Bounded Parametric Model Checking for Elementary Net Systems*

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

[1] Institute of Computer Science, PAS, J.K. Ordona 21, 01-237 Warszawa, Poland
{Michal.Knapik,mszreter}@ipipan.waw.pl
[2] Institute of Informatics, Podlasie Academy, Sienkiewicza 51, 08-110 Siedlce, Poland
penczek@ipipan.waw.pl

Abstract. Bounded Model Checking (BMC) is an efficient verification method for reactive systems. BMC has been applied so far to verification of properties expressed in (timed) modal logics, but never to their parametric extensions. In this paper we show, for the first time that BMC can be extended to PRTECTL – a parametric extension of the existential version of CTL. To this aim we define a bounded semantics and a translation from PRTECTL to SAT. The implementation of the algorithm for Elementary Net Systems is presented, together with some experimental results.

1 Introduction

Bounded Model Checking (BMC) [4] is a method of performing verification by stepwise unwinding a verified model and translating the resulting fragment, as well as the property in question, to a propositional formula. The resulting formula is then checked by means of efficient external tools, i.e., SAT-solvers. This method is usually incomplete from the practical point of view, but can find counterexamples in systems that appear too large for other approaches.

BMC was invented in late 1990s, and since then has become an established method among verification approaches. BMC is applied to verification of properties specified in temporal, dynamic, epistemic, and timed logics [3], [2], [7], [11], [13]. In fact, for many system specifications and property languages devised for explicit-state model checking, the BMC counterparts have been developed. In this paper we show how parametric model checking can be performed by means of BMC.

The rest of the paper is organized as follows. In Section 2 we shortly explore the motivations for the choice of the parameterized temporal logics vRTCTL and PRTCTL to which the BMC method is applied. Referenced and cited works are mentioned along with an outline of the contents. Section 3 recalls from [6] the syntax and semantics of the logics used in this work. In Section 3 we define existential fragments of the considered logics – vRTECTL and PRTECTL,

* Partly supported by the Polish Ministry of Science and Higher Education under the grant No. N N206 258035.

respectively. Section 4 introduces k-models together with bounded semantics for vRTECTL and PRTECTL. In Section 5 a translation of a model and a property under investigation is presented together with an algorithm for BMC. Section 6 contains an application of the above method to Elementary Net Systems. We choose three standard problems: the Mutual Exclusion, the Dining Philosophers, and the Generic Pipelining Paradigm. Some associated parameterized properties are verified in Section 7. The concluding remarks and an outline of some future work are in Section 8.

2 Related Work

The work presented in this paper falls into a broad area of Parametric Model Checking – an ambiguous term which may mean that we deal with the parameters in models (as in [1] and [8]), in logics (as in [6] and [5]) or in both (as in [12]). There are two reasons limiting the practical applications of Parametric Model Checking. The first – computational complexity of the problem – is the result of the presence of satisfiability in the Presburger Arithmetic (PA) as a subproblem. In case of the translation of the existential fragment of TCTL to PA formulae proposed in [5], the joint complexity of the solution is 3EXPTIME. The second – undecidability of the problem for Parametric Timed Automata in general [1] – results in a fact that some of the proposed algorithms do not need to stop [8].

In this paper we consider the parametric extension of Computation Tree Logic (CTL), introduced in [6] – namely PRTCTL (Parametric CTL). The logic is interpreted in Kripke structures, and we assume that traversing a transition takes one unit of time. As motivated in [6], such models, while less sophisticated than many other approaches, are often sufficient in systems modelling and analysis. The PRTCTL model checking problem is decidable, and does not contain PA-satisfiability as a subproblem. The Kripke models (marking graphs) induced by elementary Petri nets tend to be very large, which motivates our decision to apply bounded model checking ([4]) methods to the problem. The application of BMC to the existential fragment of the CTL originates from [11] with a further optimization in [14].

To the best knowledge of the authors, this paper presents the first extension of BMC to parameterized temporal logics.

3 Parameterized Temporal Logics

In this section we recall the temporal logics vRTCTL and PRTCTL, first defined in [6], both being extensions of CTL. The logic vRTCTL allows superscripts of form $^{\leq\eta}$, where η is a linear expression over path quantifiers of CTL. An example of a formula of this logic is $EF^{\leq\Theta_1+\Theta_2}(w_1 \wedge EG\neg c_1)$. The formulae of PRTCTL are built from formulae of vRTCTL by adding additional existential or universal quantifiers which may be restricted or unrestricted. As an example of a PRTCTL formula consider $\exists_{\Theta_1\leq 1}\forall_{\Theta_2\leq 2}EF^{\leq\Theta_1+\Theta_2}(w_1 \wedge EG\neg c_1)$. Following

E. A. Emerson's approach [6], the formulae are interpreted in standard Kripke structures, which seem to be appropriate for application in many computer science fields. The logics mentioned above essentially extend CTL, as they allow to formulate properties involving lengths of paths in a model. We interpret superscripts as time bounds, assuming that a transition in a model takes one unit of time. Throughout this paper by $\mathbb{N}$ we denote the set of all natural numbers (including 0). By a *sentence* of a logic we mean a formula without free variables, and by $\alpha(\Theta_1, \ldots, \Theta_n)$ we point out that the formula α contains free parameters $\Theta_1, \ldots, \Theta_n$.

3.1 Syntax

Let $\Theta_1, \ldots, \Theta_n$ be natural variables, called here *parameters*. An expression of the form $\eta = \sum_{i=1}^{n} c_i \cdot \Theta_i + c_0$, where $c_0, \ldots, c_n \in \mathbb{N}$, is called a *linear expression*. A function $\upsilon : \{\Theta_1, \Theta_2, \ldots, \Theta_n\} \longrightarrow \mathbb{N}$ is called a *parameter valuation*. Let Υ be the set of all the parameter valuations.

Definition 1. *Let $\mathcal{PV}$ be a set of propositional variables (propositions). Define inductively the formulae of* vRTCTL :

1. *every member of $\mathcal{PV}$ is a formula,*
2. *if α and β are formulae, then so are $\neg\alpha$, $\alpha \wedge \beta$ and $\alpha \vee \beta$,*
3. *if α and β are formulae, then so are $EX\alpha$, $EG\alpha$, and $E\alpha U\beta$,*
4. *if η is a linear expression, α and β, then so are $EG^{\leq\eta}\alpha$, $E\alpha U^{\leq\eta}\beta$.*

The conditions $1, 2$, and 3 alone define CTL. Notice that η is allowed to be a constant. The logic defined by a modification of the above definition, where $\eta = a$ for $a \in \mathbb{N}$, is called RTCTL in [6]. For example $EF^{\leq 3}(w_1 \wedge EG\neg c_1)$ is an RTCTL formula.
Next, we extend vRTCTL with quantifiers.

Definition 2. *The formulae of* PRTCTL *are defined as follows:*

1. *if $\alpha \in$* vRTCTL, *then $\alpha \in$* PRTCTL,
2. *if $\alpha(\Theta) \in$* PRTCTL, *where Θ is a free parameter, then $\forall_\Theta \alpha(\Theta), \exists_\Theta \alpha(\Theta), \forall_{\Theta \leq a} \alpha(\Theta), \exists_{\Theta \leq a} \alpha(\Theta) \in$* PRTCTL *for $a \in \mathbb{N}$.*

Notice that the following inclusions hold: CTL$\subseteq$RTCTL$\subseteq$vRTCTL$\subseteq$PRTCTL. In this paper we consider only sentences of PRTCTL.

By *true* we mean the formula $p \vee \neg p$, for some proposition p. Additionally we use the derived modalities: $EF\alpha \stackrel{def}{=} E(trueU\alpha)$, $AF\alpha \stackrel{def}{=} \neg EG\neg\alpha$, $AX\alpha \stackrel{def}{=} \neg EX\neg\alpha$, $AG\alpha \stackrel{def}{=} \neg EF\neg\alpha$ (CTL modalities) and $EF^{\leq\eta}\alpha \stackrel{def}{=} E(trueU^{\leq\eta}\alpha)$, $AF^{\leq\eta}\alpha \stackrel{def}{=} \neg EG^{\leq\eta}\neg\alpha$, $AG^{\leq\eta}\alpha \stackrel{def}{=} \neg EF^{\leq\eta}\neg\alpha$. Each modality of CTL has an intuitive meaning. The path quantifier A stands for "on every path" and E means "there exists a path". The modality X means "in the next state", G stands for "in the all states", F means "in some state", and U has a meaning of "until".

The introduced superscripts will become clear when the semantics of vRTCTL is presented. As to give an example of the intuitive meaning of an RTCTL formula, $EG^{\leq 3}p$ may be perceived as the statement "there exists a path such that in the first four states of this path p holds". The logic vRTCTL adds a possibility of expressing similar properties under parameter valuations, while PRTCTL allows for stating that some property holds in a model under some class of parameter valuations.

Definition 3. *The logics* vRTECTL, RTECTL, *and* PRTECTL *are defined as the restrictions of, respectively,* vRTCTL, RTCTL, *and the set of sentences of* PRTCTL *such that the negation can be applied to the propositions only.*

The main idea of bounded model checking is to unwind the computation tree of a model up to some finite depth, therefore in general with this approach it is not possible to verify the properties which deal with all the possible paths. This motivates our choice of the logics in Definition 3 (see [11]) – the formulae of the above restrictions contain the non-negated existential path quantifiers only.

3.2 Semantics

We evaluate the truth of the sentences and the formulae accompanied with parameter valuations in Kripke structures.

Definition 4. *Let $\mathcal{PV}$ be a set of propositional variables. A Kripke structure (a* model*) is defined as a tuple $M = (S, \rightarrow, \mathcal{L})$, where:*

1. *S is a finite set of* states*,*
2. *$\rightarrow \subseteq S \times S$ is a transition relation such that for every $s \in S$ there exists $s' \in S$ with $s \rightarrow s'$ (i.e., the relation is total),*
3. *$\mathcal{L} : S \longrightarrow 2^{\mathcal{PV} \cup \{true\}}$ is a labelling function satisfying $true \in \mathcal{L}(s)$ for $s \in S$.*

Notice that totality of the transition relation guarantees that there are no deadlocks (we need this assumption as the considered logics are interpreted over infinite runs). The labelling function assigns to an each state s a set of propositions which are assumed to be true at s. An infinite sequence $\pi = (s_0, s_1, \ldots)$ of states of a model such that $s_i \rightarrow s_{i+1}$ for $i \in \mathbb{N}$ is called a *path*. By $\pi(i)$ we denote the i–th position on a path π. The number of the states of model M is called the size of M and denoted by $|M|$. For a parameter valuation υ and a linear expression η, by $\upsilon(\eta)$ we mean the evaluation of η under υ.

Definition 5 (Semantics of vRTCTL**).** *Let* M *be a model, s – a state, α, β – formulae of* vRTCTL*. $M, s \models_\upsilon \alpha$ denotes that α is true at the state s in the model M under the parameter valuation υ. We omit M where it is implicitly understood. The relation $\models_\upsilon$ is defined inductively as follows:*

1. *$s \models_\upsilon p \iff p \in \mathcal{L}(s)$*
2. *$s \models_\upsilon \neg p \iff p \notin \mathcal{L}(s)$,*
3. *$s \models_\upsilon \alpha \wedge \beta \iff s \models_\upsilon \alpha$ and $s \models_\upsilon \beta$,*

4. $s \models_\upsilon \alpha \vee \beta \iff s \models_\upsilon \alpha$ *or* $s \models_\upsilon \beta$,
5. $s \models_\upsilon EX\alpha \iff \exists_\pi\big(\pi(0) = s \wedge \pi(1) \models_\upsilon \alpha\big)$,
6. $s \models_\upsilon EG\alpha \iff \exists_\pi\big(\pi(0) = s \wedge \forall_{i\geq 0}\pi(i) \models_\upsilon \alpha\big)$,
7. $s \models_\upsilon E\alpha U\beta \iff \exists_\pi\big(\pi(0) = s \wedge \exists_{i\geq 0}\big[\pi(i) \models_\upsilon \beta \wedge \forall_{j<i}\pi(j) \models_\upsilon \alpha\big]\big)$,
8. $s \models_\upsilon EG^{\leq\eta}\alpha \iff \exists_\pi\big(\pi(0) = s \wedge \forall_{0\leq i\leq \upsilon(\eta)}\pi(i) \models_\upsilon \alpha\big)$,
9. $s \models_\upsilon E\alpha U^{\leq\eta}\beta \iff \exists_\pi\big(\pi(0) = s \wedge \exists_{0\leq i\leq \upsilon(\eta)}\big[\pi(i) \models_\upsilon \beta \wedge \forall_{j<i}\pi(j) \models_\upsilon \alpha\big]\big)$.

If α is a formula of RTCTL, then the validity of $s \models_\upsilon \alpha$ does not depend on the parameter valuation υ, as there are no parameters in the formula. In this case we write $M, s \models \alpha$ omitting the parameter valuation subscript.

Observe that for every formula α of RTCTL there exists a formula β of vRTCTL and a parameter valuation υ such that $\alpha = \upsilon(\beta)$, where $\upsilon(\beta)$ denotes the formula obtained by substituting all the linear expressions with their evaluations under υ. For example the formula $EF^{\leq 5}(w_1 \wedge EG\neg c_1)$ can be obtained from $EF^{\leq\Theta_1}(w_1 \wedge EG\neg c_1)$ by a valuation υ such that $\upsilon(\Theta_1) = 5$ or from $EF^{\leq\Theta_1+\Theta_2}(w_1 \wedge EG\neg c_1)$ by a valuation υ' such that $\upsilon'(\Theta_1) = 3$ and $\upsilon'(\Theta_2) = 2$.

The semantics of PRTCTL is defined in such a way that by eliminating the quantifiers we eventually arrive at a sequence of conjunctions and/or disjunctions of RTCTL formulae. By a *fresh (integer) variable* we mean a new variable which is not a parameter and is not present in the considered formula.

Definition 6 (Semantics of PRTCTL**).** *Let* M *be a model,* s *– a state, and* α *– a formula of* PRTCTL. $M, s \models \alpha$ *denotes that* α *holds at the state* s *in the model* M. *The relation* $\models$ *is defined inductively as follows:*

1. $s \models \forall_\Theta \alpha(\Theta)$ *iff* $\bigwedge_{i_\Theta\geq 0} s \models \alpha(i_\Theta)$,
2. $s \models \forall_{\Theta\leq a} \alpha(\Theta)$ *iff* $\bigwedge_{0\leq i_\Theta\leq a} s \models \alpha(i_\Theta)$,
3. $s \models \exists_\Theta \alpha(\Theta)$ *iff* $\bigvee_{i_\Theta\geq 0} s \models \alpha(i_\Theta)$,
4. $s \models \exists_{\Theta\leq a} \alpha(\Theta)$ *iff* $\bigvee_{0\leq i_\Theta\leq a} s \models \alpha(i_\Theta)$,

where i_Θ *is a fresh integer variable.*

For example:

$$M, s \models \forall_{\Theta_1\leq 1}\exists_{\Theta_2\leq 2} EF^{\leq\Theta_1+\Theta_2}(w_1 \wedge EG\neg c_1)$$

$$\iff \bigwedge_{0\leq i_{\Theta_1}\leq 1} \bigvee_{0\leq i_{\Theta_2}\leq 2} M, s \models EF^{\leq i_{\Theta_1}+i_{\Theta_2}}(w_1 \wedge EG\neg c_1).$$

It is straightforward to check that for a model M and a state s, $M, s \models_\upsilon EG\alpha$ iff $M, s \models_\upsilon EG^{\leq|M|}\alpha$ and $M, s \models_\upsilon E\alpha U\beta$ iff $M, s \models_\upsilon E\alpha U^{\leq|M|}\beta$. The proof of this fact is based on the observation that in every path prefix of length greater or equal than $|M|$ there is a state that occurs (at least) two times, i.e., the path contains a loop.
Let's recall Theorem 1 from [6]:

Theorem 1. *Let* M *be a model and* $Q_{1\Theta_1}\ldots Q_{n\Theta_n}\alpha(\Theta_1, \ldots, \Theta_n)$, *where* $Q_i \in \{\forall, \exists\}$ *and* $\alpha(\Theta_1, \ldots, \Theta_n) \in$ vRTCTL, *be a* PRTCTL *sentence. Then,* $M, s \models Q_{1\Theta_1}\ldots Q_{n\Theta_n}\alpha(\Theta_1, \ldots, \Theta_n)$ *iff* $M, s \models Q_{1\Theta_1\leq|M|}\ldots Q_{n\Theta_n\leq|M|}\alpha(\Theta_1, \ldots, \Theta_n)$.

Next, we enhance the above theorem by the following lemma.

Lemma 1. *Let M be a model and $Q_{1\Theta_1\leq c_1}\dots Q_{n\Theta_n\leq c_n}\alpha(\Theta_1,\dots,\Theta_n)$ where $Q_i\in\{\forall,\exists\}$, $c_i\in\mathbb{N}$, and $\alpha(\Theta_1,\dots,\Theta_n)\in$ vRTCTL be a sentence of* PRTCTL. *Then $M,s\models Q_{1\Theta_1\leq c_1}\dots Q_{n\Theta_n\leq c_n}\alpha(\Theta_1,\dots,\Theta_n)$ iff*
$M,s\models Q_{1\Theta_1\leq min(c_1,|M|)}\dots Q_{n\Theta_n\leq min(c_n,|M|)}\alpha(\Theta_1,\dots,\Theta_n)$.

Proof. See the Appendix 1.

Basically, Theorem 1 allows for replacing the unrestricted quantifiers with their versions bounded with the size of the model, and Lemma 1 states that it suffices to consider the bounds not greater that $|M|$. Therefore, in the rest of this paper we restrict our research to the vRTCTL and PRTCTL formulae with superscripted modalities and restricted quantifiers. Such an approach is typical for model checking, which aims at verifying a given property in a fixed model. Notice, however, that PRTCTL allows for unbounded quantifiers – hence it is possible to express general quantitative properties abstracting from the underlying system. Let us recall an exemplary formula from [6]:

$$\forall_\Theta\big(AG(request\Rightarrow AF^{\leq\Theta}receive)\Rightarrow AG(request\Rightarrow AF^{\leq 2\cdot\Theta}grant)\big).$$

In a scenario where a system consists of a communicating client and a server, the meaning of the above property is that for each time value Θ if it takes at most Θ time steps for a request from the client to reach the server, then it takes at most $2\cdot\Theta$ time steps to obtain the grant (e.g. to some resources). Notice that when we deal with a fixed model, we can easily create a CTL equivalent of the above formula. However, it is not possible to present its general non-parameterized counterpart due to the presence of an unbounded quantifier.

3.3 Example

In Figure 1 the states of the model M are drawn as circles, whereas the values of the labelling function (a set of propositions assumed to be true) are rendered inside. The transitions are drawn as arrows connecting states. The presented Kripke structure is induced by the Petri net modelling the classical problem of Mutual Exclusion for 2 processes (see Subsection 7.1). It is straightforward to check that:

$$M,start\models\forall_{\Theta_1\leq 1}\exists_{\Theta_2\leq 2}EF^{\leq\Theta_1+\Theta_2}(w_1\wedge EG\neg c_1),$$
$$M,start\models\exists_{\Theta_1\leq 3}\forall_{\Theta_2}E\big(w_1U^{\leq\Theta_1}EG^{\leq\Theta_2}r_2\big).$$

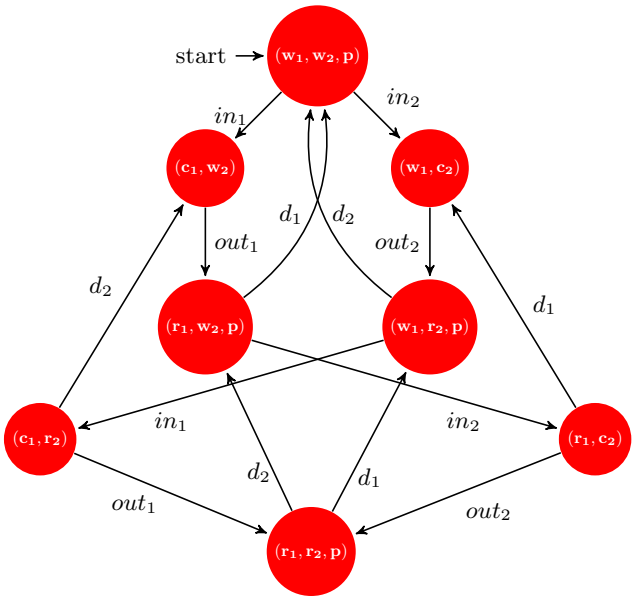
Notice that in the first formula there is no superscript over EG, nevertheless, as we have shown it can be rewritten in the following equivalent form:

$$M,start\models\forall_{\Theta_1\leq 1}\exists_{\Theta_2\leq 2}EF^{\leq\Theta_1+\Theta_2}(w_1\wedge EG^{\leq 8}\neg c_1).$$

Similarly, the second formula can be rewritten in an equivalent form, with the parameter Θ_2 bounded by $|M|$ as follows:

$$M,start\models\exists_{\Theta_1\leq 3}\forall_{\Theta_2\leq 8}E\big(w_1U^{\leq\Theta_1}EG^{\leq\Theta_2}r_2\big).$$

Fig. 1.



4 Bounded Semantics

The idea of bounded model checking is based on a concept of unfolding the computation tree of a given model only to a limited depth. In order to make things more clear we need the following definitions.

Definition 7. *Let M be a model and $k \in \mathbb{N}$. Let $Path_k$ be the set of all sequences $(s_0, \ldots, s_k)$ of states of M, where $s_i \to s_{i+1}$ for each $0 \leq i < k$. The pair $(Path_k, \mathcal{L})$ is called the* k-model *of M and is denoted by M_k.*

An element of $Path_k$ is called a *k-path* and denoted by π_k.

Definition 8. *Let M_k be the k-model of M and $\pi_k \in Path_k$. Define a function $loop : Path_k \longrightarrow 2^{\mathbb{N}}$ as:*

$$loop(\pi_k) = \{l \mid l \leq k \text{ and } \pi_k(k) \to \pi_k(l)\}.$$

A k-path π_k is called a *loop* if $loop(\pi_k) \neq \emptyset$. Observe that loops are essentially a way of representing some infinite paths in a finite way.

Definition 9 (Bounded semantics for vRTECTL**).** *Let M_k be the k-model, s – a state, $\alpha, \beta \in$* vRTECTL*, p – a propositional variable, η – a linear expression, and υ – a parameter valuation. By $M_k, s \models_\upsilon \alpha$ let us denote that α is true (valid) at the state s of M_k. Again, M_k is omitted if it is implicitly understood. Define the relation $\models_\upsilon$ as follows:*

1. $s \models_\upsilon p$ *iff* $p \in \mathcal{L}(s)$
2. $s \models_\upsilon \neg p$ *iff* $p \notin \mathcal{L}(s)$,
3. $s \models_\upsilon \alpha \wedge \beta$ *iff* $s \models_\upsilon \alpha$ *and* $s \models_\upsilon \beta$,
4. $s \models_\upsilon \alpha \vee \beta$ *iff* $s \models_\upsilon \alpha$ *or* $s \models_\upsilon \beta$,
5. $s \models_\upsilon EX\alpha$ *iff* $\exists_{\pi_k \in Path_k}(\pi_k(0) = s \wedge \pi_k(1) \models_\upsilon \alpha)$,
6. $s \models_\upsilon EG^{\leq \eta}\alpha$ *iff* $\exists_{\pi_k \in Path_k}\big(\pi_k(0) = s \wedge \big[((\upsilon(\eta) \leq k) \wedge \bigwedge_{0 \leq i \leq \upsilon(\eta)} \pi_k(i) \models_\upsilon \alpha) \vee ((\upsilon(\eta) > k) \wedge \bigwedge_{0 \leq i \leq k} \pi_k(i) \models_\upsilon \alpha \wedge loop(\pi_k) \neq \emptyset)\big]\big)$,
7. $s \models_\upsilon E(\alpha U^{\leq \eta}\beta)$ *iff* $\exists_{\pi_k \in Path_k}\big(\pi_k(0) = s \wedge \exists_{0 \leq i \leq min(k,\upsilon(\eta))}\big[\pi_k(i) \models_\upsilon \beta \wedge \bigwedge_{0 \leq j < i} \pi_k(i) \models_\upsilon \alpha\big]\big)$.

The above definition differs from its counterpart for ECTL ([11]) in the points 6 and 7. In case of the point 6, we need to consider two subcases. The first subcase deals with the situation when α is checked along a finite path of length $\upsilon(\eta)$ smaller or equal than the depth k of the unfolding of the model. Each such a finite path is then a prefix of some k-path. In the second subcase we deal with the situation when α should be checked along a finite path of length strictly greater than k. Therefore, we have to check α along the loop – hence we have the *loop* condition. Both the subcases are combined in a disjunction. In case of the point 7, we check the existence of such a k-path π_k that the subformula β is valid on its position $\pi_k(i)$ where $i \leq min(k, \upsilon(\eta))$, and for all the positions $\pi_k(j)$ where $j < i$ we have $\pi_k(j) \models_\upsilon \alpha$.

Definition 10 (Bounded semantics for PRTECTL**).** *Let M_k be the k-model of M, s – a state, α – a sentence of* PRTECTL *and $a \in \mathbb{N}$. Define the relation $\models$ as follows:*

1. $M_k, s \models \forall_\Theta \alpha(\Theta)$ *iff* $\bigwedge_{i_\Theta \geq 0} M_k, s \models \alpha(i_\Theta)$,
2. $M_k, s \models \forall_{\Theta \leq a} \alpha(\Theta)$ *iff* $\bigwedge_{0 \leq i_\Theta \leq a} M_k, s \models \alpha(i_\Theta)$,
3. $M_k, s \models \exists_\Theta \alpha(\Theta)$ *iff* $\bigvee_{i_\Theta \geq 0} M_k, s \models \alpha(i_\Theta)$,
4. $M_k, s \models \exists_{\Theta \leq a} \alpha(\Theta)$ *iff* $\bigvee_{0 \leq i_\Theta \leq min(a,k)} M_k, s \models \alpha(i_\Theta)$,

where i_Θ is a fresh integer variable.

The next two lemmas bring forward the essential properties of bounded semantics. Basically they state that the truth of a formula in some k-model is maintained also in a larger l-model and in the whole model M. Therefore if we prove that a formula holds in the k-model (hopefully k is much smaller than $|M|$), then we obtain also the validity of the formula in the model M. These lemmas form a base for the idea of Bounded Model Checking. Namely, we start the search for a proof in the k-model with $k = 0$, then the length k of the paths is incremented until the proof is found or k reaches $|M|$. Then, the conditions 2 of Lemmas 2 and 3 guard that the property holds also in the model M. On the other hand, the conditions 3 of Lemmas 2 and 3 show that if $k = |M|$ is reached and no proof was found, the considered property is not valid in M.

Lemma 2. *Let M_k be the k-model of M, s – a state, υ – a parameter valuation, and α – a formula of* vRTECTL*. Then, the following conditions hold:*

1. $\forall_{l \geq k}\ \big(M_k, s \models_v \alpha \textit{ implies } M_l, s \models_v \alpha\big)$,
2. $M_k, s \models_v \alpha \textit{ implies } M, s \models_v \alpha$,
3. $M, s \models_v \alpha \textit{ implies } M_{|M|}, s \models_v \alpha$.

Notice that Lemma 2 has its counterpart concerning PRTECTL as stated below.

Lemma 3. *Let M be a model, s – a state and α – a* PRTECTL *sentence. Then, the following conditions hold:*

1. $\forall_{l \geq k}\ \big(M_k, s \models \alpha \textit{ implies } M_l, s \models \alpha\big)$,
2. $M_k, s \models \alpha \textit{ implies } M, s \models \alpha$,
3. $M, s \models \alpha \textit{ implies } M_{|M|}, s \models \alpha$.

Proofs of both the above lemmas can be found in the Appendix 1.

Both the above lemmas also hold when $|M|$ is substituted with the diameter of the model. However, while the BMC method is complete for PRTECL (see clause 3 in Lemma 3), in practice for complex models (e.g., induced by Petri nets) unwinding of the computation tree up to the size (or diameter) of the model is not possible. The main appeal of Bounded Model Checking is the ability to verify properties in a partial unfolding of a model which is especially valuable when looking for counterexamples, e.g., errors in system design.

4.1 Example

Recall the formulae and the model M from Example 3.3. One can check that

$$M_2, start \models \forall_{\Theta_1 \leq 1} \exists_{\Theta_2 \leq 2} EF^{\leq \Theta_1 + \Theta_2}(w_1 \wedge EG\neg c_1),$$

while this property does not hold in the bounded semantics for the k–models with k strictly smaller than 2. Similarly, we have

$$M_2, start \models \exists_{\Theta_1 \leq 3} \forall_{\Theta_2} E\big(w_1 U^{\leq \Theta_1} EG^{\leq \Theta_2} r_2\big),$$

while this does not hold for the k–models with k strictly smaller than 2.

5 Bounded Model Checking

The BMC algorithm is based on the idea of a translation of a part of the given model together with a temporal formula to a propositional formula. Satisfiability of the result means that the formula is true in the model. The first part of this section gives definitions and theorems concerning submodels, the second part presents the rules for the translation, whereas the last part includes the description of the BMC algorithm.

5.1 Submodels

We aim at giving a method of checking the validity of temporal formulae in k-models. In order to obtain an acceptable efficiency, the algorithm works on submodels of the k-model.

Definition 11. *Let $M_k = (Path_k, \mathcal{L})$ be the k-model of a model M. A substructure $M'_k = (Path'_k, \mathcal{L}')$, where $Path'_k \subseteq Path_k$ and $\mathcal{L}'$ is the restriction of $\mathcal{L}$ to the states present in the paths of $Path'_k$ is called a* submodel *of M_k.*

The bounded semantics of vRTECTL formulae and PRTECTL sentences over submodels is defined as for k-models. If $M'_k = (Path'_k, \mathcal{L}')$ and $M''_k = (Path''_k, \mathcal{L}'')$ are submodels of some k-model M_k, such that $Path''_k \subseteq Path'_k$, we write $M''_k \subseteq M'_k$.

Lemma 4. *Let M_k be the k-model of a model M, M'_k and M''_k – its submodels, such that $M''_k \subseteq M'_k$ and s a state present in some path of M''_k. Then, we have:*

1. $M''_k, s \models_\upsilon \alpha \Rightarrow M'_k, s \models_\upsilon \alpha$ *for* $\alpha \in$ vRTECTL *and any parameter valuation* υ,
2. $M''_k, s \models \alpha \Rightarrow M'_k, s \models \alpha$, *for* $\alpha \in$ PRTECTL.

Proof. The first part of the lemma is easily proven by the structural induction. In order to prove the second part, notice that in the bounded semantics the non-modal quantifiers are rewritten as, respectively, conjunctions or disjunctions, and use the result of the first part.

It was proven in [11] that in order to determine the truth of an ECTL formula in M_k it is sufficient to consider only submodels of a size (that is – the value of $|Path'_k|$) given by a special function on the checked formula. We extend these results to vRTECTL and PRTECTL.

Definition 12. *Let $\alpha, \beta \in$ vRTECTL, p – a propositional variable, η – a linear expression, and υ – a parameter valuation. Recall that Υ is the set of all parameter valuations. We define recursively the special function $g_k :$ vRTECTL$\times\Upsilon \longrightarrow \mathbb{N}$ as follows:*

1. $g_k(p, \upsilon) = g_k(\neg p, \upsilon) = 0$,
2. $g_k(\alpha \vee \beta, \upsilon) = max(g_k(\alpha, \upsilon), g_k(\beta, \upsilon))$,
3. $g_k(\alpha \wedge \beta, \upsilon) = g_k(\alpha, \upsilon) + g_k(\beta, \upsilon)$,
4. $g_k(EX\alpha, \upsilon) = g_k(\alpha, \upsilon) + 1$,
5. $g_k(EG^{\leq\eta}\alpha, \upsilon) = (min(\upsilon(\eta), k) + 1) \cdot g_k(\alpha, \upsilon) + 1$,
6. $g_k(E\alpha U^{\leq\eta}\beta, \upsilon) = min(\upsilon(\eta), k) \cdot g_k(\alpha, \upsilon) + g_k(\beta, \upsilon) + 1$.

Let us build some intuitions on the above function. For a given k, let M_k be the k–model, s – a fixed state present in some path of M_k, α – a vRTECTL formula, and υ – a parameter valuation. The function $g_k(\alpha, \upsilon)$ returns the size of some submodel M'_k of M_k, such that $M_k, s \models_\upsilon \alpha$ iff $M'_k, s \models_\upsilon \alpha$. Let us focus on the fifth clause of the above definition. If $EG^{\leq\eta}\alpha$ holds in M_k under valuation υ, then there exists a k–path containing $min(\upsilon(\eta), k) + 1$ states along which α is satisfied. This path, together with $(min(\upsilon(\eta), k) + 1) \cdot g_k(\alpha, \upsilon)$ others, form the $Path'_k$ set of such a model M'_k that $M'_k, s \models_\upsilon EG^{\leq\eta}\alpha$. For all the remaining cases, consult the proof of Lemma 5.

Definition 13. *Let* $\alpha \in$ PRTECTL. *We define recursively the special function* f_k : PRTECTL $\longrightarrow \mathbb{N}$ *as follows:*

1. *if* $\alpha \in$ RTCTL *then* $f_k(\alpha) = g_k(\alpha, \upsilon)$ *for any* υ,
2. *if* $\alpha = \forall_{\Theta \leq c}\beta(\Theta)$ *then* $f_k(\alpha) = \sum_{i_\Theta \leq c} f_k(\beta(i_\Theta))$,
3. *if* $\alpha = \exists_{\Theta \leq c}\beta(\Theta)$ *then* $f_k(\alpha) = max_{i_\Theta \leq min(c,k)}\{f_k(\beta(i_\Theta))\}$,

where i_Θ *is a fresh integer variable.*

As the RTCTL formulae considered in the condition 1 of Definition 13 contain no free parameters, the above definition is unambiguous. The following lemmas state that we can determine the truth of the vRTECTL and PRTECTL formulae in the k-model using submodels of size bounded by the value of the appropriate function f_k or g_k.

Lemma 5. *Let* $\alpha \in$ vRTECTL, M_k *be the* k*-model of a model* M*, and* υ *– a parameter valuation. For any state* s *present in some path of* M_k, $M_k, s \models_\upsilon \alpha$ *if and only if there exists a submodel* M'_k *of* M_k *such that* $M'_k, s \models_\upsilon \alpha$ *and* $|Path'_k| \leq g_k(\alpha, \upsilon)$.

Proof. See the Appendix 1.

Lemma 6. *Let* β *be a* PRTECTL *sentence and* M_k *be the* k*-model of a model* M*. For any state* s *present in some path of* M_k, $M_k, s \models \beta$ *if and only if there exists a submodel* M'_k *of* M_k *such that* $M'_k, s \models \beta$ *and* $|Path'_k| \leq f_k(\beta)$.

Proof. See the Appendix 1.

From Lemmas 5,6, Lemma 4 (notice that the k-model is also a submodel) and Lemmas 2 and 3 we obtain that the truth of a formula in some submodel of size bounded by the appropriate g_k or f_k function implies the truth in a model. On the other hand, Lemmas 2 and 3 state that if a formula is true in a model, then it is also true in some k-model, or equivalently, by Lemmas 2 and 3 in its submodel of size bounded by the value of appropriately g_k or f_k.

5.2 Translation to SAT

In order to translate the problem of validity of a sentence $\alpha \in$ PRTECTL in the submodel M'_k to the problem of satisfiability of a propositional formula $\left[\alpha\right]_k$ we have to encode M'_k and α, and then combine the results together. We present an adapted version of the efficient translation introduced in [14].

Consider a model M. As the number of the states of M is finite, they can be perceived as bit vectors of the length $r = \lceil log|M| \rceil$. Therefore, we can represent the states as the valuations of the vector $w = (w_1, \ldots, w_r)$. This vector is called a *global state variable* while each its member w_i is called a *state variable*. Denote by $\mathcal{SV}$ the set of state variables, then a valuation $V : \mathcal{SV} \longrightarrow \{0, 1\}$ naturally extends to the valuation of the global state variables $\hat{V} : \mathcal{SV}^r \longrightarrow \{0, 1\}^r$ in such a way that $\hat{V}(w_1, \ldots, w_r) = (V(w_1), \ldots, V(w_r))$. With a slight notational

abuse, we denote by $\hat{V}(w)$ a state encoded by bit vector. The *symbolic k-path* is a vector of global state variables. As we need a number of symbolic k-paths to represent the k-paths in a translated submodel, by $(w_{0,i}, w_{1,i}, \ldots, w_{r,i})$ we denote the i-th symbolic k-path, where $w_{j,i}$ is a global state variable.

Let w, w' be global state variables, s a state and p a proposition. In the rules of the translation the following propositional formulae are used:

1. $p(w)$ denotes a formula such that $V \models p(w)$ iff $p \in \mathcal{L}(\hat{V}(w))$,
2. $T(w, w')$ denotes a formula such that $V \models T(w, w')$ iff $\hat{V}(w) \rightarrow \hat{V}(w')$ (i.e., there exists a transition between $\hat{V}(w)$ and $\hat{V}(w')$ in the model M),
3. $H(w, w')$ is a formula such that $V \models H(w, w')$ iff $\hat{V}(w) = \hat{V}(w')$ (encoding the equality of states),
4. $L_k(j) = \bigvee_{i=0}^{k} T(w_{k,j}, w_{i,j})$ encodes a loop, that is $V \models L_k(j)$ iff $loop((V(w_{0,j}), \ldots, V(w_{k,j}))) \neq \emptyset$,
5. $I_s(w)$ is a formula such that $V \models I_s(w)$ iff $\hat{V}(w) = s$ (encoding the initial state).

Let M be a model and let A be a finite subset of $\mathbb{N}$. Then, the *unfolding of the transition relation* is defined as

$$\left[M\right]_k^A := \bigwedge_{j \in A} \bigwedge_{i=0}^{k-1} T(w_{i,j}, w_{i+1,j}).$$

It is easy to see that $V \models \left[M\right]_k^A$ iff for each $j \in A$, $(V(w_{0,j}), \ldots, V(w_{k,j}))$ is a k-path in M. As the translation introduced in [14] was an essential improvement over the original one of [11], we follow A. Zbrzezny's approach in our work. We recall the following definitions from [14].

Let A and B be finite subsets of $\mathbb{N}$. By $A \prec B$ we denote that $x < y$ for all $x \in A$ and $y \in B$. Let $k, m, p \in \mathbb{N}$ and $m \leq |A|$, then:

1. $\hat{g}_L(A, m)$ is the subset B of A such that $|B| = m$ and $B \prec A \backslash B$,
2. $\hat{g}_R(A, m)$ denotes the subset B of A such that $|B| = m$ and $A \backslash B \prec B$,
3. $h_X(A)$ is the set $A \backslash \{min(A)\}$,
4. if $k+1$ divides $|A| - 1$ then $h_G(A, k)$ is the sequence of sets $(B_0, \ldots, B_k)$ such that $\bigcup_{i=0}^{k} B_i = A \backslash \{min(A)\}$, $|B_i| = |B_j|$ and $B_i \prec B_j$ for every $0 \leq i < j \leq k$,
5. if k divides $|A| - 1 - p$, then $h_U(A, k, p)$ denotes the sequence of sets $(B_0, \ldots, B_k)$ such that $\bigcup_{i=0}^{k} B_i = A \backslash \{min(A)\}$, $B_i \prec B_j$ for every $0 \leq i < j \leq k$, $|B_0| = \ldots = |B_{k-1}|$ and $|B_k| = p$.

We also need a sequence element selector, that is if $h_G(A, k) = (B_0, \ldots, B_k)$ then define $h_G(A, k)(i) = B_i$ for $0 \leq i \leq k$ and if $h_U(A, k, p) = (B_0, \ldots, B_k)$, define $h_U(A, k, p)(i) = B_i$ for $0 \leq i \leq k$.

The functions $\hat{g}_L$ and $\hat{g}_R$ are used to divide the set of path indices into the two parts of the sizes sufficient to perform the independent translation of subformulae α and β of formula $\alpha \wedge \beta$. Similarly, the functions h_G and h_U are used to divide

the set of path indices into the sequences (hence the use of the selector) of subsets which are of the sizes sufficient to perform the translation of subformulae α and α together with β of, respectively, formulae $EG^{\leq\eta}\alpha$ and $E\alpha U^{\eta}\beta$. These functions were not present in [11], where all the proper subformulae of a given formula α of ECTL were translated using the full set $\{i \in \mathbb{N} \mid 1 \leq i \leq g_k(\alpha)\}$ of indices. For a more in depth description we refer the reader to [14].

Definition 14 (Translation of vRTECTL**).** *Let $\alpha, \beta \in$ vRTECTL, p – a propositional variable, υ – a parameter valuation, η – a linear expression, $(m,n) \in \mathbb{N} \times \mathbb{N}$, and $A \subseteq \mathbb{N}$.*

$$\big[p\big]_k^{[m,n,A,\upsilon]} := p(w_{m,n}) \text{ and } \big[\neg p\big]_k^{[m,n,A,\upsilon]} := \neg p(w_{m,n}),$$

$$\big[\alpha \wedge \beta\big]^{[m,n,A,\upsilon]} := \big[\alpha\big]^{[m,n,\hat{g}_L(A,g_k(\alpha,\upsilon)),\upsilon]} \wedge \big[\beta\big]^{[m,n,\hat{g}_R(A,g_k(\beta,\upsilon)),\upsilon]},$$

$$\big[\alpha \vee \beta\big]^{[m,n,A,\upsilon]} := \big[\alpha\big]^{[m,n,\hat{g}_L(A,g_k(\alpha,\upsilon)),\upsilon]} \wedge \big[\beta\big]^{[m,n,\hat{g}_L(A,g_k(\beta,\upsilon)),\upsilon]},$$

$$\big[EX\alpha\big]^{[m,n,A,\upsilon]} := H(w_{m,n}, w_{0,min(A)}) \wedge \big[\alpha\big]_k^{[1,min(A),h_X(A),\upsilon]}.$$

The translation of the formula $EG^{\leq\eta}\alpha$ depends on the value of $\upsilon(\eta)$. If $\upsilon(\eta) > k$, then:

$$\big[EG^{\leq\eta}\alpha\big]^{[m,n,A,\upsilon]} := H(w_{m,n}, w_{0,min(A)}) \wedge L_k(min(A)) \wedge \bigwedge_{j=0}^{k} \big[\alpha\big]_k^{[j,min(A),h_G(A,k)(j),\upsilon]}$$

and if $\upsilon(\eta) \leq k$, then

$$\big[EG^{\leq\eta}\alpha\big]^{[m,n,A,\upsilon]} := H(w_{m,n}, w_{0,min(A)}) \wedge \bigwedge_{j=0}^{\upsilon(\eta)} \big[\alpha\big]_k^{[j,min(A),h_G(A,\upsilon(\eta))(j),\upsilon]}.$$

The translation of $E\alpha U^{\leq\eta}\beta$ is defined as follows:

$$\big[E\alpha U^{\leq\eta}\beta\big]^{[m,n,A,\upsilon]} := H(w_{m,n}, w_{0,min(A)})$$

$$\wedge \bigvee_{i=0}^{min(\upsilon(\eta),k)} \Big(\big[\beta\big]_k^{[i,min(A),h_U(A,min(\upsilon(\eta),k),g_k(\beta,\upsilon))(min(\upsilon(\eta),k)),\upsilon]}$$

$$\wedge \bigwedge_{j=0}^{i-1} \big[\alpha\big]_k^{[j,min(A),h_U(A,min(\upsilon(\eta),k),g_k(\beta,\upsilon))(j),\upsilon]}\Big).$$

The above encoding is based on the definition of the bounded semantics for vRTECTL – see Definition 9 together with the associated comment.

Definition 15 (Translation of PRTECTL**).** *Let $\alpha \in$ PRTECTL, $A \subseteq \mathbb{N}$, $(m,n) \in \mathbb{N} \times \mathbb{N}$, and $c \in \mathbb{N}$. If α contains no quantifiers and no free parameters, then:*

$$\big[\alpha\big]_k^{[m,n,A]} := \big[\alpha\big]_k^{[m,n,A,\upsilon]}, \text{ where } \upsilon \text{ is any parameter valuation.}$$

As in the above case $\alpha \in$ vRTECTL and it contains no free parameters, the choice of υ is irrelevant.

$$\big[\forall_{\Theta \leq c}\alpha(\Theta)\big]_k^{[m,n,A]} := \big[\alpha(c)\big]_k^{[m,n,\hat{g}_L(A,f_k(\alpha(c)))]} \wedge \big[\forall_{\Theta \leq c-1}\alpha(\Theta)\big]_k^{[m,n,\hat{g}_R(A,f_k(\forall_{\Theta \leq c-1}\alpha(\Theta)))]},$$

Let $d = min(c, k)$, then:

$$\left[\exists_{\Theta \leq c} \alpha(\Theta)\right]_k^{[m,n,A]} := \left[\alpha(d)\right]_k^{[m,n,\hat{g}_L(A, f_k(\alpha(d)))]} \vee \left[\exists_{\Theta \leq d-1} \alpha(\Theta)\right]_k^{[m,n,\hat{g}_L(A, f_k(\exists_{\Theta \leq d-1} \alpha(\Theta)))]}.$$

Let M_k be the k-model of a model M. If $\alpha \in$ vRTECTL and υ is a parameter valuation, then define $G_k(\alpha, \upsilon) := \{i \in \mathbb{N} \mid 1 \leq i \leq g_k(\alpha, \upsilon)\}$. Similarly, if $\beta \in$ PRTECTL, then define $F_k(\beta) := \{i \in \mathbb{N} \mid 1 \leq i \leq f_k(\beta)\}$. The sets G_k and F_k contain the indices of symbolic k-paths used to perform the translation. The formulae $\left[M\right]_k^{G_k(\alpha,\upsilon)}$ and $\left[M\right]_k^{F_k(\beta)}$ encode all the M_k submodels of the size not greater than needed to validate the truth of formulae α, β as indicated in Lemmas 5, 6.

Now, we are in the position to complete the translation of the problem of validity in vRTECTL and PRTECTL to the problem of satisfiability of propositional formulae. Let M_k be the k-model of a model M, $\alpha \in$ vRTECTL and υ be a parameter valuation. Denote

$$\left[M\right]_k^{\alpha,\upsilon} := \left[M\right]_k^{G_k(\alpha,\upsilon)} \wedge I_s(w_{0,0}) \wedge \left[\alpha\right]_k^{[0,0,G_k(\alpha,\upsilon),\upsilon]}.$$

Similarly, let $\beta \in$ PRTECTL, then denote

$$\left[M\right]_k^{\beta} := \left[M\right]_k^{F_k(\beta)} \wedge I_s(w_{0,0}) \wedge \left[\beta\right]_k^{[0,0,F_k(\beta)]}.$$

The following theorems ensure completeness and correctness of the translation.

Theorem 2. *Let M_k be the k-model of M, υ – a parameter valuation, α – a formula of* vRTECTL *containing at least one modality, and s a state. Then, the following equivalence holds: $M_k, s \models_\upsilon \alpha$ iff $\left[M\right]_k^{\alpha,\upsilon}$ is satisfiable.*

Proof. See the Appendix 2.

Theorem 3. *Let M_k be the k-model of M, β – a sentence of* PRTECTL *containing at least one modality, and s – a state. Then, the following equivalence holds: $M_k, s \models \beta$ iff $\left[M\right]_k^{\beta}$ is satisfiable.*

Proof. Replace the non-modal quantifiers in a formula of PRETCTL with, appropriately, conjunctions or disjunctions. To conclude use Theorem 2.

5.3 Example

Consider the model M from Example 3.3 and the formula:

$$\alpha = \forall_{\Theta_1 \leq 1} \exists_{\Theta_2 \leq 2} EF^{\leq \Theta_1 + \Theta_2}(w_1 \wedge EG \neg c_1).$$

The number of the paths needed to encode α in the 2–model is computed as follows:

$$f_k(\alpha) = \sum_{i_{\Theta_1} \leq 1} max_{i_{\Theta_2} \leq 2} \{f_k(EF^{\leq i_{\Theta_1} + i_{\Theta_2}}(w_1 \wedge EG \neg c_1))\}.$$

Let $\beta = EF^{\leq i_{\Theta_1}+i_{\Theta_2}}(w_1 \wedge EG\neg c_1))$, and observe that if $i_{\Theta_1} \leq 1$ and $i_{\Theta_2} \leq 2$ are fixed, then $f_k(\beta) = g_k(\beta, \upsilon)$ where $\upsilon(\Theta_1) = i_{\Theta_1}$ and $\upsilon(\Theta_2) = i_{\Theta_2}$. As $g_k(true, \upsilon) = 0$, we have $g_k(\beta, \upsilon) = g_k(w_1 \wedge EG\neg c_1, \upsilon)+1 = 2$, therefore $f_k(\alpha) = 4$. Thus, the encoding in the 2–model of M is as follows:

$$\big[\forall_{\Theta_1\leq 1}\exists_{\Theta_2\leq 2} EF^{\leq\Theta_1+\Theta_2}(w_1 \wedge EG\neg c_1)\big]_2^{[0,0,\{1,2,3,4\}]}$$

$$= \big[\exists_{\Theta_2\leq 2} EF^{\leq\Theta_2}(w_1 \wedge EG\neg c_1)\big]_2^{[0,0,\{1,2\}]} \wedge \big[\exists_{\Theta_2\leq 2} EF^{\leq 1+\Theta_2}(w_1 \wedge EG\neg c_1)\big]_2^{[0,0,\{3,4\}]}$$

$$= \bigvee_{i=0}^{2} \big[EF^{\leq i}(w_1 \wedge EG\neg c_1)\big]_2^{[0,0,\{1,2\}]} \wedge \bigvee_{j=1}^{2} \big[EF^{\leq j}(w_1 \wedge EG\neg c_1)\big]_2^{[0,0,\{3,4\}]}.$$

As the illustration of the further steps of the translation, consider:

$$\big[EF^{\leq 2}(w_1 \wedge EG\neg c_1)\big]_2^{[0,0,\{3,4\}]} = H(w_{0,0}, w_{0,3}) \wedge \bigvee_{i=0}^{2} \big[w_1 \wedge EG\neg c_1\big]_2^{[i,3,\{3,4\}]}$$

$$= H(w_{0,0}, w_{0,3}) \wedge \bigvee_{i=0}^{2} \big(\big[w_1\big]^{[i,3,\emptyset]} \wedge \big[EG\neg c_1\big]_2^{[i,3,\{4\}]}\big)$$

$$= H(w_{0,0}, w_{0,3}) \wedge \bigvee_{i=0}^{2} \big(p_{w_1}(w_{i,3}) \wedge H(w_{i,3}, w_{0,4}) \wedge L_2(4) \wedge \bigwedge_{j=0}^{2} \neg p_{c_1}(w_{j,4})\big).$$

5.4 The BMC algorithm

Let M be a model and $\alpha \in$ PRTECTL.

```
BMCverifyPRTECTL(α)
     for k := 1 to |M|
         compute the translation [M]_k^{α,υ}
         if [M]_k^{α,υ} is satisfiable return true
     end for
     return false
```

Checking satisfiability of a propositional formula is delegated to an efficient SAT-solver. Obviously the algorithm terminates in a finite number of iterations. By Theorem 2 and Lemma 3 the result is positive (that is – the translation of the formula α is satisfiable) if and only if α is valid in the state s of a model M.

It is easy to present a similar algorithm for checking the validity of vRTECTL formulae under a parameter valuation υ – the only difference is the choice of the appropriate translation.

6 Implementation of Parametric BMC for Elementary Net Systems

In this section we recall some basic definitions concerning Elementary Net Systems (called also Elementary Petri Nets) and present the implementation of BMC for a model generated by a net. The formulations of this section originate from [11]. We consider only the *safe* Petri Nets, i.e., Petri Nets, where each place can be marked with at the most one token.

6.1 Elementary Net Systems

Definition 16. *A* net *is a triple $N = (B, E, F)$, where B (the* places*) and E (the* transitions*) are finite sets satisfying $B \cap E = \emptyset$, the relation (called a* flow relation*) $F \subseteq (B \times E) \cup (E \times B)$ has the property that for every $t \in E$ there exists $p, q \in B$ such that $(p, t), (t, q) \in F$.*

Let N be a net and $t \in E$, then $\bullet t = \{p \in B \mid (p, t) \in F\}$ is called the *pre-set* of t and $t\bullet = \{p \in B \mid (t, p) \in F\}$ is called the *post-set* of t. A *configuration* of a net $N = (B, E, F)$ is a subset C of B. An usual method of visualisation of nets is where the places are rendered as circles, the transitions as boxes, the elements of flow relation as arrows, and the configuration C is represented by placing a token in every circle corresponding to a place in C. A place not marked by a token is called *free*.

Definition 17. *A quadruple $EN = (B, E, F, C_{in})$, where (B, E, F) is a net and $C_{in} \subseteq B$ is the initial configuration, is called an* elementary net system.

Definition 18. *Let $EN = (B, E, F, C_{in})$ be an elementary net system and $t \in E$.*

1. *Let $C \subseteq B$ be a configuration. If t is a transition, $\bullet t \subseteq C$, and $(t\bullet \setminus \bullet t) \cap C = \emptyset$, then the transition t is* enabled *in C (denoted by $C[t\rangle$).*
2. *Let $C, D \subseteq B$ be configurations. A transition t* fires *from C to D (denoted by $C[t\rangle D$) if $C[t\rangle$ and $D = (C \setminus \bullet t) \cup t\bullet$.*
3. *A configuration $C \subseteq B$ is* reachable *if there are configurations $C_0, C_1, \ldots, C_n \subseteq B$ with $C_0 = C_{in}, C_n = C$ and transitions $t_1, \ldots, t_n \in E$ such that $C_{i-1}[t_i\rangle C_i$ for all $1 \leq i \leq n$. We denote the set of all the reachable configuration by C_{EN}.*

Informally, the arrows of the flow relation can be thought of as the directed paths of movement of tokens. If there is an arrow directed from a place b to a transition t, then we say that b *enters* t. If there exists an arrow directed from a transition t to a place b, then we say that t *fills* b. The transition t is enabled if all the places entering t are marked with tokens and all the places filled by t and not entering the transition t are free. If a transition t fires, then the tokens from all the places entering t disappear and appear in all the places filled by t.

6.2 Implementation

Our goal is to construct a Kripke model reflecting the states (markings) and actions (firings) in an elementary net system. Consider an elementary net system $EN = (B, E, F, C_{in})$ and number the places of the net with integers smaller or equal than $n = |B|$. We use a set $\{p_1, \ldots, p_n\}$ of propositions, where p_i is interpreted as the presence of a token in the place number i. If w is a state, then by $p_i \in w$ we mean that the i-th place is marked in the corresponding configuration.

We define the model $M = (S, \rightarrow, \mathcal{L})$ for EN by placing $S = C_{EN}$ (the reachable configurations are the states), $w \rightarrow v$ iff there exists $t \in E$ such that $w[t\rangle v$ (the transitions model the firings) for $w, v \in S$, and $p_i \in \mathcal{L}(w)$ iff $p_i \in w$ (the labelling models the markings).

It is easy to see that we can encode the states of S by valuations of a vector of the state variables $w = (w[1], \ldots, w[n])$, where $w[i] = p_i$ for $0 \leq i \leq n$. Moreover, let $P = \{1, \ldots, n\}$ and let $pre(t), post(t) \subseteq P$ be finite sets of the indices of the places of, respectively, $pre{-}set(t)$ and $post{-}set(t)$. Let $\xi(C_{in}) \subseteq P$ be the set of indices of the places in C_{in}.

Now, we are in the position to present the definitions:

1. $I_{C_{in}}(w) := \bigwedge_{i \in \xi(C_{in})} w[i] \wedge \bigwedge_{i \in P \setminus \xi(C_{in})} \neg w[i]$,
2. $T(w, v) := \bigvee_{t \in E} \big(\bigwedge_{i \in pre(t)} w[i] \wedge \bigwedge_{i \in (post(t) \setminus pre(t))} \neg w[i] \wedge \bigwedge_{i \in (pre(t) \setminus post(t))} \neg v[i]$
   $\wedge \bigwedge_{i \in post(t)} v[i] \wedge \bigwedge_{i \in (P \setminus (pre(t) \cup post(t))) \cup (pre(t) \cap post(t))} w[i] \iff v[i] \big)$,
3. $p_i(w) := w[i]$,
4. $H(w, v) := \bigwedge_{1 \leq i \leq n} w[i] \iff v[i]$.

7 Experimental Results

We have implemented the presented algorithm on top of the BMC module of Verics. Elementary Net Systems are used as an input specification formalism, whereas PRTECTL is used as an input logic.

In order to show the performance and present some case studies we use standard scalable benchmarks. The detailed descriptions of these examples can be found in [11] and [10]. We have implemented our methods exactly as they are presented in this paper, the only optimization consists in using an efficient translation originating from [14].

The tables with results show the following data in the columns from left to right: the formula verified, the number of processes (denoted by NoP), the depth k of the unfolding of the model, the size of the corresponding propositional formula (numbers of variables and clauses) together with the description of how much resources (time and memory) does the translation take, the time it took for MiniSat SAT solver to check the satisfiability, and finally the SAT? column indicating whether the tested formula is satisfiable ($\surd$) or not satisfiable ($\times$).

The experiments have been performed on a Linux machine with dual core 1.6 GHz processor. We tested satisfiability using the MiniSAT solver [9]. The

presented models are relatively simple, yet classical, and the considered formulae have been chosen as to show the difference between the expressive power of CTL and PRECTL. As our work is still in its preliminary stage, we do not include any real-world example, however it should be mentioned that many of problems lead to models similar to presented in Examples 7.1 and 7.2. Tables 1 and 2 show some quantitative details of the experiments.

7.1 Mutual Exclusion

The elementary net system of Figure 2 models the well-known mutual exclusion problem. The system consists of $n + 1$ processes (where $n \geq 2$) of which n compete for the access to the shared resource and one, called the permission process, guards so that no two processes use the resource simultaneously. The presence of a token in the place labelled by w_i means that the i-th process is waiting for the access to the critical section while the token in c_i means that the i-th process has acquired the permission and entered the critical section. The place r_i models the unguarded part of the process and the presence of token in place p indicates that the resource is available.

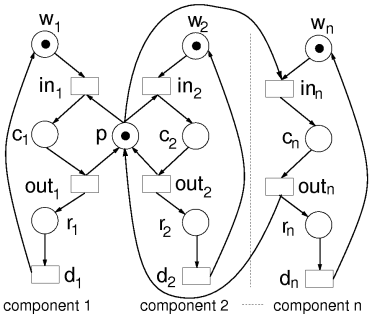


Fig. 2. Mutual exclusion

The Kripke structure constructed for 3 processes along the lines of Subsection 6.2 is presented in Figure 1. Let us consider the formula $\varphi_1^b = \forall_{\Theta \leq b} EF(\neg p \wedge EG^{\leq \Theta} c_1)$. We explore the validity of this formula with respect to the value of b. We can see that in order for the restricted EG operator to hold we need to have a path on which the first process enters its critical section and then other processes execute their local transitions d_i.

Let us explain how the verification works for this formula. For example, for 3 processes and $b = 2$, first the processes 2 and 3 enter their places r_2 and r_3 resp., then the process 1 enters its place c_1 and then 2 and 3 execute d_2 and d_3 respectively along the path of the length 2 on which c_1 holds. Notice that for $b = 3$ this formula does not hold in this model. Note that the non-parameterized counterpart of the formula φ_1, i.e., $EF(\neg p \wedge EGc_1)$ does not hold in our model as there is no cycle in which c_1 is true starting in a state where p is false.

formula	NoP	k	PBMC				MiniSAT	SAT?
			vars	clauses	sec	MB	sec	√/×
φ_1^1	3	2	1063	2920	0.01	1.3	0.003	×
φ_1^1	3	3	1505	4164	0.01	1.5	0.008	√
φ_1^2	3	4	2930	8144	0.01	1.5	0.01	×
φ_1^2	3	5	3593	10010	0.01	1.6	0.03	√
φ_1^2	30	4	37825	108371	0.3	7.4	0.2	×
φ_1^2	30	5	46688	133955	0.4	8.9	0.52	√
φ_1^3	4	6	8001	22378	0.06	2.5	0.04	×
φ_1^3	4	7	9244	25886	0.05	2.8	0.05	√

Table 1. Mutual exclusion, testing the formula φ_1^b

7.2 Dining Philosophers

Another benchmark we consider is the Dining Philosophers Problem. Consider n ($n \geq 2$) philosophers sitting around a round table. Each philosopher has a plate in front of him, and between the two neighbouring plates there lies a fork. Whenever a philosopher eats, he uses both the forks from both the sides of his plate. When a philosopher has finished eating, he lays back both of his forks on the table and starts thinking. The elementary net system modelling the system described above is shown in Fig. 3. The conditions r_i, w_i, s_i denote that i-th philosopher is thinking, waiting for both the forks and eating, respectively; c_i represents that the i-th fork is not taken.

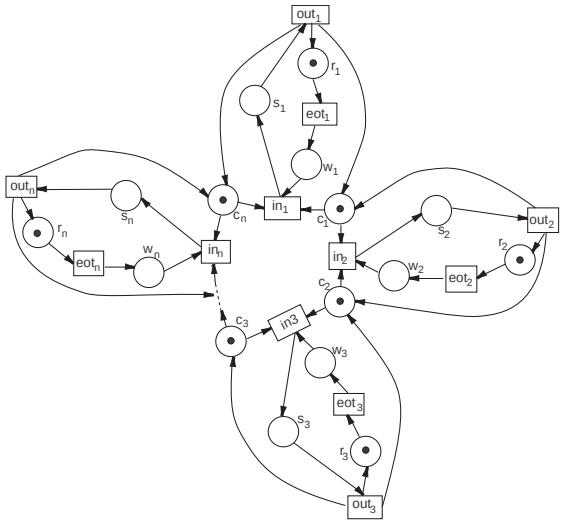


Fig. 3. Dining Philosophers

Let us consider the following properties: $\varphi_2^b = \forall_{\Theta \leq b} EF(s_1 \wedge EG^{\leq \Theta}(\neg c_1 \wedge \neg c_n \wedge \bigwedge_{1<i<n} c_i))$ and $\varphi_3^b = \forall_{\Theta \leq b} EF(s_1 \wedge EG^{\leq \Theta} \bigwedge_{1 \leq i \leq n} \neg c_i)$. The formula φ_2^b expresses that it is possible that in the future there exists a state where for b time units the first philosopher has eaten (therefore his forks are taken) while all the remaining forks are laid on the table. The formula φ_2^b states a similar property, namely that there exists a future state in which for b time units the first philosopher has eaten while all the remaining forks are taken.

Note that φ_3^3 does not hold in the model because there is no path of length 3 along which the first process can stay in the s_1 state.

formula	NoP	k	PBMC				MiniSAT	SAT?
			vars	clauses	sec	MB	sec	√/×
φ_2^1	4	1	1240	3347	0.01	1.5	0.008	×
φ_2^1	4	2	2124	5839	0.02	1.64	0.004	√
φ_2^3	4	1	2518	6821	0.01	1.8	0.004	×
φ_2^3	4	2	4298	11837	0.01	2.01	0.01	√
φ_3^1	4	3	3014	8343	0.02	1.8	0.1	×
φ_3^1	4	4	3898	10385	0.03	1.9	0.2	√
φ_3^2	4	3	4549	12600	0.04	2.07	0.008	×
φ_3^2	4	4	5875	16338	0.06	2.32	0.04	√
φ_3^2	10	9	37981	107724	0.25	7.3	3.78	×
φ_3^2	10	10	42043	119310	0.28	8	8.97	√

Table 2. Dining philosophers, testing the formulae φ_2^b and φ_3^b

7.3 Generic Pipeline Paradigm

The final benchmark we consider is the Generic Pipeline Paradigm model [10]. It consists of three parts, namely: Producer which is able to produce data ($ProdReady$), Consumer being able to receive data ($ConsReady$) and a chain of n intermediate Nodes – having data receiving ($Node_iReady$), processing ($Node_iProc_j$), and sending ($Node_iSend$) capabilities. Notice that the example

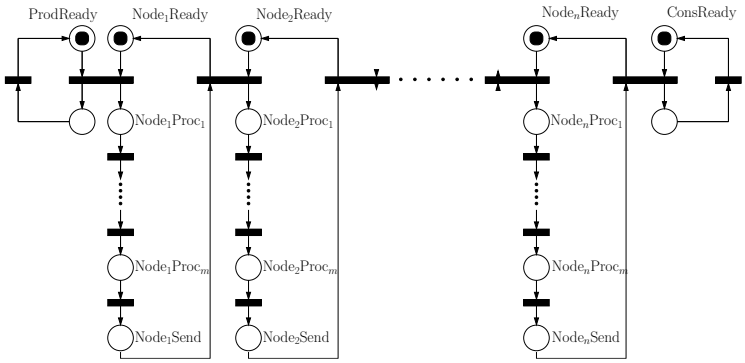


Fig. 4. Generic Pipeline Paradigm

can be scaled in order to see how the size of the system influences performance, and whether the truth of the verified formulae is affected. In particular, we extend the "processing" states with subsequent transitions, which model time needed to process data.

Let us consider the following property:

$$\varphi_4^{n,m} = \forall_{\Theta \leq nm-1} EFEG^{\leq \Theta}(\neg ProdReady \wedge \neg ConsReady \wedge \bigvee_{i=1}^{n} \neg Node_i Ready).$$

The intuitive meaning is that it is possible that for some state of the system, during some time (bounded by $nm-1$) neither Producer is able to produce, nor Consumer is able to receive, while the intermediate Node chain is processing or transferring data. Note that the CTL nonsuperscripted counterpart does not hold in the model, as the data always eventually will reach Consumer.

formula	n	m	k	PBMC				MiniSAT	SAT?
				vars	clauses	sec	MB	sec	√/×
$\varphi_4^{2,1}$	2	1	6	4086	11315	0.03	2.07	0.008	×
$\varphi_4^{2,1}$	2	1	7	4696	13079	0.036	2.7	0.02	√
$\varphi_4^{2,2}$	2	2	8	5980	16811	0.06	2.6	0.01	×
$\varphi_4^{2,2}$	2	2	9	13484	37927	0.08	2.7	0.06	√
$\varphi_4^{2,3}$	2	3	10	8844	24873	0.07	2.9	0.02	×
$\varphi_4^{2,3}$	2	3	11	9776	27509	0.08	3.1	0.2	√
$\varphi_4^{3,1}$	3	1	8	7416	20739	0.04	2.6	0.01	×
$\varphi_4^{3,1}$	3	1	9	8292	23207	0.03	2.7	0.07	√
$\varphi_4^{3,2}$	3	2	11	20025	56568	0.12	4.6	0.04	×
$\varphi_4^{3,2}$	3	2	12	21768	61517	0.14	4.7	0.43	√
$\varphi_4^{10,1}$	10	1	24	74488	212315	0.4	13.2	0.43	×
$\varphi_4^{10,1}$	10	1	25	77548	221055	0.52	13.6	19.2	√
$\varphi_4^{10,2}$	10	2	32	111844	320863	1.6	37.3	0.98	×
$\varphi_4^{10,2}$	10	2	33	230812	662175	1.64	38.4	23.1	√

Table 3. Generic Pipeline Paradigm, testing the formula $\varphi_4^{n,m}$

8 Conclusions

In this paper we showed how parametric model checking can be performed by means of Bounded Model Checking. We presented an implementation and tested it against some benchmarks. Our work is still in its preliminary phase and can be extended in several directions. One of them is to investigate the remaining parametric logics presented in [6], of which General Parametric CTL (GPCTL) seems to be the most interesting. The formulae of GPCTL allow for referring to the number of occurrences of some event. In case of GPCTL, the computational

complexity of the model checking problem is at least NP-complete, which is likely to make the BMC approach especially fruitful. Another possibility is to include parameters to the model. Introducing the real time can also be considered, given it has been done for non-parametric BMC.

References

1. R. Alur, T. Henzinger, and M. Vardi. Parametric real-time reasoning. In *Proc. of the 25th Ann. Symp. on Theory of Computing (STOC'93)*, pages 592–601. ACM, 1993.
2. M. Benedetti and A. Cimatti. Bounded model checking for Past LTL. In *Proc. of the 9th Int. Conf. on Tools and Algorithms for the Construction and Analysis of Systems (TACAS'03)*, volume 2619 of *LNCS*, pages 18–33. Springer-Verlag, 2003.
3. A. Biere, A. Cimatti, E. Clarke, M.Fujita, and Y. Zhu. Symbolic model checking using SAT procedures instead of BDDs. In *Proc. of the ACM/IEEE Design Automation Conference (DAC'99)*, pages 317–320, 1999.
4. A. Biere, A. Cimatti, E. Clarke, and Y. Zhu. Symbolic model checking without BDDs. In *Proc. of the 5th Int. Conf. on Tools and Algorithms for the Construction and Analysis of Systems (TACAS'99)*, volume 1579 of *LNCS*, pages 193–207. Springer-Verlag, 1999.
5. V. Bruyére, E. Dall'Olio, and J-F. Raskin. Durations and parametric model-checking in timed automata. *ACM Transactions on Computational Logic*, 9(2):1–21, 2008.
6. E. A. Emerson and R. Trefler. Parametric quantitative temporal reasoning. In *Proc. of the 14th Symp. on Logic in Computer Science (LICS'99)*, pages 336–343. IEEE Computer Society, July 1999.
7. K. Heljanko. Bounded reachability checking with process semantics. In *Proc. of the 12th Int. Conf. on Concurrency Theory (CONCUR'01)*, volume 2154 of *LNCS*, pages 218–232. Springer-Verlag, 2001.
8. T. Hune, J. Romijn, M. Stoelinga, and F. Vaandrager. 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*, pages 189–203. Springer-Verlag, 2001.
9. MiniSat. http://www.cs.chalmers.se/Cs/Research/FormalMethods/MiniSat, 2006.
10. D. Peled. All from one, one for all: on model checking using representatives. In *Computer Aided Verification, 5th International Conference (CAV'93)*, pages 9409–423. Springer-Verlag, July 1993.
11. W. Penczek, B. Woźna, and A. Zbrzezny. Bounded model checking for the universal fragment of CTL. *Fundamenta Informaticae*, 51(1-2):135–156, 2002.
12. J-F. Raskin and V. Bruyère. Real-time model checking: Parameters everywhere. In *Proc. of the 23rd Conf. on Foundations of Software Technology and Theoretical Computer Science (FSTTCS'03)*, volume 2914 of *LNCS*, pages 100–111. Springer-Verlag, 2003.
13. B. Woźna. ACTL* properties and bounded model checking. In L. Czaja, editor, *Proc. of the Int. Workshop on Concurrency, Specification and Programming (CS&P'03)*, volume 2, pages 591–605. Warsaw University, 2003.
14. A. Zbrzezny. Improving the translation from ECTL to SAT. *Fundamenta Informaticae*, 85(1-4):513–531, 2008.

9 Appendix 1

Let η be a linear expression. In what follows, by $Parameters(\eta)$ we denote the set of all the parameter variables present in η. If $i \in \mathbb{N}$ and Θ is a parameter variable, then by v^i_Θ we denote a valuation satisfying $v^i_\Theta(\Theta) = i$. A formula $\alpha(\Theta)$ containing exactly one free variable Θ is called a *one–parameter* formula.

Lemma 1. *Let M be a model and $Q_{1\Theta_1 \leq c_1} \dots Q_{n\Theta_n \leq c_n} \alpha(\Theta_1, \dots, \Theta_n)$ be a sentence of* PRTCTL*, where $Q_i \in \{\forall, \exists\}$, $c_i \in \mathbb{N}$, and $\alpha(\Theta_1, \dots, \Theta_n) \in$* vRTCTL*. Then $M, s \models Q_{1\Theta_1 \leq c_1} \dots Q_{n\Theta_n \leq c_n} \alpha(\Theta_1, \dots, \Theta_n)$ iff*
$M, s \models Q_{1\Theta_1 \leq min(c_1, |M|)} \dots Q_{n\Theta_n \leq min(c_n, |M|)} \alpha(\Theta_1, \dots, \Theta_n)$.

Proof. Throughout this proof we assume $k = |M|$. Let ψ be a formula of vRTCTL. Let υ be a parameter valuation such that $\upsilon(\Theta') > k$ for some parameter Θ'. Define another parameter valuation υ' such that:

$$\upsilon'(\Theta) = \begin{cases} \upsilon(\Theta), & \text{for } \Theta \neq \Theta' \\ k, & \text{for } \Theta = \Theta'. \end{cases} \tag{1}$$

We prove that $M, s \models_\upsilon \psi \iff M, s \models_{\upsilon'} \psi$ for each state $s \in S$. The proof goes by structural induction. The cases of $\psi = p$, $\psi = \neg\alpha$, $\psi = \alpha \vee \gamma$, $\psi = \alpha \wedge \gamma$ and $\psi = EX\alpha$ are easy to prove.

Let us focus on proving $M, s \models_\upsilon EG^{\leq\eta}\alpha \iff M, s \models_{\upsilon'} EG^{\leq\eta}\alpha$. If $\Theta' \notin Parameters(\eta)$, then the equivalence is valid by $\upsilon(\eta) = \upsilon'(\eta)$ and the inductive assumption. Assume that $\Theta' \in Parameters(\eta)$, and π is a path such that $\pi(0) = s$ and $M, \pi(i) \models_\upsilon \alpha$ for all $i \leq \upsilon(\eta)$. As $\upsilon'(\eta) \geq k$, there exists an $l \leq \upsilon'(\eta)$ such that $\pi(l) = \pi(n)$ for some $n < l$. Therefore we define the path π' as follows:

$$\pi'(i) = \begin{cases} \pi(i), & \text{for } i < l \\ \pi(n + (i-l) mod (l-n)), & \text{for } i \geq l. \end{cases} \tag{2}$$

As $\pi'(0) = \pi(0) = s$ and $M, \pi'(i) \models_{\upsilon'} \alpha$ for all $i \in \mathbb{N}$, by the inductive assumption we obtain $M, s \models_{\upsilon'} EG^{\leq\eta}\alpha$.

Now, let us move to the case of $\psi = E\alpha U^{\leq\eta}\beta$. We deal with the case of $\Theta' \in Parameters(\eta)$ only. If $M, s \models_\upsilon E\alpha U^{\leq\eta}\beta$, then there exists a path π with $\pi(0) = s$, such that for some $i \leq \upsilon(\eta)$ we have that $M, \pi(i) \models_\upsilon \beta$ and $M, \pi(j) \models_\upsilon \alpha$ for all $j < i$. If $i \leq \upsilon'(\eta)$, then $M, s \models_{\upsilon'} E\alpha U^{\leq\eta}\beta$ follows immediately from the inductive assumption. If $i > \upsilon'(\eta)$, then notice that from $\upsilon'(\eta) \geq k$ we get $i > k$. Therefore, there exist $n, m \in \mathbb{N}$ such that $n < m \leq i$ and $\pi(n) = \pi(m)$. By removing the subsequence $\pi(n+1), \dots, \pi(m)$ from π we obtain a new path consisting of the states $\pi(0), \pi(1), \dots, \pi(n), \pi(m+1), \pi(m+2), \dots$. By consecutive elimination of such subsequences we eventually arrive at the path π' such that $\pi'(0) = s$, $\pi'(j) \models_\upsilon \beta$ for some $j < k$ and $\pi'(l) \models_\upsilon \alpha$ for all $l < j$. Therefore, by the inductive assumption we obtain $M, s \models_{\upsilon'} E\alpha U^{\leq\eta}\beta$.
By induction on the number of parameters we get that for formulae $\psi \in$ vRTCTL, the parameter valuation υ and valuation υ' defined as $\upsilon'(\Theta) = min(\upsilon(\Theta), k)$ for all the parameters Θ, we have $M, s \models_\upsilon \psi \iff M, s \models_{\upsilon'} \psi$.

In order to prove the general case, consider a one–parameter vRTCTL formula $g(\Theta)$. We have

$$M, s \models \forall_{\Theta \leq c} g(\Theta) \iff \bigwedge_{0 \leq i \leq c} M, s \models_{v^i_\Theta} g(\Theta).$$

Based on what we have already proven concerning vRTCTL formulae, we can substitute v^i_Θ by $v^i_\Theta[\Theta \leftarrow min(i,k)]$ in the right-hand side of the above formula, obtaining:

$$\bigwedge_{0 \leq i \leq c} M, s \models_{v^i_\Theta[\Theta \leftarrow min(i,k)]} g(\Theta) \iff \bigwedge_{0 \leq i \leq min(c,k)} M, s \models_{v^i_\Theta} g(\Theta).$$

Therefore, we have $M, s \models \forall_{\Theta \leq c} g(\Theta) \iff M, s \models \forall_{\Theta \leq min(c,k)} g(\Theta)$. The equivalence $M, s \models \exists_{\Theta \leq c} g(\Theta) \iff M, s \models \exists_{\Theta \leq min(c,k)} g(\Theta)$ is proven in the similar way. To conclude, notice that for formula of PRTCTL

$$h = Q_1 \Theta_1 \leq min(c_1,k) \ldots Q_t \Theta_t \leq min(c_t,k) f(\Theta_1, \ldots, \Theta_t),$$

where $Q_i \in \{\exists, \forall\}$ and $f \in$ vRTCTL, we can define an one–parameter subformula $\mu(\Theta_1) = Q_2 \Theta_2 \leq min(c_2,k) \ldots Q_t \Theta_t \leq min(c_t,k) f(\Theta_1)$. The thesis of the lemma follows by induction on the number of parameters.

Lemma 2. *Let M_k be the k-model of M, s – a state, υ – a parameter valuation, and α – a formula of* vRTECTL. *Then, the following conditions hold:*

1. $\forall_{l \geq k} \left(M_k, s \models_\upsilon \alpha \text{ implies } M_l, s \models_\upsilon \alpha \right)$,
2. $M_k, s \models_\upsilon \alpha$ *implies* $M, s \models_\upsilon \alpha$,
3. $M, s \models_\upsilon \alpha$ *implies* $M_{|M|}, s \models_\upsilon \alpha$.

Proof. We start with the first implication. We omit the basic case of propositional variables and their negations as trivial. Let α, β be formulae satisfying the considered property, then $M_k, s \models_\upsilon \alpha \wedge \beta$ iff $M_k, s \models_\upsilon \alpha$ and $M_k, s \models_\upsilon \beta$, from which by the inductive assumption follows that $M_l, s \models_\upsilon \alpha$ and $M_l, s \models_\upsilon \beta$ which is equivalent to $M_l, s \models_\upsilon \alpha \wedge \beta$. The case of disjunction follows similarly. For the case of $M_k, s \models_\upsilon EX\alpha$ let us notice that a finite path π_k in M_k such that $\pi_k(0) = s$ and $M_k, \pi_k(1) \models_\upsilon \alpha$ is a prefix of some finite path in M_l, therefore $M_l, s \models_\upsilon EX\alpha$. Now, the case of $M_k, s \models_\upsilon EG^{\leq \eta}\alpha$ can be divided into two subcases. If $\upsilon(\eta) < k$, then define π_l as any finite path in M_l containing π_k as a prefix. If $\upsilon(\eta) \geq k$, then there exists an infinite path π along which α holds (see the proof of Lemma 1). Define π_l as a prefix of length l of π. In both the cases α holds along π_l, therefore $M_l, s \models_\upsilon EG^{\leq \eta}\alpha$. In the final case of $M_k, s \models_\upsilon E\alpha U^{\leq \eta}\beta$ let us notice that if for some path π_k in M_k we have $\pi_k(0) = s$, $M_k, \pi_k(i) \models_\upsilon \beta$, $M_k, \pi_k(j) \models_\upsilon \alpha$ for all $0 \leq j < i$ and $i \leq \upsilon(\eta)$, then the same path is a prefix of some path in M_l satisfying $M_l, s \models_\upsilon E\alpha U^{\leq \eta}\beta$.

The second implication is proven by the structural induction, similarly to the above reasoning. The only nontrivial case is when we consider $M_k, s \models_\upsilon EG^{\leq \eta}\alpha$. Notice that if $\upsilon(\eta) < k$, then a finite path along which α is satisfied up to $\upsilon(\eta)$

steps can be extended to an infinite path (due to the fact that the transition relation is total). On the other hand, if $\upsilon(\eta) \geq k$, then the finite path along which α is satisfied contains a loop, and can be transformed into an infinite path by traversing the loop, as in the proof of Lemma 1.

In the proof of the last implication we focus on two modalities, starting from the case of $M, s \models_\upsilon EG^{\leq\eta}\alpha$. If this formula is valid, then there exists an infinite path π in M such that $\pi(0) = s$ and $M, \pi(i) \models_\upsilon \alpha$ for all $0 \leq i \leq \upsilon(\eta)$. There are two possible subcases. We omit the easier subcase when $\upsilon(\eta) \leq |M|$. If $\upsilon(\eta) > |M|$, then the path π contains a loop with a return from position $l \leq |M|$. Therefore, by unwinding a loop in π we can create an infinite path such that α is satisfied in each of its position – also along a prefix of length $|M|$. Considering $M, s \models_\upsilon E\alpha U^{\leq\eta}\beta$, notice that there exists an infinite path π in M such that $\pi(0) = s$ and for some $l \leq \upsilon(\eta)$ we have $M, \pi(l) \models_\upsilon \beta$ and $M, \pi(i) \models_\upsilon \alpha$ for all $0 \leq i < l$. Again, there are two subcases. We omit the easier case of $l \leq |M|$. If $l > |M|$, then there exist $n, m \in \mathbb{N}$ such that $n < m \leq l$ and $\pi(n) = \pi(m)$. By consecutive elimination of blocks of type $\pi(n+1), \ldots, \pi(m)$, as in the proof of Lemma 1, we eventually obtain the path π' such that $\pi'(0) = s$, $\pi'(j) \models_\upsilon \beta$ for some $j \leq |M|$, and $\pi'(i) \models_\upsilon \alpha$ for all $i < j$.

Lemma 3. *Let M be a model, s – a state and α – a* PRTECTL *sentence. Then, the following conditions hold:*

1. $\forall_{l \geq k}\ \big(M_k, s \models \alpha$ *implies* $M_l, s \models \alpha\big)$,
2. $M_k, s \models \alpha$ *implies* $M, s \models \alpha$,
3. $M, s \models \alpha$ *implies* $M_{|M|}, s \models \alpha$.

Proof. Notice that due to the results of Lemma 1 and Theorem 1 we can assume that all the quantifiers are bounded in α.

Let us focus on the first implication. If α is a closed formula without existential or universal quantifiers, then α can be obtained from some vRTECTL formula ϕ through substitution of parameters by means of some parameter valuation υ. Therefore, as $M_k, s \models \alpha$ iff $M_k, s \models_\upsilon \phi$, then using Lemma 2 we obtain $M_l, s \models_\upsilon \phi$, from which $M_l, s \models \alpha$ follows. Now, assume that $\alpha(\Theta)$ is a one–parameter vRTECTL formula satisfying the property considered. Then

$$M_k, s \models \forall_{\Theta \leq c}\alpha(\Theta) \iff \bigwedge_{0 \leq i \leq c} M_k, s \models_{\upsilon^i_\Theta} \alpha(\Theta) \Rightarrow$$

$$\bigwedge_{0 \leq i \leq c} (M_l, s \models_{\upsilon^i_\Theta} \alpha(\Theta)) \iff M_l, s \models \forall_{\Theta \leq c}\alpha(\Theta).$$

We deal with the existential quantifier in a similar way, noticing that $\min(c, k) \leq \min(c, l)$. The proof follows by induction on the number of quantifiers in a formula. The remaining two implications are proven in the similar manner.

Lemma 5. *Let $\alpha \in$* vRTECTL*, M_k be the k-model of M, and υ – a parameter valuation. For any state s present in some path of M_k, $M_k, s \models_\upsilon \alpha$ if and only if there exists a submodel M'_k of M_k such that $M'_k, s \models_\upsilon \alpha$ and $|Path'_k| \leq g_k(\alpha, \upsilon)$.*

Proof. The "if" part follows directly from Lemma 4. For the "only if" part, we use structural induction. The base cases of $M_k, s \models_\upsilon p$ and $M_k, s \models_\upsilon \neg p$ are trivial. Notice that $M_k, s \models_\upsilon \alpha \vee \beta$ iff $M_k, s \models_\upsilon \alpha$ or $M_k, s \models_\upsilon \beta$. From the inductive assumption there is M'_k such that $M'_k, s \models_\upsilon \alpha$ and $|Path'_k| \leq g_k(\alpha, \upsilon)$, or $M'_k, s \models_\upsilon \beta$ and $|Path'_k| \leq g_k(\beta, \upsilon)$. Thus $M'_k, s \models_\upsilon \alpha \vee \beta$ and $|Path'_k| \leq max(g_k(\alpha, \upsilon), g_k(\beta, \upsilon)) = g_k(\alpha \vee \beta, \upsilon)$.

Recall that $M_k, s \models_\upsilon \alpha \wedge \beta$ iff $M_k, s \models_\upsilon \alpha$ and $M_k, s \models_\upsilon \beta$. By the inductive assumption there exist submodels M''_k and M'''_k of M_k such that $M''_k, s \models_\upsilon \alpha, |Path''_k| \leq g_k(\alpha, \upsilon)$ and $M'''_k, s \models_\upsilon \beta, |Path'''_k| \leq g_k(\beta, \upsilon)$. Consider the submodel M'_k such that $Path'_k = Path''_k \cup Path'''_k$, then from Lemma 4 and the inclusions $M''_k \subseteq M'_k, M'''_k \subseteq M'_k$ we obtain $M'_k, s \models_\upsilon \alpha \wedge \beta$. Moreover, $|Path'_k| \leq |Path''_k| + |Path'''_k| \leq g_k(\alpha, \upsilon) + g_k(\beta, \upsilon) = g_k(\alpha \wedge \beta, \upsilon)$.

Now consider the case of $M_k, s \models_\upsilon EX\alpha$. From the definition of bounded semantics we obtain that there is some k–path $\pi_k \in Path_k$, such that $\pi_k(0) = s$ and $M_k, \pi_k(1) \models_\upsilon \alpha$. From the inductive assumption there exists a submodel M''_k such that $M''_k, \pi_k(1) \models_\upsilon \alpha$ and $|Path''_k| \leq g_k(\alpha, \upsilon)$. Define the submodel M'_k having $Path'_k = Path''_k \cup \{\pi_k\}$. Then from $\pi_k \in Path'_k$ and Lemma 4 we obtain $M'_k, \pi_k(1) \models_\upsilon \alpha$, therefore $M'_k, s \models_\upsilon EX\alpha$. Moreover, $|Path'_k| \leq |Path''_k| + 1 \leq g_k(\alpha, \upsilon) + 1 = g_k(EX\alpha, \upsilon)$.

Let us move to the case of $M_k, s \models_\upsilon EG^{\leq \eta}\alpha$. We have to consider two subcases. In the first case of $\upsilon(\eta) \leq k$, there exists a path $\pi_k \in Path_k$ such that $\pi_k(0) = s$ and $M_k, \pi_k(i) \models_\upsilon \alpha$ for all $0 \leq i \leq \upsilon(\eta)$. Let M^i_k denote submodel of M_k such that $M^i_k, \pi_k(i) \models_\upsilon \alpha$ and $|Path^i_k| \leq g_k(\alpha, \upsilon)$ for $0 \leq i \leq \upsilon(\eta)$. Define the submodel M'_k such that $Path'_k = \bigcup_{0 \leq i \leq \upsilon(\eta)} Path^i_k \cup \{\pi_k\}$. Then by Lemma 4 we have $M'_k, \pi_k(i) \models_\upsilon \alpha$ for all $0 \leq i \leq \upsilon(\eta)$, thus $M'_k, s \models_\upsilon EG^{\leq \eta}\alpha$. From the inductive assumption we obtain

$$|Path'_k| \leq \sum_{0 \leq i \leq \upsilon(\eta)} |Path^i_k| + 1 \leq (\upsilon(\eta) + 1) g_k(\alpha, \upsilon) + 1 = g_k(EG^{\leq \eta}\alpha, \upsilon).$$

We deal with the subcase of $\upsilon(\eta) > k$ in a similar way.

In the final case of $M_k, s \models_\upsilon E\alpha U^{\leq \eta}\beta$ there exists a path $\pi_k \in Path_k$ and $0 \leq j < min(\upsilon(\eta), k)$ such that $M_k, \pi_k(j) \models_\upsilon \beta$ and $M_k, \pi_k(i) \models_\upsilon \alpha$ for all $0 \leq i \leq j$. From the inductive assumption there exists a submodel M^j_k satisfying $M^j_k, \pi_k(j) \models_\upsilon \beta$ and $|Path^j_k| \leq g_k(\beta, \upsilon)$ and the submodels M^i_k such that $M^i_k, \pi_k(i) \models_\upsilon \alpha$ for all $0 \leq i < j$ and $|Path^i_k| \leq g_k(\alpha, \upsilon)$. Define M'_k such that $Path'_k = \bigcup_{0 \leq l \leq j} Path^l_k \cup \{\pi_k\}$. Then, by Lemma 4 we have $M'_k, \pi_k(i) \models_\upsilon \alpha$ for all $0 \leq i < j$, and $M'_k, \pi_k(j) \models_\upsilon \beta$. As from the latter follows $M'_k, s \models_\upsilon E\alpha U^{\leq \eta}\beta$ and

$$|Path'_k| \leq \sum_{0 \leq i < j} |Path'_k| + |Path^j_k| + 1 \leq j \cdot g_k(\alpha, \upsilon) + g_k(\beta, \upsilon) + 1$$

$$\leq min(\upsilon(\eta), k) g_k(\alpha, \upsilon) + g_k(\beta, \upsilon) + 1 = g_k(E\alpha U^{\leq \eta}\beta, \upsilon),$$

we conclude the proof of this case and of the lemma.

Lemma 6. *Let β be a* PRTECTL *sentence and M_k be the k-model of M. For any state s present in some path of M_k, $M_k, s \models \beta$ if and only if there exists a submodel M'_k of M_k such that $M'_k, s \models \beta$ and $|Path'_k| \leq f_k(\beta)$.*

Proof. Consider a one–parameter formula $\alpha(\Theta) \in$ vRTECTL, and let $i \in \mathbb{N}$. It follows from Lemma 5 that if $M_k, s \models_{v^i_\Theta} \alpha(\Theta)$, then there exists a submodel M^i_k such that $M^i_k, s \models_{v^i_\Theta} \alpha(\Theta)$ and $|Path^i_k| \leq g_k(\alpha, v^i_\Theta)$. Recall that $M_k, s \models \forall_{\Theta \leq c}\alpha(\Theta)$ iff $M_k, s \models_{v^i_\Theta} \alpha(\Theta)$ for all $0 \leq i \leq c$. Using the above observation we obtain that $M^i_k, s \models_{v^i_\Theta} \alpha(\Theta)$ for all $0 \leq i \leq c$, and $|Path^i_k| \leq g_k(\alpha, v^i_\Theta)$. Define M'_k as a submodel such that $Path'_k = \bigcup_{0 \leq i \leq c} Path^i_k$, then by Lemma 4 we have $M'_k, s \models_{v^i_\Theta} \alpha(\Theta)$ for all $0 \leq i \leq c$, therefore $M_k, s \models \forall_{\Theta \leq c}\alpha(\Theta)$. Moreover

$$|Path'_k| \leq \sum_{0 \leq i \leq c} |Path^i_k| \leq \sum_{0 \leq i \leq c} g_k(\alpha, v^i_\Theta) = \sum_{0 \leq i \leq c} f_k(\alpha[\Theta \leftarrow i])$$

$$= f_k(\forall_{\Theta \leq c}\alpha(\Theta)).$$

Similarly, notice that $M_k, s \models \exists_{\Theta \leq c}\alpha(\Theta)$ iff $\bigvee_{0 \leq i \leq min(c,k)} M_k, s \models_{v^i_\Theta} \alpha(\Theta)$. Therefore if the right side of the above equivalence holds true, then based on Lemma 5 we obtain that for some $0 \leq j \leq min(c, k)$ there exists a submodel M'_k such that $M'_k, s \models_{v^i_\Theta} \alpha(\Theta)$ and $|Path'_k| \leq g_k(\alpha, v^i_\Theta)$, from which $M'_k, s \models \exists_{\Theta \leq c}\alpha(\Theta)$ and

$$|Path'_k| \leq max_{i \leq min(c,k)}(g_k(\alpha, v^i_\Theta) = f_k(\exists_{\Theta \leq c}\alpha(\Theta)).$$

In order to prove the general case, notice that the PRTECTL sentence $h = Q_{1\Theta_1 \leq c_1} \ldots Q_{t\Theta_t \leq c_t} f(\Theta_1, \ldots, \Theta_t)$, where f is a formula of vRTECTL and $Q \in \{\forall, \exists\}$ can be rewritten in a form of a one–parameter formula and use the induction on the number of parameters, similarly as in the proof of Lemma 1.

10 Appendix 2

Throughout this section by $\mathbb{N}_+$ we denote the set of positive naturals. Let M be a model, α a formula of vRTECTL, and β a subformula of α. By $[\beta]_k^{[\alpha,m,n,A,\upsilon]}$ we denote the propositional formula $[M]_k^{G_k(\alpha,\upsilon)} \wedge [\beta]_k^{[m,n,A,\upsilon]}$.

Theorem 2. *Let M_k be the k-model of M, υ – a parameter valuation, α – a formula of* vRTECTL *containing at least one modality, and s a state. Then, the following equivalence holds: $M_k, s \models_\upsilon \alpha$ iff $\left[M\right]_k^{\alpha,\upsilon}$ is satisfiable.*

In order to prove the above theorem, we need the following two lemmas. The first one is a counterpart of Lemma 3.1 from [14], and deals with the correctness of a translation.

Lemma A. *Let M be a model, α – a formula of* vRTECTL*, υ – a parameter valuation, and $k \in \mathbb{N}$. For every subformula β of the formula α, every $A \subseteq G_k(\alpha,\upsilon)$ such that $|A| = g_k(\beta,\upsilon)$, every $(m,n) \in \{(0,0)\} \cup \{0,\ldots,k\} \times \mathbb{N}_+$, and every state variables valuation V such that $\hat{V}(w_{m,n})$ is a state of M, the following condition holds: if $V \models [\beta]_k^{[\alpha,m,n,A,\upsilon]}$, then $M_k, \hat{V}(w_{m,n}) \models_\upsilon \beta$.*

Proof. The proof is by induction on the complexity of β. Let A, (m,n), and V be as in the thesis of the theorem.

For the base case of $\beta = p$, where p is an atomic proposition, notice that from $V \models [p]_k^{[\alpha,m,n,A,\upsilon]}$ it follows that $V \models p(w_{m,n})$, i.e., $p \in \mathcal{L}(\hat{V}(w_{m,n}))$, therefore $M_k, \hat{V}(w_{m,n}) \models_\upsilon p$. The case of $\beta = \neg p$ is proven in a similar way.

Now let $\beta = \gamma \wedge \phi$, and let $B = \hat{g}_L(A, g_k(\gamma,\upsilon))$, and $C = \hat{g}_R(A, g_k(\phi,\upsilon))$. Then from $V \models [\gamma \wedge \phi]_k^{[\alpha,m,n,A,\upsilon]}$ it follows that $V \models [\gamma]_k^{[\alpha,m,n,B,\upsilon]}$ and $V \models [\phi]_k^{[\alpha,m,n,C,\upsilon]}$. By the inductive argument we obtain that $M_k, \hat{V}(w_{m,n}) \models_\upsilon \gamma$, and $M_k, \hat{V}(w_{m,n}) \models_\upsilon \phi$, thus $M_k, \hat{V}(w_{m,n}) \models_\upsilon \gamma \wedge \phi$. The case of $\beta = \gamma \vee \phi$ follows similarly.

Notice that the sequence $(\hat{V}(w_{0,j}), \ldots, \hat{V}(w_{k,j}))$, where $j \in G_k(\alpha,\upsilon)$, is a k–path, as it satisfies the propositional formula $[M]_k^{G_k(\alpha,\upsilon)}$. In what follows we denote this k–path by π_j. In the case of $\beta = EX\gamma$, we have $V \models [EX\gamma]_k^{[\alpha,m,n,A,\upsilon]}$ iff $V \models [M]_k^{G_k(\alpha,\upsilon)} \wedge H(w_{m,n}, w_{0,min(A)}) \wedge [\gamma]_k^{[1,min(A),h_X(A),\upsilon]}$. The latter means that $\pi_{min(A)}(0) = \hat{V}(w_{m,n})$, and by the inductive argument $M_k, \pi_{min(A)}(1) \models_\upsilon \gamma$, therefore $M_k, \hat{V}(w_{m,n}) \models_\upsilon EX\gamma$.

In the case of $\beta = EG^{\leq\eta}\gamma$ we have to consider two subcases. In the first subcase, when $\upsilon(\eta) > k$, we have $V \models [EG^{\leq\eta}\gamma]_k^{[\alpha,m,n,A,\upsilon]}$ iff:

$$V \models [M]_k^{G_k(\alpha,\upsilon)} \wedge H(w_{m,n}, w_{0,min(A)}) \wedge L_k(min(A)) \wedge \bigwedge_{j=0}^{k} [\gamma]_k^{[\alpha,j,min(A),h_G(A,k)(j),\upsilon]}.$$

Now the k–path $\pi_{min(A)}$ satisfies $\pi_{min(A)}(0) = \hat{V}(w_{m,n})$, by the inductive argument $M_k, \pi_{min(A)}(j) \models_\upsilon \gamma$ for all $0 \leq j \leq k$, and for some $0 \leq i < k$ we have $\pi_{min(A)}(k) \rightarrow \pi_{min(A)}(i)$ (the k–path is a loop). Therefore, $M_k, \hat{V}(w_{m,n}) \models_\upsilon$

$EG^{\leq\eta}\gamma$. The second subcase, when $\upsilon(\eta) \leq k$, is proven in a similar way, namely in this subcase $V \models [EG^{\leq\eta}\gamma]_k^{[\alpha,m,n,A,\upsilon]}$ means that:

$$V \models [M]_k^{G_k(\alpha,\upsilon)} \wedge H(w_{m,n}, w_{0,min(A)}) \wedge \bigwedge_{j=0}^{\upsilon(\eta)} [\gamma]_k^{[j,min(A),h_G(A,\upsilon(\eta))(j),\upsilon]}.$$

As previously we have $\pi_{min(A)}(0) = \hat{V}(w_{m,n})$, and $M_k, \pi_{min(A)}(j) \models_\upsilon \gamma$ for all $0 \leq j \leq \upsilon(\eta)$, therefore $M_k, \hat{V}(w_{m,n}) \models_\upsilon EG^{\leq\eta}\gamma$.

The final case is when $\beta = E\gamma U^{\leq\eta}\phi$. If $V \models [E\gamma U^{\leq\eta}\phi]_k^{[\alpha,m,n,A,\upsilon]}$, then:

$$V \models [M]_k^{G_k(\alpha,\upsilon)} \wedge \bigvee_{i=0}^{min(\upsilon(\eta),k)} \big([\phi]_k^{[i,min(A),h_U(A,min(\upsilon(\eta),k),g_k(\phi,\upsilon))(min(\upsilon(\eta),k)),\upsilon]}$$

$$\wedge \bigwedge_{j=0}^{i-1} [\gamma]_k^{[j,min(A),h_U(A,min(\upsilon(\eta),k),g_k(\phi,\upsilon))(j),\upsilon]}\big) \wedge H(w_{m,n}, w_{0,min(A)}).$$

Now it suffices to notice that $\pi_{min(A)}$ is such a k–path that $\pi_{min(A)}(0) = \hat{V}(w_{m,n})$, and by the inductive argument there exists $0 \leq j \leq min(\upsilon(\eta), k)$ such that $M_k, \pi_{min(A)}(j) \models_\upsilon \phi$, and $M_k, \pi_{min(A)}(i) \models_\upsilon \gamma$ for all $0 \leq i < j$. Therefore $M_k, \hat{V}(w_{m,n}) \models_\upsilon E\gamma U^{\leq\eta}\phi$.

The following lemma is a counterpart of Lemma 3.2 [14], and deals with completeness of the translation. By $Var(\psi)$ we denote the set of variables present in a propositional formula ψ.

Lemma B. *Let M be a model, α – a formula of* vRTECTL*, υ – a parameter valuation, and $k \in \mathbb{N}$. For every subformula β of the formula α, every $A \subseteq G_k(\alpha,\upsilon)$ such that $|A| = g_k(\beta,\upsilon)$, every $(m,n) \in \{(0,0)\} \cup \{0,\ldots,k\}\times(\mathbb{N}_+ \setminus A)$, and every state s of M, the following condition holds: if $M_k, s \models_\upsilon \beta$, then there exists a valuation V such that $\hat{V}(w_{m,n}) = s$ and $V \models [\beta]_k^{[\alpha,m,n,A,\upsilon]}$.*

Proof. The proof proceeds by induction on the complexity of β. Let A, (m,n), and V be as in the thesis of the theorem, and s be any state of M.

In the base case of $\beta = p$, where p is an atomic proposition, it suffices to take any valuation V, such that $\hat{V}(w_{m,n}) = s$. As $M_k, s \models_\upsilon p$ means that $p \in \mathcal{L}(\hat{V}(w_{m,n}))$, we have $V \models p(w_{m,n})$, therefore $V \models [p]_k^{[\alpha,m,n,A,\upsilon]}$. The case of $\beta = \neg p$ is proven in a similar way.

Now let us consider the case of $\beta = \gamma \wedge \phi$, and let $B = \hat{g}_L(A, g_k(\gamma,\upsilon))$, and $C = \hat{g}_R(A, g_k(\phi,\upsilon))$. By the inductive argument there exist valuations V_1 and V_2 such that $V_1 \models [\gamma]_k^{[\alpha,m,n,B,\upsilon]}$, $V_2 \models [\phi]_k^{[\alpha,m,n,C,\upsilon]}$, and $V_1(w_{m,n}) = V_2(w_{m,n}) = s$. Now it suffices to notice that $Var([\gamma]_k^{[m,n,B,\upsilon]}) \cap Var([\phi]_k^{[m,n,C,\upsilon]}) = w_{m,n}$, therefore there exists a valuation V such that $V \models [\gamma]_k^{[\alpha,m,n,B,\upsilon]} \wedge [\phi]_k^{[\alpha,m,n,C,\upsilon]}$ which means that $V \models [\gamma\wedge\phi]_k^{[\alpha,m,n,A,\upsilon]}$. The case of $\beta = \gamma\vee\phi$ is proven similarly.

Let us move to the case of $\beta = EX\gamma$. If $M_k, s \models_\upsilon EX\gamma$, then there exists a k–path π_k such that $\pi_k(0) = s$, and $M_k, \pi_k(1) \models_\upsilon \gamma$. By the inductive argument there exists a valuation V_1 such that $\hat{V}_1(w_{1,min(A)}) = \pi_k(1)$, and

$V_1 \models [\gamma]_k^{[\alpha,1,min(A),h_X(A)]}$. Notice that $Var([\gamma]_k^{[1,min(A),h_X(A)]})$ contains at most one state variable of form $w_{i,min(A)}$ – namely $w_{1,min(A)}$. Therefore it is possible to find such a valuation V that $\pi_k = (\hat{V}(w_{0,min(A)}), \ldots, \hat{V}(w_{k,min(A)}))$, $\hat{V}(w_{m,n}) = \pi_k(0) = s$, $\hat{V}(w_{1,min(A)}) = \pi_k(1)$ and $V(w_{r,t}) = V_1(w_{r,t})$ for all $w_{r,t} \in Var([\gamma]_k^{[1,min(A),h_X(A)]})$. As this means that $V \models [EX]_k^{[\alpha,m,n,A,\upsilon]}$, we conclude the case.

Now let us consider the case of $\beta = EG^{\leq\eta}\gamma$. For all $0 \leq j \leq min(\upsilon(\eta))$ let us denote $B_j = h_G(A, min(\upsilon(\eta), k))(j)$. If $M_k, s \models_\upsilon EG^{\leq\eta}\gamma$, then there exists a k–path π_k such that $\pi_k(0) = s$, and $M_k, \pi(j) \models_\upsilon \gamma$ for all $0 \leq j \leq min(\upsilon(\eta), k)$. Notice that for all $0 \leq j \leq min(\upsilon(\eta), k)$ we have $|B_j| = g_k(\gamma, \upsilon)$, and $(j, min(A)) \notin B_j$, thus by the inductive argument there exist such valuations V_j that $V_j \models [\gamma]_k^{[\alpha,j,min(A),B_j,\upsilon]}$ for all $0 \leq j \leq min(\upsilon(\eta), k)$. Notice that $Var([\gamma]_k^{[r,min(A),B_r,\upsilon]}) \cap Var([\gamma]_k^{[t,min(A),B_t,\upsilon]}) = \emptyset$ for all $0 \leq r, t \leq min(\upsilon(\eta), k)$ such that $r \neq t$. It means that there exists such a valuation V that $(\hat{V}(w_{0,min(A)}), \ldots, \hat{V}(w_{k,min(A)}))$ is a k–path where for all $0 \leq j \leq min(\upsilon(\eta), k)$ it holds that $\hat{V}(w_{j,min(A)}) = \pi_k(j)$, and $V(w_{r,t}) = V_j(w_{r,t})$ given that $w_{r,t} \in Var([\gamma]_k^{[j,min(A),B_j,\upsilon]})$. As it is easy to see that $V \models L_k(min(A))$ iff π_k is a loop, the case is proven.

The final case of $\beta = E\gamma U^{\leq\eta}\phi$ is proven similarly to the previous one. From $M_k, s \models_\upsilon E\gamma U^{\leq\eta}\phi$ we obtain the existence of such a k–path π_k that $\pi_k(0) = s$, $M_k, \pi_k(j) \models_\upsilon \phi$ for some $0 \leq j \leq min(\upsilon(\eta), k)$, and $M_k, \pi_k(i) \models_\upsilon \gamma$ for all $0 \leq i < j$. Let us denote $B_j = h_U(A, min(\upsilon(\eta), k), g_k(\phi, \upsilon))(j)$ for all $0 \leq j \leq min(\upsilon(\eta), k)$. As previously, notice that $|B_{min(\upsilon(\eta),k)}| = g_k(\phi, \upsilon)$, $(min(\upsilon(\eta), k), min(A)) \notin B_{min(\upsilon(\eta),k)}$, and $|B_i| = g_k(\gamma, \upsilon)$, $(i, min(A)) \notin B_i$ for all $0 \leq i < min(\upsilon(\eta), k)$. To conclude, use the inductive assumption, and the similar reasoning as in the case of $\beta = EG^{\leq\eta}\gamma$.

Now, in order to obtain the proof of Theorem 2 it suffices to apply both the above lemmas with $(m, n) = (0, 0)$, and $\hat{V}(w_{0,0}) = s$.