Fundamenta Informaticae XX (2014) 1–16
IOS Press

Parameter Synthesis for Timed Kripke Structures*

Michał Knapik
Institute of Computer Science, PAS, Warsaw, Poland
mknapik@ipipan.waw.pl

Wojciech Penczek
Institute of Computer Science, PAS, Warsaw, Poland and
University of Natural Sciences and Humanities, II, Siedlce, Poland
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 $\mathrm{RTCTL_P}$, holds in a parametric timed Kripke structure. We prove the decidability of parameter synthesis for $\mathrm{RTCTL_P}$ by showing how to restrict the infinite space of parameter valuations to its finite subset and employ a brute-force algorithm. The brute-force approach soon becomes intractable, therefore we propose a symbolic algorithm for $\mathrm{RTCTL_P}$ parameter synthesis. Similarly to the fixed-point symbolic model checking approach, we introduce special operators which stabilise on the solution. The process of stabilisation is essentially a translation from the $\mathrm{RTCTL_P}$ parameter synthesis problem to a discrete optimization task. We show that the proposed method is sound and complete and provide some complexity results. We argue that this approach 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 flaws found 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

*This work is partially funded by DEC-2012/07/N/ST6/03426 NCN Preludium 4 grant

consisting of a model and a formula is an 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 a 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 each transition is augmented with an additional label specifying how long it takes to traverse it. The input logic is an extension of Computation Tree Logic, denoted $\mathrm{RTCTL_P}$, which allows for expressing properties over restricted fragments of the paths. To be more precise, for a model M and a formula $\phi \in \mathrm{RTCTL_P}$ we describe all the parameter valuations ω under which ϕ holds in M. We first show that the space of the parameter valuations considered can be limited to its finite subset. Therefore, the synthesis problem for $\mathrm{RTCTL_P}$ can be solved using the exhaustive brute-force approach. This, however, may become soon intractable, so we propose a symbolic framework that can alleviate the problem of the exponential blowup in some cases. We also provide preliminary complexity results for both the brute-force and the symbolic approach.

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 [ET99]. As we show, the problem of parameter synthesis is decidable for $\mathrm{RTCTL_P}$. It is however not decidable, even, for as simple properties as reachability for many other models, e.g., parametric timed automata (PTA) [AHV93, Doy07] and bounded parametric time Petri nets [TLR08]. Difference bound matrix-based semi-algorithms for reachability were extended to the PTA case in [HRSV01] and implemented in UPPAAL-PMC. In [KP12] we showed how to synthesise by means of bounded model checking a part of the set of the valuations for PTA reachability. The problem of synthesis of bounded integer valuations for PTA is analysed in [JLR13] and shown to be in PSPACE. In [ACEF09] 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 the IMITATOR tool [AFKS12]. A parametric analysis is also possible with HyTech [HHWT97] by means of hybrid automata.

In the next section we introduce the $\mathrm{RTCTL_P}$ logic and its models. In Section 3 we show that the synthesis problem for $\mathrm{RTCTL_P}$ is decidable by limiting the space of the parameter valuations to its finite subset and applying a brute-force approach. In Section 4 we show how to solve the synthesis problem via a translation to sets of linear inequalities over natural numbers and provide some remarks on the complexity of the solution. We conclude the work with a comment on the possible benefits and downsides of our approach and some 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 its i–th element.

2.1. The Syntax of $\mathrm{RTCTL_P}$

The Real Time CTL [ET99] allows to express branching-time temporal properties involving the integer time-step depth of considered paths.

Definition 2.1. (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 $\neg\alpha$ and $\alpha \wedge \beta$,
3. if α and β are formulae, then so are $EX^{\leq k}\alpha$, $EG^{\leq k}\alpha$, and $E\alpha U^{\leq k}\beta$ for $k \in \mathbb{N}$.

As to give an example of the meaning of an $\mathrm{RTCTL_P}$ 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 $\mathrm{RTCTL_P}$

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

Definition 2.2. 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 the following condition: $true \in \mathcal{L}(s)$ 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 $in(s)$, $out(s)$, and $link(s, s')$ 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 $\omega : T \to \mathbb{N}$ 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. Let $\Pi(\mathrm{M}, s)$ denote the set of all the sequences π such that $\pi_0 = s$, and let $\Pi(\mathrm{M})$ be the set of all the sequences in M; we omit the model symbol whenever it is known from the context. We define the *time distance function* between the positions π_0 and π_j on a sequence π as $\delta^j_\pi = \sum_{i=0}^{j-1} t_i$, and we assume that $\delta^0_\pi = 0$. If ω is a parameter valuation, then let $\delta^j_\pi(\omega) = \sum_{i=0}^{j-1} \omega(t_i)$. For simplicity, we use the same symbol to denote the formal sums and the arithmetic sums. This should not lead to ambiguity as the case is always easy to identify from its context.

Definition 2.3. (Semantics of $\mathrm{RTCTL_P}$)
Let $\mathrm{M} = (S, s^0, T, \to, \mathcal{L})$ be a model and $s \in S$. Let $\alpha, \beta \in \mathrm{RTCTL_P}$, $\omega \in \Omega$ 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. $\pi_0 = s, \delta^1_\pi(\omega) \leq k,$ and $\pi_1 \models_\omega \alpha$,
5. $s \models_\omega EG^{\leq k}\alpha$ iff there exists a path π s.t. $\pi_0 = s$ and $\lim_{j\to\infty} \delta^j_\pi(\omega) > k,$ 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 $\mathrm{RTCTL_P}$ logic slightly differs from RTCTL presented in [ET99]. 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, in the semantics of $EG^{\leq k}$ we explicitly assume that the time along the path exceeds k. The reason for this is that while the proposed algorithms are based on computing fixed-points, they are easy to adapt to the bounded model checking framework for the existential fragment of $\mathrm{RTCTL_P}$: it suffices to replace ∞ with a chosen depth of the unfolding in the infinite sums in Definition 4.3 and 4.4. We do not expand upon this here, as this is outside of the scope of the paper. However, for the completeness of the discourse let us briefly show how the methods presented in this work can be easily extended to deal with the original RTCTL. Let $\models^L_\omega$ denote the satisfiability in a selected logic $L \in \{\mathrm{RTCTL}, \mathrm{RTCTL_P}\}$, and let $\alpha \in \mathrm{RTCTL_P}$, $\omega \in \Omega$, and $s \in S$. We introduce a new $EG^{=0}$ modality such that $s \models_\omega^{\mathrm{RTCTL_P}} EG^{=0}\alpha$ iff there exists a path π such that $\pi_0 = s$ and $\lim_{j\to\infty} \delta^j_\pi(\omega) = 0$, and $\pi_i \models_\omega \alpha$ for all $i \geq 0$. It is easy to see that [ET99] $s \models_\omega^{\mathrm{RTCTL}} EG^{\leq k}\alpha \iff s \models_\omega^{\mathrm{RTCTL_P}} EG^{\leq k}\alpha \vee E\alpha U^{\leq k} EG^{=0}\alpha$, and that the new modality satisfies $s \models_\omega^{\mathrm{RTCTL_P}} EG^{=0}\alpha \iff s \models_\omega^{\mathrm{RTCTL_P}} \alpha \wedge EX^{\leq 0} EG^{=0}\alpha$. The last equality can be used as a basis for a simple fixed-point algorithm, similarly as it is done for CTL.

Algorithm 1 $BruteForceSynthesize(M, \phi)$

Input: a model $M = (S, s^0, T, \to, \mathcal{L})$, $\phi \in \mathrm{RTCTL_P}$

```
1: k := greatest superscript in φ
2: for ω ∈ {0, ..., k+1}^T do
3:   if M, s^0 ⊨_ω φ {a call to non-parametric checker} then
4:     for t ∈ T do
5:       if ω(t) < k+1 then ω_R(t) := ω(t)
6:       else ω_R(t) := ∗ {any value greater than k}
7:     end for
8:     print(ω_R)
9:   end if
10: end for
```

3. Brute-force Algorithm

Under a fixed parameter valuation $\omega \in \Omega$ the model M becomes non-parametric, and the question whether $M, s^0 \models_\omega \phi$ becomes a decidable decision problem with the binary (*yes/no*) answer. The naïve brute-force approach to parameter synthesis would consist of the exhaustive iteration through all the parameter valuations, performing non-parametric checks, and collecting the answers. Such a straighforward procedure is not possible however, due to the infiniteness of Ω. In this section we show that for a given model M and $\phi \in \mathrm{RTCTL_P}$ it suffices to check only a finite subspace of Ω to perform the full description of all $\omega \in \Omega$ such that $M, s^0 \models_\omega \phi$. This allows us to present the modified brute-force algorithm for $\mathrm{RTCTL_P}$ synthesis, thus showing that the problem is decidable. For each $\phi \in \mathrm{RTCTL_P}$ let $|\phi|$ denote the length of the formula ϕ. Let $k \in \mathbb{N}$ and $\omega, \omega' \in \Omega$. Define the relation $\sim_k \subseteq \Omega \times \Omega$ such that $\omega \sim_k \omega'$ iff $\forall_{t \in T}[(\omega(t) > k \iff \omega'(t) > k) \wedge (\omega(t) \leq k \implies \omega'(t) = \omega(t))]$, i.e., the valuations agree on the parameters for which k is not exceeded.

Lemma 3.1. Let $M = (S, s^0, T, \to, \mathcal{L})$ be a model, $\phi \in \mathrm{RTCTL_P}$ contain at least one temporal modality, $k \in \mathbb{N}$ be the greatest superscript appearing in ϕ, and $\omega, \omega' \in \Omega$. If $\omega \sim_k \omega'$ then $(s \models_\omega \phi$ iff $s \models_{\omega'} \phi)$ for all $s \in S$.

Proof: The proof follows by the induction on the structure of ϕ. The case when ϕ does not contain any temporal modalities is obvious. The cases of $\phi = \alpha \wedge \beta$ and $\phi = \neg\alpha$ follow easily from the definition and the inductive assumption. Let $l \leq k$ and observe $(\star)$ that $\forall_{i \in \mathbb{N}}(\delta^i_\pi(\omega) > l \iff \delta^i_\pi(\omega') > l)$. The cases of $\phi \in \{EG^{\leq l}\alpha, E\alpha U^{\leq l}\beta\}$ follow immediately from $(\star)$ and the inductive assumption. □

The above lemma can be applied to limit the space of the parameter valuations that need to be checked in order to obtain the solution to the parameter synthesis problem by means of the brute-force approach. This is exploited in Algorithm 1.

Theorem 3.1. Let $M = (S, s^0, T, \to, \mathcal{L})$ be a model, $\phi \in \mathrm{RTCTL_P}$, and k denote the greatest superscript appearing in ϕ. The brute-force enumerative Algorithm 1 prints out the representations of all the abstraction classes of the relation $\sim_k$. Its time complexity is $O((|S| + |\to|)^2 |\phi| \times (k+2)^{|T|})$.

Proof: It follows from Lemma 3.1 that it suffices to check only those $(k+2)^{|T|}$ valuations that do not assign to any of the parameters the values greater that $k+1$. Under a parameter valuation, the model M becomes a non-parametric timed Kripke structure for which the problem of $\mathrm{RTCTL_P}$ verification can be solved in $O((|S|+|\rightarrow|)^2|\phi|)$ [ET99]. In Algorithm 1 for each $\omega \in \Omega$ such that $M, s^0 \models_\omega \phi$ the enhanced valuation ω_R is a representation of the abstraction class $[\omega]_{\sim_k}$ with the values greater than k replaced by the symbol $*$. □

4. Symbolic Algorithm: a 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 any 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. This 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: $(s, A) \in Flatten(B)$ iff $A = \bigcup\{C \mid (s, C) \in B\}$, $A \neq \emptyset$. To make this definition clearer, consider $B = \{(s_0, C_1), (s_0, C_2), (s_1, C_3), (s_1, C_4), (s_2, C_5)\}$. 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.

4.1. 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 $s \models_\omega$

ϕ iff $\omega \in A_\phi(s)$. This is done recursively in Algorithm 2, with respect to the formula structure. As we show, 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 4.1. 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 4.1. 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 4.1. Consider the following sets: $A_\phi = \{(s_0, \Omega), (s_1, \{\omega \mid \omega(t_1) + 3\omega(t_2) < 5\})\}$, and $A_\psi = \{(s_1, \{\omega \mid 2\omega(t_1) + 3\omega(t_3) < 4\})\}$. We have $A_\phi * A_\psi = \{(s_1, \{\omega \mid \omega(t_1) + 3\omega(t_2) < 5 \wedge 2\omega(t_1) + 3\omega(t_3) < 4\})\}$.

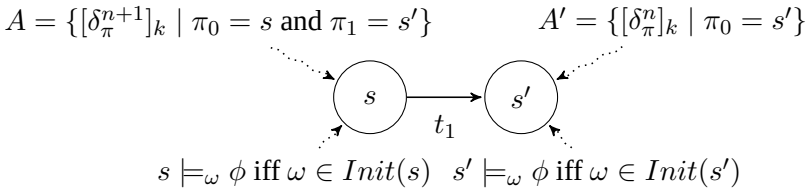
In the translation of $EG^{\leq k}$ and $EU^{\leq k}$ we make use of the *bounded backstep operation*. This operation is defined on sets of triples (s, A, C), where s is a state, A is a set of linear statements used to track possible constraints on parameters, and C is a set of parameter valuations used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

Definition 4.2. 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 $(s', A', C') \in D$.



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 4.2. Consider the following sets: $C_1 = \{\omega \mid \omega(t_1) > 2\}$, $C_2 = \{\omega \mid \omega(t_2) + \omega(t_3) \leq 4\}$, and $D = \{(s_1, \{6t_1 + 8t_2\}, C_1), (s_2, \{4t_2 + 7t_3, t_4\}, C_2)\}$, 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 $BackStep_5(D, Init)$. Notice 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 $K_0, K_1, \ldots$ *stabilises* if there exists $i \geq 0$ such that $K_j = K_i$ for all $j > i$, and denote this as $lim_{j \to \infty} K_j = K_i$. 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 $K_0 = D$, and $K_{i+1} = K_i \cup BackStep_k(K_i, Init)$ stabilises. 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 4.3. 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.

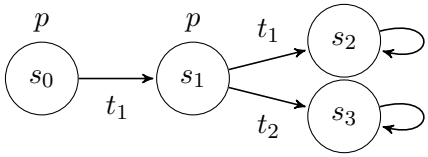


Figure 1: A simple model

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 4.3. 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 4.1. Let ϕ be a formula of $\mathrm{RTCTL_P}$, 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 $0 \leq i \leq n$.

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

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 $\omega \in (\mathcal{E}\mathcal{G}^{\leq k}A_\phi)(s)$. 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 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 $[\delta^m_{\pi'}]_k = [t_m + t_{m-1} + \ldots + t_0]_k \in A_m$, and $[\delta^m_{\pi'}]_k(\omega) > k$. Notice that the latter is equivalent to $\delta^m_{\pi'}(\omega) > k$, 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 $\delta^m_\pi(\omega) > k$. This means that $s \models_\omega EG^{\leq k}\phi$, which concludes the proof. □

Definition 4.4. 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 stabilise.

Theorem 4.2. 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 $s \models_\omega E\phi U^{\leq k}\psi$. 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 4.1 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 we can infer the existence of 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 $out(s')$) such 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 4.5. 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 $\mathcal{EX}^{\leq k} A = Flatten([I_k(A)]_{\leq k})$.

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. This simple observation in summarised in the following corollary.

Corollary 4.2. Let ϕ be a formula of $\mathrm{RTCTL_P}$, 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 the nonnegated expression. To complete the theory we show how to deal with negations.

Definition 4.6. 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 4.3. 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 $\omega \in (\imath A_\phi)(s)$.

The overall Algorithm 2 employs recursive calls and the operations introduced earlier to provide the entry point for the exhaustive parameter synthesis for $\mathrm{RTCTL_P}$; its purpose is summarised in the following theorem.

Theorem 4.3. Let M be a model and $\phi \in \mathrm{RTCTL_P}$, then $s \models_\omega \phi$ iff $\omega \in Synthesize(M, \phi)(s)$ for any state $s \in S$.

Proof: The proof follows immediately from the properties of recursively called operations, Corollaries 4.1 and 4.3, and Theorems 4.2 and 4.1. □

Now we move to the preliminary analysis of the complexity of Algorithm 2. Let $diam(M)$ be the *diameter* of a model M, i.e., the greatest among the lengths n of all such finite sequences $\pi = (s_0, t_0, s_1, t_1, \ldots, s_n, t_n)$ for which no pair (s_i, t_i) repeats for $0 \leq i < n$. Moreover, let $outd(M) = max_{s \in S}\{|out(s)|\}$ denote the maximal output degree of M.

Lemma 4.1. Let $M = (S, s^0, T, \to, \mathcal{L})$ be a model, $\phi \in \mathrm{RTCTL_P}$, and $k \in \mathbb{N}$ be the greatest superscript appearing in ϕ. The pessimistic time complexity of Algorithm 2 for $\mathrm{RTCTL_P}$ synthesis is in $O(|S| outd(M)^{diam(M)(k+1)|T|+2}|\phi|)$.

Proof: The computations of $\mathcal{EG}^{\leq k} A_\phi$ and $\mathcal{E} A_\phi \mathcal{U}^{\leq k} A_\psi$ are the most costly operations in the parameter synthesis for $\mathrm{RTCTL_P}$. Both of these are based on computing the increasing sequences of sets, namely for $\phi, \psi \in \mathrm{RTCTL_P}$ and $k, l \in \mathbb{N}$ we deal with the sums $SG^l(A_\phi) = \bigcup_{i=0}^{l} G_i(A_\phi)$ and $SH^l(A_\phi, A_\psi) = \bigcup_{i=0}^{l} H_i(A_\phi, A_\psi)$. Notice that both $G_i(A_\phi)$ and $H_i(A_\phi, A_\psi)$ have a similar structure, i.e. in the worst case:

$$G_i(A_\phi) = \bigcup_{\pi \in \Pi(M)} \{(\pi_0, [\delta_\pi^{i+1}]_k, \bigcap_{j=0}^{i} A_\phi(\pi_j))\},$$

$$H_i(A_\phi, A_\psi) = \bigcup_{\pi \in \Pi(M)} \{(\pi_0, [\delta_\pi^{i}]_k, \bigcap_{j=0}^{i-1} A_\phi(\pi_j) \cap A_\psi(\pi_i))\}$$

Algorithm 2 $Synthesize(M, \phi)$

Input: a model M, $\phi \in \mathrm{RTCTL_P}$
Output: $A_\phi : S \to 2^\Omega$ s.t. $s \models_\omega \phi \iff \omega \in A_\phi(s)$

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 $\mathcal{E}\mathcal{X}^{\leq k} A_\alpha$
end if
if $\phi = EG^{\leq k}\alpha$ **then**
  $A_\alpha = Synthesize(M, \alpha)$
  return $\mathcal{E}\mathcal{G}^{\leq k} A_\alpha$
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

for $i \in \mathbb{N}$. We can therefore focus on $G_i(A_\phi)$, as the other case follows with very slight alterations only. By a straightforward inductive argument we obtain that $|G_i(A_\phi)| \leq |S| outd(M)^i$. If we preserve the i–th positions of the sequences, i.e., define $G^*_i(A_\phi) = \bigcup_{\pi\in\Pi(M)}\{(\pi_0, \pi_i, [\delta^{i+1}_\pi]_k, \bigcap_{j=0}^{i} A_\phi(\pi_j))\}$, then $G^*_{i+1}(A_\phi)$ can be computed from $G^*_i(A_\phi)$ via $|S|outd(M)^{i+1}$ combined k-additions and set joins. Note that we do not really need to perform the real joins: it suffices to keep track of the labels of the sets present in the join operation and reduce the possible repetitions, which can be done in a constant time using a hashtable, therefore we can assume that the costs of the join and the k-addition are constant. Observe that both $SG^l(A_\phi)$ and $SH^l(A_\phi, A_\psi)$ can be computed in $O(\sum_{i=0}^{l} |S|outd(M)^{i+1}) = O(|S|outd(M)^{l+2})$. Denote $L = diam(M)((k+1)|T| + |S|)$. Let $\pi \in \Pi(M)$ and $i \in \mathbb{N}$ and, for our convenience, define $\Delta^{i,k}_\pi = ([\delta^i_\pi]_k, \bigcap_{j=0}^{i} A_\phi(\pi_j))$. It is easier to work on a derived model $M' = (S', \rightsquigarrow)$ with $S' = \bigcup_{s\in S}\{(s, t, A_\phi(s)) \mid t \in out(s)\}$ and the transition relation $\rightsquigarrow$ such that $\forall_{(s,t,A_\phi(s)),(s',t',A_\phi(s'))\in S'}\big((s, t, A_\phi(s)) \rightsquigarrow (s', t', A_\phi(s'))$ iff $s \xrightarrow{t} s'\big)$. Intuitively, the M' graph is built from the model M by removing the time step parameters from the transitions and attaching them to the states, together with the constraints A_ϕ under which ϕ holds in a given state. Let $\Pi(M')$ denote the set of all the sequences in M'. There is a clear correspondence between the sequences in $\Pi(M)$

and the sequences in $\Pi(M')$, i.e., for each $\pi = (s_0, t_0, s_1, t_1, \ldots)$ we define the unique path $\pi' = ((s_0, t_0, A_\phi(s_0)), (s_1, t_1, A_\phi(s_1)), \ldots)$ and we trivially extend to $\Pi(M')$ the function $\Delta^{n,k}_{\pi'} := \Delta^{n,k}_{\pi}$. It is easy to see that $diam(M)$ can be computed as the length of a maximal path in M' on which no vertices repeat. Now we prove that for each path $\pi' \in \Pi(M')$ we can find another path $\pi''' \in \Pi(M')$, such that $\Delta^{i,k}_{\pi'} = \Delta^{i,k}_{\pi'''}$ for sufficiently large $i \in \mathbb{N}$ and $\Delta^{i,k}_{\pi'''}$ stabilizes on some $i \leq L$. First, consider a path $\pi' \in \Pi(M')$ such that $\Delta^{i,k}_{\pi'} = \Delta^{i+diam(M),k}_{\pi'}$ for some $i \in \mathbb{N}$.

By the pigeonhole principle, there are $i \leq j_1 < j_2 \leq i + diam(M)$ such that $\pi'_{j_1} = \pi'_{j_2}$; we can therefore build another path π'' such that $\pi''_i = \pi'_i$ for $0 \leq i \leq j_1$ and $\pi''_i = \pi'_{j_2-j_1+i}$ for $i > j_1$ and $\Delta^{L,k}_{\pi'} = \Delta^{L,k}_{\pi''}$. In other words the interval $(j_1, j_2]$ can be removed from π'' for our purposes. By consecutively repeating this procedure we can build a path π''' such that $\Delta^{l,k}_{\pi'''} = \Delta^{l,k}_{\pi'}$ for sufficiently large $l \in \mathbb{N}$, and such that if $\Delta^{j_1,k}_{\pi'''}$ is not stable and $\Delta^{j_1,k}_{\pi'''} = \Delta^{j_2,k}_{\pi'''}$ and if $j_1 < j_2$, then $j_1 - j_2 \leq diam(M)$. Intuitively, this means that unless $\Delta^{l}_{\pi'''}$ is stable, it has to change within every $diam(M)$ steps. As there are at most $(k+1)|T| + |S|$ possible changes (each parameter can be incremented and the chain of set joins is of length at most $|S|$), $\Delta^{((k+1)|T|+|S|)diam(M),k}_{\pi'''}$ is stable.

We have thus proven that it suffices to unwind the model M up to depth $L = diam(M)((k+1)|T| + |S|)$ to find the limits $lim_{l\to\infty} SG^l(A_\phi) = SG^L(A_\phi)$ and $lim_{l\to\infty} SH^l(A_\phi) = SH^L(A_\phi)$ that can be used to compute $\mathcal{EG}^{\leq k} A_\phi$ and $\mathcal{E} A_\phi \mathcal{U}^{\leq k} A_\psi$; recall that the cost of this is in $O(|S| outd(M)^{L+2})$. The thesis of the theorem follows via induction on the number of modalities in ϕ. □

The above result gives only an upper bound on the time complexity of Algorithm 2 and we expect the algorithm properly implemented to essentially outperform the brute-force approach (Algorithm 1), similarly as in case of BDD-based model checking algorithms, see [JKLP12, JLKP13]. Notice also that the time complexity of Algorithm 1 given in Theorem 3.1 concerns also the average case (as, here, the worst case equals the average one), while the time complexity of Algorithm 2 given in Lemma 4.1 concerns the worst case only, as the worst case is not equal to the average one. The following example illustrates it, showing some possible benefits of the presented symbolic approach.

Example 4.4. Let $n \in \mathbb{N}$ and $M_n = (S, s^0, T, \to, \mathcal{L})$ be a model such that $S = \bigcup_{0\leq i\leq n}\{s_i, s_{ir}\} \cup \{s_d\}$, $T = \{t_0, t_1, \ldots, t_{n+1}\} \cup \{t_r, t_d\}$, and the transition relation such that $\forall_{0\leq i\leq n}(s_i \xrightarrow{t_{i+1}} s_{ir} \wedge s_{ir} \xrightarrow{t_r} s_i \wedge s_{ir} \xrightarrow{t_d} s_d)$ and $s_d \xrightarrow{t_d} s_d$ and $\forall_{0\leq i<n} s_i \xrightarrow{t_0} s_{i+1}$, and $\mathcal{L}$ such that $\forall_{s\neq s_d}\mathcal{L}(s_i) = \{p\} \wedge \mathcal{L}(s_d) = \emptyset$ (see Fig. 2). The model allows to traverse along the paths of variable lengths globally marked with the proposition p; the state s_d is the only one that is not labeled with p and it serves as a sink, i.e., once visited it cannot be left. For each $0 \leq i \leq n$ define the following sequences: $Tr_i = s_0 \xrightarrow{t_0} s_1 \xrightarrow{t_0} \ldots \xrightarrow{t_0} s_i \xrightarrow{t_{i+1}} s_{ir}$, and $Tr_iB = Tr_i \xrightarrow{t_r} s_0$ (called components of TrB type), and $Tr_iD = Tr_i \xrightarrow{t_d} s_d \xrightarrow{t_d} \ldots$ (called components of type TrD). Notice that each path $\pi \in \Pi(\mathrm{M}, s_0)$ can be presented as an infinite sequence of components of type TrB or a finite sequence of components of type TrB ended by a component TrD. We are interested in estimating the depth of the unwinding needed to synthesise all parameter valuations under which $s_0 \models_\omega \mathcal{EG}^{\leq k} p$. The order of the TrB components in the sequence is not essential in this case, thus it can be assumed that they appear in the increasing order of $Tr^{j_1}_{i_1} Tr^{j_2}_{i_2} \ldots$, where $i_1 < i_2 < \ldots$; it is also easy to see that due to the properties of k–bounded addition we can assume that none of the components appears more than $k+1$ times, i.e., $j_1, j_2, \ldots \leq k+1$. As for all $0 \leq i \leq n$ the sizes of Tr_iB and Tr_iD are $i+2$ and $i+3$, respectively, to perform the synthesis for $\mathcal{EG}^{\leq k} p$ it

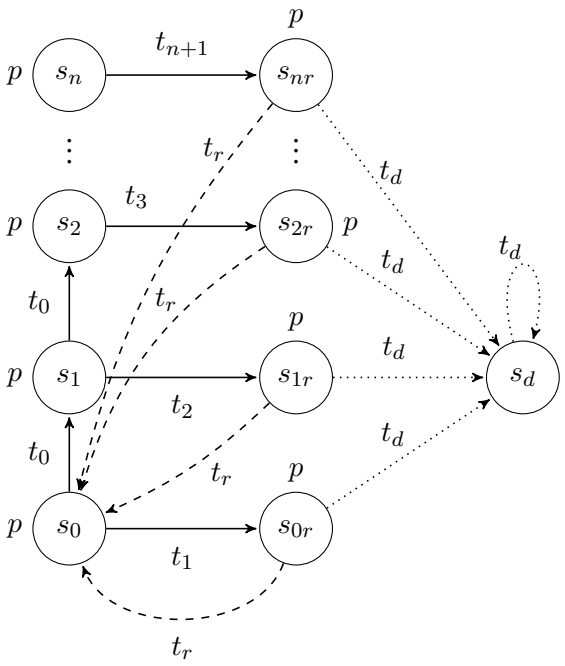


Figure 2: The M_n model for Example 4.4

suffices to consider the sequences of length at most $(k+1)\sum_{i=0}^{n}((i+1)+(i+2)) = O((k+1)n^2)$. This means that it suffices to perform $O((k+1)n^2)$ symbolic $BackStep_k$ operations in Algorithm 2 to synthesise all parameter valuations for $\mathcal{EG}^{\leq k}p$, in contrast to $(k+1)^{n+3}$ independent non-parametric checks in Algorithm 1.
An efficient and compact implementation of the $BackStep_k$ operator and of the state and solution spaces remains our future work.

5. Conclusions

The method presented in this paper allows for the synthesis of parameter values in timed Kripke structures for the properties expressed in the $\mathrm{RTCTL_P}$ logic. To be more precise, for a given property ϕ the result of the synthesis is the set A_ϕ of constraints on the 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 the $\mathrm{RTCTL_P}$ 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.

We have shown that for a given $\mathrm{RTCTL_P}$ 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 the 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 the state-space and the operations on its subsets.

Our future work will consist in implementing the presented symbolic approach using various versions of decision diagrams and SMT-theories, and in comparing experimental results.

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

[ACEF09] É. André, T. Chatain, E. Encrenaz, and L. Fribourg, *An inverse method for parametric timed automata*, International Journal of Foundations of Computer Science **20** (2009), no. 5, 819–836.

[AFKS12] É. André, L. Fribourg, U. Kühne, and R. Soulat, *Imitator 2.5: A tool for analyzing robustness in scheduling problems*, Proc. of the Formal Methods - 18th International Symposium, LNCS, vol. 7436, Springer-Verlag, 2012, pp. 33–36.

[AHV93] R. Alur, T. Henzinger, and M. Vardi, *Parametric real-time reasoning*, Proc. of the 25th Ann. Symp. on Theory of Computing (STOC'93), ACM, 1993, pp. 592–601.

[Doy07] L. Doyen, *Robust parametric reachability for timed automata*, Inf. Process. Lett. **102** (2007), 208–213.

[ET99] E. A. Emerson and R. Trefler, *Parametric quantitative temporal reasoning*, Proc. of the 14th Symp. on Logic in Computer Science (LICS'99), IEEE Computer Society, July 1999, pp. 336–343.

[HHWT97] T. Henzinger, P. Ho, and H. Wong-Toi, *HyTech: A model checker for hybrid systems*, Proc. of the 9th Int. Conf. on Computer Aided Verification (CAV'97), LNCS, vol. 1254, Springer-Verlag, 1997, pp. 460–463.

[HRSV01] T. Hune, J. Romijn, M. Stoelinga, and F. Vaandrager, *Linear parametric model checking of timed automata*, Proc. of the 7th Int. Conf. on Tools and Algorithms for the Construction and Analysis of Systems (TACAS'01), LNCS, vol. 2031, Springer-Verlag, 2001, pp. 189–203.

[JKLP12] A. V. Jones, M. Knapik, A. Lomuscio, and W. Penczek, *Group synthesis for parametric temporal-epistemic logic*, Proc. of the 11th Int. Conf. on Autonomous Agents and Multi-Agent Systems (AAMAS'12), IFAAMAS, 2012, pp. 1107–1114.

[JLKP13] A. V. Jones, A. Lomuscio, M. Knapik, and W. Penczek, *Group synthesis for alternating-time temporal logic*, Tech. report, Department of Computing, Imperial College, London, 2013.

[JLR13] A. Jovanović, D. Lime, and O. H. Roux, *Integer parameter synthesis for timed automata*, Proc. of the 19th international conference on Tools and Algorithms for the Construction and Analysis of Systems (Berlin, Heidelberg), TACAS'13, Springer-Verlag, 2013, pp. 401–415.

[KP12] M. Knapik and W. Penczek, *Bounded model checking for parametric timed automata*, T. Petri Nets and Other Models of Concurrency **5** (2012), 141–159.

[TLR08] L-M. Tranouez, D. Lime, and O. H. Roux, *Parametric model checking of time Petri nets with stopwatches using the state-class graph*, Proc. of the 6th Int. Workshop on Formal Analysis and Modeling of Timed Systems (FORMATS'08), LNCS, vol. 5215, Springer-Verlag, 2008, pp. 280–294.