Technical Sciences, 2017, 20(2), 131–147

UNIWERSYTET WARMIŃSKO-MAZURSKI W OLSZTYNIE

Quarterly peer-reviewed scientific journal
ISSN 1505-4675
e-ISSN 2083-4527
TECHNICAL SCIENCES
Homepage: www.uwm.edu.pl/techsci/

COMPARING SAT-BASED BOUNDED MODEL CHECKING RTECTL AND ECTL PROPERTIES

Agnieszka M. Zbrzezny

Institute of Mathematics and Computer Science
Jan Długosz University in Częstochowa

Received 24 October 2016, accepted 2 March 2017, available online 21 March 2017

Key words: SAT-based bounded model checking, ECTL, RTECTL, translation.

Abstract

We compare two SAT-based bounded model checking algorithms for the properties expressed in the existential fragment of a soft real-time computation tree logic (RTECTL) and in the existential fragment of computation tree logic (ECTL). To this end, we use the generic pipeline paradigm (GPP) and the train controller system (TC), the classic concurrency problems, which we formalise by means of a finite transition system. We consider several properties of the problems that can be expressed in both RTECTL and ECTL, and we present the performance evaluation of the mentioned bounded model checking methods by means of the running time and the memory used.

Introduction

The problem of model checking (CLARKE et al. 1999) is to check automatically whether a structure M defines a model for a modal (temporal, epistemic, etc.) formula α. The practical applicability of model checking is strongly limited by the state explosion problem, which means that the number of model states grows exponentially in the size of the system representation. To avoid this problem a number of state reduction techniques and symbolic model checking approaches have been developed, among others (BRYANT 1986, CLARKE et al. 1986, LOMUSCIO et al. 2010, MCMILLAN 1993). The SAT-based bounded model

Correspondence: Agnieszka Zbrzezny, Instytut Matematyki i Informatyki, Wydział Matematyczno-Przyrodniczy, Akademia im. Jana Długosza, 42-200 Częstochowa, al. Armii Krajowej 13/15, phone: 34 361 49 18, e-mail: aga.zbrzezny@gmail.com

checking (SAT-BMC) is one of the symbolic model checking technique designed for finding witnesses for existential properties or counterexamples for universal properties. The first BMC method was proposed in (BIERE et al. 1999), and it was designed for linear time properties. Next, in (PENCZEK et al. 2002) the method has been extended to handle branching time properties. Further extensions of the SAT-based BMC method for real-time systems and multi-agent systems can be found, among others, in PENCZEK and LOMUSCIO (2003), WOŹNA and ZBRZEZNY (2004).

The existential fragment of the computation tree logic (ECTL) (PENCZEK et al. 2002) is a formalism that allow for specification of properties such as: "there is a computation such that α will eventually request", or "there is a computation such that α will never be true", but it is impossible to directly express bounded properties like for example "there is a computation such that α will be true in less than 15 unit time", or "there is a computation such that α will never be averted after 15 unit time", or "there is a computation such that α will always be averted between 10 and 20 unit time". Note however that this bounded properties can be formalised in ECTL by using nested applications of the next state operators, but the resulting ECTL formulae can be very complicated and problematic to work with. The existential fragment of the soft real-time CTL (RTECTL) (EMERSON et al. 1992) defeats this restriction by introducing time-bounded temporal operators, and it supplies a much more compact and convenient way of expressing time-bounded properties. The purpose of this paper is to compare the SAT-based BMC of RTECTL (WOŹNA-SZCZEŚNIAK et al. 2011) properties and the SAT-based BMC of equivalent ECTL (ZBRZEZNY 2008) properties and to present a correct and complete translation from RTECTL to ECTL.

We have expected that the BMC for RTECTL would give better performance than the BMC for the ECTL formulae that result from the translation of RTECTL formulae. This is because the size of these ECTL formulae is exponential in the size of the original RTECTL formulae. However, our experiments have shown that for certain RTECTL formulae BMC can be less effective than BMC for their translation into ECTL formulae. These are the formulae for which the interval at the operator **EG** is finite: then after the translation to ECTL, **EG** operator disappears, and there are more paths, but they are much shorter.

Preliminaries

In this section we first define a Transition System (TS), then we recall syntax and semantics of two logics ECTL and RTECTL.

Transition system. The transition system (BAIER, KATOEN 2008) (also called a model) is a tuple $M = (S, Act, \rightarrow, s^0, AP, L)$ where: S is a set of states, Act is a set of actions, $\rightarrow \subseteq S \times Act \times S$ is a transition relation, $s^0 \subseteq S$ is the initial state, AP is a set of atomic propositions, and $L: S \rightarrow 2^{AP}$ is a labelling function. The transition system is called finite if S, Act and AP are finite. For convenience, we write $s \xrightarrow{\sigma} s'$ instead of $(s,\sigma, s') \in \rightarrow$. Moreover, we write $s \rightarrow s'$ if $s \xrightarrow{\sigma} s'$, or some $\sigma \in Act$.

From now on we assume that a considered model has no terminal states, i.e. for every $s \in S$ there exist $s' \in S$ such that $s \rightarrow s'$. The set of all natural numbers is denoted by N, and the set of all positive natural numbers by N_+. A path in M is an infinite sequence $\pi = (s_0, s_1, ...)$ of states such that $s_i \rightarrow s_{i+1}$ for each $i \in N$. For a path $\pi = (s_0, s_1, ...)$ and $i \in N$, the i-th state of π is defined as $\pi(i)$. The i-th prefix of π is defined as $\pi[...i] = (s_0, s_1, ..., s_i)$, and the i-th suffix of π, denoted by π^i, is defined as $\pi^i = (s_i, s_{i+1}, ...)$. Note that if π is a path in M then the suffix π^i is also a path in M. By $\Pi(s)$ we denote the set of all the paths starting at $s \in S$, and by $\Pi(M)$ we denote the set of all the paths in M.

Syntax of ECTL. The syntax of ECTL formulae over the set AP of atomic propositions is defined by the following grammar:

$$\varphi := true \mid false \mid p \mid \neg p \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \boldsymbol{EX}\varphi \mid \boldsymbol{E}(\varphi \boldsymbol{U} \varphi) \mid \boldsymbol{EG}\varphi,$$

where $p \in AP$ and φ is a formula. The symbols $\boldsymbol{X}$, $\boldsymbol{U}$ and $\boldsymbol{G}$ are the modal operators for "next time", "Until" and "Globally", respectively. The symbol $\boldsymbol{E}$ is the existential path quantifier. The derived basic modalities are defined as follows:

$$\boldsymbol{EF}\alpha \stackrel{\text{def}}{=} \boldsymbol{E}(true \boldsymbol{U} \alpha),\ \boldsymbol{E}(\alpha \boldsymbol{R} \beta) \stackrel{\text{def}}{=} \boldsymbol{E}(\beta \boldsymbol{U} (\alpha \wedge \beta)) \vee \boldsymbol{EG}\beta.$$

Semantics of ECTL. Let M be a model, and φ an ECTL formula. An ECTL formula φ is true in the model M (in symbols $M \models \varphi$) iff $M, s^0 \models \varphi$ (i.e., φ is true at the initial state of the model M), where:

- $M, s \models true$, $M, s, \not\models false$,
- $M, s \models p$ iff $p \in L(s)$,
- $M, s \models \neg p$ iff $p \notin L(s)$,
- $M, s \models \alpha \wedge \beta$ iff $M, s \models \alpha$ and $M, s \models \beta$,
- $M, s \models \alpha \vee \beta$ iff $M, s \models \alpha$ or $M, s \models \beta$,
- $M, s \models \boldsymbol{EX}\alpha$ iff $(\exists \pi \in \Pi(s))(M, \pi(1) \models \alpha)$,
- $M, s \models \boldsymbol{E}(\alpha \boldsymbol{U} \beta)$ iff $(\exists \pi \in \Pi(s))(\exists\, m \geq 0)(M, \pi(m) \models \beta)$ and $(\forall\, j < m)(M, \pi(j) \models \alpha)$,
- $M, s \models \boldsymbol{EG}\alpha$ iff $(\exists \pi \in \Pi(s))(\forall\, m \geq 0)(M, \pi(m) \models \alpha)$.

Syntax of RTECTL. Let $p \in AP$ and I be an interval in $N = \{1, 2, ...\}$ of the form: $[a, b)$ and $[a, \infty)$, for $a, b \in N$. Note that the remaining forms of intervals

can be defined by means of $[a, b)$ and $[a, \infty)$. The language RTECTL is defined by the following grammar:

$$\varphi := true \mid false \mid p \mid \neg p \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \boldsymbol{EX}\varphi \mid \boldsymbol{E}(\varphi \boldsymbol{U_I} \varphi) \mid \boldsymbol{EG_I}\varphi$$

The derived basic modalities are defined as follows:

$$\boldsymbol{EF_I}\alpha = \boldsymbol{E}(true \boldsymbol{U_I} \alpha),\ \boldsymbol{E}(\alpha \boldsymbol{R_I} \beta) = \boldsymbol{E}(\beta \boldsymbol{U_I} (\alpha \wedge \beta)) \vee \boldsymbol{EG_I}\beta.$$

Semantics of RTECTL. Let M be a model, and φ an RTECTL formula. An RTECTL formula φ is true in the model M (in symbols $M \models_{rt} \varphi$) iff $M, s^0 \models_{rt} \varphi$ (i.e., φ is true at the initial state of the model M), where:

- $M, s \models_{rt} true$, $M, s, \models_{rt} false$,
- $M, s \models_{rt} p$ iff $p \in L(s)$,
- $M, s \models_{rt} \neg p$ iff $p \notin L(s)$,
- $M, s \models_{rt} \alpha \wedge \beta$ iff $M, s \models_{rt} \alpha$ and $M, s \models_{rt} \beta$,
- $M, s \models_{rt} \alpha \vee \beta$ iff $M, s \models_{rt} \alpha$ or $M, s \models_{rt} \beta$,
- $M, s \models_{rt} \boldsymbol{EX}\alpha$ iff $(\exists \pi \in \Pi(s))(M, \pi(1) \models_{rt} \alpha)$,
- $M, s \models_{rt} \boldsymbol{E}(\alpha \boldsymbol{U_I} \beta)$ iff $(\exists \pi \in \Pi(s))(\exists\, m \in I)(M, \pi(m) \models_{rt} \beta)$ and $(\forall\, i < m)(M, \pi(i) \models_{rt} \alpha)$,
- $M, s \models_{rt} \boldsymbol{EG_I}\alpha$ iff $(\exists \pi \in \Pi(s))(\forall\, m \in I)(M, \pi(m) \models_{rt} \alpha)$.

Translation from RTECTL into ECTL

In this section we present a translation from the RTECTL language to the ECTL language that is based on the intuitive description of such a translation presented in (CAMPOS 1996, EMERSON et al. 1992). We focused on the existential part of RTCTL only, because we use the BMC method. Moreover, we would like to stress that our semantics of the operator $\boldsymbol{G_I}$ is different than the one presented in (CAMPOS 1996, EMERSON et al. 1992), which was not correctly defined. Further, unlike in (CAMPOS 1996, EMERSON et al. 1992), we present the translation for all types of intervals that can appear together with the temporal operators $\boldsymbol{U_I}$ and $\boldsymbol{G_I}$.

Let $p \in AP$, α and β be formulae of RTECTL, $a \in N$, $b \in N \cup \{\infty\}$ and $a < b$. We define the translation from RTECTL into ECTL as a function $tr: RTECTL \rightarrow ECTL$ in the following way:

- $tr(p) = p$, $tr(\neg p) = \neg p$, $tr(\alpha \wedge \beta) = tr(\alpha) \wedge tr(\beta)$, $tr(\alpha \vee \beta) = tr(\alpha) \vee tr(\beta)$,
- $tr(\boldsymbol{EX}\alpha) = \boldsymbol{EX} tr(\alpha)$,

$$- \; tr(\boldsymbol{E}\alpha\boldsymbol{U}_{[a,b)}\beta) \quad \begin{cases} tr(\alpha) \wedge \boldsymbol{EX}(tr(\boldsymbol{E}(\alpha\boldsymbol{U}_{[a-1,\, b-1)}\beta))) \text{ if } a > 0 \text{ and } 1 < b < \infty \\ tr(\beta) \vee tr(\alpha) \wedge \boldsymbol{EX}(tr(\boldsymbol{E}(\alpha\boldsymbol{U}_{[0,b-1)}\beta))) \text{ if } a > 0 \text{ and } 1 < b < \infty \\ tr(\alpha) \wedge \boldsymbol{EX}(tr(\boldsymbol{E}(\alpha\boldsymbol{U}_{[a-1,\infty)}\beta))) \text{ if } a > 0 \text{ and } b = \infty \\ \boldsymbol{E}(tr(\alpha)\boldsymbol{U}tr(\beta)) \text{ if } a > 0 \text{ and } b = \infty \\ tr(\beta)) \text{ if } a > 0 \text{ and } b = 1 \end{cases}$$

$$- \; tr(\boldsymbol{EG}_{[a,b)}\alpha) \quad \begin{cases} \boldsymbol{EX}(tr(\boldsymbol{EG}_{[a-1,\, b-1)}\alpha)) \text{ if } a > 0 \text{ and } 1 < b < \infty \\ tr(\alpha) \wedge \boldsymbol{EX}(tr(\boldsymbol{EG}_{[a-1,\, b-1)}\alpha)) \text{ if } a = 0 \text{ and } 1 < b < \infty \\ \boldsymbol{EX}(tr(\boldsymbol{EG}_{[a-1,\infty)}\alpha)) \text{ if } a > 0 \text{ and } b = \infty \\ \boldsymbol{EG}tr(\alpha) \text{ if } a = 0 \text{ and } b = \infty \\ tr(\alpha) \text{ if } a = 0 \text{ and } b = 1 \end{cases}$$

Because $\boldsymbol{EF}_{\boldsymbol{I}}\alpha \stackrel{\text{def}}{=} \boldsymbol{E}(true\boldsymbol{U}_{\boldsymbol{I}}\alpha)$ the translation for $\boldsymbol{EF}_{[a,b)}$ can be defined using the translation for $\boldsymbol{E}(\alpha\boldsymbol{U}_{[a,b)}\beta)$.

The following theorem, which can be proven by induction on the length of an RTECTL formula, states correctness of the above translation.

Theorem 1. [Correctness of the translation] Let M be a model, and φ an RTECTL formula. Then $M \models_{rt} \varphi$ if, and only if $M \models tr(\varphi)$.

Proposition 1. Let α be an RTECTL formula, and $a \in N$. Then, $tr(\boldsymbol{EG}_{[a,\infty)}\alpha) = \underbrace{\boldsymbol{EX} \ldots \boldsymbol{EX}}_{a}(\boldsymbol{EG}tr(\alpha))$.

Example 1. $tr(\boldsymbol{EG}_{[3,\infty)}\alpha) = \boldsymbol{EXEXEX}(\boldsymbol{EG}tr(\alpha))$.

Proposition 2. Let α be an RTECTL formula, and $a, b \in N$. If $a < b < \infty$, then $tr(\boldsymbol{EG}_{[a,b)}\alpha) = \underbrace{\boldsymbol{EX} \ldots \boldsymbol{EX}}_{a}\, (tr(\alpha) \wedge \underbrace{\boldsymbol{EX}(tr(\alpha) \wedge \boldsymbol{EX}(tr(\alpha) \ldots))}_{b-a-1})$.

Example 2. $tr(\boldsymbol{EG}_{[3,6)}\alpha) = \boldsymbol{EXEXEX}\, (tr(\alpha) \wedge \boldsymbol{EX}(tr(\alpha) \wedge \boldsymbol{EX}(tr(\alpha))))$.

Proposition 3. Let α and β be RTECTL formulae, and $a \in N$. If $a > 0$, then $tr(\boldsymbol{E}(\alpha\boldsymbol{U}_{[a,\infty)}\beta)) = tr(\underbrace{\alpha) \wedge \boldsymbol{EX}(tr(\alpha) \wedge \boldsymbol{EX}(tr(\alpha)}_{a-1}) \ldots \wedge \boldsymbol{E}(tr(\alpha)\, \boldsymbol{U}(tr(\beta)))))$.

Example 3. $tr(\boldsymbol{E}(\alpha\boldsymbol{U}_{[3,\infty)}\beta)) = tr(\alpha) \wedge \boldsymbol{EX}(tr(\alpha) \wedge \boldsymbol{EX}(tr(\alpha) \wedge \boldsymbol{E}(tr(\alpha)\boldsymbol{U}(tr(\beta)))))$.

Proposition 4. Let α and β be RTECTL formulae and $a, b \in N$. If $a < b < \infty$, then $tr(\boldsymbol{E}(\alpha\boldsymbol{U}_{[a,b)}\beta))$
$= tr(\alpha) \wedge \underbrace{\boldsymbol{EX}(tr(\alpha) \wedge \boldsymbol{EX}(tr(\alpha) \wedge \ldots}_{a} \wedge \underbrace{\boldsymbol{EX}(tr(\beta) \vee tr(\alpha) \wedge \boldsymbol{EX}(tr(\beta) \vee tr(\alpha)) \ldots \wedge \boldsymbol{EX}(tr(\beta)))))))}_{b-a-1}$.

Example 4. $tr(\boldsymbol{E}(\alpha\boldsymbol{U}_{[a,b)}\beta)) =$
$tr(\alpha) \wedge \boldsymbol{EX}(tr(\alpha) \wedge \boldsymbol{EX}(tr(\alpha) \wedge \boldsymbol{EX}(tr(\alpha) \wedge \boldsymbol{EX}(tr(\beta) \vee tr(\alpha) \wedge \boldsymbol{EX}(tr(\beta) \vee tr(\alpha) \wedge \boldsymbol{EX}(tr(\beta)))))))$.

Lemma 1. Let M be a model, and φ an RTECTL formula. Then

$$(\forall s \in S)(M, s \models_{rt} \varphi \Rightarrow M, s \models tr(\varphi)).$$

Proof 1. [Lemma 1] We proceed by induction on the length of formulae. Let $s \in S$, and assume that $M, s \models_{rt} \varphi$. Now consider the following cases:

1. $\varphi \in AP$. Then, since $tr(\varphi) = \varphi$, we have that $tr(\varphi) \in AP$. Therefore, $M, s \models_{rt} \varphi \Leftrightarrow \varphi \in L(s) \Leftrightarrow tr(\varphi) \in L(s) \Leftrightarrow M, s \models tr(\varphi)$.
2. $\varphi = \neg p$, where $p \in AP$. Then, $tr(\varphi) = \varphi$. Therefore, $M, s \models_{rt} \varphi \Leftrightarrow M, s \models_{rt} \neg p \Leftrightarrow p \notin L(s) \Leftrightarrow M, s \models \neg p \Leftrightarrow M,s \models tr(\varphi)$.
3. $\varphi = \alpha \wedge \beta$. By the definition of the satisfiability relation we have that $M, s \models_{rt} \alpha$ and $M, s \models_{rt} \beta$. By the inductive hypothesis, we get $M, s \models$ and $M, s \models tr(\beta)$ Thus, $M, s \models tr(\alpha) \wedge tr(\beta)$, and therefore $M, s \models tr(\alpha \wedge \beta)$.
4. $\varphi = \alpha \vee \beta$. By the definition of the satisfiability relation we have that $M, s \models_{rt} \alpha$ or $M, s \models_{rt} \beta$. By the inductive hypothesis, we get $M, s \models tr(\alpha)$ or $M, s \models tr(\beta)$. Thus, $M, s \models tr(\alpha) \vee tr(\beta)$, and therefore $M, s \models tr(\alpha \vee \beta)$.
5. $\varphi = \boldsymbol{EX}\alpha$. By the definition of the satisfiability relation we have that there exists $\pi \in \Pi(s)$ such that $M, \pi(1) \models_{rt} \alpha$. By the inductive hypothesis, we conclude that $M, \pi(1) \models tr(\alpha)$. Thus, $M \models \boldsymbol{EX} tr(\alpha)$, since $\pi(0) = s$. Therefore $M \models tr(\boldsymbol{EX}\alpha)$, since $tr(\boldsymbol{EX}\alpha) = \boldsymbol{EX} tr(\alpha)$.
6. $\varphi = \boldsymbol{EG_I}\alpha$. By the definition of the satisfiability relation we have that there exists $\pi \in \Pi(s)$ such that $(\forall m \in I)(M, \pi(m) \models_{rt} \alpha)$. By the inductive hypothesis, we conclude that there exists $\pi \in \Pi(s)$ such that
   $(\forall m \in I)(M, \pi(m) \models tr(\alpha))$. (1)
   Now consider the following cases:
   a. $I = [a, \infty)$. From (1) we conclude that $M, s \models \underbrace{\boldsymbol{EX} \ldots \boldsymbol{EX}}_{a} (\boldsymbol{EG} tr(\alpha))$. From this, by the Proposition 1, it follows that $M, s \models tr(\boldsymbol{EG}_{[a,\infty)}\alpha)$.
   b. $I = [a, b)$, where $a < b < \infty$. From (1) we conclude that $M, s \models \underbrace{\boldsymbol{EX} \ldots \boldsymbol{EX}}_{a}(tr(\alpha) \wedge \boldsymbol{EX}(tr(\alpha) \wedge \ldots \wedge \underbrace{\boldsymbol{EX} \ldots \boldsymbol{EX}}_{b-a}(tr(\alpha)))$. From this, by the Proposition 2, we get $M, s \models tr(\boldsymbol{EG}_{[a,b)}\alpha)$.
7. $\varphi = \boldsymbol{E}(\alpha \boldsymbol{U_I} \beta)$. By the definition of the satisfiability relation we have that there exists $\pi \in \Pi(s)$ such that $(\exists m \in I)(M, \pi(m) \models_{rt} \beta)$ and $(\forall i < m)(M, \pi(i) \models_{rt} \alpha)$. By the inductive hypothesis, we conclude that
   $(\exists m \in I)(M, \pi(m) \models_{rt} \beta)$ and $(\forall i < m)(M, \pi(i) \models_{rt} \alpha)$. (2)
   Now consider the following cases:
   a. $I = [a, \infty)$. From (2) we conclude that $M, s \models tr(\alpha) \wedge \underbrace{\boldsymbol{EX}(tr(\alpha) \wedge \boldsymbol{EX}(tr(\alpha)\ldots}_{a-1}$ $\wedge \boldsymbol{E}(tr(\alpha)\boldsymbol{U}(tr(\beta))))$. From this, by Proposition 3, it follows that $M, s \models tr(\boldsymbol{E}(\alpha \boldsymbol{U}_{[a,\infty)}\beta))$.

b. $I = [a, b)$, where $a < b < \infty$. From (2) we conclude that

$$M, s \models tr(\alpha) \wedge \underbrace{\boldsymbol{EX}(tr(\alpha) \wedge \boldsymbol{EX}(tr(\alpha) \wedge \ldots}_{a}$$
$$\wedge \underbrace{\boldsymbol{EX}(tr(\beta) \vee tr(\alpha) \wedge \boldsymbol{EX}(tr(\beta) \vee tr(\alpha) \ldots}_{b-a-1} \wedge \boldsymbol{EX}(tr(\beta)))))).$$

From this, by Proposition 4, it follows that $M, s \models tr(\boldsymbol{E}(\alpha \boldsymbol{U}_{[a,b)} \beta))$.

Lemma 2. Let M be a model, and φ an RTECTL formula. Then

$$(\forall s \in S)(M, s \models tr(\varphi) \Rightarrow M, s \models_{rt} \varphi).$$

Proof 2. [Lemma 2] We proceed by induction on the length of formulae. Let $s \in S$, and assume that $M, s \models tr(\varphi)$. Now consider the following cases:

1. $\varphi \in AP$. Then, since $\varphi = tr(\varphi)$, we have that $tr(\varphi) \in AP$. Therefore, $M, s \models tr(\varphi) \Leftrightarrow \varphi \in L(s) \Leftrightarrow tr(\varphi) \in L(s) \Leftrightarrow M, s \models_{rt} \varphi$.
2. $\varphi = \neg p$, where $p \in AP$. Then, $tr(\varphi) = \varphi$. Therefore, $M, s \models tr(\varphi) \Leftrightarrow M, s \models tr(\neg p) \Leftrightarrow p \notin L(s) \Leftrightarrow M, s \models_{rt} \neg p \Leftrightarrow M,s \models_{rt} \varphi$.
3. $\varphi = \alpha \wedge \beta$. Then, $tr(\varphi) = tr(\alpha \wedge \beta) = tr(\alpha) \wedge tr(\beta)$. By the definition of the satisfiability relation for ECTL we have that $M, s \models tr(\alpha)$ and $M, s \models tr(\beta)$. By the inductive hypothesis, we get $M, s \models_{rt} \alpha$ and $M, s \models_{rt} \beta$ and therefore $M, s \models_{rt} tr(\alpha \wedge \beta)$.
4. $\varphi = \alpha \vee \beta$. Then, $tr(\varphi) = tr(\alpha \vee \beta) = tr(\alpha) \vee tr(\beta)$. By the definition of the satisfiability relation for ECTL we have that $M, s \models tr(\alpha)$ or $M, s \models tr(\beta)$. By the inductive hypothesis, we get $M, s \models_{rt} \alpha$ or $M, s \models_{rt} \beta$ and therefore $M, s \models_{rt} tr(\alpha \vee \beta)$.
5. $\varphi = \boldsymbol{EX}\alpha$. Then, $tr(\varphi) = tr(\boldsymbol{EX}\alpha) = \boldsymbol{EX}tr(\alpha)$. By the definition of the satisfiability relation we have that there exists $\pi \in \Pi(s)$ such that $M, \pi(1) \models tr(\alpha)$. By the inductive hypothesis, we conclude that $M, \pi(1) \models_{rt} \alpha$. Thus, $M, s \models_{rt} \boldsymbol{EX}\alpha$, since $\pi(0) = s$.
6. $\varphi = \boldsymbol{EG_I}\alpha$. Consider the following cases:
   a. $I = [a, \infty)$. From (1) we conclude that $M, s \models \underbrace{\boldsymbol{EX} \ldots \boldsymbol{EX}}_{a} (\boldsymbol{EG}tr(\alpha))$. From this it follows that there exists $\pi \in \Pi(s)$ such that $(\forall m \geq a)(M, \pi(m) \models tr(\alpha))$. By the inductive hypothesis, we conclude that $(\forall m \geq a)(M, \pi(m) \models_{rt} \alpha)$, and therefore $M, s \models_{rt} \boldsymbol{EG_I}\alpha$.
   b. $I = [a, b)$, where $a < b < \infty$. From (2) we get that $M, s \models \underbrace{\boldsymbol{EX} \ldots \boldsymbol{EX}}_{a}(tr(\alpha) \wedge \boldsymbol{EX}(tr(\alpha)) \wedge \ldots \wedge \underbrace{\boldsymbol{EX} \ldots \boldsymbol{EX}}_{b-a}(tr(\alpha)))$. From this, it follows that there exists $\pi \in \Pi(s)$ such that $(\forall m \in [a, b))(M, \pi(m) \models tr(\alpha))$. By the inductive hypothesis, we conclude that $(\forall m \in [a, b))(M, \pi(m) \models_{rt} \alpha)$, Therefore $M \models_{rt} \boldsymbol{EG_I}\alpha$.

7. $\varphi = \boldsymbol{E}(\alpha \boldsymbol{U}_{\boldsymbol{I}} \beta)$. Consider the following cases:
   a. $I = [a, \infty)$. From the Proposition 3, we get that $M, s \models tr(\boldsymbol{E}(\alpha \boldsymbol{U}_{[a,\infty)}\beta)) = tr(\alpha) \wedge \underbrace{\boldsymbol{EX}(tr(\alpha) \wedge \boldsymbol{EX}(tr(\alpha)...}_{a-1} \wedge \boldsymbol{E}(tr(\alpha)\boldsymbol{U}(tr(\beta))))$. From this it follows that there exists $\pi \in \Pi(s)$ such that $(\forall m < a)(M, \pi(m) \models tr(\alpha))$ and $(\exists m \geq a)(M, \pi(m) \models \boldsymbol{E}(tr(\alpha)\boldsymbol{U}tr(\beta))$. By the inductive hypothesis, we conclude that $(\forall m < a)(M, \pi(m) \models_{rt} \alpha)$ and $(\exists m \geq a)(M, \pi(m) \models_{rt} \boldsymbol{E}(\alpha \boldsymbol{U} \beta)$, and therefore $M, s \models \boldsymbol{E}(\alpha \boldsymbol{U}_{\boldsymbol{I}} \beta)$.
   b. $I = [a, b)$, where $a < b < \infty$. From Proposition 4 we get that

$$tr(\boldsymbol{E}(\alpha \boldsymbol{U}_{[a,b)}\beta) = tr(\alpha) \wedge \underbrace{\boldsymbol{EX}(tr(\alpha) \wedge \boldsymbol{EX}(tr(\alpha) \wedge ...}_{a}$$
$$\wedge \underbrace{\boldsymbol{EX}(tr(\beta) \vee tr(\alpha) \wedge \boldsymbol{EX}(tr(\beta) \vee tr(\alpha) ...}_{b-a-1} \wedge \boldsymbol{EX}(tr(\beta)))))))).$$

From this it follows that there exists $\pi \in \Pi(s)$ such that $(\forall m < a)(M, \pi(m) \models tr(\alpha))$ and $(\exists b - a - 1 > m)(M, \pi(m) \models tr(\alpha) \vee tr(\beta))$ and $(\forall m < b)(M, \pi(m) \models tr(\beta))$. By the inductive hypothesis, we conclude that $(\forall m < a)(M, \pi(m) \models_{rt} (\alpha))$ and $(\exists b - a - 1 > m)(M, \pi(m) \models_{rt} \alpha \vee \beta)$ and $(\forall m < b)(M, \pi(m) \models_{rt} \beta)$, and therefore $M, s \models \boldsymbol{E}(\alpha \boldsymbol{U}_{\boldsymbol{I}} \beta)$.

Proof 3. [Theorem 1] This follows from Lemma 1 and Lemma 2.

Bounded model checking

The SAT-based Bounded Model Checking (BMC) is a popular model checking technique for the verification of concurrent systems. Given a model M, an existential modal formula φ, and a non-negative bound k, the SAT-based BMC consists in searching for a non-empty set of paths of the length k that constitute a witness for the checked property φ. In particular, the bounded model checking algorithms generate a propositional formula which is satisfiable if and only if the mentioned set of paths exist. The propositional formula is usually obtained as a combination of a propositional encoding of the unfolding of the transition relation of the given model, and a propositional encoding of the property in question. If the generated propositional formula is not satisfiable, then k is incremented until either the problem becomes intractable due to the complexity of solving the corresponding SAT instance, or k reaches the upper bound of the bounded model checking problem for the language under consideration.

All the SAT-based BMC, so the one for ECTL and RTECTL as well, are based on so called bounded semantics, which are the base of translations of specifications to the SAT-problem. In definitions of the bounded semantics one needs to represent cycles in models in a special way. To this aim k-paths, i.e., finite paths of the length k, and loops are defined. These definitions have

evolved over the last decade, and they have had a major impact on the effectiveness of the BMC encodings.

The SAT-based BMC method for ECTL was introduced in (PENCZEK et al. 2002), and then it was improved in (ZBRZEZNY 2008). In this paper we use the definition and implementation of the SAT-based BMC for ECTL that was presented in (ZBRZEZNY 2008). The SAT-based BMC method for RTECTL was introduced in (WOŹNA-SZCZEŚNIAK 2009), and then it was improved in (WOŹNA-SZCZEŚNIAK et al. 2011). In this paper we use the definition and implementation of the SAT-based BMC for RTECTL that was presented in (WOŹNA-SZCZEŚNIAK et al. 2011).

Experimental results

A Train Controller System

In this section we present a comparison of a performance evaluation of two SAT-based BMC algorithms: for RTECTL, and for ECTL. In order to evaluate the behaviour of the algorithms, we have tested it on several RTECTL properties and equivalent ECTL properties. An evaluation of both BMC algorithms, which have been implemented in C++ is given by means of the running time, the memory used, and the number of generated variables and clauses.

To evaluate the BMC technique for RTECTL and ECTL, we analyse a scalable concurrent system, which is a train controller system (TC). The system consists of a controller, and n trains (for $n \geq 2$), and it is assumed that each train uses its own circular track for travelling in one direction. All trains have to pass through a tunnel, but because there is only one track in the tunnel, arriving trains cannot use it simultaneously. There are signal lights on both sides of the tunnel, which can be either red or green. All trains notify the controller when they request entry to the tunnel or when they leave the tunnel. The controller controls the colour of the signal lights. The controller does not ensure the mutual exclusion property: two trains never occupy the tunnel at the same time.

An automata model of the TC system is shown in the Figure 1.

The specifications for it are given in the existential form, i.e., they are expressed in the RTECTL language:

- $\varphi_1 = \boldsymbol{EF}_{[0,\infty)} (InTunnel_1 \wedge \boldsymbol{EG}_{[1,n+2)}(\bigwedge_{n=1}^{n} \neg InTunnel_i))$ – states that there exists the case that Train_1 is in the tunnel and either it and other train will not be in the tunnel during the next $n+1$ time units, where n is the number of trains.
- $\varphi_2 = \boldsymbol{EF}_{[0,\infty)} (InTunnel_1 \wedge \boldsymbol{EG}_{[1,n+2)}(\bigwedge_{n=1}^{n} \neg InTunnel_i))$ – expresses that there exists the case that Train_1 is in the tunnel or either it or other train will not

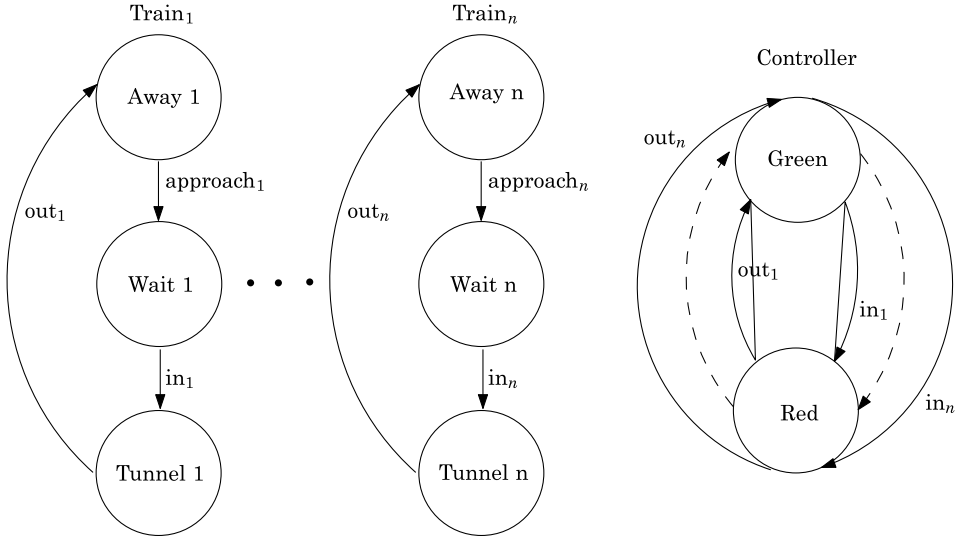

Fig. 1. A network of automata for a train controller system

be in the tunnel during the next $n+1$ time units, where n is the number of trains.

The equivalent ECTL formulae for $n=2$ are:

– $tr(\varphi_1) = \boldsymbol{EF}(InTunnel_1 \wedge \boldsymbol{EX}((\neg\, InTunnel_1 \wedge \neg InTunnel_2) \wedge \boldsymbol{EX}((\neg InTunnel_1 \wedge \neg InTunnel_2) \wedge \boldsymbol{EX}(\neg InTunnel_1 \wedge \neg InTunnel_2)))$,

– $tr(\varphi_2) = \boldsymbol{EF}(InTunnel_1 \vee \boldsymbol{EX}((\neg\, InTunnel_1 \vee \neg InTunnel_2) \wedge \boldsymbol{EX}((\neg InTunnel_1 \vee \neg InTunnel_2) \wedge \boldsymbol{EX}(\neg InTunnel_1 \vee \neg InTunnel_2)))$.

In the Figure 2 we present a comparison of total time usage and total memory usage for the formula φ_1. An analysis of the experimental results for the formula φ_1 leads to the conclusion that BMC for ECTL uses less time and memory comparing to BMC for RTECTL. The reason is that although BMC needs much more paths for the verification in case of ECTL, but these paths are significantly shorter.

In the Figure 3 we present a comparison of total time usage and total memory usage for the formula φ_2. An observation of the experimental results for the formula φ_2 leads to the conclusion that the BMC for RTECTL uses less time and memory comparing to the BMC for ECTL. The reason is that the BMC needs much more paths for verification in case of ECTL.

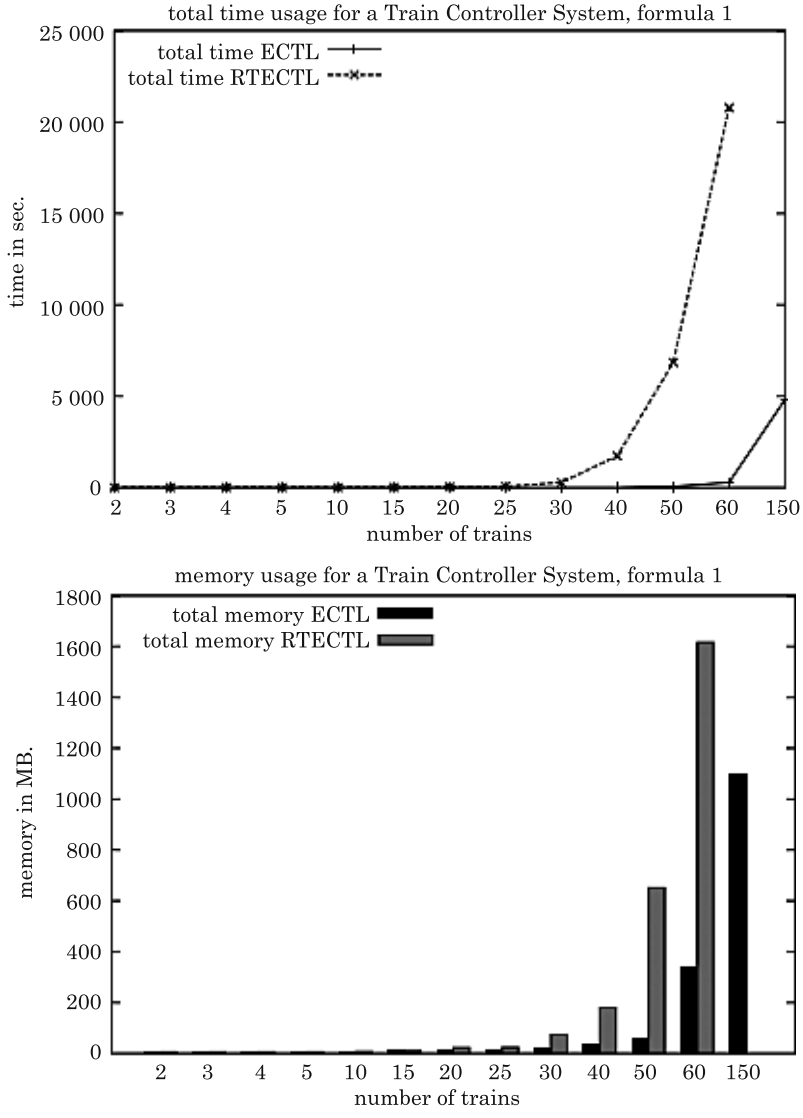

Fig. 2. A comparison of total time usage and total memory usage for the formula φ_1

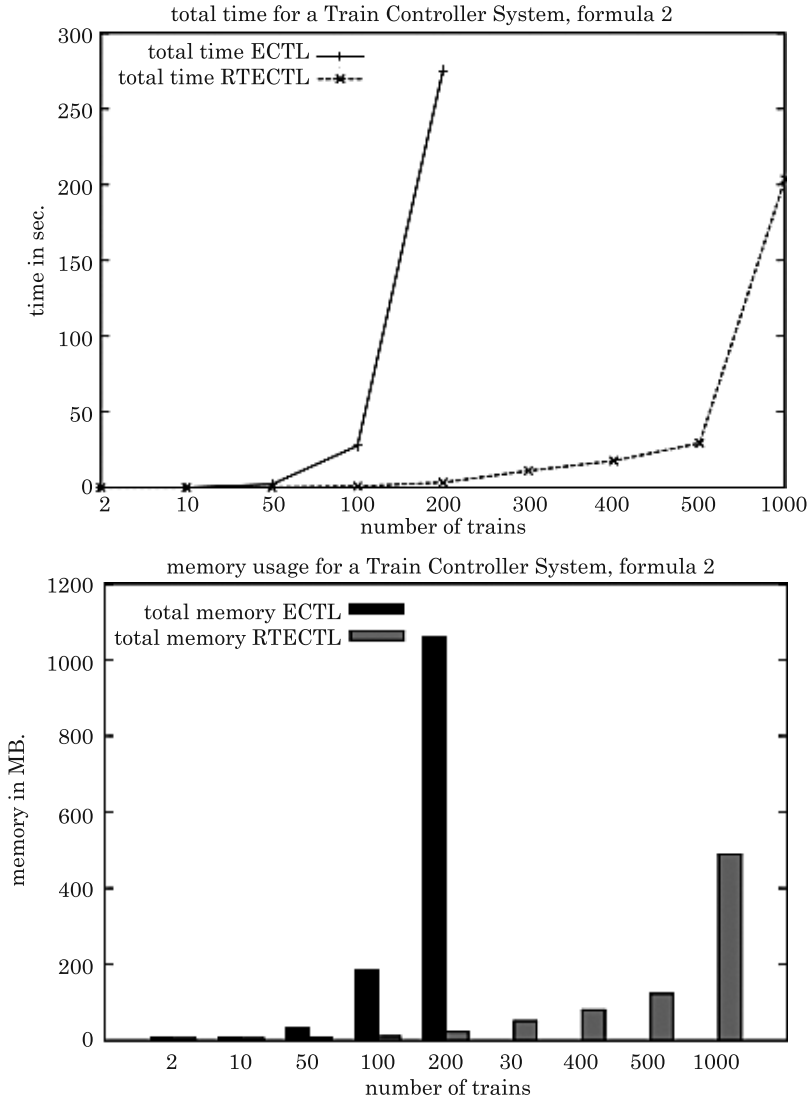

Fig. 3. A comparison of total time usage and total memory usage for the formula φ_2

Generic Pipeline Paradigm

The benchmark we consider is the generic pipeline paradigm (GPP) (PELED 1993), which consists of three parts: Producer producing data, Consumer receiving data, and a chain of n intermediate Nodes that transmit data

produced by Producer to Consumer. The local states for each component (Producer, Consumer, and intermediate Nodes), and their protocols are shown in the Figure 4.

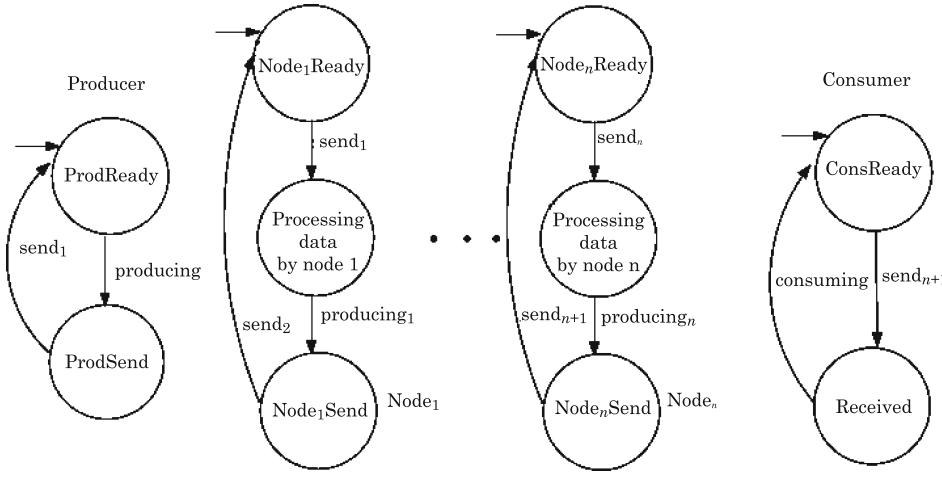

Fig. 4. The GPP system

The comparison of both BMC algorithms for RTECTL and ECTL with respect to the GPP system has been done by means of the following RTECTL specification:

– $\varphi_3 = \boldsymbol{EG}_{[0,\infty)}((\neg ProdSend \vee \boldsymbol{EF}_{[2n+1,2n+2)}(Received))$,
– $\varphi_4 = \boldsymbol{EG}_{[0,n^2+2n+1)}(Received)$, where n is the number of nodes.

The equivalent ECTL formulae for $n=2$ are:

– $tr(\varphi_3) = \boldsymbol{EG}(\neg ProdSend \vee \boldsymbol{EX}(\boldsymbol{EX}(\boldsymbol{EX}(\boldsymbol{EX}(\boldsymbol{EX}(Received))))))$,
– $tr(\varphi_4) = \boldsymbol{EX}(Received \wedge \boldsymbol{EX}(Received \wedge \boldsymbol{EX}(Received \wedge \boldsymbol{EX}(Received \wedge \boldsymbol{EX}(Received \wedge \boldsymbol{EX}(Received \wedge \boldsymbol{EX}(Received \wedge \boldsymbol{EX}(Received \wedge \boldsymbol{EX}(Received \wedge Received)))))))))$.

In the Figure 5 we present a comparison of total time usage and total memory usage for the formula φ_3. An observation of the experimental results for the formula φ_3 leads to the conclusion that the BMC for RTECTL uses less time and memory comparing to the BMC for ECTL. The reason is that although BMC needs much more paths for verification in case of ECTL.

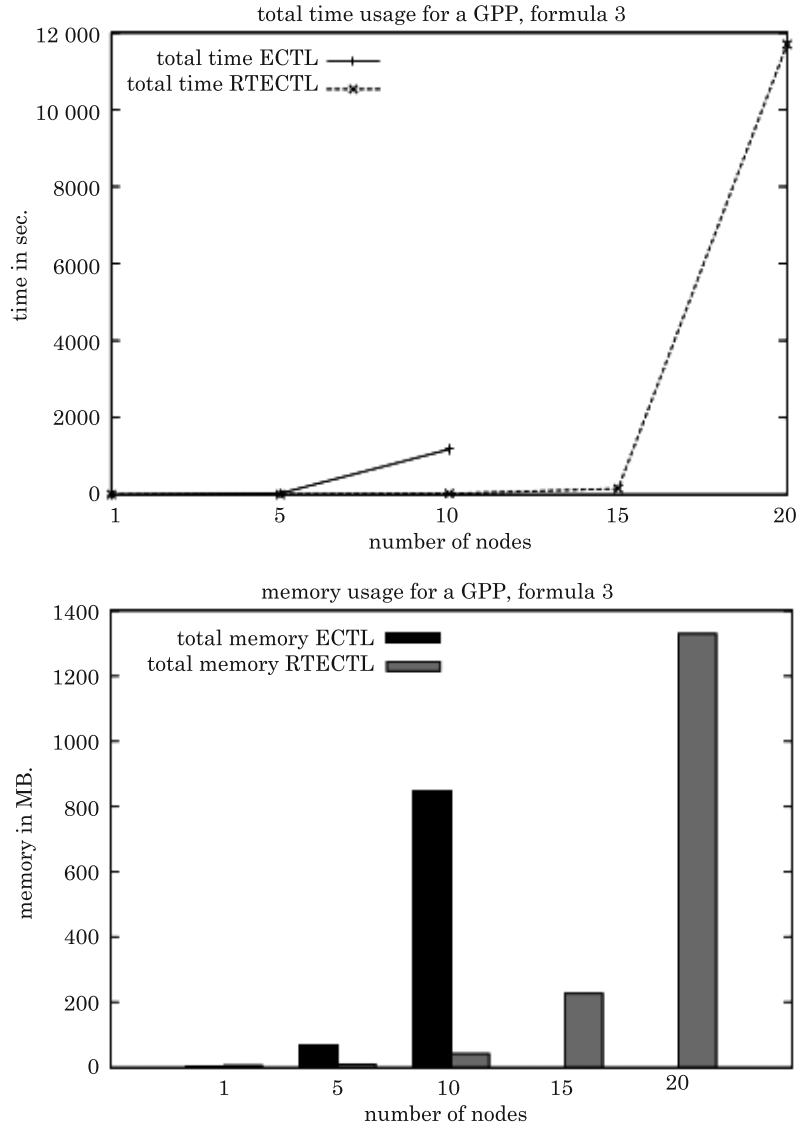

Fig. 5. A comparison of total time usage and total memory usage for the formula φ_3

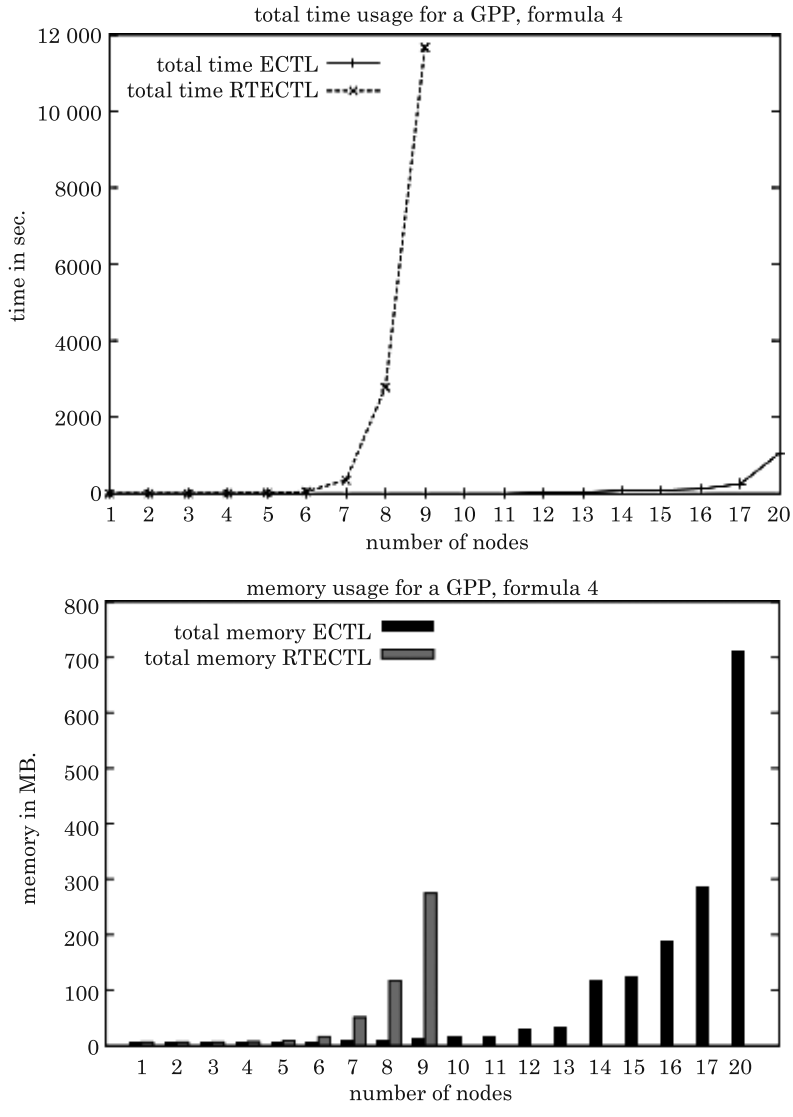

Fig. 6. A comparison of total time usage and total memory usage for the formula φ_4

In the Figure 6 we present a comparison of total time usage and total memory usage for the formula φ_4. An analysis of experimental results for the formula φ_4 leads to the conclusion that the BMC for ECTL uses less time and memory comparing to the BMC for RTECTL. The reason is that although BMC needs much more paths for verification in case of ECTL, but these paths are significantly shorter.

For the tests we have used a computer equipped with AMD Phenom™ 9550 Quad-Core 2200 MHz processor and 8 GB of RAM, running Ubuntu Linux with kernel version 3.5.0-17-generic. We have used the state of the art SAT-solver MiniSat 2 (EÉN, SÖRENSSON 2005, *MiniSat* 2006), which is one of the best SAT-solver.

Conclusions

In this paper we have presented a comparison between the BMC method for ECTL and the BMC method for RTECTL. Moreover, we have presented a correct translation for operator $\boldsymbol{EG_I}$, which was not defined in (CAMPOS 1996, EMERSON et al. 1992).

Further, we have tested and compared with each other on to standard benchmarks the BMC translations for RTECTL and ECTL incremented in (WOŹNA-SZCZEŚNIAK et al. 2011, ZBRZEZNY 2008). We have observed that for the RTECTL formulae for which the interval at the operator $\boldsymbol{EG}$ is finite, the BMC can be less effective than BMC for their translation into ECTL formulae.

Acknowledgements

The author wishes to thank Bożena Woźna-Szcześniak and Andrzej Zbrzezny for valuable comments and numerous suggestions which helped to improve this paper.

Partly supported by National Science Centre under the grant No. 2014/15/N/ST6/05079.

References

BAIER C., KATOEN J.-P. 2008. *Principles of model checking*. MIT Press.

BIERE A., CIMATTI A., CLARKE E., ZHU Y. 1999. *Symbolic Model Checking without BDDs*. Proceedings of the 5th Int. Conf. on Tools and Algorithms for the Construction and Analysis of Systems (TACAS;99), LNCS. Springer-Verlag, p. 193–207.

BRYANT R., 1986. *Graph-Based Algorithms for Boolean Function Manipulation*. IEEE Trans. Comput, 35(8): 677–691.

CAMPOS S.V.A. 1996. *A quantitative approach to the formal verification of real-time systems*. School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, USA.

CLARKE E., EMERSON E.A., SISTLA A.P. 1986. *Automatic Verification of Finite State Concurrent Systems Using Temporal Logic Specifications: A Practical Approach*. ACM Trans. Program. Lang. Syst., 8(2): 244–263.

CLARKE E., GRUMBERG O., PELED D. 1999. *Model Checking*. MIT Press.

EMERSON E.A., MOK A.K., SISTLA A.P., SRINIVASAN J. 1992. *Quantitative Temporal Reasoning*. Real-Time Syst., 4: 331–352.

LOMUSCIO A., PENCZEK W., QU H. 2010. *Partial order reduction for model checking interleaved multi-agent systems*. Proceedings of the 9th International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS;2010). IFAAMAS Press, p. 659–666.

MCMILLAN K.L. 1993. *Symbolic Model Checking*. Kluwer Academic Publishers.

PENCZEK W., LOMUSCIO A. 2003. *Verifying Epistemic Properties of Multi-Agent Systems via Bounded Model Checking*. Fundam. Informaticae, 55: 167–185.

PENCZEK W., WOŹNA B., ZBRZEZNY A. 2002. *Bounded Model Checking for the Universal Fragment of CTL*. Fundam. Informaticae, 51(1–2): 135–156.

WOŹNA B., ZBRZEZNY A. 2004. *Checking ACTL* Properties of Discrete Timed Automata via Bounded Model Checking*. Proceedings of the 1st Int. Workshop on Formal Analysis and Modeling of Timed Systems (FORMATS'03), LNCS. Springer-Verlag, p. 18–33.

WOŹNA-SZCZEŚNIAK B., ZBRZEZNY A.M., ZBRZEZNY A. 2011. *The BMC method for the existential part of RTCTLK and interleaved interpreted systems*. Proceedings of the 15th Portuguese Conference on Artificial Intelligence (EPIA;2011), LNAI. Springer-Verlag, p. 551–565.

ZBRZEZNY A. 2008. *Improving the Translation from ECTL to SAT*. Fundam. Informaticae 85(1–4): 513–531.