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 discrete-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 discrete-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 [1] the analysis of timed automata where clocks are compared to parameters. They show 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. Hune et al study in [7] a subclass of parametric timed automata (L/U automata) such that each parameter occurs either as a lower

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

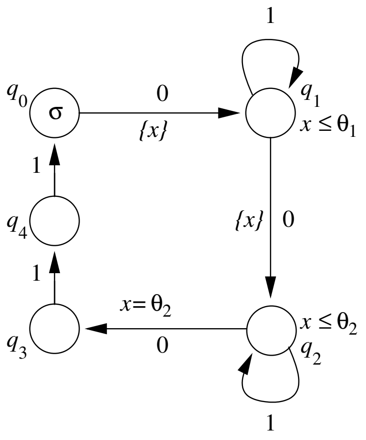

Figure 1: A parametric timed automaton

bound or as an upper bound. Wang in [9, 10], Emerson et al in [6], Alur et al in [2] and the authors of this paper in [4] 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 *discrete*-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 automaton. The automaton $\mathcal{A}$ of Figure 1 is a discrete-timed automaton with one clock x and two parameters θ_1 and θ_2. Here we explicitly model the elapse of time by transitions labeled by 0 or 1. State q_0 is labeled 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 to 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 $(q_0, 0)$, i.e. the control is in state q_0 and clock x has value 0:

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

(ii) $\forall\theta_1\forall\theta_2 \cdot (\theta_2 \leq \theta_1 \rightarrow \forall\Box(\sigma \rightarrow \forall\Diamond_{\leq 2\theta_1+2}\sigma))$

(iii) $\forall\theta_1 \cdot (\theta_1 \geq 5 \rightarrow \forall\Box(\sigma \rightarrow \forall\Diamond_{<2\theta_1+2}\sigma))$

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 $(q_0, 0)$. By observing the model and the formula, we can deduce the following constraint on the parameters: $\theta_3 \geq \theta_1 + \theta_2 + 2$. This means that any cycle through 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 $(q_0, 0)$. 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 $\theta_1 \geq 5$, a cycle through the four states lasts at most $2\theta_1 + 1$ time units". This is again a parameter synthesis problem and the answer is $\theta_2 \leq 4$.

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 [4]) to express durations of runs in a timed automaton using Presburger arithmetic. As a corollary, we obtain the decidability of the reachability problem for timed automata with one parametric clock proved by Alur et al in [1]. 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 to 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 [1] with a *discrete* time domain and *one* parametric clock. The logic is Parametric Timed CTL Logic as defined in [4].

Notation 1 Let Θ be a fixed finite set of *parameters* θ that are *shared* by the automaton and the logical formulae. A *parameter valuation* for Θ is a function $v : \Theta \to \mathbb{N}$ 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 $\{\theta_i | i \in I\} \subseteq \Theta$. 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 $\sim \in \{=, <, \leq, >, \geq\}$. 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 [1]. We make the hypothesis that non-parametric clocks have all been eliminated by a technique related to the region construction, see [1] 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*, $\mathcal{L} : Q \to 2^{\Sigma}$ is a *labeling* function and $\mathcal{I} : Q \to \mathcal{G}$ assigns an *invariant* $\mathcal{I}(q) \in \mathcal{G}$ 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'), with time increment $\tau \in \{0, 1\}$, 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 $r = \varnothing$.[2]

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 $\Sigma_{i \geq 0} \tau_i = \infty$.[3] 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 $\mathrm{D}_\rho(q', x')$.

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

[2]Note that time increment τ is first added to x, guard g is then tested, and finally x is reset according to r.

[3]Non Zenoness property.

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 of Θ 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 $\Theta_\varphi = \Theta$ (recall that Θ is the set of parameters that appear 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 parameter 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[4] 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 $j < i$

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

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 $\Theta_{f'}$ by $v' = v$ on Θ_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 $\Theta_{f'}$ by $v' = v$ on Θ_f and $v'(\theta) = c$.

2.3 Problems

The problems that we want to solve in this paper are the following 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 $(q,x) \models_v f$.[5]

Example We consider the example given in the introduction with the parametric timed automaton $\mathcal{A}$ of Figure 1 and the PTCTL formulae equal to

$$f : \forall\theta_1\forall\theta_2 \cdot (\theta_2 \leq \theta_1 \rightarrow \forall\Box(\sigma \rightarrow \forall\Diamond_{\leq 2\theta_1+2}\, \sigma))$$

and

$$g : \forall\theta_1 \cdot (\theta_1 \geq 5 \rightarrow \forall\Box(\sigma \rightarrow \forall\Diamond_{<2\theta_1+2}\, \sigma)).$$

Then $\Theta = \{\theta_1, \theta_2\}$, $\Theta_f = \varnothing$ and $\Theta_g = \{\theta_2\}$. The model-checking problem "does $(q_0, 0) \models f$ hold" has a YES answer. The parameter synthesis problem "for which parameter valuations v on Θ_g does $(q_0, 0) \models_v g$ hold" receives the answer $\theta_2 \leq 4$.

3 Decision Problems

In this section, we will prove that the model-checking problem is undecidable. The undecidability comes from the use of equality of the operators $\exists \mathrm{U}_{\sim\alpha}$ and $\forall \mathrm{U}_{\sim\alpha}$. When equality is forbidden in these operators, we will prove that the model-checking problem becomes decidable. In this case, we will also positively

[5] For instance this representation could be given in a decidable logical formalism.

solve the parameter synthesis problem. Our proofs use Presburger arithmetics and its extension with integer divisibility.

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

3.1 Undecidability Result for Equality

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

Presburger arithmetic with divisibility is an extension of Presburger arithmetic with integer divisibility relation. The additional divisibility relation is denoted by $z|z'$ and means "z divides z'". Every formula of Presburger arithmetic with divisibility can be put into *normal form*:

$$Qz_1Qz_2\dots Qz_n \cdot (\neg)\phi_1 \star (\neg)\phi_2 \star \cdots \star (\neg)\phi_m$$

where $\star$ belongs to $\{\vee, \wedge\}$, $(\neg)$ means that negation is optional and each ϕ_i is one of the following atomic formulae: (i) $\alpha = z$, (ii) $\alpha < z$, (iii) $z|z'$ such that α is a linear term and $z' > 0$. While Presburger arithmetic has a decidable theory, Presburger arithmetic with divisibility is undecidable [3].

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

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

$$Qz_1Qz_2\dots Qz_n \cdot (\neg)\phi_1 \star (\neg)\phi_2 \star \cdots \star (\neg)\phi_m.$$

We are going to construct a $\mathrm{PTCTL}_{\{=\}}$ formula f and a parametric timed automaton $\mathcal{A}$. The set Θ of parameters is equal to the set of variables used in Φ.

For each subformula ϕ_l of the form $\alpha = z$ or $\alpha < z$, we define the $\mathrm{PTCTL}_{\{=\}}$ formula $\hat{\phi}_l = \phi_l$. For each subformula ϕ_l of the form $z|z'$, we construct the next parametric timed automaton $\mathcal{A}_{\phi_l}$ and $\mathrm{PTCTL}_{\{=\}}$ formula $\hat{\phi}_l$. The automaton $\mathcal{A}_{\phi_l}$ is given in Figure 2. We label the unique initial state i_l of this automaton by σ_1^l and the unique final f_l state by σ_2^l. It is easy to see that there is a run ρ from the initial configuration $(i_l, 0)$ to the final configuration (f_l, z) with duration D_ρ iff $z|D_\rho$. For formula $\hat{\phi}_l$, we take $\sigma_1^l \wedge \exists\Diamond_{=z'}\sigma_2^l$.

Now we construct formula f as follows

$$Qz_1Qz_2\dots Qz_n \cdot (\neg)\hat{\phi}_1 \star (\neg)\hat{\phi}_2 \star \cdots \star (\neg)\hat{\phi}_m.$$

We construct the automaton $\mathcal{A}$ by first taking the union of all the previous automata $\mathcal{A}_{\phi_l}$ (introduced for the divisibility subformulae). We then merge

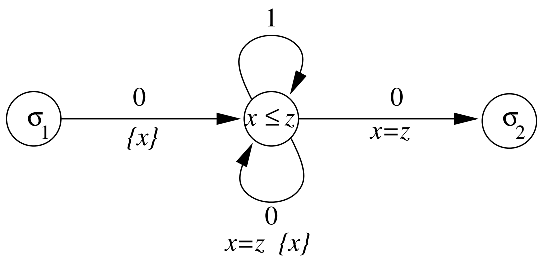

Figure 2: Automaton for $z|z'$

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^l. Finally, we add a new state q' to $\mathcal{A}$ and an edge $(f_l, 0, \top, \varnothing, q')$ from any final state f_l of $\mathcal{A}_{\phi_l}$ to state q' labeled 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. $\square$

As a direct consequence of Theorem 8, we have:

Corollary 9 *The model-checking problem for* $\mathrm{PTCTL}_{\{=\}}$ *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 $\mathrm{PTCTL}_{\{<,\leq,>,\geq\}}$. Our approach is as follows. Given a formula φ of QF-PTCTL we will construct a Presburger formula $\Delta_{q,\varphi}(x, \Theta)$ with free variables x and all $\theta \in \Theta$ such that

$$(q, x_0) \models_v \varphi \quad \text{iff} \quad \Delta_{q,\varphi}(x_0, v(\Theta)) \text{ is TRUE}$$

for *any* valuation v on Θ and *any* value x_0 of the clock (see Theorem 11). Solutions to Problems 6 and 7 will be obtained as a corollary (see Corollaries 14 and 15). For instance, the decidability of the model-checking problem will derive from the decidability of Presburger arithmetic. Indeed, if we denote by $Q\Theta\ \varphi$ a PTCTL formula f with no free parameters, then to test if $(q, x_0) \models f$ is equivalent to test if the sentence $Q\Theta\ \Delta_{q,\varphi}(x_0, \Theta)$ is TRUE.

Example Consider the parametric timed automaton of Figure 1 and the QF-PTCTL formula φ equal to $\forall\Box(\sigma \rightarrow \forall\Diamond_{\leq\theta_3}\sigma)$. Then $\Theta = \{\theta_1, \theta_2, \theta_3\}$. Presburger formula $\Delta_{q_0,\varphi}(x, \Theta)$ is here equal to $\theta_1 + \theta_2 + 2 \leq \theta_3$ with no reference to x since it is reset along the edge from q_0 to q_1. Thus $(q, x_0) \models_v \varphi$ for any clock value x_0 and any valuation v such that $v(\theta_1) + v(\theta_2) + 2 \leq v(\theta_3)$. The model-checking problem $(q, x_0) \models \forall\theta_1\forall\theta_2\exists\theta_3\varphi$ has a YES answer for any x_0 because the sentence $\forall\theta_1\forall\theta_2\exists\theta_3 \cdot (\theta_1 + \theta_2 + 2 \leq \theta_3)$ is TRUE in Presburger arithmetic. If clock x was not reset along the edge from q_0 to q_1, then the formula $\Delta_{q_0,\varphi}(x, \Theta)$

would be equal to $(\theta_1 + \theta_2 + 2 \leq \theta_3) \wedge (x \leq \theta_1)$ and the above model-checking problem would have a YES answer iff $\forall\theta_1\forall\theta_2\exists\theta_3 \cdot (\theta_1 + \theta_2 + 2 \leq \theta_3) \wedge (x_0 \leq \theta_1)$, that is $x_0 = 0$.

As indicated by this example, the Presburger formula $\Delta_{q,\varphi}(x, \Theta)$ constructed from the QF-PTCTL formula φ is a boolean combination of terms of the form $\theta \sim \alpha$ or $x \sim \alpha$ where θ is a parameter, x is the clock and α is a linear term over parameters. Formula $\Delta_{q,\varphi}(x, \Theta)$ must be seen as a *syntactic* translation of formula φ to Presburger arithmetic. The question "does $(q, x_0) \models f$ hold" with $f = Q\Theta\ \varphi$ is translated into the question "is the Presburger sentence $Q\Theta\ \Delta_{q,\varphi}(x_0, \Theta)$ TRUE". At this point only, *semantic* inconsistencies inside $Q\Theta\ \Delta_{q,\varphi}(x_0, \Theta)$ are looked for to check if this sentence is TRUE or not.

Our proofs require to work with a set $\mathcal{G}$ of guards that is more general than in Notation 1.

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 $\mathbb{N}$). Comparison symbol $\sim$ used in expressions like $x \sim \alpha$ and $\alpha \sim \beta$ belongs to the extended set $\{=, <, \leq, >, \geq, \equiv_{a,\leq}, \equiv_{a,\geq}\}$. For any constant $a \in \mathbb{N}^+$, notation $z \equiv_{a,\leq} z'$ means $z \equiv z'$ mod a and $z \leq z'$. Equivalently, this means that there exists $y \in \mathbb{N}$ such that $z + ay = z'$. Notation $z \equiv_{a,\geq} z'$ means $z \equiv z'$ mod a and $z \geq z'$.

Any $x \sim \alpha$ is called an *x-atom*, any $\alpha \sim \beta$ is called a *θ-atom*. An *x-conjunction* is any conjunction of x-atoms, and a *θ-conjunction* is any conjunction of θ-atoms. We denote by $\mathcal{B}_{x,\Theta}$ the set of *boolean combinations* of x-atoms and θ-atoms. A *guard* is any element of $\mathcal{B}_{x,\Theta}$. Thus the set $\mathcal{G}$ of Notation 1 is now equal to the set $\mathcal{B}_{x,\Theta}$.

From now on, it is supposed that the guards and the invariants appearing in parametric timed automata belong to the generalized set $\mathcal{G} = \mathcal{B}_{x,\Theta}$. It should be noted that the extension of $\sim$ to $\{=, <, \leq, >, \geq, \equiv_{a,\leq}, \equiv_{a,\geq}\}$ is only valid inside automata, and not inside PTCTL formulae. We shortly call *automaton* any parametric timed automaton $\mathcal{A}$. Let us state our main result.

Theorem 11 *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*

$$(q, x_0) \models_v \varphi \quad \textit{iff} \quad \Delta_{q,\varphi}(x_0, v(\Theta)) \textit{ is } \text{TRUE}$$

for any valuation v on Θ and any clock value x_0. The construction of formula $\Delta_{q,\varphi}$ is effective.

The proof of Theorem 11 is by induction on the way formula φ is constructed. The main ideas are roughly the following ones. First, suppose for instance that along a run $\rho = (q_i, x_i)_{i\geq 0}$ of $\mathcal{A}^v$ showing that $(q_0, x_0) \models_v \varphi$, some configuration, say (q_j, x_j), needs to satisfy $(q_j, x_j) \models_v \psi$ with ψ a subformula of φ. The automaton $\mathcal{A}$ is modified into $\mathcal{A}'$ such that the invariant $\mathcal{I}(q_j)$ is *augmented*[6] by the $\mathcal{B}_{x,\Theta}$ formula $\Delta_{q_j,\psi}$ constructed by induction. Along the run ρ seen in the modified automaton $\mathcal{A}'$, the satisfaction relation $(q_j, x_j) \models_v \psi$ holds automatically thanks to the augmented invariant of q_j. Second, what we also need is a $\mathcal{B}_{x,\Theta}$ formula that expresses the existence of an infinite run starting at a given configuration (for operator $\exists\Box$ for instance) and another one that expresses the existence of a finite run ρ starting and ending at given configurations such that $\mathrm{D}_\rho \sim v(\alpha)$ (for operator $\exists\mathrm{U}_{\sim\alpha}$ for instance). This is possible by the next two propositions. Their proof is postponed till Section 4.

Proposition 12 *Let $\mathcal{A}$ be an automaton and q be a state. Then there exists a $\mathcal{B}_{x,\Theta}$ formula* $\mathrm{Run}_q(x,\Theta)$ *such that for any valuation v and any clock value x_0,*

$$\mathrm{Run}_q(x_0, v(\Theta)) \textit{ is } \textsc{true}$$

iff there exists an infinite run in $\mathcal{A}^v$ starting with (q, x_0). The construction of $\mathrm{Run}_q(x,\Theta)$ *is effective.*

Proposition 13 *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* $\mathrm{Duration}^{\sim\alpha}_{q,q'}(x,\Theta)$ *such that for any valuation v and any clock value x_0,*

$$\mathrm{Duration}^{\sim\alpha}_{q,q'}(x_0, v(\Theta)) \textit{ is } \textsc{true}$$

iff there exists a finite run $\rho = (q, x_0) \rightsquigarrow (q', \cdot)$ in $\mathcal{A}^v$ with $D_\rho \sim v(\alpha)$. The construction of $\mathrm{Duration}^{\sim\alpha}_{q,q'}(x,\Theta)$ *is effective.*

For the proof of Theorem 11, 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 formula $\exists\Box_{\leq\alpha}\varphi$ 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 $(q,x) = (q_0, x_0)$, there exists $j \geq 0$ such that $D_\rho(q_j, x_j) \geq v(\alpha)$ and $(q_i, x_i) \models_v \varphi$ for any $i < j$.

Proof of Theorem 11 (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

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

[6] Such kind of invariant is allowed in Notation 10.

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

$$\Delta_{q,\varphi}(x, \Theta) \;=\; (\alpha \sim \beta) \wedge \mathrm{Run}_q(x, \Theta).$$

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 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 ρ'. Let (q, τ, g, r, q') be the edge of E that has lead to the transition $(q, x_0) \xrightarrow{\tau} (q', x'_0)$. Then (see Definition 3), $x'_0 = 0$ 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 $(q', x'_0) \models_v \psi$. The automaton $\mathcal{A}$ is modified into an automaton $\overline{\mathcal{A}}$ as follows. A copy[7] $\overline{q}'$ of q' is added to Q such that $\mathcal{L}(\overline{q}') = \mathcal{L}(q')$, $\mathcal{I}(\overline{q}') = \mathcal{I}(q') \wedge \Delta_{q',\psi}(x, \Theta)$. A copy $(\overline{q}', \tau', g', r', p)$ is also added for any edge (q', τ', r', g', p) leaving q'. By Proposition 12 applied to $\overline{\mathcal{A}}$ and $\overline{q}'$, we get a $\mathcal{B}_{x,\theta}$ formula $\mathrm{Run}_{\overline{q}'}$ such that $\mathrm{Run}_{\overline{q}'}(x'_0, v(\Theta))$ is TRUE iff there exists an infinite run in $\overline{\mathcal{A}}^v$ starting with $(\overline{q}', x'_0)$. By construction of $\overline{q}'$, equivalently there exists an infinite run in $\mathcal{A}^v$ starting with (q', x'_0) and such that $(q', x'_0) \models_v \psi$. Hence, the expected formula $\Delta_{q,\varphi}(x, \Theta)$ is equal to

$$\begin{array}{rcl} \Delta_{q,\varphi}(x, \Theta) & = & \bigvee_{(q,\tau,g,r,q') \in E_R} (\mathcal{I}(q) \wedge \mathrm{Run}_{\overline{q}'}(0, \Theta)) \\ & \vee & \bigvee_{(q,\tau,g,r,q') \in E \setminus E_R} (\mathcal{I}(q) \wedge \mathrm{Run}_{\overline{q}'}(x + \tau, \Theta)) \end{array}$$

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

The construction of formula $\Delta_{q,\varphi}$ for $\varphi = \exists\Box\psi$ is in the same vein as the previous one. Recall that $(q, x_0) \models_v \varphi$ iff there is an infinite run in $\mathcal{A}^v$ with first configuration (q, x_0) such that all its configurations satisfy ψ. The automaton $\mathcal{A}$ is here modified into $\overline{\mathcal{A}}$ as follows. For any state $p \in Q$, $\mathcal{I}(p)$ is replaced by $\mathcal{I}(p) \wedge \Delta_{p,\psi}(x, \Theta)$. By Proposition 12 applied to $\overline{\mathcal{A}}$, we get a formula Run_q such that $\mathrm{Run}_q(x_0, v(\Theta))$ is TRUE iff there exists an infinite run in $\mathcal{A}^v$ starting with (q, x_0) and such that all its configurations satisfy ψ. Therefore formula $\Delta_{q,\varphi}(x, \Theta)$ is equal to

$$\mathrm{Run}_q(x, \Theta).$$

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 $\rho = (q, x_0) \rightsquigarrow (q', x'_0)$ such that $\mathrm{D}_\rho \sim v(\alpha)$, ψ is satisfied at any configuration of ρ distinct from (q', x'_0), ϕ is satisfied at (q', x'_0) and (q', x'_0) 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 case (1), with the same construction of $\overline{\mathcal{A}}$ as done before for operator $\exists\bigcirc$ (with q, ϕ instead of q', ψ), we have the next formula

$$(0 \sim \alpha) \wedge \mathrm{Run}_{\overline{q}}(x, \Theta).$$

Case (2) is more involved. The automaton $\mathcal{A}$ is first modified into $\overline{\mathcal{A}}$ as for operator $\exists\bigcirc$ (with q', ϕ instead of q', ψ) to get formula $\mathrm{Run}_{\overline{q}'}$ such that $\mathrm{Run}_{\overline{q}'}(x'_0, v(\Theta))$

[7] The copy $\overline{q}'$ of q' is needed to focus on the first configuration (q', x'_0) of ρ'.

is TRUE iff there exists an infinite run in $\mathcal{A}^v$ starting with (q', x'_0) and such that $(q', x'_0) \models_v \phi$. The automaton $\mathcal{A}$ is then modified in another automaton $\underline{\mathcal{A}}$ in the following way. A copy $\underline{q}'$ of q' is added to Q as well as a copy of any edge of E entering q' as entering $\underline{q}'$; we define $\mathcal{L}(\underline{q}') = \mathcal{L}(q')$ and $\mathcal{I}(\underline{q}') = \mathcal{I}(q') \wedge \mathrm{Run}_{\overline{q}'}(x, \Theta)$.[8] For any state p of Q, $\mathcal{I}(p)$ is replaced by $\mathcal{I}(p) \wedge \Delta_{p,\psi}(x, \Theta)$. Thanks to Proposition 13 applied to $\underline{\mathcal{A}}$, we obtain a formula $\mathrm{Duration}^{\sim\alpha}_{q,\underline{q}'}(x, \Theta)$ expressing the following: $\mathrm{Duration}^{\sim\alpha}_{q,\underline{q}'}(x_0, v(\Theta))$ is TRUE iff there exists in $\underline{\mathcal{A}}^v$ a finite run $\underline{\rho} = (q, x_0) \rightsquigarrow (\underline{q}', x'_0)$ with $\mathrm{D}_{\underline{\rho}} \sim v(\alpha)$. Equivalently there exists in $\mathcal{A}^v$ a finite run $\rho = (q, x_0) \rightsquigarrow (q', x'_0)$ with $\mathrm{D}_\rho \sim v(\alpha)$ such that ψ is satisfied at any configuration of ρ distinct from (q', x'_0), ϕ is satisfied at (q', x'_0) and (q', x'_0) is the first configuration of an infinite run. For case (2), the expected formula is thus the disjunction

$$\bigvee_{q' \in Q} \mathrm{Duration}^{\sim\alpha}_{q,\underline{q}'}(x, \Theta).$$

Therefore, formula $\Delta_{q,\varphi}$ is the disjunction

$$((0 \sim \alpha) \wedge \mathrm{Run}_{\overline{q}}(x, \Theta)) \quad \vee \quad \bigvee_{q' \in Q} \mathrm{Duration}^{\sim\alpha}_{q,\underline{q}'}(x, \Theta).$$

Finally, 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 done just before, $\mathcal{A}$ is modified into $\underline{\mathcal{A}}$ except that we use $\mathrm{Run}_{q'}$ instead of $\mathrm{Run}_{\overline{q}'}$ in the definition of $\mathcal{I}(\underline{q}')$. By Proposition 13, formula $\Delta_{q,\varphi}$ is equal to

$$\bigvee_{q' \in Q} \mathrm{Duration}^{\geq\alpha}_{q,\underline{q}'}(x, \Theta).$$

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

Solutions to the model-checking problem and the parameter synthesis problem are obtained as a corollary of Theorem 11.

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

Proof Let $Q\Theta\ \varphi$ be a PTCTL formula f with no free parameters. By Theorem 11, $(q, x_0) \models f$ iff $Q\Theta\ \Delta_{q,\varphi}(x_0, \Theta)$ is TRUE. As Presburger arithmetic has a decidable theory and $Q\Theta\ \Delta_{q,\varphi}(x_0, \Theta)$ is Presburger sentence, the model-checking problem is decidable. □

The next corollary is straightforward. It states that the parameter synthesis problem is solvable.

[8] The copy $\underline{q}'$ of q' is needed to focus on the last configuration (q', x'_0) of ρ; the augmented invariant is needed to express that ϕ is satisfied at (q', x'_0) and (q', x'_0) is the first configuration of an infinite run.

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* $\mathrm{PTCTL}_{\{<,\leq,>,\geq\}}$ *formula. Then the Presburger formula $Q_1\theta_1 \ \cdots \ Q_k\theta_k \ \Delta_{q,\varphi}(x_0, \Theta)$ with free parameters in Θ_f is an effective characterization of the set of valuations v on Θ_f such that $(q, x_0) \models_v f$.*

Let us denote by $V(\mathcal{A}, f, q, x_0)$ the set of valuations v on Θ_f such that $(q, x_0) \models_v f$. Let Θ_f be equal to $\{\theta'_1, \ldots, \theta'_l\}$. Presburger arithmetics has an effective quantifier elimination, by adding to the operations $+$ and $\leq$ all the congruences $\equiv \mathrm{mod} a, a \in \mathbb{N}^+$. It follows the characterization of $V(\mathcal{A}, f, q, x_0)$ given above by $Q_1\theta_1 \ \cdots \ Q_k\theta_k \ \Delta_{q,\varphi}(x, \Theta)$ can be effectively rewritten without any quantifier. On the other hand, since Presburger arithmetic has a decidable theory, any question formulated in this logic about $V(\mathcal{A}, f, q, x_0)$ is decidable. For instance, the question "Is the set $V(\mathcal{A}, f, q, x_0)$ non empty" is decidable as it is formulated in Presburger arithmetic by $\exists\theta'_1 \ \cdots \ \exists\theta'_l \ Q_1\theta_1 \ \cdots \ Q_k\theta_k \ \Delta_{q,\varphi}(x, \Theta)$. The question "Does the set $V(\mathcal{A}, f, q, x_0)$ contain all the valuations on Θ_f" is also decidable as it can be formulated as $\forall\theta'_1 \ \cdots \ \forall\theta'_l \ Q_1\theta_1 \ \cdots \ Q_k\theta_k \ \Delta_{q,\varphi}(x, \Theta)$. The question "Is the set $V(\mathcal{A}, f, q, x_0)$ finite" is translated into

$$\exists z \forall\theta'_1 \ \cdots \ \forall\theta'_l \ Q_1\theta_1 \ \cdots \ Q_k\theta_k \ \cdot (\Delta_{q,\varphi}(x, \Theta) \ \Rightarrow \ \wedge_i \theta'_i \leq z).$$

And so on.

4 Durations

The aim of this section is a proof of Propositions 12 and 13. This will be achieved thanks to a precise description of the possible durations of finite runs in an automaton. Several steps are necessary for this purpose.

In the first subsection, we show that we can work with automata put in some normal form. This normalization allows a simplified presentation of the proofs of the next subsections.

In Subsections 4.2 and 4.3, we restrict to *reset-free* normalized automata $\mathcal{A}$, that is automata in which there is no reset of the clock. For this family of automata, we study the runs of the form $(i, x_0) \rightsquigarrow (f, \cdot)$ such that $i \in I$, $f \in F$ with I, F being two fixed subsets of states, and x_0 is a fixed clock value. In Subsection 4.2, a sequence of transformations is performed on the automaton such that the x-atoms used in the automaton are limited to equalities $x = \alpha$. These simplifications lead in Subsection 4.3 to the description by a Presburger formula of the durations D_ρ of runs $\rho = (i, x_0) \rightsquigarrow (f, \cdot)$, $i \in I$, $f \in F$.

In the last subsection, we remove the reset-free restriction imposed to $\mathcal{A}$ and we study in details the durations D_ρ of runs $\rho = (q, x_0) \rightsquigarrow (q', \cdot)$ between two fixed states q and q'. Any such run ρ can be decomposed into a sequence of runs ρ_j, $1 \leq j \leq k$, according to the reset of the clock, that is the clock is reset at the beginning and the end of ρ_j but not inside of ρ_j. The duration D_ρ of ρ is thus the sum of the durations D_{ρ_j}, $1 \leq j \leq k$. Any D_{ρ_j} falls into durations being studied in Section 4.3. Thanks to this description of any duration D_ρ in terms of durations in reset-free automata, we are finally able to prove Propositions 12 and 13.

In Subsections 4.1, 4.2 and 4.3, we are going to perform a sequence of transformations on $\mathcal{A}$ that will *preserve* the set of runs in $\mathcal{A}^v$ for any valuation v, 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[9] up to a *renaming* of any $\overline{q}_j$ into q.

4.1 Normalized Automata

In this subsection, the automata are put in some normal form. The aim of this normalization is a simplified presentation of the proofs in the rest of the paper.

Definition 16 An automaton $\mathcal{A}$ is *normalized* if

- The guards labeling the edges and used in the invariants are limited to conjunctions of x-atoms and θ-atoms with $\sim \in \{=, \leq, \geq, \equiv_{a,\leq}, \equiv_{a,\geq}\}$,
- for any state $q \in Q$, the edges (p, τ, g, r, q) entering q are all labeled by the same g and the same r (however τ can vary).

Proposition 17 *Any automaton $\mathcal{A}$ can be effectively normalized such that the set of runs in $\mathcal{A}^v$ is preserved for any valuation v.*

Before giving the proof of this proposition, we need the following result.

Lemma 18 *Any $\mathcal{B}_{x,\Theta}$ formula is a Presburger formula. It can be rewritten as a disjunction of conjunctions of x-atoms and θ-atoms with $\sim$ limited to $\{=, \leq, \geq, \equiv_{a,\leq}, \equiv_{a,\geq}\}$.*

Proof Operators $\equiv_{a,\leq}$ and $\equiv_{a,\geq}$ are easily rewritten in Presburger arithmetic. Even if linear terms $\alpha, \beta, \ldots$ contain constants in $\mathbb{Z}$, any $x \sim \alpha$ and $\alpha \sim \beta$ can also be rewritten in Presburger arithmetic. This shows that any $\mathcal{B}_{x,\Theta}$ formula is a Presburger formula. To rewrite it as described in the lemma, it is first put into disjunctive normal form. Second negation is suppressed 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 $\neg(z \equiv_{a,\leq} z')$ is equivalent to $(z > z') \vee (\bigvee_{0<b<a} z + b \equiv_{a,\leq} z')$. Similarly for $\neg(z \equiv_{a,\geq} z')$. Third all inequalities $z < z'$ and $z > z'$ are replaced respectively by $z \leq z' - 1$ and $z \geq z' + 1$. Finally this formula is put into disjunctive normal form. $\square$

Proof of Proposition 17. Let $g \in \mathcal{B}_{x,\Theta}$ be a guard. By Lemma 18, it can be rewritten as a disjunction of k formulae δ_j, $1 \leq j \leq k$, where each δ_j is a conjunction of x-atoms and θ-atoms with $\sim \in \{=, \leq, \geq, \equiv_{a,\leq}, \equiv_{a,\geq}\}$. If g labels the edge (q, τ, g, r, q') of $\mathcal{A}$, then we modify $\mathcal{A}$ by splitting this edge into k edges $(q, \tau, \delta_j, r, q')$, $1 \leq j \leq k$. If $g = \mathcal{I}(q)$ for some state q, we modify $\mathcal{A}$ by splitting

[9] Such an identification of runs is already present in the proof of Theorem 11.

q into k states $\overline{q}_j$, $1 \leq j \leq k$, such that $\mathcal{L}(\overline{q}_j) = \mathcal{L}(q)$, $\mathcal{I}(\overline{q}_j) = \delta_j$ and we accordingly split any edge that enters or leaves state q. The first condition of Proposition 17 is therefore satisfied.

For the second condition, the construction is similar. Suppose that are several edges (p, τ, g, r, q) entering state q with distinct couples (g, r). Then q is splitted into several copies (one copy for one couple (g, r)) and all the edges entering q are redirected to each copy, according to the couples (g, r). The copies of q have the same $\mathcal{L}(q)$ and $\mathcal{I}(q)$ as q. $\square$

4.2 Transformations of Reset-free Automata

In all this subsection, we assume the next hypothesis.

Hypothesis ($*$) We assume that $\mathcal{A} = (Q, I, F, E, \mathcal{L}, \mathcal{I})$ is a *reset-free normalized* 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 $f \in F$.

Remark As $\mathcal{A}$ is normalized and reset-free, given a state q, all edges (p, τ, g, r, q) entering q have the same guard g and satisfy $r = \varnothing$. It follows that we can move guard g from these edges to the invariant $\mathcal{I}(q)$ of q. Indeed g is simply erased from all the edges entering q and added as a conjunction to $\mathcal{I}(q)$. By this construction, the set E of edges of $\mathcal{A}$ can be rewritten as a subset of $Q \times \{0, 1\} \times Q$, instead of $Q \times \{0, 1\} \times \mathcal{G} \times 2^{\{x\}} \times Q$ (see Definition 2).

On the other hand, as $\mathcal{A}$ is normalized, the invariant $\mathcal{I}(q)$ of any state q is a conjunction of x-atoms and θ-atoms. We can view $\mathcal{I}(q)$ as a *set* of x-atoms and θ-atoms (instead of a conjunction) and we can say that an x-atom or a θ-atom *belongs* to q (instead of $\mathcal{I}(q)$) or *appears* in q.

Given a valuation v and a clock value x_0, 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 $\mathcal{R}(\mathcal{A}^v, x_0)$. The aim of these transformations is to simplify the form of invariants used in the automaton. The invariant $\mathcal{I}(q)$ of any state $q \in Q \setminus (I \cup F)$ will be a conjunction of at most one x-atom (of the form $x = \alpha$) and one θ-conjunction. This simplification will be possible mainly because the automaton is reset-free (see Proposition 20).

Definition 19 A reset-free normalized automaton $\mathcal{A}$ is *simplified* if

- for all $q \in Q$, the invariant $\mathcal{I}(q)$ is equal to $\mathcal{I}_x(q) \wedge \mathcal{I}_\theta(q)$ such that $\mathcal{I}_x(q)$ is an x-conjunction and $\mathcal{I}_\theta(q)$ is a θ-conjunction. Among the x-atoms $x \sim \alpha$ of $\mathcal{I}_x(q)$, at most one is an equality $x = \alpha$. Moreover, if $q \notin I \cup F$, then $\mathcal{I}_x(q)$ contains no other x-atom $x \sim \beta$ with $\sim \in \{\leq, \geq, \equiv_{a,\leq}, \equiv_{a,\geq}\}$, and if $q \in I$ (resp. $q \in F$), then the other x-atoms of $\mathcal{I}_x(q)$ are of the form $x \geq \beta$ (resp. $x \leq \beta$).

- 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 $\mathcal{I}_x(q')$ contains $x = \alpha$.

Proposition 20 *Any reset-free normalized automaton $\mathcal{A}$ can be effectively simplified such that the set $\mathcal{R}(\mathcal{A}^v, x_0)$ is preserved for any valuation v and any clock value x_0.*

Proof The proof of Proposition 20 needs several steps. The transformations described in the proof are based on standard constructions of automata theory. Each of them will preserve $\mathcal{R}(\mathcal{A}^v, x_0)$ for any valuation v and any clock value x_0. In the first step, we are going to eliminate in $\mathcal{I}_x(q)$ all x-atoms of the form $x \equiv_{a,\leq} \alpha$.

First step x-atoms $x \equiv_{a,\leq} \alpha$.

Let us show that any x-atom $x \equiv_{a,\leq} \alpha$ belonging to some state q can be eliminated at the cost of a new x-atom $x \leq \alpha$. The idea is the following. If $\alpha \equiv b \bmod a$ for some $b \in \{0, 1, \ldots, a-1\}$, then

$$x \equiv_{a,\leq} \alpha \quad \text{iff} \quad x \equiv b \bmod a \text{ and } x \leq \alpha.$$

The automaton is transformed in a way to compute modulo a. New states are of the form (q, c) with $q \in Q$ and $c \in \{0, \ldots, a-1\}$ expressing that $x \equiv c \bmod a$. 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 $c' \equiv c + \tau \bmod a$. 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 $x \equiv_{a,\leq} \alpha$, eliminate this state if $c \neq b$, replace $x \equiv_{a,\leq} \alpha$ by $x \leq \alpha$ if $c = b$. If $(q, c) \in I'$, add the x-atom $x \equiv_{a,\geq} c$ and the θ-atom $\alpha \equiv_{a,\geq} b$ to recall that $\alpha \equiv b \bmod a$ and $x \equiv c \bmod a$ initially. As α depends on the parameter valuation, value b such that $\alpha \equiv b \bmod a$ is not known in advance. Therefore the final automaton is the disjoint union of the automata $\mathcal{A}_b$, with $b \in \{0, \ldots, a-1\}$.

The elimination of x-atoms $x \equiv_{a,\geq} \alpha$ in any $\mathcal{I}_x(q)$ is performed similarly. In the next step, we are going to eliminate x-atoms $x \geq \alpha$. This will be possible except inside states $q \in I$.

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

Let us consider a fixed x-atom $x \geq \alpha$. Recall that the automaton is reset-free. Along a run $\rho \in \mathcal{R}(\mathcal{A}^v, x_0)$, as soon as $x \geq \alpha$ is satisfied at some configuration of ρ, the next occurrences of $x \geq \alpha$ are automatically satisfied and can be thus eliminated. The automaton is transformed in a way to count occurrences of $x \geq \alpha$ thanks to a counter equal to 0 (1 or 2 resp.) in case of 0 (1 or 2 and more resp.) occurrence(s) of $x \geq \alpha$ is (are) encountered. Thus when the counter c has value 2, any incrementation $c + 1$ 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 $x \geq \alpha$ 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 $x \geq \alpha$, and $c' = c$ otherwise. Finally, we suppress $x \geq \alpha$ in any state $(q, 2)$ containing it.

Now, consider a run $\rho' \in \mathcal{R}(\mathcal{A}'^v, x_0)$ equal to $(q_i, c_i, x_i)_{0 \leq i \leq n}$ such that some state (q_k, c_k) contains $x \geq \alpha$. Necessarily, $c_k = 1$ and $c_i = 0$ for $0 \leq i < k$ by construction of $\mathcal{A}'$. So x-atom $x \geq \alpha$ is satisfied at configuration (q_k, c_k, x_k) iff either $x \geq \alpha$ is satisfied at configuration (q_0, c_0, x_0) or $x = \alpha$ is satisfied at some configuration (q_i, c_i, x_i) of ρ' such that $0 < i \leq k$. Therefore, x-atom $x \geq \alpha$ can be eliminated at the cost of a new x-atom $x = \alpha$, except inside the initial state (q_0, c_0). This can be achieved by modifying $\mathcal{A}'$ into an automaton $\mathcal{A}''$ thanks to a construction which is not difficult but tedious, this will be not fully detailed. A first part of $\mathcal{A}''$ has to deal with paths of $\mathcal{A}'$ that only contain states (q, c) with $c = 0$; it is obtained from $\mathcal{A}'$ by erasing all states (q, c) with $c = 1$. A second part of $\mathcal{A}''$ has to deal with paths of $\mathcal{A}'$ starting with (q, c) such that $c = 1$; it is obtained from $\mathcal{A}'$ by erasing all states (q, c) such that $q \notin I$, $c = 1$ and all states (q, c) such that $q \in I$, $c = 0$. A third part of $\mathcal{A}''$ has to deal with paths of $\mathcal{A}'$ containing some state (q, c) such that $q \notin I$, $c = 1$; such paths are call *special*. The special paths of $\mathcal{A}'$ must be modified into two kinds of paths : either the x-atom $x \geq \alpha$ is added to the initial state of the path, or the x-atom $x = \alpha$ is added to some intermediate state of the path, which is situated between the initial state (not included) and state (q, c) (included). In both cases, the x-atom $x \geq \alpha$ must be deleted from (q, c). The third part of $\mathcal{A}''$, first case, is obtained from $\mathcal{A}'$ by adding the x-atom $x \geq \alpha$ to any state (q, c) such that $q \in I$, $c = 0$ and by deleting the x-atom $x \geq \alpha$ from any state (q, c) such that $q \notin I$, $c = 1$; it is also necessary to use a marker to verify that each accepting path of $\mathcal{A}''$ corresponds to a special path of $\mathcal{A}'$. The third part of $\mathcal{A}''$, second case, is obtained from $\mathcal{A}'$ as follows : the x-atom $x \geq \alpha$ is deleted from any state (q, c) such that $q \notin I$, $c = 1$, all states (q, c) with $q \notin I$, $c = 0$ are duplicated (together with the edges entering and leaving (q, c)) such that the x-atom $x = \alpha$ is added to one of the two copies of (q, c); it is also necessary to use a marker to verify that each accepting path of $\mathcal{A}''$ corresponds to a special path of $\mathcal{A}'$ and passes through exactly one state containing the x-atom $x = \alpha$.

The elimination of x-atoms $x \leq \alpha$ can be performed in a similar way. Note that here, as soon as the last (instead of the first) occurrence of $x \leq \alpha$ is satisfied along a run $\rho \in \mathcal{R}(\mathcal{A}^v, x_0)$, then the previous occurrences of $x \leq \alpha$ are automatically satisfied. It follows that x-atoms $x \leq \alpha$ can be eliminated except inside states $q \in F$.

At this point of the proof, for any state q, (1) if $q \notin I \cup F$, then the x-atoms contained in q are of the form $x = \alpha$, (2) if $q \in I$, then they are of the form $x = \alpha$ or $x \geq \alpha$, and (3) if $q \in F$, then they are the form $x = \alpha$ or $x \leq \alpha$.

It remains to prove two facts about x-atoms which are equalities. First for any state $q \in Q$, among the x-atoms contained in q, at most one is an equality $x = \alpha$. Second, 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 $\mathcal{I}_x(q')$ contains $x = \alpha$.

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

The first fact can be easily proved. 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 $\mathcal{I}_\theta(q) \wedge \bigwedge_{\alpha \in A}(\alpha' = \alpha)$.

Let us prove the second fact. 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 resp.) in case of 0 (1 or 2 and more resp.) 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

$\tau \backslash c$	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'

(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 $x = \alpha$, and the presence of index $+$ means a progress of time since the first occurrence of $x = \alpha$. $\square$

4.3 Durations in Reset-free Automata

In this subsection, we again make Hypothesis $(*)$. By Proposition 20, we know that the reset-free normalized automaton $\mathcal{A}$ can be supposed simplified. Thanks to this property of $\mathcal{A}$, we are going to construct a Presburger formula describing all the possible durations of runs in $\mathcal{R}(\mathcal{A}^v, x_0)$ in terms of the parameters. We need the next notation.

Notation 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*

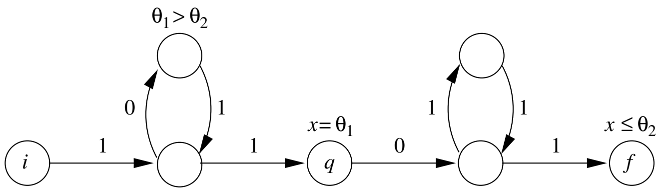

Figure 3: A reset-free normalized automaton which is simplified

if it is of the form

$$\begin{aligned}&t = \alpha,\\ &t \equiv_{a,\geq} \alpha,\\ &t = \alpha - x,\\ &t \equiv_{a,\geq} \alpha - x,\end{aligned}$$

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

$$t \leq \alpha - x.$$

A *t-conjunction* is a conjunction of t-atoms of second type.

Proposition 21 *Let $\mathcal{A}$ be an automaton. There exists a Presburger formula $\lambda(t, x, \Theta)$ such that for any valuation v and any clock value x_0,*

$$\lambda(t_0, x_0, v(\Theta)) \textit{ is } \text{TRUE}$$

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

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

where λ_t is a first type t-atom , $\lambda_\leq$ is a t-conjunction, λ_x is an x-conjunction and λ_θ is a θ-conjunction. Its construction is effective.

Let us explain this proposition on the next example.

Example Consider the simplified automaton $\mathcal{A}$ of Figure 3 with one initial state i and one final state f. We denote by t_0 the duration of any run $(i, x_0) \rightsquigarrow (f, \cdot)$ in $\mathcal{R}(\mathcal{A}^v, x_0)$, where v is a fixed parameter valuation. Every run has to pass through state q which contains the x-atom $x = \theta_1$. Let us study the possible durations t_1 of runs $\rho_1 = (i, x_0) \rightsquigarrow (q, \cdot)$. Each duration t_1 must be equal to $v(\theta_1) - x_0$. For runs ρ_1 using the cycle, constraint $v(\theta_1) > v(\theta_2)$ holds and t_1 has the form $m + 3$, $m \geq 0$. The unique run ρ_1 not using the cycle is not constrained and its duration equals $t_1 = 2$. Now any duration t_0 can be decomposed as $t_0 = t_1 + 2n + 1 = v(\theta_1) - x_0 + 2n + 1$, $n \geq 0$. Due to the x-atom $x \leq \theta_2$ of state f, we get another constraint $x_0 + t_0 \leq v(\theta_2)$. In summary, we have

$$\begin{array}{ll} & [(v(\theta_1) - x_0 \equiv_{1,\geq} 3 \ \wedge \ v(\theta_1) > v(\theta_2)) \quad \vee \quad v(\theta_1) - x_0 = 2] \\ \wedge & [t_0 \equiv_{2,\geq} v(\theta_1) - x_0 + 1] \\ \wedge & [x_0 + t_0 \leq v(\theta_2)] \end{array}$$

We get the next Presburger formula $\lambda(t, x, \Theta)$

$$\begin{array}{ll} & [(x \equiv_{1,\leq} \theta_1 - 3 \ \wedge \ \theta_1 > \theta_2) \quad \vee \quad x = \theta_1 - 2] \\ \wedge & [t \equiv_{2,\geq} \theta_1 + 1 - x] \\ \wedge & [t \leq \theta_2 - x] \end{array}$$

such that there exists a run in $\mathcal{R}(\mathcal{A}^v, x_0)$ with duration t_0 iff $\lambda(t_0, x_0, v(\Theta))$ is TRUE. This formula is in the form of Proposition 21 when it is rewritten as a disjunction of conjunctions of t-atoms, x-atoms and θ-atoms.

Thanks to the previous example, we can give some ideas of the proof of Proposition 21. Except for the initial and final states, the states of a simplified automaton contain at most one x-atom which is of the form $x = \alpha$. The proof will be by induction on these x-atoms. Given an x-atom $x = \alpha$ contained in some state q, any run ρ in $\mathcal{R}(\mathcal{A}^v, x_0)$ passing through this state q can be decomposed as $(i, x_0) \rightsquigarrow (q, x_1)$ and $(q, x_1) \rightsquigarrow (f, x_2)$. Its duration t_0 can also be decomposed as $t_1 + t_2$ with the constraint that the clock value $x_0 + t_1$ must satisfy $x = \alpha$. It follows that $t_0 = v(\alpha) - x_0 + t_2$. The durations t_1 and t_2 and the related constraints will be computed by induction. When there is no x-atom in the automaton (base case), only θ-atoms can appear in states. Runs will therefore be partitioned according to the set of θ-atoms that constrain them. Their durations will be described as fixed values or arithmetic progressions.

Proof of Proposition 21. It is supposed that $\mathcal{A} = (Q, I, F, E, \mathcal{L}, \mathcal{I})$ is simplified.

(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 make the assumption that i contains no x-atom and f contains no x-atom $x \leq \alpha$. 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 $q \in Q$. The formula $\lambda(t, x, \Theta)$ that we will construct will have *no* t-conjunction, that is $\lambda(t, x, \Theta)$ will be a disjunction of formulae of the form $\lambda_t \wedge \lambda_x \wedge \lambda_\theta$.

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 a constraint which is the conjunction of the θ-atoms contained in the states of the run.

The proof is based on the classical Kleene theorem [8] using the particular alphabet

$$B = \{(\tau, \varsigma) \mid \tau \in \{0, 1\}, \varsigma \in \{\mathcal{I}_\theta(q), q \in Q\}\}.$$

To any edge (q, τ, q') of $\mathcal{A}$ corresponds the letter $(\tau, \mathcal{I}_\theta(q'))$ of B. The concatenation $\cdot$ of two letters (τ_1, ς_1) and (τ_2, ς_2) is defined as $(\tau_1 + \tau_2, \varsigma_1 \wedge \varsigma_2)$. Thus a word over B is equal to (t, ς) where t is a positive integer (a duration) and ς is a θ-conjunction (a constraint on the parameters). In particular, the empty word is equal to $(0, \top)$. The star operation $*$ is defined as usual and the plus operation $+$ is defined by $L^+ = L^* \setminus \{(0, \top)\}$. We denote by $\mathrm{Rat}_B(\cdot, +)$ the smallest family of languages containing B and closed under $\cdot$ and $+$. The elements of a set $L \in \mathrm{Rat}_B(\cdot, +)$ have a simple form. The second components of these elements are all identical because operation $\wedge$ is idempotent. The first components constitute a set which is the union of a finite set and a finite number of arithmetic progressions [5]. In other words L is described by a disjunction of formulae of the form $\lambda_t \wedge \lambda_\theta$ such that λ_θ equals a fixed θ-conjunction ς and λ_t equals either $t = \alpha$ or $t \equiv_{a,\geq} \alpha$ with $\alpha \in \mathbb{N}$.

Now by Kleene's theorem applied to $\mathcal{A}$, we get a rational language over B whose first components describe the durations of all runs of $\mathcal{R}(\mathcal{A}^v, x_0)$ and the second components describe the related constraints. It is not difficult to prove that this rational language can be rewritten as a finite union of languages in $\mathrm{Rat}_B(\cdot, +)$. We thus get the required formula $\lambda(t, x, \Theta)$ as a disjunction of formulae $\lambda_t \wedge \lambda_\theta$ where λ_t is a first-type t-atom and λ_θ is a θ-conjunction.

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 (see the second part of Definition 19). 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 one occurrence of the state p of P.

All runs containing no state of P constitute the 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. The automaton $\mathcal{A}$ is modified into $\mathcal{A}^{p,1}$ by erasing states of $P \setminus \{p\}$ and edges leaving p. Invariant $\mathcal{I}_x(p)$ is replaced by $\top$. The new unique final state is p. The new automaton has one x-atom less, so $\lambda^{p,1}(t, x, \Theta)$ can be constructed by induction hypothesis such that $\lambda^{p,1}(t_1, x_0, v(\Theta))$ is TRUE. Formula $\lambda^{p,1}$ is a disjunction of formulae $\lambda_t^1 \wedge \lambda_x^1 \wedge \lambda_\theta^1$ where λ_t^1 is a first type t-atom, λ_x^1 is an x-conjunction and λ_θ^1 is a θ-conjunction. Suppose that λ_t^1 is one among

$$t = \alpha_1, \quad t \equiv_{a,\geq} \alpha_1, \quad t = \alpha_1 - x, \quad t \equiv_{a,\geq} \alpha_1 - x. \tag{1}$$

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

$$x_1 = v(\alpha), \quad t_1 = v(\alpha) - x_0. \tag{2}$$

So in (1), t can be replaced by $\alpha - x$ and (1) becomes

$$\alpha - x = \alpha_1, \quad \alpha - x \equiv_{a,\geq} \alpha_1, \quad \alpha = \alpha_1, \quad \alpha \equiv_{a,\geq} \alpha_1.$$

Thus λ_t^1 becomes an x-atom or a θ-atom. The modified formula $\lambda_t^1 \wedge \lambda_x^1 \wedge \lambda_\theta^1$ is denoted by

$$\lambda_x'^1 \wedge \lambda_\theta'^1. \tag{3}$$

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, $\lambda^{p,2}(t, x, \Theta)$ is constructed as a disjunction of formulae $\lambda_t^2 \wedge \lambda_x^2 \wedge \lambda_\theta^2$ where λ_t^2 is one among

$$t = \alpha_2, \quad t \equiv_{a,\geq} \alpha_2, \quad t = \alpha_2 - x, \quad t \equiv_{a,\geq} \alpha_2 - x. \tag{4}$$

Recall that $\lambda^{p,2}(t, x, \Theta)$ describes the duration t_2 of runs $\rho_2 = (p, x_1) \rightsquigarrow (f, x_2)$ for which x_1 satisfies $x = \alpha$. Thus in (4), x can be replaced by α and (4) becomes

$$t = \alpha_2, \quad t \equiv_{a,\geq} \alpha_2, \quad t = \alpha_2 - \alpha, \quad t \equiv_{a,\geq} \alpha_2 - \alpha.$$

This shows that λ_t^2 is now of the form

$$t = \beta \quad \text{or} \quad t \equiv_{a,\geq} \beta. \tag{5}$$

Moreover λ_x^2 becomes a θ-conjunction when x is replaced by α. The modified formula $\lambda_x^2 \wedge \lambda_\theta^2$ is denoted by

$$\lambda_\theta'^2. \tag{6}$$

Finally, we can describe $t_0 = t_1 + t_2$. By (2) and (5), it has the form

$$t_0 = v(\alpha) - x_0 + v(\beta) \quad \text{or} \quad t_0 \equiv_{a,\geq} v(\alpha) - x_0 + v(\beta). \tag{7}$$

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 (see (7)) $t = \alpha - x + \beta$ or $t \equiv_{a,\geq} \alpha - x + \beta$ and $\lambda_x \wedge \lambda_\theta$ has the form (see (3 and (6)) $\lambda_x'^1 \wedge \lambda_\theta'^1 \wedge \lambda_\theta'^2$.

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

$$\lambda(t, x, \Theta) \wedge \mathcal{I}_x(i)(x, \Theta) \wedge \bigwedge_{x \leq \alpha \in f} t \leq \alpha - x. \tag{8}$$

$\square$

Remark 22 Suppose that $\mathcal{A}$ is an automaton such that $\mathcal{I}(i)$ equals $x = 0$ for each initial state $i \in I$. Then formula $\lambda(t, x, \Theta)$ of Proposition 21 contains the x-atom $x = 0$ (see (8)). Hence, if $\lambda(t_0, x_0, v(\Theta))$ is TRUE, then necessarily $x_0 = 0$, which can been interpreted as a reset of the clock. This remark will be used in the next subsection.

4.4 Durations in General

This subsection is devoted to the proofs of Propositions 12 and 13. Here there is no longer a restriction on the automaton: it is *any* automaton as in Definition 2. This automaton is supposed to be normalized by Proposition 17. Thus, given a state q, the edges (p, τ, g, r, q) entering q all have the same r. We call q a *reset-state* in case $r = \{x\}$. The set of reset-states of $\mathcal{A}$ is denoted by Q_R.

Let $\mathcal{A} = (Q, E, \mathcal{L}, \mathcal{I})$ be an automaton. Let us fix two states q, q', a parameter valuation v, a clock value x_0. We denote by

$$\mathcal{R}_{q,q'}(\mathcal{A}^v, x_0)$$

the set of runs $\rho = (q, x_0) \rightsquigarrow (q', \cdot)$ in $\mathcal{A}^v$. Let us study this set.

A run ρ in $\mathcal{R}_{q,q'}(\mathcal{A}^v, x_0)$ 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 possibly 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 21. 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 $\lambda_t \wedge \lambda_{\leq} \wedge \lambda_x \wedge \lambda_\theta$. So the total duration D_ρ is equal to the sum $\Sigma_{1 \leq j \leq k} \mathrm{D}_{\rho_j}$. We will see that the durations D_ρ of runs $\rho \in \mathcal{R}_{q,q'}(\mathcal{A}^v, x_0)$ can be symbolically represented thanks to rational expressions on an alphabet whose letters are the formulae $\lambda_t \wedge \lambda_{\leq} \wedge \lambda_x \wedge \lambda_\theta$ that appear in the $\lambda^j(t, x, \Theta)$'s. Thanks to this symbolic description and because our logic is restricted to $\mathrm{PTCTL}_{\{<,\leq,>,\geq\}}$, we will be able to prove Propositions 12 and 13. It should be noted that the durations D_ρ of runs $\rho \in \mathcal{R}_{q,q'}(\mathcal{A}^v, x_0)$ cannot be described by a Presburger formula as in Proposition 21, otherwise the model-checking problem for PTCTL would be decidable (see Corollary 9).

Let us now explain in details all these ideas.

In a first step, we construct from $\mathcal{A}$ several reset-free normalized automata as in Hypothesis $(*)$. The construction is a standard one in automata theory. Runs ρ_j mentioned before will be runs in these automata and their durations will be described thanks to Proposition 21.

First construction For each couple (p, p') of states of $\mathcal{A}$ such that $p \in \{q\} \cup Q_R$ and $p' \in \{q'\} \cup Q_R$, we construct from $\mathcal{A}$ the following reset-free automaton $\mathcal{A}_{p,p'} = (Q', I', F', E', \mathcal{L}', \mathcal{I}')$. The set Q' of states is $(Q \setminus Q_R) \cup \{\overline{p}, \overline{p}'\}$ where $\overline{p}, \overline{p}'$ are copies of p, p'. The unique initial state is $\overline{p}$ and the unique final state is $\overline{p}'$. Let $\mathcal{L}'(\overline{p}) = \mathcal{L}(p)$ and $\mathcal{L}'(\overline{p}') = \mathcal{L}(p')$. Let $\mathcal{I}'(\overline{p})$ be equal to $\mathcal{I}(p)$ if $p = q$ and to $(\mathcal{I}(p) \wedge x = 0)$[10] if $p \neq q$. Let $\mathcal{I}'(\overline{p}')$ be equal to $\mathcal{I}(p')$ if $p' \notin Q_R$ and to $(\mathcal{I}(p') \wedge x = 0)$[11] if $p' \in Q_R$. The set E' of edges is the union of E restricted to $Q \setminus Q_R$ with the next set of new edges[11]

$$\begin{array}{ll} (\overline{p}, \tau, g, r, p_1) & \text{if } (p, \tau, g, r, p_1) \in E \\ (p_1, \tau, g, \varnothing, \overline{p}') & \text{if } (p_1, \tau, g, r, p') \in E \\ (\overline{p}, \tau, g, \varnothing, \overline{p}') & \text{if } (p, \tau, g, r, p') \in E. \end{array}$$

[10] The x-atom $x = 0$ imposes a reset of the clock at state p (see Remark 22)

[11] As $\mathcal{A}_{p,p'}$ must satisfy Hypothesis $(*)$, no reset can appears on the edges

In this way, automaton $\mathcal{A}_{p,p'}$ satisfies Hypothesis $(*)$.

Let $p \in \{q\} \cup Q_R$ and $p' \in \{q'\} \cup Q_R$. We define x_1 to be equal to x_0 if $p = q$, and to 0 if $p \neq q$. 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 states of the run). The durations of runs in $\mathcal{R}(\mathcal{A}^v_{p,p'}, x_1)$ are described by formula $\lambda^{p,p'}(t, x, \Theta)$ of Proposition 21. This formula is a disjunction $\bigvee_j \lambda^{p,p',j}$ of formulae

$$\lambda^{p,p',j} = \lambda^{p,p',j}_t \wedge \lambda^{p,p',j}_{\leq} \wedge \lambda^{p,p',j}_x \wedge \lambda^{p,p',j}_\theta. \tag{9}$$

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.

In a second step, we construct another automaton from $\mathcal{A}$ in a way to show how a run of $\mathcal{R}_{q,q'}(\mathcal{A}^v, x_0)$ is decomposed into a sequence of runs ρ_j according to reset-states of $\mathcal{A}$. This automaton will be a classical automaton [8].

Second construction We construct an automaton $\mathcal{B}$ over the alphabet B as follows. The set of states equals $Q_R \cup \{q, q'\}$ 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').

So, any run ρ of $\mathcal{R}_{q,q'}(\mathcal{A}^v, x_0)$ is map into a path in $\mathcal{B}$ from q to q' which indicates how ρ is decomposed according to reset-states of $\mathcal{A}$. The duration of ρ is symbolically represented by the word that labels the corresponding path in $\mathcal{B}$. Hence the set of durations of runs of $\mathcal{R}_{q,q'}(\mathcal{A}^v, x_0)$ is *symbolically represented* by the rational subset accepted by $\mathcal{B}$. We denote by

$$L_{q,q'}$$

this subset of B^*. Any word of $L_{q,q'}$ has *at most one* letter that is non reset (the first letter of the word).

We now study in details rational expressions over the alphabet B and in particular the rational expression defining $L_{q,q'}$.

Rational expressions Let L^+ be denoting $L^* \backslash \{\epsilon\}$ with ϵ denoting the empty word and $\mathrm{Rat}_B(\cdot, +)$ be the smallest family closed under $\cdot$ and $+$, and containing B. One can 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

$$L_{q,q'} = \bigcup_i L_i \tag{10}$$

with

$$L_i = \{\epsilon\} \quad \text{or} \quad L_i = \{b_i\} \quad \text{or} \quad L_i = b_i \cdot K_i$$

such that $b_i \in B, K_i \in \mathrm{Rat}_{B_R}(\cdot,+)$. The set $\mathcal{R}_{q,q'}(\mathcal{A}^v, x_0)$ is decomposed into

$$\mathcal{R}_{q,q'}(\mathcal{A}^v, x_0) = \bigcup_i \mathcal{R}_i \tag{11}$$

according to (10).

An non empty word of $L_{q,q'}$ is a sequence $b_1 b_2 \cdots b_n \in B^+$. The first letter b_1 describes runs from state q to some reset-state p_1, the clock value at q is x_0. Each letter b_i, $i \geq 2$, is a reset-letter. If $2 \leq i < n$, b_i describes runs from reset-state p_{i-1} to reset-state p_i, the clock value at p_{i-1} is 0. If $i = n$, b_i describes runs from reset-state p_{n-1} to state q', the clock value at p_{n-1} is 0. Let

$$\lambda_t^i \wedge \lambda_\leq^i \wedge \lambda_x^i \wedge \lambda_\theta^i \tag{12}$$

be the formula associated to each letter b_i, $i \geq 1$ (see (9)). Whenever $i \geq 2$, λ_x^i contains the x-atom $x = 0$ by Remark 22 and Definition of automaton $\mathcal{A}_{p,p'}$. In this case, we prefer[12] to work with the equivalent formula

$$\kappa_t^i \wedge \kappa_\leq^i \wedge \kappa_\theta^i \tag{13}$$

such that x has been replaced by 0 in (12) (in particular, λ_x becomes a θ-conjunction). In this formula κ_t^i is a t-atom of the form $t = \alpha$ or $t \equiv_{a,\geq} \alpha$, $\kappa_\leq^i$ is a conjunction of t-atoms of the form $t \leq \alpha$ and κ_θ^i is a θ-conjunction.

The concatenation $b_1 \cdot b_2 \cdot \cdots \cdot b_n$ is interpreted as follows. It is the *sum* $t_1 + t_2 + \cdots + t_n$ of the durations $t_1, t_2, \ldots, t_n$ respectively described by $\lambda_t^1, \kappa_t^2, \ldots, \kappa_t^n$. It is the *conjunction* of the related constraints

$$(\lambda_\leq^1 \wedge \kappa_\leq^2 \wedge \cdots \kappa_\leq^n) \wedge \lambda_x^1 \wedge (\lambda_\theta^1 \wedge \kappa_\theta^2 \wedge \cdots \kappa_\theta^n).$$

Formulae $\lambda_\leq^1, \kappa_\leq^2, \ldots \kappa_\leq^n$ impose upper bounds on $t_1, t_2, \ldots, t_n$. The x-conjunction imposes constraints on the clock value x_0. The θ-conjunction $(\lambda_\theta^1 \wedge \kappa_\theta^2 \wedge \cdots \kappa_\theta^n)$ impose constraints on the parameters.

In the next lemmas, we show that certain properties of runs in $\mathcal{R}_i$ can be expressed in Presburger arithmetics thanks to the symbolic representation L_i of $\mathcal{R}_i$ (see (10) and (11)). After these lemmas, we will be fully equipped to prove Propositions 12 and 13. Note that Proposition 13 can only be proved with $\sim$ limited to $\{<, \leq, >, \geq\}$, otherwise the model-checking problem for PTCTL would be decidable.

Lemma 23 *One can construct a $\mathcal{B}_{x,\Theta}$ formula* $\mathrm{NonEmpty}_{L_i}(x, \theta)$ *such that for any valuation v and any clock value x_0,* $\mathrm{NonEmpty}_{L_i}(x_0, v(\theta))$ *is* TRUE *iff $\mathcal{R}_i$ is non empty.*

Proof Runs of $\mathcal{R}_i$ have durations that are symbolically represented by the words of L_i. Let us construct formula $\mathrm{NonEmpty}_{L_i}$ by induction on the rational

[12] The sequence $b_1 b_2 \cdots b_n$ symbolically represents certain runs of $\mathcal{R}_{q,q'}(\mathcal{A}^v, x_0)$. We are only interested in the initial clock value x_0 treated by formula λ_x^i of b_1.

expression defining L_i (see (10)). This formula will be equal to $\eta_x \wedge \eta_\theta$ with η_x an x-conjunction imposing constraints on the clock and η_θ a θ-conjunction imposing constraints on the parameters.

Suppose $L_i = \{\epsilon\}$, then $\mathrm{NonEmpty}_{L_i}(x,\Theta)$ equals $x = 0$ is q is a reset-state and $\mathcal{I}(q)(x,\Theta)$ otherwise. Indeed, under these constraints, $\mathcal{R}_i$ is non empty since it contains the empty run with the null duration. Suppose that $L_i = \{b_i\}$ with $b_i \in B$ and associated formula $\lambda^i_t \wedge \lambda^i_\leq \wedge \lambda^i_x \wedge \lambda^i_\theta$. Recall that λ^i_t is one among the t-atoms $t = \alpha$, $t = \alpha - x$, $t \equiv_{a,\geq} \alpha$ or $t \equiv_{a,\geq} \alpha - x$ and that $\lambda^i_\leq$ is of the form $\bigwedge_\beta t \leq \beta - x$. It follows that the non emptiness of $\mathcal{R}_i$ can be expressed thanks to the minimum duration $t = \alpha$ ($t = \alpha - x$ resp.) of runs in $\mathcal{R}_i$. Then

$$\begin{aligned} \mathrm{NonEmpty}_{L_i}(x,\Theta) &= (\bigwedge_\beta \alpha \leq \beta - x) \wedge \lambda_x \wedge \lambda_\theta \\ (&= (\bigwedge_\beta \alpha \leq \beta) \wedge \lambda_x \wedge \lambda_\theta \quad \text{resp.}) \end{aligned} \tag{14}$$

Suppose now that $L_i = b_i \cdot K_i$ with $b_i \in B$ and $K_i \in \mathrm{Rat}_{B_R}(\cdot,+)$. Let us first prove by induction on the rational expression defining K_i that $\mathrm{NonEmpty}_{K_i}(\Theta)$ equals some θ-conjunction η_θ.[13] Let $K_i = \{b_i\}$ with $b_i \in B_R$. We obtain a formula similar to (14) where x is replaced by 0 (see(13)), so

$$\mathrm{NonEmpty}_{K_i}(\Theta) = (\bigwedge_\beta \alpha \leq \beta) \wedge \kappa_\theta.$$

Suppose that $K_i = K \cdot K'$ and formulae $\mathrm{NonEmpty}_K$, $\mathrm{NonEmpty}_{K'}$ have been constructed by induction. Then $\mathrm{NonEmpty}_{K_i}(\Theta) = \mathrm{NonEmpty}_K(\Theta) \wedge \mathrm{NonEmpty}_{K'}(\Theta)$ because the non emptiness of $\mathcal{R}_i$ requires the non emptiness of both K and K'. If $K_i = K^+$, then $\mathrm{NonEmpty}_{K_i}(\Theta) = \mathrm{NonEmpty}_K(\Theta)$ because conjunction in an idempotent operation. Finally for $L_i = b_i \cdot K_i$, we get $\mathrm{NonEmpty}_{L_i}(x,\Theta) = \mathrm{NonEmpty}_{\{b_i\}}(x,\Theta) \wedge \eta_\theta$ where $\mathrm{NonEmpty}_{\{b_i\}}(x,\Theta)$ is formula (14) and η_θ is the formula just constructed for K_i. □

Lemma 24 *One can construct a $\mathcal{B}_{x,\Theta}$ formula $\mathrm{NonNull}_{L_i}(x,\theta)$ such that for any valuation v and any clock value x_0, $\mathrm{NonNull}_{L_i}(x_0, v(\theta))$ is* TRUE *iff $\mathcal{R}_i$ contains a run with a non null duration.*

Proof The proof is in the same vein as for Lemma 23 with a similar form $\eta_x \wedge \eta_\theta$ for $\mathrm{NonNull}_{L_i}(x,\theta)$.

If $L_i = \{\epsilon\}$, then clearly $\mathrm{NonNull}_{L_i}(x,\theta) = \perp$. If $L_i = \{b_i\}$ with $b_i \in B$ and associated formula $\lambda^i_t \wedge \lambda^i_\leq \wedge \lambda^i_x \wedge \lambda^i_\theta$. Let us study as before formulae λ^i_t and $\lambda^i_\leq$, where $\lambda^i_\leq = \bigwedge_\beta (t \leq \beta - x)$. If λ^i_t equals $t = \alpha$, then t is non null iff $\alpha > 0$. Then $\mathrm{NonNull}_{L_i}(x,\Theta)$ is the formula $(\alpha > 0) \wedge (\bigwedge_\beta \alpha \leq \beta - x) \wedge \lambda^i_x \wedge \lambda^i_\theta$. When λ^i_t is $t = \alpha - x$, we have a similar formula with t non null if $\alpha - x > 0$. If λ^i_t equals $t \equiv_{a,\geq} \alpha$, then a possible non null value for t is either α if $\alpha > 0$

[13]There is no term η_x since $K_i \subseteq B^+_R$, that is, $x = 0$ (see (13)).

or a if $\alpha = 0$. We get formula $\mathrm{NonNull}_{L_i}(x, \Theta)$ equal to $((\alpha > 0 \wedge \bigwedge_\beta(\alpha \leq \beta - x)) \vee (\alpha = 0 \wedge \bigwedge_\beta(a \leq \beta - x))) \wedge \lambda_x^i \wedge \lambda_\theta^i$. A similar argument holds if λ_t^i equals $t \equiv_{a,\geq} \alpha - x$.

Let $L_i = b_i \cdot K_i$, with $b_i \in B$ and $K_i \in \mathrm{Rat}_{B_R}(\cdot, +)$. Let us first construct formula $\mathrm{NonNull}_{K_i}(\Theta)$ by induction on K_i. This formula will be a θ-conjunction. If $K_i = \{b_i\}$ with $b_i \in B_R$, we get a formula $\mathrm{NonNull}_{K_i}$ as for the case $L_i = \{b_i\}$ such that x is replaced by 0. If $K_i = K \cdot K'$, then there exists a non null duration in K_i iff there exists some duration in K and some other in K' and one of them is non null. Thus $\mathrm{NonNull}_{K_i}(\Theta)$ equals $(\mathrm{NonNull}_K(\Theta) \wedge \mathrm{NonEmpty}_{K'}(\Theta)) \vee (\mathrm{NonEmpty}_K(\Theta) \wedge \mathrm{NonNull}_{K'}(\Theta))$. If $K_i = K^+$, then $\mathrm{NonNull}_{K_i}(\Theta) = \mathrm{NonNull}_K(\Theta)$. Finally, for $L_i = b_i \cdot K_i$, we get the formula $(\mathrm{NonNull}_{\{b_i\}}(x, \Theta) \wedge \mathrm{NonEmpty}_{K_i}(\Theta)) \vee (\mathrm{NonEmpty}_{\{b_i\}}(x, \Theta) \wedge \mathrm{NonNull}_{K_i}(\Theta))$.
□

Lemma 25 *One can construct a $\mathcal{B}_{x,\Theta}$ formula* $\mathrm{NonZeno}_{L_i}(x, \theta)$ *such that for any valuation v and any clock value x_0,* $\mathrm{NonZeno}_{L_i}(x_0, v(\theta))$ *is* TRUE *iff $\mathcal{R}_i$ contains runs with arbitrarily large durations.*

Proof The proof is again similar.

Suppose $L_i = \{\epsilon\}$, then clearly $\mathrm{NonZeno}_{L_i}(x, \Theta) = \bot$. Let $L_i = \{b_i\}$ with $b_i \in B$ and associated formula $\lambda_t^i \wedge \lambda_\leq^i \wedge \lambda_x^i \wedge \lambda_\theta^i$. If λ_t^i equals $t = \alpha$ or $t = \alpha - x$, then $\mathrm{NonZeno}_{L_i}(x, \Theta) = \bot$. If λ_t^i equals $t \equiv_{a,\geq} \alpha$ or $t \equiv_{a,\geq} \alpha - x$, then t is arbitrarily large iff $\lambda_\leq^i = \top$. In this case, $\mathrm{NonZeno}_{L_i}(x, \Theta) = \lambda_x^i \wedge \lambda_\theta^i$, otherwise $\mathrm{NonZeno}_{L_i}(x, \Theta) = \bot$.

Suppose now that $L_i = b_i \cdot K_i$. We begin to construct a θ-conjunction $\mathrm{NonZeno}_{K_i}(\Theta)$ by induction on K_i. If $K_i = \{b_i\}$ with $b_i \in B_R$, then the formula is as in the case $L_i = \{b_i\}$ with x replaced by 0. If $K_i = K \cdot K'$, then $\mathrm{NonZeno}_{K_i}(\Theta)$ equals $(\mathrm{NonZeno}_K(\Theta) \wedge \mathrm{NonEmpty}_{K'}(\Theta)) \vee (\mathrm{NonEmpty}_K(\Theta) \wedge \mathrm{NonZeno}_{K'}(\Theta))$. If $K_i = K^+$, then K_i has arbitrarily large durations iff K contains a non null duration, that is $\mathrm{NonZeno}_{K_i}(\Theta) = \mathrm{NonNull}_K(\Theta)$. Thus we get for $L_i = b_i \cdot K_i$ the formula $(\mathrm{NonZeno}_{\{b_i\}}(x, \Theta) \wedge \mathrm{NonEmpty}_{K_i}(\Theta)) \vee (\mathrm{NonEmpty}_{\{b_i\}}(x, \Theta) \wedge \mathrm{NonZeno}_{K_i}(\Theta))$. □

Lemma 26 *One can construct a Presburger formula* $\mathrm{Min}_{L_i}(t, x, \theta)$ *such that for any valuation v and any clock value x_0,* $\mathrm{Min}_{L_i}(t_0, x_0, v(\theta))$ *is* TRUE *iff t_0 is the minimum duration of runs of $\mathcal{R}_i$. This formula is equal to $\mu_t \wedge \mu_x \wedge \mu_\theta$ such that μ_t is of the form $t = \alpha$ or $t = \alpha - x$, μ_x is an x-conjunction and μ_θ is a θ-conjunction.*

Proof In this proof, we have to describe the minimum duration by the variable t and the constraints on it by μ_x and μ_θ.

Let $L_i = \{\epsilon\}$, then $\mathrm{Min}_{L_i}(t, x, \Theta)$ is equal to $(t = 0) \wedge (x = 0)$ if q is a reset-state, and to $(t = 0) \wedge \mathcal{I}(q)(x, \Theta)$ otherwise. Let $L_i = \{b_i\}$ with $b_i \in B$. Then looking at the form of λ_t^i, the minimum duration equals α ($\alpha - x$ resp.)

(see (14)). Therefore formula $\mathrm{Min}_{L_i}(t,x,\Theta)$ is equal to

$$(t=\alpha)\wedge(\bigwedge_{\beta}\alpha\leq\beta-x)\wedge\lambda^i_x\wedge\lambda^i_\theta \tag{15}$$

$$(\qquad (t=\alpha-x)\wedge(\bigwedge_{\beta}\alpha\leq\beta)\wedge\lambda^i_x\wedge\lambda^i_\theta \quad \text{resp.})$$

Suppose $L_i=b_i\cdot K_i$. Let us begin to construct formula $\mathrm{Min}_{K_i}(t,\Theta)$ the form of which will be $\mu_t\wedge\mu_\theta$. If $K_i=\{b_i\}$ with $b_i\in B_R$, then $\mathrm{Min}_{K_i}(t,\Theta)$ equals (15) with x replaced by 0. If $K_i=K\cdot K'$, then the minimum duration in K_i equals the sum of the minimum durations in K and K'. Hence, if $\mathrm{Min}_K(t,\Theta)=(t=\alpha)\wedge\mu_\theta$ and $\mathrm{Min}_{K'}=(t=\alpha')\wedge\mu'_\theta$, then $\mathrm{Min}_{K_i}(t,\Theta)$ is equal to $(t=\alpha+\alpha')\wedge\mu_\theta\wedge\mu'_\theta$. If $K_i=K^+$, then the minimum duration in K_i is the minimum duration in K, i.e. $\mathrm{Min}_{K_i}(t,\Theta)=\mathrm{Min}_K(t,\Theta)$. Let us come back to $L_i=b_i\cdot K_i$. Let $\mathrm{Min}_{\{b_i\}}(t,x,\Theta)$ be equal to (15) and $\mathrm{Min}_{K_i}(t,\Theta)$ be equal $(t=\alpha')\wedge\mu_\theta$. Then $\mathrm{Min}_{L_i}(t,\Theta)$ is equal to $(t=\alpha+\alpha')\wedge(\bigwedge_\beta\alpha\leq\beta-x)\wedge\lambda^i_x\wedge\lambda^i_\theta\wedge\mu_\theta$ (resp. $(t=\alpha+\alpha'-x)\wedge(\bigwedge_\beta\alpha\leq\beta)\wedge\lambda^i_x\wedge\lambda^i_\theta\wedge\mu_\theta)$. □

In the next lemma, we are going to construct a formula $\mathrm{Max}_{L_i}(t,x,\Theta)$ that describes the maximum duration t in L_i. Note that durations t in L_i can be arbitrarily large (see Lemma 25). We will thus denote symbolically by $t=\infty$ the (non existing) maximum duration.

Lemma 27 *One can construct a formula* $\mathrm{Max}_{L_i}(t,x,\theta)$ *such that for any valuation v and any clock value x_0,* $\mathrm{Max}_{L_i}(t_0,x_0,v(\theta))$ *is* TRUE *iff t_0 is the maximum duration of runs of $\mathcal{R}_i$. This formula is equal to a disjunction of formulae $M_t\wedge M_x\wedge M_\theta$ such that M_t is of the form $t=\alpha$, $t=\alpha-x$ or $t=\infty$, M_x is an x-conjunction and M_θ is a θ-conjunction.*

Proof If $L_i=\{\epsilon\}$, then Max_{L_i} is $(t=0)\wedge(x=0)$ if q is a reset-state, and to $(t=0)\wedge\mathcal{I}(q)(x,\Theta)$ otherwise. Let $L_i=\{b_i\}$ with $b_i\in B$. Let us study λ^i_t and $\lambda^i_\leq$ equal to $\bigwedge_\beta(t\leq\beta-x)$. If λ^i_t is $t=\alpha$, then $\mathrm{Max}_L(t,x,\Theta)$ equals $\lambda^i_t\wedge\bigwedge_\beta(\alpha\leq\beta-x)\wedge\lambda^i_x\wedge\lambda^i_\theta$. A similar formula holds when λ^i_t equals $t=\alpha-x$. If λ^i_t is $t\equiv_{a,\geq}\alpha$ with $\lambda^i_\leq=\top$, then $\mathrm{Max}_L(t,x,\Theta)$ equals $(t=\infty)\wedge\lambda^i_x\wedge\lambda^i_\theta$. Suppose that λ^i_t is $t\equiv_{a,\geq}\alpha$ with $\lambda^i_\leq$ being a non empty conjunction $\bigwedge_\beta(t\leq\beta-x)$. Then the maximum duration is the greatest value $\alpha+ay$, for some $y\in\mathbb{N}$, which is less than the smallest among the $\beta-x$'s, denoted by $\beta'-x$. Assume that $\beta'-x\equiv b \bmod a$ and $\alpha\equiv c \bmod a$ 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 $t=\beta'-x-(b-c)$ under the condition m_θ equal to $t\geq\alpha$, i.e. $\beta'-x-(b-c)\geq\alpha$. If $b<c$, then M_t equals $t=\beta'-x-(a+b-c)$ under the condition m_θ equal to $\beta'-x-(a+b-c)\geq\alpha$. Thus $\mathrm{Max}_L(t,x,\Theta)$ is a disjunction over the different possible values of β', b and c of formulae

$$M_t\wedge m_\theta\wedge\lambda_\theta\wedge M_{\beta',x,b,c}$$

such that $M_{\beta',b,c}$ is the conjunction

$$(\bigwedge_{\beta} \beta' \leq \beta) \wedge (\beta' - x \equiv_{a,\geq} b) \wedge (\alpha \equiv_{a,\geq} c).$$

A similar argument can be done when λ_t^i is $t \equiv_{a,\geq} \alpha - x$.

Let $L_i = b_i \cdot K_i$. Let us first construct Max_{K_i}. This formula will contain no M_x. If $K_i = \{b_i\}$ with $b_i \in B_R$, then all the proof done before for $L_i = \{b_i\}$ can be repeated with x replaced by 0. Suppose that $K_i = K \cdot K'$ and that $\mathrm{Max}_K(t, \Theta)$ and $\mathrm{Max}_{K'}(t, \Theta)$ are a disjunction of formulae $M_t \wedge M_\theta$ and $M'_t \wedge M'_\theta$ respectively. If $M_t = (t = \alpha)$ and $M'_t = (t = \alpha')$, then $\mathrm{Max}_{K_i}(t, \Theta)$ contains the conjunction $(t = \alpha + \alpha') \wedge M_\theta \wedge M'_\theta$. If $M_t = (t = \infty)$ or $M'_t = (t = \infty)$, then $\mathrm{Max}_{K_i}(t, \Theta)$ contains the conjunction $(t = \infty) \wedge M_\theta \wedge M'_\theta$. Suppose that $K_i = K^+$, then the maximum duration equals ∞ if L contains a non null duration (see Lemma 24), and 0 otherwise. Thus $\mathrm{Max}_{K_i}(t, \Theta)$ is the formula $((t = \infty) \wedge \mathrm{NonNull}_K(\Theta)) \vee ((t = 0) \wedge \neg \mathrm{NonNull}_K(\Theta))$. Formula $\mathrm{Max}_{L_i}(t, x, \Theta)$ for $L_i = b_i \cdot K_i$ can be easily constructed (as done before for $K \cdot K'$). □

Proof of Proposition 12. Let us prove that one can construct a $\mathcal{B}_{x,\Theta}$ formula $\mathrm{Run}_q(x, \Theta)$ such that for any valuation v and any clock value x_0, $\mathrm{Run}_q(x_0, v(\Theta))$ is TRUE iff there exists an infinite run in $\mathcal{A}^v$ starting with (q, x_0). Such a run exists iff for some $q' \in Q$, there exist runs in $\mathcal{R}_{q,q'}(\mathcal{A}^v, x_0)$ with arbitrarily large durations. As $\mathcal{R}_{q,q'}(\mathcal{A}^v, x_0) = \bigcup_i \mathcal{R}_i$, this is equivalent to say that some $\mathcal{R}_i$ contains runs with arbitrarily large durations. By Lemma 25, it follows that formula $\mathrm{Run}_q(x, \Theta)$ is equal to $\bigvee_{q' \in Q} \bigvee_i \mathrm{NonZeno}_{L_i}(x, \Theta)$. □

Proof of Proposition 13. Let γ be a linear term and $\sim \in \{<, \leq, >, \geq\}$. We have to show that there exists a $\mathcal{B}_{x,\Theta}$ formula $\mathrm{Duration}^{\sim\gamma}_{q,q'}(x, \Theta)$ such that for any valuation v and any clock value x_0, $\mathrm{Duration}^{\sim\gamma}_{q,q'}(x_0, v(\Theta))$ is TRUE iff there exists a run in $\mathcal{R}_{q,q'}(\mathcal{A}^v, x_0)$ with duration $t \sim v(\gamma)$.

(1) We begin with $\sim \in \{<, \leq\}$. To test if there exists a run in $\mathcal{R}_{q,q'}(\mathcal{A}^v, x_0)$ with duration $t \sim v(\gamma)$ is equivalent to test that $t_{min} \sim v(\gamma)$ with t_{min} being the minimum duration of runs in $\mathcal{R}_{q,q'}(\mathcal{A}^v, x_0)$. By Lemma 26, the minimum duration for each $\mathcal{R}_i$ is expressed by formula $\mathrm{Min}_{L_i}(t, x, \Theta)$. This formula is of the form $\mu_t \wedge \mu_x \wedge \mu_\theta$ with μ_t equal to $t = \alpha$ or $t = \alpha - x$. Therefore $\mathrm{Duration}^{\sim\gamma}_{q,q'}(x, \Theta)$ is equal to $\bigvee_i \mathrm{Duration}_i$, where each $\mathrm{Duration}_i$ is obtained by modifying Min_{L_i} as follows: any formula μ_t equal to $t = \alpha$ ($t = \alpha - x$ resp.) is replaced by formula $\alpha \sim \gamma$ ($\alpha - x \sim \gamma$ resp.).

(2) We now turn to $\sim \in \{>, \geq\}$. The approach is similar but with the maximum (instead of minimum) duration. By Lemma 27, the maximum duration for each $\mathcal{R}_i$ is expressed by formula $\mathrm{Max}_{L_i}(t, x, \Theta)$. This formula is a disjunction of formulae $M_t \wedge M_x \wedge M_\theta$ with M_t equal to $t = \alpha$, $t = \alpha - x$ or $t = \infty$. It follows that $\mathrm{Duration}^{\sim\gamma}_{q,q'}(x, \Theta)$ is equal to $\bigvee_i \mathrm{Duration}_i$, where each $\mathrm{Duration}_i$ is obtained by modifying Max_{L_i} in the following way. If M_t equals $t = \alpha$, $t = \alpha - x$ or $t = \infty$, then it is replaced by formula $\alpha \sim \gamma$, $\alpha - x \sim \gamma$ or $\top$ respectively. □

5 Conclusion

In this paper, we have completely studied the model-checking problem and the parameter synthesis problem of the logic PTCTL, an extension of TCTL with parameters, over one parametric clock timed automata. On the negative side, we showed that the model-checking problem is undecidable. 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 [4]) to express durations of runs of a timed automaton using Presburger arithmetic. With this approach, the model-checking problem and the parameter synthesis problem are syntactically translated into Presburger arithmetic which has a decidable theory and an effective quantifier elimination. The model checking problem is translated into a Presburger sentence inside which the Presburger decidability process looks for semantic inconsistencies between the parameters and the parametric clock. The parameter synthesis problem asks for which values of the parameters is a PTCTL formula true at a given configuration of the timed automaton. Thanks to Presburger quantifier elimination, this problem is solved by expressing the values of the parameters in terms of the operations $+$, $\leq$ and $\equiv \mathrm{mod} a$, $a \in \mathbb{N}^+$.

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 illustrated in the introduction the kind of properties that can be expressed and automatically verified in our framework. With the Presburger approach, we obtained complete solutions and clearly indicated the borderline between decidability and undecidability.

Future works could be the following ones. No complexities issues are given in this paper and only the discrete time is considered. Presburger theory is decidable with the high 3ExpTime complexity. More efficient algorithms should be designed for particular fragments of $\mathrm{PTCTL}_{\{<,\leq,>,\geq\}}$. The extension to dense timed models of the method proposed in this paper should be investigated.

References

[1] 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.

[2] 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.

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

[4] V. Bruyère, E. Dall'olio, and J.-F. Raskin. Durations, parametric model-checking in timed automata with Presburger arithmetic. In *STACS 2003, 20th Annual Symposium on Theoretical Aspects of Computer Science, Berlin, Germany, February 27 - March 1, 2003, Proceedings*, volume 2607 of *Lecture Notes in Computer Science*, pages 687–698. Springer, 2003.

[5] S. Eilenberg. *Automata, Languages, and Machines*, volume A. Academic Press, 1974.

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

[7] Thomas Hune, Judi Romijn, Marielle Stoelinga, and Frits Vaandrager. Linear parametric model checking of timed automata. *Journal of Logic and Algebraic Programming*, 52-53:183–220, 2002.

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

[9] 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.

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