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 – the parametric extension of the existential version of CTL. To this aim we define a bounded semantics and a translation to SAT for PRTECTL. The implementation of the algorithm for Elementary Net Systems is presented together with some experimental results.

1 Introduction

Bounded Model Checking (BMC) [BCCZ99] 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 [BCC+99], [BC03], [Hel01], [PWZ02], [Woź03]. 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 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 [ET99] 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 two standard problems: the Mutual Exclusion and the Dining Philosophers and test some associated parameterized properties in Section 7. The concluding remarks and an outline of some future work are in Section 8.

2 Related Work

The logics investigated in this paper were introduced in [ET99] while the application of BMC to the existential fragment of the CTL originates from [PWZ02] with a further optimization in [Zbr08]. The work presented in this paper falls into a broad area of the Parametric Model Checking – an ambiguous term which may mean that we deal with the parameters in models (as in [AHV93] and [HRSV01]), in logics (as in [ET99] and [BDR08]) or in both (as in [RB03]). There are two reasons limiting the practical applications of the 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 [BDR08], the joint complexity of the solution is 3EXPTIME. The second – undecidability of the problem for Parametric Timed Automata in general [AHV93] – results in a fact that some of the proposed algorithms need not to stop [HRSV01]. 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 [ET99], both being extensions of Computation Tree Logic (CTL) introduced in [EC82]. 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 [ET99], the formulae are interpreted in standard Kripke structures, which seem to be appropriate for application in many computer science fields, as motivated in [ET99]. 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 the 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 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 a set of all the parameter valuations.

Definition 1. *Let $\mathcal{PV}$ be a set of propositional variables containing the symbol* true. *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 $EX\alpha$, $EG\alpha$, $E\alpha U\beta$,*
4. *if η is a linear expression, α and β are formulae of* vRTCTL, *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 [ET99]. For example $EF^{\leq 3}(w_1 \wedge EG\neg c_1)$ is an RTCTL formula.

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

1. *if $\alpha \in$* vRTCTL, *then $\alpha \in$* PRTCTL,
2. *if $\alpha(\Theta) \in$* vRTCTL *or $\alpha(\Theta) \in$* 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 $a \in \mathbb{N}$.*

The following inclusions hold: CTL $\subseteq$ RTCTL $\subseteq$ vRTCTL $\subseteq$ PRTCTL. In this paper we consider only sentences of PRTCTL.
Additionally we use the derived modalities: $EF\alpha \overset{def}{=} E(trueU\alpha)$, $AF\alpha \overset{def}{=} \neg EG\neg\alpha$, $AX\alpha \overset{def}{=} \neg EX\neg\alpha$, $AG\alpha \overset{def}{=} \neg EF\neg\alpha$ (CTL modalities) and $EF^{\leq\eta}\alpha \overset{def}{=} E(trueU^{\leq\eta}\alpha)$, $AF^{\leq\eta}\alpha \overset{def}{=} \neg EG^{\leq\eta}\neg\alpha$, $AG^{\leq\eta}\alpha \overset{def}{=} \neg EF^{\leq\eta}\neg\alpha$. 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, and 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.*

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 containing the symbol* true. *A Kripke structure (a* model*) is defined as a tuple $(S, \rightarrow, \mathcal{L})$ 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. *$\mathcal{L} : S \longrightarrow 2^{\mathcal{PV}}$ is a labelling function satisfying true $\in \mathcal{L}(s)$ for $s \in S$.*

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 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. $s \models_\upsilon \alpha \wedge \beta \iff s \models_\upsilon \alpha \wedge s \models_\upsilon \beta$,
4. $s \models_\upsilon \alpha \vee \beta \iff s \models_\upsilon \alpha \vee s \models_\upsilon \beta$,
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 valuation υ such that $\upsilon(\Theta_1) = 5$ or from $EF^{\leq \Theta_1 + \Theta_2}(w_1 \wedge EG\neg c_1)$ by 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, $M, s \models_v EG\alpha \iff M, s \models_v EG^{\leq |M|}\alpha$ and $M, s \models_v E\alpha U\beta \iff M, s \models_v E\alpha U^{\leq |M|}\beta$. The proof of this fact is based on the observation that in every path a prefix of length greater or equal than $|M|$ contains a loop.
Recall Theorem 1 from [ET99]:

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

In this paper 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* $Q_i \in \{\forall, \exists\}$ *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.

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 vRTCTL and PRTCTL formulae with superscripted modalities and restricted quantifiers.

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 3 processes (see Subsection 7.1). It is straightforward to check that:

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

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

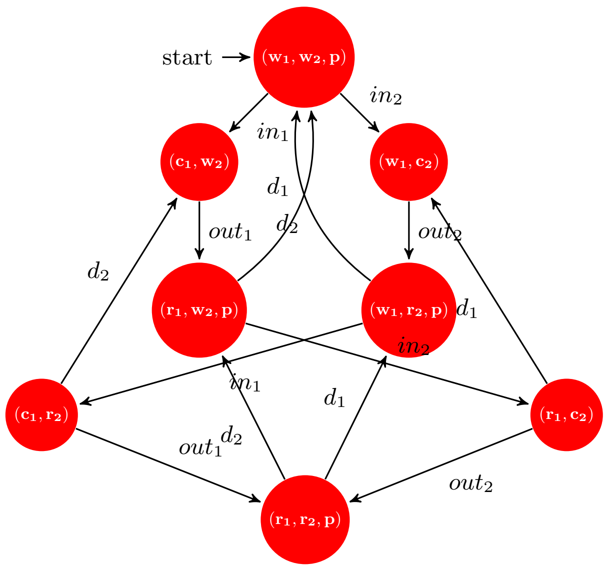
Notice that in the first formula there is no superscript over EG, nevertheless, as we have shown it can be rewritten in the equivalent form:

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

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

$$M, start \models \exists_{\Theta_1 \leq 3} \forall_{\Theta_2 \leq 8} E\big(w_1 U^{\leq \Theta_1} EG^{\leq \Theta_2} r_2\big).$$

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 $s_i \rightarrow s_{i+1}$ 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 a k-model of M and $\pi_k \in Path_k$. Define a function $loop : Path_k \longrightarrow 2^{\mathbb{N}}$ as:*

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

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 a k-model, s – a state, $\alpha, \beta \in$ vRTECTL, p – an atomic proposition, η – a linear expression, and υ – a parameter valuation. By $M_k, s \models_\upsilon \alpha$ let 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 ([PWZ02]) in the points 6 and 7. In case of the point 6, we need to consider two cases. The first case 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 case 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 cases are combined in the 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 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 a 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 a 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 a* k*-model of* M, s *– a state,* υ *– a parameter valuation, and* α *– a formula of* vRTECTL. *Then, the following conditions hold:*

1. $\forall_{l \geq k} \big(M_k, s \models_\upsilon \alpha$ *implies* $M_l, s \models_\upsilon \alpha\big)$,
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. (Sketch) The proof is straightforward. The first and second condition is proved by induction on the length of a formula. In order to prove the third condition notice that each infinite path in the model M contains a looped prefix of length smaller or equal than $|M|$.

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. $\forall_{l \geq k} \big(M_k, s \models \alpha$ *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. (Sketch) The proof is based on the observation that the existential and universal quantifiers can be replaced by disjunctions and conjunctions, respectively. Then, the results of Lemma 2 are applied.

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 algorithm of the bounded model checking is based on the idea of a translation of a part of the model and the formula to a propositional formula. Satisfiability of the result means that the translated formula is true in the model. In the first part of this section we formulate 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 the acceptable efficiency, the algorithm works on submodels of the k-model.

Definition 11. *Let $M_k = (Path_k, \mathcal{L})$ be the k-model. 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 a k-model, 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. (Sketch) The first part of the lemma is easily proved 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 [PWZ02] that in order to determine the truth of an ECTL formula in M_k it is sufficient to consider only submodels of a size 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 – an atomic proposition, 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. $g_k(E\alpha U^{\leq\eta}\beta, \upsilon) = min(\upsilon(\eta), k) \cdot g_k(\alpha, \upsilon) + g_k(\beta, \upsilon) + 1$.

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

1. *if* $\alpha \in \mathrm{RTCTL}$ *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 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* $\alpha \in \mathrm{vRTECTL}$, M_k *be the* k*-model 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. (Sketch) The "if" part follows directly from Lemma 4. For the "only if" part, use induction on the length of a formula and Lemma 4.

Lemma 6. *Let* β *be a* PRTECTL *sentence and* M_k *be the* k*-model. 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. (Sketch) The proof uses the similar observation as in the proof of Lemma 1 – by recalling the results of Lemma 5 for one-parameter vRTECTL formulae and the structural induction on the number of the nonmodal quantifiers.

From Lemmas 5,6, Lemma 4 (notice that the k-model is also a submodel) and Lemmas 2,3 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 2,3 state that if a formula is true in a model, then it is also true in some k-model, or equivalently, by Lemmas 2,3 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 $\alpha \in \mathrm{PRTECTL}$ in the submodel M'_k to the problem of satisfiability of a propositional formula $\lceil\alpha\rfloor_k$ 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 [Zbr08].

Consider the model M. As the number of the states of M is finite, they can be perceived as a bit vectors of the length $r = \lceil log|M|\rceil$. Therefore, we can perceive

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 $\mathcal{SV}$ a set of state variables, then a valuation $V : \mathcal{SV} \longrightarrow \{0,1\}$ naturally extends to the valuation of 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 $\hat{V}(w) \to \hat{V}(w')$ (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 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 [Zbr08] was an essential improvement over the original one of [PWZ02], we follow A. Zbrzezny's approach in our work. We recall the following definitions from [Zbr08].
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 $E\alpha U^{\eta}\beta$. For a more in-depth description we refer to [Zbr08].

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

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

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

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

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

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

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

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

The above encoding is based on the definition of the bounded semantics for vRTECTL – see the 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:

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

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

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

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. 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 a k-model, $\alpha \in$ 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 a* 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. (Sketch) The modification of the proof of Theorem 3.1 from [Zbr08]. The proof is divided into two parts – the proof of correctness and the proof of completeness of the translation, both obtained by the induction on the length of the formula.

Theorem 3. *Let* M_k *be a* 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. (Sketch) 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 following:

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

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:

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

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

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

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

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

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

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

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 the 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 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 [PWZ02]. We consider only the *safe* Petri Nets, i.e., 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 $C[t\rangle$) if $C[t\rangle$ and $D = (C \setminus \bullet t) \cup t\bullet$.*
3. *Let $t_1, \ldots, t_n \in E$. 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 $C_{i-1}[t_i\rangle C_i$ 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 the tokens 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 $post - set(t)$. Denote the initial state C_{in} by s and let $\xi(s) \subseteq P$ be the set of indices of the places in s.
We are in the position to present the definitions:

1. $I_s(w) := \bigwedge_{i \in \xi(s)} w[i] \wedge \bigwedge_{i \in P \setminus \xi(s)} \neg w[i]$,
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 model checking tool. The Elementary Net Systems are used as an input specification formalism, and 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 [PWZ02].

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 were performed on a Linux machine with dual core 1.6 GHz processor. We tested satisfiability using the MiniSAT solver [Min06]. The presented models are relatively simple, yet classical, and the considered formulae were 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.

The Kripke structure constructed for 3 processes along the lines of Subsection

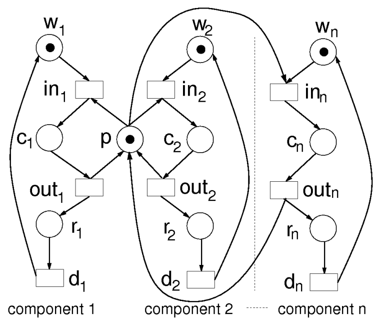


Fig. 2. Mutual exclusion

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. Of course, the order between 2 and 3 may be different. 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	$\surd/\times$
φ_1^1	3	2	1063	2920	0.01	1.3	0.003	$\times$
φ_1^1	3	3	1505	4164	0.01	1.5	0.008	$\surd$
φ_1^2	3	4	2930	8144	0.01	1.5	0.01	$\times$
φ_1^2	3	5	3593	10010	0.01	1.6	0.03	$\surd$
φ_1^2	30	4	37825	108371	0.3	7.4	0.2	$\times$
φ_1^2	30	5	46688	133955	0.4	8.9	0.52	$\surd$
φ_1^3	4	6	8001	22378	0.06	2.5	0.04	$\times$
φ_1^3	4	7	9244	25886	0.05	2.8	0.05	$\surd$

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 backs 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.

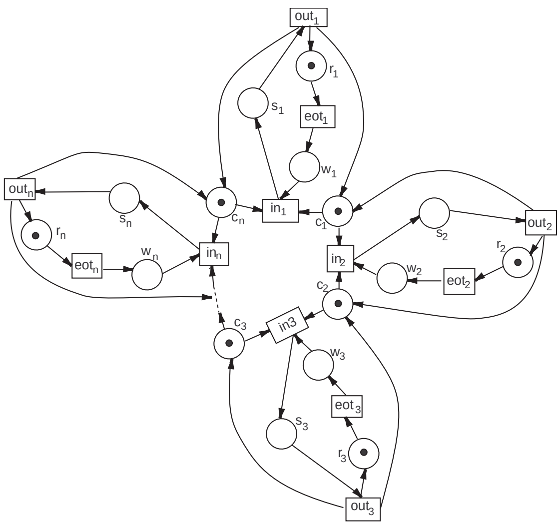


Fig. 3. Dining Philosophers

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 the b time units the first philosopher is eating (therefore his forks are taken) while all the remaining forks are laid on the table. The formula φ_2^b states the similar property, namely that there exists a future state in which for the b time units the first philosopher eats 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	$\surd/\times$
φ_2^1	4	1	1240	3347	0.01	1.5	0.008	$\times$
φ_2^1	4	2	2124	5839	0.02	1.64	0.004	$\surd$
φ_2^3	4	1	2518	6821	0.01	1.8	0.004	$\times$
φ_2^3	4	2	4298	11837	0.01	2.01	0.01	$\surd$
φ_3^1	4	3	3014	8343	0.02	1.8	0.1	$\times$
φ_3^1	4	4	3898	10385	0.03	1.9	0.2	$\surd$
φ_3^2	4	3	4549	12600	0.04	2.07	0.008	$\times$
φ_3^2	4	4	5875	16338	0.06	2.32	0.04	$\surd$
φ_3^2	10	9	37981	107724	0.25	7.3	3.78	$\times$
φ_3^2	10	10	42043	119310	0.28	8	8.97	$\surd$

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

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 [ET99], 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 the parameters to the model. Introducing the real time can also be considered, given that it has been done for non-parametric BMC.

References

[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.

[BC03] M. Benedetti and A. Cimatti, *Bounded model checking for Past LTL*, Proc. of the 9th Int. Conf. on Tools and Algorithms for the Construction and Analysis of Systems (TACAS'03), LNCS, vol. 2619, Springer-Verlag, 2003, pp. 18–33.

[BCC$^+$99] A. Biere, A. Cimatti, E. Clarke, M.Fujita, and Y. Zhu, *Symbolic model checking using SAT procedures instead of BDDs*, Proc. of the ACM/IEEE Design Automation Conference (DAC'99), 1999, pp. 317–320.

[BCCZ99] A. Biere, A. Cimatti, E. Clarke, and Y. Zhu, *Symbolic model checking without BDDs*, Proc. of the 5th Int. Conf. on Tools and Algorithms for the Construction and Analysis of Systems (TACAS'99), LNCS, vol. 1579, Springer-Verlag, 1999, pp. 193–207.

[BDR08] V. Bruyére, E. Dall'Olio, and J-F. Raskin, *Durations and parametric model-checking in timed automata*, ACM Transactions on Computational Logic **9(2)** (2008), 1–21.

[EC82] E. A. Emerson and E. Clarke, *Using branching-time temporal logic to synthesize synchronization skeletons*, Science of Computer Programming **2(3)** (1982), 241–266.

[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.

[Hel01] K. Heljanko, *Bounded reachability checking with process semantics*, Proc. of the 12th Int. Conf. on Concurrency Theory (CONCUR'01), LNCS, vol. 2154, Springer-Verlag, 2001, pp. 218–232.

[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.

[Min06] *MiniSat*, http://www.cs.chalmers.se/Cs/Research/FormalMethods/MiniSat, 2006.

[PWZ02] W. Penczek, B. Woźna, and A. Zbrzezny, *Bounded model checking for the universal fragment of CTL*, Fundamenta Informaticae **51(1-2)** (2002), 135–156.

[RB03] J-F. Raskin and V. Bruyère, *Real-time model checking: Parameters everywhere*, Proc. of the 23rd Conf. on Foundations of Software Technology and Theoretical Computer Science (FSTTCS'03), LNCS, vol. 2914, Springer-Verlag, 2003, pp. 100–111.

[Woź03] B. Woźna, *ACTL* properties and bounded model checking*, Proc. of the Int. Workshop on Concurrency, Specification and Programming (CS&P'03) (L. Czaja, ed.), vol. 2, Warsaw University, 2003, pp. 591–605.

[Zbr08] A. Zbrzezny, *Improving the translation from ECTL to SAT*, Fundamenta Informaticae **85(1-4)** (2008), 513–531.

9 Appendix

The proof of Lemma 1:

Proof. Throughout this proof we denote $k = |M|$. We start with formulae ψ of vPRTCTL. Let υ be a parameter valuation such that $\upsilon(\Theta') = c > k$. Define another valuation

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

We prove that for each state s, $M, s \models_{\upsilon} \psi \iff M, s \models_{\upsilon'} \psi$. The proof goes by the 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 $M, s \models_{\upsilon} EG^{\leq\beta}\alpha \iff M, s \models_{\upsilon'} EG^{\leq\beta}\alpha$.
If $\Theta' \notin Parameters(\beta)$, then the equivalence is valid by $\upsilon(\beta) = \upsilon'(\beta)$ and the inductive assumption. Assume that $\Theta \in Parameters(\beta)$.
If $M, s \models_{\upsilon} EG^{\leq\beta}\alpha$, then there exists a path π such that $\pi(0) = s$ and $M, \pi(i) \models_{\upsilon} \alpha$ for all $i \leq \upsilon(\beta)$. Now, from $\upsilon'(\beta) < \upsilon(\beta)$ and the inductive assumption we have $M, s \models_{\upsilon'} EG^{\leq\beta}\alpha$. Similarly, if $M, s \models_{\upsilon'} EG^{\leq\beta}\alpha$, then there exists a path π such that $\pi(0) = s$ and $M, \pi(i) \models_{\upsilon'} \alpha$ for all $i \leq \upsilon'(\beta)$. As $\upsilon'(\beta) \geq k$, there exists a $l \leq \upsilon'(\beta)$ such that $\pi(l) = \pi(n)$ for some $n < l$. Therefore we can define a path π' as follows:

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

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 $M, s \models_{\upsilon} \alpha$.
Now, let us move to the case of $\psi = E\alpha U^{\leq\beta}\gamma$. We deal with the case of $\Theta' \in Parameters(\beta)$ only. If $M, s \models_{\upsilon} E\alpha U^{\leq\beta}\gamma$, then there exists a path π having $\pi(0) = s$, such that for some $i \leq \upsilon(\beta)$ it occurs that $M, \pi(i) \models_{\upsilon} \gamma$ and $M, \pi(j) \models_{\upsilon} \alpha$ for all $j < i$. If $i \leq \upsilon'(\beta)$, then $M, s \models_{\upsilon'} E\alpha U^{\leq\beta}\gamma$ follows instantly from the inductive assumption. If $i > \upsilon'(\beta)$, notice that from $\upsilon'(\beta) > k$ we get $\pi(i) > k$, thus $\pi(i) = \pi(n)$ for some $n < k < \upsilon'(\beta)$ from which follows $M, s \models_{\upsilon'} E\alpha U^{\leq\beta}\gamma$.
Therefore, by induction on the number of the parameters we get that for formulae $\psi \in$ vPRTCTL, the parameter valuation υ and valuation υ' defined as $\upsilon'(\Theta) = min(\upsilon(\Theta), k)$ we have $M, s \models_{\upsilon} \psi \iff M, s \models_{\upsilon'} \psi$.
In order to prove the general case, consider a one–parameter vPRTCTL formula $g(\Theta)$. We have

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

Based on what we have already proven concerning vPRTCTL formulae, we can substitute $\{\Theta := i\}$ by $\{\Theta := min(i, k)\}$ in the right-hand side of the above formula, obtaining:

$$\bigwedge_{0 \leq i \leq c} M, s \models_{\{\Theta := min(i,k)\}} g(\Theta) \iff \bigwedge_{0 \leq i \leq min(c,k)} M, s \models_{\{\Theta := i\}} g(\Theta).$$

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 proved in the similar way.
Finally, notice that for the formula $h = Q_{1\Theta_1 \leq min(c_1,k)} \ldots Q_{t\Theta_t \leq min(c_t,k)} f$ of PRTCTL where $f \in$ vPRTCTL we can define a one–parameter subformula $\mu(\Theta_1) = Q_{2\Theta_2 \leq min(c_2,k)} \ldots Q_{t\Theta_t \leq min(c_t,k)} f(\Theta_1)$. Now, this formula can be rewritten as a vPRTCTL formula $\hat{\mu}(\Theta_1)$ by substituting universal and existential quantifiers with, appropriately, conjunctions and disjunctions. Therefore, by induction on the number of parameters, the thesis of the lemma follows.