Quantitative Robustness Analysis of Flat Timed Automata*

Rémi Jaubert** and Pierre-Alain Reynier

LIF, Université Aix-Marseille & CNRS, France
{remi.jaubert,pierre-alain.reynier}@lif.univ-mrs.fr

Abstract. Whereas formal verification of timed systems has become a very active field of research, the idealized mathematical semantics of timed automata cannot be faithfully implemented. Recently, several works have studied a parametric semantics of timed automata related to implementability: if the specification is met for some positive value of the parameter, then there exists a correct implementation. In addition, the value of the parameter gives lower bounds on sufficient resources for the implementation. In this work, we present a symbolic algorithm for the computation of the parametric reachability set under this semantics for flat timed automata. As a consequence, we can compute the largest value of the parameter for a timed automaton to be safe.

1 Introduction

Verification of real-time systems. In the last thirty years, formal verification of reactive, critical, or embedded systems has become a very active field of research in computer science. It aims at checking that (the model of) a system satisfies (a formula expressing) its specifications. The importance of taking real-time constraints into account in verification has quickly been understood, and the model of timed automata [AD94] has become one of the most established models for real-time systems, with a well studied underlying theory, the development of mature model-checking tools (UPPAAL [BDL04], KRONOS [BDM+98], ...), and numerous success stories.

Implementation of real-time systems. Implementing mathematical models on physical machines is an important step for applying theoretical results on practical examples. This step is well-understood for many untimed models that have been studied (*e.g.*, finite automata, pushdown automata). In the timed setting, while timed automata are widely-accepted as a framework for modelling real-time aspects of systems, it is known that they cannot be faithfully implemented on finite-speed CPUs [CHR02]. Studying the "implementability" of timed automata is thus a challenging issue of obvious theoretical and practical interest.

A semantical approach. Timed automata are governed by a mathematical, idealized semantics, which does not fit with the digital, imprecise nature of the hardware on

* Partly supported by the French projects ECSPER (ANR-09-JCJC-0069) and DOTS (ANR-06-SETI-003), and by the European project QUASIMODO (FP7-ICT-STREP-214755).
** Funded by a doctoral grant of "Conseil Régional Provence-Alpes-Côte d'Azur".

M. Hofmann (Ed.): FOSSACS 2011, LNCS 6604, pp. 229–244, 2011.

which they will possibly be implemented. An *implementation semantics* has been defined in [DDR05] in order to take the hardware into account: that semantics models a digital CPU which, every δ_P time units (at most), reads the value of the digital clock (updated every δ_L time units), computes the values of the guards, and fires one of the available transitions. A timed automaton is then said to be *implementable* if there exist positive values for those parameters (δ_P and δ_L) for which, under this new semantics, the behaviours of the automaton satisfy its specification. In order to study it efficiently, this semantics is over-approximated by the *AASAP semantics*, which consists in "enlarging" the constraints on the clocks by some parameter δ. For instance, "$x \in [a, b]$" is transformed into "$x \in [a - \delta, b + \delta]$". Moreover, a formal link is drawn in [DDR05] between these two semantics: as soon as $\delta > 4\delta_P + 3\delta_L$, the AASAP semantics simulates the semantics of the implementation. As a consequence, implementability can be ensured by establishing the existence of some positive δ for which the AASAP semantics meets the specification.

Robustness problems. We call the above problem (existence of some positive δ) the *qualitative problem of robustness*. This problem was proven decidable for different kind of properties: the problem is PSPACE-complete for safety properties [Pur00, DDMR08] and LTL formula [BMR06]. It is EXPTIME-complete for a fragment of the timed logic MTL [BMR08]. In addition, for safety properties, it is proven in [Pur00, DDMR08] that if there exists a safe positive value of δ, then the system is also safe for a specific value of the form $1/2^{|\mathcal{A}|}$. While this allows to deduce a correct value for the parameter δ, computing the largest value of δ for which the AASAP semantics meets the specification was still an open problem. We are interested here in this last problem, which we call the *quantitative problem of robustness* for safety properties.

Our contributions. In this paper, we prove that the quantitative robustness problem for safety properties is decidable for flat timed automata (*i.e.* where each location belongs to at most one cycle). In addition, we show that the maximal safe value of δ is a rational number. To this end, we solve a more general problem: we prove that it is possible to compute, for a flat timed automaton, its reachability set parameterized by δ. We call it the parametric reachability set. For this computation, we present a forward algorithm based on parametric zones (recall that a zone is a constraint on clocks). As a parametric forward analysis does not terminate for (flat) timed automata, we need some acceleration techniques. To solve the qualitative robustness problem, different algorithms have been proposed in [Pur00, DDMR08, DK06] which compute an enlarged reachability set corresponding to states reachable for any positive perturbation, and include an acceleration of cycles. The algorithm we propose can be understood as a parametric version of the symbolic algorithm proposed in [DK06] for flat timed automata. We then tackle two issues: the termination of our procedure and its correctness. For the first aspect, as we are in a parametric setting, we need completely new arguments of termination (the number of parametric zones we compute cannot be bounded as it is the case for zones). Considering a graph representation of zones introduced in [CJ99a], we obtain proofs of termination depending only on the number of clocks, and not on the constants appearing in the automaton. To our knowledge, this constitutes an original approach in the context of timed automata. Regarding correctness, we identify under which conditions the enlarged reachability set coincides with the standard reachability set. This allows

us to propose an algorithm computing the parametric reachability set of a flat timed automaton.

Related work. Since its definition in [Pur00, DDR05], the approach based on the AASAP semantics has received much attention, and other kind of perturbations, like the drift of clocks, have been studied [DDMR08, ALM05, Dim07]. In the case of safety properties and under some natural assumptions, this perturbation is equivalent to constraint enlargement and relies on similar techniques, as proven in [DDMR08]. Also, several works have considered variants of the robustness problem. In [SF07, SFK08], the case of systems with bounded life-time or regular resynchronization of clocks is considered, while in [Dim07], a symbolic algorithm is proposed to handle strict constraints.

Many other notions of "robustness" have been proposed in the literature in order to relax the mathematical idealization of the semantics of timed automata, see for instance [GHJ97, OW03, BBB$^+$07]. Those approaches are different from ours, since they roughly consist in dropping "isolated" or "unlikely" executions, and are thus more related to language theoretical issues than to implementability issues.

Finally, our work is somewhat related to parametric timed automata. It is proven in [WT99] that emptiness is already undecidable for timed automata with three clocks and one parameter. In our setting, decidability results follow from strong restrictions on the use of the parameter. They correspond to the notion of upper parameter introduced in [HRSV02], but the problems we consider are different. In addition, to obtain termination, we introduce acceleration techniques based on [CJ99a]. Two recent works [BIL06, BIK10] also rely on [CJ99a] to propose acceleration techniques, but these concern flat counter automata with an integer-valued parameter.

Organisation of the paper. In Section 2, we introduce standard definitions. We present in Section 3 the definition of the enlarged reachability set, and a modification of the algorithm of [DK06] for its computation. In Section 4, we first recall the graph representation of constraints, then present how we use it to obtain a new acceleration technique, and finally we present our parametric algorithm and its proof of termination and of correction.

2 Definitions

2.1 Timed Automata, Zones

Let $\mathcal{X} = \{x_1, \ldots, x_n\}$ be a finite set of *clock* variables. We extend it with a fictive clock x_0, whose value will always be 0, and denote by $\overline{\mathcal{X}}$ the set $\mathcal{X} \cup \{x_0\}$. An *atomic (clock) constraint* on $\mathcal{X}$ is of the form $x - y \leq k$, where $x \neq y \in \overline{\mathcal{X}}$ and $k \in \mathbb{Q}$. Note that we only consider non-strict inequalities. This makes sense as we will later enlarge these constraints. We say that the constraint is *non-diagonal* if the comparison involves the clock x_0. We denote by $\mathcal{G}(\mathcal{X})$ (resp. $\mathcal{G}_{nd}(\mathcal{X})$) the set of *(clock) constraints* (resp. *non-diagonal constraints*) defined as conjunctions of atomic constraints (resp. non-diagonal atomic constraints).

A *(clock) valuation* v for $\mathcal{X}$ is an element of $\mathbb{R}^{\mathcal{X}}_{\geq 0}$. A valuation $v \in \mathbb{R}^{\mathcal{X}}_{\geq 0}$ is extended to $\mathbb{R}^{\overline{\mathcal{X}}}_{\geq 0}$ by $v(x_0) = 0$. If $v \in \mathbb{R}^{\mathcal{X}}_{\geq 0}$ and $t \in \mathbb{R}_{\geq 0}$, we write $v + t$ for the valuation assigning $v(x) + t$ to every clock $x \in \mathcal{X}$. If $r \subseteq \mathcal{X}$, $v[r \leftarrow 0]$ denotes the valuation

assigning 0 to every clock in r and $v(x)$ to every clock in $\mathcal{X} \setminus r$. Whether a valuation $v \in \mathbb{R}^{\mathcal{X}}_{\geq 0}$ satisfies a constraint $g \in \mathcal{G}(\mathcal{X})$, written $v \models g$, is defined inductively as follows: the conjunction is handled naturally, and $v \models x - y \leq k$ iff $v(x) - v(y) \leq k$ (recall that $v(x_0) = 0$). The set of valuations satisfying a constraint g is denoted $[\![g]\!]$.

A *zone* Z over $\mathcal{X}$ is a convex subset of $\mathbb{R}^{\mathcal{X}}_{\geq 0}$ which can be defined as the set of valuations satisfying a clock constraint, *i.e.* there exists $g \in \mathcal{G}(\mathcal{X})$ such that $Z = [\![g]\!]$. We note $\mathsf{Zones}(\mathcal{X})$ the set of zones on $\mathcal{X}$. The zone $\mathbb{R}^{\mathcal{X}}_{\geq 0}$ is denoted $\top$.

Definition 1 (Timed Automaton). *A* TA *is a tuple $\mathcal{A} = (L, \ell_0, \mathcal{X}, \Sigma, T)$ where L is a finite set of locations, $\ell_0 \in L$ is an initial location, $\mathcal{X}$ is a finite set of clocks, Σ is a finite set of actions, and $T \subseteq L \times \mathcal{G}_{nd}(\mathcal{X}) \times \Sigma \times 2^{\mathcal{X}} \times L$ is a finite set of transitions.*

We define the semantics of $\mathcal{A}$ as a timed transition system $[\![\mathcal{A}]\!] = \langle S, S_0, \Sigma, \rightarrow \rangle$. The set S of states of $[\![\mathcal{A}]\!]$ is $L \times \mathbb{R}^{\mathcal{X}}_{\geq 0}$ and $S_0 = \{(\ell_0, v_0) \mid v_0(x) = v_0(y),\ \forall x, y \in \mathcal{X}\}$. A transition in $[\![\mathcal{A}]\!]$ is composed either of a delay move $(\ell, v) \xrightarrow{d} (\ell, v + d)$, with $d \in \mathbb{R}_{>0}$, or of a discrete move $(\ell, v) \xrightarrow{\sigma} (\ell', v')$ when there exists a transition $(\ell, g, \sigma, r, \ell') \in T$ with $v \models g$, and $v' = v[r \leftarrow 0]$. The graph $[\![\mathcal{A}]\!]$ is thus an infinite transition system. A *run* of $[\![\mathcal{A}]\!]$ is a finite or infinite sequence $(\ell_0, v_0) \xrightarrow{\sigma_1} (\ell_1, v_1) \xrightarrow{d_1} (\ell_1, v_1 + d_1) \xrightarrow{\sigma_2} (\ell_2, v_2) \ldots$ where for each $i \geq 1$, $d_i \in \mathbb{R}_{\geq 0}$, and $(\ell_0, v_0) \in S_0$. A state (ℓ, v) is *reachable* in $[\![\mathcal{A}]\!]$ iff there exists a run from an initial state $(\ell_0, v_0) \in S_0$ to (ℓ, v); the set of reachable states is denoted $\mathsf{Reach}(\mathcal{A})$.

Note that standard definitions of timed automata also allow invariants on locations which restrict time elapsing. For the sake of simplicity, we do not consider this technical addition here, however all our results hold in presence of invariants.

A *cycle* of $\mathcal{A}$ is a finite sequence of transitions corresponding to a cycle of the underlying finite state automaton. We say that a timed automaton is *flat* if each location belongs to at most one cycle. A *progress cycle* is a cycle where each clock is reset at least once. We say $\mathcal{A}$ is *progressive* if it only contains progress cycles.

Assumptions. As our results rely on previous works on robustness in TA [Pur00, DDMR08], we assume that our TA are progressive, and that all the clocks are always bounded by some constant M. In addition, as the algorithm we propose is based on [DK06], we also require our timed automata to be flat.

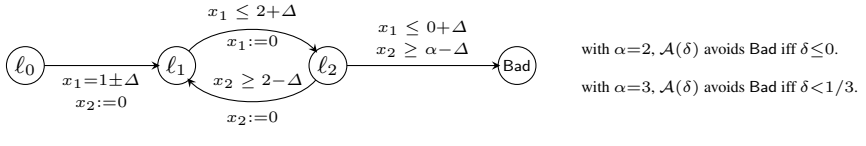


Fig. 1. A timed automaton $\mathcal{A}$, with its parametric semantics

2.2 Parametric Objects

We define the parametric semantics introduced in [Pur00] that enlarges the set of runs of timed automata. This semantics can be defined in terms of timed automata extended with one parameter, denoted Δ, with syntactic constraints on the use of this parameter.

We denote by $\mathcal{PG}(\mathcal{X})$ the set of *parametric (clock) constraints* generated by the grammar [1] $g ::= g \wedge g \mid x - y \leq k + b\Delta$, where $x \neq y \in \overline{\mathcal{X}}$, $k \in \mathbb{Q}$ and $b \in \mathbb{N}$ (Δ represents a delay and b represents the accumulation of this delay, see Example 1). Given a parametric constraint g and $\delta \in \mathbb{Q}_{\geq 0}$, we denote by $g(\delta)$ the constraint obtained by evaluating the parameter Δ in δ. As the parameter helps in "relaxing" the clock constraint, we have that $\delta \leq \delta'$ implies $[\![g(\delta)]\!] \subseteq [\![g(\delta')]\!]$.

Definition 2 (Parametric Zone). *A* parametric zone $\mathcal{Z}$ *over* $\mathcal{X}$ *is a partial mapping from* $\mathbb{Q}_{\geq 0}$ *to zones over* $\mathcal{X}$*, which satisfies the following properties:* (i) *its domain* $\mathsf{dom}(\mathcal{Z})$ *is an interval with rational bounds, and* (ii) *it can be defined as the parametric satisfiability set of a parametric clock constraint,* i.e. *there exists* $g \in \mathcal{PG}(\mathcal{X})$ *such that for all* $\delta \in \mathsf{dom}(\mathcal{Z}), \mathcal{Z}(\delta) = [\![g(\delta)]\!]$*. We denote by* $\mathsf{PZones}(\mathcal{X})$ *the set of parametric zones on* $\mathcal{X}$[2].

By default the considered domain for a parametric zone is $\mathbb{Q}_{\geq 0}$. Given a rational interval I, we denote $\mathcal{Z}_{|I}$ the parametric zone whose domain is restricted to I *i.e.*, $\mathsf{dom}(\mathcal{Z}_{|I}) = \mathsf{dom}(\mathcal{Z}) \cap I$, and which coincides with $\mathcal{Z}$ on $\mathsf{dom}(\mathcal{Z}_{|I})$. Given $\mathcal{Z}, \mathcal{Z}' \in \mathsf{PZones}(\mathcal{X})$, we define $\mathcal{Z} \subseteq \mathcal{Z}'$ if, and only if, we have $\mathsf{dom}(\mathcal{Z}) \subseteq \mathsf{dom}(\mathcal{Z}')$, and for any $\delta \in \mathsf{dom}(\mathcal{Z})$, $\mathcal{Z}(\delta) \subseteq \mathcal{Z}'(\delta)$. We say that a parametric zone $\mathcal{Z}$ is non-empty if there exists $\delta \in \mathsf{dom}(\mathcal{Z})$ such that $\mathcal{Z}(\delta) \neq \varnothing$. Let $\mathcal{Z}$ be a non-empty parametric zone. As the mapping represented by $\mathcal{Z}$ is monotone, we define $\delta_{\neg\varnothing}(\mathcal{Z}) = \inf\{\delta \geq 0 \mid \mathcal{Z}(\delta) \neq \varnothing\}$ the minimal value of the parameter for the zone it denotes to be nonempty. As $\mathcal{Z}$ only involves non-strict linear inequalities, $\delta_{\neg\varnothing}(\mathcal{Z})$ is a rational number and we have $\mathcal{Z}(\delta_{\neg\varnothing}(\mathcal{Z})) \neq \varnothing$ (provided that $\delta_{\neg\varnothing}(\mathcal{Z}) \in \mathsf{dom}(\mathcal{Z})$).

Definition 3 (Parametric Semantics [Pur00, DDMR08]). *Let* $\mathcal{A} = (L, \ell_0, \mathcal{X}, \Sigma, T)$ *be a TA. The* parametric semantics of $\mathcal{A}$ *consists in replacing each constraint* $g \in \mathcal{G}_{nd}(\mathcal{X})$ *appearing in some transition of* $\mathcal{A}$ *by the parametric constraint obtained by enlarging it with the parameter* Δ*. Formally, each atomic constraint of the form* $x - y \leq k$ *is replaced by the parametric constraint* $x - y \leq k + \Delta$*.*

Given $\delta \in \mathbb{Q}_{\geq 0}$, the instantiation of all constraints of $\mathcal{A}$ in δ leads to a timed automaton that we denote by $\mathcal{A}(\delta)$. The semantics used implies the following monotonicity property: $\delta \leq \delta' \Rightarrow \mathsf{Reach}(\mathcal{A}(\delta)) \subseteq \mathsf{Reach}(\mathcal{A}(\delta'))$. An example of timed automaton is shown in Figure 1.

2.3 Symbolic Computations Using (Parametric) Zones

A *symbolic state* is a pair $(\ell, Z) \in L \times \mathsf{Zones}(\mathcal{X})$. Consider a transition $t = (\ell, g, \sigma, r, \ell') \in T$ of a TA $\mathcal{A}$. We define the operator Post^t computing the symbolic successors over t starting from the zone Z, with $Z \in \mathsf{Zones}(\mathcal{X})$, by $\mathsf{Post}^t(Z) = \{v' \in \mathbb{R}^{\mathcal{X}}_{\geq 0} \mid \exists v \in Z, \exists d \in \mathbb{R}_{>0} : v \models g \wedge v' = v[r \leftarrow 0] + d\}$. It is well known that $\mathsf{Post}^t(Z)$ is still a zone. We define similarly the operator Pre^t for the set of predecessors by t. Given a sequence of transitions ϱ, we define the operators Post^ϱ and Pre^ϱ as the compositions of these operators for each transition of ϱ. We define the set of successors from a symbolic state by $\mathsf{Succ}(\ell, Z) = \{(\ell', Z') \in L \times \mathsf{Zones}(\mathcal{X}) \mid \exists t = (\ell, g, \sigma, r, \ell') \in T \text{ s.t. } Z' = \mathsf{Post}^t(Z)\}$.

[1] Compared with L/U TA introduced in [HRSV02], our parameter is "upper".

[2] In the sequel, Z and Y denote a zone, while $\mathcal{Z}$ and $\mathcal{Y}$ denote a parametric zone.

In order to perform parametric computations, we will use parametric zones. Our parametric constraints are less expressive[3] than those considered in [AAB00]. In particular, we can perform the operations of intersection, time elapsing, clock reset, inclusion checking... and extend operators Post^{ϱ} and Pre^{ϱ} to a parametric setting. We denote these extensions by PPost^{ϱ} and PPre^{ϱ}. We also define the operator $\mathsf{Succ}(\ell, \mathcal{Z})$, where $\mathcal{Z} \in \mathsf{PZones}(\mathcal{X})$, using the PPost operator.

3 The Enlarged Reachability Set $\mathsf{Reach}^*(\mathcal{A})$

Definition of $\mathsf{Reach}^*(\mathcal{A})$. We are interested here in the *quantitative problem of robustness* for safety properties: given a set of states Bad to be avoided, compute the maximal value of δ for the system to be safe, *i.e.* the value $\delta_{\max} = \sup\{\delta \geq 0 \mid \mathsf{Reach}(\mathcal{A}(\delta)) \cap \mathsf{Bad} = \varnothing\}$ (recall the monotonicity of $\mathsf{Reach}(\mathcal{A}(\delta))$ *w.r.t.* δ). Note that the value $\delta_{\max}$ may be safe or not (see Examples in [JR10]).

In this paper, we propose an algorithm that computes a representation of the parametric reachability set of a flat timed automaton. It is then easy to derive the optimal value $\delta_{\max}$. A forward parametric computation of reachable states does not terminate in general for timed automata. Indeed, the coefficient on parameter Δ (coefficient b in Definition 2) cannot always be bounded (see Example 1). Such a phenomenon is due to cycles: it can be the case that a state (ℓ, v) is reachable for any $\delta > 0$, but the length of paths allowing to reach (ℓ, v) in $\mathcal{A}(\delta)$ diverges when δ converges to 0.

Example 1. Consider the timed automaton represented on Figure 1. State (ℓ_2, v) with $v(x_1) = 0$ and $v(x_2) = 2$ is reachable in $[\![\mathcal{A}(\delta)]\!]$ for any $\delta > 0$. Let us denote by t_1 (resp. t_2) the transition from ℓ_1 to ℓ_2 (resp. from ℓ_2 to ℓ_1), and let $\varrho = t_1 t_2$. In $[\![\mathcal{A}(\delta)]\!]$, this state is reachable only after $\lceil \frac{1}{2\delta} \rceil$ iterations of the cycle ϱ (see Figure 2).

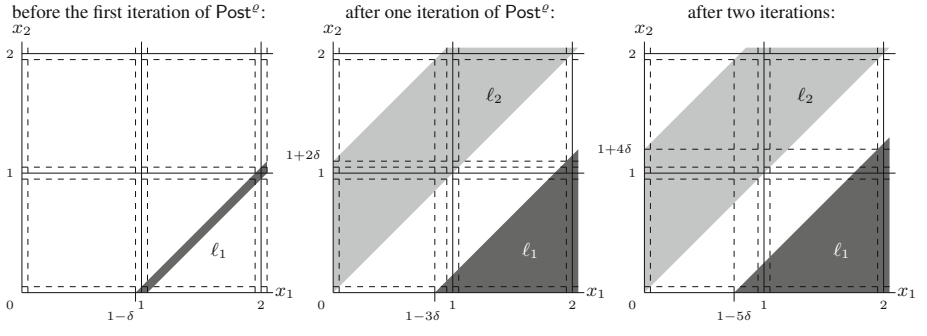


Fig. 2. Reachable states during the parametric forward analysis of $\mathcal{A}(\delta)$

[3] Note that in our setting, one can define a data structure more specific than parametric DBMs considered in [AAB00]. Indeed, we do not need to split DBMs as the constraints only involve conjunctions. Moreover, we can perform basic operations (future, reset, intersection with an atomic constraint) in quadratic time, as for DBMs, see [Jau09].

This difficulty has first been identified by Puri in [Pur00] when studying the qualitative robustness problem, and solved by computing the enlarged reachability set defined as $\mathsf{Reach}^*(\mathcal{A}) \stackrel{def}{=} \bigcap_{\delta \in \mathbb{Q}_{>0}} \mathsf{Reach}(\mathcal{A}(\delta))$. It is the set of states of the automaton reachable by an arbitrarily small value of the parameter. While [Pur00] proposed an algorithm based on the region graph, we use an algorithm proposed in [DK06] which relies on zones, as it is better suited for a parametric setting. The drawback of [DK06] is that it requires the timed automaton to be flat as it enumerates cycles of the automaton.

Algorithm 1. Computation of $\mathsf{Reach}^*(\mathcal{A})$

Require: a progressive flat timed automaton $\mathcal{A}$ with bounded clocks.
Ensure: the set $\mathsf{Reach}^*(\mathcal{A})$.

1: Compute $\nu Y.\mathsf{Pre}^{\varrho}(Y)$, $\nu Y.\mathsf{Post}^{\varrho}(Y)$, for each cycle ϱ in $\mathcal{A}$.
2: Wait $= \{(\ell_0, Z_0)\}$; *// Initial states*
3: Passed $= \varnothing$;
4: **while** Wait $\neq \varnothing$ **do**
5: pop (ℓ, Z) from Wait ;
6: **if** $\forall (\ell, Z') \in$ Passed, $Z \not\subseteq Z'$ **then**
7: **if** there exists a cycle ϱ around location ℓ **then**
8: **if** $Z \cap \nu Y.\mathsf{Pre}^{\varrho}(Y) \neq \varnothing$ **then**
9: Wait $=$ Wait $\cup\, \mathsf{Succ}(\ell, \nu Y.\mathsf{Post}^{\varrho}(Y))$;
10: Passed $=$ Passed $\cup \{(\ell, \nu Y.\mathsf{Post}^{\varrho}(Y))\}$;
11: Wait $=$ Wait $\cup\, \mathsf{Succ}(\ell, Z)$;
12: Passed $=$ Passed $\cup \{(\ell, Z)\}$;
13: **return** Passed ;

A new procedure for the computation of Reach^*. We present Algorithm 1 which is a modification of the algorithm proposed in [DK06] to compute Reach^*. This modification allows us in Section 4 to prove the termination of a parametric version of this algorithm.

The original algorithm proposed in [DK06] relies on the notion of *stable zone* of a cycle ϱ. This zone represents states having infinitely many predecessors and successors by ϱ, and is defined as the intersection of two greatest fixpoints: $W_{\varrho} = \nu Y.\mathsf{Post}^{\varrho}(Y) \cap \nu Y.\mathsf{Pre}^{\varrho}(Y)$. Then, the algorithm is obtained by the following modifications of the standard forward analysis of the timed automaton: for each new symbolic state (ℓ, Z) considered, if there exists a cycle ϱ around location ℓ, and if Z intersects the stable zone W_{ϱ}, then the stable zone is marked as reachable. The correction of this algorithm relies on the following property of the stable zone: given two valuations $v, v' \in W_{\varrho}$, for any $\delta > 0$, there exists a path in $[\![\mathcal{A}(\delta)]\!]$ from state (ℓ, v) to state (ℓ, v') (while such a path may not exist in $[\![\mathcal{A}]\!]$). The addition of the stable zone can be viewed as the acceleration of cycle ϱ.

Our new algorithm is obtained as follows: (i) at line 8, we test the intersection of Z with $\nu Y.\mathsf{Pre}^{\varrho}(Y)$ instead of W_{ϱ}, and (ii) at line 9 and 10, instead of declaring W_{ϱ} as reachable, we declare $\nu Y.\mathsf{Post}^{\varrho}(Y)$ reachable. We state below that this modification is correct.

Theorem 1. *Algorithm 1 is sound and complete.*

Proof. We show that Algorithm 1 is equivalent to that of [DK06]. As W_ϱ is included in both greatest fixpoints, the completeness of the algorithm is trivial. To prove the soundness, let us consider the region graph construction (see for instance [AD94]). We do not recall this standard construction as it will only be used in this proof. As there are finitely many regions, it is easy to verify that if a region is included in $\nu Y.\mathsf{Pre}^\varrho(Y)$, it has infinitely many successors by ϱ and then one of them is included in W_ϱ. In other terms, the test of line 8 of intersection with $\nu Y.\mathsf{Pre}^\varrho(Y)$ instead of W_ϱ simply anticipates the acceleration of the cycle ϱ. Similarly, any region included in $\nu Y.\mathsf{Post}^\varrho(Y)$ is the successor of a region included in W_ϱ. Thus, our modification can be understood as a speed-up of the original algorithm of [DK06]. □

We also state the following Lemma whose proof follows from a similar reasoning:

Lemma 1. *Let ϱ be a cycle of a TA $\mathcal{A}$. Then we have:*
$\nu Y.\mathsf{Pre}^\varrho(Y) \neq \varnothing \Leftrightarrow \nu Y.\mathsf{Pre}^\varrho(Y) \cap \nu Y.\mathsf{Post}^\varrho(Y) \neq \varnothing \Leftrightarrow \nu Y.\mathsf{Post}^\varrho(Y) \neq \varnothing$

4 Parametric Computation of Reach($\mathcal{A}(\delta)$)

4.1 Representing Constraints as a Graph

In the sequel, we will use a representation of clock constraints as a weighted directed graph introduced in [CJ99a, CJ99b]. Due to lack of space, we recall here only succinctly its definition. Intuitively, the value of a clock can be recovered from its date of reset and the current time. The vertices of the graph represent these values, with one duplicate for each fired transition. Constraints on clock values are expressed as weights on arcs.

More formally, recall that $n = |\mathcal{X}|$, $\mathcal{X} = \{x_1, \ldots, x_n\}$, x_0 is a fictive clock whose value is always zero, and $\overline{\mathcal{X}} = \mathcal{X} \cup \{x_0\}$. We introduce a new clock τ which is never reset and thus represents the total elapsed time. In addition, for each clock $x_i \in \overline{\mathcal{X}}$ we let variable X_i denote $X_i = \tau - x_i$. Note that for $x_i \in \mathcal{X}$, X_i thus represents last date of reset of clock x_i. For the special case of x_0, we have $X_0 = \tau$ (as x_0 always has value 0). We denote $\overrightarrow{V}$ the vector defined as $(\tau, X_1, \ldots, X_n)$ which is a vector of (symbolic) variables. For a transition $t = (\ell, g, \sigma, r, \ell')$, we define the formula $T^t(\overrightarrow{V}, \overrightarrow{V'})$ which expresses the relationship between values of the variables before (represented by $\overrightarrow{V}$) and after the firing of the transition (represented by $\overrightarrow{V'} = (\tau', X'_1, \ldots, X'_n)$):

$$T^t(\overrightarrow{V}, \overrightarrow{V'}) := \bigwedge_{i=1}^{n} \left(X_i \leq \tau \wedge X'_i \leq \tau' \wedge \tau \leq \tau' \wedge \bigwedge_{x_i \in r} \tau = X'_i \wedge \bigwedge_{x_i \notin r} X_i = X'_i \wedge \overline{g} \right)$$

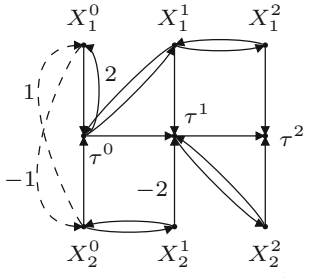
where $\overline{g}$ is the constraint g where for any i, clock x_i is replaced by $\tau - X_i$.

Let $\varrho = t_1 \ldots t_m$ be a sequence of transitions. For $j \in \{0, \ldots, m\}$, we denote by $\overrightarrow{V^j}$ the vector $(\tau^j, X_1^j, \ldots, X_n^j)$. Then we define formula $T^\varrho(\overrightarrow{V^0}, \overrightarrow{V^m})$ expressing the constraints between variables before and after the firing of the sequence ϱ as follows:

$$T^\varrho(\overrightarrow{V^0}, \overrightarrow{V^m}) := \exists \overrightarrow{V^1}, \ldots, \overrightarrow{V^{m-1}}. \bigwedge_{j=0}^{m-1} T^{t_{j+1}}(\overrightarrow{V^j}, \overrightarrow{V^{j+1}})$$

We associate with formula $T^{\varrho}(\overrightarrow{V^0}, \overrightarrow{V^m})$ a weighted directed graph whose vertices are variables used to define the formula, and arcs represent constraints of the formula:

Definition 4 (Graph $G_{\varrho}^{\top}$). *Let $\varrho = t_1 \dots t_m$ be a sequence of transitions. The weighted directed graph $G_{\varrho}^{\top}$ has a set of vertices $\mathcal{S} = \bigcup_{j=0}^{m} V^j$, where $V^j = \{\tau^j, X_1^j, \dots, X_n^j\}$. Given two vertices $X, X' \in \mathcal{S}$ and a weight $c \in \mathbb{Q}$, there is an arc from X to X' labelled by c if and only if the formula $T^{\varrho}(\overrightarrow{V^0}, \overrightarrow{V^m})$ contains the constraint $X - X' \leq c$.*



Example 2. Consider the sequence of transitions $\varrho = t_1 t_2$ in the TA of Figure 1 defined in Example 1. The graph depicted on the left-side figure with plain arcs represents $G_{\varrho}^{\top}$ (arcs without label have weight 0). For instance, the arc from vertex X_2^1 to vertex τ^1, labelled by -2, represents the lower bound for the clock x_2 in t_2 which means: $x_2 \geq 2$.

For any path p, we write $w(p)$ the total weight of the path. Suppose now that there is no cycle of negative weight in graph $G_{\varrho}^{\top}$. Let P_{beg}^{ϱ} (resp. P_{end}^{ϱ}) denote the set of minimal weighted paths from a vertex in V^0 (resp. in $V^{|\varrho|}$) to another vertex in V^0 (resp. in $V^{|\varrho|}$). We define the following mapping which interprets these shortest paths as clock constraints:

$$C(p) = x_l - x_i \leq w(p) \text{ if } \begin{cases} p \in P_{beg}^{\varrho} \text{ starts in } X_i^0 \text{ and ends in } X_l^0 \\ p \in P_{end}^{\varrho} \text{ starts in } X_i^{|\varrho|} \text{ and ends in } X_l^{|\varrho|} \end{cases}$$

From Propositions 12 and 13 of [CJ99b], we have the following properties:

Proposition 1. *Let ϱ be a sequence of transitions. Then we have:*

- *there exists a cycle γ with $w(\gamma) < 0$ in $G_{\varrho}^{\top}$ $\Leftrightarrow$ $\mathsf{Post}^{\varrho}(\top) = \varnothing \Leftrightarrow \mathsf{Pre}^{\varrho}(\top) = \varnothing$*
- *if there is no cycle of negative weight, then:*
  $[\![\bigwedge_{p \in P_{end}^{\varrho}} C(p)]\!] = \mathsf{Post}^{\varrho}(\top)$ *and* $[\![\bigwedge_{p \in P_{beg}^{\varrho}} C(p)]\!] = \mathsf{Pre}^{\varrho}(\top)$

More generally, given a zone Z, we define the graph denoted G_{ϱ}^{Z} by adding the constraints of Z on the vertices in V^0. Mapping C applied on paths in P_{end}^{ϱ} then defines the zone $\mathsf{Post}^{\varrho}(Z)$. Similarly, the zone $\mathsf{Pre}^{\varrho}(Z)$ can be represented by adding constraints of Z on vertices in $V^{|\varrho|}$.

Example 3 (Example 2 continued.). Consider now the zone $Z = [\![x_1 - x_2 = 1]\!]$ (it corresponds to the set of reachable valuations after firing transition $\ell_0 \to \ell_1$ in TA of Figure 1), then additional dotted arcs allow to represent G_{ϱ}^{Z}.

It is easy to verify that this construction extends to a parametric setting: considering parametric constraints on arcs, we obtain a graph representation of the parametric computation of symbolic successors or predecessors. Note that a path p in this context will have a weight of the form $k + b\Delta$, where $b \in \mathbb{N}$ represents the number of atomic constraints of the TA used in p. In particular, while the value of a path depends on the value of Δ, its existence does not.

Given a zone defined as the result of the firing of a sequence of transitions, this representation allows to recover how the constraints are obtained. Thus, the graph stores the complete history of the constraints.

In the sequel, we use this construction in the particular case of the iteration of a cycle ϱ, given as a sequence of transitions of a TA. Let Z_{init} be a zone. We consider two sequences of zones $(Z_k^\top)_{k\geq 0}$ (resp. $(Z_k^{init})_{k\geq 0}$) defined by $Z_0^\top = \top$ (resp. $Z_0^{init} = Z_{init}$) and $Z_{k+1}^* = \mathsf{Post}^\varrho(Z_k^*)$ (where * denotes either $^\top$ or ^{init}). Note that by monotonicity of Post^ϱ, the sequence $(Z_k^\top)_{k\geq 0}$ is decreasing and converges towards $Z_\infty^\top = \nu Y.\mathsf{Post}^\varrho(Y)$. According to Proposition 1, note that constraints defining zone $Z_k^\top$ (resp. Z_k^{init}) can be obtained from shortest paths in graph $G_{\varrho^k}^\top$ (resp. $G_{\varrho^k}^{init}$). As the cycle ϱ will be clear from the context, we will omit to mention it in the subscript, and use notations $G_k^\top$ and G_k^{init} respectively.

Moreover, we will only be interested in vertices at the frontier between the different copies of the graph of ϱ. Then, given a clock $x_i \in \overline{\mathcal{X}}$ and an index $j \leq k$, *vertex X_i^j now denotes the date of reset of clock x_i after the j-th execution of ϱ* (this notation is a shorthand for the notation $X_i^{j\times|\varrho|}$, as this last notation will never be used anymore).

Definition 5. *Let $N = |\overline{\mathcal{X}}|^2$. A* return path *is a pair $r = (p_1, p_2)$ of paths in the graph $G_N^\top$ such that there exist two clocks $x_u, x_v \in \overline{\mathcal{X}}$ and two indices $0 \leq i < j \leq N$ verifying:*

- *p_1 and p_2 are included in the subgraph associated with i-th to j-th copies of ϱ*
- *p_1 is a shortest path from vertex X_u^j to vertex X_u^i*
- *p_2 is a shortest path from vertex X_v^i to vertex X_v^j*

The weight of *r is defined as $w(r) = w(p_1) + w(p_2)$. The set of return paths is finite and is denoted $\mathcal{R}$.*

4.2 Accelerating Computations of Greatest Fixpoints

Let ϱ be a cycle. In this subsection, we only consider the operator Post^ϱ, but all our results also apply to the operator Pre^ϱ. We consider the decreasing sequence $(Z_k^\top)_{k\geq 0}$ converging towards $Z_\infty^\top = \nu Y.\mathsf{Post}^\varrho(Y) = \bigcap_{k\geq 0} Z_k^\top$. We prove the following lemma which provides a bound for termination only dependant on the number of clocks. Note that this result does not require the cycle ϱ to be progressive neither the clocks to be bounded.

Lemma 2. *Let $N = |\overline{\mathcal{X}}|^2$, and $k \geq N$. If $Z_{k+1}^\top \subsetneq Z_k^\top$, then we have $Z_\infty^\top = \varnothing$.*

Proof. First, we prove that $Z_{k+1}^\top \subsetneq Z_k^\top$ implies that there exists $r \in \mathcal{R}$ used in some shortest path of $Z_{k+1}^\top$ witness of the disequality. Indeed, as $Z_{k+1}^\top \subsetneq Z_k^\top$, there exists a bound $b = "x_p - x_q \leq \cdot"$ with $0 \leq p \neq q \leq n$, whose constraint is strictly smaller in $Z_{k+1}^\top$ than in $Z_k^\top$. In $Z_{k+1}^\top$, the constraint on b is obtained as a shortest path between vertices X_p^{k+1} and X_q^{k+1} in the graph $G_{k+1}^\top$. Let c be such a path. By definition of $G_k^\top$ and $G_{k+1}^\top$, the path c must use arcs in $G_1^\top$ (otherwise c would also exist in $G_k^\top$). The graph $G_{k+1}^\top$ is the concatenation of $k+1$ copies of the graph of ϱ. For each occurrence of ϱ, c goes through a pair of vertices when it enters/leaves it. Finally, as $k+1 > N = |\overline{\mathcal{X}}|^2$, there exists a pair that occurs twice, we denote these

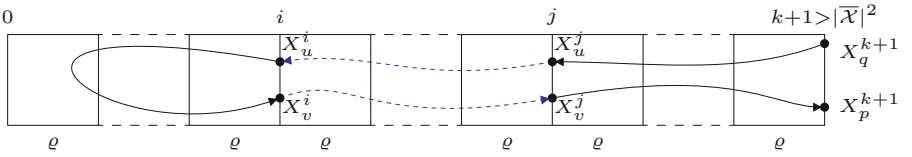


Fig. 3. Pumping lemma : a path from X_q^{k+1} to X_p^{k+1} using arcs in $G_1^\top$ exhibits a return path between pairs of vertices (X_u^i, X_v^i) and (X_u^j, X_v^j)

two clocks x_u and x_v. Thus c contains a return path $r \in \mathcal{R}$ (see Figure 3 representing the graph $G_{k+1}^\top$ and the return path r in the shortest path c).

Second, as $Z_{k+1}^\top \subsetneq Z_k^\top$, we have $w(r) < 0$. By contradiction, if $w(r) > 0$ then c would not be a shortest path and if $w(r) = 0$ then c would also exist in $G_k^\top$.

Finally, the existence of a return path $r \in \mathcal{R}$ such that $w(r) < 0$ implies that $Z_\infty^\top = \varnothing$ $(= \nu Y.\mathsf{Post}^\varrho(Y))$. When k grows, one can build new paths by repeating this return path. As its weight is negative, the weights of the paths we build diverge towards $-\infty$. In particular, the constraint of the zone $Z_\infty^\top$ on the clock difference $x_p - x_q$ cannot be finite (as it is the limit of a sequence diverging towards $-\infty$), and thus we obtain $Z_\infty^\top = \varnothing$. □

We can now compute, in the parametric setting, the greatest fixpoint of PPost^ϱ for every cycle ϱ of the automaton. We first evaluate the parametric zones $\mathcal{Z} = \mathsf{PPost}^{\varrho^N}(\top)$ and $\mathcal{Z}' = \mathsf{PPost}^\varrho(\mathcal{Z})$. Then, we determine the minimal value $\delta_0 = \min\{\delta \geq 0 \mid \mathcal{Z}(\delta) = \mathcal{Z}'(\delta)\}$. This definition is correct as $\mathcal{Z}' \subseteq \mathcal{Z}$ and, for large enough values of δ, all parametric constraints are equivalent to $\top$. Thus, we have $\nu\mathcal{Y}.\mathsf{PPost}^\varrho(\mathcal{Y})(\delta) \neq \varnothing$ which implies by Lemma 2 that $\mathcal{Z}(\delta) = \mathcal{Z}'(\delta)$. Finally the greatest fixpoint can be represented by $\mathcal{Z}_{|[\delta_0;+\infty[}$ as Lemma 2 ensures that the fixpoint is empty for all $\delta < \delta_0$.

4.3 Parametric Forward Analysis with Acceleration

We present Algorithm 2 for the parametric computation of $\mathsf{Reach}(\mathcal{A}(\delta))$. It can be understood as an adaptation in a parametric setting of Algorithm 1. First, at line 1 we perform parametric computation of greatest fixpoints using the procedure proposed in Section 4.2. Second, the test of intersection between the current zone and the greatest fixpoint of Pre^ϱ is realized in a parametric setting by the computation at line 8 of $\delta_{\min} = \delta_{\neg\varnothing}(\mathcal{Z} \cap \nu\mathcal{Y}.\mathsf{PPre}^\varrho(\mathcal{Y}))$. Finally, we split the domain of the current parametric zone into intervals I_1 and I_2. In interval I_1, no acceleration is done for cycles and thus the set $\mathsf{Reach}(\mathcal{A}(\delta))$ is computed. Acceleration techniques are used only for interval I_2, and for these values the algorithm computes the set $\mathsf{Reach}^*(\mathcal{A}(\delta))$. We prove below that in this case, the equality $\mathsf{Reach}(\mathcal{A}(\delta)) = \mathsf{Reach}^*(\mathcal{A}(\delta))$ holds. Note that the test at line 9 allows to handle differently the particular case of value $\delta_{\min}$ which does not always require to apply acceleration.

Theorem 2. *Algorithm 2 terminates and is correct.*

In the sequel, we denote $N = |\overline{\mathcal{X}}|^2$ and $\delta_{\neg\varnothing}^\varrho = \delta_{\neg\varnothing}(\nu\mathcal{Y}.\mathsf{PPre}^\varrho(\mathcal{Y})) = \delta_{\neg\varnothing}(\nu\mathcal{Y}.\mathsf{PPost}^\varrho(\mathcal{Y}))$ (by Lemma 1). Before turning to the proof, we state the following

Algorithm 2. Parametric Computation of the Reachability Set

Require: a progressive flat timed automaton $\mathcal{A}$ with bounded clocks.
Ensure: the set $\mathsf{Reach}(\mathcal{A}(\delta))$ for all $\delta \in \mathbb{R}_{\geq 0}$.

Compute $\nu\mathcal{Y}.\mathsf{PPre}^{\varrho}(\mathcal{Y})$ and $\nu\mathcal{Y}.\mathsf{PPost}^{\varrho}(\mathcal{Y})$ for each cycle ϱ of $\mathcal{A}$.
Wait $= \{(\ell_0, \mathcal{Z}_0)\}$; *// Initial States*
Passed $= \varnothing$;
while Wait $\neq \varnothing$ **do**
  pop $(\ell, \mathcal{Z})$ from Wait ;
  if $\forall (\ell, \mathcal{Z}') \in$ Passed, $\mathcal{Z} \not\subseteq \mathcal{Z}'$ **then**
    if there exists a cycle ϱ around location ℓ **then**
      $\delta_{\min} = \delta_{\neg\varnothing}(\mathcal{Z} \cap \nu\mathcal{Y}.\mathsf{PPre}^{\varrho}(\mathcal{Y}))$;
      if $\delta_{\min} = \delta_{\neg\varnothing}(\nu\mathcal{Y}.\mathsf{PPre}^{\varrho}(\mathcal{Y}))$ **then**
        $I_1 = [0; \delta_{\min}]$; $I_2 =]\delta_{\min}; +\infty[$;
      else
        $I_1 = [0; \delta_{\min}[$; $I_2 = [\delta_{\min}; +\infty[$;
      Wait $=$ Wait $\cup\, \mathsf{Succ}(\ell, \mathcal{Z}_{|I_1}) \cup \mathsf{Succ}(\ell, \mathcal{Z}_{|I_2}) \cup \mathsf{Succ}(\ell, \nu\mathcal{Y}.\mathsf{PPost}^{\varrho}(\mathcal{Y})_{|I_2})$;
      Passed $=$ Passed $\cup\, (\ell, \mathcal{Z}_{|I_1}) \cup (\ell, \mathcal{Z}_{|I_2}) \cup (\ell, \nu\mathcal{Y}.\mathsf{PPost}^{\varrho}(\mathcal{Y})_{|I_2})$;
    else
      Wait $=$ Wait $\cup\, \mathsf{Succ}(\ell, \mathcal{Z})$;
      Passed $=$ Passed $\cup\, (\ell, \mathcal{Z})$;
return Passed ;

Lemma whose proof is given in [JR10]. Intuitively, it establishes that when all return paths have a positive weight, then either (i) the starting zone has finitely many successors and then it converges to the empty set after at most N steps, or (ii) it has infinitely many successors and then it converges towards $\nu Y.\mathsf{Post}^{\varrho}(Y)$. In this last case, the enlarged reachability set corresponds to the standard reachability set. Its proof relies on pumping techniques presented in Section 4.2. To illustrate property (ii), let consider the timed automaton of Figure 1, for which the enlarged reachability set strictly contains the standard reachability set. One can verify that there exists a return path associated with $\varrho = t_1 t_2$ which has weight 0.

Lemma 3. *Let ϱ be such that for any return path $r \in \mathcal{R}$, we have $w(r) > 0$. Then we have:*

(i) *If $Z_{init} \cap \nu Y.\mathsf{Pre}^{\varrho}(Y) = \varnothing$, then $Z_N^{init} = \varnothing$.*
(ii) *If $Z_{init} \cap \nu Y.\mathsf{Pre}^{\varrho}(Y) \neq \varnothing$, then $Z_\infty^{init} = Z_\infty^{\top} (= \nu Y.\mathsf{Post}^{\varrho}(Y))$.*

Unlike Lemma 2, we use the progress cycle assumption to prove this lemma.

Recall that the TA we consider are flat. As a consequence, in the following proofs of termination and correctness, we will only consider a simple cycle ϱ.

Termination. Consider a parametric symbolic state $(\ell, \mathcal{Z})$ and a cycle ϱ starting in ℓ. We have to prove that all the elements added to the Wait list have a finite number of successors. This is trivial for the successors of $(\ell, \nu\mathcal{Y}.\mathsf{PPost}^{\varrho}(\mathcal{Y})_{|I_2})$ as $\nu\mathcal{Y}.\mathsf{PPost}^{\varrho}(\mathcal{Y})_{|I_2}$ is by definition a fixpoint of PPost^{ϱ}. We now focus on the successors of $(\ell, \mathcal{Z}_{|I_1})$ and $(\ell, \mathcal{Z}_{|I_2})$. Note that we have $\delta_{\min} \geq \delta^{\varrho}_{\neg\varnothing}$.

- **Case of** $(\ell, \mathcal{Z}_{|I_2})$**:** We prove property $(*)$ $\mathsf{PPost}^{\varrho^N}(\mathcal{Z}_{|I_2}) \subseteq \nu\mathcal{Y}.\mathsf{PPost}^{\varrho}(\mathcal{Y})_{|I_2}$. Then the computation is stopped by the test of line 6 as the greatest fixpoint has been added to the Passed list. To prove $(*)$, we prove it holds for any $\delta \in I_2$. Fix some $\delta \in I_2$ and define $Z_{init} = \mathcal{Z}_{|I_2}(\delta)$. We consider the two sequences $(Z_i^*)_{i\geq 0}$ *w.r.t.* cycle ϱ enlarged by δ. Note that as $\delta \geq \delta_{\min} \geq \delta^{\varrho}_{\neg\varnothing}$, we have $\nu\mathcal{Y}.\mathsf{PPost}^{\varrho}(\mathcal{Y})(\delta) \neq \varnothing$. By Lemma 2, this entails $Z_N^{\top} = \nu\mathcal{Y}.\mathsf{PPost}^{\varrho}(\mathcal{Y})(\delta)$. By monotonicity of Post^{ϱ}, $Z_N^{init} \subseteq Z_N^{\top}$ holds. This yields the result.
- **Case of** $(\ell, \mathcal{Z}_{|I_1})$**:** We distinguish two cases whether $\delta_{\min} > \delta^{\varrho}_{\neg\varnothing}$ or not.
  If $\delta_{\min} > \delta^{\varrho}_{\neg\varnothing}$**:** for any $\delta \in [\delta^{\varrho}_{\neg\varnothing}, \delta_{\min}[$, Lemma 3.$(i)$ can be applied on cycle ϱ enlarged by δ. This implies that for any $\delta \in [\delta^{\varrho}_{\neg\varnothing}, \delta_{\min}[$, we have $\mathsf{PPost}^{\varrho^N}(\mathcal{Z}_{|I_1})(\delta) = \varnothing$. Then this property also holds for any $\delta \in I_1$, by monotonicity of $\mathcal{Z}$ and PPost^{ϱ}.
  If $\delta_{\min} = \delta^{\varrho}_{\neg\varnothing}$**:** the complete proof of this last case is more technical and is completely described in [JR10]. We only present here a sketch of proof. First note that for any fixed value of $\delta < \delta_{\min}$, as the zone does not intersect the greatest fixpoint of Pre^{ϱ}, the zone has finitely many successors. However, this argument cannot be lifted to a parametric setting as this number diverges when δ converges towards $\delta_{\min}$. By definition of $\delta^{\varrho}_{\neg\varnothing}$, some return paths, which we call *optimal*, have a weight equal to 0 in $\delta^{\varrho}_{\neg\varnothing}$ (and are thus strictly negative on $[0, \delta^{\varrho}_{\neg\varnothing}[$). Our proof consists in first showing that there exists some integer k for which after k steps, all shortest paths go through optimal return paths. Then, considering q as the least common multiple of lengths of optimal return paths, we can prove the following inclusion $\mathsf{PPost}^{\varrho^{k+q}}(\mathcal{Z}_{|I_1}) \subseteq \mathsf{PPost}^{\varrho^k}(\mathcal{Z}_{|I_1})$. The algorithm stops by test of line 6.

Correctness. As explained before, the algorithm is a standard forward analysis which may add some additional behaviours, according to test of line 8. We distinguish three cases:

1. **For** $\delta \in [0, \delta_{\min}[$ **:** For these values, the algorithm simply performs a forward analysis. As a consequence, the correctness is trivial.
2. **For** $\delta \in]\delta_{\min}, +\infty[$**:** For all these values, the addition occurs, and then the algorithm is equivalent to Algorithm 1. By correction of Algorithm 1, this implies that it computes the set $\mathsf{Reach}^*(\mathcal{A}(\delta))$. We will prove that for all these values, we have the equality $\mathsf{Reach}(\mathcal{A}(\delta)) = \mathsf{Reach}^*(\mathcal{A}(\delta))$. Therefore we need to prove that what has been added to obtain $\mathsf{Reach}^*(\mathcal{A}(\delta))$ was already in $\mathsf{Reach}(\mathcal{A}(\delta))$. Note that the only addition is the greatest fixpoint of Post^{ϱ}. The property is then a direct consequence of Lemma 3.(ii) as it states that the greatest fixpoint is reachable from the initial states. It is easy to verify that Lemma 3.(ii) can indeed be applied.
3. **For** $\delta = \delta_{\min}$**:** There are two cases, whether $\delta_{\min} = \delta^{\varrho}_{\neg\varnothing}$ or not. If the equality holds, then $\delta_{\min} \in I_1$ and the reasoning developed at point 1. also applies. If $\delta_{\min} > \delta^{\varrho}_{\neg\varnothing}$ holds, then $\delta_{\min} \in I_2$ and we can apply reasoning of point 2. as Lemma 3.(ii) also applies because we have $\delta_{\min} > \delta^{\varrho}_{\neg\varnothing}$.

4.4 Quantitative Safety

Once the reachable state space of the automaton is computed by Algorithm 2, it is easy to compute the maximal value of the parameter such that the system avoids some set of bad states. Simply compute the value $\delta_{\neg\varnothing}$ on each parametric zone associated with a bad location and keep the lower one: $\delta_{\max} = \min\{\delta_{\neg\varnothing}(\mathcal{Z}) \mid \exists \ell \in Bad \text{ such that } (\ell, \mathcal{Z}) \in \text{Passed}\}$. We thus obtain:

Theorem 3. *The quantitative robustness problem for safety properties is decidable for flat progressive timed automata with bounded clocks. In addition, the value $\delta_{\max}$ is a rational number.*

5 Conclusion

In this paper, we considered the quantitative robustness problem for safety properties, which aims at computing the largest value of the parameter Δ under which the TA is safe. We proposed a symbolic forward algorithm for the computation of the parametric reachability set for flat timed automata. We proved its termination by means of original arguments using a representation of zones by graphs. As a consequence, it allows us to compute the largest safe value of the parameter, and prove it is a rational number.

Among perspectives, we are first implementing the algorithm using a data structure specific to the parametric zones used in our setting. Second, we want to study the complexity of our algorithm. The difficulty is due to the argument of termination in the last case which leads to a large value and may be improved.

We also aim at enlarging the class of TA for which we can solve the quantitative robustness problem. For instance, if the parameter is not always introduced on guards with coefficient 1, but with other coefficients in $\mathbb{N}_{>0}$, we believe that our algorithm can also be applied. A challenging topic concerns the hypothesis of flatness: first, [CJ99a] proves that timed automata can be flattened, and we want to study whether their result can be combined with ours. Second, we plan to investigate a parametric extension of the algorithm introduced in [Dim07] which can be seen as an extension of that of [DK06] to non-flat TA.

Finally, we believe that it should be possible to solve the quantitative robustness problem for flat TA for other specifications like for instance LTL properties.

References

[AAB00] Annichini, A., Asarin, E., Bouajjani, A.: Symbolic techniques for parametric reasoning about counter and clock systems. In: Emerson, E.A., Sistla, A.P. (eds.) CAV 2000. LNCS, vol. 1855, pp. 419–434. Springer, Heidelberg (2000)

[AD94] Alur, R., Dill, D.: A theory of timed automata. Theoretical Computer Science 126(2), 183–235 (1994)

[ALM05] Alur, R., La Torre, S., Madhusudan, P.: Perturbed timed automata. In: Morari, M., Thiele, L. (eds.) HSCC 2005. LNCS, vol. 3414, pp. 70–85. Springer, Heidelberg (2005)

[BBB+07] Baier, C., Bertrand, N., Bouyer, P., Brihaye, T., Größer, M.: Probabilistic and topological semantics for timed automata. In: Arvind, V., Prasad, S. (eds.) FSTTCS 2007. LNCS, vol. 4855, pp. 179–191. Springer, Heidelberg (2007)

[BDL04] Behrmann, G., David, A., Larsen, K.G.: A tutorial on UPPAAL. In: Bernardo, M., Corradini, F. (eds.) SFM-RT 2004. LNCS, vol. 3185, pp. 200–236. Springer, Heidelberg (2004)

[BDM+98] Bozga, M., Daws, C., Maler, O., Olivero, A., Tripakis, S., Yovine, S.: Kronos: a model-checking tool for real-time systems. In: Y. Vardi, M. (ed.) CAV 1998. LNCS, vol. 1427, pp. 546–550. Springer, Heidelberg (1998)

[BIK10] Bozga, M., Iosif, R., Konečný, F.: Fast acceleration of ultimately periodic relations. In: Touili, T., Cook, B., Jackson, P. (eds.) CAV 2010. LNCS, vol. 6174, pp. 227–242. Springer, Heidelberg (2010)

[BIL06] Bozga, M., Iosif, R., Lakhnech, Y.: Flat parametric counter automata. In: Bugliesi, M., Preneel, B., Sassone, V., Wegener, I. (eds.) ICALP 2006. LNCS, vol. 4052, pp. 577–588. Springer, Heidelberg (2006)

[BMR06] Bouyer, P., Markey, N., Reynier, P.-A.: Robust model-checking of linear-time properties in timed automata. In: Correa, J.R., Hevia, A., Kiwi, M. (eds.) LATIN 2006. LNCS, vol. 3887, pp. 238–249. Springer, Heidelberg (2006)

[BMR08] Bouyer, P., Markey, N., Reynier, P.-A.: Robust analysis of timed automata *via* channel machines. In: Amadio, R.M. (ed.) FOSSACS 2008. LNCS, vol. 4962, pp. 157–171. Springer, Heidelberg (2008)

[CHR02] Cassez, F., Henzinger, T.A., Raskin, J.-F.: A comparison of control problems for timed and hybrid systems. In: Tomlin, C.J., Greenstreet, M.R. (eds.) HSCC 2002. LNCS, vol. 2289, pp. 134–148. Springer, Heidelberg (2002)

[CJ99a] Comon, H., Jurski, Y.: Timed automata and the theory of real numbers. In: Baeten, J.C.M., Mauw, S. (eds.) CONCUR 1999. LNCS, vol. 1664, pp. 242–257. Springer, Heidelberg (1999)

[CJ99b] Comon, H., Jurski, Y.: Timed automata and the theory of real numbers. Research Report LSV-99-6, Laboratoire Spécification et Vérification, ENS Cachan, France, 44 pages (July 1999)

[DDMR08] De Wulf, M., Doyen, L., Markey, N., Raskin, J.-F.: Robust safety of timed automata. Formal Methods in System Design 33(1-3), 45–84 (2008)

[DDR05] De Wulf, M., Doyen, L., Raskin, J.-F.: Almost ASAP semantics: from timed models to timed implementations. Formal Aspects of Computing 17(3), 319–341 (2005)

[Dim07] Dima, C.: Dynamical properties of timed automata revisited. In: Raskin, J.-F., Thiagarajan, P.S. (eds.) FORMATS 2007. LNCS, vol. 4763, pp. 130–146. Springer, Heidelberg (2007)

[DK06] Daws, C., Kordy, P.: Symbolic robustness analysis of timed automata. In: Asarin, E., Bouyer, P. (eds.) FORMATS 2006. LNCS, vol. 4202, pp. 143–155. Springer, Heidelberg (2006)

[GHJ97] Gupta, V., Henzinger, T.A., Jagadeesan, R.: Robust timed automata. In: Maler, O. (ed.) HART 1997. LNCS, vol. 1201, pp. 331–345. Springer, Heidelberg (1997)

[HRSV02] Hune, T., Romijn, J., Stoelinga, M., Vaandrager, F.: Linear parametric model checking of timed automata. Journal of Logic and Algebraic Programming (2002)

[Jau09] Jaubert, R.: Aspects quantitatifs dans la réalisation de contrôleurs temps-réels robustes. Mémoire de Master Recherche, Master Informatique Fondamentale, Marseille (2009)

[JR10] Jaubert, R., Reynier, P.-A.: Quantitative robustness analysis of flat timed automata. Research Report 00534896, HAL (2010)

[OW03] Ouaknine, J., Worrell, J.B.: Revisiting digitization, robustness and decidability for timed automata. In: Proc. LICS 2003. IEEE Computer Society Press, Los Alamitos (2003)

[Pur00] Puri, A.: Dynamical properties of timed automata. Discrete Event Dynamic Systems 10(1-2), 87–113 (2000)

[SF07] Swaminathan, M., Fränzle, M.: A symbolic decision procedure for robust safety of timed systems. In: Proc. TIME 2007, p. 192. IEEE Computer Society Press, Los Alamitos (2007)

[SFK08] Swaminathan, M., Fränzle, M., Katoen, J.-P.: The surprising robustness of (closed) timed automata against clock-drift. In: Ausiello, G., Karhumäki, J., Mauri, G., Ong, L. (eds.) Proc. TCS 2008. IFIP, vol. 273, pp. 537–553. Springer, Heidelberg (2008)

[WT99] Wong-Toi, H.: Analysis of slope-parametric rectangular automata. In: Antsaklis, P.J., Kohn, W., Lemmon, M.D., Nerode, A., Sastry, S.S. (eds.) HS 1997. LNCS, vol. 1567, pp. 390–413. Springer, Heidelberg (1999)