SAT-Based Bounded Model Checking for Deontic Interleaved Interpreted Systems*

Bożena Woźna-Szcześniak and Andrzej Zbrzezny

IMCS, Jan Długosz University, Al. Armii Krajowej 13/15, 42-200 Częstochowa, Poland
{b.wozna,a.zbrzezny}@ajd.czest.pl

Abstract. We propose a bounded model checking (BMC) method for the verification of multi-agent systems' (MASs). The MASs are modelled by deontic interleaved interpreted systems, and specifications are expressed in the logic RTECTLKD. The verification approach is based on the state of the art solutions to BMC, one of the mainstream approaches in verification of reactive systems. We test our results on a typical communication scenario: train controller problem with faults.

1 Introduction

Agents are rational and sophisticated entities that act autonomously on behalf of their users, across open and distributed environments, to solve a growing number of complex problems. *A multi-agent system* (MAS) [14] is a loosely united network of agents that interact (communicate, coordinate, cooperate, etc.) to solve problems that are beyond the individual capacities or knowledge of a single agent. *Deontic interpreted systems* (DIS), a deontic extension of interpreted systems [4], were defined in [8] to represent and reason about epistemic and correct functioning behaviour of MASs. They provide a semantics based on the computation states of the agents, on which it is possible to interpret a modality $\mathcal{O}_i\phi$, representing the fact "in all correct functioning executions of agent i, ϕ holds", as well as a traditional epistemic modalities and temporal operators. *Deontic interleaved interpreted systems* (DIIS) are a deontic extension of the formalism of interleaved interpreted systems [7]. We introduce them since they allow for the distinction between correct (or ideal, normative, etc.) and incorrect states, and they enable more efficient verification of MASs that are not so loosely coupled.

Model checking [2] is one of the mainstream techniques whose aim is to provide an algorithm determining whether an abstract model - representing, for example, a software project - satisfies a formal specification expressed as a modal formula. Moreover, if the property does not hold, the method identifies a counterexample execution that shows the source of the problem. The practical applicability of model checking in MASs settings requires the development of sophisticated means of coping with what is known as the state explosion problem. To avoid this problem a number of approaches have been developed, including BDD-based bounded [5,9] and unbounded [13,12] model checking, SAT-based bounded [10,11,15] and unbounded [6] model checking.

* Partly supported by National Science Center under the grant No. 2011/01/B/ST6/05317.

G. Jezic et al. (Eds.): KES-AMSTA 2012, LNAI 7327, pp. 494–503, 2012.

The RTCTLKD language is an epistemic and deontic extension of RTCTL [3], which allows for the representation of the quantitative temporal evolution of epistemic states of the agents, as well as their correct and incorrect functioning behaviour.

In past research we have provided a theoretical underpinnings of a bounded model checking (BMC) algorithm for DIS and an existential part of CTLKD (ECTLKD) [15]. However, the method have not been implemented and experimentally evaluated. Moreover, it was not tailored to the DIISs settings, and it was not based on the state-of-the art BMC method for ECTL [17], which uses a reduced number of paths, what results in significantly smaller and less complicated propositional formulae that encode the ECTLKD properties. In this paper we provide a new SAT-based BMC technique for the existential part of RTCTLKD (thus, for ECTLKD as well) by means of which we can automatically verify not only epistemic and temporal properties but also deontic and quantitative temporal properties that express compliance of a MAS, modelled by DIIS, with respect to specifications.

The structure of the paper is as follows. In Section 2 we shortly introduce DIISs, the RTCTLKD language together with its existential (RTECTLKD) and universal fragments (RTACTLKD), unbounded and bounded semantics. In Section 3 we define a BMC method for RTECTLKD. In Section 4 we present performance evaluation of our newly developed SAT-based BMC algorithm and we conclude the paper.

2 Preliminaries

DIIS. We assume that a MAS consists of n agents, and by $Ag = \{1, \ldots, n\}$ we denote the non-empty set of agents; note that we do not consider the environment component. This may be added with no technical difficulty at the price of heavier notation. We assume that each agent $c \in Ag$ is in some particular local state at a given point in time, and that a set L_c of local states for agent $c \in Ag$ is non-empty and finite (this is required by the model checking algorithms). We assume that for each agent $c \in Ag$, its set L_c can be partitioned into *faultless (green)* and *faulty (red)* states. For n agents and n mutually disjoint and non-empty sets $\mathcal{G}_1, \ldots, \mathcal{G}_n$ we define the set G of all possible *global states* as the Cartesian product $L_1 \times \ldots \times L_n$, such that $L_1 \supseteq \mathcal{G}_1, \ldots, L_n \supseteq \mathcal{G}_n$. The set $\mathcal{G}_c$ is called the set of green states for agent c. The complement of $\mathcal{G}_c$ with respect to L_c (denoted by $\mathcal{R}_c$) is called the set of red states for the agent c. Note that $\mathcal{G}_c \cup \mathcal{R}_c = L_c$ for any agent c. Further, by $l_c(g)$ we denote the local component of agent $c \in Ag$ in a global state $g = (l_1, \ldots, l_n)$.

With each agent $c \in Ag$ we associate a finite set of *possible actions* Act_c such that a special "null" action (ϵ_c) belongs to Act_c; as it will be clear below the local state of agent c remains the same, if the null action is performed. We do not assume that the sets Act_c (for all $c \in Ag$) are disjoint. Next, with each agent $c \in Ag$ we associate a protocol that defines rules, according to which actions may be performed in each local state. The protocol for agent $c \in Ag$ is a function $P_c : L_c \to 2^{Act_c}$ such that $\epsilon_c \in P_c(l)$ for any $l \in L_c$, i.e., we insist on the null action to be enabled at every local state. For each agent c, there is defined a (partial) evolution function $t_c : L_c \times Act_c \to L_c$ such that for each $l \in L_c$ and for each $a \in P_c(l)$ there exists $l' \in L_c$ such that $t_c(l, a) = l'$; moreover, $t_c(l, \epsilon_c) = l$ for each $l \in L_c$. Note that the local evolution function considered here

differs from the standard one (see [4]) by having the local action instead of the join action as the parameter. Further, we define the following sets $Act = \bigcup_{c\in Ag} Act_c$ and $Agent(a) = \{c \in Ag \mid a \in Act_c\}$.

We assumed that, in every state, agents evolve simultaneously. Thus the *global interleaved evolution function* $t : G \times Act_1 \times \cdots \times Act_n \to G$ is defined as follows: $t(g, a_1, \ldots, a_n) = g'$ iff there exists an action $a \in Act \setminus \{\epsilon_1, \ldots, \epsilon_n\}$ such that for all $c \in Agent(a)$, $a_c = a$ and $t_c(l_c(g), a) = l_c(g')$, and for all $c \in Ag \setminus Agent(a)$, $a_c = \epsilon_c$ and $t_c(l_c(g), a_c) = l_c(g)$. In brief we write the above as $g \xrightarrow{a} g'$.

Now, for a given set of agents Ag and a set of propositional variables $\mathcal{PV}$, which can be either true or false, a *deontic interleaved interpreted system* is a tuple: $DIIS = (\iota, < L_c, \mathcal{G}_c, Act_c, P_c, t_c >_{c\in Ag}, \mathcal{V})$, where $\iota \in G$ is an initial global state, and $\mathcal{V} : G \to 2^{\mathcal{PV}}$ is a valuation function. With such a DIIS it is possible to associate a *model* $M = (\iota, S, T, \{\sim_c\}_{c\in Ag}, \{\bowtie_c\}_{c\in Ag}, \mathcal{V})$, where ι is the initial global state; $S \subseteq G$ is a set of reachable global states that is generated from ι by using the global interleaved evolution functions t; $T \subseteq S \times S$ is a global transition (temporal) relation on S defined by: sTs' iff there exists an action $a \in Act \setminus \{\epsilon_1, \ldots, \epsilon_n\}$ such that $s \xrightarrow{a} s'$. We assume that the relation is total, i.e., for any $s \in S$ there exists an $a \in Act \setminus \{\epsilon_1, \ldots, \epsilon_n\}$ such that $s \xrightarrow{a} s'$ for some $s' \in S$; $\sim_c \subseteq S \times S$ is an indistinguishability relation for agent c defined by: $s \sim_c s'$ iff $l_c(s') = l_c(s)$; $\bowtie_c \subseteq S \times S$ is a deontic relation for agent c defined by: $s \bowtie_c s'$ iff $l_c(s') \in \mathcal{G}_c$; $\mathcal{V} : S \to 2^{\mathcal{PV}}$ is the valuation function of $DIIS$ restricted to the set S. $\mathcal{V}$ assigns to each state a set of propositional variables that are assumed to be true at that state.

Syntax of RTCTLKD. Let $p \in \mathcal{PV}$, $c \in Ag$, $\Gamma \subseteq Ag$, and I be an interval in $\mathbb{N} = \{0, 1, 2, \ldots\}$ of the form: $[a, b)$ and $[a, \infty)$, for $a, b \in \mathbb{N}$; note that the remaining forms of intervals can be defined by means of $[a, b)$ and $[a, \infty)$. Hereafter, let $left(I)$ denote the left end of the interval I, and $right(I)$ the right end of the interval I. The language RTCTLKD is defined by the following grammar:

$$\varphi := \mathbf{true} \mid \mathbf{false} \mid p \mid \neg\alpha \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \mathrm{EX}\varphi \mid \mathrm{E}(\varphi \mathrm{U}_I \varphi) \mid \mathrm{EG}_I\varphi \mid$$
$$\overline{\mathrm{K}}_c\varphi \mid \overline{\mathrm{D}}_\Gamma\varphi \mid \overline{\mathrm{E}}_\Gamma\varphi \mid \overline{\mathrm{C}}_\Gamma\varphi \mid \overline{\mathcal{O}}_c\alpha \mid \widehat{\underline{\mathrm{K}}}_c^d\alpha$$

The derived basic modalities are defined as follows: $\mathrm{E}(\alpha \mathrm{R}_I \beta) \stackrel{def}{=} \mathrm{E}(\beta \mathrm{U}_I(\alpha \wedge \beta)) \vee \mathrm{EG}_I\beta$, $\mathrm{EF}_I\alpha \stackrel{def}{=} \mathrm{E}(\mathbf{true}\mathrm{U}_I\alpha)$, $\mathrm{AX}\alpha \stackrel{def}{=} \neg\mathrm{EX}\neg\alpha$, $\mathrm{AF}\alpha \stackrel{def}{=} \neg\mathrm{EG}\neg\alpha$, $\mathrm{A}(\alpha \mathrm{R}\beta) \stackrel{def}{=} \neg\mathrm{E}(\neg\alpha \mathrm{U}\neg\beta)$, $\mathrm{AG}\alpha \stackrel{def}{=} \neg\mathrm{EF}\neg\alpha$, $\mathcal{O}_c\alpha \stackrel{def}{=} \neg\overline{\mathcal{O}}_c\neg\alpha$, $\mathrm{K}_c\alpha \stackrel{def}{=} \neg\overline{\mathrm{K}}_c\neg\alpha$, $\widehat{\mathrm{K}}_c^d\alpha \stackrel{def}{=} \neg\widehat{\underline{\mathrm{K}}}_c^d\neg\alpha$, $\mathrm{D}_\Gamma\varphi \stackrel{def}{=} \neg\overline{\mathrm{D}}_\Gamma\neg\alpha$, $\mathrm{E}_\Gamma\varphi \stackrel{def}{=} \neg\overline{\mathrm{E}}_\Gamma\neg\alpha$, $\mathrm{C}_\Gamma\varphi \stackrel{def}{=} \neg\overline{\mathrm{C}}_\Gamma\neg\alpha$, where $c, d \in \mathcal{AG}$, and $\Gamma \subseteq \mathcal{AG}$. Intuitively, E and A mean, resp., there exists a computation, and for all the computations, U_I and G_I are the operators, resp., for "bounded until" and "bounded always". $\overline{\mathrm{K}}_c$ is the operator dual for the standard epistemic modality K_c ("agent c knows"), so $\overline{\mathrm{K}}_c\alpha$ is read as "agent c does not know whether or not α holds". Similarly, the modalities $\overline{\mathrm{D}}_\Gamma, \overline{\mathrm{E}}_\Gamma, \overline{\mathrm{C}}_\Gamma$ are the dual operators for $\mathrm{D}_\Gamma, \mathrm{E}_\Gamma, \mathrm{C}_\Gamma$ representing distributed knowledge in the group Γ, everyone in Γ knows, and common knowledge among agents in Γ. Further, we use the (double) indexed modal operators $\mathcal{O}_c$, $\overline{\mathcal{O}}_c$, $\widehat{\mathrm{K}}_c^d$ and $\widehat{\underline{\mathrm{K}}}_c^d$ to represent the *correctly functioning circumstances of agent* c. The formula $\mathcal{O}_c\alpha$ stands for "for all the states where agent c is functioning correctly, α holds". $\overline{\mathcal{O}}_c$ is the operator dual for the modality $\mathcal{O}_c$. The formula $\widehat{\mathrm{K}}_c^d\alpha$ is read as "agent c knows

that α under the assumption that agent d is functioning correctly". $\underline{\widehat{\mathrm{K}}}_c^d$ is the operator dual for the modality $\underline{\mathrm{K}}_c^d$. We refer to [8] for a discussion of this notion; note that the operator $\overline{\mathcal{O}}_c$ is there referred to as $\mathcal{P}_c$.

Next, we define two sublogics of RTCTLKD. The first one is the existential fragment of RTCTLKD (RTECTLKD), defined by the following grammar: $\varphi ::= p \mid \neg p \mid \alpha \wedge \beta \mid \alpha \vee \beta \mid \mathrm{EX}\alpha \mid \mathrm{E}(\alpha \mathrm{U}_I \beta) \mid \mathrm{EG}_I\alpha \mid \overline{\mathrm{K}}_c\alpha \mid \overline{\mathrm{E}}_\Gamma\alpha \mid \overline{\mathrm{D}}_\Gamma\alpha \mid \overline{\mathrm{C}}_\Gamma\alpha \mid \overline{\mathcal{O}}_c\alpha \mid \underline{\widehat{\mathrm{K}}}_c^d\alpha$. The second on is the universal fragment of RTCTLKD (RTACTLKD), defined as: $\varphi ::= p \mid \neg p \mid \alpha\wedge\beta \mid \alpha\vee\beta \mid \mathrm{AX}\alpha \mid \mathrm{A}(\alpha \mathrm{R}_I \beta) \mid \mathrm{AF}_I\alpha \mid \mathrm{K}_c\alpha \mid \mathrm{E}_\Gamma\alpha \mid \mathrm{D}_\Gamma\alpha \mid \mathrm{C}_\Gamma\alpha \mid \mathcal{O}_c\alpha \mid \widehat{\mathrm{K}}_c^d\alpha$.

Semantics of RTCTLKD. Let M be a model for $DIIS$. A *path* in M is an infinite sequence $\pi = (s_0, s_1, \ldots)$ of states such that $(s_j, s_{j+1}) \in T$ for each $j \in \mathbb{N}$. For a path π, we take $\pi(j) = s_j$. By $\Pi(s)$ we denote the set of all the paths starting at $s \in S$. For the group epistemic modalities we define the following. If $\Gamma \subseteq Ag$, then $\sim_\Gamma^E \stackrel{def}{=} \bigcup_{c\in\Gamma} \sim_c$, $\sim_\Gamma^C \stackrel{def}{=} (\sim_\Gamma^E)^+$ (the transitive closure of $\sim_\Gamma^E$), and $\sim_\Gamma^D \stackrel{def}{=} \bigcap_{c\in\Gamma} \sim_c$. Given the above, the semantics of RTCTLKD is the following:

- $M, s \models \mathbf{true}$, • $M, s \not\models \mathbf{false}$, • $M, s \models p$ iff $p \in \mathcal{V}(s)$, • $M, s \models \neg\alpha$ iff $M, s \not\models \alpha$,
- $M, s \models \alpha \wedge \beta$ iff $M, s \models \alpha$ and $M, s \models \beta$,
- $M, s \models \alpha \vee \beta$ iff $M, s \models \alpha$ or $M, s \models \beta$,
- $M, s \models \mathrm{EX}\alpha$ iff $(\exists \pi \in \Pi(s))(M, \pi(1) \models \alpha)$,
- $M, s \models \mathrm{E}(\alpha \mathrm{U}_I \beta)$ iff $(\exists \pi \in \Pi(s))(\exists m \in I)[M, \pi(m) \models \beta$ and $(\forall j < m) M, \pi(j) \models \alpha]$,
- $M, s \models \mathrm{EG}_I\alpha$ iff $(\exists \pi \in \Pi(s))$ such that $(\forall m \in I)[M, \pi(m) \models \alpha]$,
- $M, s \models \overline{\mathrm{K}}_c\alpha$ iff $(\exists s' \in S)(s \sim_c s'$ and $M, s' \models \alpha)$,
- $M, s \models \overline{Y}_\Gamma\alpha$ iff $(\exists s' \in S)(s \sim_\Gamma^Y s'$ and $M, s' \models \alpha)$, where $Y \in \{\mathrm{D}, \mathrm{E}, \mathrm{C}\}$,
- $M, s \models \overline{\mathcal{O}}_c\alpha$ iff $(\exists s' \in S)(s \bowtie_c s'$ and $M, s' \models \alpha)$,
- $M, s \models \underline{\widehat{\mathrm{K}}}_c^d\alpha$ iff $(\exists s' \in S)(s \sim_c s'$ and $s \bowtie_d s'$ and $M, s' \models \alpha)$.

An RTCTLKD formula φ is *valid* in M (denoted $M \models \varphi$) iff $M, \iota \models \varphi$, i.e., φ is true at the initial state of the model M. The *model checking problem* asks whether $M \models \varphi$.

Bounded Semantics. The proposed bounded semantics is the backbone of the SAT-based BMC method for RTECTLKD, which is presented in the next section. As usual, we start by defining *k-paths* and *loops*.

Let M be a model for $DIIS$, $k \in \mathbb{N}$, and $0 \leq l \leq k$. A *k-path* is a finite sequence $\pi = (s_0, \ldots, s_k)$ of states such that $(s_j, s_{j+1}) \in T$ for each $0 \leq j < k$. A k-path π is a *loop* if $\pi(k) = \pi(l)$ for some $l < k$. By $\Pi_k(s)$ we denote the set of all the k-paths starting at s in M. Note that although every k-path π is finite, if it is a loop, then it generates the infinite path of the following form: $u \cdot v^\omega$ with $u = (\pi(0), \ldots, \pi(l))$ and $v = (\pi(l+1), \ldots, \pi(k))$. Further, since in the bounded semantics we consider finite prefixes of paths only, the satisfiability of all the temporal operators depends on whether a considered k-path is a loop. Thus, as customary, we introduce a function $loop : \bigcup_{s\in S} \Pi_k(s) \to 2^{\mathbb{N}}$, which identifies these k-paths that are loops. The function is defined as: $loop(\pi_k) = \{l \mid 0 \leq l < k \text{ and } \pi_k(l) = \pi_k(k)\}$.

Given the above, the bounded semantics of RTECTLKD is defined as follows. Let $M, s \models_k \alpha$ denotes that α is $k-$true at the state s of M. The relation $\models_k$ is defined inductively as follows:

- $M,s \models_k \mathbf{true}$, • $M,s \not\models_k \mathbf{false}$, • $M,s \models_k p$ *iff* $p \in \mathcal{V}(s)$,
- $M,s \models_k \neg p$ *iff* $p \notin \mathcal{V}(s)$, • $M,s \models_k \alpha \vee \beta$ *iff* $M,s \models_k \alpha$ or $M,s \models_k \beta$,
- $M,s \models_k \alpha \wedge \beta$ *iff* $M,s \models_k \alpha$ and $M,s \models_k \beta$,
- $M,s \models_k \mathrm{EX}\alpha$ *iff* $k > 0$ *and* $(\exists \pi \in \Pi_k(s))M,\pi(1) \models_k \alpha$,
- $M,s \models_k \mathrm{E}(\alpha \mathrm{U}_I \beta)$ *iff* $(\exists \pi \in \Pi_k(s))(\exists 0 \leq m \leq k)(m \in I$ *and* $M,\pi(m) \models_k \beta$ *and* $(\forall 0 \leq j < m)M,\pi(j) \models_k \alpha)$,
- $M,s \models_k \mathrm{EG}_I\alpha$ *iff* $(\exists \pi \in \Pi_k(s))((k \geq right(I)$ *and* $(\forall j \in I)\ M,\pi(j) \models_k \alpha)$ *or* $(k < right(I)$ *and* $(\exists l \in loop(\pi))(\forall min(left(I),l) \leq j < k)M,\pi(j) \models_k \alpha))$,
- $M,s \models_k \overline{\mathrm{K}}_c\alpha$ *iff* $(\exists \pi \in \Pi_k(\iota))(\exists 0 \leq j \leq k)(M,\pi(j) \models_k \alpha$ *and* $s \sim_c \pi(j))$,
- $M,s \models_k \overline{\mathrm{Y}}_\Gamma\alpha$ *iff* $(\exists \pi \in \Pi_k(\iota))(\exists 0 \leq j \leq k)(M,\pi(j) \models_k \alpha$ *and* $s \sim_\Gamma^Y \pi(j))$, where $Y \in \{\mathrm{D},\mathrm{E},\mathrm{C}\}$,
- $M,s \models_k \overline{\mathcal{O}}_c\alpha$ *iff* $(\exists \pi \in \Pi_k(\iota))(\exists 0 \leq j \leq k)(M,\pi(j) \models_k \alpha$ *and* $s \bowtie_c \pi(j))$,
- $M,s \models_k \underline{\widehat{\mathrm{K}}}_c^d\alpha$ *iff* $(\exists \pi \in \Pi_k(\iota))(\exists 0 \leq j \leq k)(M,\pi(j) \models_k \alpha$ *and* $s \sim_c \pi(j)$ *and* $s \bowtie_d \pi(j))$.

An RTECTLKD formula φ is *valid in model M with bound k* (denoted $M \models_k \varphi$) iff $M,\iota \models_k \varphi$, i.e., φ is k-true at the initial state of the model M. The *bounded model checking problem* asks whether there exists $k \in \mathbb{N}$ such that $M \models_k \varphi$.

The following theorem states that for a given model and formula there exists a bound k such that the model checking problem ($M \models \varphi$) can be reduced to the bounded model checking problem ($M \models_k \varphi$). Its proof can be done by straightforward induction on the length of an RTECTLKD formula.

Theorem 1. *Let M be a model and φ an* RTECTLKD *formula. Then, the following equivalence holds: $M \models \varphi$ iff there exists $k \geq 0$ such that $M \models_k \varphi$.*

Further, by straightforward induction on the length of an RTECTLKD formula φ, we can show that φ is k-true in M if and only if φ is k-true in M with a number of k-paths reduced to $f_k(\varphi)$, where the function f_k : RTECTLKD $\to \mathbb{N}$ is defined as follows. $f_k(\mathbf{true}) = f_k(\mathbf{false}) = f_k(p) = f_k(\neg p) = 0$, where $p \in \mathcal{PV}$; $f_k(\alpha \wedge \beta) = f_k(\alpha) + f_k(\beta)$; $f_k(\alpha \vee \beta) = max\{f_k(\alpha), f_k(\beta)\}$; $f_k(\mathrm{E}(\alpha \mathrm{U}_I \beta)) = k \cdot f_k(\alpha) + f_k(\beta) + 1$; $f_k(\mathrm{EG}_I\alpha) = (k+1) \cdot f_k(\alpha) + 1$; $f_k(\overline{\mathrm{C}}_\Gamma\alpha) = f_k(\alpha) + k$; $f_k(Y\alpha) = f_k(\alpha) + 1$ for $Y \in \{\mathrm{EX}, \overline{\mathrm{K}}_c, \overline{\mathcal{O}}_c, \underline{\widehat{\mathrm{K}}}_c^d, \overline{\mathrm{D}}_\Gamma, \overline{\mathrm{E}}_\Gamma\}$.

3 SAT-Based BMC for RTECTLKD

Let $M = (\iota, S, T, \{\sim_c\}_{c \in Ag}, \{\bowtie_c\}_{c \in Ag}, \mathcal{V})$ be a model, φ an RTECTLKD formula, and $k \geq 0$ a bound. The proposed BMC method is based on the BMC encoding presented in [16], and it consists in translating the problem of checking whether $M \models_k \varphi$ holds, to the problem of checking the satisfiability of the propositional formula $[M,\varphi]_k := [M^{\varphi,\iota}]_k \ \wedge \ [\varphi]_{M,k}$. The formula $[M^{\varphi,\iota}]_k$ encodes sets of k-paths of M, whose size are equal to $f_k(\varphi)$, and in which at least one path starts at the initial state of the model M. The formula $[\varphi]_{M,k}$ encodes a number of constraints that must be satisfied on these sets of k-paths for φ to be satisfied. Once this translation is defined, checking satisfiability of an RTECTLKD formula can be done by means of a SAT-solver.

In order to define the formula $[M,\varphi]_k$ we proceed as follows. We begin with an encoding of states of the given model M. Since the set of states of M is finite, each state

s of M can be encoded by a bit-vector, whose length r depends on the number of agents' local states. Thus, each state s of M can be represented by a vector $w = (w_1, \ldots, w_r)$ (called a *symbolic state*) of propositional variables (called *state variables*). A finite sequence $(w_0, \ldots, w_k)$ of symbolic states of length k is called a *symbolic k-path*. Since in general we may need to consider more than one symbolic k-path, we introduce a notion of the j-th symbolic k-path $(w_{0,j}, \ldots, w_{k,j})$, where $w_{i,j}$ are symbolic states for $0 \leq j < f_k(\varphi)$ and $0 \leq i \leq k$. Note that the exact number of symbolic k-paths depends on the checked formula φ, and it can be calculated by means of the function f_k.

Let $\sigma : SV \to \{0,1\}$ be a *valuation of state variables* (a *valuation* for short). Each valuation induces the function $\boldsymbol{\sigma} : SV^r \to \{0,1\}^r$ defined in the following way: $\boldsymbol{\sigma}((w_1, \ldots, w_r)) = (\sigma(w_1), \ldots, \sigma(w_r))$. Moreover, let SV denote the set of all the state variables, and $SV(w)$ denote the set of all the state variables occurring in a symbolic state w. Next, let w and w' be two symbolic states such that $SV(w) \cap SV(w') = \emptyset$. We define the following auxiliary propositional formulae:

- $I_s(w)$ is a formula over $SV(w)$ that is true for a valuation σ iff $\boldsymbol{\sigma}(w) = s$.
- $p(w)$ is a formula over w that is true for a valuation σ iff $p \in \mathcal{V}(\boldsymbol{\sigma}(w))$ (encodes a set of states of M in which $p \in \mathcal{PV}$ holds).
- $H(w, w')$ is a formula over $SV(w) \cup SV(w')$ that is true for a valuation σ iff $\boldsymbol{\sigma}(w) = \boldsymbol{\sigma}(w')$ (encodes equivalence of two global states).
- $H_c(w, w')$ is a formula over $SV(w) \cup SV(w')$ that is true for a valuation σ iff $l_c(\boldsymbol{\sigma}(w)) = l_c(\boldsymbol{\sigma}(w))$ (encodes equivalence of local states of agent c).
- $HO_c(w, w')$ is a formula over $SV(w) \cup SV(w')$ that is true for a valuation σ iff $l_c(\boldsymbol{\sigma}(w')) \in \mathcal{G}_c$ (encodes an accessibility of a global state in which agent c is functioning correctly).
- $\widehat{H}_c^d(w, w') := H_c(w, w') \wedge HO_d(w, w')$.
- $\mathcal{R}(w, w')$ is a formula over $SV(w) \cup SV(w')$ that is true for a valuation σ iff $(\boldsymbol{\sigma}(w), \boldsymbol{\sigma}(w')) \in T$ (encodes the transition relation of M).
- Let $j \in \mathbb{N}$, and I be an interval. Then $In(j, I) := \mathbf{true}$ if $j \in I$, and $In(j, I) := \mathbf{false}$ if $j \notin I$.

Let $W = \{SV(w_{i,j}) \mid 0 \leq i \leq k \text{ and } 0 \leq j < f_k(\varphi)\}$ be a set of state variables. The propositional formula $[M^{\varphi,\iota}]_k$ is defined over the set W in the following way:

$$[M^{\varphi,\iota}]_k := I_\iota(w_{0,0}) \wedge \bigwedge_{j=0}^{f_k(\varphi)-1} \bigwedge_{i=0}^{k-1} \mathcal{R}(w_{i,j}, w_{i+1,j})$$

The next step of the reduction to SAT is the transformation of an RTECTLKD formula φ into a propositional formula $[\varphi]_{M,k} := [\varphi]_k^{[0,0,F_k(\varphi)]}$, where $F_k(\varphi) = \{j \in \mathbb{N} \mid 0 \leq j < f_k(\varphi)\}$, and $[\varphi]_k^{[m,n,A]}$ denotes the translation of φ at the symbolic state $w_{m,n}$ using k-paths, whose indices are in the set A.

Following [17], to translate an RTECTLKD formula with an operator Q (where $\mathrm{Q} \in \{\mathrm{EX}, \mathrm{EU}_I, \mathrm{EG}_I, \overline{\mathrm{K}}_1, \ldots, \overline{\mathrm{K}}_n, \overline{\mathcal{O}}_1, \ldots, \overline{\mathcal{O}}_n, \overline{\mathrm{D}}_\Gamma, \overline{\mathrm{E}}_\Gamma\} \cup \{\underline{\widehat{\mathrm{K}}}_c^d \mid c, d \in Ag \text{ and } c \neq d\}$), we want exactly one path to be chosen for translating the operator Q, and the remaining k-paths to be used to translate arguments of Q. To accomplish this goal we need some auxiliary functions. However, before we define them, we first recall a definition of a relation $\prec$ that is defined on the power set of $\mathbb{N}$ as follows: $A \prec B$ iff for all natural numbers x and y, if $x \in A$ and $y \in B$, then $x < y$. Notice that from the definition of $\prec$ it follows that $A \prec B$ iff either $A = \emptyset$ or $B = \emptyset$ or $A \neq \emptyset$, $B \neq \emptyset$, $A \cap B = \emptyset$ and $max(A) < min(B)$.

Now, let $A \subset \mathbb{N}$ be a finite nonempty set, $k, p \in \mathbb{N}$, and $m \in \mathbb{N}$ such that $m \leq |A|$:

- $g_l(A, m)$ denotes the subset B of A such that $|B| = m$ and $B \prec A \setminus B$.
- $g_r(A, m)$ denotes the subset C of A such that $|C| = m$ and $A \setminus C \prec C$.
- $g_s(A)$ denotes the set $A \setminus \{min(A)\}$.
- If $k+1$ divides $|A|-1$, then $h_{\mathrm{G}}(A, k)$ denotes the sequence $(B_0, \ldots, B_k)$ of subsets of $A \setminus \{min(A)\}$ such that $\bigcup_{j=0}^{k} B_j = A \setminus \{min(A)\}$, $|B_0| = \ldots = |B_k|$, and $B_i \prec B_j$ for every $0 \leq i < j \leq k$. If $h_{\mathrm{G}}(A, k) = (B_0, \ldots, B_k)$, then $h_{\mathrm{G}}(A, k)(j)$ denotes the set B_j, for every $0 \leq j \leq k$.
  Notice that if $k + 1$ does not divide $|A| - 1$, then $h_{\mathrm{G}}(A, k)$ is undefined. However, for every set A such that $|A| = f_k(\mathrm{EG}_I\alpha)$, it follows from the definition of f_k that $k + 1$ divides $|A| - 1$.
- If k divides $|A| - 1 - p$, then $h_{\mathrm{U}}(A, k, p)$ denotes the sequence $(B_0, \ldots, B_k)$ of subsets of $A \setminus \{min(A)\}$ such that $\bigcup_{j=0}^{k} B_j = A \setminus \{min(A)\}$, $|B_0| = \ldots = |B_{k-1}|$, $|B_k| = p$, and $B_i \prec B_j$ for every $0 \leq i < j \leq k$. If $h_{\mathrm{U}}(A, k, p) = (B_0, \ldots, B_k)$, then $h_{\mathrm{U}}(A, k, p)(j)$ denotes the set B_j, for every $0 \leq j \leq k$.
  Notice that if k does not divide $|A|-1-p$, then $h_{\mathrm{U}}(A, k, p)$ is undefined. However, for every set A such that $|A| = f_k(\mathrm{E}(\alpha \mathrm{U}_I \beta))$, it follows from the definition of f_k that k divides $|A| - 1 - f_k(\beta)$.

Let φ be an RTECTLKD formula, and $k \geq 0$ a bound. We define inductively the translation of φ over path number $n \in F_k(\varphi)$ starting at symbolic state $w_{m,n}$ as shown below. Let $min(A) = A'$, then:

- $[\mathbf{true}]_k^{[m,n,A]} := \mathbf{true}$, • $[\mathbf{false}]_k^{[m,n,A]} := \mathbf{false}$,
- $[p]_k^{[m,n,A]} := p(w_{m,n})$, • $[\neg p]_k^{[m,n,A]} := \neg p(w_{m,n})$,
- $[\alpha \wedge \beta]_k^{[m,n,A]} := [\alpha]_k^{[m,n,g_l(A,f_k(\alpha))]} \wedge [\beta]_k^{[m,n,g_r(A,f_k(\beta))]}$,
- $[\alpha \vee \beta]_k^{[m,n,A]} := [\alpha]_k^{[m,n,g_l(A,f_k(\alpha))]} \vee [\beta]_k^{[m,n,g_l(A,f_k(\beta))]}$,
- $[\mathrm{EX}\alpha]_k^{[m,n,A]} := \begin{cases} (1)\ H(w_{m,n}, w_{0,A'}) \wedge [\alpha]_k^{[1,A',g_s(A)]}, & \text{if } k > 0 \\ (2)\ \mathbf{false}, & \text{otherwise} \end{cases}$
- $[\mathrm{E}(\alpha \mathrm{U}_I \beta)]_k^{[m,n,A]} := H(w_{m,n}, w_{0,A'}) \wedge \bigvee_{i=0}^{k}([\beta]_k^{[i,A',h_{\mathrm{U}}(A,k,f_k(\beta))(k)]} \wedge In(i, I) \wedge \bigwedge_{j=0}^{i-1}[\alpha]_k^{[j,A',h_{\mathrm{U}}(A,k,f_k(\beta))(j)]})$,
- $[\mathrm{EG}_I\alpha]_k^{[m,n,A]} := H(w_{m,n}, w_{0,A'}) \wedge$
  $(1)\ \bigwedge_{j=left(I)}^{right(I)}[\alpha]_k^{[j,A',h_{\mathrm{G}}(A,k)(j)]}$, if $right(I) \leq k$
  $(2)\ \bigvee_{l=0}^{k-1}(H(w_{k,A'}, w_{l,A'}) \wedge \bigwedge_{j=min(left(I),l)}^{k-1}[\alpha]_k^{[j,A',h_{\mathrm{G}}(A,k)(j)]})$, otherwise.
- $[\overline{\mathrm{K}}_c\alpha]_k^{[m,n,A]} := I_\iota(w_{0,A'}) \wedge \bigvee_{j=0}^{k}([\alpha]_k^{[j,A',g_s(A)]} \wedge H_c(w_{m,n}, w_{j,A'}))$,
- $[\overline{\mathcal{O}}_c\alpha]_k^{[m,n,A]} := I_\iota(w_{0,A'}) \wedge \bigvee_{j=0}^{k}([\alpha]_k^{[j,A',g_s(A)]} \wedge HO_c(w_{m,n}, w_{j,A'}))$,
- $[\widehat{\underline{\mathrm{K}}}_c^d\alpha]_k^{[m,n,A]} := I_\iota(w_{0,A'}) \wedge \bigvee_{j=0}^{k}([\alpha]_k^{[j,A',g_s(A)]} \wedge \widehat{H}_c^d(w_{m,n}, w_{j,A'}))$,
- $[\overline{\mathrm{D}}_\Gamma\alpha]_k^{[m,n,A]} := I_\iota(w_{0,A'}) \wedge \bigvee_{j=0}^{k}([\alpha]_k^{[j,A',g_s(A)]} \wedge \bigwedge_{c\in\Gamma} H_c(w_{m,n}, w_{j,A'}))$,
- $[\overline{\mathrm{E}}_\Gamma\alpha]_k^{[m,n,A]} := I_\iota(w_{0,A'}) \wedge \bigvee_{j=0}^{k}([\alpha]_k^{[j,A',g_s(A)]} \wedge \bigvee_{c\in\Gamma} H_c(w_{m,n}, w_{j,A'}))$,
- $[\overline{\mathrm{C}}_\Gamma\alpha]_k^{[m,n,A]} := [\bigvee_{j=1}^{k}(\overline{\mathrm{E}}_\Gamma)^j\alpha]_k^{[m,n,A]}$.

The theorem below states the correctness and the completeness of the presented translation. It can be proven by induction on the complexity of the given RTECTLKD formula.

Theorem 2. *Let M be a model, and φ an* RTECTLKD *formula. Then for every $k \in \mathbb{N}$, $M \models_k \varphi$ if, and only if, the propositional formula $[M, \varphi]_k$ is satisfiable.*

Now, from Theorems 1 and 2 we get the following.

Corollary 1. *Let M be a model, and φ an* RTECTLKD *formula. Then, $M \models \varphi$ if, and only if, there exists $k \in \mathbb{N}$ such that the propositional formula $[M, \varphi]_k$ is satisfiable.*

4 Experimental Results

Our implementation of the presented BMC method uses Reduced Boolean Circuits (RBC) [1] to represent the propositional formula $[M, \varphi]_k$. An RBC represents subformulae of $[M, \varphi]_k$ by fresh propositions such that each two identical subformulae correspond to the same proposition. Further, our SAT-BMC method for RTCTLKD is, to our best knowledge, the first ones formally presented in the literature. However, to assess how well the new BMC algorithm performs, we compare it with non-BMC BDD-based symbolic model checking algorithm for ECTLKD that is implemented in McMAS (`http://www-lai.doc.ic.ac.uk/mcmas/`).

The tests have been performed on a computer with Intel Xeon 2 GHz processor and 4 GB of RAM, running Linux 2.6, with the default limits of 2 GB of memory and 5400 seconds. The specifications for the described benchmark are given in the universal form, for which we verify the corresponding counterexample formula, i.e., the formula which is negated and interpreted existentially.

FTC. To evaluate our BMC techniques, we analyse a scalable multi-agent system, which is a faulty train controller system (FTC). Figure 1 presents a DIIS composed of three agents: a controller and two trains, but in general the system consists of a controller, and n trains (for $n \geq 2$) that use their own circular tracks for travelling in one direction (states Away (A)). At one point, all trains have to pass through a tunnel (states Tunnel 'T'), but because there is only one track in the tunnel, trains arriving from each direction cannot use it simultaneously. There are colour light signals on both sides of the tunnel, which can be either red (state 'R') or green (state 'G'). 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 colour light signals, however it can be faulty (state 'F'), and thereby it does not serve its purpose. In the figure, the initial states of the controller and the trains are 'G' and 'W' (Waiting in front of the tunnel) respectively, and the transitions with the same label are synchronised. Null actions are omitted in the figure.

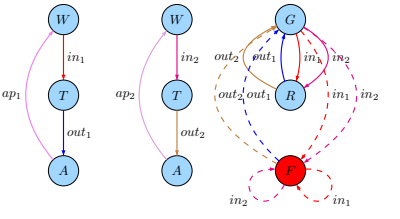


Fig. 1. A DIIS of FTC for 2 trains

Let $\mathcal{PV} = \{inTunnel_1, \ldots inTunnel_n, Red\}$ be a set of propositional variables, which we find useful in analysis of the scenario of the FTC system. A valuation function $\mathcal{V} : S \to 2^{\mathcal{PV}}$ is defined as follows. Let $Ag = \{Train1\ (T1), \ldots, TrainN\ (TN), Controller\ (C)\}$. Then, $inTunnel_c \in \mathcal{V}(s)$ if $l_c(s) = T$ and $c \in Ag \setminus \{C\}$; $Red \in \mathcal{V}(s)$ if $l_C(s) = R$. The specifications are the following:

φ_1 = $\mathrm{AG}_{[0,\infty]}\,\mathcal{O}_C(\bigwedge_{i=1}^{n-1}\bigwedge_{j=i+1}^{n}\neg(InTunnel_i \wedge InTunnel_j))$. "Always when *Controller* is functioning correctly, trains have exclusive access to the tunnel".

φ_2 = $\mathrm{AG}_{[0,\infty]}\mathcal{O}_C(inTunnel_1 \Rightarrow \mathrm{K}_{T1}(\neg inTunnel_2))$. "Always when *Controller* is functioning correctly, then if *Train1* is in the tunnel, it knows that *Train2* is not in the tunnel".

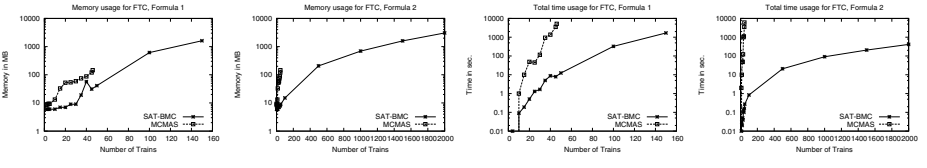
φ_3 = $\mathrm{AG}_{[0,\infty]}(inTunnel_1 \Rightarrow \widehat{\mathrm{K}}^C_{T1}(\bigwedge_{i=2}^{n}(\neg inTunnel_i)))$. "Always when *Train1* is in the tunnel, it knows under assumption that *Controller* is functioning correctly that none of the other trains is in the tunnel".

φ_4 = $\mathrm{AG}_{[0,\infty]}(inTunnel_1 \Rightarrow \widehat{\mathrm{K}}^C_{T1}(Red))$. "Always when *Train1* is in the tunnel, it knows under assumption that *Controller* is functioning correctly that the colour of the light signal for other trains is red".

φ_5 = $\mathrm{AG}_{[0,\infty]}(InTunnel_1 \Rightarrow \mathrm{K}_{T1}(\mathrm{AF}_{[1,n+1]}(\bigvee_{i=1}^{n} InTunnel_i)))$. "Always when *Train1* is in the tunnel, it knows that either he or other train will be in the tunnel during the next $n+1$ time units".

All the above properties are false in our DIIS model of the FTC system.

Performance Evaluation. The experimental results show that our SAT-based BMC significantly outperforms the BDD-based unbounded algorithm of McMAS for φ_1 and φ_2 in both the memory consumption and the execution time (as shown below in the line plots); note that both formulae are in ECTLKD. In the case of φ_1 our SAT-BMC is 3-times better than McMAS, and for φ_2 it is even 43-times better. A noticeable superiority of SAT-BMC for φ_1 and φ_2 follows from the long encoding times of the BDD for the transition relation and very short counterexamples.



Since McMAS does not support the $\widehat{\mathrm{K}}$ modality, we were not able to compare our results with McMAS for the formulae φ_3 and φ_4. Thus, we present results of our method only. Namely, for φ_3 and φ_4 we managed to compute the results for 1100 and 3000 trains, respectively, in the time of 5400 seconds (exact data for 1100 trains: $k = 4$, $f_k(\varphi_3) = 2$, encoding time (bmcT) is 210.12, memory use for encoding (bmcM) is 655.20, satisfiability checking time (satT) is 5258.43, memory use for satisfiability checking (satM) is 1412.00, bmcT+satT is 5468.55, max(bmcM,satM) is 1412.00; exact data for 3000 trains: $k = 1$, $f_k(\varphi_4) = 2$, bmcT is 170.38, bmcM is 1191.00, satT is 18.13, satM is 2356.00, bmcT+satT is 188.51, max(bmcM,satM) is 2356.00).

The formula φ_5 demonstrate that SAT-BMC is indeed a complementary technique to BDD-based unbounded model checking. McMAS was able to check φ_5 (in its equivalent ECTLKD form) for 45 trains in the time of 5400 seconds (memory use: 120MB), and our SAT-BMC succeed to compute the results only for 11 trains (exact data for 11 trains: $k = 21$, $f_k(\varphi_4) = 3$, bmcT is 1.99, bmcM is 4.47, satT is 4914.08, satM

is 224.00, bmcT+satT is 4916.07, max(bmcM,satM) is 224.00). The reason for this is that the length of the counterexamples grows with the number of trains, i.e, for n trains $k = 2n - 1$.

Our future work will involve an implementation of the method also for other models of multi-agent systems, for example for standard interpreted systems. Moreover, we are going to define a BDD-based BMC algorithm for RTECTLKD, and compare it with the method presented in this paper.

References

1. Abdulla, P.A., Bjesse, P., Eén, N.: Symbolic Reachability Analysis Based on SAT-Solvers. In: Graf, S. (ed.) TACAS 2000. LNCS, vol. 1785, pp. 411–425. Springer, Heidelberg (2000)
2. Clarke, E.M., Grumberg, O., Peled, D.A.: Model Checking. The MIT Press, Cambridge (1999)
3. Sistla, A.P., Emerson, E.A., Mok, A.K., Srinivasan, J.: Quantitative temporal reasoning. Real-Time Systems 4(4), 331–352 (1992)
4. Fagin, R., Halpern, J.Y., Moses, Y., Vardi, M.Y.: Reasoning about Knowledge. MIT Press, Cambridge (1995)
5. Jones, A., Lomuscio, A.: A BDD-based BMC approach for the verification of multi-agent systems. In: CS&P 2009, vol. 1, pp. 253–264. Warsaw University (2009)
6. Kacprzak, M., Lomuscio, A., Lasica, T., Penczek, W., Szreter, M.: Verifying Multi-agent Systems via Unbounded Model Checking. In: Hinchey, M.G., Rash, J.L., Truszkowski, W.F., Rouff, C.A. (eds.) FAABS 2004. LNCS (LNAI), vol. 3228, pp. 189–212. Springer, Heidelberg (2004)
7. Lomuscio, A., Penczek, W., Qu, H.: Partial order reduction for model checking interleaved multi-agent systems. In: AAMAS 2010, pp. 659–666. IFAAMAS Press (2010)
8. Lomuscio, A., Sergot, M.: Deontic interpreted systems. Studia Logica 75(1), 63–92 (2003)
9. Męski, A., Penczek, W., Szreter, M.: Bounded model checking linear time and knowledge using decision diagrams. In: CS&P 2011, pp. 363–375. Biaystok University of Technology (2011)
10. Penczek, W., Lomuscio, A.: Verifying epistemic properties of multi-agent systems via bounded model checking. In: AAMAS 2003, pp. 209–216. ACM (2003)
11. Penczek, W., Woźna-Szcześniak, B., Zbrzezny, A.: Towards SAT-based BMC for LTLK over interleaved interpreted systems. In: CS&P 2011, pp. 565–576. Białystok University of Technology (2011)
12. Raimondi, F., Lomuscio, A.: Symbolic Model Checking of Deontic Interpreted Systems via OBDDs. In: Lomuscio, A., Nute, D. (eds.) DEON 2004. LNCS (LNAI), vol. 3065, pp. 228–242. Springer, Heidelberg (2004)
13. Raimondi, F., Lomuscio, A.: Automatic verification of multi-agent systems by model checking via OBDDs. Journal of Applied Logic 5(2), 235–251 (2005)
14. Wooldridge, M.: An introduction to multi-agent systems. John Wiley, England (2002)
15. Woźna, B., Lomuscio, A., Penczek, W.: Bounded model checking for deontic interpreted systems. In: LCMAS 2004. ENTCS, vol. 126, pp. 93–114. Elsevier (2005)
16. Woźna-Szcześniak, B., Zbrzezny, A., Zbrzezny, A.: The BMC Method for the Existential Part of RTCTLK and Interleaved Interpreted Systems. In: Antunes, L., Pinto, H.S. (eds.) EPIA 2011. LNCS (LNAI), vol. 7026, pp. 551–565. Springer, Heidelberg (2011)
17. Zbrzezny, A.: Improving the translation from ECTL to SAT. Fundamenta Informaticae 85(1-4), 513–531 (2008)