for a QF-PTCTL formula φ, we have $\Theta_\varphi = \Theta$. So the set of free parameters of φ are the parameters that appear effectively in the formula and in the automaton.

We now give the *semantics* of PTCTL.

Definition 5 Let $\mathcal{A}$ be a parametric timed automaton and (q,x) be a configuration of $\mathcal{A}$. Let $f = Q_1\theta_1 \ \cdots \ Q_k\theta_k \ \varphi$ be a PTCTL formula. Given a paramater valuation v on Θ_f, the *satisfaction* relation $(q,x) \models_v f$ is defined inductively as follows. If $f = \varphi$, then $(q,x) \models_v \varphi$ according to following rules:

- $(q,x) \models_v \sigma$ iff there exists[2] a run $\rho = (q_i,x_i)_{i\geq 0}$ in $\mathcal{A}^v$ with $(q,x) = (q_0,x_0)$ and $\sigma \in \mathcal{L}(q)$
- $(q,x) \models_v \alpha \sim \beta$ iff there exists a run $\rho = (q_i,x_i)_{i\geq 0}$ in $\mathcal{A}^v$ with $(q,x) = (q_0,x_0)$ and $v(\alpha) \sim v(\beta)$
- $(q,x) \models_v \neg\varphi$ iff $(q,x) \not\models_v \varphi$
- $(q,x) \models_v \varphi \vee \psi$ iff $(q,x) \models_v \varphi$ or $(q,x) \models_v \psi$
- $(q,x) \models_v \exists\bigcirc\varphi$ iff there exists a run $\rho = (q_i,x_i)_{i\geq 0}$ in $\mathcal{A}^v$ with $(q,x) = (q_0,x_0)$ and $(q_1,x_1) \models_v \varphi$
- $(q,x) \models_v \varphi\exists\mathrm{U}_{\sim\alpha}\psi$ iff there exists a run $\rho = (q_i,x_i)_{i\geq 0}$ in $\mathcal{A}^v$ with $(q,x) = (q_0,x_0)$, there exists $i \geq 0$ such that $\mathrm{D}_\rho(q_i,x_i) \sim v(\alpha)$, $(q_i,x_i) \models_v \psi$ and $(q_j,x_j) \models_v \varphi$ for all $j < i$
- $(q,x) \models_v \varphi\forall\mathrm{U}_{\sim\alpha}\psi$ iff for any run $\rho = (q_i,x_i)_{i\geq 0}$ in $\mathcal{A}^v$ with $(q,x) = (q_0,x_0)$, there exists $i \geq 0$ such that $\mathrm{D}_\rho(q_i,x_i) \sim v(\alpha)$, $(q_i,x_i) \models_v \psi$ and $(q_j,x_j) \models_v \varphi$ for all $j < i$

If $f = \exists\theta f'$, then $(q,x) \models_v f$ iff there exists $c \in \mathbb{N}$ such that $(q,x) \models_{v'} f'$ where v' is defined on $P_{f'}$ by $v' = v$ on P_f and $v'(\theta) = c$. If $f = \forall\theta f'$, then $(q,x) \models_v f$ iff for all $c \in \mathbb{N}$, $(q,x) \models_{v'} f'$ where v' is defined on $P_{f'}$ by $v' = v$ on P_f and $v'(\theta) = c$.

The problems that we want to solve in this paper are the next ones. The first problem is the model-checking problem for PTCTL formulae f with *no* free parameters. In this case, we omit the index by v in the satisfaction relation $(q,x) \models f$ since no parameter (neither in the automaton nor in the formula) has to receive a valuation.

Problem 6 The *model-checking* problem is the following. Given a parametric timed automaton $\mathcal{A}$ and a PTCTL formula f such that $\Theta_f = \varnothing$, given a configuration (q,x) of $\mathcal{A}$, does $(q,x) \models f$ hold ?

The second problem is the more general problem of parameter synthesis for PTCTL formulae f such that Θ_f is *any* subset of Θ.

Problem 7 The *parameter synthesis* problem is the following. Given a parametric timed automaton $\mathcal{A}$ and a configuration (q,x) of $\mathcal{A}$, given a PTCTL formula f, compute a symbolic representation of the set of parameter valuations v on Θ_f such that $(q,x) \models_v f$. This symbolic representation should support boolean operations, projections and checking emptiness.

[2] We verify the existence of a run starting in (q,x) to ensure that time can progress in $\mathcal{A}^v$ from that configuration.

3 Decision Problems

In this section, we prove that the model-checking problem is decidable when equality is not used in operators $\exists U_{\sim\alpha}$ and $\forall U_{\sim\alpha}$. Our proof is based on a translation to Presburger arithmetic. We prove that the model-checking problem becomes undecidable as soon as equality is allowed.

When equality is not used, we solve the more general parameter synthesis problem for formulae φ of QF-PTCTL. Our approach is as follows : we construct a Presburger formula $\Delta_{q,\varphi}(x,\Theta)$ with free variables x and all $\theta \in \Theta$ such that $(q,x_0) \models_v \varphi$ iff $\Delta_{q,\varphi}(x_0, v(\Theta))$ is true, for *any* valuation v on Θ and *any* value x_0 of the clock. Therefore the model-checking problem is decidable since Presburger arithmetic has a decidable theory. Indeed, if we denote by $Q\Theta\ \varphi$ a PTCTL formula f with no free parameters, then $(q,x_0) \models f$ iff $Q\Theta\ \Delta_{q,\varphi}(x_0,\Theta)$ is true.

In the sequel, we use subscripts to indicate what are the limitations imposed to $\sim$ in operators $\exists U_{\sim\alpha}$ and$\exists U_{\sim\alpha}$. For instance, notation $\text{PTCTL}_{\{=\}}$ means that $\sim$ can only be equality.

3.1 Undecidability Results for Equality

We prove here that Problem 6 is undecidable for $\text{PTCTL}_{\{=\}}$. The proof relies on the undecidability of Presburger arithmetic with divisibility.

Presburger arithmetic with divisibility, PAD for short, is an extention of Presburger arithmetic with integer divisibility relation. The additional divisibility relation is noted as $z|z'$ and means "z divides z'". Let $z, z', z_1, z_2, \ldots, z_n$ be variables with $z' > 0$ and α be a linear term over the set of variables $z_1, z_2, \ldots, z_n$. Every formula of PAD can be put into the following form:

$$Qz_{i_1}Qz_{i_2}\ldots Qz_{i_l} \cdot (\neg)\phi_1 \star (\neg)\phi_2 \star \cdots \star (\neg)\phi_m$$

where $\star \in \{\vee, \wedge\}$ and each ϕ_i is one of the following atomic formulae: $(i)\ \alpha = z$, $(ii)\ \alpha < z$, $(iii)\ z|z'$. We say that such a formula is in *normal form*. The Presburger theory is the set of its sentences that are true. While PA is a decidable theory, PAD theory is undecidable [Bès02].

Theorem 8 *For any sentence Φ of* PAD*, we can construct a parametric timed automaton $\mathcal{A}$, a configuration (q,x_0) and a PTCTL formula f such that Φ is* TRUE *iff the answer to the model checking problem $(q,x_0) \models f$ for A is* YES.

Proof Let us make the assumption that sentence Φ is in normal form, that is

$$Qz_{i_1}Qz_{i_2}\ldots Qz_{i_n} \cdot (\neg)\phi_1 \star (\neg)\phi_2 \star \cdots \star (\neg)\phi_m.$$

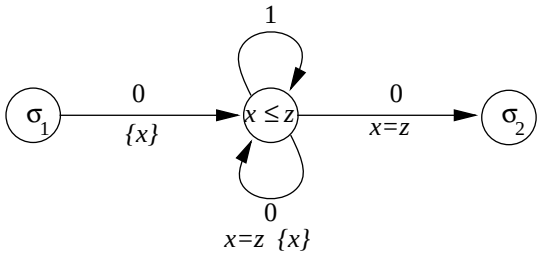

Figure 2. Automaton for $z|z'$

We are going to construct a $\text{PTCTL}_{\{=\}}$ formula f and a paramatric timed automaton $\mathcal{A}$ over the set of parameters equal to $\Theta = \{z_1, z_2, \ldots, z_n\}$.

For each subformula ϕ_i of the form $\alpha = z$ or $\alpha < z$, we define the $\text{PTCTL}_{\{=\}}$ formula $\hat{\phi}_i = \phi_i$. For each subformula ϕ_i of the form $z|z'$, we construct the next parametric timed automaton $\mathcal{A}_{\phi_i}$ and $\text{PTCTL}_{\{=\}}$ formula $\hat{\phi}_i$. The automaton $\mathcal{A}_{\phi_i}$ given in Figure 2. We label the unique initial state of this automaton by σ_1^i and the unique final state by σ_2^i. It is easy to see that there is a path from the initial state to the final state with length l iff $z|l$. For formula $\hat{\phi}_i$, we take $\sigma_1^i \wedge \exists\Diamond_{=z'}\sigma_2^i$.

Now we construct formula f as follows

$$Qz_{i_1}Qz_{i_2}\ldots Qz_{i_n} \cdot (\neg)\hat{\phi}_1 \star (\neg)\hat{\phi}_2 \star \cdots \star (\neg)\hat{\phi}_m.$$

We construct the automaton $\mathcal{A}$ by first taking the union of all the previous automata $\mathcal{A}_{\phi_i}$ (introduced for the divisibility subformulae). We then merge their initial states into a unique state of $\mathcal{A}$ that we call q. The label $\mathcal{L}(q)$ of q is the union of the labels σ_1^i. Finally, we add a new state q' to $\mathcal{A}$ and an edge $(f_i, 0, \top, \varnothing, q')$ from any final state f_i of $\mathcal{A}_{\phi_i}$ to state q' labelled with $\tau = 0$ and without guard and reset. To complete the construction, we add a self-loop $(q', 1, \top, \varnothing, q')$ on q' that allows time to progress.

It is easy to see that given $\mathcal{A}$, we have $(q,0) \models f$ iff Φ is TRUE. □

As a direct consequence of the last theorem, we have:

Corollary 9 *The model-checking problem for $\text{PTCTL}_{\{=\}}$ is undecidable.*

3.2 Decidability for PTCTL without Equality

In this section, we provide solutions to the model-checking problem and the parameter synthesis problem for $\text{PTCTL}_{\{<,\leq,>,\geq\}}$. Our solutions require to work with automata that are more general than in Definition 2.

Notation 10 Linear terms $\alpha, \beta, \ldots$ are any $\Sigma_i c_i\theta_i + c$, with $c_i, c \in \mathbb{Z}$ (instead of $\mathbb{N}$). Whenever a linear term α is used,

it is supposed that $\alpha \geq 0$. Comparison symbol $\sim$ belongs to the extended set $\{=, <, \leq, >, \geq, \leq_a, \geq_a\}$. For any constant $a \in \mathbb{N}_0$, notation $z \leq_a z'$ (resp. $z \geq_a z'$) means $z \equiv z' \mod a$ and $z \leq z'$ (resp. $z \geq z'$). Equivalently, $z \leq_a z'$ (resp. $z \geq_a z'$) iff there exists $y \in \mathbb{N}$ such that $z + ay = z'$ (resp. $z = z' + ay$).

Any $x \sim \alpha$ is called an *x-atom*, any $\alpha \sim \beta$ is called an *θ-atom*. An *x-formula* is any conjunction of x-atoms, and a *θ-formula* is any conjunction of θ-atoms. We denote by $\mathcal{B}_{x,\Theta}$ the set of *boolean combinations* of x-atoms and θ-atoms.

Lemma 11 *Any formula of $\mathcal{B}_{x,\Theta}$ can be rewritten as a disjunction of conjunctions of x-atoms and θ-atoms with $\sim$ limited to $\{=, <, >, \leq_a, \geq_a\}$.*

Proof The formula is first put into disjunctive normal form. Second we suppress negation in any $\neg(z \sim z')$ as follows. This is done easily for $\sim \in \{<, \leq, >, \geq\}$. Negation $\neg(z = z')$ is replaced by $z < z' \vee z > z'$. Negation $\neg(z \leq_a z')$ is equivalent to $(z > z') \vee (\bigvee_{0<b<a} z + b \leq_a z')$. Similarly for $\neg(z \geq_a z')$. Third we replace $\leq$ and $\geq$ thanks to $<$, $>$ and $=$. Finally this formula is put into disjunctive normal form. □

Definition 12 An *extended* parametric timed automaton $\mathcal{A} = (Q, E, \mathcal{L}, \mathcal{I})$ is defined as before except that (1) any guard $g \in \mathcal{G}$ used in E is an x-formula, (2) function $\mathcal{I} : Q \to \mathcal{B}_{x,\Theta}$ assigns to any state q an invariant in $\mathcal{B}_{x,\Theta}$.

From now on, we call *automaton* any extended parametric timed automaton $\mathcal{A}$. It should be noted that the extension of $\sim$ to $\{=, <, \leq, >, \geq, \leq_a, \geq_a\}$ is only valid inside automata, and not inside PTCTL formulae.

Solutions to the model-checking problem and the parameter synthesis problem are obtained as a corollary of the next theorem.

Theorem 13 *Let $\mathcal{A}$ be an automaton and q be a state of $\mathcal{A}$. Let φ be a QF-PTCTL$_{\{<,\leq,>,\geq\}}$ formula. Then there exists a $\mathcal{B}_{x,\Theta}$ formula $\Delta_{q,\varphi}(x, \Theta)$ with free variables x and all $\theta \in \Theta$ such that $(q, x_0) \models_v \varphi$ iff $\Delta_{q,\varphi}(x_0, v(\Theta))$ is true, for any valuation v on Θ and any clock value x_0. The construction of formula $\Delta_{q,\varphi}$ is effective.*

Corollary 14 *The model-checking problem for PTCTL$_{\{<,\leq,>,\geq\}}$ is decidable.*

Proof Let $Q\Theta\ \varphi$ be a PTCTL formula f with no free parameters. Then $(q, x_0) \models f$ iff $Q\Theta\ \Delta_{q,\varphi}(x_0, \Theta)$ is true. Note that any $\mathcal{B}_{x,\Theta}$ formula is a Presburger formula. Indeed operators $\leq_a$ and $\geq_a$ can be rewritten in Presburger arithmetic, as well as any linear term $\alpha = \Sigma_i c_i\theta_i + c$, with $c_i, c \in \mathbb{Z}$ since they satisfy $\alpha \geq 0$ (see Notation 10). As Presburger arithmetic has a decidable theory, the model-checking problem is decidable. □

Since the construction of formula $\Delta_{q,\varphi}$ is effective and that Presburger arithmetic supports boolean operations, projections and checking emptiness, the next corollary is immediate.

Corollary 15 *Let $\mathcal{A}$ be an automaton and (q, x_0) be a configuration of $\mathcal{A}$. Let $\{\theta_1, \ldots, \theta_k\} \subseteq \Theta$ with $k \geq 0$ and let $f = Q_1\theta_1\ \cdots\ Q_k\theta_k\ \varphi$ be a PTCTL$_{\{<,\leq,>,\geq\}}$ formula. Then formula $Q_1\theta_1\ \cdots\ Q_k\theta_k\ \Delta_{q,\varphi}(x, \Theta)$ with free parameters in Θ_f is an effective characterization of the set of valuations v on Θ_f such that $(q, x_0) \models_v f$.*

The proof of Theorem 13 is by induction on the way formula φ is constructed. Instead of the grammar given in Definition 4, we prefer to work with the grammar

$$\varphi ::= \sigma \mid \alpha \sim \beta \mid \neg\varphi \mid \varphi \vee \varphi \mid \exists\bigcirc\varphi \mid \varphi \exists \mathrm{U}_{\sim\alpha} \varphi \mid \exists\Box_{<\alpha}\varphi \mid \exists\Box\varphi$$

This grammar is equivalent because $\exists\Box_{\leq\alpha}\varphi$ can be replaced by $\exists\Box_{<\alpha+1}\varphi$, formula $\exists\Box_{\geq\alpha}\varphi$ by $(\exists\Box_{>\alpha-1}\varphi) \vee (\exists\Box\varphi \wedge \alpha = 0)$, formula $\exists\Box_{>\alpha}\varphi$ by $\exists\Diamond_{\leq\alpha}\exists\bigcirc\exists\Box\varphi$, and formula $\varphi\forall \mathrm{U}_{\sim\alpha}\psi$ by $\neg(\neg\psi\exists \mathrm{U}_{\sim\alpha}(\neg\varphi \wedge \neg\psi)) \vee \neg\exists\Box_{\sim\alpha}\neg\psi$.

It is not difficult to check that the semantics of the new operator $\exists\Box_{<\alpha}\varphi$ is given by

$(q, x) \models_v \exists\Box_{<\alpha}\varphi$ iff there exists a run $\rho = (q_i, x_i)_{i\geq 0}$ of $\mathcal{A}^v$ with $(q, x) = (q_0, x_0)$ such that $(q_i, x_i) \models_v \varphi$ for any $i \geq 0$ such that $\mathrm{D}_\rho(q_i, x_i) \geq v(\alpha)$.

Let us give a flavour of the proof of Theorem 13 on formula $\exists\bigcirc\varphi$. The satisfaction relation $(q, x_0) \models_v \exists\bigcirc\varphi$ holds iff in $\mathcal{A}^v$, there exists a transition $(q, x_0) \xrightarrow{\tau} (q', x_0')$ such that $(q', x_0') \models_v \varphi$ and (q', x_0') is the first configuration of an infinite run. Suppose that $\Delta_{q',\varphi}$ has been constructed by induction as a formula of $\mathcal{B}_{x,\Theta}$. If we replace invariant $\mathcal{I}(q')$ of state q' by $\mathcal{I}(q') \wedge \Delta_{q',\varphi}$, then relation $(q', x_0') \models_v \varphi$ holds automatically. So, what we still need is a $\mathcal{B}_{x,\Theta}$ formula that expresses the existence of an infinite run in $\mathcal{A}^v$ starting at configuration (q', x_0'). This is possible:

Proposition 16 *Let $\mathcal{A}$ be an automaton and q be a state. Then there exists a $\mathcal{B}_{x,\Theta}$ formula $R_q(x, \Theta)$ such that for any valuation v and any clock value x_0, $R_q(x_0, v(\Theta))$ is true iff there exists an infinite run in $\mathcal{A}^v$ starting with (q, x_0). The construction of R_q is effective.*

The proof of this proposition is postponed in Section 4.

Looking at the semantics of formula $\varphi\exists \mathrm{U}_{\sim\alpha}\psi$, we see that we also need a $\mathcal{B}_{x,\Theta}$ formulae that expresses the existence of a finite run ρ in $\mathcal{A}^v$ starting and ending at given configurations such that $\mathrm{D}_\rho \sim v(\alpha)$. This is again possible:

Notation 17 Let $\mathcal{A}$ be an automaton and v be a valuation. Given two states q, q' of $\mathcal{A}$ and x_0 a clock value, we denote by $\lambda^{x_0}_{q,q'}$ the set of durations $t \in \mathbb{N}$ such that there exists a finite run $\rho = (q, x_0) \rightsquigarrow (q', \cdot)$ in $\mathcal{A}^v$ such that $t = \mathrm{D}_\rho$.

Proposition 18 *Let $\mathcal{A}$ be an automaton and q, q' be two states. Let $\sim \in \{<, \leq, >, \geq\}$ and α be a linear term. Then there exists a $\mathcal{B}_{x,\Theta}$ formula $\lambda^{\sim\alpha}_{q,q'}(x, \Theta)$ such that for any valuation v and any clock value x_0, $\lambda^{\sim\alpha}_{q,q'}(x_0, v(\Theta))$ is true iff there exists $t \sim v(\alpha)$ in set $\lambda^{x_0}_{q,q'}$. The construction of $\lambda^{\sim\alpha}_{q,q'}$ is effective.*

The proof of this proposition is also postponed in Section 4.

We are now ready to proof Theorem 13. We first need the next variations of Propositions 16 and 18.

Lemma 19 *Let $\mathcal{A}$ be an automaton and q be a state. Let $\delta \in \mathcal{B}_{x,\Theta}$. There exists a formula $R_{q,\delta}(x, \Theta)$ in $\mathcal{B}_{x,\Theta}$ such that for any valuation v and any clock value x_0, $R_{q,\delta}(x_0, v(\Theta))$ is true iff there exists an infinite run in $\mathcal{A}^v$ with first configuration (q, x_0) and $(q, x_0) \models_v \delta$.*

Proof First $\mathcal{A}$ is modified as follows. A copy $\overline{q}$ of q is added to Q such that $\mathcal{L}(\overline{q}) = \mathcal{L}(q)$ and $\mathcal{I}(\overline{q}) = \mathcal{I}(q) \wedge \delta$. A copy $(\overline{q}, \tau, g, r, q')$ is also added for any edge (q, τ, g, r, q') leaving q. Then by Proposition 16 applied to $\overline{q}$, we get the expected formula $R_{q,\delta}(x, \Theta)$. □

Lemma 20 *Let $\mathcal{A}$ be an automaton and q be a state. Let $\delta_p \in \mathcal{B}_{x,\Theta}$ for all $p \in Q$. There exists a formula $R_{q,\delta_p}(x, \Theta)$ in $\mathcal{B}_{x,\Theta}$ such that for any valuation v and any clock value x_0, $R_{q,\delta_p}(x_0, v(\Theta))$ is true iff there exists an infinite run ρ in $\mathcal{A}^v$ such that $(p, x) \models_v \delta_p$ for each configuration (p, x) of ρ.*

Proof The proof is similar to the previous one. The automaton is modified such that for any state p of Q, $\mathcal{I}(p)$ is replaced by $\mathcal{I}(p) \wedge \delta_p$. We obtain formula $R_{q,\delta_p}(x, \Theta)$ by Proposition 16. □

Lemma 21 *Let $\mathcal{A}$ be an automaton and q, q' be two states. Let $\sim \in \{<, \leq, >, \geq\}$ and α be a linear term. Let $\delta_p \in \mathcal{B}_{x,\Theta}$, for all $p \in Q$. Let $\delta' \in \mathcal{B}_{x,\Theta}$. There exists a formula $\lambda^{\sim\alpha}_{q,q',\delta_p,\delta'}(x, \Theta)$ in $\mathcal{B}_{x,\Theta}$ such that for any valuation v and any clock value x_0, $\lambda^{\sim\alpha}_{q,q',\delta_p,\delta'}(x_0, v(\Theta))$ is true iff there exists a finite run $\rho = (q, x_0) \rightsquigarrow (q', x')$ in $\mathcal{A}$ such that $\mathrm{D}_\rho \sim v(\alpha)$, $(q', x') \models_v \delta'$ and $(p, x) \models_v \delta_p$ for every configuration (p, x) of ρ distinct from (q', x').*

Proof A copy $\overline{q}'$ of q' is added to Q as well as a copy of any edge of E entering q' as entering $\overline{q}'$. Then $\mathcal{L}(\overline{q}') = \mathcal{L}(q')$ and $\mathcal{I}(\overline{q}') = \mathcal{I}(q') \wedge \delta'$. For any state p of Q, $\mathcal{I}(p)$ is replaced by $\mathcal{I}(p) \wedge \delta_p$. We obtain formula $\lambda^{\sim\alpha}_{q,q',\delta_p,\delta'}(x, \Theta)$ thanks to Proposition 18 applied to $q, \overline{q}'$. □

Proof of Theorem 13 (by induction on φ). If $\varphi = \sigma$, then $(q, x_0) \models_v \varphi$ iff there exists an infinite run starting with (q, x_0) and $\sigma \in \mathcal{L}(q)$. Therefore

$$\begin{array}{rcll} \Delta_{q,\varphi}(x, \Theta) & = & \bot & \text{if } \sigma \notin \mathcal{L}(q) \\ & = & R_q(x, \Theta) & \text{otherwise} \end{array}$$

Similarly, if $\varphi = \alpha \sim \beta$ with $\sim \in \{=, <, \leq, >, \geq\}$, then

$$\Delta_{q,\varphi}(x, \Theta) \quad = \quad (\alpha \sim \beta) \wedge R_q(x, \Theta)$$

If $\varphi = \psi \vee \phi$, then $\Delta_{q,\varphi} = \Delta_{q,\psi} \vee \Delta_{q,\phi}$. If $\varphi = \neg\psi$, then $\Delta_{q,\varphi} = \neg\Delta_{q,\psi}$.

Let us now treat $\varphi = \exists \bigcirc \psi$. Recall that $(q, x_0) \models_v \exists\bigcirc\psi$ iff there exists a transition $(q, x_0) \xrightarrow{\tau} (q', x'_0)$ such that $(q', x'_0) \models_v \psi$ and (q', x'_0) is the first configuration of an infinite run. Let (q, τ, g, r, q') be the edge of E that has lead to transition $(q, x_0) \xrightarrow{\tau} (q', x'_0)$. Then, $x'_0 = 0$ if $r = \{x\}$, and $x'_0 = x_0 + \tau$ if $r = \varnothing$. By induction hypothesis, $\Delta_{q',\psi}$ has been constructed such that $\Delta_{q',\psi}(x'_0, v(\Theta))$ is true iff $(q', x'_0) \models_v \psi$. Thus, by Lemma 19 with $\delta = \Delta_{q',\psi}$, we get the expected formula

$$\begin{array}{l} \Delta_{q,\varphi}(x, \Theta) = \\ \bigvee_{(q,\tau,q') \in E_R} (\mathcal{I}(q) \wedge R_{q',\Delta_{q',\psi}}(0, \Theta)) \quad \vee \\ \bigvee_{(q,\tau,q') \in E \setminus E_R} (\mathcal{I}(q) \wedge R_{q',\Delta_{q',\psi}}(x + \tau, \Theta)) \end{array}$$

where E_R is the set of edges that reset the clock.

Let us turn to formula $\varphi = \psi \exists \mathrm{U}_{\sim\alpha} \phi$. We have $(q, x_0) \models_v \varphi$ iff either (1) $0 \sim v(\alpha)$, $(q, x_0) \models_v \phi$ and (q, x_0) is the first configuration of an infinite run, or (2) there exists a finite run $\rho = (q, x_0) \rightsquigarrow (q', x')$ such that $\mathrm{D}_\rho \sim v(\alpha)$, ψ is satisfied at any configuration of ρ distinct from (q', x'), ϕ is satisfied at (q', x') and (q', x') is the first configuration of an infinite run. For any state $p \in Q$, formulae $\Delta_{p,\psi}$ and $\Delta_{p,\phi}$ have been constructed by induction hypothesis. So, in the first case, by Lemma 19 with $\delta = \Delta_{q,\phi}$, we have the next formula

$$\gamma = (0 \sim \alpha) \wedge R_{q,\Delta_{q,\phi}}(x, \Theta)$$

In the second case, we first apply Lemma 19 to state q' and formula $\Delta_{q',\phi}$ to get formula $\delta' = R_{q',\Delta_{q',\phi}}(x, \Theta)$. After, by Lemma 21 with $\delta_p = \Delta_{p,\psi}$, $p \in Q$ and δ', we get the next formula

$$\gamma_{q'} = \lambda^{\sim\alpha}_{q,q',\delta_p,\delta'}(x, \Theta)$$

Therefore, formula $\Delta_{q,\varphi}$ is the disjunction $\gamma \vee \bigvee_{q' \in Q} \gamma_{q'}$.

Let φ be $\exists \Box_{<\alpha} \psi$. Then $(q, x_0) \models_v \varphi$ iff there exists a finite run $\rho = (q, x_0) \rightsquigarrow (q', x')$ such that $\mathrm{D}_\rho \geq v(\alpha)$, $(p, x) \models_v \psi$ for each configuration (p, x) of ρ distinct from (q', x') and (q', x') is the first configuration of an infinite run. As above, we construct formula

$$\gamma_{q'} = \lambda^{\geq\alpha}_{q,q',\delta_p,\delta'}(x, \Theta)$$

where $\delta_p = \Delta_{p,\psi}$, $p \in Q$ and $\delta' = R_{q'}(x, \Theta)$. Hence $\Delta_{q,\varphi}$ is formula $\bigvee_{q' \in Q} \gamma_{q'}$.

Finally, suppose that $\varphi = \exists\Box\psi$. Then $(q, x_0) \models_v \varphi$ iff there is an infinite run with first configuration (q, x_0) such that all its configurations satisfy ψ. It follows by Lemma 20 with $\delta_p = \Delta_{p,\psi}$, $p \in Q$, that $\Delta_{q,\varphi}(x, \Theta)$ is equal to $R_{q,\Delta_{p,\psi}}(x, \Theta)$.

The proof is completed since all the formulae belong to $\mathcal{B}_{x,\Theta}$ and their construction is effective. □

4 Durations

The aim of this section is a proof of Propositions 16 and 18. Several steps are necessary. In Subsections 4.1 and 4.2, we work with a *reset-free* automaton $\mathcal{A}$ such that the clock is never reset, and we concentrate on runs of the form $(i, x_0) \rightsquigarrow (f, \cdot)$, $i \in I$, $f \in F$, such that I and F are fixed subsets of states, and x_0 is a fixed clock value. In Subsection 4.1, a sequence of transformations is performed on the automaton such that x-terms used in guards and invariants are limited to equalities $x = \alpha$. These simplifications allow the description by a Presburger formula of durations of runs $(i, x_0) \rightsquigarrow (f, \cdot)$, $i \in I$, $f \in F$, in Subsection 4.2. In the last subsection, no restriction is no longer imposed to $\mathcal{A}$. The automaton is just slightly modified in a way to reset clock x inside a state (called *reset-state*), instead of along an edge. We there study in details set $\lambda^{x_0}_{q,q'}$ introduced in Notation 17. Any run $(q, x_0) \rightsquigarrow (q', \cdot)$ can be decomposed into a sequence of runs ρ_j, $1 \leq j \leq k$, going from reset-states to reset-states. Its duration is thus the sum of durations D_{ρ_j}, $1 \leq j \leq k$. Any D_{ρ_j} falls into durations being studied in Section 4.2. Therefore, set $\lambda^{x_0}_{q,q'}$ is symbolically described by a rational expression whose letters are durations in reset-free automata. Thanks to this rational expression, we are finally able to prove Propositions 16 and 18.

4.1 Automata Transformations

Hypothesis (∗) We assume that $\mathcal{A} = (Q, I, F, E, \mathcal{L}, \mathcal{I})$ is a *reset-free* automaton with a set $I \subseteq Q$ of *initial* states and a set $F \subseteq Q$ of *final* states. We also assume such that $I \cap F = \varnothing$, no edge enters $i \in I$ and no edge leaves $f \in F$. We suppose that x_0 is a fixed clock value.

Given a valuation v, we denote by $\mathcal{R}(\mathcal{A}^v, x_0)$ the set of runs of $\mathcal{A}^v$ of the form $(i, x_0) \rightsquigarrow (f, \cdot)$ with $i \in I$ and $f \in F$. We are going to perform a sequence of transformations on $\mathcal{A}$ that will preserve $\mathcal{R}(\mathcal{A}^v, x_0)$ in the following sense. During a transformation, state q will possibly be splitted into several copies $\overline{q}_j$. Runs before and after the splitting can be supposed identical up to a *renaming* of any $\overline{q}_j$ into q.

The aim of these transformations is a simplification of guards and invariants used in the automaton. Guards will disappear. Invariants of states $q \in Q \setminus (I \cup F)$ will be a conjunction of at most one x-atom (of the form $x = \alpha$) and of a certain number of θ-atoms. This simplification is possible, mainly because the automaton is reset-free.

Definition 22 An automaton $\mathcal{A}$ is *normalized* if (1) no guard labels the edges, that is $E \subseteq Q \times \{0, 1\} \times Q$, (2) any invariant $\mathcal{I}(q)$ is a conjunction $\mathcal{I}_x(q) \wedge \mathcal{I}_\theta(q)$ of an x-formula $\mathcal{I}_x(q)$ and a θ-formula $\mathcal{I}_\theta(q)$ with $\sim$ limited to $\{=, <, >, \leq_a, \geq_a\}$, (3) $\mathcal{I}_x(q) = \mathcal{I}_x(q')$ for any edge $(q, \tau, q') \in E$ such that $\tau = 0$.

Lemma 23 *Any automaton $\mathcal{A}$ can be effectively normalized with set $\mathcal{R}(\mathcal{A}^v, x_0)$ preserved for any valuation v.*

Proof We can assume that all the edges entering state q are labelled with the same guard g. Indeed, every state q not satisfying this property is split into several copies and the edges entering q are redirected to each copy according to the guards labelling those edges. The copies of q have the same $\mathcal{L}(q)$ and $\mathcal{I}(q)$ as q, they all belong to F if $q \in F$. Therefore, guard g can be erased from all the edges entering q and added as a conjunction to $\mathcal{I}(q)$. Property (1) follows. Now by Lemma 11, the new formula $\mathcal{I}(q) \wedge g$ is rewritten as a disjunction of k formulae δ_j, $1 \leq j \leq k$, where each δ_j is a conjunction of an x-formula and a θ-formula with $\sim \in \{=, <, >, \leq_a, \geq_a\}$. We modify $\mathcal{A}$ by splitting state q into k states $\overline{q}_j$, $1 \leq j \leq k$, such that $\mathcal{L}(\overline{q}_j) = \mathcal{L}(q)$ and $\mathcal{I}(\overline{q}_j) = \delta_j$. Accordingly, we split any edge of E that enters or leaves state q. The k copies $\overline{q}_j$ all belong to I (resp. F) if q belongs to I (resp. F). Property (2) thus holds. To get Property (3), for any edge $(q, 0, q')$, replace $\mathcal{I}_x(q)$ and $\mathcal{I}_x(q')$ by the x-formula $\mathcal{I}_x(q) \wedge \mathcal{I}_x(q')$. All these transformations preserve $\mathcal{R}(\mathcal{A}^v, x_0)$. □

Definition 24 A normalized automaton $\mathcal{A}$ is called *simplified* if (1) $\mathcal{I}_x(q)$ is a conjunction of x-atoms $x = \alpha$ if $q \notin I \cup F$, and of x-atoms $x = \alpha$, $x < \beta$ if $q \in F$[3], (2) for any state $q \in Q$, $\mathcal{I}_x(q)$ contains at most one x-atom $x \sim \alpha$ with $\sim$ equal to $=$, (3) for any run $\rho \in \mathcal{R}(\mathcal{A}^v, x_0)$, for any x-atom $x = \alpha$, there exists at most one configuration (q, x) of ρ such that $\mathcal{I}_x(q)$ equals $x = \alpha$.

Proposition 25 *Any normalized automaton $\mathcal{A}$ can be effectively simplified with set $\mathcal{R}(\mathcal{A}^v, x_0)$ preserved for any valuation v.*

Proof The proof of Proposition 25 needs several steps. At the end of each step, the automaton will be *normalized* if necessary. Each transformation described in the proof will preserve set $\mathcal{R}(\mathcal{A}^v, x_0)$ for any valuation v. Given a state q, we will often view $\mathcal{I}(q)$ as a *set* of x-atoms and θ-atoms (instead of a conjunction) and we will say that such a term *belongs* to q (instead of $\mathcal{I}(q)$) or *appears* in q.

[3] Let us note that $\mathcal{I}_x(q)$ remains any x-formula if $q \in I$.

First step x-atoms $x \leq_a \alpha$.

Assume that $x \leq_a \alpha$ belongs to some state of $\mathcal{A}$. Let us show that this x-atom can be eliminated at the cost of a new x-atom $x \leq \alpha$. The idea is the following. If $\alpha \equiv b \mod a$ for some $b \in \{0, 1, \cdots, a-1\}$, then

$$x \leq_a \alpha \quad \text{iff} \quad x \equiv b \mod a \text{ and } x \leq \alpha.$$

The automaton is transformed in a way to compute modulo a. Formally we construct $\mathcal{A}_b = (Q', I', F', E', \mathcal{L}', \mathcal{I}')$ where $Q' = Q \times \{0, \ldots, a-1\}$, $I' = I \times \{0, \ldots, a-1\}$, $F' = F \times \{0, \ldots, a-1\}$, $\mathcal{L}'(q, c) = \mathcal{L}(q)$ and $((q, c), \tau, (q', c')) \in E'$ iff $(q, \tau, q') \in E$ and $c' \equiv c + \tau \mod a$. Function $\mathcal{I}'$ is defined as follows. For any $(q, c) \in Q'$, let $\mathcal{I}'(q, c) = \mathcal{I}(q)$. If (q, c) contains $x \leq_a \alpha$, eliminate this state if $c \neq b$, replace $x \leq_a \alpha$ by $x \leq \alpha$ if $c = b$. If $(q, c) \in I'$, add x-atom $x \geq_a c$ and θ-atom $\alpha \geq_a b$. As α depends on the parameter valuation, value b such that $\alpha \equiv b \mod a$ is not known in advance. Therefore the final automaton is the disjoint union of the automata $\mathcal{A}_b$, with $b \in \{0, \cdots, a-1\}$.

The elimination of x-atoms $x \geq_a \alpha$ is performed similarly.

Second step x-atoms $x > \alpha$.

Let us fix an x-atom $x > \alpha$. Recall that the automaton is reset-free. Along a run $\rho \in \mathcal{R}(\mathcal{A}^v, x_0)$, as soon as $x > \alpha$ is satisfied at some configuration of ρ, the next occurrences of $x > \alpha$ are automatically satisfied and can be thus eliminated. The automaton is transformed in a way to count occurrences of $x > \alpha$ thanks to a counter equal to $0, 1$ or 2 in case of $0, 1$ or 2 and more occurrences of $x > \alpha$. Thus when the counter has value 2, any incrementation lets it at value 2. Formally we construct $\mathcal{A}' = (Q', I', F', E', \mathcal{L}', \mathcal{I}')$ where $Q' = Q \times \{0, 1, 2\}$, $F' = F \times \{0, 1, 2\}$, $\mathcal{L}'(q, c) = \mathcal{L}(q)$ and $\mathcal{I}'(q, c) = \mathcal{I}(q)$. Sets I' and E' are defined as follows. For any $q \in I$, state (q, c) belongs to I' with $c = 1$ if $x > \alpha$ belongs to q, and $c = 0$ otherwise. For any $(q, \tau, q') \in E$, edge $((q, c), \tau, (q', c'))$ belongs to E' with $c' = c + 1$ if q' contains $x > \alpha$, and $c' = c$ otherwise. Finally, we suppress $x > \alpha$ in any state $(q, 2)$ containing it.

Now, consider a run $\rho' \in \mathcal{R}(\mathcal{A}'^v, x_0)$ starting with configuration (q_0, c_0, x_0) and having a transition $(q, c, x) \xrightarrow{\tau} (q', c', x')$ such that (q', c') contains $x > \alpha$. Necessarily $c' = 1$ by construction of $\mathcal{A}'$, $(q', c') \notin I'$ by Hypothesis $(*)$ and $\tau = 1$ since $\mathcal{A}$ is normalized (Property (3)). So x-atom $x > \alpha$ is satisfied at configuration (q', c', x') iff x-atom $x = \alpha$ is satisfied at some configuration of ρ' between (q_0, c_0, x_0) and (q, c, x). Therefore, x-atom $x > \alpha$ can be eliminated at the cost of a new x-atom $x = \alpha$. This can be achieved by modifying $\mathcal{A}'$ in the following way (see Figure 3). We add to Q' a copy $(q, \overline{0})$ of any state $(q, 0)$ together with a copy $\mathcal{L}(q, \overline{0})$ of $\mathcal{L}(q, 0)$ and a copy $\mathcal{I}(q, \overline{0})$ of $\mathcal{I}(q, 0)$. Sets I' and F' are let unchanged. To any edge

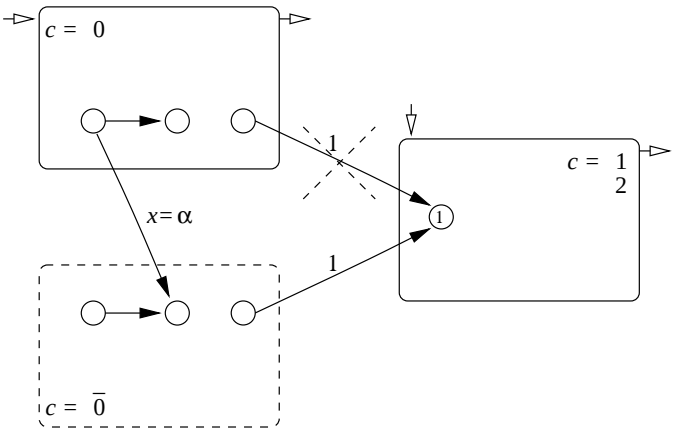

Figure 3. Automaton for x-atoms $x > \alpha$

$((q, 0), \tau, (q', 0)) \in E'$, we add a copy $((q, \overline{0}), \tau, (q', \overline{0}))$ and another copy $((q, 0), \tau, (q', \overline{0}))$ with label $x = \alpha$. Every edge $((q, 0), 1, (q', 1)) \in E'$ is replaced by the edge $((q, \overline{0}), 1, (q', 1))$. At last, we suppress x-atom $x > \alpha$ from any state $(q, 1) \notin I'$ that contains it.

Third step x-atoms $x < \alpha$.

Let us given an x-atom $x < \alpha$. Here, as soon as the last (instead of first) occurrence of $x < \alpha$ is satisfied along a run $\rho \in \mathcal{R}(\mathcal{A}^v, x_0)$, then the previous occurrences of $x < \alpha$ are automatically satisfied. Thus $\mathcal{A}$ is transformed twice as before with the difference that runs are treated in the reverse direction. The first transformation is defined as follows. Let $\mathcal{A}' = (Q', I', F', E', \mathcal{L}', \mathcal{I}')$ with $Q' = Q \times \{0, 1, 2\}$, $I' = I \times \{0, 1, 2\}$, $\mathcal{L}'(q, c) = \mathcal{L}(q)$ and $\mathcal{I}'(q, c) = \mathcal{I}(q)$. For any $q \in F$, state (q, c) belongs to F' with $c = 1$ if $x < \alpha$ belongs to q, and $c = 0$ otherwise. For any $(q, \tau, q') \in E$, edge $((q, c), \tau, (q', c'))$ belongs to E' with $c = c' + 1$ if q contains $x < \alpha$, and $c = c'$ otherwise. In any state $(q, 2)$ that contains $x < \alpha$, this x-atom is eliminated.

Let ρ' be a run in $\mathcal{R}(\mathcal{A}'^v, x_0)$ ending with configuration (q_f, c_f, x_f) and having a transition $(q, c, x) \xrightarrow{\tau} (q', c', x')$ such that (q, c) contains $x < \alpha$. Then $c = 1$, $(q, c) \notin F'$ and $\tau = 1$. Therefore $x < \alpha$ is satisfied at configuration (q, c, x) iff either $x < \alpha$ is satisfied at configuration (q_f, c_f, x_f), or $x = \alpha$ is satisfied at some configuration of ρ' between (q', c', x') and (q_f, c_f, x_f). The second transformation of $\mathcal{A}$ works as follows (see Figure 4).

We add to Q' a copy $(q, \overline{0})$ of any state $(q, 0)$ together with a copy of $\mathcal{L}(q, 0)$ and $\mathcal{I}(q, 0)$. Any initial state $(q, 0)$ is replaced by the initial state $(q, \overline{0})$. Any final state $(q, 0)$ has its copy $(q, \overline{0})$. To any edge $((q, 0), \tau, (q', 0)) \in E'$, we add a copy $((q, \overline{0}), \tau, (q', \overline{0}))$ and another copy $((q, 0), \tau, (q', \overline{0}))$ with label $x = \alpha$. We suppress $x < \alpha$ from any state $(q, 1) \notin F'$ that contains it and we add $x < \alpha$ to any state

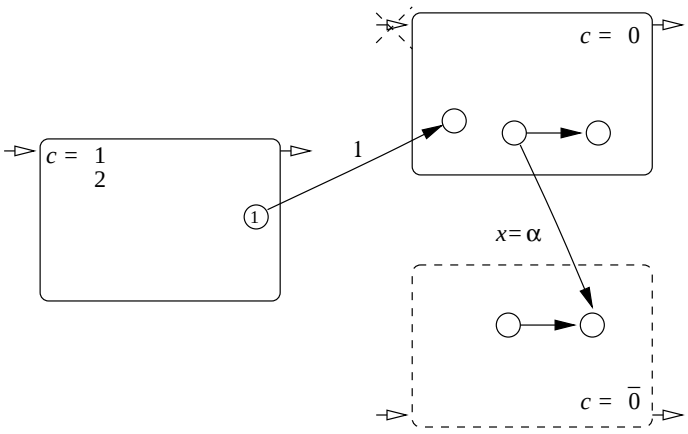

Figure 4. Automaton for x-atoms $x < \alpha$

$(q, 0) \in F'$.

Fourth step x-atoms $x = \alpha$.

At this point of the proof, state q only contains x-atoms of the form $x = \alpha$ if $q \notin I \cup F$, and of the form $x = \alpha$ or $x < \alpha$ if $q \in F$. Hence Property (1) of Proposition 25 holds. Let us prove Property (2). Suppose that $\mathcal{I}_x(q) = \bigwedge_{\alpha \in A}(x = \alpha)$ for some set A of linear terms. Let $\alpha' \in A$. Then $\mathcal{I}_x(q)$ is equivalent to

$$(x = \alpha') \wedge \bigwedge_{\alpha \in A} (\alpha' = \alpha).$$

Thus $\mathcal{I}_x(q)$ can be replaced by $x = \alpha'$ and $\mathcal{I}_\theta(q)$ by $\mathcal{I}_\theta(q) \wedge \bigwedge_{\alpha \in A}(\alpha' = \alpha)$. It remains to prove Property (3). Let ρ be a run in $\mathcal{R}(\mathcal{A}^v, x_0)$. Assume that there are several configurations (q_j, x_j), $1 \leq j \leq k$, in ρ such that q_j contains a given x-atom $x = \alpha$. Time does not progress from (q_1, x_1) to (q_k, x_k), that is, $x_j = x_1$ for all j. Only the first occurrence of $x = \alpha$ at state q_1 is useful, the next ones can be forgotten. Therefore, $\mathcal{A}$ is transformed in a way to count occurrences of $x = \alpha$ and to remember any progress of time. As before, a counter has value $0, 1$ or 2 in case of $0, 1$ or 2 and more occurrences of $x = \alpha$. Moreover, values 1 and 2 are indexed by $+$ if time has progressed since the first occurrence of $x = \alpha$. Formally we construct $\mathcal{A}' = (Q', I', F', E', \mathcal{L}', \mathcal{I}')$ where $Q' = Q \times \{0, 1, 1_+, 2, 2_+\}$, $F' = F \times \{0, 1, 1_+, 2, 2_+\}$, $\mathcal{L}'(q, c) = \mathcal{L}(q)$ and $\mathcal{I}'(q, c) = \mathcal{I}(q)$. For any $q \in I$, state (q, c) belongs to I' with $c = 1$ if $x = \alpha$ belongs to q, and $c = 0$ otherwise. For any $(q, \tau, q') \in E$, edge $((q, c), \tau, (q', c'))$ belongs to E' where c' is computed according Table 1. Finally, for any state (q, c) containing $x = \alpha$, we suppress this state if $c = 2_+$, we suppress $x = \alpha$ from this state if $c = 2$. Indeed recall that counter 2 indicates that it is at least the second occurrence of $x = \alpha$. Since

$\tau \backslash c$	0	1	1_+	2	2_+
0	1	2	2_+	2	2_+
1	1	2_+	2_+	2_+	2_+

if q' contains $x = \alpha$

$\tau \backslash c$	0	1	1_+	2	2_+
0	0	1	1_+	2	2_+
1	0	1_+	1_+	2_+	2_+

otherwise

Table 1. Computation of c'

the presence of index $+$ means a progress of time since the first occurrence of $x = \alpha$, this x-term cannot be satisfied at state $(q, 2_+)$ and can be suppressed at state $(q, 2)$. □

4.2 Durations in Reset-Free Automata

In this subsection, we again make Hypothesis $(*)$. When $\mathcal{A}$ is a reset-free automaton that has been simplified according to proposition 25 , we are going to construct a Presburger formula describing all the possible durations of runs in $\mathcal{R}(\mathcal{A}^v, x_0)$, in other words a Presburger formula for $\bigcup_{i \in I, f \in F} \lambda_{i,f}^{x_0}$ (see Notation 17). This formula will impose constraints on durations t_0 like $t_0 = b$, $t_0 \geq_a b$, $t_0 = b - x_0$, $t_0 \geq_a b - x_0$ and $t_0 < b - x_0$, with $a, b \in \mathbb{N}$.

Notation 26 Let t be a variable used to denote a duration and x be a variable for a clock value. We call *t-atom* any $t \sim \alpha$ or $t \sim \alpha - x$. A t-atom is of *first type* if is of the form

$$\begin{array}{l} t = \alpha, \\ t \geq_a \alpha, \\ t = \alpha - x, \\ t \geq_a \alpha - x, \end{array}$$

it is of *second type* if it of the form

$$t < \alpha - x.$$

A *t-formula* is either of first type, or of second type. A *first type* t-formula is an t-atom of first type, a *second type* t-formula is a conjunction of t-atoms of second type.

Proposition 27 *Let $\mathcal{A} = (Q, I, F, E, \mathcal{L}, \mathcal{I})$ be a simplified automaton. There exists a Presburger formula $\lambda(t, x, \Theta)$ such that for any valuation v and any clock value x_0, $\lambda(t_0, x_0, v(\Theta))$ is true iff there exists a run in $\mathcal{R}(\mathcal{A}^v, x_0)$ with duration t_0. This formula is a disjunction of formulae of the form*

$$\lambda_t \wedge \lambda_< \wedge \lambda_x \wedge \lambda_\theta,$$

that is, a conjunction of a first type t-formula λ_t, a second type t-formula $\lambda_<$, an x-formula λ_x and a θ-formula λ_θ. Its construction is effective.

Proof In this proof, as already done before, we often view an x-formula as a set of x-atoms and conversely.

(1) We can suppose that I is reduced to one initial state i and F to one final state f. At the end of the proof, it will remain to take a disjunction over $i \in I$ and $f \in F$ of the constructed formulae. From now on, we suppose that $I = \{i\}$ and $F = \{f\}$.

(2) *Assumption*. We first make the assumption that i contains no x-atom and f contains no x-atom $x < \alpha$. As $\mathcal{A}$ is simplified, this means that for any state $q \in Q$, either $\mathcal{I}_x(q) = \top$ or $\mathcal{I}_x(q)$ equals some $x = \alpha$. The proof is done by induction on the x-atoms $x = \alpha$ that appear as $\mathcal{I}_x(q)$ with $q \in Q$. Let us show how to construct $\lambda(t, x, \Theta)$. It will have *no* t-formula of second type due to the assumption.

Base case. Suppose that $\mathcal{I}_x(q) = \top$ for all $q \in Q$, that is $\mathcal{I}(q) = \mathcal{I}_\theta(q)$. Durations of runs in $\mathcal{R}(\mathcal{A}^v, x_0)$ are thus independent on the clock values. They are simply equal to the number of edges labeled by $\tau = 1$ along runs from i to f. And to any of these runs is associated the set of θ-atoms contained in the states of the run. Therefore the proof consists in a kind of determinization [LP98] of $\mathcal{A}$ that has to take into account the formulae $\mathcal{I}_\theta(q)$. The alphabet is reduced to one letter $\tau = 1$, while label $\tau = 0$ is understood as the empty word ϵ. Formally, let $\mathcal{T}$ be the set of θ-atoms that belong to the states of Q. Define $\mathcal{Q} = Q \times 2^{\mathcal{T}}$. For $(q, T) \in \mathcal{Q}$, we define Closure(q, T) as the set of $(q', T') \in \mathcal{Q}$ such that there exists a run ρ from q to q' with duration 0 and $T' = T \cup \bigcup_{p \in \rho} \mathcal{I}_\theta(p)$.

We define a classical [LP98] automaton $\mathcal{B}$ with $2^{\mathcal{Q}}$ as set of states, one initial state $\mathcal{P}_0$ equal to Closure$(i, \mathcal{I}_\theta(i))$ and set of final states equal to

$$\{\mathcal{F} \subseteq \mathcal{Q} \mid (f, T) \in \mathcal{F} \text{ for some } T\}.$$

Any edge is defined as $(\mathcal{P}, 1, \mathcal{P}')$ with $\mathcal{P}, \mathcal{P}' \subseteq \mathcal{Q}$, $\mathcal{P}' =$ Closure$(\mathcal{S})$ such that

$$\mathcal{S} = \{(q', T') \mid (q, T) \in \mathcal{P}, (q, 1, q') \in E, T' = T \cup \mathcal{I}_\theta(q')\}.$$

There exists a path in $\mathcal{B}$ from $\mathcal{P}_0$ to state $\mathcal{P}$ iff there exists a run ρ in $\mathcal{A}$ from i to q such that $(q, T) \in \mathcal{P}$ and T is the set of x-atoms contained in the states of ρ. Moreover, the duration of ρ is the length of the corresponding path in $\mathcal{B}$. As the alphabet is the one-letter alphabet $\{1\}$, automaton $\mathcal{B}$ has the special structure of a "frying pan" automaton (see Figure 5). This shows that formula $\lambda(t, x, \Theta)$ is a disjunction of formulae $\lambda_t \wedge \lambda_\theta$ constructed in the following way. Let $\mathcal{F}$ be a final state of $\mathcal{B}$, let $(f, T) \in \mathcal{F}$. If $\mathcal{F}$ does not belong to the cycle of $\mathcal{B}$, then $\lambda_\theta = T$ and λ_t equals $t = b$ where b is the length of the path from $\mathcal{P}_0$ to $\mathcal{F}$. If $\mathcal{F}$ belongs to the cycle of $\mathcal{B}$, then $\lambda_\theta = T$ and λ_t equals $t \geq_a b$ where b is the length of the shortest path from $\mathcal{P}_0$ to $\mathcal{F}$ and a is the length of the cycle of $\mathcal{B}$.

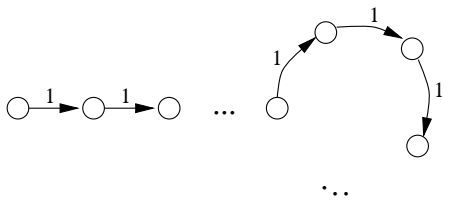

Figure 5. Frying pan automaton

General case. Now consider a particular x-atom $x = \alpha$. Let us denote by P the set of states q such that $\mathcal{I}_x(q)$ is equal to $x = \alpha$. As $\mathcal{A}$ is simplified, any run ρ of $\mathcal{R}(\mathcal{A}^v, x_0)$ contains 0 or 1 state of P. We are going to prove that the expected formula $\lambda(t, x, \Theta)$ is equal to

$$\lambda_{Q \setminus P}(t, x, \Theta) \vee \bigvee_{p \in P} \lambda_p(t, x, \Theta)$$

where $\lambda_{Q \setminus P}$ describes durations of runs containing no state of P, and λ_p describes durations of runs containing exactly state p of P.

All runs containing no state of P constitute set $\mathcal{R}(\mathcal{A}'^v, x_0)$ of an automaton $\mathcal{A}'$ obtained from $\mathcal{A}$ by erasing all states in P. As $\mathcal{A}'$ has one x-atom less, $\lambda_{Q \setminus P}(t, x, \Theta)$ can be constructed by induction hypothesis.

Let us now fix $p \in P$ and a run $\rho \in \mathcal{R}(\mathcal{A}^v, x_0)$ that contains it. This run is decomposed into a run $\rho_1 = (i, x_0) \rightsquigarrow (p, x_1)$ with duration t_1 and a run $\rho_2 = (p, x_1) \rightsquigarrow (f, x_2)$ with duration t_2. Duration t_0 of ρ is equal to $t_1 + t_2$ such that $x_1 = x_0 + t_1$, $x_2 = x_1 + t_2$ and x_1 satisfies $x = \alpha$. Durations t_1 and t_2 can be computed by induction in the following way.

Let us begin with t_1. Automaton $\mathcal{A}$ is modified into $\mathcal{A}_{p,1}$ by erasing states of $P \setminus \{p\}$ and edges leaving p. Formula $\mathcal{I}_x(p)$ is replaced by $\top$. The unique final state is p. The new automaton has one x-atom less, so $\lambda_{\mathcal{A}_{p,1}}(t, x, \Theta)$ can be constructed by induction hypothesis such that $\lambda_{\mathcal{A}_{p,1}}(t_1, x_0, v(\Theta))$ is true. Recall that $\lambda_{\mathcal{A}_{p,1}}$ is a disjunction of formulae $\lambda_{1,t} \wedge \lambda_{1,x} \wedge \lambda_{1,\theta}$ where $\lambda_{1,t}$ is a first type t-atom, $\lambda_{1,x}$ is an x-formula and $\lambda_{1,\theta}$ is a θ-formula. Suppose that $\lambda_{1,t}$ is one among

$$t = \alpha_1, \quad t \geq_a \alpha_1, \quad t = \alpha_1 - x, \quad t \geq_a \alpha_1 - x.$$

As x_1 satisfies $x = \alpha$ and $x_1 = x_0 + t_1$, then

$$x_1 = v(\alpha), \quad t_1 = v(\alpha) - x_0.$$

So $\lambda_{1,t}$ must be replaced by one among the formulae

$$\alpha - x = \alpha_1, \quad \alpha - x \geq_a \alpha_1, \quad \alpha = \alpha_1, \quad \alpha \geq_a \alpha_1.$$

It becomes a conjunction of x-atoms and θ-atoms. The modified formula $\lambda_{1,t} \wedge \lambda_{1,x} \wedge \lambda_{1,\theta}$ is denoted by $\lambda'_{1,x} \wedge \lambda'_{1,\theta}$.

Let us now describe t_2. We modify $\mathcal{A}$ into $\mathcal{A}_{p,2}$ by erasing states of $P \setminus \{p\}$ and edges entering p. Formula $\mathcal{I}_x(p)$ is replaced by $\top$. The new unique initial state is p. By induction hypothesis, $\lambda_{\mathcal{A}_{p,2}}(t,x,\Theta)$ is constructed as a disjunction of formulae $\lambda_{2,t} \wedge \lambda_{2,x} \wedge \lambda_{2,\theta}$ where $\lambda_{2,t}$ is one among

$$t = \alpha_2, \quad t \geq_a \alpha_2, \quad t = \alpha_2 - x, \quad t \geq_a \alpha_2 - x.$$

Formula $\lambda_{\mathcal{A}_{p,2}}(t,x,\Theta)$ is true with $(t_2, x_1 = v(\alpha), v(\Theta))$. Thus any $\lambda_{2,t}$ in $\lambda_{\mathcal{A}_{p,2}}$ is one among the formulae

$$t = \alpha_2, \quad t \geq_a \alpha_2, \quad t = \alpha_2 - \alpha, \quad t \geq_a \alpha_2 - \alpha,$$

that is t-atoms of the form

$$t = \beta \quad \text{or} \quad t \geq_a \beta.$$

Moreover any $\lambda_{2,x}$ in $\lambda_{\mathcal{A}_{p,2}}$ becomes a θ-formula. The modified formula $\lambda_{2,x} \wedge \lambda_{2,\theta}$ is denoted by $\lambda'_{2,\theta}$.
Finally, we can describe $t_0 = t_1 + t_2$. It has the form

$$t_0 = v(\alpha) - x_0 + v(\beta) \quad \text{or} \quad t_0 \geq_a v(\alpha) - x_0 + v(\beta).$$

Hence formula $\lambda_p(t,x,\Theta)$ for t_0 is a disjunction of formulae $\lambda_t \wedge \lambda_x \wedge \lambda_\theta$ such that λ_t has the form

$$t = \alpha - x + \beta \quad \text{or} \quad t \geq_a \alpha - x + \beta$$

and $\lambda_x \wedge \lambda_\theta$ has the form

$$\lambda'_{1,x} \ \wedge \ \lambda'_{1,\theta} \ \wedge \ \lambda'_{2,\theta}.$$

(3) Under the assumption that i contains no x-atoms and f contains no x-atom $x < \alpha$, we have constructed a formula $\lambda(t,x,\Theta)$ with no t-formula of second type. So we have to take into account x-formula $\mathcal{I}_x(i)$ and x-atoms $x < \alpha$ appearing in f. Thus x_0 must satisfy $\mathcal{I}_x(i)$ and $x_0 + t_0$ must satisfy any $x < \alpha$ in f. It follows that the final formula is equal to

$$\lambda(t,x,\Theta) \wedge \mathcal{I}_x(i) \wedge \bigwedge_{x < \alpha \in f} t < \alpha - x.$$

□

4.3 Runs and Durations

This subsection is devoted to the proofs of Propositions 16 and 18. Here $\mathcal{A}$ is *any* automaton $(Q, E, \mathcal{L}, \mathcal{I})$. Without lost of generality, we can suppose that $\mathcal{A}$ is *state-reset*, that is for any q of $\mathcal{A}$, the edges entering the state q either all reset clock x or none resets it, so clock x can be seen to be reset at states instead of along edges (see Lemma 28 below). Thus in the sequel we will consider the set E of edges as a subset of $Q \times \{0,1\} \times \mathcal{G} \times Q$ and we call *reset-state* any state that resets the clock.

Given two states q, q', a valuation v and a clock value x_0, a run $\rho = (q, x_0) \rightsquigarrow (q', \cdot)$ in $\mathcal{A}^v$ possibly contains some reset-states. It thus decomposes as a sequence of $k \geq 1$ runs ρ_j, $1 \leq j \leq k$, such that for any j, ρ_j contains no reset-state, except for the first and the last configurations of ρ_j. The duration D_{ρ_j} of each ρ_j can be computed thanks to Proposition 27 with the initial clock value equal to x_0 if $j = 1$ and equal to 0 if $j > 1$. For any j, $1 \leq j \leq k$, let us denote by $\lambda_j(t,x,\Theta)$ the Presburger formula corresponding to D_{ρ_j} which is a disjunction of formulae

$$\lambda_t \wedge \lambda_< \wedge \lambda_x \wedge \lambda_\theta.$$

So the total duration D_ρ is equal to the sum $\Sigma_{1 \leq j \leq k} \mathrm{D}_{\rho_j}$. Durations D_ρ of runs $\rho = (q, x_0) \rightsquigarrow (q', \cdot)$, i.e. set $\lambda^{x_0}_{q,q'}$, can be symbolically represented thanks to rational expressions on an alphabet whose letters are formulae $\lambda_t \wedge \lambda_< \wedge \lambda_x \wedge \lambda_\theta$ that appear in the $\lambda_j(t,x,\Theta)$'s. Thanks to this symbolic description of $\lambda^{x_0}_{q,q'}$, we are able to prove Propositions 16 and 18. Let us explain in details all these ideas.

Lemma 28 *Any automaton $\mathcal{A}$ can be effectively transformed into a state-reset automaton.*

Proof The proof is similar to the proof of Lemma 23. If two edges enter a given state p, with one resetting the clock and the other not, p is splitted into two states and the edges entering p are redirected to one or the other copy according to the fact that x is reset or not. We can thus suppose that the edges entering p all reset the clock or none does reset it. Resetting the clock can thus be done at state p instead of along an edge entering p. □

Let $\mathcal{A} = (Q, E, \mathcal{L}, \mathcal{I})$ be a state-reset automaton with $E \subseteq Q \times \{0,1\} \times \mathcal{G} \times Q$ and $R \subseteq Q$ the set of its reset-states. Let us fix two states q, q', a parameter valuation v, a clock value x_0 and let us study all the runs $(q, x_0) \rightsquigarrow (q', \cdot)$ in $\mathcal{A}^v$.

First construction For each couple (p, p') of states of $\mathcal{A}$ such that $p \in \{q\} \cup R$ and $p' \in \{q'\} \cup R$, we construct from $\mathcal{A}$ the following reset-free automaton $\mathcal{A}_{p,p'} = (Q', I', F', E', \mathcal{L}', \mathcal{I}')$ with initial and final states. The set Q' of states is $(Q \setminus R) \cup \{\overline{p}, \overline{p}'\}$ where $\overline{p}, \overline{p}'$ are copies of p, p' with $\mathcal{L}'(\overline{p}) = \mathcal{L}(p)$, $\mathcal{L}'(\overline{p}') = \mathcal{L}(p')$, $\mathcal{I}'(\overline{p}) = \mathcal{I}(p)$ and $\mathcal{I}'(\overline{p}') = \mathcal{I}(p')$. States $\overline{p}, \overline{p}'$ are non reset. The set E' of edges is the union of E restricted to $Q \setminus R$ with the next set of copied edges

$$\begin{array}{lll} (\overline{p}, \tau, g, p_1) & \text{with} & (p, \tau, g, p_1) \in E \\ (p_1, \tau, g, \overline{p}') & \text{with} & (p_1, \tau, g, p') \in E \\ (\overline{p}, \tau, g, \overline{p}') & \text{with} & (p, \tau, g, p') \in E. \end{array}$$

The unique initial state is $\overline{p}$ and the unique final state is $\overline{p}'$. Automaton $\mathcal{A}_{p,p'}$ satisfies Hypothesis $(*)$. Let x_1 be a clock

value such that $x_1 = 0$ if p is a reset-state, $x_1 = x_0$ otherwise. The runs of $\mathcal{R}(\mathcal{A}^v_{p,p'}, x_1)$ are exactly the non-empty runs $(p, x_1) \rightsquigarrow (p', \cdot)$ of $\mathcal{A}^v$ that pass through no reset-state (except possibly the first and the last state of the run). By Proposition 27, the durations of runs in $\mathcal{R}(\mathcal{A}^v_{p,p'}, x_1)$ are described by formula $\lambda_{p,p'}(t, x, \Theta)$ equal to some disjunction $\bigvee_j \lambda^{p,p',j}$ where

$$\lambda^{p,p',j} = \lambda_t^{p,p',j} \wedge \lambda_<^{p,p',j} \wedge \lambda_x^{p,p',j} \wedge \lambda_\theta^{p,p',j}.$$

Note that in formula $\lambda_{p,p'}(t, x, \Theta)$, if p is a reset-state, then *necessarily* $x = 0$. Hence, due to instantiation of value 0 to variable x, any formula $\lambda^{p,p',j}$ has no longer free variable x, $\lambda_t^{p,p',j}$ is one of the t-atoms $t = \alpha$ or $t \geq_a \alpha$, $\lambda_<^{p,p',j}$ is a conjunction of t-atom of the form $t < \alpha$, and $\lambda_x^{p,p',j}$ becomes a θ-formula.

For each couple (p, p') and each j, we associate a distinct letter $b_{p,p',j}$ to each formula $\lambda^{p,p',j}$. The set of all these letters is denoted by B. We say that letter $b_{p,p',j}$ is a *reset-letter* if p is a reset-state. The set of reset-letters is denoted B_R.

Second construction We construct a classical automaton $\mathcal{B}$ over the alphabet B as follows. The set of states equals $R \cup \{q, q'\}$ and the set of edges equals $\{(p, b, p') \mid b = b_{p,p',j} \text{ for some } j\}$. The unique initial (resp. final) state is q (resp. q'). A path in $\mathcal{B}$ from q to q' indicates how a run $\rho = (q, x_0) \rightsquigarrow (q', \cdot)$ in $\mathcal{A}^v$ is decomposed according to reset-states of $\mathcal{A}$. The duration of such a run ρ can symbolically be viewed as a word over B that labels the corresponding path in $\mathcal{B}$. Set $\lambda^{x_0}_{q,q'}$ is thus symbolically represented as a rational set over B that we denote by $L_{q,q'}$. It is important to note that any word of $L_{q,q'}$ has *at most one* letter that is non reset (the first letter of the word). If q is a reset-state, then $L_{q,q'} \subseteq B_R^*$, otherwise $L_{q,q'} \subseteq (B \setminus B_R) \cdot B_R^* \cup \{\epsilon\}$.

Rational expressions The concatenation of two letters b_{p_1,p_1',j_1} and b_{p_2,p_2',j_2} of B is interpreted as follows. It is the *sum* $t_1 + t_2$ of durations t_1 and t_2 described by $\lambda_t^{p_1,p_1',j_1} \wedge \lambda_<^{p_1,p_1',j_1}$ and $\lambda_t^{p_2,p_2',j_2} \wedge \lambda_<^{r_2,r_2',l_2}$. It is the *conjunction* of x-formulae and θ-formulae described by $\lambda_x^{p_1,p_1',j_1} \wedge \lambda_\theta^{p_1,p_1',j_1}$ and $\lambda_x^{p_2,p_2',j_2} \wedge \lambda_\theta^{p_2,p_2',j_2}$.

Let L^+ be denoting $L^* \setminus \{\epsilon\}$ and $\mathrm{Rat}_B(\cdot, +)$ be the smallest family closed under $\cdot$ and $+$, and containing B. As sum and conjunction operations are *commutative*, it is not difficult to prove that any rational language over B can be effectively rewritten as a finite union of languages in $\{\epsilon\} \cup \mathrm{Rat}_B(\cdot, +)$. Therefore

$$L_{q,q'} = \bigcup_k L_k$$

with $L_k = \{\epsilon\}$ or $L_k \in \mathrm{Rat}_B(\cdot, +)$. More precisely, if q is a reset-state, then

$$L_k \in \{\epsilon\} \cup \mathrm{Rat}_{B_R}(\cdot, +),$$

otherwise, if $L_k \neq \{\epsilon\}$,

$$L_k = b_k \cdot L_k' \quad \text{with} \quad b_k \in B \setminus B_R, L_k' \in \{\epsilon\} \cup \mathrm{Rat}_{B_R}(\cdot, +).$$

We are now fully equiped to prove Proposition 16 and Proposition 18.

Proof of Proposition 16. Let us prove that one can construct a $\mathcal{B}_{x,\Theta}$ formula $R_q(x, \Theta)$ such that there exists an infinite run starting with (q, x_0) iff $R_q(x_0, v(\Theta))$ is true. Such a run exists iff for some $q' \in Q$, set $\lambda^{x_0}_{q,q'}$ contains arbitrarily large durations t. We say that $\lambda^{x_0}_{q,q'}$ (or equivalently $L_{q,q'}$) is *non Zeno* if this property holds. By induction on the rational expression defining $L_{q,q'}$, we are going to construct a formula $\mathrm{NZ}_{L_{q,q'}}(x, \Theta) \in \mathcal{B}_{x,\Theta}$ such that $\mathrm{NZ}_{L_{q,q'}}(x_0, v(\Theta))$ is true iff $L_{q,q'}$ is non Zeno. Thus formula $R_q(x, \Theta)$ is equal to $\bigvee_{q' \in Q} \mathrm{NZ}_{L_{q,q'}}(x, \Theta)$.

As $L_{q,q'}$ is a finite union $\bigcup_k L_k$ of sets $L_k \in \{\epsilon\} \cup \mathrm{Rat}_B(\cdot, +)$, it is non Zeno iff one among the L_k's is non Zeno. Thus $\mathrm{NZ}_{L_{q,q'}} = \bigvee_k \mathrm{NZ}_{L_k}$. In case $L_k = \{\epsilon\}$, then clearly $\mathrm{NZ}_{L_k} = \bot$. Let $L \in \mathrm{Rat}_B(\cdot, +)$. Let $L = \{b_{p,p',j}\}$ for some $b_{p,p',j} \in B$. Recall that $\lambda^{p,p',j}$ is a conjunction $\lambda_t \wedge \lambda_< \wedge \lambda_x \wedge \lambda_\theta$ between a t-atom of first type, a t-formula of second type, an x-formula and a θ-formula. If λ_t equals $t = \alpha$ or $t = \alpha - x$, then $\mathrm{NZ}_L = \bot$. If λ_t equals $t \geq_a \alpha$ or $t \geq_a \alpha - x$, then t is arbitrarily large if $\lambda_< = \top$. In this case, $\mathrm{NZ}_L = \lambda_x \wedge \lambda_\theta$, otherwise $\mathrm{NZ}_L = \bot$. Suppose now that $L = L_1 \cdot L_2$, then L is non Zeno if L_1 or L_2 is non Zeno, that is $\mathrm{NZ}_L = \mathrm{NZ}_{L_1} \vee \mathrm{NZ}_{L_2}$. If $L = L_1^+$, then L is non Zeno if L_1 is *evolutive*, that is, contains a non null duration. This property is studied hereafter.

Let us study when $L \in \mathrm{Rat}_B(\cdot, +)$ is evolutive and denote by $\mathrm{E}_L(x, \Theta)$ the related formula of $\mathcal{B}_{x,\Theta}$. Consider case $L = \{b_{p,p',j}\}$ for some $b_{p,p',j} \in B$. As above let us study formulae λ_t and $\lambda_<$. Suppose that $\lambda_< = \bigwedge_\beta (t < \beta - x)$. If λ_t equals $t = \alpha$, then t is non null iff $\alpha > 0$. Then E_L is the formula of $\mathcal{B}_{x,\Theta}$ equal to $\alpha > 0 \wedge \bigwedge_\beta (\alpha < \beta - x) \wedge \lambda_x \wedge \lambda_\theta$. When λ_t is $t = \alpha - x$, we have a similar formula such that α is replaced by $\alpha - x$. If λ_t equals $t \geq_a \alpha$, then a possible non null value for t is either α if $\alpha > 0$ or a if $\alpha = 0$. We get formula E_L of $\mathcal{B}_{x,\Theta}$ equal to $\Big((\alpha > 0 \wedge \bigwedge_\beta (\alpha < \beta - x)) \vee (\alpha = 0 \wedge \bigwedge_\beta (a < \beta - x))\Big) \wedge \lambda_x \wedge \lambda_\theta$. A similar argument holds if $t \geq_a \alpha - x$. Finally if $X = L_1 \cdot L_2$, then $\mathrm{E}_L = \mathrm{E}_{L_1} \vee \mathrm{E}_{L_2}$, and if $L = L_1^+$, then $\mathrm{E}_L = \mathrm{E}_{L_1}$. □

Proof of Proposition 18. Let γ be a linear term and $\sim \in \{<, \leq, >, \geq\}$. We have to show that there exists a formula $\lambda^{\sim\gamma}_{q,q'}(x, \Theta)$ of $\mathcal{B}_{x,\Theta}$ that describes the existence of a duration $t \sim v(\gamma)$ in set $\lambda^{x_0}_{q,q'}$.

(1) We begin with $\sim \in \{<, \leq\}$. To test if there exists $t \sim v(\gamma)$ in $\lambda^{x_0}_{q,q'}$ is equivalent to test that $t_{min} \sim v(\gamma)$ with t_{min} being the *minimum* duration of $\lambda^{x_0}_{q,q'}$.

We first suppose that q is a reset-state. Let $L_{q,q'} = \bigcup_k L_k$ with sets L_k in $\{\epsilon\} \cup \text{Rat}_{B_R}(\cdot, +)$. By induction on the rational expression defining any L_k, let us construct a Presburger formula $\text{Min}_{L_k}(t, \Theta)$ that describes the minimum duration t in L_k. It has no free variable x since q is a reset-state. This formula will be a conjunction

$$\mu_t \wedge \mu_< \wedge \mu_\theta$$

where μ_t is a t-atom of the form $t = \alpha$, $\mu_<$ is a t-formula of the form $\bigwedge_\beta t < \beta$ and μ_θ is a θ-formula. Therefore formula $\lambda^{\sim\gamma}_{q,q'}$ is equal to $\bigvee_k \lambda_k$, such that each formula λ_k is obtained by modifying Min_{L_k} as follows. Any formula $\mu_t \wedge \mu_<$ equal to $(t = \alpha) \wedge \bigwedge_\beta (t < \beta)$ is replaced by formula $(\alpha \sim \gamma) \wedge \bigwedge_\beta (\alpha < \beta)$. The resulting formula λ_k belongs to $\mathcal{B}_{x,\Theta}$.
In case $L_k = \{\epsilon\}$, then $q = q'$. Thus Min_{L_k} is equal to $(t = 0) \wedge \mathcal{I}(q)$ such that x is instantiated to 0 inside $\mathcal{I}(q)$. Let $L \in \text{Rat}_{B_R}(\cdot, +)$. Consider case $L = \{b_{p,p',j}\}$ for some reset-letter $b_{p,p',j} \in B_R$. Suppose that $\lambda^{p,p',j} = \lambda_t \wedge \lambda_< \wedge \lambda_\theta$. As t-atom λ_t is either $t = \alpha$ or $t \geq_a \alpha$, then the minimum duration equals α. So formula Min_L is equal to $(t = \alpha) \wedge \lambda_< \wedge \lambda_\theta$. If $L = L_1 \cdot L_2$, then the minimum duration in L equals the sum of the minimum durations in L_1 and L_2. Let $\text{Min}_{L_1} = \mu^1_t \wedge \mu^1_< \wedge \mu^1_\theta$ where μ^1_t equals $t = \alpha_1$ and $\mu^1_<$ equals $\bigwedge_{\beta_1}(t < \beta_1)$. Let $\text{Min}_{L_2} = \mu^2_t \wedge \mu^2_< \wedge \mu^2_\theta$ where μ^2_t equals $t = \alpha_2$ and $\mu^2_<$ equals $\bigwedge_{\beta_2}(t < \beta_2)$. Thus Min_L is equal to $(t = \alpha_1 + \alpha_2) \wedge \bigwedge_{\beta_1,\beta_2}(t < \beta_1 + \beta_2) \wedge \mu^1_\theta \wedge \mu^2_\theta$. If $L = L_1^+$, then the minimum duration in L is the minimum duration in L_1, i.e. $\text{Min}_L = \text{Min}_{L_1}$.

We now suppose that q is not a reset-state. Let $L_{q,q'} = \bigcup_k L_k$ such that $L_k = \{\epsilon\}$ or $L_k = b_k \cdot L'_k$ with $L'_k \in \{\epsilon\} \cup \text{Rat}_{B_R}(\cdot, +)$. For each k, the Presburger formula $\text{Min}_{L_k}(t, x, \Theta)$ that we want to construct has a slightly different form. It has now free variable x, it is a conjunction

$$\mu_t \wedge \mu_< \wedge \mu_x \wedge \mu_\theta$$

where μ_t is a t-atom of the form $t = \alpha$ or $t = \alpha - x$, $\mu_<$ is a t-formula of the form $\bigwedge_\beta (t < \beta - x)$, μ_x is an x-formula and μ_θ is a θ-formula. Consequently, formula $\lambda^{\sim\gamma}_{q,q'}$ will now be equal to $\bigvee_k \lambda_k$, such that each formula λ_k is obtained by modifying Min_{L_k} as follows. Any formula $\mu_t \wedge \mu_<$ respectively equal to

$$(t = \alpha) \wedge \bigwedge_\beta (t < \beta - x) \quad \text{or} \quad (t = \alpha - x) \wedge \bigwedge_\beta (t < \beta - x)$$

is replaced by formula

$$(\alpha \sim \gamma) \wedge \bigwedge_\beta (\alpha < \beta - x) \quad \text{or} \quad (\alpha - x \sim \gamma) \wedge \bigwedge_\beta (\alpha < \beta - x).$$

If $L_k = \{\epsilon\}$, then $q = q'$ and Min_{L_k} is $(t = 0) \wedge \mathcal{I}(q)$. Let $L = b \cdot L'$ with $b \in B \setminus B_R$ and $L' \in \{\epsilon\} \cup \text{Rat}_{B_R}(\cdot, +)$. By the first part of the proof, the minimum duration of L' is expressed by formula $\text{Min}_{L'}(t, \Theta)$ equal to $\mu'_t \wedge \mu'_< \wedge \mu'_\theta$ with $\mu'_t = (t = \alpha')$ and $\mu'_< = \bigwedge_{\beta'}(t < \beta')$. For $b = b_{p,p',j}$, we have an associated formula $\lambda^{p,p',j} = \lambda_t \wedge \lambda_< \wedge \lambda_x \wedge \lambda_\theta$ such that t-atom λ_t is one among $t = \alpha$, $t = \alpha - x$, $t \geq_a \alpha$, $t \geq_a \alpha - x$, and $\lambda_<$ is $\bigwedge_\beta (t < \beta - x)$. The minimum duration for $\lambda^{p,p',j}$ is $t = \alpha$ or $t = \alpha - x$. Hence for concatenation $L = b \cdot L'$, we have formula $\text{Min}_L = \mu_t \wedge \mu_< \wedge \mu_x \wedge \mu_\theta$ such that μ_t is $t = \alpha + \alpha'$ or $t = \alpha + \alpha' - x$, $\mu_<$ is $\bigwedge_{\beta,\beta'}(t < \beta + \beta' - x)$, μ_x is λ_x and μ_θ is $\lambda_\theta \wedge \mu'_\theta$.

(2) We now turn to $\sim \in \{>, \geq\}$. The approach is similar but with the *maximum* (instead of minimum) duration. Again we consider the two cases : q is or is not a reset-state.

Suppose that q is a reset-state. Let $L_{q,q'} = \bigcup_k L_k$ with any L_k in $\{\epsilon\} \cup \text{Rat}_{B_R}(\cdot, +)$. By induction on the rational expression defining any L_k, we will construct a formula $\text{Max}_{L_k}(t, \Theta)$ (without free variable x) that describes the maximum duration t in L_k. Note that when L_k is non Zeno (see proof of Proposition 16), durations t in L_k can be arbitrarily large. We will thus denote symbolically by $t = \infty$ the (non existing) maximum duration. Formula Max_{L_k} will be a disjunction of conjunctions

$$M_t \wedge M_\geq \wedge M_\theta$$

such that M_t is either $t = \alpha$ or $t = \infty$, $M_\geq$ is either $t \geq \beta$ or $\top$ (equivalently $t \geq \beta$ with $\beta = 0$) and M_θ is a θ-formula. We will get the next formula $\lambda^{\sim\gamma}_{q,q'}$ equal to $\bigvee_k \lambda_k$ where each formula λ_k is obtained by modifying Max_{L_k} in the following way. If M_t equals $t = \infty$, then λ_k equals M_θ. If M_t equals $t = \alpha$, then λ_k equals $(\alpha \sim \gamma) \wedge (\alpha \geq \beta) \wedge M_\theta$. If $L_k = \{\epsilon\}$, then Max_{L_k} is $(t = 0) \wedge \mathcal{I}(q)$ such that $x = 0$ in $\mathcal{I}(q)$. Let $L \in \text{Rat}_{B_R}(\cdot, +)$. Let us study case $L = \{b_{p,p',j}\}$ with $b_{p,p',j} \in B_R$ a reset-letter. Let $\lambda^{p,p',j} = \lambda_t \wedge \lambda_< \wedge \lambda_\theta$ the associated formula. Let us study λ_t and $\lambda_< = \bigwedge_\beta (t < \beta)$. If λ_t is $t = \alpha$, then Max_L equals $\lambda_t \wedge \bigwedge_\beta (\alpha < \beta) \wedge \lambda_\theta$. If λ_t is $t \geq_a \alpha$ with $\lambda_< = \top$, then Max_L equals $(t = \infty) \wedge \lambda_\theta$. Finally, suppose that λ_t is $t \geq_a \alpha$ with $\lambda_<$ being a non empty conjunction $\bigwedge_\beta (t < \beta)$. Then the maximum duration is the greatest value $\alpha + ay$, for some $y \in \mathbb{N}$, which is strictly less than the smallest β, denoted by β'. Assume that $\beta' \equiv b \mod a$ and $\alpha \equiv c \mod a$ for some $b, c \in \{0, \cdots, a-1\}$. If $b \geq c$, then the maximum duration is given by formula M_t equal to $t = \beta' - (b - c)$ with condition $M_\geq$ equal to $t \geq \alpha$. If $b < c$, then M_t equals $t = \beta' - (a + b - c)$ and $M_\geq$ equal to $t \geq \alpha$. Thus Max_L is a disjunction over the different possible values of β', b and c of formulae $M_t \wedge M_\geq \wedge \lambda_\theta \wedge M_{\beta',b,c}$ such that $M_{\beta',b,c}$ is the conjunction between $\bigwedge_\beta (\beta' \leq \beta)$, $\beta' \geq_a b$ and $\alpha \geq_a c$. Let us now treat case $L = L_1 \cdot L_2$. If Max_{L_1} contains a conjunction $(t = \alpha_1) \wedge (t \geq \beta_1) \wedge M_{\theta_1}$ and Max_{L_2} contains a conjunction $(t = \alpha_2) \wedge (t \geq \beta_2) \wedge M^2_\theta$,

then Max_L contains the conjunction $(t = \alpha_1 + \alpha_2) \wedge (t \geq \beta_1 + \beta_2) \wedge M^1_\theta \wedge M^2_\theta$. If $t = \infty$ in Max_{L_1} or $t = \infty$ in Max_{L_2}, then Max_L contains the conjunction $(t = \infty) \wedge M^1_\theta \wedge M^2_\theta$. Finally if $L = L_1^+$, then the maximum duration equals ∞ if L_1 is evolutive (see Proof of Proposition 16), and 0 otherwise. Thus Max_L is the formula $((t = \infty) \wedge \mathrm{E}_L) \vee ((t = 0) \wedge \neg \mathrm{E}_L)$. It has the right form because $\mathrm{E}_L \in \mathcal{B}_{x,\Theta}$ and by Lemma 11.

We now suppose that q is not a reset-state. Let $L_{q,q'} = \bigcup_k L_k$ such that $L_k = \{\epsilon\}$ or $L_k = b_k \cdot L'_k$ with $L'_k \in \{\epsilon\} \cup \mathrm{Rat}_{B_R}(\cdot, +)$. For each k, formula $\mathrm{Max}_{L_k}(t, x, \Theta)$ (with free variable x) is now a disjunction of conjunctions

$$M_t \wedge M_{\geq} \wedge M_x \wedge M_\theta$$

such that M_t is either $t = \alpha$ or $t = \alpha - x$ or $t = \infty$, $M_{\geq}$ is $t \geq \beta$, M_x is an x-formula and M_θ is a θ-formula. The requested formula $\lambda^{\sim\gamma}_{q,q'}$ is obtained in a way similar as done before.

If $L_k = \{\epsilon\}$, then $\mathrm{Max}_{L_k} = (t = 0) \wedge \mathcal{I}(q)$. Let $L = b \cdot L'$ with $b \in B \setminus B_R$ and $L' \in \{\epsilon\} \cup \mathrm{Rat}_{B_R}(\cdot, +)$. We already know that the maximum duration of L' can be expressed by a formula $\mathrm{Max}_{L'}(t, \Theta)$ equal to a disjunction of $M'_t \wedge M'_{\geq} \wedge M'_\theta$ with M'_t equal to $(t = \alpha')$ or $(t = \infty)$. For $b = b_{p,p',j}$, the maximum duration of $\lambda^{p,p',j} = \lambda_t \wedge \lambda_< \wedge \lambda_x \wedge \lambda_\theta$ can be studied as before, including two additional cases $t = \alpha - x$ and $t \geq_a \alpha - x$ for λ_t. It is not difficult to see that we obtain a formula $\mathrm{Max}_b(t, x, \Theta)$ which is a disjunction of formulae of the form $M_t \wedge M_{\geq} \wedge M_x \wedge M_\theta$. It remains to study the maximum duration in concatenation $L = b \cdot L'$. Again this can be done as before according to the different cases for M'_t and M_t. The resulting formula $\mathrm{Max}_L(t, x, \Theta)$ has the right form. □

5 Conclusion

In this paper, we have studied algorithmic solutions to the model-checking and parameter synthesis problems of the logic PTCTL over *one parametric clock* timed automata. On the negative side, we showed that the model-checking problem of TCTL extended with parameters is *undecidable* over timed automata with *only one* parametric clock. The undecidability result needs *equality in the logic*. On the positive side, we showed that when equality is not allowed in the logic, the model-checking problem becomes *decidable* and the parameter synthesis problem is solvable. Our algorithm is based on automata theoretic principles and an extension of our method (see [BDR02]) to express durations of paths of a timed automata using Presburger arithmetic.

To the best of our knowledge, this is the first work that studies the model-checking and parameter synthesis problems with parameters both in the model (timed automaton) and in the property (PTCTL formula). The problems solved in this paper are important as it is very natural to refer in the properties of the system to parameters appearing in the model of the system. We have illustrated in the introduction the kind of properties that can be expressed and automatically verified in our framework.

As future works, we plan to investigate if the method proposed here can be extended from discrete to dense timed models.

References

[AETP99] Rajeev Alur, Kousha Etessami, Salvatore La Torre, and Doron Peled. Parametric temporal logic for "model measuring". *Lecture Notes in Computer Science*, 1644:159–173, 1999.

[AHV93] R. Alur, T.A. Henzinger, and M.Y. Vardi. Parametric real-time reasoning. In *Proceedings of the 25th Annual Symposium on Theory of Computing*, pages 592–601. ACM Press, 1993.

[BDR02] V. Bruyère, E. Dall'olio, and J.-F. Raskin. Durations, parametric model-checking in timed automata with Presburger arithmetic. Technical Report CFV-2002-1, Centre Fédéré en Vérification (Belgique), 2002. `http://www.ulb.ac.be/di/ssd/cfv`, a short version to appear in Proc. of "Stacs'03".

[Bès02] Alexis Bès. A survey of arithmetical definability. *A tribute to Maurice Boffa, Special Issue of Belg. Math. Soc.*, pages 1–54, 2002.

[ET99] E. Allen Emerson and Richard J. Trefler. Parametric quantitative temporal reasoning. In *Logic in Computer Science*, pages 336–343, 1999.

[LP98] Harry Lewis and Christos Papadimitriou. *Elements of the theory of computation*. Prentice Hall, 1998.

[Wan95] Farn Wang. Timing behavior analysis for real-time systems. In *In Proceedings of the Tenth IEEE Symposium on Logic in Computer Science*, pages 112–122, 1995.

[WH97] Farn Wang and Pao-Ann Hsiung. Parametric analysis of computer systems. In *Algebraic Methodology and Software Technology*, pages 539–553, 1997.