SAT-Based BMC for Deontic Metric Temporal Logic and 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 consider multi-agent systems' (MASs) modelled by deontic interleaved interpreted systems and we provide a new SAT-based bounded model checking (BMC) method for these systems. The properties of MASs are expressed by means of the metric temporal logic with discrete semantics and extended to include epistemic and deontic operators. The proposed BMC approach is based on the state of the art solutions to BMC. We test our results on a typical MASs scenario: train controller problem with faults.

1 Introduction

By *agents* we usually mean rational, independent, intelligent and high-tech 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) [27] is a system composed of multiple interacting (communicating, coordinating, cooperating, etc.) agents which can be used to solve problems that are beyond the individual capacities or knowledge of a single agent.

Deontic interpreted systems (DISs) [18] are models of MASs that make possible reasoning about epistemic and correct functioning behaviour of MASs. They provide a computationally grounded semantics on which it is possible to interpret the $\mathcal{O}_i\alpha$ modality, representing the fact "in all correct functioning executions of agent i, α holds", as well as a traditional epistemic modalities and temporal operators. By *deontic interleaved interpreted systems* (DIISs) we mean a restriction of DISs that enforce the executions of agents to be interleaved. Thus we assume that agents act as network of synchronised automata; note that one can see DIISs as a deontic extension of the formalism of interleaved interpreted systems [17]. We consider DIISs since they allow for the distinction between correct (or ideal, normative, etc.) and incorrect states, and they enable more efficient verification of MASs, the behaviour of which is as the behaviour of synchronised automata. Note that although our method is described for DIISs, it can be applied to DISs [10] as well; as it will be clear below the main difference between DIISs and DISs is in the definition of the global evolution function. Thus, to apply our method to DISs it is enough to change the definition a propositional formula that encodes the transition relation. However, only DIISs can be combined with partial order reductions allowing for more efficient verification of MASs that are not so loosely coupled.

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

M. Baldoni et al. (Eds.): DALT 2012, LNAI 7784, pp. 170–189, 2013.

Model checking [6,24] has been developed as a method for automatic verification of finite state concurrent systems, and impressive strides have been made on this problem over the past thirty years. The main aim of model checking 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 discovers a counterexample execution that shows the source of the problem. The practical applicability of model checking in MASs settings requires the development of algorithms hacking the state explosion problem. In particular, to avoid this problem the following approaches have been developed: BDD-based bounded [13,19,20] and unbounded [26,25] model checking, SAT-based bounded [22,23,29,31,30] and unbounded [14] model checking.

To express the requirements of MASs, various extensions of temporal [9] or real time [2] temporal logics with epistemic (to represent knowledge) [10], doxastic (to represent beliefs) [16], and deontic (to represent norms and prescriptions) [18,3] components have been proposed. In this paper we consider a deontic and epistemic extension of Metric Temporal Logic (MTL) [15], which we call MTLKD, and interpret over discrete-time models; note that over the adopted discrete-time model, MTL is simply LTL, but with an exponentially succinct encoding [11]. MTLKD allows for the representation of the quantitative temporal evolution of epistemic states of the agents, as well as their correct and incorrect functioning behaviour. It can express multiple timing constraints on computations, which is really interesting for writing specifications. For example, MTLKD allows to express property asserting that whenever the system finds itself in a p-state, then agent c knows that the system will be in a q-state precisely one time unit later; note that this can be specified by the formula $\mathrm{G}_{[0,\infty)}(p \Rightarrow \mathrm{K}_c\mathrm{F}_{[1,1]}q)$.

In our past research we have provided a theoretical underpinnings of a preliminary bounded model checking (BMC) algorithm for DIS and an existential part of a computation tree logic extended to include an epistemic and deontic modalities (ECTLKD) [29]. 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 an existential part of a computation tree logic (ECTL) [32], which uses a reduced number of paths, what results in significantly smaller and less complicated propositional formulae that encode the ECTLKD properties. In [30] we have defined and experimentally evaluated a BMC algorithm for the existential part of an epistemic and deontic extension of real time CTL (RTCTLKD) [8] 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 main contributions of the paper are as follows. First, we introduce the MTLKD language. Second, we propose a SAT-based BMC technique for DIISs and the existential part of MTLKD. This is the first time when the BMC method for linear time epistemic (and deontic) logics uses a reduced number of paths to evaluate epistemic and deontic components what results in significantly smaller and less complicated propositional formulae that encode the MTLKD properties. Third, we implement the proposed BMC method and evaluate it experimentally. To the best of our knowledge, this

is the first work which provides a practical (bounded) model checking algorithm for the MTLKD language, and the MTL itself.

The structure of the paper is the following. In Section 2 we shortly introduce DIISs and the MTLKD language. In Section 3 we define a bounded semantics for EMTLKD (the existential part of MTLKD) and prove that there is a bound such that both bounded and unbounded semantics for EMTLKD are equivalent. In Section 4 we define a BMC method for MTLKD. In Section 5 we present performance evaluation of our newly developed SAT-based BMC algorithm. In Section 6 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 because 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 S of all possible *global states* as the Cartesian product $\prod_{c=1}^{n} L_c$, such that $L_c \supseteq \mathcal{G}_c$. The set $\mathcal{G}_c$ represents the set of green states for agent c. The complement of $\mathcal{G}_c$ with respect to L_c (denoted by $\mathcal{R}_c$) represents the set of red states for agent c. Note that for any agent c, $L_c = \mathcal{G}_c \cup \mathcal{R}_c$. Further, by $l_c(s)$ we denote the local component of agent $c \in Ag$ in a global state $s = (\ell_1, \ldots, \ell_n)$.

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 $\epsilon_c \in P_c(\ell)$ for any $\ell \in L_c$, i.e., we insist on the null action to be enabled at every local state. For each agent c, there is a (partial) evolution function $t_c : L_c \times Act_c \to L_c$ such that for each $\ell \in L_c$ and for each $a \in P_c(\ell)$ there exists $\ell' \in L_c$ such that $t_c(\ell, a) = \ell'$; moreover, $t_c(\ell, \epsilon_c) = l$ for each $\ell \in L_c$. Note that the local evolution function considered here differs from the standard one (see [10]) 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 $Agent(a) = \{c \in Ag \mid a \in Act_c\}$.

The *global interleaved evolution function* $t : S \times \prod_{i=1}^{n} Act_i \to S$ is defined as follows: $t(s, a_1, \ldots, a_n) = s'$ 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 $t_c(l_c(s), a) = l_c(s')$, and for all $c \in Ag \setminus Agent(a)$, $a_c = \epsilon_c$ and $t_c(l_c(s), a_c) = l_c(s)$. In brief we write the above as $s \xrightarrow{a} s'$.

Note that similarly to blocking synchronisation in automata, the above insists on all agents performing the same non-null action in a global transition; additionally, note that if an agent has the action being performed in its repertoire, it must be performed, for the global transition to be allowed. This assumes that the local protocols are defined to permit this; if a local protocol does not allow it, then the local action cannot be performed

and therefore the global transition does not comply with the definition of interleaving above. As we formally clarify below, we only consider interleaved transitions here.

Now, for a given set of agents Ag and a set of propositional variables $\mathcal{PV}$ we define a *deontic interleaved interpreted system* DIIS as a tuple $(\iota, \{L_c, \mathcal{G}_c, Act_c, P_c, t_c\}_{c\in Ag}, \mathcal{V})$, where $\iota \in S$ is an initial global state, and $\mathcal{V} : S \to 2^{\mathcal{PV}}$ is a valuation function. With such a DIIS we associate a Kripke *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 is the set of global states; $T \subseteq S \times S$ is a global transition (temporal) relation defined by: $(s, s') \in T$ 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. $\mathcal{V}$ assigns to each state a set of propositional variables that are assumed to be true at that state.

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

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

The derived basic modalities are defined as follows: $\alpha \mathrm{R}_I \beta \stackrel{def}{=} \beta \mathrm{U}_I(\alpha \wedge \beta) \vee \mathrm{G}_I\beta$, $\mathrm{F}_I\alpha \stackrel{def}{=} \mathbf{true}\mathrm{U}_I\alpha$, $\mathcal{O}_c\alpha \stackrel{def}{=} \neg\overline{\mathcal{O}}_c\neg\alpha$, $\mathrm{K}_c\alpha \stackrel{def}{=} \neg\overline{\mathrm{K}}_c\neg\alpha$, $\widehat{\mathrm{K}}_c^d\alpha \stackrel{def}{=} \neg\underline{\widehat{\mathrm{K}}}_c^d\neg\alpha$, $\mathrm{D}_\Gamma\alpha \stackrel{def}{=} \neg\overline{\mathrm{D}}_\Gamma\neg\alpha$, $\mathrm{E}_\Gamma\alpha \stackrel{def}{=} \neg\overline{\mathrm{E}}_\Gamma\neg\alpha$, $\mathrm{C}_\Gamma\alpha \stackrel{def}{=} \neg\overline{\mathrm{C}}_\Gamma\neg\alpha$, where $c, d \in \mathcal{AG}$, and $\Gamma \subseteq \mathcal{AG}$.

Intuitively, X, U_I and G_I are the operators, respectively, for "neXt time", "bounded until", and "bounded always". $\mathrm{X}\alpha$ is true in a computation if α is true at the second state of the computation, $\alpha \mathrm{U}_I \beta$ is true in a computation if β is true in the interval I at least in one state and always earlier α holds, and $\mathrm{G}_I\alpha$ is true in a computation if α is true at all the states of the computation that are in the interval I. $\overline{\mathrm{K}}_c$ 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 $\underline{\widehat{\mathrm{K}}}_c^d$ 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". The formula $\overline{\mathcal{O}}_c\alpha$ can be read as "there is a state where agent c is functioning correctly, and in which α holds". 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 $\widehat{\mathrm{K}}_c^d$. We refer to [18] for a discussion of this notion; note that the operator $\overline{\mathcal{O}}_c$ is there referred to as $\mathcal{P}_c$.

The existential fragment of MTLKD (denoted by EMTLKD) is defined by the following grammar:

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

Semantics of MTLKD. Let $M = (\iota, S, T, \{\sim_c\}_{c\in Ag}, \{\bowtie_c\}_{c\in Ag}, \mathcal{V})$ be a model for DIIS. A *path* in M is an infinite sequence $\pi = (s_0, s_1, \ldots)$ of states such that $(s_m, s_{m+1}) \in T$ for each $m \in \mathbb{N}$. For a path π and $m \in \mathbb{N}$, we take $\pi(m) = s_m$. Moreover, the m-th suffix of the path π is defined in the standard way: $\pi^m = (s_m, s_{m+1}, \ldots)$, and the m-th prefix of π is also defined in the standard way: $\pi[..m] = (s_0, s_1, \ldots, s_m)$. 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 MTLKD is the following.

Definition 1. *Let I be an interval in $\mathbb{N}$ of the form: $[a, b)$ or $[a, \infty)$ for $a, b \in \mathbb{N}$, and $m \in \mathbb{N}$. Then, $I + m \stackrel{df}{=} [a + m, b + m)$ if $I = [a, b)$, and $I + m \stackrel{df}{=} [a + m, \infty)$ if $I = [a, \infty)$. A MTLKD formula φ is true (valid) along the path π (in symbols $M, \pi \models \varphi$) iff $M, \pi^0 \models \varphi$, where*

$M, \pi^m \models \mathbf{true}$, $M, \pi^m \not\models \mathbf{false}$,
$M, \pi^m \models p$ *iff* $p \in \mathcal{V}(\pi(m))$, $M, \pi^m \models \neg\alpha$ *iff* $M, \pi^m \not\models \alpha$,
$M, \pi^m \models \alpha \wedge \beta$ *iff* $M, \pi^m \models \alpha$ and $M, \pi^m \models \beta$,
$M, \pi^m \models \alpha \vee \beta$ *iff* $M, \pi^m \models \alpha$ or $M, \pi^m \models \beta$,
$M, \pi^m \models \mathrm{X}\alpha$ *iff* $M, \pi^{m+1} \models \alpha$,
$M, \pi^m \models \alpha \mathrm{U}_I \beta$ *iff* $(\exists i \geqslant m)[i \in I + m$ and $M, \pi^i \models \beta$ and $(\forall m \leqslant j < i) M, \pi^j \models \alpha]$,
$M, \pi^m \models \mathrm{G}_I\alpha$ *iff* $(\forall i \in I + m)[M, \pi^i \models \alpha]$,
$M, \pi^m \models \overline{\mathrm{K}}_c\alpha$ *iff* $(\exists \pi' \in \Pi(\iota))(\exists i \geqslant 0)[\pi(m) \sim_c \pi'(i)$ and $M, \pi'^i \models \alpha]$,
$M, \pi^m \models \overline{\mathcal{O}}_c\alpha$ *iff* $(\exists \pi' \in \Pi(\iota))(\exists i \geqslant 0)[\pi(m) \bowtie_c \pi'(i)$ and $M, \pi'^i \models \alpha]$,
$M, \pi^m \models \underline{\widehat{\mathrm{K}}}_c^d\alpha$ *iff* $(\exists \pi' \in \Pi(\iota))(\exists i \geqslant 0)[\pi(m) \sim_c \pi'(i)$ and $\pi(m) \bowtie_d \pi'(i)$ and $M, \pi'^i \models \alpha]$,
$M, \pi^m \models \overline{Y}_\Gamma\alpha$ *iff* $(\exists \pi' \in \Pi(\iota))(\exists i \geqslant 0)[\pi(m) \sim_\Gamma^Y \pi'(i)$ and $M, \pi'^i \models \alpha]$, where $Y \in \{\mathrm{D}, \mathrm{E}, \mathrm{C}\}$.

A MTLKD formula φ holds in the model M (denoted $M \models \varphi$) iff $M, \pi \models \varphi$ for all the paths $\pi \in \Pi(\iota)$. An EMTLKD formula φ holds in the model M, denoted $M \models^\exists \varphi$, iff $M, \pi \models \varphi$ for some path $\pi \in \Pi(\iota)$. The *existential model checking problem* asks whether $M \models^\exists \varphi$.

3 Bounded Semantics for EMTLKD

The proposed bounded semantics is the backbone of the SAT-based BMC method for EMTLKD, which is presented in the next section. The temporal part of this semantics is based on the bounded semantics presented in [28,33]. As usual, we start by defining *k-paths* and *loops*.

Let $M = (\iota, S, T, \{\sim_c\}_{c\in Ag}, \{\bowtie_c\}_{c\in Ag}, \mathcal{V})$ be a model for DIIS, $k \in \mathbb{N}$, and $0 \leqslant l \leqslant k$. A *$k$-path* π_l is a pair (π, l), where π is a finite sequence $\pi = (s_0, \ldots, s_k)$ of

states such that $(s_j, s_{j+1}) \in T$ for each $0 \leqslant j < k$. A k-path π_l is a *loop* if $l < k$ and $\pi(k) = \pi(l)$. Note that if a k-path π_l is a loop, then it represents the infinite path of the form uv^ω, where $u = (\pi(0), \ldots, \pi(l))$ and $v = (\pi(l+1), \ldots, \pi(k))$. We denote this unique path by $\varrho(\pi_l)$. Note that for each $j \in \mathbb{N}$, $\varrho(\pi_l)^{l+j} = \varrho(\pi_l)^{k+j}$. By $\Pi_k(s)$ we denote the set of all the k-paths starting at s in M.

Let $k \in \mathbb{N}$ be a bound, $0 \leqslant m \leqslant k$, $0 \leqslant l \leqslant k$, and φ an EMTLKD formula. As in the definition of semantics we need to define the satisfiability relation on suffixes of k-paths, we denote by π_l^m the pair (π_l, m), i.e., the k-path π_l together with the designated starting point m. Further, $M, \pi_l^m \models_k \varphi$ denotes that the formula φ is k-true along the suffix $(\pi(m), \ldots, \pi(k))$ of π.

Definition 2. *An EMTLKD formula φ is k-true along the $k-$path π_l (in symbols $M, \pi_l \models_k \varphi$) iff $M, \pi_l^0 \models_k \varphi$, where*

$M, \pi_l^m \models_k \mathbf{true}$, $\quad M, \pi_l^m \not\models_k \mathbf{false}$,

$M, \pi_l^m \models_k p$ *iff* $p \in \mathcal{V}(\pi(m))$, $\quad M, \pi_l^m \models_k \neg p$ *iff* $p \notin \mathcal{V}(\pi(m))$,

$M, \pi_l^m \models_k \alpha \wedge \beta$ *iff* $M, \pi_l^m \models_k \alpha$ and $M, \pi_l^m \models_k \beta$,

$M, \pi_l^m \models_k \alpha \vee \beta$ *iff* $M, \pi_l^m \models_k \alpha$ or $M, \pi_l^m \models_k \beta$,

$M, \pi_l^m \models_k \mathrm{X}\alpha$ *iff* $(m < k$ and $M, \pi_l^{m+1} \models_k \alpha)$ or
$(m = k$ and $l < k$ and $\pi(k) = \pi(l)$ and $M, \pi_l^{l+1} \models_k \alpha)$,

$M, \pi_l^m \models_k \alpha \mathrm{U}_I \beta$ *iff* $(\exists m \leqslant j \leqslant k)(j \in I + m$ and $M, \pi_l^j \models_k \beta$ and $(\forall m \leqslant i < j)$
$M, \pi_l^i \models_k \alpha)$ *or* $(l < m$ and $\pi(k) = \pi(l)$ and $(\exists l < j < m)$
$(j + k - l \in I + m$ and $M, \pi_l^j \models_k \beta$ and $(\forall l < i < j) M, \pi_l^i \models \alpha$
and $(\forall m \leqslant i \leqslant k) M, \pi_l^i \models_k \alpha))$,

$M, \pi_l^m \models_k \mathrm{G}_I \alpha$ *iff* $(k \geqslant \mathit{right}(I+m)$ and $(\forall j \in I + m)\ (M, \pi_l^j \models_k \alpha))$ or
$(k < \mathit{right}(I+m)$ and $\pi(k) = \pi(l)$ and $(\forall \mathit{max} \leqslant j < k)$
$M, \pi_l^j \models_k \alpha$ and $(\forall l \leqslant j < \mathit{max})\ (j + k - l \in I + m$ *implies*
$M, \pi_l^j \models_k \alpha))$, *where* $\mathit{max} = \mathit{max}(\mathit{left}(I + m), m)$,

$M, \pi_l^m \models_k \overline{\mathrm{K}}_c \alpha$ *iff* $(\exists \pi'_{l'} \in \Pi_k(\iota))(\exists 0 \leqslant j \leqslant k)\ (M, {\pi'_{l'}}^j \models_k \alpha$ and $\pi(m) \sim_c \pi'(j))$,

$M, \pi_l^m \models_k \overline{Y}_\Gamma \alpha$ *iff* $(\exists \pi'_{l'} \in \Pi_k(\iota))\ (\exists 0 \leqslant j \leqslant k)\ (M, {\pi'_{l'}}^j \models_k \alpha$ and $\pi(m) \sim_\Gamma^Y \pi'(j))$,

$M, \pi_l^m \models_k \overline{\mathcal{O}}_c \alpha$ *iff* $(\exists \pi'_{l'} \in \Pi_k(\iota))\ (\exists 0 \leqslant j \leqslant k)\ (M, {\pi'_{l'}}^j \models_k \alpha$ and $\pi(m) \bowtie_c \pi'(j))$,

$M, \pi_l^m \models_k \widehat{\underline{\mathrm{K}}}_c^d \alpha$ *iff* $(\exists \pi'_{l'} \in \Pi_k(\iota))\ (\exists 0 \leqslant j \leqslant k)\ (M, {\pi'_{l'}}^j \models_k \alpha$ and $\pi(m) \sim_c \pi'(j)$
and $\pi(m) \bowtie_d \pi'(j))$.

Let M be a model, and φ an EMTLKD formula. We use the following notations: $M \models_k^\exists \varphi$ iff $M, \pi_l \models_k \varphi$ for some $\pi_l \in \Pi_k(\iota)$. The *bounded model checking problem* asks whether there exists $k \in \mathbb{N}$ such that $M \models_k^\exists \varphi$.

Equivalence of the Bounded and Unbounded Semantics. Now, we show that for some particular bound the bounded and unbounded semantics are equivalent.

Lemma 1. *Let M be a model, φ an* EMTLKD *formula, $k \geqslant 0$ a bound, π_l a k-path in M, and $0 \leqslant m \leqslant k$. Then, $M, \pi_l^m \models_k \varphi$ implies*

1. *if π_l is not a loop, then $M, \rho^m \models \varphi$ for each path $\rho \in M$ such that $\rho[..k] = \pi$.*
2. *if π_l is a loop, then $M, \varrho(\pi_l)^m \models \varphi$.*

Proof. (Induction on the length of φ) The lemma follows directly for the propositional variables and their negations. Assume that $M, \pi_l^m \models_k \varphi$ and consider the following cases:

1. $\varphi = \alpha \wedge \beta \mid \alpha \vee \beta \mid \mathrm{X}\alpha$. See the proof of Lemma 2.1 of [33].
2. $\varphi = \alpha \mathrm{U}_I \beta$. Assume that π_l is not a loop. Then $(\exists m \leqslant j \leqslant k)(j \in I + m$ and $M, \pi_l^j \models_k \beta$ and $(\forall m \leqslant i < j)(M, \pi_l^i \models_k \alpha)$. By inductive hypothesis, for every path ρ in M such that $\rho[..k] = \pi$, $(\exists m \leqslant j \leqslant k)(j \in I + m$ and $M, \rho^j \models \beta$ and $(\forall m \leqslant i < j) M, \rho^i \models \alpha)$. Thus, for every path ρ in M such that $\rho[..k] = \pi$, $M, \rho^m \models \varphi$.
   Now assume that π_l is a loop. Then $l < m$ and $\pi(k) = \pi(l)$ and $(\exists l < j < m)$ $(j+k-l \in I+m$ and $M, \pi_l^j \models_k \beta$ and $(\forall l < i < j) M, \pi_l^i \models \alpha$ and $(\forall m \leqslant i \leqslant k)$ $M, \pi_l^i \models_k \alpha)$. By inductive hypothesis, $(\exists l < j < m)(j + k - l \in I + m$ and $M, \varrho(\pi_l)^j \models \beta$ and $(\forall l < i < j)$ $M, \varrho(\pi_l)^i \models \alpha$ and $(\forall m \leqslant i \leqslant k)$ $M, \varrho(\pi_l)^i \models \alpha)$. Since for each $n \in \mathbb{N}$, $\varrho(\pi_l)^{l+n} = \varrho(\pi_l)^{k+n}$, it follows that $M, \varrho(\pi_l)^{j+k-l} \models \beta$ and $(\forall k < i < j + k - l)$ $(M, \varrho(\pi_l)^i \models \alpha)$ and $(\forall m \leqslant i \leqslant k)$ $(M, \varrho(\pi_l)^i \models \alpha)$. Hence, $\varrho(\pi_l)^{j+k-l} \models \beta$ and $(\forall m \leqslant i < j + k - l)$ $(M, \varrho(\pi_l)^i \models \alpha)$. Thus, $M, \varrho(\pi_l)^m \models \varphi$.
3. $\varphi = \mathrm{G}_I \alpha$. Assume that π_l is not a loop. Then $k \geq \mathit{right}(I + m)$ and $(\forall j \in I + m)$ $(M, \pi_l^j \models_k \alpha)$. By inductive hypothesis, for every path ρ in M such that $\rho[..k] = \pi$, $(\forall j \in I + m)(M, \rho^j \models \alpha)$. Thus, for every path ρ in M such that $\rho[..k] = \pi$, $M, \rho^m \models \varphi$.
   Now assume that π_l is a loop, and $max = max(\mathit{left}(I + m), m)$. Then, $k < \mathit{right}(I + m)$ and $\pi(k) = \pi(l)$ and $(\forall max \leqslant j < k)$ $M, \pi_l^j \models_k \alpha$ and $(\forall l \leqslant j < max)$ $(j + k - l \in I + m$ implies $M, \pi_l^j \models_k \alpha)$. By inductive hypothesis, $(\forall max \leqslant j < k)$ $M, \varrho(\pi_l)^j \models \alpha$ and $(\forall l \leqslant j < max)$ $(j + k - l \in I + m$ implies $M, \varrho(\pi_l)^j \models \alpha)$. Since for each $n \in \mathbb{N}$, $\varrho(\pi_l)^{l+n} = \varrho(\pi_l)^{k+n}$, it follows that $(\forall n \in \mathbb{N})$ $(\forall j \geqslant l + n)$ $(j + k - l \in I + m$ implies $M, \varrho(\pi_l)^j \models \alpha)$. Thus, $M, \varrho(\pi_l)^m \models \varphi$.
4. $\varphi = \overline{\mathrm{K}}_c \alpha$. From $M, \pi_l^m \models_k \varphi$ it follows that $(\exists \pi'_{l'} \in \Pi_k(\iota))(\exists 0 \leqslant j \leqslant k)$ $(M, \pi'^j_{l'} \models_k \alpha$ and $\pi(m) \sim_c \pi'(j))$. Assume that both π_l and $\pi'_{l'}$ are not loops. By inductive hypothesis, for every path ρ' in M such that $\rho'[..k] = \pi'$, $(\exists 0 \leqslant j \leqslant k)$ $(M, \rho'^j \models \alpha$ and $\pi(m) \sim_c \rho'(j))$. Further, for every path ρ in M such that $\rho[..k] = \pi$, we have that $\rho(m) \sim_c \rho'(j))$. Thus, for every path ρ in M such that $\rho[..k] = \pi$, $M, \rho^m \models \varphi$.
   Now assume that $\pi'_{l'}$ is not a loop, and π_l is a loop. By inductive hypothesis, for every path ρ' in M such that $\rho'[..k] = \pi'$, $(\exists 0 \leqslant j \leqslant k)(M, \rho'^j \models \alpha$ and $\pi(m) \sim_c \rho'(j))$. Further, observe that $\varrho(\pi_l)(m) = \pi(m)$, thus $M, \varrho(\pi_l)^m \models \varphi$.
   Now assume that both π_l and $\pi'_{l'}$ are loops. By inductive hypothesis, $(\exists 0 \leqslant j \leqslant k)$ $(M, \varrho(\pi'_{l'})^j \models \alpha$ and $\pi(m) \sim_c \varrho(\pi'_{l'})(j))$. Further, observe that $\varrho(\pi_l)(m) = \pi(m)$, thus $M, \varrho(\pi_l)^m \models \varphi$.
   Now assume that $\pi'_{l'}$ is a loop, and π_l is not a loop. By inductive hypothesis, $(\exists 0 \leqslant j \leqslant k)$ $(M, \varrho(\pi'_{l'})^j \models \alpha$ and $\pi(m) \sim_c \varrho(\pi'_{l'})(j))$. Further, for every path ρ in M such that $\rho[..k] = \pi$, we have that $\rho(m) \sim_c \varrho(\pi'_{l'})(j))$. Thus, for every path ρ in M such that $\rho[..k] = \pi$, $M, \rho^m \models \varphi$.
5. Let $\varphi = \overline{Y}_I \alpha$, where $Y \in \{\mathrm{D}, \mathrm{E}, \mathrm{C}\}$, or $\varphi = \overline{\mathcal{O}}_c \alpha$, or $\varphi = \underline{\widehat{\mathrm{K}}}_c^d \alpha$. These cases can be proven analogously to the case 4.

Lemma 2. (Theorem 3.1 of [4]) *Let M be a model, α an LTL formula, and π a path. Then, the following implication holds: $M, \pi \models \alpha$ implies that for some $k \geqslant 0$ and $0 \leqslant l \leqslant k$, $M, \rho_l \models_k \alpha$ with $\pi[..k] = \rho$.*

Since MTL is simply LTL with an exponentially succinct encoding ([11]), every MTL formula γ can be translated into an LTL formula α_γ. Thus, by Lemma 2 we have that the following lemma holds:

Lemma 3. *Let M be a model, α an MTL formula, and π a path. Then, the following implication holds: $M, \pi \models \alpha$ implies that for some $k \geqslant 0$ and $0 \leqslant l \leqslant k$, $M, \rho_l \models_k \alpha$ with $\pi[..k] = \rho$.*

Lemma 4. *Let M be a model, α an* MTL *formula, $Y \in \{\overline{\mathrm{K}}_c, \overline{\mathrm{D}}_\Gamma, \overline{\mathrm{E}}_\Gamma, \overline{\mathrm{C}}_\Gamma, \overline{\mathcal{O}}_c, \widehat{\underline{\mathrm{K}}}_c^d\}$, and π a path. Then, the following implication holds: $M, \pi \models Y\alpha$ implies that for some $k \geqslant 0$ and $0 \leqslant l \leqslant k$, $M, \rho_l \models_k Y\alpha$ with $\pi[..k] = \rho$.*

Proof. Let X^j denote the neXt time operator applied j times, i.e., $X^j = \underbrace{X \ldots X}_{j}$.

1. Let $Y = \overline{\mathrm{K}}_c$. Then $M, \pi \models \overline{\mathrm{K}}_c\alpha$ iff $M, \pi^0 \models \overline{\mathrm{K}}_c\alpha$ iff $(\exists \pi' \in \Pi(\iota))\ (\exists j \geqslant 0)[\pi'(j) \sim_c \pi(0)$ and $M, \pi'^j \models \alpha]$. Since $\pi'(j)$ is reachable from the initial state of M, the checking of $M, \pi'^j \models \alpha$ is equivalent to the checking of $M, \pi'^0 \models \mathrm{X}^j\alpha$. Now since $\mathrm{X}^j\alpha$ is a pure MTL formula, by Lemma 3 we have that for some $k \geqslant 0$ and $0 \leqslant l \leqslant k$, $M, \rho'^0_l \models_k \mathrm{X}^j\alpha$ with $\pi'[..k] = \rho'$. This implies that $M, \rho'^j_l \models_k \alpha$ with $\pi'[..k] = \rho'$, for some $k \geqslant 0$ and $0 \leqslant l \leqslant k$. Now, since $\pi'(j) \sim_c \pi(0)$, we have $\rho'(j) \sim_c \rho(0)$. Thus, by the bounded semantics we have that for some $k \geqslant 0$ and $0 \leqslant l \leqslant k$, $M, \rho_l \models_k \overline{\mathrm{K}}_c\alpha$ with $\pi[..k] = \rho$.
2. Let $Y = \overline{\mathrm{D}}_\Gamma$. Then $M, \pi \models \overline{\mathrm{D}}_\Gamma\alpha$ iff $M, \pi^0 \models \overline{\mathrm{D}}_\Gamma\alpha$ iff $(\exists \pi' \in \Pi(\iota))(\exists j \geqslant 0)[\pi'(j) \sim^{\mathrm{D}}_\Gamma \pi(0)$ and $M, \pi'^j \models \alpha]$. Since $\pi'(j)$ is reachable from the initial state of M, the checking of $M, \pi'^j \models \alpha$ is equivalent to the checking of $M, \pi'^0 \models \mathrm{X}^j\alpha$. Now since $\mathrm{X}^j\alpha$ is a pure MTL formula, by Lemma 3 we have that for some $k \geqslant 0$ and $0 \leqslant l \leqslant k$, $M, \rho'^0_l \models_k \mathrm{X}^j\alpha$ with $\pi'[..k] = \rho'$. This implies that $M, \rho'^j_l \models_k \alpha$ with $\pi'[..k] = \rho'$, for some $k \geqslant 0$ and $0 \leqslant l \leqslant k$. Now, since $\pi'(j) \sim^{\mathrm{D}}_\Gamma \pi(0)$, we have $\rho'(j) \sim^{\mathrm{D}}_\Gamma \rho(0)$. Thus, by the bounded semantics we have that for some $k \geqslant 0$ and $0 \leqslant l \leqslant k$, $M, \rho_l \models_k \overline{\mathrm{D}}_\Gamma\alpha$ with $\pi[..k] = \rho$.
3. Let $Y = \overline{\mathrm{E}}_\Gamma$. Since $\overline{\mathrm{E}}_\Gamma\alpha = \bigvee_{c\in\Gamma} \overline{\mathrm{K}}_c\alpha$, the lemma follows from the case 1.
4. Let $Y = \overline{\mathrm{C}}_\Gamma$. Since $\overline{\mathrm{C}}_\Gamma\alpha = \bigvee_{i=1}^{n} (\overline{\mathrm{E}}_\Gamma)^i\alpha$, where n is the size of the model M, the lemma follows from the case 3.
5. Let $Y = \overline{\mathcal{O}}_c$. Then $M, \pi \models \overline{\mathcal{O}}_c\alpha$ iff $M, \pi^0 \models \overline{\mathcal{O}}_c\alpha$ iff $(\exists \pi' \in \Pi(\iota))\ (\exists j \geqslant 0)[\pi(0) \bowtie_c \pi'(j)$ and $M, \pi'^j \models \alpha]$. Since $\pi'(j)$ is reachable from the initial state of M, the checking of $M, \pi'^j \models \alpha$ is equivalent to the checking of $M, \pi'^0 \models \mathrm{X}^j\alpha$. Now since $\mathrm{X}^j\alpha$ is a pure MTL formula, by Lemma 3 we have that for some $k \geqslant 0$ and $0 \leqslant l \leqslant k$, $M, \rho'^0_l \models_k \mathrm{X}^j\alpha$ with $\pi'[..k] = \rho'$. This implies that $M, \rho'^j_l \models_k \alpha$ with $\pi'[..k] = \rho'$, for some $k \geqslant 0$ and $0 \leqslant l \leqslant k$. Now, since $\pi(0) \bowtie_c \pi'(j)$, we have $\rho(0) \bowtie_c \rho'(j)$ Thus, by the bounded semantics we have that for some $k \geqslant 0$ and $0 \leqslant l \leqslant k$, $M, \rho_l \models_k \overline{\mathcal{O}}_c\alpha$ with $\pi[..k] = \rho$.
6. Let $Y = \widehat{\underline{\mathrm{K}}}_c^d$. This case can be proven analogously to the case 1 and 5.

Lemma 5. *Let M be a model, φ an* EMTLKD *formula, and π a path. The following implication holds: $M, \pi \models \varphi$ implies that there exists $k \geqslant 0$ and $0 \leqslant l \leqslant k$ such that $M, \rho_l \models_k \varphi$ with $\rho[..k] = \pi$.*

Proof. (Induction on the length of φ) The lemma follows directly for the propositional variables and their negations. Assume that the hypothesis holds for all the proper subformulas of φ and consider φ to be of the following form:

1. $\varphi = \psi_1 \vee \psi_2 \mid \psi_1 \wedge \psi_2 \mid \mathrm{X}\psi \mid \psi_1 \mathrm{U}_I \psi_2 \mid \mathrm{G}_I \psi$. Straightforward by the induction hypothesis and Lemma 3.
2. Let $\varphi = Y\alpha$, and $Y, Y_1, \ldots, Y_n, Z \in \{\overline{\mathrm{K}}_c, \overline{\mathrm{D}}_\Gamma, \overline{\mathrm{E}}_\Gamma, \overline{\mathrm{C}}_\Gamma, \overline{\mathcal{O}}_c, \underline{\widehat{\mathrm{K}}}_c^d\}$. Moreover, let $Y_1\alpha_1, \ldots, Y_n\alpha_n$ be the list of all "top level" proper Y-subformulas of α (i.e., each $Y_i\alpha_i$ is a subformula of $Y\alpha$, but it is not a subformula of any subformula $Z\beta$ of $Y\alpha$, where $Z\beta$ is different from $Y\alpha$ and from $Y\alpha_i$ for $i = 1, \ldots, n$).
   If this list is empty, then α is a "pure" MTL formula with no nested epistemic modalities. Hence, by Lemma 4 we have $M, \pi \models \varphi$ implies that there exists $k \geqslant 0$ and $0 \leqslant l \leqslant k$ such that $M, \rho_l \models_k \varphi$ with $\rho[..k] = \pi$.
   Otherwise, introduce for each $Y_i\alpha_i$ a new proposition q_i, where $i = 1, \ldots, n$. Using first a translation of MTL formulae to LTL formulae, and then a symbolic state labelling algorithm presented in [7] (for LTL modalities) and [26] (for epistemic and deontic modalities), we can augment with q_i the labelling of each state s of M initialising some run along which the epistemic formula $Y_i\alpha_i$ holds, and then translate the formula α to the formula α', which instead of each subformula $Y_i\alpha_i$ contains adequate propositions q_i. Therefore, we obtain "pure" LTL formula. Hence, by Lemma 4 we have $M, \pi \models \varphi$ implies that there exists $k \geqslant 0$ and $0 \leqslant l \leqslant k$ such that $M, \rho_l \models_k \varphi$ with $\rho[..k] = \pi$.

The following theorem, whose proof follows directly from Lemma 1 and Lemma 5, states that for a given model and an EMTLKD formula there exists a bound k such that the model checking problem ($M \models^\exists \varphi$) can be reduced to the bounded model checking problem ($M \models_k^\exists \varphi$).

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

Further, by straightforward induction on the length of an EMTLKD 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 : EMTLKD $\to \mathbb{N}$ gives a bound on the number of k-paths, which are sufficient to validate a given EMTLKD formula.

In the definition of f_k we assume that each EMTLKD formula is preceded by the "path" quantifier E with the meaning "there exists a path in $\Pi_k(\iota)$"; this assumption is only technical and it makes the definition of f_k easy to implement. Note that in the BMC method we deal with the existential validity ($\models^\exists$) only, so the above assumption is just another way to express this fact. More precisely, let φ be an EMTLKD formula. To calculate the value of $f_k(\varphi)$, we first extend the formula φ to the formula $\varphi' = \mathrm{E}\varphi$. Next, we calculate the value of f_k for φ' in the following way: $f_k(\mathrm{E}\varphi) = f_k(\varphi) + 1$; $f_k(\mathbf{true}) = f_k(\mathbf{false}) = 0$; $f_k(p) = f_k(\neg p) = 0$ for $p \in \mathcal{PV}$; $f_k(\alpha \wedge \beta) = f_k(\alpha) + f_k(\beta)$; $f_k(\alpha \vee \beta) = max\{f_k(\alpha), f_k(\beta)\}$; $f_k(\mathrm{X}\alpha) = f_k(\alpha)$; $f_k(\alpha \mathrm{U}_I \beta) = k \cdot f_k(\alpha) +$

$f_k(\beta)$; $f_k(\mathrm{G}_I\alpha) = (k+1)\cdot f_k(\alpha)$; $f_k(\overline{\mathrm{C}}_\Gamma\alpha) = f_k(\alpha)+k$; $f_k(Y\alpha) = f_k(\alpha)+1$ for $Y \in \{\overline{\mathrm{K}}_c, \overline{\mathcal{O}}_c, \underline{\widehat{\mathrm{K}}}^d_c, \overline{\mathrm{D}}_\Gamma, \overline{\mathrm{E}}_\Gamma\}$.

4 SAT-Based BMC for EMTLKD

Let $M = (\iota, S, T, \{\sim_c\}_{c\in Ag}, \{\bowtie_c\}_{c\in Ag}, \mathcal{V})$ be a model, φ an EMTLKD formula, and $k \geqslant 0$ a bound. The proposed BMC method is based on the BMC encoding presented in [33], and it consists in translating the problem of checking whether $M \models^{\exists}_k \varphi$ holds, to the problem of checking the satisfiability of the propositional formula

$$[M,\varphi]_k \coloneqq [M^{\varphi,\iota}]_k \ \wedge\ [\varphi]_{M,k}$$

The formula $[M^{\varphi,\iota}]_k$ encodes sets of k-paths of M, whose size equals 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. Note that our translation, like the translation from [33], does not require that either all the k-paths used in the translation are loops or none is a loop. Once this translation is defined, checking satisfiability of an EMTLKD 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 $w = (\mathtt{w}_1, \ldots, \mathtt{w}_r)$ (called a *symbolic state*) of propositional variables (called *state variables*). The set of all the propositional state variables we denote by SV.

Since any k-path (π, l) is a pair consisting of a finite sequence of states of length k and a number $l \leqslant k$, to encode it by propositional formula, it suffices to take a finite sequence of symbolic states of length k and a formula that encodes the position $l \leqslant k$. The designated position l can be encoded as a bit vector of the length $t = max(1, \lceil log_2(k+1)\rceil)$. Thus, the position l can be represented by a valuation of a vector $u = (\mathtt{u}_1, \ldots, \mathtt{u}_t)$ (called a *symbolic number*) of propositional variables (called *propositional natural variables*), which not appear among propositional state variables. The set of all the propositional natural variables we denote by NV, and we assume that $SV \cap NV = \emptyset$. Given the above we can define a *symbolic k-path* as a pair $((w_0, \ldots, w_k), u)$ consisting of a finite sequence of symbolic states of length k and a symbolic number. Since in general we may need to consider more than one symbolic k-path, therefore we introduce a notion of the j-th symbolic k-path $\boldsymbol{\pi}_j = ((w_{0,j}, \ldots, w_{k,j}), u_j)$, where $w_{i,j}$ are symbolic states for $0 \leqslant j < f_k(\varphi)$ and $0 \leqslant i \leqslant k$, and u_j is a symbolic number for $0 \leqslant j < f_k(\varphi)$. 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 $PV = SV \cup NV$, and $V : PV \to \{0,1\}$ be a *valuation of propositional variables* (a *valuation* for short). Each valuation induces the functions $\mathbf{S} : SV^r \to \{0,1\}^r$ and $\mathbf{J} : NV^t \to \mathbb{N}$ defined in the following way: $\mathbf{S}((\mathtt{w}_1, \ldots, \mathtt{w}_r)) = (V(\mathtt{w}_1), \ldots, V(\mathtt{w}_r))$, $\mathbf{J}((\mathtt{u}_1, \ldots, \mathtt{u}_t)) = \sum_{i=1}^{t} V(\mathtt{u}_i)\cdot 2^{i-1}$. Moreover, for a symbolic state w and a symbolic number u, by $SV(w)$ and $NV(u)$ we denote, respectively, the set of all the state variables occurring in w, and the set of all the natural variables occurring in u.

Next, let w and w' be two symbolic states such that $SV(w) \cap SV(w') = \emptyset$, and u be a symbolic number. We define the following auxiliary propositional formulae:

- $I_s(w)$ is a formula over $SV(w)$ that is true for a valuation V iff $\mathbf{S}(w) = s$.
- $p(w)$ is a formula over $SV(w)$ that is true for a valuation V iff $p \in \mathcal{V}(\mathbf{S}(w))$ (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 V iff $\mathbf{S}(w) = \mathbf{S}(w')$ (encodes equality of two global states).
- $H_c(w, w')$ is a formula over $SV(w) \cup SV(w')$ that is true for a valuation V iff $l_c(\mathbf{S}(w)) = l_c(\mathbf{S}(w))$ (encodes equality 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 V iff $l_c(\mathbf{S}(w')) \in \mathcal{G}_c$ (encodes an accessibility of a global state in which agent c is functioning correctly).
- $\widehat{H}_c^d(w, w') := H_c(w, w') \wedge HO_d(w, w')$.
- $\mathcal{T}(w, w')$ is a formula over $SV(w) \cup SV(w')$ that is true for a valuation V iff $(\mathbf{S}(w), \mathbf{S}(w')) \in T$ (encodes the transition relation of M).
- $\mathcal{B}_j^{\sim}(u)$ is a formula over $NV(u)$ that is true for a valuation V iff $j \sim \mathbf{J}(u)$, where $\sim \in \{<, \leqslant, =, \geqslant, >\}$.
- $\mathcal{L}_k^l(\boldsymbol{\pi}_j) := \mathcal{B}_k^{>}(u_j) \wedge H(w_{k,j}, w_{l,j})$.

Moreover, let $j \in \mathbb{N}$, and I be an interval. Then,

$$In(j, I) = \begin{cases} \mathbf{true}, & \text{if } j \in I \\ \mathbf{false}, & \text{if } j \notin I \end{cases}$$

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

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

The next step of the reduction to SAT is the transformation of an EMTLKD formula φ into a propositional formula $[\varphi]_{M,k} := [\varphi]_k^{[0,0,F_k(\varphi)]}$, where $F_k(\varphi) = \{j \in \mathbb{N} \mid 0 \leqslant j < f_k(\varphi)\}$, and $[\varphi]_k^{[m,n,A]}$ denotes the translation of φ along the symbolic path $\boldsymbol{\pi}_{m,n}$ with starting point m by using the set A.

For every EMTLKD formula φ the function f_k determines how many symbolic k-paths are needed for translating the formula φ. Given a formula φ and a set A of k-paths such that $|A| = f_k(\varphi)$, we divide the set A into subsets needed for translating the subformulae of φ. To accomplish this goal we need some auxiliary functions that were defined in [33]. We recall the definitions of these functions.

The relation $\prec$ 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 $x < y$.

Now, let $A \subset \mathbb{N}$ be a finite nonempty set, and $n, d \in \mathbb{N}$, where $d \leqslant |A|$. Then,

- $g_l(A, d)$ denotes the subset B of A such that $|B| = d$ and $B \prec A \setminus B$.
- $g_r(A, d)$ denotes the subset C of A such that $|C| = d$ and $A \setminus C \prec C$.
- $g_s(A)$ denotes the set $A \setminus \{min(A)\}$.

- if n divides $|A| - d$, then $hp(A, d, n)$ denotes the sequence $(B_0, \ldots, B_n)$ of subsets of A such that $\bigcup_{j=0}^{n} B_j = A$, $|B_0| = \ldots = |B_{n-1}|$, $|B_n| = d$, and $B_i \prec B_j$ for every $0 \leqslant i < j \leqslant n$.

Now let $h_k^{\mathrm{U}}(A, d) \stackrel{df}{=} hp(A, d, k)$ and $h_k^{\mathrm{G}}(A) \stackrel{df}{=} hp(A, 0, k)$. Note that if $h_k^{\mathrm{U}}(A, d) = (B_0, \ldots, B_k)$, then $h_k^{\mathrm{U}}(A, d)(j)$ denotes the set B_j, for every $0 \leqslant j \leqslant k$. Similarly, if $h_k^{\mathrm{G}}(A) = (B_0, \ldots, B_k)$, then $h_k^{\mathrm{G}}(A)(j)$ denotes the set B_j, for every $0 \leqslant j \leqslant k$.

The functions g_l and g_r are used in the translation of the formulae with the main connective being either conjunction or disjunction. For a given EMTLKD formula $\alpha \wedge \beta$, if a set A is used to translate this formula, then the set $g_l(A, f_k(\alpha))$ is used to translate the subformula α and the set $g_r(A, f_k(\beta))$ is used to translate the subformula β; for a given EMTLKD formula $\alpha \vee \beta$, if a set A is used to translate this formula, then the set $g_l(A, f_k(\alpha))$ is used to translate the subformula α and the set $g_l(A, f_k(\beta))$ is used to translate the subformula β.

The function g_s is used in the translation of the formulae with the main connective $\mathrm{Q} \in \{\overline{\mathrm{K}}_c, \widehat{\underline{\mathrm{K}}}_c^{j}, \mathcal{O}_c, \overline{\mathrm{D}}_\Gamma, \overline{\mathrm{E}}_\Gamma\}$. For a given EMTLKD formula $\mathrm{Q}\alpha$, if a set A is used to translate this formula, then the path of the number $min(A)$ is used to translate the operator Q and the set $g_s(A)$ is used to translate the subformula α.

The function h_k^{U} is used in the translation of subformulae of the form $\alpha \mathrm{U}_I \beta$. If a set A is used to translate the subformula $\alpha \mathrm{U}_I \beta$ at the symbolic k-path $\boldsymbol{\pi}_n$ (with starting point m), then for every j such that $m \leqslant j \leqslant k$, the set $h_k^{\mathrm{U}}(A, f_k(\beta))(k)$ is used to translate the formula β along the symbolic path $\boldsymbol{\pi}_n$ with starting point j; moreover, for every i such that $m \leqslant i < j$, the set $h_k^{\mathrm{U}}(A, f_k(\beta))(i)$ is used to translate the formula α along the symbolic path $\boldsymbol{\pi}_n$ with starting point i. Notice that if k does not divide $|A| - d$, then $h_k^{\mathrm{U}}(A, d)$ is undefined. However, for every set A such that $|A| = f_k(\alpha \mathrm{U}_I \beta)$, it is clear from the definition of f_k that k divides $|A| - f_k(\beta)$.

The function h_k^{G} is used in the translation of subformulae of the form $\mathrm{G}_I \alpha$. If a set A is used to translate the subformula $\mathrm{G}_I \alpha$ along a symbolic k-path $\boldsymbol{\pi}_n$ (with starting point m), then for every j such that $m \leqslant j \leqslant k$ and $j \in I$, the set $h_k^{\mathrm{G}}(A)(j)$, is used to translate the formula α along the symbolic paths $\boldsymbol{\pi}_n$ with starting point j; Notice that if $k + 1$ does not divide $|A|$, then $h_k^{\mathrm{G}}(A)$ is undefined. However, for every set A such that $|A| = f_k(\mathrm{G}_I \alpha)$, it is clear from the definition of f_k that $k + 1$ divides $|A|$.

Let φ be an EMTLKD formula, and $k \geqslant 0$ a bound. We can 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 $A' = min(A)$, $h_k^{\mathrm{U}} = h_k^{\mathrm{U}}(A, f_k(\beta))$ and $h_k^{\mathrm{G}} = h_k^{\mathrm{G}}(A)$, then:

$$
\begin{array}{l}
[\mathbf{true}]_k^{[m,n,A]} := \mathbf{true},\ [\mathbf{false}]_k^{[m,n,A]} := \mathbf{false}, \\
[p]_k^{[m,n,A]} := p(w_{m,n}),\ [\neg p]_k^{[m,n,A]} := \neg p(w_{m,n}), \\
[\alpha \wedge \beta]_k^{[m,n,A]} := [\alpha]_k^{[m,n,g_l(A,f_k(\alpha))]} \wedge [\beta]_k^{[m,n,g_r(A,f_k(\beta))]}, \\
[\alpha \vee \beta]_k^{[m,n,A]} := [\alpha]_k^{[m,n,g_l(A,f_k(\alpha))]} \vee [\beta]_k^{[m,n,g_l(A,f_k(\beta))]}, \\
[\mathrm{X}\alpha]_k^{[m,n,A]} := [\alpha]_k^{[m+1,n,A]}, \text{ if } m < k \\
\qquad\qquad \bigvee_{l=0}^{k-1} (\mathcal{L}_k^l(\boldsymbol{\pi}_n) \wedge [\alpha]_k^{[l+1,n,A]}), \text{ if } m = k \\
[\alpha \mathrm{U}_I \beta]_k^{[m,n,A]} := \bigvee_{j=m}^{k} (In(j, I + m) \wedge [\beta]_k^{[j,n,h_k^{\mathrm{U}}(k)]} \wedge \bigwedge_{i=m}^{j-1} [\alpha]_k^{[i,n,h_k^{\mathrm{U}}(i)]}) \vee \\
\qquad\qquad (\bigvee_{l=0}^{m-1} (\mathcal{L}_k^l(\boldsymbol{\pi}_n)) \wedge \bigvee_{j=0}^{m-1} (\mathcal{B}_j^{>}(u_n) \wedge [\beta]_k^{[j,n,h_k^{\mathrm{U}}(k)]} \wedge \\
\qquad\qquad (\bigvee_{l=0}^{m-1} (\mathcal{B}_l^{=}(u_n) \wedge In(j + k - l, I + m))) \wedge
\end{array}
$$

$$
\begin{array}{ll}
 & \bigwedge_{i=0}^{j-1}(\mathcal{B}_i^{>}(u_n) \rightarrow [\alpha]_k^{[i,n,h_k^{\mathrm{U}}(i)]}) \wedge \bigwedge_{i=m}^{k}[\alpha]_k^{[i,n,h_k^{\mathrm{U}}(i)]})),\\
[\mathrm{G}_I\alpha]_k^{[m,n,A]} & := \text{if } right(I+m) \leqslant k, \text{ then } \bigwedge_{j=max}^{right(I+m)}[\alpha]_k^{[j,n,h_k^{\mathrm{G}}(j)]},\\
 & \text{if } right(I+m) > k, \text{ then } \bigvee_{l=0}^{k-1}(\mathcal{L}_k^l(\boldsymbol{\pi}_n)) \wedge \bigwedge_{j=max}^{k-1}[\alpha]_k^{[j,n,h_k^{\mathrm{G}}(j)]} \wedge\\
 & \bigwedge_{j=0}^{max-1}((\mathcal{B}_j^{\geqslant}(u_n) \wedge (\bigvee_{l=0}^{max-1}(\mathcal{B}_l^{=}(u_n) \wedge In(j+k-l, I+m))))\\
 & \rightarrow [\alpha]_k^{[j,n,h_k^{\mathrm{G}}(j)]}), \text{ where } max = max(left(I+m), m)\\
[\overline{\mathrm{K}}_c\alpha]_k^{[m,n,A]} & := I_\iota(w_{0,A'}) \wedge \bigvee_{j=0}^{k}([\alpha]_k^{[j,A',g_s(A)]} \wedge H_c(w_{m,n}, w_{j,A'})),\\
[\overline{\mathcal{O}}_c\alpha]_k^{[m,n,A]} & := I_\iota(w_{0,A'}) \wedge \bigvee_{j=0}^{k}([\alpha]_k^{[j,A',g_s(A)]} \wedge HO_c(w_{m,n}, w_{j,A'})),\\
[\widehat{\underline{\mathrm{K}}}_c^d\alpha]_k^{[m,n,A]} & := I_\iota(w_{0,A'}) \wedge \bigvee_{j=0}^{k}([\alpha]_k^{[j,A',g_s(A)]} \wedge \widehat{H}_c^d(w_{m,n}, w_{j,A'})),\\
[\overline{\mathrm{D}}_\Gamma\alpha]_k^{[m,n,A]} & := I_\iota(w_{0,A'}) \wedge \bigvee_{j=0}^{k}([\alpha]_k^{[j,A',g_s(A)]} \wedge \bigwedge_{c\in\Gamma} H_c(w_{m,n}, w_{j,A'})),\\
[\overline{\mathrm{E}}_\Gamma\alpha]_k^{[m,n,A]} & := I_\iota(w_{0,A'}) \wedge \bigvee_{j=0}^{k}([\alpha]_k^{[j,A',g_s(A)]} \wedge \bigvee_{c\in\Gamma} H_c(w_{m,n}, w_{j,A'})),\\
[\overline{\mathrm{C}}_\Gamma\alpha]_k^{[m,n,A]} & := [\bigvee_{j=1}^{k}(\overline{\mathrm{E}}_\Gamma)^j\alpha]_k^{[m,n,A]}.
\end{array}
$$

Now, let α be an EMTLKD formula. For every subformula φ of α, we denote by $[\varphi]_k^{[\alpha,m,n,A]}$ the propositional formula $[M]_k^{F_k(\alpha)} \wedge [\varphi]_k^{[m,n,A]}$, where $[M]_k^{F_k(\alpha)} = \bigwedge_{j=0}^{f_k(\alpha)-1}\bigwedge_{i=0}^{k-1} \mathcal{T}(w_{i,j}, w_{i+1,j}) \wedge \bigwedge_{j=0}^{f_k(\alpha)-1}\bigvee_{l=0}^{k} B_l^{=}(u_j)$. We write $V \Vdash \xi$ to mean that the valuation V satisfies the propositional formula ξ. Moreover, we write $s_{i,j}$ instead of $\mathbf{S}(w_{i,j})$, and l_j instead of $\mathbf{J}(u_j)$.

The lemmas below state the correctness and the completeness of the presented translation respectively.

Lemma 6. *Correctness of the translation Let M be a model, α an* EMTLKD *formula, and $k \in \mathbb{N}$. For every subformula φ of the formula α, every $(m,n) \in \{0,\ldots,k\} \times F_k(\alpha)$, every $A \subseteq F_k(\alpha) \setminus \{n\}$ such that $|A| = f_k(\varphi)$, and every valuation V, the following condition holds: $V \Vdash [\varphi]_k^{[\alpha,m,n,A]}$ implies $M, ((s_{0,n},\ldots,s_{k,n}), l_n)^m \models_k \varphi$.*

Proof. Let $n \in F_k(\alpha)$, A be a set such that $A \subseteq F_k(\alpha) \setminus \{n\}$ and $|A| = f_k(\varphi)$, m be a natural number such that $0 \leqslant m \leqslant k$ and V a valuation. Suppose that $V \Vdash [\varphi]_k^{[\alpha,m,n,A]}$ and consider the following cases:

1. Let $\varphi = p \mid \neg p \mid \psi_1 \vee \psi_2 \mid \psi_1 \wedge \psi_2 \mid \mathrm{X}\psi$ with $p \in \mathcal{PV}$. See Lemma 3.1. of [33].
2. $\varphi = \psi_1 \mathrm{U}_I \psi_2$. Denote by A_1 the propositional formula $\bigvee_{j=m}^{k}\big(In(j, I+m) \wedge [\psi_2]_k^{[j,n,h_k^{\mathrm{U}}(A,f_k(\psi_2))(k)]} \wedge \bigwedge_{i=m}^{j-1}[\psi_1]_k^{[i,n,h_k^{\mathrm{U}}(A,f_k(\psi_2))(i)]}\big)$, and by A_2 the propositional formula $\bigvee_{l=0}^{m-1}(\mathcal{L}_k^l(\boldsymbol{\pi}_n)) \wedge \bigvee_{j=0}^{m-1}\big(\mathcal{B}_j^{>}(u_n) \wedge [\beta]_k^{[j,n,h_k^{\mathrm{U}}(A,f_k(\beta))(k)]} \wedge (\bigvee_{l=0}^{m-1}(\mathcal{B}_l^{=}(u_n) \wedge In(j+k-l, I+m))) \wedge \bigwedge_{i=0}^{j-1}(\mathcal{B}_i^{>}(u_n) \rightarrow [\alpha]_k^{[i,n,h_k^{\mathrm{U}}(A,f_k(\beta))(i)]}) \wedge \bigwedge_{i=m}^{k}[\alpha]_k^{[i,n,h_k^{\mathrm{U}}(A,f_k(\beta))(i)]}\big)$. Observe that $V \Vdash [\psi_1 \mathrm{U}_I \psi_2]_k^{[\alpha,m,n,A]}$ iff $V \Vdash A_1 \vee A_2$ iff $V \Vdash A_1$ or $V \Vdash A_2$. Let us denote by π_l the k-path $((s_{0,n},\ldots,s_{k,n}), l_n)$, and consider two cases:
   (a) $V \Vdash A_1$. From this we get: $(\exists m \leqslant j \leqslant k)\ \big(j \in I+m$ and $M, \pi_l^j \models_k \psi_2$ and $(\forall m \leqslant i < j) M, \pi_l^i \models_k \psi_1\big)$. Hence $M, \pi_l^m \models_k \psi_1 \mathrm{U}_I \psi_2$.
   (b) $V \Vdash A_2$. From this we get: $l < m$ and $\pi(k) = \pi(l)$ and $(\forall m \leqslant i \leqslant k) M, \pi_l^i \models_k \psi_1$ and $(\exists l < j < m)(j+k-l \in I+m$ and $M, \pi_l^j \models_k \psi_2$ and $(\forall l < i < j) M, \pi_l^i \models \psi_1)$. Hence $M, \pi_l^m \models_k \psi_1 \mathrm{U}_I \psi_2$.

3. $\varphi = \mathrm{G}_I\psi$. If $right(I+m) \leqslant k$, then by B_1 we denote the propositional formula: $\bigwedge_{j=max(left(I+m),m)}^{right(I+m)}[\psi]_k^{[j,n,h_k^{\mathrm{G}}(A)(j)]}$. If $right(I+m) > k$, then by B_2 we denote the propositional formula: $\bigvee_{l=0}^{k-1}(\mathcal{L}_k^l(\boldsymbol{\pi}_n)) \wedge \bigwedge_{j=max(left(I+m),m)}^{k-1}[\psi]_k^{[j,n,h_k^{\mathrm{G}}(A)(j)]} \wedge \bigwedge_{j=0}^{max(left(I+m),m)-1}\big((\mathcal{B}_j^{\geqslant}(u_n) \wedge (\bigvee_{l=0}^{max(left(I+m),m)-1}(\mathcal{B}_l^{=}(u_n) \wedge In(j+k-l, I+m))))\big) \rightarrow [\psi]_k^{[j,n,h_k^{\mathrm{G}}(A)(j)]}\big)$. Observe that $V \Vdash [\mathrm{G}_I\psi]_k^{[\alpha,m,n,A]}$ iff $V \Vdash B_1 \vee B_2$ iff $V \Vdash B_1$ or $V \Vdash B_2$. Let us denote by π_l the k-path $((s_{0,n}, \ldots, s_{k,n}), l_n)$, and consider two cases:
   (a) $V \Vdash B_1$. From this we get: $k \geqslant right(I+m)$ and $(\forall j \in I+m)(M, \pi_l^j \models_k \psi)$. Thus $M, \pi_l^m \models_k \mathrm{G}_I\psi$.
   (b) $V \Vdash B_2$. From this we get: $k < right(I+m)$ and $\pi(k) = \pi(l)$ and $(\forall max \leqslant j < k)\ M, \pi_l^j \models_k \psi$ and $(\forall l \leqslant j < max)(j+k-l \in I+m$ implies $M, \pi_l^j \models_k \psi)$, where $max = max(left(I+m), m)$. Thus $M, \pi_l^m \models_k \mathrm{G}_I\psi$.
4. Let $\varphi = \overline{\mathrm{K}}_c\psi$. Let $n' = \min(A)$, and $\widetilde{\pi}_{l'}$ denotes the k-path $((g_{0,n'}, \ldots, g_{k,n'}), l_{n'})$. By the definition of the translation we have $V \Vdash [\overline{\mathrm{K}}_c\psi]_k^{[\alpha,m,n,A]}$ implies $V \Vdash I_\iota(w_{0,n'}) \wedge \bigvee_{j=0}^{k}([\psi]_k^{[\alpha,j,n',g_s(A)]} \wedge H_c(w_{m,n}, w_{j,n'}))$. Since $V \Vdash H_c(w_{m,n}, w_{j,n'})$ holds, we have $g_{m,n} \sim_c g'_{j,n'}$, for some $j \in \{0, \ldots, k\}$. Therefore, by inductive hypotheses we get $(\exists 0 \leqslant j \leqslant k)(M, \widetilde{\pi}_{l'}^j \models_k \psi$ and $g_{m,n} \sim_c g'_{j,n'})$. Thus we have $M, ((g_{0,n}, \ldots, g_{k,n}), l_n)^m \models_k \overline{\mathrm{K}}_c\psi$.
5. Let $\varphi = \overline{Y}_I\psi$, where $Y \in \{\mathrm{D}, \mathrm{E}, \mathrm{C}\}$, or $\varphi = \overline{\mathcal{O}}_c\psi$, or $\varphi = \underline{\widehat{\mathrm{K}}}_c^d\psi$. These can be proven analogously to Case 4.

Let B and C be two finite sets of indices. Then, by $Var(B)$ we denote the set of all the state variables appearing in all the symbolic states of all the symbolic k-paths whose indices are taken from the set B. Moreover, for every valuation V and every set of indices B, by $V \uparrow B$ we denote the restriction of the valuation V to the set $Var(B)$. Notice that if $B \cap C = \emptyset$, then $Var(B) \cap Var(C) = \emptyset$. This property is used in the proof of the following lemma.

Lemma 7. *Completeness of the translation Let M be a model, $k \in \mathbb{N}$, and α an* EMTLKD *formula such that $f_k(\alpha) > 0$. For every subformula φ of the formula α, every $(m,n) \in \{(0,0)\} \cup \{0, \ldots, k\} \times F_k(\alpha)$, every $A \subseteq F_k(\alpha) \setminus \{n\}$ such that $|A| = f_k(\varphi)$, and every k-path π_l, the following condition holds: $M, \pi_l^m \models_k \varphi$ implies that there exists a valuation V such that $\pi_l = ((s_{0,n}, \ldots, s_{k,n}), l_n)$ and $V \Vdash [\varphi]_k^{[\alpha,m,n,A]}$.*

Proof. First, note that given an EMTLKD formula α, and natural numbers k, m, n with $0 \leqslant m \leqslant k$ and $n \in F_k(\alpha)$, there exists a valuation V such $V \Vdash [M]_k^{F_k(\alpha)}$. This is because M has no terminal states. Now we proceed by induction on the complexity of φ. Let $n \in F_k(\alpha)$, A be a set such that $A \subseteq F_k(\alpha) \setminus \{n\}$ and $|A| = f_k(\varphi)$, ρ_l be a k-path in M, and m be a natural number such that $0 \leqslant m \leqslant k$. Suppose that $M, \pi_l^m \models_k \varphi$ and consider the following cases:

1. Let $\varphi = p \mid \neg p \mid \psi_1 \vee \psi_2 \mid \psi_1 \wedge \psi_2 \mid \mathrm{X}\psi$ with $p \in \mathcal{PV}$. See the proof of Lemma 3.3. of [33].
2. $\varphi = \psi_1 \mathrm{U}_I \psi_2$. Let $A_j = h_k^{\mathrm{U}}(A, f_k(\psi_2))(j)$, for each $0 \leqslant j \leqslant k$. We have to consider two cases:

(a) $(\exists m \leqslant j \leqslant k)\big(j \in I+m$ and $M, \pi_l^j \models_k \psi_2$ and $(\forall m \leqslant i < j) M, \pi_l^i \models_k \psi_1\big)$. By inductive hypothesis, there exist valuations $V_0, V_1, \ldots, V_k$, such that $(\exists m \leqslant j \leqslant k)\big(V_k \Vdash [\psi_2]_k^{[\alpha,j,n,A_k]}$ and $(\forall m \leqslant i < j) V_i \Vdash [\psi_2]_k^{[\alpha,i,n,A_i]}\big)$. Since the family of sets $\{A_j\}_{0\leqslant j\leqslant k}$ is pairwise disjoint and the formula $In(j, I+m)$ has always a constant value equal to **true** or **false**, then there exists a valuation V such that $(\exists m \leqslant j \leqslant k)\big(V \Vdash In(j, I+m) \wedge [\psi_2]_k^{[\alpha,j,n,A_k]}$ and $(\forall m \leqslant i < j) V \Vdash [\psi_2]_k^{[\alpha,i,n,A_i]}\big)$. From this we get that $(\exists m \leqslant j \leqslant k)\ \big(V \Vdash In(j, I+m) \wedge [\psi_2]_k^{[\alpha,j,n,A_k]}$ and $V \Vdash \bigwedge_{i=m}^{j-1} [\psi_1]_k^{[\alpha,i,n,A_i]}\big)$. Hence, $V \Vdash \bigvee_{j=m}^{k} (In(j, I+m) \wedge [\psi_2]_k^{[\alpha,j,n,A_k]} \wedge \bigwedge_{i=m}^{j-1} [\psi_2]_k^{[\alpha,i,n,A_i]})$. Thus, $V \Vdash [\psi_1 \mathrm{U}_I \psi_2]_k^{[\alpha,m,n,A]}$.

(b) $(\exists l < j < m)(j+k-l \in I+m$ and $M, \pi_l^j \models_k \psi_2$ and $(\forall l < i < j) M, \pi_l^i \models \psi_1)$ and $(\forall m \leqslant i \leqslant k) M, \pi_l^i \models_k \psi_1$ and $l < m$ and $\pi(k) = \pi(l)$. By inductive hypothesis, there exist valuations $V_0, V_1, \ldots, V_k$, such that $(\exists l < j < m)(V_k \Vdash [\psi_2]_k^{[\alpha,j,n,A_k]}$ and $(\forall m \leqslant i < j)\, V_i \Vdash [\psi_2]_k^{[\alpha,i,n,A_i]})$ and $(\forall m \leqslant i \leqslant k)\, V_i \Vdash [\psi_2]_k^{[\alpha,i,n,A_i]}$. Since the family of sets $\{A_j\}_{0\leqslant j\leqslant k}$ is pairwise disjoint, and the formula $In(j+k-l, I+m)$ has always a constant value equal to **true** or **false**, there exists a valuation V such that $(\exists l < j < m)(V \Vdash In(j+k-l, I+m) \wedge [\psi_2]_k^{[\alpha,j,n,A_k]}$ and $(\forall l < i < j)\, V \Vdash [\psi_2]_k^{[\alpha,i,n,A_i]})$ and $(\forall m \leqslant i \leqslant k)\, V \Vdash [\psi_2]_k^{[\alpha,i,n,A_i]}$. Moreover, $V \Vdash \mathcal{L}_k^l(\boldsymbol{\pi}_n)$. From this we get: $V \Vdash \bigvee_{j=0}^{m-1} (\mathcal{B}_j^{>}(u_n) \wedge [\psi_2]_k^{[j,n,h_k^{\mathrm{U}}(A,f_k(\psi_2))(k)]} \wedge (\bigvee_{l=0}^{m-1} (\mathcal{B}_l^{=}(u_n) \wedge In(j+k-l, I+m))) \wedge \bigwedge_{i=0}^{j-1} (\mathcal{B}_i^{>}(u_n) \rightarrow [\psi_1]_k^{[i,n,h_k^{\mathrm{U}}(A,f_k(\psi_2))(i)]}) \wedge \bigwedge_{i=m}^{k} [\psi_1]_k^{[i,n,h_k^{\mathrm{U}}(A,f_k(\psi_2))(i)]})$ and $V \Vdash \bigvee_{l=0}^{m-1} \mathcal{L}_k^l(\boldsymbol{\pi}_n)$. Hence, $V \Vdash [\psi_1 \mathrm{U}_I \psi_2]_k^{[\alpha,m,n,A]}$.

3. $\varphi = \mathrm{G}_I \psi$. Let $A_j = h_k^{\mathrm{G}}(A)(j)$, for each $0 \leqslant j \leqslant k$. We have to consider two cases:

(a) $k \geq right(I+m)$ and $(\forall j \in I+m)(M, \pi_l^j \models_k \alpha)$. By inductive hypothesis, there exist valuations $V_0, V_1, \ldots, V_k$, such that $(\forall j \in I+m)\, V_j \Vdash [\psi]_k^{[\alpha,j,n,A_j]}$. Since the family of sets $\{A_j\}_{0\leqslant j\leqslant k}$ is pairwise disjoint, there exists a valuation V such that $(\forall j \in I+m)\, V \Vdash [\psi]_k^{[\alpha,j,n,A_j]}$. From this we get: $V \Vdash \bigwedge_{j=max(left(I+m),m)}^{right(I+m)} [\alpha]_k^{[j,n,h_k^{\mathrm{G}}(A)(j)]}$. Hence, $V \Vdash [\mathrm{G}_I \psi]_k^{[\alpha,m,n,A]}$.

(b) $k < right(I+m)$ and $\pi(k) = \pi(l)$ and $(\forall max \leqslant j < k)\ M, \pi_l^j \models_k \alpha$ and $(\forall l \leqslant j < max)(j+k-l \in I+m$ implies $M, \pi_l^j \models_k \alpha)$, where $max = max(left(I+m), m)$. By inductive hypothesis, there exist valuations $V_0, V_1, \ldots, V_k$, such that $(\forall max \leqslant j < k)\, V_j \Vdash [\psi]_k^{[\alpha,j,n,A_j]}$ and $(\forall l \leqslant j < max)(j+k-l \in I+m$ implies $V_j \Vdash [\psi]_k^{[\alpha,j,n,A_j]})$. Since the family of sets $\{A_j\}_{0\leqslant j\leqslant k}$ is pairwise disjoint, there exists a valuation V such that $(\forall max \leqslant j < k)\, V \Vdash [\psi]_k^{[\alpha,j,n,A_j]}$ and $(\forall l \leqslant j < max)(j+k-l \in I+m$ implies $V \Vdash [\psi]_k^{[\alpha,j,n,A_j]})$. Moreover, $V \Vdash \mathcal{L}_k^l(\boldsymbol{\pi}_n)$. From this we get: $V \Vdash \bigvee_{l=0}^{k-1} (\mathcal{L}_k^l(\boldsymbol{\pi}_n)) \wedge \bigwedge_{j=max}^{k-1} [\psi]_k^{[j,n,h_k^{\mathrm{G}}(A)(j)]} \wedge \bigwedge_{j=0}^{max-1} ((\mathcal{B}_j^{\geqslant}(u_n) \wedge (\bigvee_{l=0}^{max-1} (\mathcal{B}_l^{=}(u_n) \wedge In(j+k-l, I+m)))) \rightarrow [\psi]_k^{[j,n,h_k^{\mathrm{G}}(A)(j)]})$. Hence, $V \Vdash [\mathrm{G}_I \psi]_k^{[\alpha,m,n,A]}$.

4. Let $\varphi = \overline{\mathrm{K}}_c\psi$. Since $M, \pi_l^m \models_k \overline{\mathrm{K}}_c\psi$, we have that $(\exists \pi'_{l'} \in \Pi_k(\iota))(\exists 0 \leqslant j \leqslant k)$ $(M, \pi'^j_{l'} \models_k \psi$ and $\pi(m) \sim_c \pi'(j))$. Let $n' = \min(A)$ and $B = g_s(A)$. By the inductive hypothesis and the definition of the formula H_c, there exists a valuation V' such that $V' \Vdash [M]_k^{F_k(\alpha)}$ and $V' \Vdash [\psi]_k^{[j,n',B]} \wedge H_c(w_{m,n}, w_{j,n'})$ for some $j \in \{0, \ldots, k\}$. Hence we have $V' \Vdash \bigvee_{j=0}^{k}([\psi]_k^{[j,n',B]} \wedge H_c(w_{m,n}, w_{j,n'}))$. Further, since $\pi'_{l'} \in \Pi_k(\iota)$, $\pi'_{l'}(0) = \iota$. Thus, by the definition of the formula I, we get that $V' \Vdash I_\iota(w_{0,n'})$. Therefore we have $V' \Vdash I_\iota(w_{0,n'}) \wedge \bigvee_{j=0}^{k}([\psi]_k^{[j,n',B]} \wedge H_c(w_{m,n}, w_{j,n'}))$, which implies that $V' \Vdash [\overline{\mathrm{K}}_c\psi]_k^{[m,n,A]}$. Since $n' \notin B$ and $n \notin A$, there exists a valuation V such that $V \uparrow B = V' \uparrow B$ and moreover $V \Vdash [M]_k^{F_k(\alpha)}$ and $V \Vdash [\overline{\mathrm{K}}_c\psi]_k^{[m,n,A]}$. Therefore we get $V \Vdash [\overline{\mathrm{K}}_c\psi]_k^{[\alpha,m,n,A]}$.
5. Let $\varphi = \overline{Y}_\Gamma\psi$, where $Y \in \{\mathrm{D}, \mathrm{E}, \mathrm{C}\}$, or $\varphi = \overline{\mathcal{O}}_c\psi$, or $\varphi = \widehat{\underline{\mathrm{K}}}_c^d\psi$. These can be proven analogously to Case 4.

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

Proof. ($\Longrightarrow$) Let $k \in \mathbb{N}$ and $M, \pi_l \models_k \varphi$ for some $\pi_l \in \Pi_k(\iota)$. By Lemma 7 it follows that there exists a valuation V such that $\pi_l = ((s_{0,0}, \ldots, s_{k,0}), l_0)$ with $\mathbf{S}(w_{0,0}) = s_{0,0} = \iota$ and $V \Vdash [\varphi]_k^{[\varphi,0,0,F_k(\varphi)]}$. Hence, $V \Vdash I(w_{0,0}) \wedge [M]_k^{F_k(\varphi)} \wedge [\varphi]_k^{[0,0,F_k(\varphi)]}$. Thus $V \Vdash [M, \varphi]_k$.
($\Longleftarrow$) Let $k \in \mathbb{N}$ and $[M, \varphi]_k$ is satisfiable. It means that there exists a valuation V such that $V \Vdash [M, \varphi]_k$. So, $V \Vdash I(w_{0,0})$ and $V \Vdash [M]_k^{F_k(\varphi)} \wedge [\varphi]_k^{[0,0,F_k(\varphi)]}$. Hence, by Lemma 6 it follows that $M, ((s_{0,0}, \ldots, s_{k,0}), l_0) \models_k \varphi$ and $\mathbf{S}(w_{0,0}) = s_{0,0} = \iota$. Thus $M \models_k^\exists \varphi$.

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

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

5 Experimental Results

Our SAT-base BMC method for EMTLKD is, to our best knowledge, the first one formally presented in the literature, and moreover there is no any other model checking technique for the considered EMTLKD language. Further, 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[1]. For the tests we have used a computer with Intel Core i3-2125 processor,

[1] Following van der Meyden at al. [12], instead of using RBCs, we could directly encode $[M, \varphi]_k$ in such a way that each subformula ψ of $[M, \varphi]_k$ occurring within a scope of a k-element disjunction or conjunction is replaced with a propositional variable p_ψ and the reduced formula $[M, \varphi]_k$ is conjuncted with the implication $p_\psi \Rightarrow \psi$. However, in this case our method, as the one proposed in [12], would not be complete.

8 GB of RAM, and running Linux 2.6. We set the timeout to 5400 seconds, and memory limit to 8GB, and we used the state of the art SAT-solver MiniSat 2. 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.

To evaluate our technique, we have analysed 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 $n \geqslant 2$) 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'), i.e., a faulty traffic light remains green when a train enters the tunnel, 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.

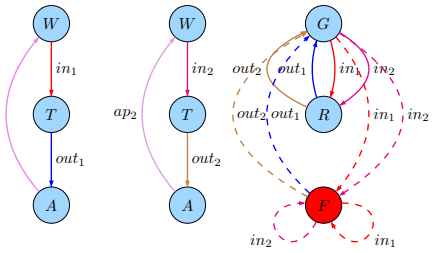


Fig. 1. An DIIS of FTC for two trains. Null actions are omitted.

Let $\mathcal{PV} = \{inT_1, \ldots inT_n, Red\}$ 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, $inT_c \in \mathcal{V}(s)$ 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:

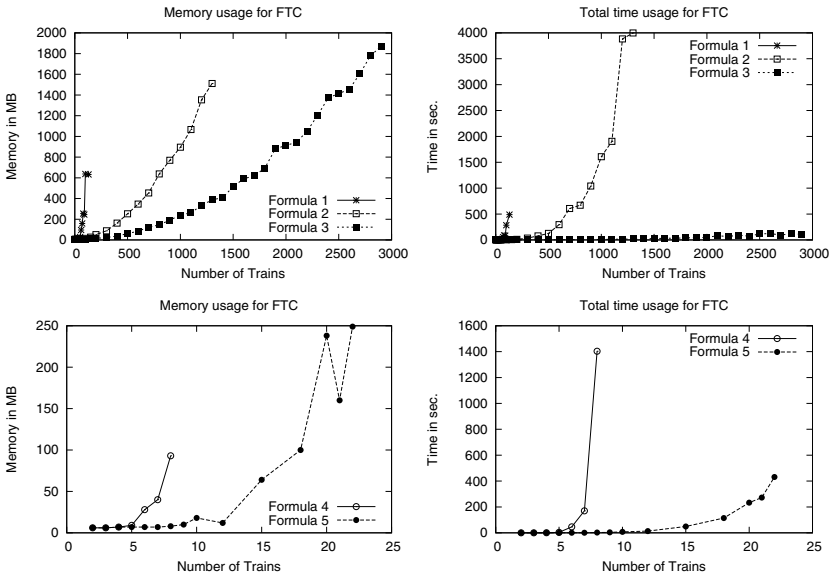
φ_1 = $\mathrm{G}_{[0,\infty]}\ \mathcal{O}_C(\bigwedge_{i=1}^{n-1}\bigwedge_{j=i+1}^{n}\neg(InT_i \wedge InT_j))$. "Always when *Controller* is functioning correctly, trains have exclusive access to the tunnel".

φ_2 = $\mathrm{G}_{[0,\infty]}(inT_1 \Rightarrow \widehat{\mathrm{K}}^C_{T1}(\bigwedge_{i=2}^{n}(\neg inT_i)))$. "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_3 = $\mathrm{G}_{[0,\infty]}(inT_1 \Rightarrow \widehat{\mathrm{K}}^C_{T1}(Red))$. "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_4 = $\mathrm{G}_{[0,\infty]}(InT_1 \Rightarrow \mathrm{K}_{T1}(\mathrm{F}_{[1,n+1]}(\bigvee_{i=1}^{n} InT_i)))$. "Always when *Train1* is in the tunnel, it knows that either it or other train will be in the tunnel during the next $n+1$ time units".

φ_5 = $\mathrm{G}_{[0,\infty]}(InT_1 \Rightarrow \mathrm{K}_{T1}(\mathrm{G}_{[3m-2,3m-2]}InT_1 \vee \mathrm{F}_{[1,n+1]}(\bigvee_{i=2}^{n} InT_i)))$, where $m \geqslant 2$. "Always when *Train1* is in the tunnel, it knows that either he is in the tunnel every $3m-2$ time units 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. Since there is no model checker that supports the EMTLKD properties, we were not able to compare our results with others for the above formulae; McMAS [25] is the only model checker that supports deontic modalities, however it is designated for branching time logics only. Thus, we present results of our method only. An evaluation is given by means of the running time and the memory used, and it is presented on the included line-charts. It can be observed that for φ_1, φ_2, φ_3, φ_4 and φ_5 we managed to compute the results for 130, 1300, 2900, 8, and 22 trains, respectively, in the time of 5400 seconds. The exact data for the mentioned maximal number of trains are the following:

φ_1: $k = 4$, $f_k(\varphi_1) = 2$, bmcT is 5.44, bmcM is 14.00, satT is 483.61, satM is 632.00, bmcT+satT is 489.05, max(bmcM,satM) is 632.00;

φ_2: $k = 4$, $f_k(\varphi_2) = 2$, bmcT is 148.02, bmcM is 909.00, satT is 3850.09, satM 1511.00, bmcT+satT is 3998.11, max(bmcM,satM) is 1511.00;

φ_3: $k = 1$, $f_k(\varphi_3) = 2$, bmcT is 98.89, bmcM is 1114.00, satT is 9.69, satM 1869.00, bmcT+satT is 108.58, max(bmcM,satM) is 1869.00;

φ_4: $k = 24$, $f_k(\varphi_4) = 2$, bmcT is 2.00, bmcM is 3.57, satT is 1401.24, satM 93.00, bmcT+satT is 1403.24, max(bmcM,satM) is 93.00;

φ_5: $k = 65$, $f_k(\varphi_5) = 2$, bmcT is 281.50, bmcM is 18.13, satT is 149.59, satM 249.00, bmcT+satT is 431.10, max(bmcM,satM) is 249.00,

where k is the bound, $f_k(\varphi)$ is the number of symbolic paths, bmcT is the encoding time, bmcM is memory use for encoding, satT is satisfiability checking time, satM is memory use for satisfiability checking.

The formulae φ_1, φ_2 and φ_3 corroborates the efficiency of the SAT-based BMC methods when the length of the counterexamples does not grow with the number of agents (trains). On the other hand the formulae φ_4 and φ_5 demonstrate that SAT-based

BMC becomes inefficient when the the length of the counterexamples grows with the number of agents (trains).

6 Conclusions

We have proposed, implemented, and experimentally evaluated a BMC method for EMTLKD interpreted over deontic interleaved interpreted systems. The experimental results show that the method is very promising.

In [21] it has been shown that the BDD- and SAT-based BMC approaches for ELTLK (an existential part of LTL that is extended with epistemic operators) are complementary. This result is consistent with comparisons for pure temporal logics [5]. Thus, in the future we are going to check whether the same results we can get for DIIS and EMTLKD. Therefore, we are going to define and implement a BDD-based BMC algorithm for EMTLKD, and compare it with the method presented in this paper.

In [20] the semantics of interpreted systems (IS) and interleaved interpreted systems (IIS) were experimentally evaluated by means of the BDD-based bounded model checking method for LTLK. IIS restrict IS by enforcing asynchronous semantics. The paper shows that the modelling approach has a very strong impact on the efficiency of verification by means of BMC. Thus, our future work will involve an implementation of the method for deontic interpreted systems and a comparison of the SAT-based BMC for DIS 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. Alur, R., Henzinger, T.A.: Logics and Models of Real Time: A Survey. In: Huizing, C., de Bakker, J.W., Rozenberg, G., de Roever, W.-P. (eds.) REX 1991. LNCS, vol. 600, pp. 74–106. Springer, Heidelberg (1992)
3. Aqvist, L.: Deontic logic. In: Handbook of Philosophical Logic. Extensions of Classical Logic, vol. II, pp. 605–714. Reidel, Dordrecht (1984)
4. Biere, A., Heljanko, K., Junttila, T., Latvala, T., Schuppan, V.: Linear encodings of bounded LTL model checking. Logical Methods in Computer Science 2(5:5), 1–64 (2006)
5. Cabodi, G., Camurati, P., Quer, S.: Can BDD compete with SAT solvers on bounded model checking? In: Proceedings of DAC 2002, pp. 117–122 (2002)
6. Clarke, E.M., Allen Emerson, E.: Design and Synthesis of Synchronization Skeletons for Branching-Time Temporal Logic. In: Kozen, D. (ed.) Logic of Programs 1981. LNCS, vol. 131, pp. 52–71. Springer, Heidelberg (1982)
7. Clarke, E., Grumberg, O., Hamaguchi, K.: Another Look at LTL Model Checking. In: Dill, D.L. (ed.) CAV 1994. LNCS, vol. 818, pp. 415–427. Springer, Heidelberg (1994)
8. Sistla, A.P., Emerson, E.A., Mok, A.K., Srinivasan, J.: Quantitative temporal reasoning. Real-Time Systems 4(4), 331–352 (1992)
9. Emerson, E.A.: Temporal and modal logic. In: Handbook of Theoretical Computer Science, vol. B, ch. 16, pp. 996–1071. Elsevier Science Publishers (1990)
10. Fagin, R., Halpern, J.Y., Moses, Y., Vardi, M.Y.: Reasoning about Knowledge. MIT Press, Cambridge (1995)
11. Furia, C.A., Spoletini, P.: Tomorrow and All our Yesterdays: MTL Satisfiability over the Integers. In: Fitzgerald, J.S., Haxthausen, A.E., Yenigun, H. (eds.) ICTAC 2008. LNCS, vol. 5160, pp. 126–140. Springer, Heidelberg (2008)

12. Huang, X., Luo, C., van der Meyden, R.: Improved Bounded Model Checking for a Fair Branching-Time Temporal Epistemic Logic. In: van der Meyden, R., Smaus, J.-G. (eds.) MoChArt 2010. LNCS, vol. 6572, pp. 95–111. Springer, Heidelberg (2011)
13. Jones, A., Lomuscio, A.: A BDD-based BMC approach for the verification of multi-agent systems. In: Proceedings of CS&P 2009, vol. 1, pp. 253–264. Warsaw University (2009)
14. 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)
15. Koymans, R.: Specifying real-time properties with metric temporal logic. Real-Time Systems 2(4), 255–299 (1990)
16. Levesque, H.: A logic of implicit and explicit belief. In: Proceedings of the 6th National Conference of the AAAI, pp. 198–202. Morgan Kaufman (1984)
17. Lomuscio, A., Penczek, W., Qu, H.: Partial order reduction for model checking interleaved multi-agent systems. In: AAMAS, pp. 659–666. IFAAMAS Press (2010)
18. Lomuscio, A., Sergot, M.: Deontic interpreted systems. Studia Logica 75(1), 63–92 (2003)
19. Męski, A., Penczek, W., Szreter, M.: Bounded model checking linear time and knowledge using decision diagrams. In: Proceedings of CS&P 2011, pp. 363–375 (2011)
20. Męski, A., Penczek, W., Szreter, M.: BDD-based Bounded Model Checking for LTLK over Two Variants of Interpreted Systems. In: Proceedings of LAM 2012, pp. 35–50 (2012)
21. Męski, A., Penczek, W., Szreter, M., Woźna-Szcześniak, B., Zbrzezny, A.: Two Approaches to Bounded Model Checking for Linear Time Logic with Knowledge. In: Jezic, G., Kusek, M., Nguyen, N.-T., Howlett, R.J., Jain, L.C. (eds.) KES-AMSTA 2012. LNCS, vol. 7327, pp. 514–523. Springer, Heidelberg (2012)
22. Penczek, W., Lomuscio, A.: Verifying epistemic properties of multi-agent systems via bounded model checking. In: Proceedings of AAMAS 2003, pp. 209–216. ACM (2003)
23. Penczek, W., Woźna-Szcześniak, B., Zbrzezny, A.: Towards SAT-based BMC for LTLK over interleaved interpreted systems. Fundamenta Informaticae 119(3-4), 373–392 (2012)
24. Quielle, J.P., Sifakis, J.: Specification and Verification of Concurrent Systems in CESAR. In: Dezani-Ciancaglini, M., Montanari, U. (eds.) Programming 1982. LNCS, vol. 137, pp. 337–351. Springer, Heidelberg (1982)
25. Raimondi, F., Lomuscio, A.: Automatic Verification of Deontic Properties of Multi-agent Systems. In: Lomuscio, A., Nute, D. (eds.) DEON 2004. LNCS (LNAI), vol. 3065, pp. 228–242. Springer, Heidelberg (2004)
26. Raimondi, F., Lomuscio, A.: Automatic verification of multi-agent systems by model checking via OBDDs. Journal of Applied Logic 5(2), 235–251 (2005)
27. Wooldridge, M.: An introduction to multi-agent systems. John Wiley, England (2002)
28. Woźna, B.: Bounded Model Checking for the universal fragment of CTL*. Fundamenta Informaticae 63(1), 65–87 (2004)
29. Woźna, B., Lomuscio, A., Penczek, W.: Bounded model checking for deontic interpreted systems. In: Proceedings of LCMAS 2004. ENTCS, vol. 126, pp. 93–114. Elsevier (2005)
30. Woźna-Szcześniak, B., Zbrzezny, A.: SAT-Based Bounded Model Checking for Deontic Interleaved Interpreted Systems. In: Jezic, G., Kusek, M., Nguyen, N.-T., Howlett, R.J., Jain, L.C. (eds.) KES-AMSTA 2012. LNCS, vol. 7327, pp. 494–503. Springer, Heidelberg (2012)
31. 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, vol. 7026, pp. 551–565. Springer, Heidelberg (2011)
32. Zbrzezny, A.: Improving the translation from ECTL to SAT. Fundamenta Informaticae 85(1-4), 513–531 (2008)
33. Zbrzezny, A.: A new translation from ECTL* to SAT. Fundamenta Informaticae 120(3-4), 377–397 (2012)