Verifying Epistemic Properties of Multi-agent Systems via Bounded Model Checking*

W. Penczek
Institute of Computer Science, PAS
01-237 Warsaw, ul. Ordona 21, Poland
email: penczek@ipipan.waw.pl
and
Podlasie Academy
Institute of Informatics, Siedlce, Poland

A. Lomuscio
Department of Computer Science
King's College London, London WC2R 2LS, United Kingdom
email: alessio@dcs.kcl.ac.uk

August 14, 2003

Abstract

We present a framework for verifying temporal and epistemic properties of multi-agent systems by means of bounded model checking. We use interpreted systems as underlying semantics. We give details of the proposed technique, and show how it can be applied to the "attacking generals problem", a typical example of coordination in multi-agent systems.

1 Introduction

The field of multi-agent systems (MAS) theories is traditionally concerned with the formal representation of the mental attitudes of autonomous entities, or *agents*, in a distributed system. For this task several modal logics have been developed in the past 20 years, the most studied being logics for knowledge, beliefs, desires, goals, and intentions.

These logics are seen as *specifications* of particular classes of MAS systems. Their aim is to offer a description of the *macroscopic* mental properties (such as knowledge, beliefs, etc.) that a MAS should exhibit in a specific class of scenarios. Sometimes, *interaction properties* are studied. For example, in an epistemic and doxastic model of agency it often makes sense to impose that knowledge is "true belief". This leads to a logic with two families of modalities, $\{K_i\}_{i\in A}, \{B_i\}_{i\in A}$, where B_i is a KD45-modality, K_i is an S5-modality, and the interaction axiom $K_i p \rightarrow B_i p$ expresses the intended interplay between the two informational operators. A considerable number of these formal studies are available in the literature and temporal extensions of these (i.e., modal combinations of CTL [5] or LTL [20] with the modalities for the mental attitudes) have appeared recently. The typical technical contribution of this line of work is to explore the metalogical properties of these logics, e.g., completeness, decidability, and computational complexity.

*Partly supported by the Polish State Committee for Scientific Research under grant 7T11C 00620, by the EU Framework V research project ALFEBIITE (IST-1999-10298), and by the Nuffield Foundation under grant NAL/00690/G. This paper extends with proofs and further technical details material appearing in [25], and appears in similar form in Fundamenta Informaticae, Volume 55, Number 2.

While these investigations are conceptually valuable, a deep computational analysis of the system is often missing. Typically, the semantics given for these logics is a plain Kripke models semantics which abstracts from how the *actual computation* proceeds in the MAS. Attempts have been made to solve this problem by giving a clear computational model on which to define informational modalities. The framework of *interpreted systems* [7] is one of such formalisms. There, the notions of actions, protocols, and transitions are given a prominent role, and epistemic modalities are defined, not on abstract possible worlds, but on the sets of runs that actually encode all the possible computations of the system. This has provided a formal basis for an in-depth analysis of the epistemic properties that arise in a MAS when the system enjoys particular properties such as synchronicity, asynchronicity, perfect recall, no learning, broadcasting, etc [8, 9, 22, 17].

The advantage of having a computationally grounded semantics such as the one of interpreted systems is that it allows for the possibility of exploring the issue of *verifying* MAS. In particular, interpreted systems are very promising because they are built upon standard CTL semantics, allowing for possible integration with model checking tools [4]. Indeed, some investigations along these lines have been carried out already. In particular, [11, 12] analyse respectively application of SPIN and MOCHA to model checking of LTL and ATL extended by epistemic modalities, whereas [21] studies the complexity of the model checking problem for essentially infinite state systems of knowledge and time.

Model checking provides for a promising set of techniques for hardware and software verification, but it suffers from what is known as the state explosion problem. Essentially, in this formalism checking that a property follows from a specification amounts to checking whether or not a modal formula is valid on a model representing all possible computations of the system. Encoding such a model is not problematic for small tailored examples, but it quickly becomes unfeasible as the number of states increases. Moreover, and perhaps most importantly, a full generation of the model is simply not possible if the specification actually generates an infinite computational model. One of the most promising solutions to these two problems is bounded model checking (BMC) [1, 3]. In this line of work, a methodology is developed to explore only the part of the model that is sufficient to validate the particular formula that needs to be checked. Then, the model checking problem over the part of the model is translated into a test of propositional satisfiability, for which refined tools already exist [24].

The aim of this paper is to report on recent progress on the application of bounded model checking to verifying not just temporal, but also epistemic properties of a MAS. Our approach is novel in the way it combines CTL with knowledge modalities in a fully automatic fashion.

The rest of the paper is organised as follows. Section 2 introduces interpreted system semantics. The logic **CTLK** is defined in Section 3, with its bounded semantics defined in Section 4. Section 5 describes the BMC algorithm for **CTLK**; its correctness is investigated in Section 6. In Section 7 we put these ideas to the test by evaluating its applicability on an example popular in the MAS literature: the "attacking generals problem". In the final section we point to future and related work.

2 Interpreted systems semantics

We assume familiarity with interpreted system semantics [7]. This can be succinctly defined as follows. Assume a set of agents $A = \{1, \ldots, n\}$, a set of local states L_i and possible actions Act_i for each agent $i \in A$, and a set L_e and Act_e of local states and actions for the environment. The set of global states for the system is defined as $G \subseteq L_1 \times \ldots \times L_n \times L_e$, where each element $(l_1, \ldots, l_n, l_e)$ of G represents a computational state for the whole system. Further assume a set of protocols $P_i : L_i \to 2^{Act_i}$, for $i = 1, \ldots, n$, representing the functioning behaviour of every agent, and a function $P_e : L_e \to 2^{Act_e}$ for the environment. Note that this defines a non-deterministic system. We can model the computation taking place in the system by means of a transition function $t : G \times Act \to G$, where $Act \subseteq Act_1 \times \ldots \times Act_n \times Act_E$ is the set of joint actions. Intuitively, given an initial state s_0, the sets of protocols, and the transition function, we can build a (possibly infinite) structure that represents all the possible computations of the system. Many representations can be given to this structure; since in this paper we are only concerned with temporal epistemic properties, we shall find the following to be a useful one.

Definition 1 *Given a set of agents $A = \{1, \ldots, n\}$ a temporal epistemic* model *(or simply a model) is a pair* $\mathrm{M} = (\mathcal{K}, \mathcal{V})$ *with* $\mathcal{K} = (W, T, \sim_1, \ldots, \sim_n, \iota)$*, where*

- *W is a finite[1] set of reachable global states for the system (henceforth called simply "states"),*
- *$T \subseteq W \times W$ is a total[2] binary (successor) relation on W,*
- *$\sim_i \subseteq W \times W$ $(i \in A)$ is an* epistemic accessibility relation *for each agent $i \in A$ defined by $w \sim_i w'$ iff $l_i(w') = l_i(w)$, where the function $l_i : W \to L_i$ returns the local state of agent i from a global state w. Obviously $\sim_i$ is an equivalence relation.*
- *$\iota \in W$ is the initial state,*
- *$\mathcal{V} : W \longrightarrow 2^{\mathcal{PV}_K}$ is a valuation function for a set of propositional variables $\mathcal{PV}_K$ such that* **true** $\in \mathcal{V}(s)$ *for all $s \in W$. $\mathcal{V}$ assigns to each state a set of propositional variables that are assumed to be true at that state.*

By $|\mathrm{M}|$ we denote the number of states of M, whereas $\mathbb{N} = \{0, 1, 2, \ldots\}$ indicates the set of natural numbers, and $\mathbb{N}_+ = \{1, 2, \ldots\}$ the set of positive natural numbers.

Epistemic relations. Let $\Gamma \subseteq A$. The union of Γ's accessibility relations is defined as $\sim^E_\Gamma = \bigcup_{i \in \Gamma} \sim_i$. By $\sim^C_\Gamma$ we denote the transitive closure of $\sim^E_\Gamma$, whereas $\sim^D_\Gamma = \bigcap_{i \in \Gamma} \sim_i$. The above relations are later used to give semantics to the "common knowledge", "everyone knows", and "distributed knowledge" modalities of the logic [7].

Computations paths. A *computation* in M is a possibly infinite sequence of states $\pi = (s_0, s_1, \ldots)$ such that $(s_i, s_{i+1}) \in T$ for each $i \in \mathbb{N}$. Specifically, we assume that $(s_i, s_{i+1}) \in T$ iff $s_{i+1} = t(s_i, act_i)$, i.e., s_{i+1} is the result of applying the transition function t to the global state s_i, and a joint action act_i. Each of the components of act_i is prescribed by the corresponding protocol P_j at $l_j(s_i)$, for $j \in A$. In the following we abstract from the transition function, the actions, and the protocols, and simply use the relation T, but it should be clear that this is uniquely determined by the interpreted system under consideration. In interpreted systems terminology a computation is a part of a run, sometimes referred to as finite trace. A k-computation is a computation of length k. For a computation $\pi = (s_0, s_1, \ldots)$, let $\pi(k) = s_k$, and $\pi_k = (s_0, \ldots, s_k)$, for each $k \in \mathbb{N}$. By $\Pi(s)$ we denote the set of all the infinite computations starting at s in M, whereas by $\Pi_k(s)$ the set of all the k-computations starting at s.

3 Computation Tree Logic of Knowledge (CTLK)

Interpreted systems are traditionally used to give a semantics to an epistemic language enriched with temporal connectives based on linear time [7]. Here we use CTL by Emerson and Clarke [6] as our basic temporal language and add an epistemic component to it. We call the resulting logic Computation Tree Logic of Knowledge (**CTLK**).

Definition 2 (Syntax of CTLK) *Let $\mathcal{PV}_K$ be a set of propositional variables containing the symbol* **true***. The set of* **CTLK** *formulas $\mathcal{FORM}$ is defined inductively as follows:*

[1] For simplicity in this paper we use finite models but the formalism presented here can be extended to infinite models with no difficulty by considering different sets of languages for the local states depending on the size of the bounded model under consideration.

[2] By "total" we mean that given any w there exists a w' such that wRw' (this condition is also called "seriality" in other areas of modal logic).

- *every member p of $\mathcal{PV}_K$ is a formula,*
- *if α and β are formulas, then so are $\neg\alpha$, $\alpha \wedge \beta$ and $\alpha \vee \beta$,*
- *if α is formula, then so are $\mathrm{EX}\alpha$, $\mathrm{EG}\alpha$ and $\mathrm{E}(\alpha \mathrm{U} \beta)$,*
- *if α is formula, then so is $\overline{\mathrm{K}}_i\alpha$, for $i \in A$,*
- *if α is formula, then so are $\overline{\mathrm{D}}_\Gamma\alpha$, $\overline{\mathrm{C}}_\Gamma\alpha$, and $\overline{\mathrm{E}}_\Gamma\alpha$, for $\Gamma \subseteq A$.*

The basic modalities are defined by derivation as follows: $\mathrm{F}\alpha \stackrel{def}{=} \mathbf{true}\mathrm{U}\alpha$, $\mathrm{A}(\alpha\mathrm{R}\beta) \stackrel{def}{=} \neg\mathrm{E}(\neg\alpha\mathrm{U}\neg\beta)$, $\mathrm{AX}\alpha \stackrel{def}{=} \neg\mathrm{EX}\neg\alpha$, $\mathrm{AG}\alpha \stackrel{def}{=} \neg\mathrm{EF}\neg\alpha$, $\mathrm{D}_\Gamma\alpha \stackrel{def}{=} \neg\overline{\mathrm{D}}_\Gamma\neg\alpha$, $C_\Gamma\alpha \stackrel{def}{=} \neg\overline{\mathrm{C}}_\Gamma\neg\alpha$, $\mathrm{E}_\Gamma\alpha \stackrel{def}{=} \neg\overline{\mathrm{E}}_\Gamma\neg\alpha$. Moreover, $\alpha \rightarrow \beta \stackrel{def}{=} \neg\alpha \vee \beta$. We omit the subscribe Γ of the epistemic modalities if $\Gamma = A$, i.e., Γ is the set of all the agents.

The logic **ECTLK** is the restriction of **CTLK** such that negation can be applied only to elements of $\mathcal{PV}_K$ — the definition of **ECTLK** is identical to Definition 2 except for $\neg p$ replacing $\neg\alpha$ in the second itemised paragraph.

The logic **ACTLK** is the restriction of **CTLK** such that its language is defined as $\{\neg\varphi \mid \varphi \in$ **ECTLK**$\}$. It is easy to see that **ACTLK** formulas can be written as follows: $\mathrm{AX}\alpha$, $\mathrm{A}(\alpha\mathrm{R}\beta)$, $\mathrm{AF}\alpha$, $\mathrm{K}_i\alpha$, $\mathrm{D}_\Gamma\alpha$, $C_\Gamma\alpha$, and $\mathrm{E}_\Gamma\alpha$.

Definition 3 (Satisfaction of CTLK) *Let M be a model, s be a state, π be a computation, and α, β formulas of* **CTLK**. *$\mathrm{M}, s \models \alpha$ denotes that α is true at the state s in the model M. M is omitted, if it is implicitly understood. The relation $\models$ is defined inductively as follows:*

$s \models p$	*iff*	$p \in \mathcal{V}(s)$, *for* $p \in \mathcal{PV}_K$,
$s \models \alpha \vee \beta$	*iff*	$s \models \alpha$ *or* $s \models \beta$,
$s \models \neg\alpha$	*iff*	$s \not\models \alpha$,
$s \models \alpha \wedge \beta$	*iff*	$s \models \alpha$ *and* $s \models \beta$,
$s \models \mathrm{EX}\alpha$	*iff*	$\exists \pi \in \Pi(s)\ \pi(1) \models \alpha$,
$s \models \mathrm{EG}\alpha$	*iff*	$\exists \pi \in \Pi(s)\ \forall_{m \geq 0}\ \pi(m) \models \alpha$,
$s \models \mathrm{E}(\alpha\mathrm{U}\beta)$	*iff*	$\exists \pi \in \Pi(s)\ (\exists_{m \geq 0}\ [\pi(m) \models \beta$ *and* $\forall_{j<m}\ \pi(j) \models \alpha])$,
$s \models \overline{\mathrm{K}}_i\alpha$	*iff*	$\exists s' \in W\ (s \sim_i s'$ *and* $s' \models \alpha)$,
$s \models \overline{\mathrm{D}}\alpha$	*iff*	$\exists s' \in W\ (s \sim_\Gamma^D s'$ *and* $s' \models \alpha)$,
$s \models \overline{\mathrm{E}}_\Gamma\alpha$	*iff*	$\exists s' \in W\ (s \sim_\Gamma^E s'$ *and* $s' \models \alpha)$,
$s \models \overline{\mathrm{C}}_\Gamma\alpha$	*iff*	$\exists s' \in W\ (s \sim_\Gamma^C s'$ *and* $s' \models \alpha)$.

Definition 4 (Validity) *A* **CTLK** *formula φ is valid on $\mathrm{M} = (\mathcal{K}, \mathcal{V})$ (denoted $\mathrm{M} \models \varphi$) iff $\mathrm{M}, \iota \models \varphi$, i.e., φ is true at the initial state of the model M.*

4 Bounded Semantics for CTLK

In this section we give a *bounded semantics* for **CTLK** in order to define the *bounded model checking problem* for **ECTLK**, and to translate it subsequently into a satisfiability problem. This formalism is an extension of the one presented in [26].

Definition 5 (k−model) *Let $\mathrm{M} = (\mathcal{K}, \mathcal{V})$ be a model and $k \in \mathbb{N}_+$. A structure $\mathrm{M}_k = ((W, P_k, \sim_1, \ldots, \sim_n, \iota), \mathcal{V})$ is a k−model for M, where P_k is the set of all the k-computations of M, i.e., $P_k = \bigcup_{s \in W} \Pi_k(s)$.*

Satisfaction for the temporal operators in the bounded case depends on whether or not the computation π defines a loop, i.e., whether $loop(\pi) \neq \emptyset$, where *loop* is defined below.

Definition 6 *Let $\mathrm{M}_k = ((W, P_k, \sim_1, \ldots, \sim_n, \iota), \mathcal{V})$ be a k−model for a model M, and $\pi \in P_k$ a k−computation. The function $loop : P_k \rightarrow 2^{\mathbb{N}}$ is defined as: $loop(\pi) = \{l \mid 0 \leq l \leq k$ and $(\pi(k), \pi(l)) \in T\}$.*

The main reason for reformulating the semantics of the modalities in the following definition in terms of elements of k-computations rather than elements of W or Π is to restrict the semantics to a part of the model. Note that the interpretation of the temporal modalities on bounded semantics is different from the one of Definition 3.

Definition 7 (Bounded semantics) *Let* M_k *be a* $k-$*model and* α, β *be* **ECTLK** *formulas.* $\mathrm{M}_k, s \models \alpha$ *denotes that* α *is true at the state* s *of* M_k. M_k *is omitted if it is clear from the context. The relation* $\models$ *is defined inductively as follows:*

$s \models p$ *iff* $p \in \mathcal{V}(s)$, *for* $p \in \mathcal{PV}_K$,
$s \models \alpha \vee \beta$ *iff* $s \models \alpha$ *or* $s \models \beta$,
$s \models \neg p$ *iff* $p \notin \mathcal{V}(s)$,
$s \models \alpha \wedge \beta$ *iff* $s \models \alpha$ *and* $s \models \beta$,
$s \models \mathrm{EX}\alpha$ *iff* $\exists \pi \in P_k \left(\pi(0) = s \text{ and } \pi(1) \models \alpha\right)$,
$s \models \mathrm{EG}\alpha$ *iff* $\exists \pi \in P_k \left(\pi(0) = s \text{ and } \forall_{0 \leq j \leq k}\ \pi(j) \models \alpha\right)$ *and* $loop(\pi) \neq \emptyset$,
$s \models \mathrm{E}(\alpha \mathrm{U} \beta)$ *iff* $\exists \pi \in P_k \left(\pi(0) = s \text{ and } \exists_{0 \leq j \leq k}\left(\pi(j) \models \beta \text{ and } \forall_{0 \leq i < j} \pi(i) \models \alpha\right)\right)$,
$s \models \overline{\mathrm{K}}_i\alpha$ *iff* $\exists \pi \in P_k \left(\pi(0) = \iota \text{ and } \exists_{0 \leq j \leq k}\left(\pi(j) \models \alpha \text{ and } s \sim_i \pi(j)\right)\right)$,
$s \models \overline{\mathrm{D}}_\Gamma\alpha$ *iff* $\exists \pi \in P_k \left(\pi(0) = \iota \text{ and } \exists_{0 \leq j \leq k}\left(\pi(j) \models \alpha \text{ and } s \sim_\Gamma^D \pi(j)\right)\right)$,
$s \models \overline{\mathrm{E}}_\Gamma\alpha$ *iff* $\exists \pi \in P_k \left(\pi(0) = \iota \text{ and } \exists_{0 \leq j \leq k}\left(\pi(j) \models \alpha \text{ and } s \sim_\Gamma^E \pi(j)\right)\right)$,
$s \models \overline{\mathrm{C}}_\Gamma\alpha$ *iff* $\exists \pi \in P_k \left(\pi(0) = \iota \text{ and } \exists_{0 \leq j \leq k}\left(\pi(j) \models \alpha \text{ and } s \sim_\Gamma^C \pi(j)\right)\right)$.

The above extends to knowledge modalities the bounded semantics of [26]. Note that given Definition 1, the epistemic relations used above are constructed on the basis of the *internal structure of the global states* of the system (i.e., they are defined on the basis of the local states of the agents), and not by means of an ad-hoc construction by the modeller. Note also that while the conditions for the temporal components require the states to be reachable from the state in consideration, this is not the case for the epistemic conditions, where we consider whether or not there is *a* computation from the initial state that results in a state that is epistemically indistinguishable for agent i from the global state under consideration. This guarantees reachability of such a state and corresponds to the usual interpretation of epistemic modalities according to which an agent considers as epistemically possible global states of computation resulting from different traces, as long as its local state is the same.

Definition 8 (Validity for Bounded Semantics) *An* **ECTLK** *formula* φ *is valid on a k-model* M_k *(denoted* $M \models_k \varphi$*) iff* $\mathrm{M}_k, \iota \models \varphi$.

Next, we describe how the model checking problem ($M \models \varphi$) can be reduced to the bounded model checking problem ($M \models_k \varphi$).

Lemma 1 *Let* M *be a model,* s *be a state of* M*, and* φ *be an* **ECTLK** *formula. Then, the following two conditions hold:*
a) $\mathrm{M}_k, s \models \varphi$ *implies* $M_l, s \models \varphi$, *for* $l \geq k$,
b) $\mathrm{M}_k, s \models \varphi$ *implies* $M, s \models \varphi$.

Proof 4.1 *Straightforward by induction on the length of* φ.

Lemma 2 *Let* M *be a model,* φ *be an* **ECTLK** *formula,* s *be a state of* M*, and* $k = |M|$*. If* $M, s \models \varphi$*, then* $\mathrm{M}_k, s \models \varphi$.

Proof 4.2 *By induction on the length of* φ*. The lemma follows directly for the propositional variables and their negations.*

Next, assume that the hypothesis holds for all the proper sub-formulas of φ*. If* φ *is equal to either* $\alpha \wedge \beta$ *or* $\alpha \vee \beta$*, then it is easy to check that the lemma holds. Consider* φ *to be of the following forms:*

- $\varphi = \mathrm{EX}\alpha \mid \mathrm{EG}\alpha \mid \mathrm{E}(\alpha \mathrm{U} \beta)$*. By induction hypothesis — see [26] page 139.*

- $\varphi = \overline{\mathrm{K}}_i\alpha$. *By definition, there is a state* s' *in* M *such that* $l_i(s) = l_i(s')$ *and* $\mathrm{M}, s' \models \alpha$. *By the inductive assumption, we have that* $\mathrm{M}_k, s' \models \alpha$. *Since* s' *is reachable, it is reachable from* ι *in at most* k *steps as* $k = |M|$. *Thus, there is a* $k-$*computation* $\pi \in P_k$ *such that* $\pi(1) = \iota$ *and* $\pi(i) = s'$ *for some* $i \leq k$. *So, we have* $\mathrm{M}_k, s \models \overline{\mathrm{K}}_i\alpha$.

- $\varphi = \overline{\mathrm{E}}_\Gamma\alpha$. $\varphi = \overline{\mathrm{E}}_\Gamma\alpha = \bigvee_{i\in A} \overline{\mathrm{K}}_i\alpha$. *Therefore the result follows from the case above for a specific* $i \in A$, *and the basic case for the boolean connectives.*

- $\varphi = \overline{\mathrm{D}}_\Gamma\alpha$. *Straightforward by definition from the case* $\varphi = \overline{\mathrm{K}}_i\alpha$.

- $\varphi = \overline{\mathrm{C}}_\Gamma\alpha$. *Note that* $M, s \models \overline{\mathrm{C}}_\Gamma\alpha$ *iff* $M, s \models \bigvee_{i\leq|M|}(\overline{\mathrm{E}}_\Gamma)^i\alpha$. *So, by induction and the former case, we have* $\mathrm{M}_k, s \models \overline{\mathrm{C}}_\Gamma\alpha$.

In this setting we can prove that in some circumstances satisfiability in the $|M|$-bounded semantics is equivalent to the unbounded one.

Theorem 1 *Let* $\mathrm{M} = ((W, T, \sim_1, \ldots, \sim_n, \iota), \mathcal{V})$ *be a model,* φ *be an* **ECTLK** *formula and* $k = |M|$. *Then,* $M \models \varphi$ *iff* $M \models_k \varphi$.

Proof 4.3 *Straightforward from Lemma 1 and Lemma 2 above.*

Given that we reasoned on a bounded model of size $|M|$ there is nothing surprising about the results above. The rationale behind the method is that for particular examples checking satisfiability of a formula can be done on a small fragment of the model.

5 The BMC algorithm for ECTLK

In this section we present a method of BMC for **ECTLK**. This is an extension of the method appearing in [26]. We assume the following two definitions.

Definition 9 *Let* $\mathrm{M}_k = ((W, P_k, \sim_1, \ldots, \sim_n, \iota), \mathcal{V})$ *be a k-model of* M. *We say that a structure* $\mathrm{M}'_k = ((W', P'_k, \sim'_1, \ldots, \sim'_n, \iota), \mathcal{V}')$ *is a* submodel *of* M_k *if* $P'_k \subseteq P_k$, $States(P'_k) \subseteq W' \subseteq W$, $\sim'_i = \sim_i \cap (W' \times W')$, *for* $i \in A$, *and* $\mathcal{V}' = \mathcal{V}|_{W'}$, *where* $States(P'_k)$ *defines the set of states reached in all computations in* P'_k, *and* $\mathcal{V}|_{W'}$ *denotes the restriction of the interpretation function* $\mathcal{V}$ *to* W', *a subset of* W *(upon which* $\mathcal{V}$ *is defined).*

For technical reasons we allow for having states in W', which may not be reached in P'_k, but obviously all the states of W' are reachable in M_k as $W' \subseteq W$.

The bounded semantics of **ECTLK** over submodels M'_k can still be defined as for M_k (see Def. 7). Our present aim is give a bound for the number of k-computations in M'_k such that the validity of φ in M_k is equivalent to the validity of φ in M'_k.

Definition 10 *Define a function* $f_k : \mathcal{FORM} \rightarrow \mathbb{N}$ *as follows:*

- $f_k(p) = f_k(\neg p) = 0$, *where* $p \in \mathcal{PV}_K$,
- $f_k(\alpha \vee \beta) = max\{f_k(\alpha), f_k(\beta)\}$,
- $f_k(\alpha \wedge \beta) = f_k(\alpha) + f_k(\beta)$,
- $f_k(Y\alpha) = f_k(\alpha) + 1$, *for* $Y \in \{\mathrm{EX}, \overline{\mathrm{K}}_i, \overline{\mathrm{D}}_\Gamma, \overline{\mathrm{E}}_\Gamma\}$,
- $f_k(\overline{\mathrm{C}}_\Gamma\alpha) = f_k(\alpha) + k$,
- $f_k(\mathrm{EG}\alpha) = (k + 1) \cdot f_k(\alpha) + 1$,
- $f_k(\mathrm{E}(\alpha\mathrm{U}\beta)) = k \cdot f_k(\alpha) + f_k(\beta) + 1$.

The function f_k determines the number of k-computations of a submodel M'_k sufficient for checking an **ECTLK** formula. Here we take this bound as given, but we provide a proof of the adequacy of this in the next section.

The main idea is that we can check φ over M_k by checking the satisfiability of a propositional formula $[M,\varphi]_k = [M^{\varphi,\iota}]_k \wedge [\varphi]_{\mathrm{M}_k}$, where the first conjunct represents (part of) the model under consideration and the second a number of constraints that must be satisfied on M_k for φ to be satisfied. Once this translation is defined, checking satisfiability of an **ECTLK** formula can be done by means of a SAT-checker. Although from a theoretical point of view the complexity of this operation is no easier, in practice the efficiency of modern SAT-checkers makes the process worthwhile in many instances. In this process, an important decision to take is the size k of the truncation. We do not discuss this issue in this paper, but we do point out the fact that there are heuristics that can be developed for particular classes of examples.

A trivial mechanism, for instance, would be to start with $k := 1$, test satisfiability for the translation, and increase k by one either until $[M^{\varphi,\iota}]_k \wedge [\varphi]_{\mathrm{M}_k}$ becomes satisfiable or k reaches $|M|$.

Definition 11 *BMC algorithm for* **ECTLK***:*

- *Let $\varphi = \neg\psi$ (where ψ is an* **ACTLK** *formula).*
- *Iterate for $k := 1$ to $|M|$.*
- *Select the $k-$model* M_k.
- *Select the submodels* M'_k *of* M_k *with* $|P'_k| \leq f_k(\varphi)$.
- *Translate the transition relation of the $k-$computations of all of the submodels* M'_k *into a propositional formula* $[M^{\varphi,\iota}]_k$.
- *Translate φ over all* M'_k *into a propositional formula* $[\varphi]_{\mathrm{M}_k}$.
- *Check the satisfiability of* $[\mathrm{M},\varphi]_k := [\mathrm{M}^{\varphi,\iota}]_k \wedge [\varphi]_{\mathrm{M}_k}$.

We now give details of this translation. We begin with the encoding of the transitions in the interpreted system under consideration. Recall that the set of possible global states $G = \times_{i=1}^{n} L_i$ is the Cartesian product of the set of local states. We assume $L_i \subseteq \{0,1\}^{n_i}$, where $n_i = \lceil log_2(|L_i|) \rceil$, and let $n_1 + \ldots + n_n = m$ for some m. Moreover, let I_i be a set of the indexes of the bits of the local states of each agent i in the global states, i.e., $I_1 = \{1,\ldots,n_1\},\ldots,I_n = \{m-n_n+1,\ldots,m\}$. So, each global state $s = (l_1,\ldots,l_n) = (s[1],\ldots,s[m])$ can be represented by $w = (w[1],\ldots,w[m])$ (which we shall call a *global state variable*), where each $w[i]$ for $i = 1,\ldots,m$ is a propositional variable. (Notice that we distinguish between global states being sequences of binary digits and their representations in terms of propositional variables $w[i]$). A finite sequence $(w_0,\ldots,w_k)$ of global state variables is called a *symbolic $k-$path*. In general we shall need to consider not just one but a number of symbolic $k-$paths. This number depends on the formula φ under investigation, and it is returned as the value $f_k(\varphi)$ of the function f_k. We refer to [26] for more details. To construct $[\mathrm{M},\varphi]_k$, we first define a propositional formula $[\mathrm{M}^{\varphi,\iota}]_k$ that constrains the $f_k(\varphi)$ symbolic k-paths to be valid k-computations of M_k. For $1 \leq j \leq f_k(\varphi)$, the j-th symbolic $k-$computation is denoted as $w_{0,j},\ldots,w_{k,j}$, where $w_{i,j}$ for $i \in \{0,\ldots,k\}$ are global state variables.

Let $\mathcal{PV}$ be a set of propositional variables, $\mathcal{FORM}$ be a set of propositional formulas over $\mathcal{PV}$, and let $lit : \{0,1\} \times \mathcal{PV} \to \mathcal{FORM}$ be a function defined as follows: $lit(0,p) = \neg p$ and $lit(1,p) = p$. Furthermore, let w, v be global state variables. We define the following propositional formulas:

- $I_s(w) := \bigwedge_{i=1}^{m} lit(s[i], w[i])$.

  This formula encodes the state s of the model, i.e., $s[i] = 1$ is encoded by $w[i]$, and $s[i] = 0$ is encoded by $\neg w[i]$.

- $p(w)$ is a formula over $w[1],\ldots,w[m]$, which is true for a valuation $(s_1,\ldots,s_m)\in\{0,1\}^m$ of $(w[1],\ldots,w[m])$ iff $p\in\mathcal{V}((s_1,\ldots,s_m))$, where $p\in\mathcal{PV}_K$.

  This formula encodes a proposition p of **ECTLK**.

- $H(w,v) := \bigwedge_{i=1}^{m} w[i]\Leftrightarrow v[i]$.

  This formula represents logical equivalence between global state encodings, representing the fact that they represent the same state.

- $H_l(w,v) := \bigwedge_{i\in I_l} w[i]\Leftrightarrow v[i]$.

  This formula represents logical equivalence between l-local state encodings, representing the fact that they represent the same local state, i.e., the local state in the two states is the same.

- $T(w,v)$ is a formula over the propositions $w[1],\ldots,w[m]$, $v[1],\ldots,v[m]$, which is true for a valuation $(s_1,\ldots,s_m)$ of $(w[1],\ldots,w[m])$ and a valuation $(s'_1,\ldots,s'_m)$ of $(v[1],\ldots,v[m])$ iff $((s_1,\ldots,s_m),(s'_1,\ldots,s'_m))\in T$.

- $L_{k,j}(l) := T(w_{k,j},w_{l,j})$,

  This formula encodes a backward loop connecting the k-th state to the l-th state in the symbolic $k-$computation j, for $0\le l\le k$.

The propositional formula $[\mathrm{M}^{\varphi,\iota}]_k$, representing the transitions in the k-model, is given by the following definition.

Definition 12 (Unfolding of Transition Relation) *Let* $\mathrm{M}_k = ((W,P_k,\sim_1,\ldots,\sim_n,\iota),\mathcal{V})$ *be the* $k-$*model of* M, $\iota\in W$, *and* φ *be an* ECTL *formula. The propositional formula* $[\mathrm{M}^{\varphi,\iota}]_k$ *is defined as follows:*

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

where $w_{0,0}$, *and* $w_{i,j}$ *for* $0\le i\le k$ *and* $1\le j\le f_k(\varphi)$ *are global state variables.* $[\mathrm{M}^{\varphi,\iota}]_k$ *encodes the initial state* ι *by* $w_{0,0}$ *and constrains the* $f_k(\varphi)$ *symbolic* k*-paths to be valid* k*-computations in* M_k.

The next step of the algorithm consists in translating an **ECTLK** formula φ into a propositional formula.

Definition 13 (Translation of ECTLK formulas) *Let a model* M_k *with initial state* ι*, and an* **ECTLK** *formula* φ *be given. We inductively define the translation of* φ *at state* $w_{m,n}$ *into the propositional formula* $[\varphi]_k^{[m,n]}$ *as follows:*

$$
\begin{array}{lcl}
[p]_k^{[m,n]} & := & p(w_{m,n}),\\
[\neg p]_k^{[m,n]} & := & \neg p(w_{m,n}),\\
[\alpha\wedge\beta]_k^{[m,n]} & := & [\alpha]_k^{[m,n]} \wedge [\beta]_k^{[m,n]},\\
[\alpha\vee\beta]_k^{[m,n]} & := & [\alpha]_k^{[m,n]} \vee [\beta]_k^{[m,n]},\\
[\mathrm{EX}\alpha]_k^{[m,n]} & := & \bigvee_{1\le i\le f_k(\varphi)} \Big(H(w_{m,n},w_{0,i}) \;\wedge\; [\alpha]_k^{[1,i]}\Big),\\
[\mathrm{EG}\alpha]_k^{[m,n]} & := & \bigvee_{1\le i\le f_k(\varphi)} \Big(H(w_{m,n},w_{0,i}) \;\wedge\; \bigvee_{l=0}^{k} L_{k,i}(l) \;\wedge\; \bigwedge_{j=0}^{k}[\alpha]_k^{[j,i]}\Big),\\
[\mathrm{E}(\alpha\mathrm{U}\beta)]_k^{[m,n]} & := & \bigvee_{1\le i\le f_k(\varphi)} \Big(H(w_{m,n},w_{0,i}) \;\wedge\; \bigvee_{j=0}^{k}\big([\beta]_k^{[j,i]} \;\wedge\; \bigwedge_{t=0}^{j-1}[\alpha]_k^{[t,i]}\big)\Big),\\
[\overline{\mathrm{K}}_l\alpha]_k^{[m,n]} & := & \bigvee_{1\le i\le f_k(\varphi)} \Big(I_\iota(w_{0,i}) \;\wedge\; \bigvee_{j=0}^{k}\big([\alpha]_k^{[j,i]} \;\wedge\; H_l(w_{m,n},w_{j,i})\big)\Big),\\
[\overline{\mathrm{D}}_\Gamma\alpha]_k^{[m,n]} & := & \bigvee_{1\le i\le f_k(\varphi)} \Big(I_\iota(w_{0,i}) \;\wedge\; \bigvee_{j=0}^{k}\big([\alpha]_k^{[j,i]} \;\wedge\; \bigwedge_{l\in\Gamma} H_l(w_{m,n},w_{j,i})\big)\Big),\\
[\overline{\mathrm{E}}_\Gamma\alpha]_k^{[m,n]} & := & \bigvee_{1\le i\le f_k(\varphi)} \Big(I_\iota(w_{0,i}) \;\wedge\; \bigvee_{j=0}^{k}\big([\alpha]_k^{[j,i]} \;\wedge\; \bigvee_{l\in\Gamma} H_l(w_{m,n},w_{j,i})\big)\Big),\\
[\overline{\mathrm{C}}_\Gamma\alpha]_k^{[m,n]} & := & [\bigvee_{1\le i\le k}(\overline{\mathrm{E}}_\Gamma)^i\alpha]_k^{[m,n]}.
\end{array}
$$

The meaning of the translations above can be intuitively reconstructed from the definition of propositional formulas presented earlier. For example, the formula $[\mathrm{EX}\alpha]_k^{[m,n]}$ expresses the condition that there exists a sub-path starting from $w_{m,n}$ in which the first point $w_{0,i}$ in this computation satisfies α. For $[\overline{\mathrm{K}}_l\alpha]_k^{[m,n]}$ we insist on the existence of a point $w_{j,i}$ which has the same local state for agent l, and that it is accessible from the initial state by some computation. The other cases are variations of these.

Given the translations above, we can now check φ over M_k by checking the satisfiability of the propositional formula $[M^{\varphi,\iota}]_k \wedge [\varphi]_{\mathrm{M}_k}$, where $[\varphi]_{\mathrm{M}_k} = [\varphi]_k^{[0,0]}$. The translation presented above is shown to be correct and complete in the next section.

6 Correctness of the translation

In this section we prove the correctness of the translation of the model checking problem into the SAT-problem as defined by Definition 12.

Lemma 3 $\mathrm{M}_k, s \models \varphi$ *iff there is a submodel* M'_k *of* M_k *with* $|P'_k| \leq f_k(\varphi)$ *such that* $\mathrm{M}'_k, s \models \varphi$.

Proof 6.1 (=>) *By structural induction on* φ. *The lemma follows directly for the propositional variables and their negations.*

Assume that the hypothesis holds for all the proper subformulas of φ.

- $\varphi = \alpha \vee \beta \mid \alpha \wedge \beta$. *Straightforward.*
- $\varphi = \mathrm{EX}\alpha \mid \mathrm{EG}\alpha \mid \mathrm{E}(\alpha \mathrm{U} \beta)$. *By induction hypothesis — see [26] page 143.*
- *Let* $\varphi = \overline{\mathrm{K}}_i\alpha$. *If* $\mathrm{M}_k, s \models \overline{\mathrm{K}}_i\alpha$, *then by definition:* $(\exists\pi \in P_k)\big(\pi(0) = \iota$ *and* $\exists_{0\leq j\leq k}(s \sim_i \pi(j))$ *and* $\pi(j) \models \alpha)\big)$. *By the inductive assumption there is a submodel* $\mathrm{M}'_k = ((W', P'_k, \sim'_1, \ldots, \sim'_n, \iota), \mathcal{V}')$ *of* M_k *such that* $|P'_k| \leq f_k(\alpha)$ *and* $\mathrm{M}'_k, \pi(j) \models \alpha$.

  Consider a submodel $\mathrm{M}''_k = ((W'', P''_k, \sim''_1, \ldots, \sim''_n, \iota), \mathcal{V}'')$ *of* M_k, *where* $P''_k = P'_k \cup \{\pi\}$ *and* $W'' = States(P''_k) \cup \{s\}$. *Since* π *belongs to* P''_k, *by the construction of* M''_k *and the definition of the bounded semantics, we have that* $\mathrm{M}''_k, s \models \overline{\mathrm{K}}_i\alpha$ *and* $|P''_k| \leq f_k(\varphi) = f_k(\alpha) + 1$.
- $\varphi = \overline{\mathrm{E}}_\Gamma\alpha \mid \overline{\mathrm{D}}_\Gamma\alpha$. *This case can be proven similarly.*
- *Let* $\varphi = \overline{\mathrm{C}}_\Gamma\alpha$. *The proof follows by induction using two cases: disjunction and* $\varphi = \overline{\mathrm{E}}_\Gamma\alpha$. *Below, we only prove that* $f_k(\overline{\mathrm{C}}_\Gamma\alpha) = f_k(\alpha) + k$ *is a sufficient number of paths in a submodel* M'_k *validating* φ. *The actual construction of* M'_k *can be given similarly to the case* $\varphi = \overline{\mathrm{K}}_i\alpha$. *Notice that* $\overline{\mathrm{C}}_\Gamma\alpha = \bigvee_{1\leq i\leq k}(\overline{\mathrm{E}}_\Gamma)^i\alpha$. *So, as shown in [26] for the formulas in the form of disjunction we have* $f_k(\varphi) = max_{1\leq i\leq k} f_k((\overline{\mathrm{E}}_\Gamma)^i\alpha)$. *Consider* $f_k((\overline{\mathrm{E}}_\Gamma)^i\alpha)$. *We know that* $f_k((\overline{\mathrm{E}}_\Gamma)\alpha) = f_k(\alpha) + 1$. *Therefore, by induction we have that* $f_k((\overline{\mathrm{E}}_\Gamma)^i\alpha) = f_k(\alpha) + i$. *This implies that* $f_k(\varphi) = max_{1\leq i\leq k}(f_k(\alpha) + i) = f_k(\alpha) + k$.

(<=) *The proof is straightforward.*

From Lemma 3 we can now derive the following.

Corollary 1 $\mathrm{M} \models_k \varphi$ *iff there is a submodel* M'_k *of* M_k *with* $|P'_k| \leq f_k(\varphi)$ *such that* $\mathrm{M}'_k, \iota \models \varphi$.

Proof 6.2 *It follows from Definition 8, and Lemma 3, by using* $s = \iota$.

Note that for what concerns Corollary 1, in the submodel M'_k all the states W' can be reached in computations of P'_k.

Lemma 4 *For each state* s *of* M, *the following condition holds:* $[M^{\varphi,s}]_k \wedge [\varphi]_{\mathrm{M}_k}$ *is satisfiable iff there is a submodel* M'_k *of* M_k *with* $|P'_k| \leq f_k(\varphi)$ *such that* $\mathrm{M}'_k, s \models \varphi$.

Proof 6.3 (=>) *Let* $[\mathrm{M}^{\varphi,s}]_k \wedge [\varphi]_{\mathrm{M}_k}$ *be satisfiable. By the definition of the translation, the propositional formula* $[\varphi]_{\mathrm{M}_k}$ *encodes all the sets of* $k-$*computations of size* $f_k(\varphi)$ *which satisfy the formula* φ*. By the definition of the unfolding of the transition relation, the propositional formula* $[M^{\varphi,s}]_k$ *encodes* $f_k(\varphi)$ *symbolic* k*-paths to be valid* $k-$*computations of* M_k*. Hence, there is a set of* $k-$*computations in* M_k*, which satisfies the formula* φ *of size smaller or equal to* $f_k(\varphi)$*. Thus, we conclude that there is a submodel* M'_k *of* M_k *with* $|P'_k| \leq f_k(\varphi)$ *and* $\mathrm{M}'_k, s \models \varphi$*. The actual definition of* M'_k *can be reconstructed from Definition 13 and Definition 12.*

(<=) *The proof is by induction on the length of* φ*. The lemma follows directly for the propositional variables and their negations. Consider the following cases:*

- *For* $\varphi = \alpha \vee \beta, \alpha \wedge \beta$ *or the temporal operators the proof is like in [26].*
- *Let* $\varphi = \overline{\mathrm{K}}_l\alpha$*. If* $\mathrm{M}'_k, s \models \overline{\mathrm{K}}_l\alpha$ *with* $|P'_k| \leq f_k(\overline{\mathrm{K}}_l\alpha)$*, then by Definition 7 we have that there is a* $k-$*computation* π *such that* $\pi(0) = \iota$ *and* $(\exists j \leq k)\ s \sim_l \pi(j))$ *and* $M'_k, \pi(j) \models \alpha$*. Hence, by induction we obtain that for some* $j \leq k$ *the propositional formula* $[\alpha]_k^{[0,0]} \wedge [M^{\alpha,\pi(j)}]_k$ *is satisfiable. Let* $ii = f_k(\alpha) + 1$ *be the index of a new symbolic* $k-$*path which satisfies the formula* $I_\iota(w_{0,ii})$*. Therefore, by the construction above, it follows that the propositional formula* $I_\iota(w_{0,ii}) \ \wedge \ \bigvee_{j=0}^{k}\left([\alpha]_k^{[j,ii]} \wedge H_l(w_{0,0}, w_{j,ii})\right) \ \wedge [M^{\overline{\mathrm{K}}_l\alpha,s}]_k$ *is satisfiable. Therefore, the following propositional formula is satisfiable:*
  $\bigvee_{1\leq i\leq f_k(\overline{\mathrm{K}}_l\alpha)}\Big(I_\iota(w_{0,i}) \ \wedge \ \bigvee_{j=0}^{k}\left([\alpha]_k^{[j,i]} \wedge H_l(w_{0,0}, w_{j,i})\right) \wedge [M^{\overline{\mathrm{K}}_l\alpha,s}]_k\Big)$.
  Hence, by the definition of the translation of an **ECTLK** *formula, the above formula is equal to the propositional formula* $[\overline{\mathrm{K}}_l\alpha]_k^{[0,0]} \wedge [M^{\overline{\mathrm{K}}_l\alpha,s}]_k$.
- *The other proofs are similar.*

Theorem 2 *Let* M *be a model,* M_k *be a* $k-$*model of* M*, and* φ *be an* **ECTLK** *formula. Then,* $M \models_k \varphi$ *iff* $[\varphi]_{\mathrm{M}_k} \wedge [M^{\varphi,\iota}]_k$ *is satisfiable.*

Proof 6.4 *Follows from Lemmas 3 and 4.*

Corollary 2 $M \models_k \neg\varphi$ *iff* $[\varphi]_{\mathrm{M}_k} \wedge [M^{\varphi,\iota}]_k$ *is unsatisfiable for* $k = |M|$*.*

This concludes our analysis of the translation technique. We now give an example to demonstrate how it can be put into practice.

7 MAS coordination and attacking generals

The framework described in the previous sections allows us to verify the temporal epistemic properties of MAS. In principle, by means of BMC on **CTLK** we can check formulas representing:

- Private and group knowledge of a MAS about a changing world,
- Temporal evolution of knowledge in a MAS,
- Any combination of the above.

In practice the technique above is most useful when the following prerequisites are observed. First we should be able to specify fully the system under consideration. This can be done for instance by giving a complete description of it in terms of interpreted systems, i.e., by spelling out the sets of local states, actions, protocols, and transition function. In this way we can build the model in an automatic way (details of how this can be done are not presented in this paper). Second, the benefits of the BMC machinery are more evident when the task is to check that:

1. an **ACTLK** formula is false (on an interpreted system).
2. an **ECTLK** formula is true (on an interpreted system).

We perform 1) when we would like to check the model for faults, i.e., we would check whether some particular formula is actually false in the model. For instance we may want to check whether a particular interpreted system does not guarantee that common knowledge of a particular fact is always maintained in the future. This would amount to checking whether the interpreted system is a counter model for a formula of type $\mathrm{AG}C\phi$.

We perform 2) when we would like to check whether the model provides for a realisation of a formula. For example we might want to check whether there is a trace in the temporal evolution of the system where common knowledge is not obtained. That would amount to checking whether the formula $\mathrm{EG}\neg C\phi$ holds.

In MAS literature examples of the type above appear in a variety of scenarios. Rather than producing yet another ad-hoc example, in order to demonstrate the technique, we revisit a widely discussed scenario: the coordinated attack problem. This is an example discussed in MAS, in distributed computing, as well as in epistemic logic. It concerns coordination of agents in the presence of unreliable communication.

> Two divisions of an army, each commanded by a general, are camped on the hilltops overlooking a valley. In the valley awaits the enemy. It is clear that if both divisions attack the enemy simultaneously, they will win the battle. While if one division attacks, it will be defeated. As a result neither general will attack unless he is absolutely sure the other will attack with him. In particular, one general will not attack if he receives no messages. The commander of the first division wishes to coordinate a simultaneous attack (at some point the next day). The generals can only communicate by means of messengers. Normally it takes a messenger one hour to get from one encampment to the other. However, it is possible that he will get lost in the dark or, worse yet, be captured by the enemy. Fortunately, on this particular night, everything goes smoothly. How long will it take them to coordinate an attack?
>
> ([7] page 176).

This example is appealing for at least two reasons. First, it is an instance of a recurring problem in coordination for action in MAS. Second, it can be formally analysed by means of interpreted systems and temporal epistemic logic. Crucially two key properties can be proven about the scenario above.

- No general will attack before it is common knowledge that they will both attack.
- No joint protocol can establish common knowledge, unless the delay with which the messages may be delivered is bounded.

From this one can infer that the generals will not attack on the night, even though the messengers do deliver the messages in exactly one hour. We refer to the literature [10] for a comprehensive analysis of the example, which turns out to be more subtle than it may appear at first. What we point out here is that the problem resides with the agents being forced to contemplate the possibility of the messenger getting lost at each round. This makes it impossible for common knowledge to be obtained in this circumstance[3].

Obviously it is problematic to perform model checking on the scenario as described above. The reason is that it is in fact a description for a *family* of joint protocols for the generals (the question of how long it will take to coordinate is more technically posed as "what joint protocol should the generals be running"). Indeed it looks difficult to prove in any way other than analytically as in the literature impossibility results of the kind mentioned above.

For the purpose of this paper, we choose a particular joint protocol for the scenario above and verify the truth and falsity of particular formulas that capture the key characteristics of the scenario.

The variant we analyse is the following:

[3]One should not infer from this, as it is sometimes mistakenly done, that common knowledge can *never* been achieved by message passing. The key element here is that messages may be delayed without a bound.

After having studied the opportunity of doing so, general A may issue a request-to-attack order to General B. If so, A will then wait to receive an acknowledgement from B, and will attack immediately after having received it. General B will not issue request-to-attack orders himself, but if his assistance is requested, he will acknowledge the request, and will attack after a suitable time for his messenger to reach A (*assuming no delays*) has elapsed. A joint attack guarantees success, and any non-coordinated attack causes defeat for the army involved.

We can model the example above with an interpreted system as follows. The local states for the agents are:

- $L_A = \{plan, wait, go, win, defeat\}$, $L_B = \{wait, attacking, go, win, defeat\}$,
- $L_E = \{\epsilon, deliver_B, deliver_A\}$.

The sets of actions available to the agents are as follows: $Act_A = Act_B = \{\lambda, attack, fight\}$, $Act_E = \{transmit, delay\}$. The protocols the agents are running are as follows:

- $P_A(plan) = \{\lambda, attack\}, P_A(wait) = \{\lambda\},\ P_A(go) = \{attack\}, P_A(win) = P_A(defeat) = \{\lambda\}$,
- $P_B(win) = P_B(defeat) = \{\lambda\}, P_B(attacking) = \{attack\},\ P_B(wait) = \{\lambda\}, P_B(go) = \{fight\}$,
- $P_E(\epsilon) = P_E(deliver_B) = P_E(deliver_A) = \{delay, transmit\}$.

It should be straightforward to infer the transition system that is induced by the informal description of the scenario we considered above together with the local states and protocols defined above. For example the following is a trace of the system. Other traces ending in states of failure are immediate to derive, and involve settings in which the acknowledgement from B is delayed by the environment.

$$(plan, wait, \epsilon) \overset{\lambda,\lambda,transmit}{\longrightarrow} (plan, wait, \epsilon) \overset{attack,\lambda,delay}{\longrightarrow} (wait, wait, deliver_b)$$
$$\overset{\lambda,\lambda,delay}{\longrightarrow} (wait, wait, deliver_b) \overset{\lambda,\lambda,transmit}{\longrightarrow} (wait, attacking, \epsilon) \overset{\lambda,attack,transmit}{\longrightarrow} (go, go, \epsilon)$$
$$\overset{fight,fight,transmit}{\longrightarrow} (win, win, \epsilon)$$

We now encode the local states in binary form in order to use them in the model checking technique. Given that A can be in 5 different local states we shall need 3 bits to encode its state; we take: $(0,0,0) = plan, (0,0,1) = wait, (0,1,0) = go, (0,1,1) = win, (1,0,0) = defeat$. Similarly for B: $(0,0,0) = wait, (0,0,1) = attacking, (0,1,0) = go, (0,1,1) = win, (1,0,0) = defeat$. The modelling of the environment E requires only two bits: $(0,0) = \epsilon, (0,1) = deliver_B, (1,0) = deliver_A$.

In view of this a global state is modelled by a byte: $g = (s[1], s[2], s[3], s[4], s[5], s[6], s[7], s[8])$. For instance the initial state $\iota = (plan, wait, \epsilon)$ is represented as a tuple of eight 0's. If we are to represent it in terms of propositional atoms, we shall have to insist on the atoms coding the state to be in the state of false. In other words, we would encode the initial state as follows: $I_\iota(w_{0,0}) = \bigwedge_{i=1}^{8} \neg w_{0,0}[i]$.

Some properties we may be interested in checking for the example above are the following:

1. $M \models \mathrm{AG}(\mathbf{attack{-}order} \rightarrow \mathbf{K_B attack{-}order})$
2. $M \models \mathrm{AG}(\mathbf{attack{-}ack} \rightarrow \mathbf{K_A K_B attack{-}order})$
3. $M \models \mathrm{EF}\mathbf{fail} \wedge \mathrm{EF}\mathbf{success}$
4. $M \models \mathrm{EG}\overline{\mathrm{C}}\neg(\mathbf{attack{-}ack})$

where the proposition **attack−order** is true on all the states of the model M (for the interpreted system) with the exception of $(plan, wait, \epsilon)$ and $(wait, wait, deliver_b)$. **attack−ack** is true on all states in which A is either in *go* state or anything that follows it in the run, that is either in

$(go, *, *)$, or $(win, *, *)$ or $(defeat, *, *)$. The propositions **fail** and **success** are true at the global states $(defeat, defeat, *)$ and $(win, win, *)$, respectively, where $*$ denotes an arbitrary local state.

Property 1) states that whenever the order has been received, agent B knows about it. Property 2) says that when the order has been acknowledged agent A knows that agent B knows of the order. Property 3) states that there exist (separate) evolutions leading to success and failure. Property 4) states that there exists a computation path in which common knowledge of the attack-order having been sent is never achieved.

Formulas 1-4 are true on the interpreted system in consideration. Formulas 1, and 2, are **ACTLK** formulas, so in order to check them we shall have to encode the whole model. We can do this in the BMC technique reported above, but as mentioned already the benefits of BMC are most apparent when only a *fraction* of the model is generated. For example this happens in formulas 3, 4, where we need to check validity of an **ECTLK** formula in the model. For the purposes of this paper we check validity of formula 4, by means of the technique presented here.

$$\varphi := \mathrm{EG}\overline{\mathrm{C}}\neg(\mathbf{attack-ack}).$$

The other translations are similar.

The translation of the proposition used in φ is as follows: $\mathbf{attack-ack(w)} := (\neg\mathbf{w[1]} \wedge \mathbf{w[2]}) \vee (\mathbf{w[1]} \wedge \neg\mathbf{w[2]} \wedge \neg\mathbf{w[3]})$, which means that **attack−ack** holds at all the global states with the first local state equal to $(0,1,0)$, $(0,1,1)$ or $(1,1,0)$.

The translation of the equality of the i-local states, for $i = 1, 2, 3$, is as follows (here 1 stands for A, 2 for B, and 3 for E): $H_1(w,v) = \bigwedge_{i=1}^{3} w[i] \Leftrightarrow v[i]$, $H_2(w,v) = \bigwedge_{i=4}^{6} w[i] \Leftrightarrow v[i]$, and $H_3(w,v) = \bigwedge_{i=7}^{8} w[i] \Leftrightarrow v[i]$.

We calculate that $f_1(\varphi) = 3$ (see Definition 10), so we need to exploit three symbolic paths for the translation with $k = 1$. To proceed with the translation, the first thing we need to translate is the initial state $\iota = (plan, wait, \epsilon)$, where ι is binary represented by $(0, \ldots, 0)$. With the representation above this will be encoded by the propositional formula $I_\iota(w_{0,j}) := \bigwedge_{i=1}^{8} \neg w_{0,j}[i]$, for $1 \leq j \leq 2$.

The next step is to translate the transitions $T(w_{i,j}, w_{i+1,j})$; for simplicity we report only on one transition for the case $k = 1$, and in particular only the formula $T(w_{0,1}, w_{1,1})$ representing the first transition of the first path. The remaining formula $T(w_{0,2}, w_{1,2})$

Consider then the joint actions $(\lambda, \lambda, \lambda)$, $(attack, \lambda, delay)$, $(attack, \lambda, transmit)$. They generate three possible successors of the initial state:
$(plan, wait, \epsilon)$, $(wait, wait, deliver_B)$, $(wait, attacking, \epsilon)$, respectively.

The corresponding formula is $T(w_{0,1}, w_{1,1}) := \bigwedge_{i=1}^{8} \neg w_{0,1}[i] \wedge ((\bigwedge_{i=1}^{8} \neg w_{1,1}[i]) \vee (\bigwedge_{i=1,2,4,5,6,7} \neg w_{1,1}[i] \wedge \bigwedge_{i=3,8} w_{1,1}[i]) \vee (\bigwedge_{i=1,2,4,5,7,8} \neg w_{1,1}[i] \wedge \bigwedge_{i=3,6} w_{1,1}[i]))$.

To encode the whole example we should model all the transitions for all the k's starting from $k := 1$. We do not do it here.

Let us now encode the formula φ we would like to check. $[\varphi]_1^{[0,0]} :=$

$$\bigvee_{1 \leq i \leq 3} \Big(\bigwedge_{l=1}^{8} (w_{0,0}[l] \Leftrightarrow w_{0,i}[l]) \wedge \bigvee_{l=0}^{1} T(w_{1,i}, w_{l,i}) \ \wedge \ \bigwedge_{j=0}^{1} [\overline{\mathrm{C}}\neg(\mathbf{attack-ack})]_{\mathbf{1}}^{[\mathbf{j,i}]} \Big).$$

Next:

$$[\overline{\mathrm{C}}\neg(\mathbf{attack-ack})]_{\mathbf{1}}^{[\mathbf{j,i}]} := [\overline{\mathrm{E}}\neg(\mathbf{attack-ack})]_{\mathbf{1}}^{[\mathbf{j,i}]} :=$$

$$\bigvee_{1 \leq n \leq 3} \Big(I_\iota(w_{0,n}) \ \wedge \ \bigvee_{m=0}^{1} \big(\neg[\mathbf{attack-ack}]_{\mathbf{1}}^{[\mathbf{m,n}]} \ \wedge \bigvee_{\mathbf{1 \leq l \leq 3}} \mathbf{H_l(w_{j,i}, w_{m,n})} \big) \Big),$$

where $[\mathbf{attack-ack}]_{\mathbf{1}}^{[\mathbf{m,n}]} = \mathbf{attack-ack(w_{m,n})}$ (as defined above).

Checking that the coordinated attack protocol satisfies the temporal epistemic formula above can now be done by feeding a SAT solver with the propositional formula generated in this method. This would produce a solution, thereby proving that the propositional formula is satisfiable.

8 Conclusions

The field of MAS theories has traditionally been concerned with the *specification* of MAS. In this line of work, the theorems of particular modal logics are seen as specifying macroscopic properties of agents such as their knowledge, belief, intention, and the temporal evolution of these.

More recently, the importance of MAS verification has been highlighted by a number of papers in the area [11, 12]. One of the problems of verifying MAS is that a plain temporal logic like CTL is not sufficient to represent the mental states of the agents in a MAS. Enriching CTL with modalities for knowledge, belief, and intention raises the question of what semantics to use to interpret these modalities. It has long been argued [28] that plain Kripke semantics is not adequate to perform this task. If we aim to *verify* MAS, we need to find an intuitive computational correspondence for these notions. In this paper we have used the semantic model of interpreted systems, and integrated it with the verification technique of bounded model checking, one of the currently most prominent techniques from verification of distributed systems.

In this line of work, we have recently integrated the theory presented here into a fully-automated model checker, so that experimental results can be produced. Preliminary results appear encouraging and are reported in [16]. The system allows the user will to give a full characterisation of the system in terms of a variant of Estelle [14]. This can be a rather lengthy and error-prone process, so, at the same time, we are working on a translator from a specification given in interpreted systems to an Estelle program. Ultimately it is hoped that it will be possible to verify properties expressible in **CTLK** on a specification given directly in terms of interpreted systems.

Finally we should like to stress that this paper belongs to a line of research on model checking time and knowledge encompassing theoretical investigations [21, 11, 12], as well as experimental work [18, 13]. A comparison of the practical results achievable with the various techniques seems to be a fruitful avenue for further work.

Ultimately, verification of multi-agent systems will have to involve checking not only informational properties as the ones treated here but also motivational ones such as desires and intentions [2, 27, 15], and normative [23]. This task is made particularly complex by the fact that these modalities are typically not given an interpretation in terms of local states as it is the case for knowledge as discussed here. For the case of correct functioning behaviour we see potential in applying the technique presented here to the case of deontic interpreted systems [19]. We leave this for further work.

Acknowledgements. The authors are grateful to the reviewers of the second joint conference on autonomous agents and multi-agent systems (AAMAS-03) for extensive comments on a shorter version of this paper.

References

[1] A. Biere, A. Cimatti, E. Clarke, and Y. Zhu. Symbolic model checking without BDDs. In *Proc. of TACAS'99*, volume 1579 of *LNCS*, pages 193–207. Springer-Verlag, 1999.

[2] M. E. Bratman. *Intentions, Plans, and Practical Reason.* Harvard University Press: Cambridge, MA, 1987.

[3] E. Clarke, A. Biere, R. Raimi, and Y. Zhu. Bounded model checking using satisfiability solving. *Formal Methods in System Design*, 19(1):7–34, 2001.

[4] E. M. Clarke, O. Grumberg, and D. Peled. *Model Checking.* MIT Press, 1999.

[5] E. A. Emerson. *Handbook of Theoretical Computer Science*, volume B: Formal Methods and Semantics, chapter Temporal and Modal Logic, pages 995–1067. Elsevier, 1990.

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

[7] R. Fagin, J. Y. Halpern, Y. Moses, and M. Y. Vardi. *Reasoning about Knowledge*. MIT Press, Cambridge, 1995.

[8] R. Fagin, J. Y. Halpern, and M. Y. Vardi. What can machines know? On the properties of knowledge in distributed systems. *Journal of the ACM*, 39(2):328–376, Apr. 1992.

[9] J. Halpern, R. Meyden, and M. Y. Vardi. Complete axiomatisations for reasoning about knowledge and time. *SIAM Journal on Computing*, 2003. To Appear.

[10] J. Halpern and Y. Moses. Knowledge and common knowledge in a distributed environment. *Journal of the ACM*, 37(3):549–587, 1990. A preliminary version appeared in *Proc. 3rd ACM Symposium on Principles of Distributed Computing*, 1984.

[11] W. van der Hoek and M. Wooldridge. Model checking knowledge and time. In *Proc. of the 9th Int. SPIN Workshop (SPIN'02)*, volume 2318 of *LNCS*, pages 95–111. Springer-Verlag, 2002.

[12] W. van der Hoek and M. Wooldridge. Tractable multiagent planning for epistemic goals. In *Proc. of the 1st Int. Conf. on Autonomous Agents and Multi-Agent Systems (AAMAS'02)*, July 2002. to appear.

[13] M.-P. Huget, M. Wooldridge, M. Fisher and S. Parsons. Model checking multiagent systems with mable. In *Proceedings of the First International Conference on Autonomous Agents and Multiagent Systems (AAMAS-02)*, Bologna, Italy, July 2002.

[14] ISO/IEC 9074(E), Estelle - a formal description technique based on an extended state-transition model. International Standards Organization, 1997.

[15] B. van Linder, W. van der Hoek, and J.-J. Meyer. Seeing is believing, *and so are hearing and jumping. Journal of Logic, Language, and Information*, 6(1):33–61, 1997.

[16] A. Lomuscio, T. Łasica, and W. Penczek. Bounded model checking for interpreted systems: preliminary experimental results. In M. Hinchey, editor, *Proceedings of FAABS II*, volume 2699 of *LNCS*. Springer Verlag, 2003.

[17] A. Lomuscio, R. Meyden, and M. Ryan. Knowledge in multi-agent systems: Initial configurations and broadcast. *ACM Transactions of Computational Logic*, 1(2), 2000.

[18] A. Lomuscio, F. Raimondi, and M. Sergot. Towards model checking interpreted systems. In *Proceedings of Mochart — First International Workshop on Model Checking and Artificial Intelligence*, 2002.

[19] A. Lomuscio and M. Sergot. Deontic interpreted systems. *Studia Logica*, 75, 2003.

[20] Z. Manna and A. Pnueli. *The temporal logic of reactive and concurrent systems*, volume 1. Springer-Verlag, Berlin/New York, 1992.

[21] R. van der Meyden and H. Shilov. Model checking knowledge and time in systems with perfect knowledge. In *Proceedings of Proc. of FST&TCS*, volume 1738 of *Lecture Notes in Computer Science*, pages 432–445, Hyderabad, India, 1999.

[22] R. v. Meyden and K. Wong. Complete axiomatizations for reasoning about knowledge and branching time. *Studia Logica*, 75, 2003.

[23] J.-J. C. Meyer and R. J. Wieringa. Deontic logic: A concise overview. In *Deontic Logic in Computer Science*, Wiley Professional Computing Series, chapter 1, pages 3–16. John Wiley and Sons, Chichester, UK, 1993.

[24] M. Moskewicz, C. Madigan, Y. Zhao, L. Zhang, and S. Malik. Chaff: Engineering an efficient SAT solver. In *Proc. of the 38th Design Automation Conference (DAC01)*, pages 530–535, June 2001.

[25] W. Penczek and A. Lomuscio. Verifying epistemic properties of multi-agent systems via bounded model checking. In T. Sandholm, editor, *Proceedings of the Second International Joint Conference on Autonomous Agents and Multi-agent systems (AAMAS-03)*, 2003.

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

[27] A. S. Rao and M. P. Georgeff. Decision procedures for BDI logics. *Journal of Logic and Computation*, 8(3):293–343, June 1998.

[28] M. Wooldridge. Computationally grounded theories of agency. In E. Durfee, editor, *Proceedings of ICMAS, International Conference of Multi-Agent Systems*. IEEE Press, 2000.