Reachability Solution Characterization of Parametric Real-Time Systems

Farn Wang and Hsu-Chun Yen

Dept. of Electrical Engineering, National Taiwan University
Taipei, Taiwan 106, Republic of China
{farn, yen}@cc.ee.ntu.edu.tw

Abstract

We investigate the problem of characterizing the solution spaces for timed automata augmented by unknown timing parameters (called *timing parameter automata* (TPA)). The main contribution of this paper is that we identify three non-trivial subclasses of TPAs, namely, *upper-bound, lower-bound* and *bipartite* TPAs, and analyze how hard it is to characterize the solution space. As it turns out, we are able to give complexity bounds for the sizes of the minimal (resp., maximal) elements which completely characterize the upward-closed (resp., downward-closed) solution spaces of upper-bound (resp., lower-bound) TPAs. For bipartite TPAs, it is shown that their solution spaces are not semilinear in general. We also extend our analysis to TPAs equipped with counters without zero-test capabilities.

1 Introduction

Timed automata have been a popular model in the research of formal description and verification of real-time systems [1,2]. In real-world applications, systems are usually described with unknown parameters to be analyzed. Here we use the term *timing parameters* to refer to those parameters which are compared with clocks in either timed automata [4] or parametric TCTL formulae [14–17]. A timed automaton extended with unknown timing parameters is called a *timing parameter automaton (TPA)*. A *valuation* of unknown parameters making the goal state reachable in a TPA is called a *solution*. In this paper, we are mainly concerned with the following problem:

- The *reachability solution characterization (RSC)* problem: Given a TPA A and a goal predicate η, formulate a representation for the solution space of A with respect to η.

In [4], it has been shown that the emptiness problem becomes undecidable when three or more clocks are compared with unknown parameters in TPAs. Knowing such a limitation, a line of subsequent research has been focused on the solution characterization problem for a number of restricted versions of TPAs. See, e.g., [3,7,14–17]. The positive results obtained in the last few years have all been focused on unknown timing parameters in the specification of logic formulae. But in practice, it is more likely that engineers will use unknown parameters in the system behaviour descriptions. Moreover, engineers will be more interested at knowing the condition for solution parameters valuations than at knowing whether there exists a solution parameter valuation. In this work, we identify three subclasses of TPAs and investigate the complexity issue of their RSC problems. The three subclasses are called *upper-bound TPAs*, *lower-bound TPAs*, and *bipartite TPAs*. Consider a TPA and w.l.o.g., we assume that only $\leq$ and $<$ are used in the predicates of the TPA. An *upper-bound parameter* θ is one that only appears to the right of an inequality operator (e.g., $x - y \leq \theta, x < \theta$), whereas a *lower-bound parameter* θ appears to the left of an inequality operator (e.g., $\theta \leq x - y, \theta < x$). *Upper-bound* (resp. *lower-bound*) TPAs are those whose unknown parameters are all *upper-bound* (resp. *lower-bound*) parameters. *Bipartite* TPAs refer to those for which every unknown parameter is either a lower-bound parameter or an upper-bound parameter, but not both. Bipartite TPAs were considered in a recent article [10] in which the emptiness problem (undecidable for general TPA [4]) was shown to be decidable for such automata.

Intuitively, what makes *upper-bound* (resp. *lower-bound*) TPAs easier to analyze, in comparison with their general counterparts, lies in the fact that for each of such TPAs, the solution space is *upward-closed* (resp. *downward-closed*). (A set S over k-dimensional vectors of natural numbers, for some k, is called *upward-closed* (resp., *downward-closed*) if $\forall u \in S, v \geq u \Longrightarrow v \in S$ (resp., $\forall u \in S, v \leq u \Longrightarrow v \in S$)). It is well known that an upward-closed set (resp., downward-closed set) is completely characterized by its *minimal* (resp., *maximal*) elements, which always form a finite set although the set might not be effectively computable in general. As we shall see later in this paper, we are able to give a complexity bound for the sizes of the minimal elements for a given upper-bound TPA. Our analysis is carried out in a way similar to a strategy proposed in [13] (by Valk and Jantzen), in which a sufficient and necessary condition was derived under which the set of minimal elements of an upward-closed set is guaranteed to be effectively computable. (Note, however, that [13] reveals no complexity bounds for the sizes of the minimal elements.) Taking advantage of certain properties offered by timed automata, we are able to refine Valk and Jantzen's approach to yield complexity bounds for the sizes of the minimal elements for the upward-closed sets associated with upper-bound TPAs, allowing us to characterize their solution spaces. This in turn answers the *RSC* problem for upper-bound TPAs. To a certain extent, our result supplements the work of [10] (in which the emptiness problem was shown to be decidable for bipartite TPAs) by tackling a more general problem. We are also able to extend our analysis to the model of upper-bound *timing parameter vector addition systems with states (TPVASSs)*, each of which can be viewed as a TPA equipped with counters without zero-test capabilities. Once the sizes of minimal elements become available, finding all such elements can be done by exhaustive search using the region graph technique, although it would clearly be interesting to develop smarter (and more efficient) algorithms. Some complexity results are also derived for lower-bound TPAs. For bipartite TPAs, we are able to show that their solution spaces are not semilinear in general, in spite of the fact that the emptiness problem is decidable [10].

We feel that the method developed in this paper for analyzing upward-closed sets is interesting in its own right. Our technique refines the strategy of [13] (which deals with computing the minimal elements of upward-closed sets) in the following sense. Although the approach proposed in [13] is powerful for showing decidability for a variety of problems in a unified framework, the lack of information regarding the nature of the underlying system makes the calculation of the size of the associated minimal elements infeasible. Our study shows that if a key step in the algorithm of [13] meets certain conditions, then the sizes of the minimal elements can be deduced. It would be interesting to seek additional applications of our technique.

2 Models of Parametric Timed Systems

Let Z (N and R^+, resp.) be the set of all integers (nonnegative integers, and nonnegative reals, resp.), and Z^k (N^k, resp.) be the set of k-dimensional vectors of integers (nonnegative integers, resp.). Let $\mathbf{0}$ be the *zero vector*. Let $v(i), 1 \leq i \leq k$, denote the i-th component of a k-dimensional vector v. Given two vectors u and $v(\in N^k)$, $u \leq v$ if $\forall 1 \leq i \leq k, u(i) \leq v(i)$, and $u < v$ if $u \leq v$ and $u \neq v$. We define the *norm* of v, denoted by $||v||$, to be $max\{|v(i)| \mid 1 \leq i \leq k\}$, i.e., the absolute value of the largest component in v. For a set of vectors $V = \{v_1, ..., v_m\}$, the *norm* of V is defined to be $max\{||v_i|| \mid 1 \leq i \leq m\}$. In our subsequent discussion, we let $N_\infty = N \cup \{\infty\}$ (∞ is a new element capturing the notion of something being 'arbitrarily large', and for every $t \in N, t < \infty$ holds). We also let $N^k_\infty = (N \cup \{\infty\})^k = \{(v_1, ..., v_k) \mid v_i \in (N \cup \{\infty\}), 1 \leq i \leq k\}$. For a $v \in N^k_\infty$, we also write $||v||$ to denote $max\{v(i) \mid v(i) \neq \infty\}$, (i.e., the largest component in v excluding ∞) if $v \neq (\infty, ..., \infty)$; $||(\infty, ..., \infty)|| = 1$. Unless stated otherwise, we always assume that numbers are represented in *binary*, and the *size* of a number $t \in N$ is $\lceil log_2 t \rceil$.

A set $U(\subseteq N^k)$ is called *upward-closed* if $\forall u \in U$, $\forall v, v \geq u \Longrightarrow v \in U$. An element u ($\in N^k$) is said to be *minimal* if there is no v ($\neq u$) $\in U$ such that

$v < u$. We write $min(U)$ to denote the set of minimal elements of U. For an element $v \in N_\infty^k$, let $reg(v) = \{w \in N^k \mid w \le v\}$. A set $D(\subseteq N^k)$ is called *downward-closed* if $\forall u \in D, \forall v, \mathbf{0} \le v \le u \Longrightarrow v \in D$. An element $u\ (\in N_\infty^k)$ is said to be *maximal* if there is no $v\ (\neq u)\ \in D$ such that $v > u$. We write $max(D)$ to denote the set of maximal elements of D. For a given dimension, it is well known that every upward-closed (resp., downward-closed) set has a finite number of minimal (resp., maximal) elements. However, such finite sets may not be effectively computable in general. In an article [13] by Valk and Jantzen, the following result was proven which suggests a sufficient and necessary condition under which the set of minimal elements of an upward-closed set is effectively computable:

THEOREM 1 *([13]) For each upward-closed set $U(\subseteq N^k)$, $min(U)$ is effectively computable iff for every $v \in N_\infty^k$, the problem '$reg(v) \cap U \neq \emptyset$?' is decidable.*

It should be noted that in Theorem 1, the lack of information regarding the nature of the underlying system makes the calculation of the sizes of the associated minimal elements infeasible.

2.1 Timing parameter automata (TPA)

Given a set P of basic propositions, a set X of clocks, and a set H of unknown parameters, a *state predicate* η of P, X, and H has the following syntax rules.

$$\eta ::= \mathit{false} \mid p \mid x \sim c \mid x \sim \theta \mid \eta_1 \vee \eta_2 \mid \neg\eta_1$$

where $p \in P$, $x, y \in X$, $c \in N$, $\theta \in H$, $\sim \in \{\le, <, =, \ge, >\}$, and η_1, η_2 are state predicates. Notationally, we let $B(P, X, H)$ be the set of all state predicates on P, X, and H. Parentheses and traditional shorthands like $\Rightarrow, \wedge$ can also be used.

Definition 1 (Timing parameter automata, state and interpretation): A *timing parameter automaton* (TPA) is a tuple $(Q, q_0, X, H, \mu, E, \tau, \pi)$, where Q is a finite set of *modes* (operation modes, or control locations), $q_0 \in Q$ is the *initial mode*, X is a set of clocks with readings in R^+, H is a set of parameter variables with values in N, μ is a mapping from Q such that for each $q \in Q$, $\mu(q) \in B(\emptyset, X, H)$ is the invariance condition true in q, $E \subseteq Q \times Q$ is the set of transitions, $\tau : E \mapsto B(\emptyset, X, H)$ is a mapping which defines the transition-triggering conditions, and $\pi : E \mapsto 2^X$ defines the set of clocks to be reset during each transition. A *state* of TPA $A = (Q, q_0, X, H, \mu, E, \tau, \pi)$ is a pair (q, ν) such that $q \in Q$ and ν is a mapping from X to R^+ (i.e., ν represents the current clock readings). Let U_A be the state set of A. An *interpretation* I

for H is a mapping from $N \cup H$ to N such that for all $c \in N$, $I(c) = c$. ■

Given a TPA A and an interpretation I, A^I is the timed automaton obtained from A with all parameters interpreted according to I. Given a predicate $\eta \in B(P, X, H)$ and an interpretation I, η^I is the new predicate obtained from η with all parameters interpreted according to I.

Definition 2 (Satisfaction of state predicate with interpretation): A state (q, ν) satisfies state predicate $\eta \in B(Q, X, H)$ with interpretation I, written as $(q, \nu) \models_I \eta$, iff

- $(q, \nu) \not\models_I$ *false*;
- $(q, \nu) \models_I q'$ iff $q = q'$
- $(q, \nu) \models_I x \sim \theta$ iff $\nu(x) \sim I(\theta)$ where $\theta \in N \cup H$;
- $(q, \nu) \models_I \eta_1 \vee \eta_2$ iff $(q, \nu) \models_I \eta_1$ or $(q, \nu) \models_I \eta_2$; and
- $(q, \nu) \models_I \neg\eta_1$ iff $(q, \nu) \not\models_I \eta_1$.

If for all I, we have $(q, \nu) \models_I \eta$, then we may write $(q, \nu) \models \eta$. ■

Definition 3 (Transitions): Given two states $(q, \nu), (q', \nu')$, there is a *mode transition* from (q, ν) to (q', ν') in A with interpretation I, in symbols $(q, \nu) \rightarrow_I (q', \nu')$, iff $(q, q') \in E$, $(q, \nu) \models_I \mu(q) \wedge \tau(q, q')$, $(q', \nu') \models_I \mu(q')$, $\forall x \in \pi(q, q')(\nu'(x) = 0)$, and $\forall x \notin \pi(q, q')(\nu'(x) = \nu(x))$. ■

For ease of expression, given a state ν and a $\delta \in R^+$, we let $(q, \nu)+\delta = (q, \nu+\delta)$ be the state that agrees with (q, ν) in every aspect except for all $x \in X$, $\nu(x) + \delta = (\nu + \delta)(x)$.

Definition 4 ((q, ν)-run of interpreted TPA):
An infinite computation of $A=(Q, q_0, X, H, \mu, E, \tau, \pi)$ starting at state (q, ν) with interpretation I is called a (q, ν)-*run* and is a sequence $((q_1, \nu_1, t_1), (q_2, \nu_2, t_2), \ldots)$ such that

- $q = q_1$ and $\nu = \nu_1$;
- $t_1 t_2 \ldots$ is a monotonically increasing divergent sequence such that for each $t \in R^+$, there is an $i \in N$ such that $t_i \geq t$ (meaning that the run is diverging);
- for each integer $i \geq 1$ and for each real $0 \leq \delta \leq t_{i+1} - t_i$, $(q_i, \nu_i) + \delta \models_I \mu(q_i)$ (meaning that the invariance condition $\mu(q_i)$ continuously holds throughout the time interval $[t_i, t_{i+1}]$); and
- for each $i \geq 1$, A goes from (q_i, ν_i) to (q_{i+1}, ν_{i+1}) because of
  - a mode transition, i.e., $t_i = t_{i+1} \wedge (q_i, \nu_i) \rightarrow_I (q_{i+1}, \nu_{i+1})$; or
  - time passage, i.e., $t_i < t_{i+1} \wedge (q_i, \nu_i) + t_{i+1} - t_i = (q_{i+1}, \nu_{i+1})$. ■

Let $\langle 0 \rangle$ be the mapping that maps everything to zero. The initial state of a TPA $A = (Q, q_0, X, H, \mu, E, \tau, \pi)$ is $(q_0, \langle 0 \rangle)$. Given a TPA $A = (Q, q_0, X, H, \mu, E, \tau, \pi)$, a goal state-predicate $\eta \in B(Q, X, H)$, and an interpretation I, we say η is *reachable* in A with I, in symbols $A \rightsquigarrow_I \eta$, iff there exist a $(q_0, \langle 0 \rangle)$-run $= ((q_1, \nu_1, t_1), (q_2, \nu_2, t_2), \dots)$ in A, an $i \geq 1$, and a $\delta \in [0, t_{i+1} - t_i]$, s.t. $(q_i, \nu_i) + \delta \models_I \eta$. An interpretation I that makes $A \rightsquigarrow_I \eta$ is called a *solution* for A and η. The set of all solutions forms the so-called *solution space*. With respect to a given pair of A and $\eta \in B(Q, X, H)$, the problem of *finding a proper characterization for the solution space of A with respect to η* arises naturally in many real-world applications. Such a problem is called the *Reachability Solution Characterization (RSC)* problem. Throughout the rest of this paper, we write $RSC(A, \eta)$ to denote the solution space of automaton A with respect to predicate η.

2.3 Lower-bound, upper-bound, and bipartite TPAs

One of the major motivations in this work is to find practical classes of TPAs for which we can develop algorithms with known complexities for its RSC problem. First we need the following concepts. A predicate $\eta \in B(P, X, H)$ is in *literal form* iff in η, negation symbols only appear in front of elements in P; there is no negative signs immediately before inquality literals; and only $\leq$ and $<$ are used in inequality atoms. Every predicate can be transformed to a literal form in linear time. A TPA $A = (Q, q_0, X, H, \mu, E, \tau, \pi)$ is in called *literal TPA* iff $\mu(q)$ is in literal form for all $q \in Q$; and $\tau(q, q')$ is also in literal form for all $q, q' \in Q$. Every TPA can also be transformed to a literal TPA in linear time. In a literal TPA, if an unknown parameter θ appears to the right of an inequality operator in a literal (e.g. $x - y \leq \theta$, $x - y < \theta$, $x \leq \theta$, $x < \theta$), then θ is called an *upper-bound parameter*. If it appears to the left of an inequality operator in a literal (e.g. $\theta \leq x - y$, $\theta < x - y$, $\theta \leq x$, $\theta < x$), then it is called a *lower-bound parameter*.

Definition 5 *(***Bipartite, lower-bound, and upper-bound TPAs***):*
A *bipartite* TPA A=$(Q$,q_0,X,H,μ,E,τ,$\pi)$ is a literal TPA such that its set $\underline{H}$ of lower-bound parameters and set $\overline{H}$ of upper-bound parameters are disjoint, i.e. $\underline{H} \cap \overline{H} = \emptyset$. If $\underline{H} = \emptyset$, then A is also called an *upper-bound TPA.* If $\overline{H} = \emptyset$, then A is also called a *lower-bound TPA.* A predicate η is called an *upper-bound* (resp., *lower-bound*) predicate if all of its constituent parameters are *upper-bound* (resp., *lower-bound*) parameters. ■

There are two interpretations on bipartite TPA which serve importantly in defining the computability of the RSC problem. The first is the *maximum interpretation* I^M with which $I^M(\underline{\theta}) = 0$ for all $\underline{\theta} \in \underline{H}$; and $I^M(\overline{\theta}) = \infty$ for all $\overline{\theta} \in \overline{H}$. The second is the *minimum interpretation* I^m with which $I^m(\underline{\theta}) = \infty$ for all $\underline{\theta} \in \underline{H}$; and $I^m(\overline{\theta}) = 0$ for all $\overline{\theta} \in \overline{H}$. Note that maximum and minimum interpretations are not really interpretations as we defined in Definition 1 which does not map parameters to ∞. While interpreting $(q, \nu) \models_{I^M} \eta$ and $(q, \nu) \models_{I^m} \eta$, we shall assume $c \sim \infty = \mathit{true}$ and $\infty \sim c = \mathit{false}$, where $\sim \in \{<, \leq\}$.

We assume the basic knowledge of *region graph* constructions for timed automata presented in [1]. Suppose the biggest timing constant used in A and η is $C_{A:\eta}$. In [1], given a timed automaton A (or a TPA with an interpretation), a *region* for a state (q, ν) is a triple (q, γ, ϕ) such that γ records the integer parts of clock readings, at (q, ν), up to $C_{A:\eta}$ (when a clock reading is bigger than $C_{A:\eta}$, it is represented as ∞), and ϕ records the total ordering of the fractional parts of zero and clock readings at (q, ν). A *region graph* for a TCTL model-checking problem instance is constructed with region set as the node set and (timed and discrete) transition relation from regions to regions as the arc set. TCTL model-checking problems can be answered with CTL model-checking algorithms on *region graphs*. A rough bound on the number of regions in a region graph for A^{I_M} was computed as $\Lambda(|Q|, C_{A:\eta}, |X|) = |Q|((2+C_{A:\eta})(|X|+1))^{|X|}$. It is possible to get a tighter bound.

LEMMA 2 *$A \rightsquigarrow_{I^M} \eta$ iff there is a run, of less than $(\Lambda(|Q|, C_{A:\eta}, |X|) - 1)/2$ time units long, from the initial state $(q_0, \mathbf{0})$ to a state satisfying η.*

From now on, we shall let $\Gamma_{A:\eta} = (\Lambda(|Q|, C_{A:\eta}, |X|) - 1)/2$ for convenience.

In subsequent sections, we consider the problem of characterizing $RSC(A, \eta)$ when both TPA A and goal predicate η are upper-bound (or lower-bound). It is worthy of noting that in either case, the clock constraints in η can be built into TPA A. Consequently, the RSC problem can further be modified into one with only state reachability.

3 Computing Minimal Elements for Systems with Upward-Closed Solution Spaces

3.1 Upper-bound TPAs

Now consider upper-bound TPAs with upper-bound goal predicates. By establishing an ordering on the elements of H (i.e., $H = \{\theta_1, ..., \theta_k\}$, for some k), an interpretation can now be regarded as a $k-$dimensional vector in N^k. With a slight abuse of notation, for an interpretation I we write $I(\theta)$ to denote $I(i)$, where $\theta = \theta_i$. Given an interpretation I and a $\Delta \geq \mathbf{0} (\Delta \in N^k)$, we define $I+\Delta$ as the new interpretation such that for all $\theta \in H$, $(I+\Delta)(\theta) = I(\theta) + \Delta(\theta)$. The following lemma shows that the solution space for each upper-bound TPA w.r.t. an upper-bound goal predicate is *upward-closed.*

LEMMA 3 *For any upper-bound TPA A and upper-bound goal predicate η, if $A \leadsto_I \eta$ is true, then $\forall \Delta \geq \mathbf{0}$, $(A \leadsto_{I+\Delta} \eta)$. In words, the set of interpretations satisfying η is upward-closed.*

In view of the above lemma, each solution I can actually be regarded a *representative* for a convex space of solutions, called *funnel* of I. Given an interpretation I, we use $\langle I \rangle$ to represent the funnel pointing at I, i.e.,

$$\langle I \rangle = \{ I + \Delta \mid \Delta \geq \mathbf{0} \}$$

I is called the *point* of funnel $\langle I \rangle$. (Note that $\langle I \rangle$ has a unique minimal element, namely, I.) A set of funnels $\langle I_1 \rangle, \ldots, \langle I_m \rangle$ is called *mutually independent* iff each funnel is not a subset of the unions of the others, that is, $\forall 1 \leq i \leq m (\langle I_i \rangle \not\subseteq \bigcup_{1 \leq j \leq m; i \neq j} \langle I_j \rangle)$, or equivalently

$$\forall 1 \leq i < j \leq m \exists \alpha \in H \exists \alpha' \in H (I_i(\alpha) < I_j(\alpha) \wedge I_i(\alpha') > I_j(\alpha'))$$

Given an upper-bound TPA and an η, Lemma 3 suggests that $RSC(A, \eta)$ is upward-closed. Using the basic theory of timed automata (see, e.g., [2]), the problem of, given an extended interpretation $I \in N_\infty^k$, deciding '$reg(I) \cap RSC(A, \eta) \neq \emptyset$?' is clearly decidable. This observation, in conjunction with Theorem 1, yields the computability of the set of minimal elements of $RSC(A, \eta)$, although it reveals no information regarding the size of $min(A, \eta)$. In the remainder of this section, we shall take advantage of certain properties of timed automata to derive complexity bounds for computing the minimal elements of $RSC(A, \eta)$. Our analysis involves the following two steps.

- We first show that the solution space for upper-bound TPAs and η is a finite union of funnels; and
- With an inductive scheme on the number of unknown upper-bound parameters, we derive a finite bound on the magnitudes of parameter values of point solutions of the funnels in the finite union.

The position of an existent solution is important in identifying the finite structure of the solution space. Let I^a be the interpretation that maps every $\theta \in H$ to $1 + a$. Lemma 2 implies that if the solution space is nonempty, then $I^{(\Lambda(|Q|, C_{A:\eta}, |X|) - 1)/2}$ is a solution.

Let J be a *partial interpretation* of the parameters in H, that is, J is undefined for some parameters in H. For convenience, we write $J(\alpha) = \infty$, if α is undefined in J. (By doing so, J becomes a vector in N_∞^k, where $k = |H|$.) We conveniently use $\langle J \rangle$ as the union of all $\langle I \rangle$ such that $I(\alpha) = J(\alpha)$ for every α defined in J (i.e., $J(\alpha) \neq \infty$). Notice that $\langle J \rangle = \{I \in N^k \mid I(\alpha) = J(\alpha), \forall \alpha \neq \infty\}$. For instance, $\langle (5, \infty, 2, \infty) \rangle = \{(5, x, 2, y) \mid x, y \in N\}$. Thus we may also write $\langle I \rangle \subseteq \langle J \rangle$ if I agrees with J on every α defined in J.

Given a partial interpretation J, in symbols, we let $\bar{H}^J$ be the set of variables in H uninterpreted by J, that is $\bar{H}^J = \{\alpha \mid J(\alpha) \text{ is undefined}\}$. In $\langle J \rangle$, there can be non-solution interpretations for A and η. The following notation is for the characterization of those solution interpretations in $\langle J \rangle$. Given an J, we let $\Omega^J_{A:\eta}$ be the space of solutions I for $A \leadsto_I \eta$ with $\langle I \rangle \subseteq \langle J \rangle$. If J happens to be a total interpretation, then (1) $\Omega^J_{A:\eta} = \langle J \rangle = \{v \mid v \geq J\}$ in case $A \leadsto_J \eta$; and (2) $\Omega^J_{A:\eta} = \emptyset$ otherwise. For convenience, given a partial interpretation J and $a \in N$, we let $J[\alpha := a]$ be a new partial interpretation that agrees with J in every parameter except that $J[\alpha := a](\alpha) = a$.

LEMMA 4 *If J is a partial interpretation J and $I \in \Omega^J_{A:\eta}$ is a total interpretation, then*

$$\Omega^J_{A:\eta} = \langle I \rangle \cup \bigcup_{\alpha \in \bar{H}^J; 0 \leq a < I(\alpha)} \Omega^{J[\alpha:=a]}_{A:\eta} \qquad (1)$$

PROOF. We prove this by an induction on the size of $\bar{H}^J$. In the base case, $|\bar{H}^J| = 0$, J happens to be a total interpretation, and moreover $\bigcup_{\alpha \in \bar{H}^J; 0 \leq a < I(\alpha)} \Omega^{J[\alpha:=a]}_{A:\eta} = \emptyset$. Thus the base case is proven.

In the induction step, we assume that the lemma is true for all $|\bar{H}^J| \leq k$ with $k \geq 0$. Let us consider a solution interpretation $I' \in \Omega^J_{A:\eta}$. If $\langle I' \rangle \subseteq \langle I \rangle$, then the lemma is proven. Now let us consider the case that $\langle I' \rangle \not\subseteq \langle I \rangle$. In this case, this means that there is an $\alpha \in \bar{H}^J$ such that $I'(\alpha) < I(\alpha)$. By enumerating all the possibilities of $\alpha \in \bar{H}^J$ and all the possible values in $[0, I(\alpha))$, we obtain the expression of $\bigcup_{\alpha \in \bar{H}^J; 0 \leq a < I(\alpha)} \Omega^{J[\alpha:=a]}_{A:\eta}$. Thus the induction step is proven. Thus the lemma is also proven. ■

The importance of Lemma 4 is that once we can find a solution interpretation I in the solution space, the lemma suggests a way to inductively and compositionally construct the solution space by means of unions of funnels. But according to Lemma 2, we do know how to find this special interpretation I based on a given partial interpretation. As we shall see later, the ability to find such an I plays a critical role in deriving a complexity bound for the size of $min(A, \eta)$. To emphasize its importance, the following definition is given to capture the idea stated above.

- (*Small witness property*:) For a given partial interpretation J, a 'small' total interpretation $I \in \Omega^J_{A,\eta}$ can be computed effectively.

Let $(\infty, ..., \infty)$ be the partial interpretation that is undefined on everything. Lemma 4, together with the fact that $RSC(A, \eta) = \Omega_{A:\eta}^{(\infty,...,\infty)}$, suggests an algorithm for computing the constituent funnels of $RSC(A, \eta)$, provided that $I \in \Omega_{A:\eta}^{J}$ is computable for every partial interpretation J. That is, if we view $\Omega_{A:\eta}^{J}$ as a procedure-call with parameters A, η, and J, then we can construct the solution space representation by invoking $\Omega_{A:\eta}^{(\infty,...,\infty)}$.

By examining all components of Formulus (1), we find that every component in (1) is with straightforwardly known complexity except $I \in \Omega_{A:\eta}^{J}$. It is obvious that if we can find bounds on the vector $I \in \Omega_{A:\eta}^{J}$ for each J, then we can multiply and sum up all the component complexities to derive the complexity for the RSC problem. The major difficulty is to carefully account for all the component complexities so that bounds can be derived. Let A^J be the new timed automaton obtained from A by substituting every defined θ in J for $J(\theta)$; and substituting every undefined θ' in J for ∞. The bounds can be obtained by using Lemma 2. That is, we can construct the region graph for A^J and η and use the length of the longest simple path in the graph to bound the vector components in I. In the same reasoning of Lemma 2, we know that there is an interpretation $I \in \Omega_{A:\eta}^{J}$ making $A \leadsto_I \eta$ iff $A^J \leadsto \eta$. According to [1], the size of region graph is bounded by

$$2|Q| \cdot (|X| + 1)^{|X|} \cdot (C_{A^J:\eta} + 1)^{|X|} \tag{2}$$

In a region graph of this size, the time-span of the shortest path from one region to another can always be bounded by $|Q| \cdot (|X| + 1)^{|X|} \cdot (C_{A^J:\eta} + 1)^{|X|}$. According to the same reasoning of Lemma 2, we can bound each component of the I with $|Q| \cdot (|X| + 1)^{|X|} \cdot (C_{A^J:\eta} + 1)^{|X|}$.

LEMMA 5 *For every partial interpretation J, there is an $I \in \Omega_{A:\eta}^{J}$ such that for every $\theta \in \bar{H}^J$, $I(\theta) \leq |Q| \cdot (|X| + 1)^{|X|} \cdot (C_{A^J:\eta} + 1)^{|X|}$.*

We want to construct the inductive definition of the magnitude of $C_{A^J:\eta}$. That is, we want to define $C_{A^J:\eta}$ based on those partial interpretations which define one less parameters than J does. The following lemma unwinds Formulus (2) for a bound on the complexity of $I \in \Omega_{A:\eta}^{J}$ for each J. For convenience, we let $|J|$ be the number of parameters defined in J.

LEMMA 6 *In Formulus (1), for every partial interpretation J, there is an $I \in \Omega_{A:\eta}^{J}$ such that for every $\theta \in \bar{H}^J$, $I(\theta)$ is $O\big((|Q| \cdot (|X| + 1)^{|X|})^{\sum_{0 \leq i \leq |J|} |X|^i} \cdot (C_{A:\eta} + 1)^{|X|^{1+|J|}}\big)$.*

PROOF. Base case: $|J| = 0$. In this case, the bound is $|Q| \cdot (|X|+1)^{|X|} \cdot (C_{A:\eta}+1)^{|X|}$, which is exactly the size bound of the region graph for A and η with I^M. This case is true according to Lemma 2.

The inductive hypothesis is that there is an $I \in \Omega^J_{A:\eta}$ such that for every $\theta \in \bar{H}^J$, $I(\theta)$ is $O\left(\left(|Q| \cdot (|X|+1)^{|X|}\right)^{\sum_{0\leq i\leq |J|}|X|^i} \cdot (C_{A:\eta}+1)^{|X|^{1+|J|}}\right)$. This means that in the induction step, the biggest timing constant used in $A^{J[\theta:=a]}$ is of the same complexity. Thus we deduce, according to Formulus (2), that the size bound of the region graph for $A^{J[\theta:=a]}$ is

$$O(2|Q| \cdot (|X|+1)^{|X|} \cdot \left(\left(|Q| \cdot ((|X|+1)^{|X|})\right)^{\sum_{0\leq i\leq |J|}|X|^i} \cdot (C_{A:\eta}+1)^{|X|^{1+|J|}} + 1\right)^{|X|})$$
$$= O(2|Q| \cdot (|X|+1)^{|X|} \cdot \left(\left(|Q| \cdot ((|X|+1)^{|X|})\right)^{|X|\sum_{0\leq i\leq |J|}|X|^i} \cdot (C_{A:\eta}+1)^{|X|^{1+|J|}\cdot|X|}\right))$$
$$= O\left(\left(|Q| \cdot (|X|+1)^{|X|}\right)^{\sum_{0\leq i\leq 1+|J|}|X|^i} \cdot (C_{A:\eta}+1)^{|X|^{2+|J|}}\right)$$

Since $|J|+1 = |J[\theta := a]|$ and the reachability does not need a path time greater than the number of regions, thus the complexity is proven. ■

Using Lemmas 4 and 6, we have the following:

THEOREM 7 *Given an upper-bound TPA A and an upper-bound predicate η, $||min(RSC(A,\eta))||$ is bounded by $O((|Q| \cdot |X| \cdot C_{A:\eta})^{|X|^{c*|H|}})$, where c is a constant.*

Once the sizes of minimal elements become available, finding all such elements can be done by exhaustive search using the region graph technique, although it would clearly be desirable to develop smarter (and more efficient) algorithms.

3.3 An example for the super-exponential complexity

Here we demonstrate an example TPA family which has n upper-bound parameters whose solutions grow super-exponentially. This example serves to show that the bounds that we shall derive in this section is roughly tight.

The example TPA family have members $A_1, \ldots, A_n, \ldots$ where A_n has n unknown upper-bound parameters $\alpha_1, \ldots, \alpha_n$. The TPA for A_n is depicted in Figure 1 with $2n$ clocks: $x_1, y_1, \ldots, x_n, y_n$. For each A_n, there are three types of transitions. Type A transitions are used to maintain a rough period of 3 time units. Type B transitions make sure that for each $1 \leq i \leq n$, we can enforce a super minimum period of γ_i such that $\gamma_i = \Pi_{1\leq j<i}\gamma_j + 1$, which is mutually prime to each of $\gamma_1, \ldots, \gamma_{i-1}$. The i'th type B transition happens when the first through the $i-1$'st type B transitions have all happened. Also the happening of the i'th type B transition also means that the i'th component

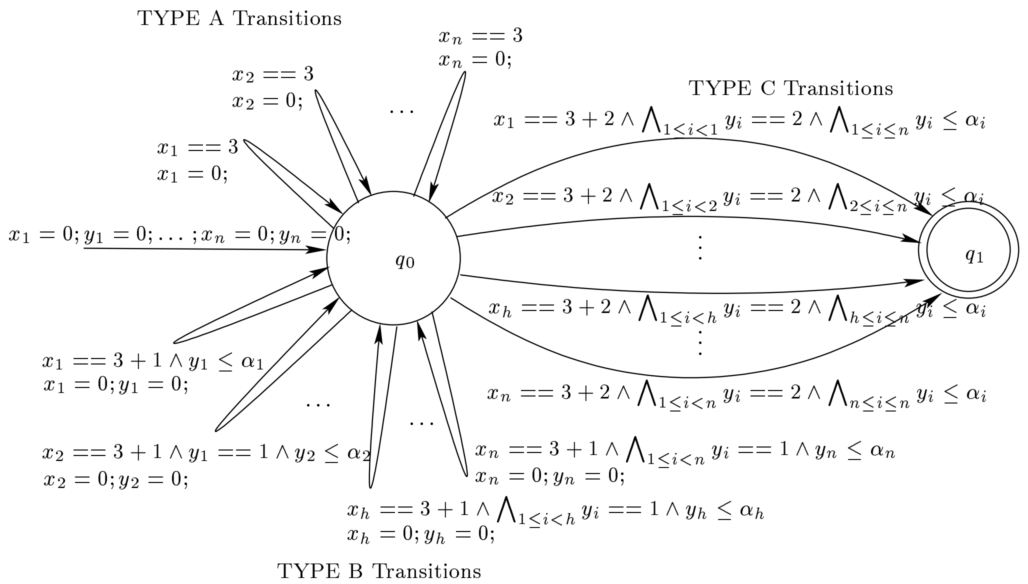

Fig. 1. A TPA with super-exponential solutions.

of all the $i+1$'st thrugh n'th minimal points can be as much as $\Pi_{1\leq j<i}\gamma_j + 1$.

Type C transitions are used to detect the goal predicate. The i'th minimal point occurs when the i'th transition of type C happens and for all $i \leq j \leq n$, the j'th component of the i'th minimal point can be as much as $\Pi_{1\leq j<i}\gamma_j + 2$.

3.4 Timing parameter vector addition systems with states

Our technique can also be applied to analyzing upper-bound *timing parameter vector addition systems with states* (TPVASSs). An *m-dimensional vector addition system* (VAS) is a pair (v_0, V) where $v_0 \in N^m$ is called the *start vector*, and V, a finite subset of Z^m, is called the set of *addition rules*. A vector $z \in N^m$ is said to be reachable in VAS (v_0, V) if $z = v_0 + v_1 + \cdots + v_j$ for some $j \geq 0$, where each $v_i (1 \leq i \leq j)$ is in V and, for each $1 \leq i \leq j$, $v_0 + v_1 + \cdots + v_i \geq \mathbf{0}$. The *covering* problem of VASs is that of, given a VAS (v_0, V) and a vector v, deciding whether there is a reachable vector z such that $z \geq v$ (i.e., z covers v). An *m-dimensional vector addition system with states* (VASS) is a 5-tuple (v_0, V, p_0, S, δ) where v_0 and V are the same as defined above, S is a finite set of *states*, $\delta (\subseteq S \times S \times V)$ is the transition relation, and p_0 ($\in S$) is the *initial state*. Elements (p, q, v) of δ are called *transitions* and are usually written as $p \to (q, v)$. A configuration of a VASS is a pair (p, u) where $p \in S$ and $u \in N^m$. (p_0, v_0) is the *initial configuration*. The transition $p \to (q, v)$ can be applied to the configuration (p, u) and yields the configuration $(q, u + v)$, provided that $u + v \geq \mathbf{0}$. The reader is referred to [9,11] for more about VASs and VASSs.

A *timing parameter VASS* (TPVASS) is a tuple $(v_0, V, Q, q_0, X, H, \mu, E, \tau, \pi)$ where v_0 and V represent the start vector and the addition rules, respectively, associated with TPA $(Q, q_0, X, H, \mu, E, \tau, \pi)$. One may view a TPVASS as a TPA equipped with counters without zero-test capabilities. The *dimension* of a TPVASS is the dimension of its constituent VAS. It is important to point out the the way *time* is introduced in this computational model differs from that in the conventional *timed* (or *time*) Petri nets. Unlike the case in the strong firing semantics of *timed* (or *time*) Petri nets, it is not required to fire all the enabled transitions at any point in time during the course of a computation. In our setting it is perfectly legal for time to elapse, causing enabled transitions to become disabled without being fired.

Using the technique of region graphs, we have:

LEMMA 8 *: Given a TPVASS $A = (v_0, V, Q, q_0, X, H, \mu, E, \tau, \pi)$, an interpretation I, and an upper-bound predicate $\eta \in B(Q, X, H)$, we can construct a VASS $M_{A,\eta,I} = (v_0, V, p_0, S, \delta)$ and a state $s \in S$ such that $A \leadsto_I \eta$ iff there is a computation from (p_0, v_0) to (s, v) in $M_{A,\eta,I}$, for some v. Furthermore, $|S| = O(2|Q| \cdot (|X|+1)^{|X|} \cdot (max\{C_{A:\eta}, ||I||\}+1)^{|X|})$.*

Consider the *RSC* problem for an m-dim TPVASS A (with respect to predicate η). From A we construct a new TPVASS of dimension $m+1$ in such a way that from each configuration satisfying η, a transition incrementing the new (i.e., $(m+1)$-th) position by one is introduced. Clearly the *RSC* problem has a solution iff in the new TPVASS, there is a computation reaching a configuration with non-zero in the $(m+1)$-th position. Based upon the discussion above and the fact that an m-dim VASS A (with s and l as the number of states and the largest integer mentioned in A, respectively) can be simulated by an $(m+3)$-dim VAS A' whose largest integer is bounded by $max\{s^2, l\}$ (from Lemma 2.1, [9]), we have:

COROLLARY 9 *: Given an m-dim TPVASS A=(v_0, V, Q, q_0, X, H, μ, E,τ, π), an interpretation I, and an upper-bound predicate $\eta \in B(Q, X, H)$, we can construct an $(m+4)$-dim VAS $W_{A,\eta} = (v_0', V')$ such that $A \leadsto_I \eta$ iff there is a computation from v_0' to a vector $v'' \geq (0, ..., 0, 1)$ in the VAS, for some v''. Furthermore, $||W_{A,\eta}||$ is bounded by $max\{(2|Q| \cdot (|X|+1)^{|X|} \cdot (max\{C_{A:\eta}, ||I||\}+1)^{|X|})^2, ||V||, ||v_0||\}$.*

It is known from [11] that given a VAS $W = (v_0, V)$, a vector v can be covered in W iff there exists a short covering path whose length is bounded by $O(2^{2^{cnlogn}})$, where c is a constant and n is the size of the VAS. By treating the *dimension* and *norm* of a VAS as two separate parameters, an improved bound of $O(s^{2^{d*m*logm}})$ for the length of the shortest covering path (where s and m are the norm and dimension of the VAS, respectively, and d is a constant) can be found in [12]. This, in conjunction with Corollary 9, allows us to derive the following result. Due to space limitations, the proof details are omitted.

THEOREM 10 *Given an upper-bound TPVASS A and an upper-bound predicate* η, $||min(RSC(A,\eta))||$ *is bounded by* $O((|Q|\cdot|X|\cdot C_{A:\eta})^{2^{c\cdot m\cdot logm}\cdot|X|^{d|H|}})$, *where* c, d *are constants.*

3.5 A unified strategy for analyzing upward-closed sets

In what follows, we propose a framework using which the sizes of the minimal elements in an upward closed set can be calculated. The idea is the following. In [13], the key in proving decidability lies in the ability of, given an arbitrary $v \in N_\infty^k$, testing whether '$reg(v) \cap U \neq \emptyset$?'. Now suppose in addition to the ability to test '$reg(v) \cap U \neq \emptyset$?', we are also able to compute the size of a witnessing vector w in $reg(v) \cap U$, if such a vector exists. That is, the small witness property holds for the system under consideration. In this case, the following result can be proven along a line similar to the proof of Theorem 1 presented in [13]. More precisely,

THEOREM 11 *For each upward-closed set* $U(\subseteq N^k)$, *if given a* $v \in N_\infty^k$, *a witness* w *for '*$reg(v) \cap U \neq \emptyset$*' (if one exists) can be computed such that* $||w|| \leq f(||v||)$, *for some function* f, *then* $||min(U)|| \leq (\overbrace{f\circ\cdots\circ f}^{k})(1)$.

It would be interesting to seek additional applications of the above result.

4 Computing Maximal Elements for Systems with Downward-Closed Solution Spaces

For TPAs with unknown lower-bound timing parameters, we still want to find characterization for the solution interpretation. In this case, there is one thing worth noting: *The solution space for the unknowns is downward-closed.* Geometrically, this means that the solution space is a union of "bottom-up" funnels. For convenience, we call such bottom-up funnels *cones*, which can be characterized by the maximal solutions of those cones. If we can find the upper-bounds for the maximal solutions, if any, of those cones, then we can shape the solution space in this case.

LEMMA 12 *Given a lower-bound TPA A and a lower-bound predicate* η, *if* $A \rightsquigarrow_I \eta$ *for some* I *such that there is an* $\underline{\theta}$ *with* $I(\underline{\theta}) > \Gamma_{A:\eta}$, *then for all* I' *such that* I' *agrees with* I *on all parameters except* $I'(\underline{\theta}) > I(\underline{\theta})$, $A \rightsquigarrow_{I'} \eta$.

PROOF. If $I(\underline{\theta}) > \Gamma_{A:\eta}$, then there is a run along which a clock x is incremented beyond $\Gamma_{A:\eta}$ and tested against condition $x \sim I(\underline{\theta})$ where $\sim \in \{\text{“}>\text{”}, \text{“}\geq\text{”}\}$. This means that there is a path in the region graph of A^{I^M} whose length of time is greater than $\Gamma_{A:\eta}$ and at the end of the path, $x \sim I(\underline{\theta})$ is tested. Let X' be the set of clocks whose reading is greater than $\Gamma_{A:\eta}$ at the end of the path. Since the length of time is $> \Gamma_{A:\eta}$, we can pick a sequence of $\Gamma_{A:\eta} + 1$ regions along the path such that two successive regions are separated by one time unit. Among these regions, there must be two identical regions and between these two regions, the time-length is no less than 1. By repeating this cycle any number of times, we get a new path (or run) ρ' along which the readings of some clocks, including all clocks in X', becomes arbitrarily large when tested against lower-bound conditions with unknowns. This ρ' is not only a run of A^{I^M} bus also a run of $A^{I'}$. This means no matter how big $I'(\underline{\theta})$ is, we can still find a path to make a run in A^{I^M} to make η^{I^M} reachable. Thus the lemma is proven. ■

Lemma 12 implies that to search for maximal solutions in cones, we only have to check the reachability of $A \rightsquigarrow_I \eta$ with $I(\theta) \leq \Gamma_{A:\eta} + 1$ for all θ. If I is a solution, then so is the cone characterized by

$$\bigwedge_{I(\theta) \leq \Gamma_{A:\eta}} \theta \leq I(\theta)$$

which puts no restrictions on those parameters θ with $I(\theta) = \Gamma_{A:\eta} + 1$. Thus a simple way to formulate the algorithm for the RSC problem of lower-bound TPAs is with the following formula for the corresponding solution space.

$$\bigvee_{I: A \rightsquigarrow_I \eta \wedge \forall \theta, 0 \leq I(\theta) \leq \Gamma_{A:\eta}+1} \bigwedge_{I(\theta) \leq \Gamma_{A:\eta}} \theta \leq I(\theta) \tag{3}$$

THEOREM 13 *Given a lower-bound TPA A and a lower-bound predicate η, the size of $||max(A, \eta)||$ can be computed in PSPACE.*

PROOF. In Expression (3), we need PSPACE to query each question of $A \rightsquigarrow_I \eta$ given that the constants used in A^I is no greater than $\Gamma_{A:\eta} + 1$. To see this, we know that each such basic query needs the search of a region graph with at most $2|Q| \cdot (|X|+1)^{|X|} \cdot (\Gamma_{A:\eta}+2)^{|X|}$ regions. Expanding the expression in complexity notation, we get

$$\begin{aligned}
& O(2|Q| \cdot (|X|+1)^{|X|} \cdot (2|Q| \cdot (|X|+1)^{|X|} \cdot (C_{A:\eta}+1)^{|X|} + 2)^{|X|}) \\
= \; & O(2|Q| \cdot (|X|+1)^{|X|} \cdot (2^{|X|}|Q|^{|X|} \cdot (|X|+1)^{|X|^2} \cdot (C_{A:\eta}+1)^{|X|^2})) \\
= \; & O\left(2^{|X|+1}|Q|^{|X|+1} \cdot (|X|+1)^{|X|^2+|X|} \cdot (C_{A:\eta}+1)^{|X|^2}\right)
\end{aligned}$$

With a counter, we can explore the region graphs in full. The number of bits in this counter is

$$O\left(\log\left(2^{|X|+1}|Q|^{|X|+1}\cdot(|X|+1)^{|X|^2+|X|}\cdot(C_{A:\eta}+1)^{|X|^2}\right)\right)$$
$$= O(|X|+1+(|X|+1)\log|Q|+(|X|^2+|X|)\log(|X|+1)+|X|^2\log(C_{A:\eta}+1))$$

As can be seen from the above, we need only polynomial space in the counters and polynomial space to record a single region in order to explore the region graph.

Then according to Formula (3), we need $|H|$ counters of $\log(\Gamma_{A:\eta}+1)$ bits each to carry out the enumeration of the outer disjunction. Thus the total memory used is still in PSPACE. ■

For lower-bound TPVASSs, the argument used in the proof of Lemma 12 does not work, since a 'loop' in the region graph may not be repeatable due to the possibility of a loss in the counter value.

So far we have seen that for restricted subclasses such as upper-bound and lower-bound TPAs, their solution spaces are upward-closed and downward-closed, respectively. This, together with a recent result of [10] showing the emptiness problem to be decidable for bipartite TPAs, leaves us to wonder whether the solution space of a bipartite TPA remains semilinear or not. Following a result in [4] that the solution spaces for general TPAs are not necessarily semilinear, it is reasonably easy to show:

THEOREM 14 *The solution spaces of bipartite TPAs are not semilinear in general.*

5 Conclusion

We have studied in detail the sizes of the minimal (maximal, resp.) elements of upward-closed (downward-closed, resp.) solution spaces associated with upper-bound (lower-bound, resp.) TPAs. A line of future research for upper-bound TPAs (and TPVASSs) is to explore the possibility of manipulating and characterizing the computations and the solution spaces in a symbolic fashion. One way to do this, perhaps, is to take a closer look at data structures designed explicitly for upward-closed sets, such as the so-called *sharing trees* of [6]. Finding how tight our complexity bounds for upper-bound and lower-bound TPAs are remains a question to be answered.

References

[1] R. Alur, C. Courcoubetis, D.L. Dill. Model-Checking in Dense Real-Time, *Information and Computation* 104(1), 2-34, 1990.

[2] R. Alur, D.L. Dill, D. Automata for Modeling Real-Time Systems, in in *Proc. 17th ICALP*, LNCS 443, pp. 332–335, 1990.

[3] R. Alur, K. Etessami, S. La Torre, D. Peled. Parametric Temporal Logic for Model Measuring, in *Proc. 26th ICALP*, LNCS 1644, pp. 169-178, 1999.

[4] R. Alur, T.A. Henzinger, M.Y. Vardi. Parametric Real-Time Reasoning, in *Proc. 25th ACM STOC*, pp. 592–601, 1993.

[5] B. Berard. Untiming Timed Languages, *Inform. Proc. Lett.*, 55:129-135,1995.

[6] G. Delzanno, J.-F. Raskin. Symbolic Representation of Upward-closed Sets, in *Proc. TACAS 2000*, LNCS 1785, pp. 426-440, 2000.

[7] E.A. Emerson, R. Trefler. Parametric Quantitative Temporal Reasoning, in *Proc. IEEE LICS*, pp. 336–343, July 1999.

[8] T.A. Henzinger, X. Nicollin, J. Sifakis, S. Yovine. Symbolic Model Checking for Real-Time Systems, *Information and Computation*, 111, 193-244, 1994.

[9] J. Hopcroft, J. Pansiot. On the Reachability Problem for 5-Dimensional Vector Addition Systems, *Theoretical Computer Science*, 8, 135-159, 1979.

[10] T. Hune, J. Romijn, M. Stoekinga, F. Vaandrager. Linear Parametric Model Checking of Timed Automata, in *Proc. TACAS 2001*, Italy, April, 2001, LNCS 2031, pp. 189-203.

[11] C. Rackoff. The Covering and Boundedness Problems for Vector Addition Systems, *Theoretical Computer Science*, 6, 223-231, 1978.

[12] L. Rosier, H. Yen. A Multiparameter Analysis of the Boundedness Problem for Vector Addition Systems, *J. Comput. System Sci.*, 32, 105-135, 1986.

[13] R. Valk, M. Jantzen. The Residue of Vector Sets with Applications to Decidability in Petri Nets, *Acta Informatica*, 21, 643-674, 1985.

[14] F. Wang. Parametric Timing Analysis for Real-Time Systems, *Information and Computation*, 130(2), 131-150, 1996. Also in *Proc. 10th IEEE LICS*, 1995.

[15] F. Wang. Parametric Analysis of Computer Systems, *Formal Methods in System Design*, 17, 39-60, 2000.

[16] F. Wang, P.-A. Hsiung. Parametric Analysis of Computer Systems, Australia, Dec. 1997, in *Proc. AMAST'97*, LNCS 1349, pp. 539-553.

[17] F. Wang, H.-C. Yen. Parametric Optimization of Open Real-Time Systems, Paris, July 2001, in *Proc. SAS 2001*, LNCS 2126, pp. 299-318.