Parameter Synthesis for Timed Kripke Structures

Extended Abstract

Michał Knapik[1] and Wojciech Penczek[1,2]

[1] Institute of Computer Science, PAS, Warsaw, Poland
[2] University of Natural Sciences and Humanities, II, Siedlce, Poland
{knapik,penczek}@ipipan.waw.pl

Abstract. We show how to synthesise parameter values under which a given property, expressed in a certain extension of CTL called $\mathrm{RTCTL}_{\mathrm{P}}$, holds in a parametric timed Kripke structure. Similarly as in fixed-point symbolic model checking approach, we introduce special operators which stabilise on the solution. The process of stabilisation is essentially a translation from $\mathrm{RTCTL}_{\mathrm{P}}$ parameter synthesis problem to a discrete optimization task. We argue that this leads to new opportunities in model checking, including the use of integer programming and related tools.

1 Introduction

Complex systems, both hardware and software, present in critical areas need to be verified. The best moment for the verification is the design phase, perhaps even before any prototype is developed. This helps to reduce errors and costs; the found flaws can also provide valuable pointers to a designer.

Model checking is one of the established methods for verification of complex, timed, and reactive systems. In this approach, a model for a verified system is built (e.g. a Kripke structure or a Petri net), and a property to be checked is specified in a version of a modal logic (e.g. CTL or TCTL). The pair consisting of a model and a formula is the input for a model checking tool. The output is simply the *property holds* or *property does not hold* answer.

However, such an approach has its drawbacks. In the beginning phases of a system design some of the features required in a model might be unknown (e.g. timing constraints), which forces the designer to substitute them with some guessed or standard values. Even if it is possible to present a full model of the system, there is no guarantee that this specification will not be subject to some changes. Often the minimal alteration of the original model may lead to violation of a checked property, therefore the process of verification has to be repeated.

A system designer using model checking methods would substantially benefit from a tool that is able to accept an underspecified model with some values abstracted as parameters. In this case the expected output consists of a set of parameter valuations under which a given property holds. This approach is called *parametric model checking* or *parameter synthesis*. Parametric model checking eliminates the needs for guessing and for performing batches of tests for ranges of values.

In this paper we show how to perform parameter synthesis for timed Kripke structures, i.e., Kripke structures where transition is augmented with an additional label specifying how long it takes to traverse it. The input logic is a certain extension of Computation Tree Logic, which allows for expressing properties over the restricted fragments of paths.

1.1 Related Work and Paper Outline

The logic considered in this paper and its models are based on the Real Time Computation Tree Logic (RTCTL) and timed Kripke structures introduced in [1].

As we show, the problem of parameter synthesis is decidable for $\mathrm{RTCTL_P}$. It is however not decidable for even as simple properties as reachability for many other models, e.g. parametric timed automata (PTA) [2, 3] and bounded parametric time Petri nets [4]. Difference bound matrix - based semi-algorithms for reachability were extended to the PTA case in [5] and implemented in UPPAAL-PMC. In [6] we showed how to synthesise by means of bounded model checking a part of the set of valuations for PTA reachability. The problem of synthesis of bounded integer valuations for PTA is analysed in [7] and shown to be in PSPACE. In [8] the authors show how to synthesise the constraints on valuations under which a PTA is *time-abstract* equivalent to some initial one; the work is implemented in IMITATOR prototype tool. Parametric analysis is also possible with HyTech [9] by means of hybrid automata.

In the next section we introduce the $\mathrm{RTCTL_P}$ logic and its models. In Section 3 we show how to solve the synthesis problem via a translation to sets of linear inequalities over natural numbers. We conclude the work with a comment on the possible benefits and downsides of our approach and future plans.

2 Parameterized Temporal Logics

Let $\mathbb{N}$ denote the set of all natural numbers (including 0), and let $P(D)$ denote the power set of a set D. For any sequence $x = (x_1, \ldots, x_n)$ and $0 \leq i \leq n$, let $x|_i = x_i$ be the projection of x on the i–th variable.

2.1 The Syntax of $\mathrm{RTCTL_P}$

The Real Time CTL [1] allows to express branching-time temporal properties involving the integer time-step depth of considered paths.

Definition 1 (Syntax of $\mathrm{RTCTL_P}$). *Let $\mathcal{PV}$ be a set of propositional variables containing the symbol* true. *The formulae of* $\mathrm{RTCTL_P}$ *are defined as follows:*

1. *every member of $\mathcal{PV}$ is a formula,*
2. *if α and β are formulae, then so are $\neg\alpha$, $\alpha \wedge \beta$,*

3. if α and β are formulae, then so are $EX^{\leq k}\alpha$, $EG^{\leq k}\alpha$, $E\alpha U^{\leq k}\beta$ for $k \in \mathbb{N}$.

As to give an example of the meaning of an RTCTL$_{\mathrm{P}}$ formula, $EG^{\leq 5}p$ states that *"there exists a path such that p holds in each state reached from the beginning in time not greater than* 5.*"*

2.2 The Semantics of RTCTL$_{\mathrm{P}}$

We evaluate the truth of the formulae in the parametric timed Kripke structures. These are standard Kripke structures with the transitions decorated by additional labels interpreted as time variables.

Definition 2. *A parametric timed Kripke structure (a* model*) is a 5-tuple* $\mathrm{M} = (S, s^0, T, \rightarrow, \mathcal{L})$ *where:*

- *S is a finite set of* states*,*
- *$s^0 \in S$ is the* initial state*,*
- *T is a set of* time step parameters *(variables),*
- *$\rightarrow \subseteq S \times T \times S$ is a transition relation such that for every $s \in S$ there exists $s' \in S$ and $t \in T$ with $(s, t, s') \in \rightarrow$ (i.e., the relation is total),*
- *$\mathcal{L} : S \longrightarrow 2^{\mathcal{PV}}$ is a valuation function satisfying true $\in \mathcal{L}(s)$ for each $s \in S$.*

Let s, s' be two states of a model, and let t be a time step parameter. By $s \xrightarrow{t} s'$ we denote that $(s, t, s') \in \rightarrow$. The intuitive meaning of $s \xrightarrow{t} s'$ is that it takes t time units to reach s' from s. We define $in(s)$, $out(s)$, $link(s, s')$ as the sets of the labels of the transitions entering s, leaving s, and connecting s with s', respectively. More formally, $in(s) = \{t \in T \mid s' \xrightarrow{t} s \text{ for } s' \in S\}$, $out(s) = \{t \in T \mid s \xrightarrow{t} s' \text{ for } s' \in S\}$, and $link(s, s') = \{t \in T \mid s \xrightarrow{t} s'\}$.

A function $\omega : T \rightarrow \mathbb{N}$ is called a *parameter valuation*. The set of all the parameter valuations is denoted by Ω. Consider an infinite sequence $\pi = (s_0, t_0, s_1, t_1, \ldots)$ such that $s_i \in S$ and $s_i \xrightarrow{t_i} s_{i+1}$ for $i \in \mathbb{N}$. By $\pi_i = s_i$ we denote the i–th state of π. We define the *time distance function* between the positions π_0 and π_j on a sequence π as $\delta_\pi^j = \sum_{i=0}^{j-1} t_i$, and we assume that $\delta_\pi^0 = 0$. If ω is a parameter valuation, then let $\delta_\pi^j(\omega) = \sum_{i=0}^{j-1} \omega(t_i)$. A sequence π is called an *ω–path* if $lim_{j\rightarrow\infty}\delta_\pi^j(\omega) = \infty$, or simply a *path* if ω is evident from the context.

Definition 3 (Semantics of RTCTL$_{\mathrm{P}}$**).** *Let* $\mathrm{M} = (S, s^0, T, \rightarrow, \mathcal{L})$ *be a model and $s \in S$. Let $\alpha, \beta \in$* RTCTL$_{\mathrm{P}}$*, let $\omega \in \Omega$ be a parameter valuation, and $k \in \mathbb{N}$. $M, s \models_\omega \alpha$ denotes that α is true at the state s of M under the valuation ω. (In what follows we omit M where it is implicitly understood.) The relation $\models_\omega$ is defined inductively as follows:*

1. $s \models_\omega p$ iff $p \in \mathcal{L}(s)$,
2. $s \models_\omega \neg\alpha$ iff $s \not\models_\omega \alpha$,
3. $s \models_\omega \alpha \wedge \beta$ iff $s \models_\omega \alpha$ and $s \models_\omega \beta$,
4. $s \models_\omega EX^{\leq k}\alpha$ iff there exists a path π s.t. $\pi_0 = s, \delta_\pi^1(\omega) \leq k$, and $\pi_1 \models_\omega \alpha$,

5. *$s \models_\omega EG^{\leq k}\alpha$ iff there exists a path π such that $\pi_0 = s$, and for all $i \geq 0$ if $\delta^i_\pi(\omega) \leq k$, then $\pi_i \models_\omega \alpha$,*
6. *$s \models_\omega E\alpha U^{\leq k}\beta$ iff there exists a path π such that $\pi_0 = s$ and for some $i \in \mathbb{N}$ it holds that $\delta^i_\pi(\omega) \leq k$ and $\pi_i \models_\omega \beta$, and $\pi_j \models_\omega \alpha$ for all $0 \leq j < i$.*

The RTCTL$_\mathrm{P}$ logic slightly differs from RTCTL presented in [1]. Firstly, we have omitted the non-superscripted modalities. It is straightforward to extend the logic with these, and to see that the standard fixpoint algorithms for *EG* and *EU* verification can be applied with no changes. Secondly, we define the semantics on ω–paths, explicitly requiring the total traversal time to grow to the infinity with the depth of the path. This is consistent with the usual requirement of progressiveness of timed systems.

3 Translation to Linear Algebra

In what follows we fix a model $M = (S, s^0, T, \rightarrow, \mathcal{L})$.

We need several simple notions concerning the sets of statements (called *linear statements*) of the form $c_1t_1 + \ldots + c_nt_n$, where $t_i \in T$ are time step parameters, $c_i \in \mathbb{N}$, and $t_i \neq t_j$ for all $1 \leq i, j \leq n$, $i \neq j$. The set of all linear statements over T is denoted by $\mathcal{LS}_T$; we omit the T subscript if it is implicitly understood. In this paper we consider only finite subsets of $\mathcal{LS}_T$.

Let $\eta = c_1t_1 + \ldots + c_nt_n$, and let $\omega \in \Omega$. We define the application of ω to η as $\eta[\omega] = c_1\omega(t_1) + \ldots + c_n\omega(t_n)$. We also define the k-bounding operation for $k \in \mathbb{N}$ as follows:

$$[\eta]_k := min(c_1, k+1)t_1 + \ldots + min(c_n, k+1)t_n.$$

To show an example, consider the statement $\eta = 6t_1 + 9t_2$ and 5-bounding $[\eta]_5 = min(6, 6)t_1 + min(9, 6)t_2 = 6t_1 + 6t_2$.

The operation of *k-bounding* has a property such that if $\approx \in \{\leq, <, >, \geq\}$, then for any $k \in \mathbb{N}$ the inequalities $\eta \approx k$ and $[\eta]_k \approx k$ have the same sets of solutions. This can be easily verified on a case-by-case basis, by noticing that if a given coefficient c_i of η exceeds $k+1$, then any nonzero value of t_i makes $\eta \approx k$ true for $\approx \in \{>, \geq\}$, while $\approx \in \{\leq, <\}$ means that only zero can be substituted for t_i.

Previous observation is crucial to the theory, as it means that every set of linear statements over the finite parameter set T, obtained by means of k-bounding with respect to some fixed natural k, is finite. We extend the $[\ ]_k$ operation to subsets $A \subseteq \mathcal{LS}$ as follows:

$$[A]_k = \{[\eta]_k \mid \eta \in A\}.$$

Let $A, B \subseteq \mathcal{LS}$, then we define $A + B = \{\eta + \mu \mid \eta \in A \text{ and } \mu \in B\}$.

Now let us consider $A \subseteq \mathcal{LS}$, $k \in \mathbb{N}$, and $\approx \in \{\leq, <, >, \geq\}$. We define $[A]_{\approx k}$ as follows:

$$[A]_{\approx k} = \bigcup_{\eta \in A} \{\omega \mid \eta[\omega] \approx k\}.$$

As to give an example, let $A = \{t_1 + 2t_2, t_3\}$, then $[A]_{<4}$ consists of all the valuations ω such that $\omega(t_1) + 2\omega(t_2) < 4$, or $\omega(t_3) < 4$.

We call the set $S \times P(\Omega)$ the *parametric state space*, and its elements are called the *parametric states*. As to give an example, consider $A \subseteq \mathcal{LS}$ such that $A = \{2t_1 + 3t_2, 2t_1 + 3t_4\}$. The pair of form $(s_0, [A]_{\leq 10})$ is a parametric state.

The last preliminary notion needed in the rest of the paper is the auxiliary operator *Flatten*. Let $B \subseteq S \times P(\Omega)$, then we define:

$$(s, A) \in Flatten(B) \text{ iff } A = \bigcup\{C \mid (s, C) \in B\},\ A \neq \emptyset.$$

To make this definition clearer, consider an example where $B = \{(s_0, C_1), (s_0, C_2), (s_1, C_3), (s_1, C_4), (s_2, C_5)\}$. In this case $Flatten(B) = \{(s_0, C_1 \cup C_2), (s_1, C_3 \cup C_4), (s_2, C_5)\}$.

If $Flatten(B) = B$, then the set B is called *flat*. If B is flat, then by $B(s)$ we denote the *parameter selector*, that is $B(s) = C$ iff $(s, C) \in B$. The parameter selector is a well defined partial function on S.

Algorithm 1 $Synthesize(M, \phi)$

```
if $\phi = p$ then
    return $A_p$
end if
if $\phi = \neg\alpha$ then
    $A_\alpha = Synthesize(M, \alpha)$
    return $\imath A_\alpha$
end if
if $\phi = \alpha \wedge \beta$ then
    $A_\alpha = Synthesize(M, \alpha)$
    $A_\beta = Synthesize(M, \beta)$
    return $A_\alpha * A_\beta$
end if
if $\phi = EX^{\leq k}\alpha$ then
    $A_\alpha = Synthesize(M, \alpha)$
    return $\mathcal{EX}^{\leq k} A_\alpha$
end if
if $\phi = EG^{\leq k}\alpha$ then
    $A_\alpha = Synthesize(M, \alpha)$
    return $\mathcal{EG}^{\leq k} A_\alpha$
end if
if $\phi = E\alpha U^{\leq k}\beta$ then
    $A_\alpha = Synthesize(M, \alpha)$
    $A_\beta = Synthesize(M, \beta)$
    return $\mathcal{E} A_\alpha \mathcal{U}^{\leq k} A_\beta$
end if
```

3.1 The translation

Our aim is to find all the valuations under which a given formula $\phi \in \mathrm{RTCTL_P}$ holds in a model M. In our solution we augment each state s with the set $A_\phi(s)$ of parameter valuations such that $s \models_\omega \phi$ iff $\omega \in A_\phi(s)$. This is done recursively in Algorithm 1, with respect to the formula structure. For each s the set $A_\phi(s)$ can be represented as a finite union of solution sets of a finite number of linear (integer) inequalities. This means that $A_\phi(s)$ has a finite representation for each s, and for this reason we call the method a translation from $\mathrm{RTCTL_P}$ parametric model checking to linear algebraic problem.

Let $p \in \mathcal{PV}$, then $A_p = \{(s, \Omega) \mid p \in \mathcal{L}(s)\}$ is the set of such pairs (s, Ω) that $p \in \mathcal{L}(s)$. Intuitively, A_p contains the pairs consisting of a state in which p holds, together with the full set Ω; this expresses the lack of restrictions on the parameter values. Obviously, A_p is flat.

In the algorithm we use several new operators that are counterparts of propositional connectives and $\mathrm{RTCTL_P}$ modalities:

1. operator $*$ – a counterpart of $\wedge$,
2. operator $\imath$ – related to $\neg$,
3. operator $\mathcal{EX}^{\leq k}$ – a counterpart of $EX^{\leq k}$,
4. operator $\mathcal{EG}^{\leq k}$ – a counterpart of $EG^{\leq k}$.
5. operator $\mathcal{EU}^{\leq k}$ – related to $EU^{\leq k}$.

The detailed description of these notions is a subject of the rest of this section, starting with the $*$ operator.

Definition 4. *Let A, B be two flat subsets of $S \times P(\Omega)$. Define:*

$$A * B = \{(s, C \cap C') \mid (s, C) \in A, \text{ and } (s, C') \in B\}.$$

The next corollary follows immediately from the above definition.

Corollary 1. *Let ϕ, ψ be $\mathrm{RTCTL_P}$ formulae, and A_ϕ, A_ψ be such flat subsets of the parametric state space that $s \models_\omega \phi$ iff $\omega \in A_\phi(s)$ and $s \models_\omega \psi$ iff $\omega \in A_\psi(s)$ for all $s \in S$. Then $s \models_\omega \phi \wedge \psi$ iff $\omega \in (A_\phi * A_\psi)(s)$.*

It should be noted that in our applications, the $*$ operation is purely symbolic, as we deal with the sets of inequalities only.

Example 1. Consider the following sets:

$$A_\phi = \{(s_0, \Omega), (s_1, \{\omega \mid \omega(t_1) + 3\omega(t_2) < 5\})\},$$
$$A_\psi = \{(s_1, \{\omega \mid 2\omega(t_1) + 3\omega(t_3) < 4\})\},$$

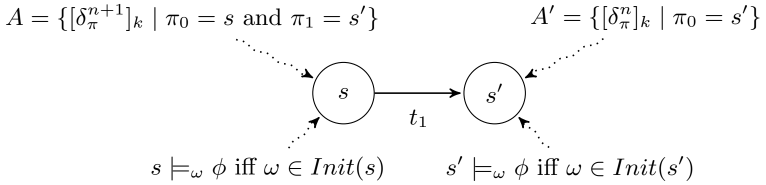
We have $A_\phi * A_\psi = \{(s_1, \{\omega \mid \omega(t_1) + 3\omega(t_2) < 5 \ \wedge \ 2\omega(t_1) + 3\omega(t_3) < 4\})\}$.

In the translation of $EG^{\leq k}$ and $EU^{\leq k}$ we make use of the *bounded backstep operation.* This operation is defined on sets of triples (s, A, C), where s is a state, A is a set of linear statements used to track possible constraints on parameters, and C is a set of parameter valuations used to track the allowed values of time step parameters.

Definition 5. *Let $D \subseteq S \times P(\mathcal{LS}) \times P(\Omega)$, $k \in \mathbb{N}$, and $Init$ be a flat subset of $S \times P(\Omega)$ such that for each $e \in D$ there is $f \in Init$ satisfying $e|_1 = f|_1$. Now, $(s, A, C) \in BackStep_k(D, Init)$ iff:*

1. *there exists $e \in D$ such that $e|_1 = s$,*
2. *for some $A' \subseteq \mathcal{LS}$, $C' \subseteq \Omega$, and $s' \in S$ there exists $(s', A', C') \in D$, such that:*
   - *(a) the set $link(s, s')$ of time step parameters (treated as linear statements) is nonempty (i.e. there is a transition from s to s'),*
   - *(b) $A = [link(s, s') + A']_k$,*
   - *(c) $C = C' \cap Init(s)$.*

While the bounded backstep operation may seem involved, it originates from a natural idea. Let ϕ be some property and let $Init$ be such a set that $s \models_\omega \phi$ iff $\omega \in Init(s)$ for each state s. Let $D \subseteq S \times P(\mathcal{LS}) \times P(\Omega)$ and $(s', A', C') \in D$.


$A = \{[\delta_\pi^{n+1}]_k \mid \pi_0 = s \text{ and } \pi_1 = s'\}$
$A' = \{[\delta_\pi^{n}]_k \mid \pi_0 = s'\}$
s
s'
t_1
$s \models_\omega \phi$ iff $\omega \in Init(s)$
$s' \models_\omega \phi$ iff $\omega \in Init(s')$

Assume that $C' = Init(s')$, let $n \in \mathbb{N}$, and A' be the set of k–bounded time distance functions for all paths leaving s' and measuring the distance up to the n–th position. It is easy to see, that $BackStep_k(D, Init)$ contains a tuple $(s, A, Init(s) \cap Init(s'))$, where $A = [link(s, s') + A']_k$. The set A consists of k–bounded time distance functions for all paths leaving s, entering s' in the next step, and measuring the distance up to the $(n + 1)$–th position. The set $Init(s) \cap Init(s')$ contains such parameter valuations ω that $s \models_\omega \phi$ and $s' \models_\omega \phi$.

Example 2. Consider the sets:

$$C_1 = \{\omega \mid \omega(t_1) > 2\},\ C_2 = \{\omega \mid \omega(t_2) + \omega(t_3) \leq 4\},$$
$$D = \{(s_1, \{6t_1 + 8t_2\}, C_1), (s_2, \{4t_2 + 7t_3, t_4\}, C_2)\},$$

and assume that the only transitions involving s_1 and s_2 are (s_1, t_1, s_2), (s_1, t_2, s_2), and let $Init = \{(s_1, C_1), (s_2, C_2)\}$. Let us compute $BackStep_5(D, Init)$. We can see that $link(s_1, s_2) = \{t_1, t_2\}$, $link(s_1, s_1) = link(s_2, s_2) = link(s_2, s_1) = \emptyset$. Let $A = [\{t_1, t_2\} + \{4t_2 + 7t_3, t_4\}]_5 = \{t_1 + 4t_2 + 6t_3, 5t_2 + 6t_3, t_1 + t_4, t_2 + t_4\}$, and $C = C_2 \cap Init(s_1) = C_2 \cap C_1 = \{\omega \mid \omega(t_1) > 2 \text{ and } \omega(t_2) + \omega(t_3) \leq 4\}$. In this case $BackStep_5(D, Init) = \{(s_1, A, C)\}$.

We say that a sequence of sets $H_0, H_1, \ldots$ *stabilizes* if there exists $i \geq 0$ such that $H_j = H_i$ for all $j > i$, and denote this as $lim_{j \to \infty} H_j = H_i$.

Let D be a finite subset of $S \times P(\mathcal{LS}) \times P(\Omega)$. Notice that if we fix some $k \in \mathbb{N}$ and $Init$, then the sequence defined by $H_0 = D$, and $H_{i+1} = H_i \cup BackStep_k(H_i, Init)$ stabilizes. This is due to the fact that there is a finite number of time parameters in a model (therefore a finite number of k-bounded expressions built with respect to $[\ ]_k$), and a finite number of parameter valuation sets in D.

Let $(s, A, C) \in S \times P(\mathcal{LS}) \times P(\Omega)$, $\approx \in \{\leq, <, >, \geq\}$, and $k \in \mathbb{N}$. Denote $[(s, A, C)]_{\approx k} = (s, [A]_{\approx k} \cap C)$. Intuitively, this encodes a state together with those parameter valuations which satisfy constraints present in $[A]_{\approx k}$ (the path length constraints), and in C (the initial constraints). We extend this notion to the space on which *BackStep* operates, by putting $[D]_{\approx k} = \{[(s, A, C)]_{\approx k} \mid (s, A, C) \in D\}$ for any $D \subseteq S \times P(\mathcal{LS}) \times P(\Omega)$.

Let us move to the first application of $BackStep_k$ operation, i.e., the translation of $EG^{\leq k}$. The following example provides some intuitions behind the parametric counterpart of this modality.

Example 3. Consider model shown in Fig. 1, where $\mathcal{L}(s_0) = \mathcal{L}(s_1) = \{p\}$, and formula $EG^{\leq 2}p$. For the simplicity, the loops on states s_2, s_3 are unlabeled.

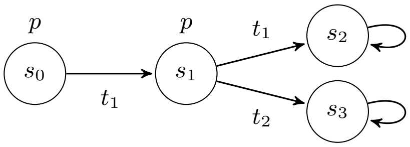


Fig. 1: A simple model

It is easy to see that $s_1 \models_\omega EG^{\leq 2}p$ iff $\omega(t_1) > 2$ or $\omega(t_2) > 2$, i.e., using the newly introduced notation, $\omega \in [out(s_1)]_{>2}$. It also holds that $s_0 \models_\omega EG^{\leq 2}p$ if $\omega \in [out(s_0)]_{>2}$, but this is not an exhaustive description of all such parameter valuations. Indeed, $s_0 \models_\omega EG^{\leq 2}p$ also if $2\omega(t_1) > 2$ or $\omega(t_1) + \omega(t_2) > 2$, i.e., $\omega \in [t_1 + out(s_1)]_{>2}$. By a straightforward case-by-case analysis we can check that $s_0 \models_\omega EG^{\leq 2}p$ iff $\omega \in [out(s_0)]_{>2} \cup [t_1 + out(s_1)]_{>2}$.

Definition 6. *Let A be a flat subset of $S \times P(\Omega)$ and $k \in \mathbb{N}$. Define:*

$$G_0(A) = \{(s, out(s), A(s)) \mid \text{there exists } e \in A \text{ such that } e|_1 = s\},$$
$$G_{j+1}(A) = BackStep_k(G_j(A), A).$$

We define $\mathcal{EG}^{\leq k}A = Flatten(\bigcup_{j=0}^{\infty}[G_j(A)]_{>k})$.

The *Flatten* operator is used only in order to obtain the result in a less complex form, where for each state s there exists at most one $e \in \mathcal{EG}^{\leq k}A$ such that $e|_1 = s$.

Theorem 1. *Let ϕ be a formula of* RTCTL$_\mathrm{P}$*, and A_ϕ be such a flat subset of $S \times P(\Omega)$ that $s \models_\omega \phi$* iff *$\omega \in A_\phi(s)$. For any state $s \in S$, $k \in \mathbb{N}$, and a parameter valuation ω we have $s \models_\omega EG^{\leq k}\phi$ iff $\omega \in (\mathcal{EG}^{\leq k}A_\phi)(s)$.*

Proof. If $s \models_\omega EG^{\leq k}\phi$, then there exists a path $\pi = (s_0, t_0, s_1, t_1, \ldots)$, such that for some $n \in \mathbb{N}$ it holds that $\pi_0 = s$, $\delta_\pi^{n+1}(\omega) > k$ and $\delta_\pi^i(\omega) \leq k$ for all

$0 \leq i \leq n$, and $\pi_i \models_\omega \phi$ for all $0 \leq i \leq n$.

$$\pi = \overbrace{\underbrace{s_0 \xrightarrow{t_0} s_1 \xrightarrow{t_1} s_2 \xrightarrow{t_2} \ldots \xrightarrow{t_{n-1}}}_{\delta_\pi^n(\omega) \leq k} s_n \xrightarrow{t_n}}^{\delta_\pi^{n+1}(\omega) > k} s_{n+1} \xrightarrow{t_{n+1}} \ldots$$

For each $0 \leq i \leq n$ we have that $\pi_i \models_\omega \phi$, therefore $A_\phi(s_i)$ is well defined for each $0 \leq i \leq n$, and $\omega \in \bigcap_{i=0}^n A_\phi(s_i)$. It is easy to see that $(s_n, out(s_n), A_\phi(s_n)) \in G_0(A_\phi)$, and $t_n \in out(s_n)$. Notice that $s_{n-1} \xrightarrow{t_{n-1}} s_n$, thus $(s_{n-1}, [link(s_{n-1}, s_n) + out(s_n)]_k,\ A_\phi(s_{n-1}) \cap A_\phi(s_n)) \in BackStep_k(G_0(A_\phi), A_\phi) = G_1(A_\phi)$. Again, we have that $[t_{n-1} + t_n]_k \in [link(s_{n-1}, s_n) + out(s_n)]_k$. After $n+1$ such inductive steps we obtain that there is a tuple $(s_0, A, \bigcap_{i=0}^n A_\phi(s_i)) \in G_n(A_\phi)$ such that $[t_0 + t_1 + \ldots + t_n]_k \in A$, and $\omega \in \bigcap_{i=0}^n A_\phi(s_i)$. Recall that $\delta_\pi^n = t_0 + t_1 + \ldots + t_n$, and as $\delta_\pi^n(\omega) > k$, we have that $[t_0 + t_1 + \ldots + t_n]_k(\omega) > k$, therefore $\omega \in [A]_{>k}$. This means that $\omega \in [A]_{>k} \cap \bigcap_{i=0}^n A_\phi(s_i)$, which in view of the fact that $[(s, A, \bigcap_{i=0}^n A_\phi(s_i))]_{>k} \in [G_n(A_\phi)]_{>k}$ concludes this part of the proof.

Now let $\omega \in (\mathcal{EG}^{\leq k} A_\phi)(s)$. This means that for some $m \in \mathbb{N}$, and $e_m = (s_m, B_m)$, where $s_m = s$ we have that $e_m \in [G_m(A_\phi)]_{>k}$, and $\omega \in B_m$. This in turn means that there is a sequence $(s_0, A_0, C_0), (s_1, A_1, C_1), \ldots, (s_m, A_m, C_m)$ such that:

1. $A_i = [link(s_i, s_{i-1}) + A_{i-1}]_k$ for all $0 < i \leq m$, and $A_0 = out(s_0)$,
2. $C_i = \bigcap_{j=0}^i A_\phi(s_j)$ and $\omega \in C_i$ for all $0 \leq i \leq m$,
3. $(s_i, A_i, C_i) \in G_i(A_\phi)$ for all $0 \leq i \leq m$,
4. $[A_n]_{>k} \cap C_m = B_m$.

From the above points it follows that there exists such a finite sequence $\pi' = (s_m, t_m, s_{m-1}, t_{m-1} \ldots, s_0, t_0)$ that $[\delta_{\pi'}^m]_k = [t_m + t_{m-1} + \ldots + t_0]_k \in A_m$, and $[\delta_{\pi'}^m]_k(\omega) > k$. Notice that the latter is equivalent to $\delta_{\pi'}^m(\omega) > k$, and that the second point implies that $s_i \models_\omega \phi$ for all $0 \leq i \leq m$. The sequence π' is a prefix of some infinite path π (due to the totality of the transition relation), such that $\pi_i \models_\omega \phi$ for all $0 \leq i \leq m$, and $\delta_\pi^m(\omega) > k$. This means that $s \models_\omega EG^{\leq k}\phi$, which concludes the proof. □

Definition 7. *Let A, B be two flat subsets of $S \times P(\Omega)$ and $k \in \mathbb{N}$. Denote:*

$$\begin{aligned} H_0(A, B) = \{&(s, link(s, s'), A(s) \cap B(s')) \mid \text{there exists } e \in B,\ e|_1 = s', \\ &\text{and } link(s, s') \neq \emptyset\}, \\ H_{i+1}(A, B) = &\ BackStep_k(H_i(A, B), A). \end{aligned}$$

We define $\mathcal{E}A\mathcal{U}^{\leq k}B = Flatten((\bigcup_{i=0}^\infty [H_i(A, B)]_{\leq k}) \cup B)$.

Again, the *Flatten* operator is used only for the convenience, and the sequence $(\bigcup_{i=0}^j H_i)_{j \geq 0}$ is guaranteed to stabilize.

Theorem 2. *Let ϕ, ψ be* $\mathrm{RTCTL_P}$ *formulae, and A_ϕ, A_ψ be such flat subsets of parametric state space that $s \models_\omega \phi$* iff *$\omega \in A_\phi(s)$ and $s \models_\omega \psi$* iff *$\omega \in A_\psi(s)$, for each state s. For any state s, any $k \in \mathbb{N}$, and parameter valuation ω it holds that $s \models_\omega E\phi U^{\leq k}\psi$ iff $\omega \in (\mathcal{E}A_\phi \mathcal{U}^{\leq k} A_\psi)(s)$.*

Proof. Assume that $s \models_\omega E\phi U^{\leq k}\psi$. This means that there exists a sequence $\pi = (s_0, t_0, s_1, t_1, \ldots, s_n, t_n, \ldots)$ such that $\pi_0 = s$, for some $n \geq 0$ we have $\delta^n_\pi(\omega) \leq k$, $\pi_n \models_\omega \psi$, and $\pi_i \models_\omega \phi$ for all $0 \leq i < n$. If $n = 0$, then $s \models_\omega \psi$, therefore $\omega \in A_\psi(s)$; now it suffices to notice that A_ψ is a (flattened) subset of $\mathcal{E}A_\phi \mathcal{U}^{\leq k} A_\psi$. We can therefore assume that $n > 0$, which means that $s_{n-1} \models_\omega \phi$, and $s_n \models_\omega \psi$, thus $\omega \in A_\phi(s_{n-1}) \cap A_\psi(s_n)$. As $t_{n-1} \in link(s_{n-1}, s_n)$, we obtain that $\big(s_{n-1}, link(s_{n-1}, s_n), (A_\phi(s_{n-1}) \cap A_\psi(s_n))\big) \in H_0(A_\phi, A_\psi)$. Similarly as in a first part of the proof of Theorem 1 we can now create a sequence $(s_0, A_0, C_0), (s_1, A_1, C_1), \ldots, (s_{n-1}, A_{n-1}, C_{n-1})$ such that for all $0 \leq i \leq n-1$:

1. $A_i = [link(s_i, s_{i+1}) + link(s_{i+1}, s_{i+2}) + \ldots + link(s_{n-1}, s_n)]_k$,
2. $C_i = \bigcap_{j=i}^{n-1} A_\phi(s_j) \cap A_\psi(s_n)$ and $\omega \in C_i$,
3. $(s_i, A_i, C_i) \in H_{n-i-1}(A_\phi, A_\psi)$.

Now let us notice that $[t_0 + t_1 + \ldots + t_{n-1}]_k \in A_0$, and as $\delta^n_\pi(\omega) \leq k$, also $[t_0 + t_1 + \ldots + t_{n-1}]_k(\omega) \leq k$. This means that $\omega \in [A_0]_{\leq k} \cap C_0$, therefore there is $e \in [H_0(A_\phi, A_\psi)]_{\leq k}$ such that $e|_1 = s_0 = s$, and $\omega \in e|_2$, which concludes the case.

Now let us assume that $\omega \in (\mathcal{E}A_\phi \mathcal{U}^{\leq k} A_\psi)(s)$. If $\omega \in A_\psi(s)$, then obviously $s \models_\omega \psi$ and $s \models_\omega E\phi U^{\leq k}\psi$, therefore let us assume that for some $m \in \mathbb{N}$ we have that $e = (s_m, B_m) \in [H_m(A_\phi), A_\psi]_{\leq k}$ where $s_m = s$, and $\omega \in B_n$. Again, this means that there exist a state s' such that $\omega \in A_\psi(s')$, and a sequence $(s_0, A_0, C_0), (s_1, A_1, C_1), \ldots, (s_m, A_m, C_m)$ such that:

1. $link(s_{i+1}, s_i) \neq \emptyset$ for all $0 \leq i < m$, and $link(s_0, s') \neq \emptyset$,
2. $A_i = [link(s_i, s_{i-1}) + link(s_{i-1}, s_{i-2}) + \ldots + link(s_0, s')]_k$ for all $0 \leq i \leq m$,
3. $C_i = \bigcap_{j=0}^{i} A_\phi(s_j) \cap A_\psi(s')$ and $\omega \in C_i$ for all $0 \leq i \leq m$,
4. $(s_i, A_i, C_i) \in H_i(A_\phi, A_\psi)$ for all $0 \leq i \leq m$,
5. $[A_m]_{\leq k} \cap C_m = B_m$.

From the above points we can infer the existence of such a finite sequence $\pi' = (s_m, t_m, s_{m-1}, t_{m-1}, \ldots, s_0, t_0, s', t')$ (the t' is an arbitrary time step parameter from $out(s')$) that:

1. $t_i \in link(s_i, s_{i-1})$ for all $0 < i \leq m$, and $t' \in link(s_0, s')$,
2. $\pi'(i) \models_\omega \phi$ for all $0 \leq i \leq m$, and $\pi'(m+1) \models_\omega \psi$,
3. $\delta^m_{\pi'}(\omega) \leq k$, as $[\delta^m_{\pi'}]_k(\omega) = [t_0 + t_1 + \ldots + t_m]_k(\omega) \leq k$.

By the virtue of the totality of the transition relation this means that $s \models_\omega E\phi U^{\leq k}\psi$, which concludes the proof. □

Definition 8. *Let A be a flat subset of $S \times P(\Omega)$, and $k \in \mathbb{N}$. Denote:*

$$I_k(A) = \{(s, link(s, s'), A(s')) \mid \text{exists } e \in A \text{ s. t. } e|_1 = s' \text{ and } link(s, s') \neq \emptyset\}.$$

We define $\mathcal{E}\mathcal{X}^{\leq k} A = Flatten([I_k(A)]_{\leq k})$.

Intuitively, in $I_k(A)$ for each state s we gather its connections with other states s' and constraints $A(s')$ imposed in s'. It suffices to ensure that these constraints are consistent with conditions of transition from s to s' in under k time units.

Corollary 2. *Let ϕ be a formula of* RTCTL$_\mathrm{P}$*, let $k \in \mathbb{N}$, and let A_ϕ be such a flat subset of $S \times P(\Omega)$ that $s \models_\omega \phi$* iff *$\omega \in A_\phi(s)$. For any state s and parameter valuation ω we have $s \models_\omega EX^{\leq k}\phi$ iff $\omega \in (\mathcal{EX}^{\leq k} A_\phi)(s)$.*

We have proved that the proposed translation is valid for all nonnegated expression. To complete the theory we show how to deal with negations.

Definition 9. *Let A be a flat subset of $S \times P(\Omega)$. We define:*

$$\imath A = Flatten(\{(s, \Omega \setminus A(s)) \mid \text{exists } e \in A \text{ such that } e|_1 = s\}$$
$$\cup\{(s, \Omega) \mid \text{there is no } e \in A \text{ such that } e|_1 = s\}).$$

Let us present some intuitions concerning the translation of the negation. Let A_ϕ characterize the states augmented with parameter valuations under which the ϕ property holds. The $\imath A_\phi$ set is built by:

1. augmenting any state s represented in A_ϕ, by those valuations under which ϕ does not hold (the complement of $A_\phi(s)$),
2. including all the states which are not represented in A_ϕ together with the full set of parameter valuations.

This gives rise to the following corollary.

Corollary 3. *Let A_ϕ be such a flat subset of $S \times P(\Omega)$ that $s \models_\omega \phi$* iff *$\omega \in A_\phi(s)$. For any state s and $\omega \in \Omega$ it holds that $s \models_\omega \neg\phi$ iff $\omega \in (\imath A_\phi)(s)$.*

4 Conclusions

The method presented in this paper allows for the synthesis of parameter values in timed Kripke structures for properties expressed in RTCTL$_\mathrm{P}$ logic. To be more precise, for a given property ϕ the result of synthesis is the set A_ϕ of constraints on time step parameters. These constraints are expressed as linear inequalities over natural numbers, therefore our method is in fact a translation from the problem of RTCTL$_\mathrm{P}$ parameter synthesis to a problem stated in the language of linear algebra. If properly implemented, this enables to take advantage of the vast work and available tools from the discrete optimization field.

It is rather straightforward to show that for a given RTCTL$_\mathrm{P}$ formula ϕ it suffices to consider only the parameter step values which do not exceed the greatest superscript in ϕ plus 1. While Ω can be limited to a finite set, an enumerative verification of all possible valuations from this set would soon prove to be intractable. A symbolic model checking approach gives a chance of alleviating these limitations via an efficient representation of statespace and operations on its subsets. We plan to research the possibilities of implementing the presented work using various versions of decision diagrams and SMT-theories.

Acknowledgements Michał Knapik is supported by the Foundation for Polish Science under International PhD Projects in Intelligent Computing. Project financed from the European Union within the Innovative Economy Operational Programme 2007-2013 and European Regional Development Fund.

References

1. Emerson, E.A., Trefler, R.: Parametric quantitative temporal reasoning. In: Proc. of the 14th Symp. on Logic in Computer Science (LICS'99), IEEE Computer Society (July 1999) 336–343
2. Alur, R., Henzinger, T., Vardi, M.: Parametric real-time reasoning. In: Proc. of the 25th Ann. Symp. on Theory of Computing (STOC'93), ACM (1993) 592–601
3. Doyen, L.: Robust parametric reachability for timed automata. Inf. Process. Lett. **102** (May 2007) 208–213
4. Tranouez, L.M., Lime, D., Roux, O.H.: Parametric model checking of time Petri nets with stopwatches using the state-class graph. In: Proc. of the 6th Int. Workshop on Formal Analysis and Modeling of Timed Systems (FORMATS'08). Volume 5215 of LNCS., Springer-Verlag (2008) 280–294
5. Hune, T., Romijn, J., Stoelinga, M., Vaandrager, F.: Linear parametric model checking of timed automata. In: Proc. of the 7th Int. Conf. on Tools and Algorithms for the Construction and Analysis of Systems (TACAS'01). Volume 2031 of LNCS., Springer-Verlag (2001) 189–203
6. Knapik, M., Penczek, W.: Bounded model checking for parametric timed automata. T. Petri Nets and Other Models of Concurrency **5** (2012) 141–159
7. Jovanović, A., Lime, D., Roux, O.H.: Integer parameter synthesis for timed automata. In: Proceedings of the 19th international conference on Tools and Algorithms for the Construction and Analysis of Systems. TACAS'13, Berlin, Heidelberg, Springer-Verlag (2013) 401–415
8. André, E., Chatain, T., Encrenaz, E., Fribourg, L.: An inverse method for parametric timed automata. International Journal of Foundations of Computer Science **20**(5) (Oct 2009) 819–836
9. Henzinger, T., Ho, P., Wong-Toi, H.: HyTech: A model checker for hybrid systems. In: Proc. of the 9th Int. Conf. on Computer Aided Verification (CAV'97). Volume 1254 of LNCS., Springer-Verlag (1997) 460–463