Towards SAT-based BMC for LTLK over Interleaved Interpreted Systems ⋆

Wojciech Penczek[1,2], Bożena Woźna-Szcześniak[3], and Andrzej Zbrzezny[3]

[1] Polish Academy of Sciences, ICS, Ordona 21, 01-237 Warsaw, Poland
[2] University of Podlasie, ICS, Sienkiewicza 51, 08-110 Siedlce, Poland
{penczek}@ipipan.waw.pl
[3] Jan Długosz University, IMCS, Armii Krajowej 13/15, 42-200 Częstochowa, Poland
{b.wozna,a.zbrzezny}@ajd.czest.pl

Abstract. This paper makes two contributions to the verification of multi agent-systems modelled by interleaved interpreted systems. Firstly, the paper presents theoretical underpinnings of the SAT-based bounded model checking (BMC) approach for LTL extended with the epistemic component (LTLK) over interleaved interpreted systems. Secondly, the BMC method has been implemented and tested on several benchmarks for MAS. The preliminary experimental results reveal advantages and disadvantages of our SAT-based BMC for LTLK and show that the method has a significant potential.

1 Introduction

Verification of multi-agent systems (MAS) is an actively developing field of research. Several approaches based on model checking [3] have been put forward for the verification of MAS. Typically, they employ combinations of the epistemic logic with either branching [11, 17] or linear time temporal logic [8, 14, 13, 6, 18]. Some approaches reduce the verification problem to the one for plain temporal logic [2, 8], while others treat typical MAS modalities such as (distributed, common) knowledge as first-class citizens and introduce novel algorithms for them [14, 13, 17].

In an attempt to alleviate the state-space explosion problem (i.e., an exponential growth of the system state space with the number of the agents) two main approaches have been proposed based on combining bounded model (BMC) with symbolic verification using translations to either ordered binary decision diagrams (BDDs) [10] or propositional logic (SAT) [16, 11]. However, the above approaches can deal with the properties expressed in CTLK (i.e., CTL extended with an epistemic component) only.

In this paper we aim at completing the picture of applying BMC-based symbolic verification to MAS by looking at LTLK (i.e., LTL extended with the epistemic component, called also CKL$_n$ [8]), interpreted on a particular class of interpreted systems called *interleaved interpreted systems* (IIS) [12]. IIS are a special class of interpreted systems [5] in which only one action at a time is performed in a global transition. Our original contribution consists in defining a novel SAT-based model checking method for

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

CONCURRENCY, SPECIFICATION AND PROGRAMMING
M. Szczuka et al. (eds.): Proceedings of the international workshop CS&P 2011
September 28-30, Pułtusk, Poland, pp. 565-576

LTLK. Our method has been implemented and tested on two benchmarks for MAS. The preliminary experimental results reveal advantages and disadvantages of our SAT-based BMC for LTLK and show that the method has a significant potential.

The rest of the paper is organised as follows. We begin in Section 2 by presenting IIS and the logic LTLK. In Section 3 we present our SAT-based BMC method for LTLK together with its proof of correctness. In Section 4 we discuss our preliminary experimental results and conclude the paper.

2 Syntax and Semantics of LTLK

In this section we introduce the basic definitions used in the paper. In particular, we define the semantics of interpreted systems and syntax and semantics of LTLK.

The semantics of *interpreted systems* provides a setting to reason about MAS by means of specifications based on knowledge and linear or branching time. We report here the basic setting as popularised in [5], restricted to interleaved interpreted systems [12]. Therefore, we assume that if more than one agent is active at a given state, all the active agents perform the same (shared) action in the round. Note that it is still possible for agents to communicate by means of shared actions.

We begin by assuming a MAS to be composed of n agents[4] $\mathcal{A} = \{1, \ldots, n\}$. We associate a set of *possible local states* L_c and *actions* Act_c to each agent $_c \in \mathcal{A}$. We assume that the special action ϵ_c, called "null", or "silent" action of agent 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. Also note that we do not assume that the sets of actions of the agents to be disjoint. We call $Act = \bigcup_{c \in \mathrm{A}} Act_c$ the union of all the sets Act_c. For each action a by $Agent(a) \subseteq \mathcal{A}$ we mean all the agents c such that $a \in Act_c$, i.e., the set of agents potentially able to perform a. Following closely the interpreted system model, we consider a *local protocol* modelling the program the agent is executing. Formally, for any agent c, the actions of the agents are selected according to a *local protocol* $P_c : L_c \to 2^{Act_c}$; we assume 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, for each $l \in L_c$, $t_c(l, \epsilon_c) = l$. The local evolution function considered here differs from the standard treatment in interpreted systems by having the local action as the only parameter. A *global state* $g = (l_1, \ldots, l_n)$ is a tuple of local states for all the agents in the MAS corresponding to an instantaneous snapshot of the system at a given time. Given a global state $g = (l_1, \ldots, l_n)$, we denote by $g^c = l_c$ the local component of agent $c \in \mathcal{A}$ in g. Let G be a set of global states. The *global interleaved evolution* function $t : G \times \prod_{c=1}^{n} Act_c \to G$ 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 $t_c(g^c, a) = g'^c$, and for all $c \in \mathcal{A} \setminus Agent(a)$, $a_c = \epsilon_c$ and $t_c(g^c, a_c) = g'^c$. In brief we write the above as $g \xrightarrow{a} g'$.

[4] Note in the present study we do not consider the environment component. This may be added with no technical difficulty at the price of heavier notation.

Similar to blocking synchronisation in automata, the above insists on all agents performing the same non-epsilon 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 local protocols are defined in such a way to permit this; if a local protocol does not allow this, 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.

We assume that the global transition relation is total, i.e., that for any $g \in G$ there exists an $a \in Act$ such that $g \xrightarrow{a} g'$ for some $g' \in G$. A sequence of global states and actions $\rho = g_0 a_0 g_1 a_1 g_2 \ldots$ is called an *interleaved path*, or an *interleaved run* (or more simply a *path* or a *run*) originating at g_0 if there is a sequence of interleaved transitions from g_0 onwards, i.e., if $g_i \xrightarrow{a_i} g_{i+1}$ for every $i \geq 0$. The set of interleaved paths originating from g is denoted as $\Pi(g)$. A state g is said to be *reachable* from g_0 if there is an interleaved path $\rho = g_0 a_0 g_1 a_1 g_2 \ldots$ such that $g = g_i$ for some $i \geq 0$.

Definition 1 (Interleaved Interpreted Systems). *Let $\mathcal{PV}$ be a set of propositions. An interleaved interpreted system (IIS), also referred to as a model, is a tuple $M = (G, \iota, \Pi, \{\sim_c\}_{c\in\mathcal{A}}, \mathcal{V})$, where G is a set of global states, $\iota \in G$ is an initial (global) state, $\Pi = \bigcup_{g\in G} \Pi(g)$ is the set of all the interleaved paths originating from all states in G, $\sim_c \subseteq G \times G$ is an* epistemic indistinguishability *relation for each agent $c \in \mathcal{A}$, defined by $g \sim_c r$ if $g^c = r^c$, and $\mathcal{V} : G \to 2^{\mathcal{PV}}$ is a valuation function.*

Combinations of linear time with knowledge have long been used in the analysis of temporal epistemic properties of systems [5]. We now recall the basic definitions here and adapt them to our purposes when needed.

Syntax. Let $\mathcal{PV}$ be a set of atomic propositions to be interpreted over the global states of a system, $p \in \mathcal{PV}$, and $\Gamma \subseteq \mathcal{A}$. Then, the syntax of LTLK is defined by the following BNF grammar:

$$\varphi ::= p \mid \neg\varphi \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \mathrm{X}\varphi \mid \varphi \mathrm{U} \varphi \mid \varphi \mathrm{R} \varphi \mid$$
$$\mathrm{K}_c\varphi \mid \overline{\mathrm{K}}_c\varphi \mid \mathrm{E}_\Gamma\varphi \mid \overline{\mathrm{E}}_\Gamma\varphi \mid \mathrm{D}_\Gamma\varphi \mid \overline{\mathrm{D}}_\Gamma\varphi \mid \mathrm{C}_\Gamma\varphi \mid \overline{\mathrm{C}}_\Gamma\varphi.$$

The temporal operators U and R are named as usual *until* and *release* respectively, X is the next step operator. The operator K_c represents "agent c knows" and $\overline{\mathrm{K}}_c$ is the corresponding dual representing "agent c does not know whether or not something holds". The epistemic operators $\mathrm{D}_\Gamma, \mathrm{E}_\Gamma$, and C_Γ represent distributed knowledge in the group Γ, "everyone in Γ knows", and common knowledge among agents in Γ. $\overline{\mathrm{D}}_\Gamma, \overline{\mathrm{E}}_\Gamma$, and $\overline{\mathrm{C}}_\Gamma$ are the corresponding dual ones.

Semantics. Let $M = (G, \iota, \Pi, \{\sim_c\}_{c\in\mathcal{A}}, \mathcal{V})$ be a model, and ρ be an interleaved path; $\rho(i)$ denote the i-th state of ρ, and $\rho[i]$ denote the path ρ with a designated formula evaluation position i, where $i \in N$. Further, let $\Gamma \subseteq \mathcal{A}$. The union of Γ's epistemic indistinguishability relations is defined as $\sim_\Gamma^E = \bigcup_{c\in\Gamma} \sim_c$, $\sim_\Gamma^C$ denotes the transitive closure of $\sim_\Gamma^E$, whereas $\sim_\Gamma^D = \bigcap_{c\in\Gamma} \sim_c$. Then a LTLK formula φ is true (valid) along the path ρ (in symbols $M, \rho \models \varphi$) iff $M, \rho[0] \models \varphi$, where

- $M, \rho[m] \models p$ iff $p \in \mathcal{V}(\rho(m))$,
- $M, \rho[m] \models \neg\alpha$ iff $M, \rho[m] \not\models \alpha$,
- $M, \rho[m] \models \alpha \wedge \beta$ iff $M, \rho[m] \models \alpha$ and $M, \rho[m] \models \beta$,

- $M, \rho[m] \models \alpha \vee \beta$ iff $M, \rho[m] \models \alpha$ or $M, \rho[m] \models \beta$,
- $M, \rho[m] \models \mathrm{X}\alpha$ iff $M, \rho[m+1] \models \alpha$,
- $M, \rho[m] \models \alpha \mathrm{U} \beta$ iff $(\exists i \geq m)[M, \rho[i] \models \beta$ and $(\forall m \leq j < i) M, \rho[j] \models \alpha]$,
- $M, \rho[m] \models \alpha \mathrm{R} \beta$ iff $(\forall i \geq m)[M, \rho[i] \models \beta]$ or $(\exists i \geq m)[M, \rho[i] \models \alpha$ and $(\forall m \leq j \leq i) M, \rho[j] \models \beta]$,
- $M, \rho[m] \models \mathrm{K}_c \alpha$ iff $(\forall \rho' \in \Pi(\iota))(\forall i \geq 0)[\rho'(i) \sim_c \rho(m)$ implies $M, \rho'[i] \models \alpha]$,
- $M, \rho[m] \models \overline{\mathrm{K}}_c \alpha$ iff $(\exists \rho' \in \Pi(\iota))(\exists i \geq 0)[\rho'(i) \sim_c \rho(m)$ and $M, \rho'[i] \models \alpha]$,
- $M, \rho[m] \models Y\alpha$ iff $(\forall \rho' \in \Pi(\iota))(\forall i \geq 0)[\rho'(i) \sim \rho(m)$ implies $M, \rho'[k] \models \alpha]$,
- $M, \rho[m] \models \overline{Y}\alpha$ iff $(\exists \rho' \in \Pi(\iota))(\exists i \geq 0)[\rho'(i) \sim \rho(m)$ and $M, \rho'[i] \models \alpha]$, where

$Y \in \{\mathrm{D}_\Gamma, \mathrm{E}_\Gamma, \mathrm{C}_\Gamma\}$, $\overline{Y} \in \{\overline{\mathrm{D}}_\Gamma, \overline{\mathrm{E}}_\Gamma, \overline{\mathrm{C}}_\Gamma\}$, and $\sim \in \{\sim_\Gamma^D, \sim_\Gamma^E, \sim_\Gamma^C\}$, resp.

LTL is the sublogic of LTLK which consists only of the formulae built without epistemic operators. ELTLK is the existential fragment of LTLK, defined by the following grammar:

$$\varphi ::= | \; p \mid \neg p \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \mathrm{X}\varphi \mid \varphi \mathrm{U} \varphi \mid \varphi \mathrm{R} \varphi \mid \overline{\mathrm{K}}_c \varphi \mid \overline{\mathrm{E}}_\Gamma \varphi \mid \overline{\mathrm{D}}_\Gamma \varphi \mid \overline{\mathrm{C}}_\Gamma \varphi$$

Moreover, an ELTLK formula φ holds in the model M, denoted $M \models^\exists \varphi$, iff $M, \rho \models \varphi$ for some path $\rho \in \Pi(\iota)$. The intuition behind this definition is that ELTLK is obtained only by restricting the syntax of the epistemic operators while the temporal ones remain the same. We get the existential version of these operators by the change from the universal ($\models$) to the existential quantification ($\models^\exists$) over the paths in the definition of the validity in the model M. Notice that this change is only necessary when φ contains a temporal operator, which is not nested in an epistemic operator.

3 Bounded Model Checking for ELTLK

In this section we present a SAT-based BMC method for LTLK, which is based on ideas from [1, 16]. We first define k-paths, and (k,l)-*loops*, and then in turn we define a bounded semantics for ELTLK, which is later used for translation to SAT.

Let $M = (G, \iota, \Pi, \{\sim_c\}_{c \in \mathcal{A}}, \mathcal{V})$ be a model, and $k \geq 0$. A k-path is the prefix of length k of a path in Π. By P_k we denote a set of all the k-paths. By $P_k(g)$ we mean a set of all the k-paths ρ with $\rho(0) = g$. We call a k-path ρ *initial* iff $\rho \in P_k(\iota)$. We call a k-path ρ a (k,l)-*loop* iff $\rho(l) = \rho(k)$, for some $0 \leq l < k$; note that (k,l)-loop ρ generates the inifinite path of the following form: $\rho = u \cdot v^\omega$ with $u = (\rho(0), \ldots, \rho(l-1))$ and $v = (\rho(l), \ldots, \rho(k-1))$. In the bounded semantics we only look at finite prefixes of paths. In particular, we only consider the first $k+1$ states of a path to define the validity of an ELTLK formula along that path. Also, we introduce a function $loop : P_k \to 2^{\mathbb{N}}$ which identifies these k-paths that are loops. The function is defined as: $loop(\rho) = \{l \mid 0 \leq l < k$ and $\rho(l) = \rho(k)\}$.

Definition 2 (Bounded semantics for a loop). *Given are a model M, an ELTLK formula φ, a bound $k \geq 0$, and a $(k,l)-$loop ρ. Let m and l be parameters indicating the current position and the loop position in the $(k,l)-$loop ρ, respectively. Then an* ELTLK *formula φ is $k-$true along ρ (in symbols $M, \rho \models_k \varphi$) iff $M, \rho[0] \models_{k,l} \varphi$ for some $l < k$, where*

- $M, \rho[m] \models_{k,l} p$ *iff* $p \in \mathcal{V}(\rho(m))$,
- $M, \rho[m] \models_{k,l} \neg p$ *iff* $p \notin \mathcal{V}(\rho(m))$,

- $M, \rho[m] \models_{k,l} \alpha \vee \beta$ *iff* $M, \rho[m] \models_{k,l} \alpha$ *or* $M, \rho[m] \models_{k,l} \beta$,
- $M, \rho[m] \models_{k,l} \alpha \wedge \beta$ *iff* $M, \rho[m] \models_{k,l} \alpha$ *and* $M, \rho[m] \models_{k,l} \beta$,
- $M, \rho[m] \models_{k,l} \mathrm{X}\alpha$ *iff* $m < k$ *and* $M, \rho[m+1] \models_{k,l} \alpha$ *or* $m = k$ *and* $M, \rho[l] \models_{k,l} \alpha$,
- $M, \rho[m] \models_{k,l} \alpha \mathrm{U} \beta$ *iff* $(\exists m \leq i \leq k)(M, \rho[i] \models_{k,l} \beta$ *and* $(\forall m \leq j < i)$ $M, \rho[j] \models_{k,l} \alpha)$,
- $M, \rho[m] \models_{k,l} \alpha \mathrm{R} \beta$ *iff* $(\forall min(l, m) \leq i \leq k)(M, \rho[i] \models_{k,l} \beta)$ *or* $(\exists m \leq i \leq k)(M, \rho[i] \models_{k,l} \alpha$ *and* $(\forall m \leq j \leq i) M, \rho[j] \models_{k,l} \beta)$,
- $M, \rho[m] \models_{k,l} \overline{\mathrm{K}}_c \alpha$ *iff* $(\exists \rho' \in P_k(\iota))(\exists 0 \leq j \leq k)\big((\exists 0 \leq ll < k)(ll \in loop(\rho')$ *and* $M, \rho'[j] \models_{k,ll} \alpha)$ *and* $\rho(m) \sim_c \rho'(j)\big)$,
- $M, \rho[m] \models_{k,l} \overline{Y} \alpha$ *iff* $(\exists \rho' \in P_k(\iota))(\exists 0 \leq j \leq k)\big((\exists 0 \leq ll < k)(M, \rho'[j] \models_{k,ll} \alpha$ *and* $ll \in loop(\rho'))$ *and* $\rho(m) \sim \rho'(j)\big)$, *where* $\overline{Y} \in \{\overline{\mathrm{D}}_\Gamma, \overline{\mathrm{E}}_\Gamma, \overline{\mathrm{C}}_\Gamma\}$ *and* $\sim \in \{\sim_\Gamma^D, \sim_\Gamma^E, \sim_\Gamma^C\}$, *resp.*

Assume that ρ is a k-path; note that the state $\rho(k)$ does not need to have a successor in this case.

Definition 3 (Bounded semantics). *Given are a model M, an* ELTLK *formula φ, a bound $k \geq 0$, and a $k-$path ρ. Let m be a parameter indicating the current position in the $k-$path ρ. Then an* ELTLK *formula φ is $k-$true along ρ (in symbols $M, \rho \models_k \varphi$) iff $M, \rho[0] \models_k \varphi$, where*

- $M, \rho[m] \models_k p$ *iff* $p \in \mathcal{V}(\rho(m))$,
- $M, \rho[m] \models_k \neg p$ *iff* $p \notin \mathcal{V}(\rho(m))$,
- $M, \rho[m] \models_k \alpha \vee \beta$ *iff* $M, \rho[m] \models_k \alpha$ *or* $M, \rho[m] \models_k \beta$,
- $M, \rho[m] \models_k \alpha \wedge \beta$ *iff* $M, \rho[m] \models_k \alpha$ *and* $M, \rho[m] \models_k \beta$,
- $M, \rho[m] \models_k \mathrm{X}\alpha$ *iff* $m < k$ *and* $M, \rho[m+1] \models_k \alpha$,
- $M, \rho[m] \models_k \alpha \mathrm{U} \beta$ *iff* $(\exists m \leq i \leq k)(M, \rho[i] \models_k \beta$ *and* $(\forall m \leq j < i) M, \rho[j] \models_k \alpha)$,
- $M, \rho[m] \models_k \alpha \mathrm{R} \beta$ *iff* $(\exists m \leq i \leq k)(M, \rho[i] \models_k \alpha$ *and* $(\forall m \leq j \leq i) M, \rho[j] \models_k \beta)$,
- $M, \rho[m] \models_k \overline{\mathrm{K}}_c \alpha$ *iff* $(\exists \rho' \in P_k(\iota))(\exists 0 \leq j \leq k)\big(M, \rho'[j] \models_k \alpha$ *and* $\rho(m) \sim_c \rho'(j)\big)$,
- $M, \rho[m] \models_k \overline{Y} \alpha$ *iff* $(\exists \rho' \in P_k(\iota))(\exists 0 \leq j \leq k)\big(M, \rho'[j] \models_k \alpha$ *and* $\rho(m) \sim \rho'(j)\big)$, *where* $\overline{Y} \in \{\overline{\mathrm{D}}_\Gamma, \overline{\mathrm{E}}_\Gamma, \overline{\mathrm{C}}_\Gamma\}$ *and* $\sim \in \{\sim_\Gamma^D, \sim_\Gamma^E, \sim_\Gamma^C\}$, *resp.*

Given are a model M, and an ELTLK formula φ. We use the following notations: $M \models_k^\exists \varphi$ iff $M, \rho \models_k \varphi$ for some $\rho \in P_k(\iota)$.

3.1 Equivalence of the bounded and unbounded semantics

Now we show that for some particular bound the bounded and unbounded semantics are equivalent.

Lemma 1. *Given are a model M, an* ELTLK *formula φ, a path ρ, a bound $k > 0$, and $m \leq k$. Then the following implication holds: for each subformula ψ of φ $M, \rho[m] \models_k \psi$ implies $M, \rho[m] \models \psi$.*

Proof. Straightforward by induction on the length of φ.

Lemma 2. *Given are a model M, an* ELTLK *formula φ, a path ρ which is a (k,l)-loop, a bound $k > 0$, and $m \leq k$. Then the following implication holds: for each subformula ψ of φ $M, \rho[m] \models_{k,l} \psi$ implies $M, \rho[m] \models \psi$.*

Proof. Straightforward by induction on the length of φ.

Lemma 3. *Given an LTL formula α and a model M. Then, the following implication holds: if $M \models^{\exists} \alpha$, then there exists $k \leq |M| \cdot |\alpha| \cdot 2^{|\alpha|}$ with $M \models^{\exists}_{k} \alpha$.*

Proof. In [7] it is shown that existential model checking problem for an LTL formula α [5] can be reduced to checking for emptiness of the product P of the original model and the Büchi automaton with at most $|\alpha| \cdot 2^{|\alpha|}$ states. So, if the LTL formula α is existentially valid in M, then there exists a path in the product P that starts with an initial state and ends with a cycle in the strongly connected component of accepting states. This path can be chosen to be a loop with k bounded by $|M| \cdot |\alpha| \cdot 2^{|\alpha|}$ which is the size of the product P. If we project this path onto its first component, the original model, then we get a path of the length k that is a loop and in addition fulfils α. By Definition of the bounded semantics this also implies $M \models^{\exists}_{k} \alpha$.

Lemma 4. *Given are a model M, an* ELTLK *formula φ, and a path ρ. Then the following implication holds: $M, \rho \models \varphi$ implies that there exists $k \geq 0$ such that $M, \rho \models_k \varphi$.*

Proof. In [8] it has been shown that it is possible to reduce the CKL_n model checking problem to the LTL model checking problem. The reduction is based on Proposition 1 of [4], which states that each epistemic modality (K_c, E_Γ, C_Γ, and D_Γ) is expressible in the Logic of Local Propositions. The CKL_n is the LTL logic augmented by an indexed set of modal operators K_c with their diamonds $\overline{\mathrm{K}}_c$, one for each agent $i \in Ag$, and common knowledge operators C_Γ with their diamonds $\overline{\mathrm{C}}_\Gamma$, where $\Gamma \subseteq Ag$.

Now, note that our ELTLK language is an "epistemically existential" fragment of CKL_n augemnted by the diamonds for D_Γ and E_Γ, representing distributed knowledge in the group Γ, and "everyone in Γ knows". Thus, to prove that the ELTLK model checking problem can be reduced to the LTL model checking problem, it is enough to observe that $\overline{\mathrm{D}}_\Gamma \alpha = \bigwedge_{c \in \Gamma} \overline{\mathrm{K}}_i \alpha$ and $\overline{\mathrm{E}}_\Gamma \alpha = \bigvee_{i \in \Gamma} \overline{\mathrm{K}}_i \alpha$. Consequently, by using Lemma 3, we can conclude that $M, \rho \models \varphi$ implies $M, \rho \models_k \varphi$ for some $k > 0$.

The following theorem, states that, if we take all possible bounds into account, then the bounded and unbounded semantics are equivalent.

Theorem 1. *Given are a model M and an* ELTLK *formula φ. Then the following equivalence holds: $M \models^{\exists} \varphi$ iff there exists $k \geq 0$ such that $M \models^{\exists}_{k} \varphi$.*

Proof. The "left-to-right" implication follows directly from Lemma 4. The "right-to-left" implication follows directly from Lemma 1 and Lemma 2.

[5] An LTL formula α is existentially valid in the model M (in symbols $M \models^{\exists} \alpha$) iff there exists a path $\rho \in P_k(\iota)$ in M with $M, \rho \models \alpha$. Determining whether an LTL formula α is existentially valid in a given model is called an existential model checking problem.

By straightforward induction on the length of a ELTLK 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 $f_k : \text{ELTLK} \to \mathbb{N}$ gives a bound on the number of k-paths, which are sufficient to validate a given ELTLK formula.

In the definition of f_k we assume that each formula of ELTLK is preceded by the "path" quantifier E with the meaning "there exists a path in $P_k(\iota)$"; 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 ELTLK 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: $f_k(\mathrm{E}\varphi) = f_k(\varphi) + 1$, $f_k(p) = f_k(\neg p) = 0$, if $p \in \mathcal{PV}$, $f_k(\alpha \vee \beta) = max\{f_k(\alpha), f_k(\beta)\}$, $f_k(\alpha \wedge \psi) = f_k(\alpha) + f_k(\beta)$, $f_k(\mathrm{X}\alpha) = f_k(\alpha)$, $f_k(\alpha \mathrm{U} \beta) = k \cdot f_k(\alpha) + f_k(\beta)$, $f_k(\alpha \mathrm{R} \beta) = (k+1) \cdot f_k(\beta) + f_k(\alpha)$, $f_k(\overline{Y}\alpha) = f_k(\alpha) + 1$, for $\overline{Y} \in \{\overline{\mathrm{K}}_c, \overline{\mathrm{D}}_\Gamma, \overline{\mathrm{E}}_\Gamma\}$, $f_k(\overline{\mathrm{C}}_\Gamma \alpha) = f_k(\alpha) + k$. Note that the truth or falsity of an ELTL formula when interpreted on a path ρ in M depends only on this path. This is not the case for the knowledge modalities as these can express properties of other paths as well. The definition of the function f_k reflects the above observation.

3.2 Translation to propositional formulae

We now show how to reduce BMC for ELTLK to the propositional satisfiability problem. This reduction allows us to use efficient SAT solvers for model checking.

Given are a model $M = (G, \iota, \Pi, \{\sim_c\}_{c \in \mathcal{A}}, \mathcal{V})$, an ELTLK formula φ, and a bound $k \geq 0$. The problem of checking whether the M is a model for φ can be translated to the problem of checking the satisfiability of the following propositional formula:

$$[M, \varphi]_k := [M^{\varphi,\iota}]_k \ \wedge \ [\varphi]_{M,k} \tag{1}$$

The formula $[M^{\varphi,\iota}]_k$ constrains the $f_k(\varphi)$ symbolic k-paths to be valid k-paths of M, while 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 ELTLK 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 the encoding of the global states in the model M. Recall that the set of reachable global states G is a subset of the Cartesian product of the agents' local states (i.e., a subset of $\prod_{i=1}^{n} L_i$). We assume that $L_i \subseteq \{0,1\}^{nl_i}$ with $nl_i = \lceil log_2(|L_i|) \rceil$, and $\sum_{i=1}^{n} nl_i = m$ for some $m \in N$. So, each global state $g = (l_1, \ldots, l_n) = (g[1], \ldots, g[m])$ of M can be represented by a vector $w = (w[1], \ldots, w[m])$ (called *symbolic state*), where each $w[i] \in \mathcal{PV}$, for $i = 1, \ldots, m$, is a propositional variable (called state variable).

A finite sequence $(w_0, \ldots, w_k)$ of symbolic states is called a *symbolic k-path*. In general, we need to consider not just one, as for pure LTL, but a number of symbolic k-paths. This number depends on the formula φ under investigation, and it is given by the value $f_k(\varphi)$. The j-th symbolic k-path is denoted by $w_{0,j}, \ldots, w_{k,j}$, where $w_{i,j}$ is a symbolic state for $0 \leq j < f_k(\varphi)$, and $0 \leq i \leq k$. For two symbolic states w, w', we define the following auxiliary propositional formulae:

- $I_g(w)$ is a formula that encodes the global state g of the model M.
- $p(w)$ is a formula that encodes a set of states where the proposition $p \in \mathcal{PV}$ holds.

- $H_c(w,w')$ is a formula that encodes equivalence of local states of agent c.
- $\mathcal{R}(w,w')$ is a formula that encodes the transition relation of M.
- $L_{l,k}(j)$ is a formula that encodes a (k,l)-loop on the $j-$th symbolic $k-$path.

The propositional formula $[M^{\varphi,\iota}]_k$ is defined as follows:

$$[M^{\varphi,\iota}]_k := \bigwedge_{j=0}^{f_k(\varphi)-1} \bigwedge_{i=0}^{k-1} I_\iota(w_{0,j}) \wedge \mathcal{R}(w_{i,j}, w_{i+1,j}) \tag{2}$$

where $w_{i,j}$ are symbolic states for $i = 0,\ldots,k$ and $j = 0,\ldots,f_k(\varphi)-1$.

The next step is a translation of an ELTLK formula φ to a propositional formula

$$[\varphi]_{M,k} := [\varphi]_k^{[0,0]} \vee \bigvee_{l=0}^{k-1} (L_{l,k}(0) \wedge [\varphi]_{k,l}^{[0,0]}) \tag{3}$$

where $[\varphi]_k^{[m,n]}$ denotes the translation of φ at the symbolic state $w_{m,n}$ using the n-th symbolic k-path, $[\varphi]_{k,l}^{[m,n]}$ denotes the translation of φ at the symbolic state $w_{m,n}$ using the n-th symbolic k-path which is a (k,l)-loop.

Definition 4 (Translation of the ELTLK formulae). *Let φ be an* ELTLK *formula, $k \geq 0$ a bound, and $0 \leq m,l \leq k$. Moreover, let $succ(m) = m+1$ for $m < k$, and $succ(m) = l$ for $m = k$.*

- $[p]_k^{[m,n]} := p(w_{m,n})$,
- $[p]_{k,l}^{[m,n]} := p(w_{m,n})$,
- $[\neg p]_k^{[m,n]} := \neg p(w_{m,n})$,
- $[\neg p]_{k,l}^{[m,n]} := \neg p(w_{m,n})$,
- $[\alpha\wedge\beta]_k^{[m,n]} := [\alpha]_k^{[m,n]} \wedge [\beta]_k^{[m,n]}$,
- $[\alpha\wedge\beta]_{k,l}^{[m,n]} := [\alpha]_{k,l}^{[m,n]} \wedge [\beta]_{k,l}^{[m,n]}$,
- $[\alpha\vee\beta]_k^{[m,n]} := [\alpha]_k^{[m,n]} \vee [\beta]_k^{[m,n]}$,
- $[\alpha\vee\beta]_{k,l}^{[m,n]} := [\alpha]_{k,l}^{[m,n]} \vee [\beta]_{k,l}^{[m,n]}$,
- $[\mathrm{X}\alpha]_k^{[m,n]} :=$ *if* $m < k$ *then* $[\alpha]_k^{[m+1,n]}$ *else* $false$,
- $[\mathrm{X}\alpha]_{k,l}^{[m,n]} := [\alpha]_{k,l}^{[succ(m),n]}$,
- $[\alpha\mathrm{U}\beta]_k^{[m,n]} := \bigvee_{i=m}^{k}([\beta]_k^{[i,n]} \wedge \bigwedge_{j=m}^{i-1}[\alpha]_k^{[j,n]})$,
- $[\alpha\mathrm{U}\beta]_{k,l}^{[m,n]} := \bigvee_{i=m}^{k}([\beta]_{k,l}^{[i,n]} \wedge \bigwedge_{j=m}^{i-1}[\alpha]_{k,l}^{[j,n]})$,
- $[\alpha\mathrm{R}\beta]_k^{[m,n]} := \bigvee_{i=m}^{k}([\alpha]_k^{[i,n]} \wedge \bigwedge_{j=m}^{i}[\beta]_k^{[j,n]})$,
- $[\alpha\mathrm{R}\beta]_{k,l}^{[m,n]} := \bigvee_{i=m}^{k}([\alpha]_{k,l}^{[i,n]} \wedge \bigwedge_{j=m}^{i}[\beta]_{k,l}^{[j,n]}) \vee \bigwedge_{i=min(l,m)}^{k}[\beta]_{k,l}^{[i,n]}$,
- $[\overline{\mathrm{K}}_c\alpha]_k^{[m,n]} := \bigvee_{t=0}^{f_k(\varphi)-1}\bigvee_{j=0}^{k}([\alpha]_k^{[j,t]} \wedge H_c(w_{m,n}, w_{j,t}))$,
- $[\overline{\mathrm{K}}_c\alpha]_{k,l}^{[m,n]} := \bigvee_{t=0}^{f_k(\varphi)-1}\bigvee_{j=0}^{k}(\bigvee_{ll=0}^{k-1}([\alpha]_{k,ll}^{[j,t]} \wedge L_{ll,k}(t)) \wedge H_c(w_{m,n}, w_{j,t}))$,
- $[\overline{\mathrm{D}}_\Gamma\alpha]_k^{[m,n]} := \bigvee_{t=0}^{f_k(\varphi)-1}\bigvee_{j=0}^{k}([\alpha]_k^{[j,t]} \wedge \bigwedge_{c\in\Gamma} H_c(w_{m,n}, w_{j,t}))$,
- $[\overline{\mathrm{D}}_\Gamma\alpha]_{k,l}^{[m,n]} := \bigvee_{t=0}^{f_k(\varphi)-1}\bigvee_{j=0}^{k}(\bigvee_{ll=0}^{k-1}([\alpha]_{k,ll}^{[j,t]} \wedge L_{ll,k}(t)) \wedge \bigwedge_{c\in\Gamma} H_c(w_{m,n}, w_{j,t}))$,
- $[\overline{\mathrm{E}}_\Gamma\alpha]_k^{[m,n]} := \bigvee_{t=0}^{f_k(\varphi)-1}\bigvee_{j=0}^{k}([\alpha]_k^{[j,t]} \wedge \bigvee_{c\in\Gamma} H_c(w_{m,n}, w_{j,t}))$,
- $[\overline{\mathrm{E}}_\Gamma\alpha]_{k,l}^{[m,n]} := \bigvee_{t=0}^{f_k(\varphi)-1}\bigvee_{j=0}^{k}(\bigvee_{ll=0}^{k-1}([\alpha]_{k,ll}^{[j,t]} \wedge L_{ll,k}(t)) \wedge \bigvee_{c\in\Gamma} H_c(w_{m,n}, w_{j,t}))$,
- $[\overline{\mathrm{C}}_\Gamma\alpha]_k^{[m,n]} := [\bigvee_{i=1}^{k}(\overline{\mathrm{E}}_\Gamma)^i\alpha]_k^{[m,n]}$,
- $[\overline{\mathrm{C}}_\Gamma\alpha]_{k,l}^{[m,n]} := [\bigvee_{i=1}^{k}(\overline{\mathrm{E}}_\Gamma)^i\alpha]_{k,l}^{[m,n]}$.

Note that in Formula 3 we either do not put any assumption on the shape of all the $f_k(\varphi)$ symbolic $k-$paths, or we require that all the $f_k(\varphi)$ symbolic $k-$paths represent $(k,l)-$loops. Such a definition is fully consistent with the definition of the bounded semantics.

Now consider a definition of the formula $[\varphi]_{M,k}$ in which we require the set of $f_k(\varphi)$ symbolic $k-$paths to represent strictly either a set of $(k,l)-$loops, or a set of $k-$paths which are not $(k,l)-$loops for any $l < k$, i.e., the following definition:

$$[\varphi]_{M,k} := ([\varphi]_k^{[0,0]} \wedge \neg L_k) \vee \bigvee_{l=0}^{k-1} (L_{l,k}(0) \wedge [\varphi]_{k,l}^{[0,0]}) \tag{4}$$

with $L_k := \bigwedge_{j=0}^{f_k(\varphi)-1}(\bigvee_{l=0}^{k-1} L_{l,k}(j))$; note that Formula 4 strictly corresponds to the second conjunct of the formula that defines the general translation of the BMC-method for LTL (see Definition 9 of [1]).

We have observed that this additional "no loop" constraint embedded in the translation implemented according to Formula 4 does not help the SAT-solver MiniSat 2 to check the resulting formula more efficiently, than in the case of the translation implemented according to Formula 3. The corresponding experimental results are presented in the next section.

Lemma 5 and Lemma 6 show the correctness of the translation implemented according to Formula 3. The corresponding lemmas for the translation implemented according to Formula 4 can be formulated and proven in similar way.

Lemma 5. *Given are a model M, an* ELTLK *formula φ, and a bound $k \geq 0$. Let $A_k \subseteq P_k(\iota)$ be a set that contains $f_k(\varphi)$ initial $k-$paths of M. Moreover, let $ix : A_k \to \{0,\ldots,f_k(\varphi)-1\}$ be a one-to-one function that for each $\rho \in A_k$ assigns a natural number less than $f_k(\varphi)$. Then, for all the subformulas γ of φ, and each $\rho \in A_k$ the following implication holds: If $M, \rho[m] \models_k \gamma$, then $[M^{\varphi,\iota}]_k \wedge [\gamma]_k^{[m,ix(\rho)]}$ is satisfiable.*

Proof. Straightforward by induction on the length of φ.

Lemma 6. *Given are a model M, an* ELTLK *formula φ, and a bound $k \geq 0$. Let $A_k \subseteq P_k(\iota)$ be a set that contains $f_k(\varphi)$ initial $k-$paths of M such that they are (k,l)-loops. Moreover, let $ix : A_k \to \{0,\ldots,f_k(\varphi)-1\}$ be a one-to-one function that for each $\rho \in A_k$ assigns a natural number less than $f_k(\varphi)$. Then, for all the subformulas γ of φ, and each $\rho \in A_k$ the following implication holds: If $M, \rho[m] \models_{k,l} \gamma$, then $[M^{\varphi,\iota}]_k \wedge \big(\bigvee_{l=0}^{k-1}(L_{l,k}(ix(\rho)) \wedge [\gamma]_{k,l}^{[m,ix(\rho)]})\big)$ is satisfiable.*

Proof. Straightforward by induction on the length of φ.

The correctness of the SAT-based translation scheme for ELTLK is guaranted by the following theorem.

Theorem 2. *Let M be a model, and φ an* ELTLK *formula. Then, $M \models^{\exists} \varphi$ iff there exists $k \geq 0$ such that $[\varphi]_{M,k} \wedge [M^{\varphi,\iota}]_k$ is satisfiable.*

Proof. By Theorem 1 we have that $M \models^{\exists} \varphi$ iff there exists $k \geq 0$ such that $M \models_k^{\exists} \varphi$. Thus, to prove the theorem, it is enough to prove the following equivalence: $M \models_k^{\exists} \varphi$ iff $[\varphi]_{M,k} \wedge [M^{\varphi,\iota}]_k$ is satisfiable.
(**Left-to-right**). This implication follows directly from Lemma 5 and Lemma 6.
(**Right-to-left**). Let $[\varphi]_{M,k} \wedge [M^{\varphi,\iota}]_k$ be satisfiable. By the definition of the translation, the propositional formula $[\varphi]_{M,k}$ encodes all the sets of initial $k-$paths of size $f_k(\varphi)$, which satisfy the formula φ. By the definition of the unfolding of the transition relation,

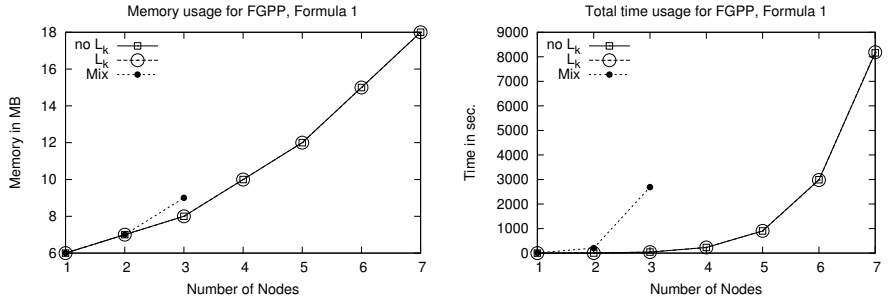
the propositional formula $[M^{\varphi,\iota}]_k$ encodes $f_k(\varphi)$ symbolic k-paths in P_k to be valid initial $k-$paths of M. Hence, there is a set of initial $k-$paths of size smaller or equal to $f_k(\varphi)$ in M which satisfies the formula φ. Thus, we can conclude that formula φ is existentially valid with bound k in M, i.e., $M \models^{\exists}_k \varphi$.

4 Experimental Results and Final Remarks

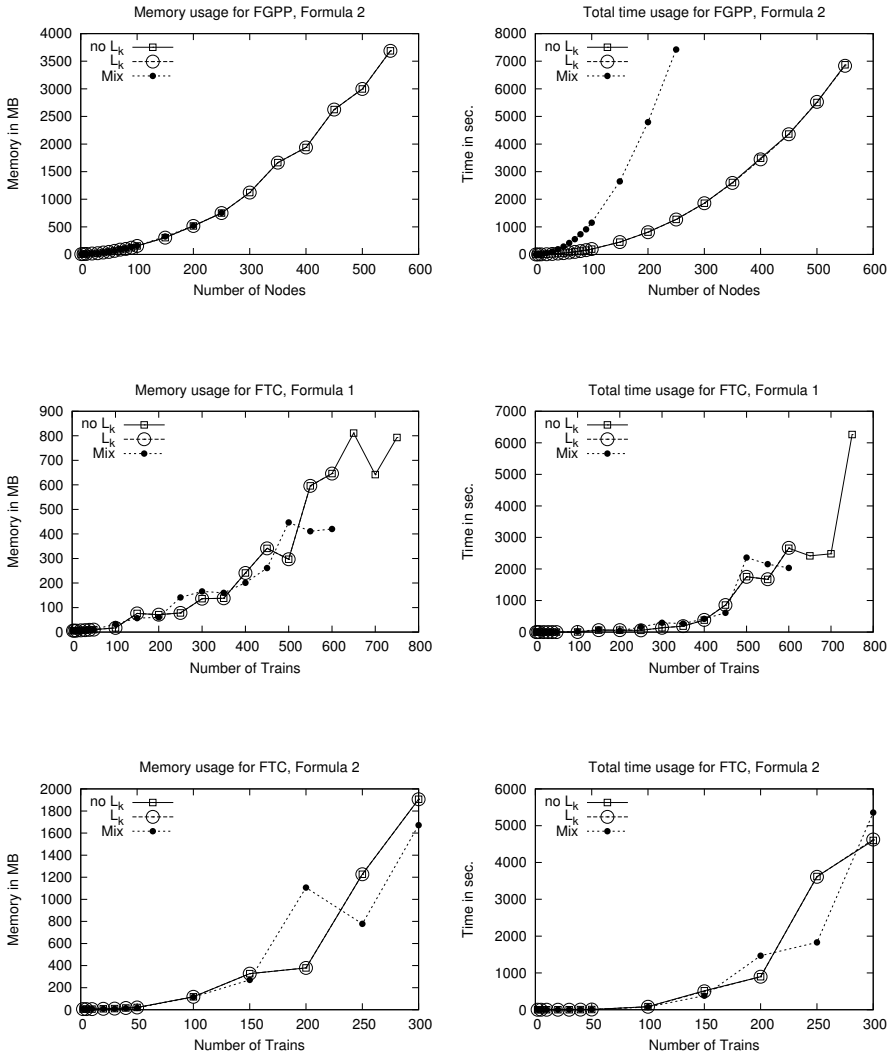
In this section we consider two scalable systems for which we give performance evaluation of our new BMC algorithms for the verification of several properties expressed in LTLK. Unfortunately, we cannot compare our experimental results to others, simply because, to the best of our knowledge, our SAT-based BMC methods for LTLK are the first one formally presented in the literature.

The specifications for the described benchmarks are given in the universal form, for which we verify the corresponding counterexample formula, i.e., the formula which is negated and interpreted existentially. Moreover, for every specification given, there exists a counterexample. All the benchmarks can be found at the webpage `http://ajd.czest.pl/~modelchecking/software.html`, together with an instruction how to repeat our experiments.

Faulty Generic Pipeline Paradigm (FGPP) (adapted from [15]) consists of Producer, Consumer, and a chain of n intermediate Nodes transmitting data, together with a chain of n Alarms enabled when some error occurs. We consider the following specifications: $\varphi_1 = \mathrm{G}(ProdSend \rightarrow \mathrm{K}_C\mathrm{K}_P ConsReady)$, and $\varphi_2 = \bigwedge_{i=1}^{n} \mathrm{G}(\mathrm{K}_P(Problem_i \rightarrow (\mathrm{F}Repair_i \vee \mathrm{G}Alarm_i Send)))$. The formula φ_1 states that if Producer produces a commodity, then Consumer knows that Producer does not know that Consumer has the commodity. The formula, φ_2 expresses that Producer knows that each time a problem occurs on a node, then either it is repaired or the alarm rings.



A faulty train controller system (FTC) (adapted from [9]) consists of a controller, and n trains (for $n \geq 2$), one of which is dysfunctional. We consider the following specifications: $\varphi_1 = \mathrm{G}(InTunnel_1 \rightarrow \mathrm{K}_{Train_1}(\bigwedge_{i=2}^{n} \neg InTunnel_i))$, and $\varphi_2 = \mathrm{G}(\mathrm{K}_{Train_1} \bigwedge_{i=1}^{n-1} \bigwedge_{j=i+1}^{n} \neg(InTunnel_i \wedge InTunnel_j))$. The formula φ_1 expresses that whenever a train is in the tunnel, it knows that the other train is not. The formula φ_2 represents that trains are aware of the fact that they have exclusive access to the tunnel.



Performance evaluation. We have implemented three different SAT-based BMC methods, but theoretically we have described only the one, which we call "no L_k". This is because it generates the least number of variables and clauses, and it performs equally efficient as the translation, which we call "L_k" (defined according to Formula 4).

Additionally to Definition 4 of the translation of LTLK formulae, we have defined and tested another translation, called "Mix", in which we have permitted to mix up between the translations $[\varphi]_k^{[m,n]}$ and $[\varphi]_{k,l}^{[m,n]}$. Namely, in both translations we have allowed in the definitions of epistemic operators to chose between the paths which are, or are not loops. For example in the case of $\overline{\mathrm{K}}_c$ the translation is the following:

$[\overline{\mathrm{K}}_c\alpha]_k^{[m,n]} = [\overline{\mathrm{K}}_c\alpha]_{k,l}^{[m,n]} := \bigvee_{t=0}^{f_k(\varphi)-1}\Big(\bigwedge_{ll=0}^{k-1}(\neg L_{ll,k}(t)) \wedge \bigvee_{j=0}^{k}([\alpha]_k^{[j,t]} \wedge H_c(w_{m,n},w_{j,t})) \vee \bigvee_{j=0}^{k}((\bigvee_{ll=0}^{k-1}([\alpha]_{k,ll}^{[j,t]} \wedge L_{ll,k}(t))) \wedge H_c(w_{m,n},w_{j,t}))\Big)$.

The comparison shows that the translation as presented in Definition 4 is clearly superior to the "mixing" one for all the tested systems and formulae (sometimes by several orders of magnitude). An evaluation is given by means of the running time and the memory used, and it is presented on the included line-charts.

For the tests we have used a computer with Intel Xeon 2 GHz processor, 4GB of RAM, and running Linux 2.6. We set the timeout to 9000 seconds, and memory limit to 4GB, and we used the state of the art SAT-solver MiniSat 2.

References

1. A. Biere, A. Cimatti, E. Clarke, O. Strichman, and Y. Zhu. Bounded model checking. *Advances in Computers*, vol. 58, pp. 118-149. Academic Press, 2003.
2. R. Bordini, M. Fisher, C. Pardavila, W. Visser, and M. Wooldridge. Model checking multi-agent programs with CASP. In *Proc. of CAV'03*, vol. 2725 of *LNCS*, pp. 110–113. Springer-Verlag, 2003.
3. E. Clarke, O. Grumberg, and D. Peled. *Model Checking*. MIT Press, 1999.
4. K. Engelhardt, R. van der Meyden, and Y. Moses. Knowledge and the logic of local propositions. In *Proc. of TARK'98*, pp. 29–41, 1998.
5. R. Fagin, J. Y. Halpern, Y. Moses, and M. Vardi. *Reasoning about Knowledge*. MIT Press, Cambridge, 1995.
6. P. Gammie and R. van der Meyden. MCK: Model checking the logic of knowledge. In *Proc. of CAV'04*, vol. 3114 of *LNCS*, pp. 479–483. Springer-Verlag, 2004.
7. P. Gastin and D. Oddoux. Fast LTL to Büchi automata translation. In *Proc. of CAV'01*, vol. 2102 of *LNCS*, pp. 53–65. Springer-Verlag, 2001.
8. W. van der Hoek and M. Wooldridge. Model checking knowledge and time. In *Proc. of SPIN'02*, vol. 2318 of *LNCS*, pp. 95–111. Springer-Verlag, 2002.
9. W. van der Hoek and M. Wooldridge. Cooperation, knowledge, and time: Alternating-time temporal epistemic logic and its applications. *Studia Logica*, 75(1):125–157, 2003.
10. A. Jones and A. Lomuscio. A BDD-based BMC approach for the verification of multi-agent systems. In *Proc. of CS&P'09*, vol. 1, pp. 253–264. Warsaw University, 2009.
11. M. Kacprzak, A. Lomuscio, and W. Penczek. From bounded to unbounded model checking for temporal epistemic logic. *Fundamenta Informaticae*, 63(2-3):221–240, 2004.
12. A. Lomuscio, W. Penczek, and H. Qu. Partial order reduction for model checking interleaved multi-agent systems. In *AAMAS, IFAAMAS Press.*, pp. 659–666, 2010.
13. R. van der Mayden and K. Su. Symbolic model checking the knowledge of the dining cryptographers. In *Proc. of CSFW-17*, pp. 280–291. IEEE Computer Society, June 2004.
14. R. van der Meyden and N. V. Shilov. Model checking knowledge and time in systems with perfect recall. In *Proc. of FSTTCS'99*, vol. 1738 of *LNCS*, pp. 432–445. Springer-Verlag, 1999.
15. D. Peled. All from one, one for all: On model checking using representatives. In *Proc. of CAV'93*, volume 697 of *LNCS*, pp. 409–423. Springer-Verlag, 1993.
16. W. Penczek and A. Lomuscio. Verifying epistemic properties of multi-agent systems via bounded model checking. *Fundamenta Informaticae*, 55(2):167–185, 2003.
17. F. Raimondi and A. Lomuscio. Automatic verification of multi-agent systems by model checking via OBDDs. *Journal of Applied Logic*, 5(2):235–251, 2007.
18. K. Su, A. Sattar, and X. Luo. Model checking temporal logics of knowledge via OBDDs. *The Computer Journal*, 50(4):403–420, 2007.