A Temporal Epistemic Logic with a Reset Operation*

Alessio Lomuscio
Department of Computing
Imperial College London
180 Queen's Gate
London SW7 2BZ, United Kingdom
email: A.Lomuscio@imperial.ac.uk

Bożena Woźna
Institute of Computer Science
Jan Długosz University
Al. Armii Krajowej 13/15
42-200 Częstochowa, Poland
email: b.wozna@ajd.czest.pl

ABSTRACT

We present an axiomatisation for an extension of a temporal epistemic logic with an epistemic "reset" operator defined on the intersection between epistemic and temporal relations. Additionally we show the logic has the finite model property, hence it is decidable.

Categories and Subject Descriptors

F.4.1 [**Mathematical Logic and Formal Languages**]: Mathematical Logic; I.2.4 [**Artificial Intelligence**]: Knowledge Representation Formalisms and Methods

General Terms

Theory

Keywords

Epistemic logic, axiomatisation, decidability

1. INTRODUCTION

Modal logics provide a formal framework to specify and reason about computations in distributed and multi-agent systems. In particular, interpreted systems [6] provide a formal Kripke-style semantics to reason about different states of knowledge of the agents. On this semantics, different concepts of knowledge have been explored, from implicit [11] to distributed, common, deductive, algorithmic knowledge, etc. All these logics are normally seen as formal specification languages for representing agents' knowledge.

A number of works have recently appeared in the literature relating to model checking techniques [4] for verifying automatically that a multi-agent system satisfies a particular temporal epistemic specification [13, 19, 9, 20]. While model checking presents some documented advantages over theorem proving, the core difficulty of the approach is the "state explosion problem", i.e. the fact the model representing the system grows very quickly to a size which is difficult to manage even when encoded symbolically.

*The research presented in this paper was supported by EPSRC (grant GR/S49353).

AAMAS'07 May 14–18 2007, Honolulu, Hawai'i, USA.

Bounded model checking [2] and other SAT-based approaches attempt to ease this problem by performing aggressive depth-first search on appropriate restricted submodels. This technique has been shown to be quite effective in temporal-epistemic logic as well [19, 21], but a key problem is that an epistemic modality defined on equality of local states (as in interpreted systems) forces to consider states reachable from *any* possible branch from the initial state, thereby limiting the advantages of the techniques. It is sometimes useful though, to reason about epistemic properties of agents that result from a "reset" operation, i.e., to reason about the knowledge regarding the future from the current time, *as if* a pruning of the model was performed at that instant and only the submodel *generated* by that point (as initial state) were to be considered. This happens, for instance, when different instances of the same property are checked a number of times over the same run, such as safe receipt of a stream of bits.

In this paper we introduce an epistemic modality R_i (for "reset") that intrinsically incorporates the concept of resetting the model at the point where the modality is considered. This is equivalent to assuming that the agents are able to distinguish (i.e., to remove from their epistemically indistinguishable set of accessible states) the current state from states in the past and from states belonging to a different computational branch from the one that terminated in the state under evaluation. Alternatively, one can see this modality as expressing standard implicit knowledge but under the assumption the system enjoys a particular form of perfect recall [18], so that the agent would be able to recognise states with the same prefix. Representing perfect recall or even weaker variants of it is particularly costly in terms of model checking as the size of local states grows rapidly with time. The alternative that this operator suggests is to use a standard semantics but evaluate the R_i on the intersection of the epistemic relation for agent i with the reflexive transitive closure of the temporal relation.

Although our research is *inspired* by efficient implementations of model checking algorithms for epistemic temporal logic, in this paper we focus on the metalogical properties of the resulting logic and leave for the future the coding of specialised verification algorithms and data structures. Specifically, in Section 2 we present syntax and semantics. In Section 3 we exemplify the formalism to the bit transmission problem. Section 4 is devoted to the construction of the underlying machinery to prove the main results of the paper. Sections 5 and 6 present this main results, namely a decidability theorem and a completeness proof for the logic;

these follow the schema of the completeness and decidability proof in [10] which we extend to deal with the logic presented here. Section 7 contains conclusions and final remarks.

2. SYNTAX AND SEMANTICS

Here we give a syntax and semantics of a temporal epistemic logic $\mathcal{L}$ with an operator that is defined in terms of the intersection of the transitive reflexive closure of a serial temporal relation and the standard epistemic relation (which is an equivalence relation); the temporal part of $\mathcal{L}$ corresponds to the well-known CTL [3], and the epistemic part is the logic $S5_n$ [6].

Syntax. Let $\mathcal{PV}$ be a set of propositional variables containing the symbol $\top$, which stands for the logical constant TRUE, $\mathcal{A}g$ a finite set of agents, $p \in \mathcal{PV}$ and $i \in \mathcal{A}g$. The language $\mathcal{L}$ is defined by the following grammar:

$$\varphi := p \mid \neg\varphi \mid \varphi \wedge \varphi \mid \mathrm{EX}\varphi \mid \mathrm{E}(\varphi \mathrm{U} \varphi) \mid \mathrm{A}(\varphi \mathrm{U} \varphi) \mid \mathrm{K}_i\varphi \mid \mathrm{R}_i\varphi$$

In the above, the operators E and A denote path quantifiers with the following meaning "*for some computation path*" and "*for all the computation paths*", respectively; the operators X and U are the path operators read as "*at the next step*" and "*until*", respectively; the operator R_i is read as "*following a reset operation agent i knows*", and K_i represents the standard epistemic modality, read as "*agent i knows*". The remaining Boolean and temporal operators can be defined in a usual way.

Let φ and ψ be $\mathcal{L}$ formulae. We say that ψ is a *subformula* of φ and write $\psi \in Sub(\varphi)$ if either (a) $\psi = \varphi$; or (b) φ is of the form $\neg\alpha$, $\mathrm{EX}\alpha$, $\mathrm{K}_i\alpha$, or $\mathrm{R}_i\alpha$, and ψ is a subformula of α; or (c) φ is of the form $\alpha \wedge \beta$, $\mathrm{E}(\alpha \mathrm{U} \beta)$, or $\mathrm{A}(\alpha \mathrm{U} \beta)$ and ψ is a subformula of either α or β. Further, we define the *length* of a formula φ to be the number of symbols (parentheses, propositional variables and connectives, modal operators) of which it consists.

Semantics. Traditionally, the semantics of temporal epistemic logics is given on interpreted systems, defined as follows [6].

Definition 1 (Interpreted system). *Given a set of of agents $\mathcal{A}g = \{1, \ldots, n\}$, consider:*

- *$L_1, \ldots, L_n$ to be the countable sets of* local states *for the agents and L_e a set of local states for the environment; Any element in $S \subseteq L_1 \times \cdots \times L_e$ is called a* global state.
- *$P_1 : L_1 \to 2^{Act_1}, \ldots, P_n : L_n \to 2^{Act_n}$ a set of* protocols *for the agents and $P_e : L_e \to 2^{Act_e}$ a protocol for the environment where Act_i (respectively, Act_e) is a set of* actions *for agent i (respectively, the environment);*
- *$\tau : S \times Act_1 \times \ldots Act_n \times Act_e \to S$ a transition function for the system;*
- *$\mathcal{V} : S \mapsto 2^{\mathcal{PV}}$ is an interpretation for the atoms in $\mathcal{PV}$.*

The tuple $IS = (S, \tau, \mathcal{V})$ is called an interpreted system. *IS is the class of all interpreted systems; we refer to [6] for more details on the above.*

We interpret the language $\mathcal{L}$ on models below.

Definition 2 (Model). *Given a set of agents $\mathcal{A}g$, a* model *is a tuple $M = (S, T, \{\sim_i\}_{i\in\mathcal{A}g}, \mathcal{V}, \{\sqcap_i\}_{i\in\mathcal{A}g})$, such that S is a countable set of states; $T \subseteq S \times S$ is a serial relation on S; for each agent $i \in \mathcal{A}g$, $\sim_i \subseteq S \times S$ is an equivalence relation; $\mathcal{V} : S \mapsto 2^{\mathcal{PV}}$ is a valuation function that assigns to each state a set of proposition variables that are assumed to be true at that state; and $\sqcap_i = \sim_i \cap T^*$ for each $i \in \mathcal{A}g$, where T^* denotes the reflexive and transitive closure of T. We call $F = (S, T, \{\sim_i\}_{i\in\mathcal{A}g}, \{\sqcap_i\}_{i\in\mathcal{A}g})$ a* frame.

A *path* in M is an infinite sequence $\pi = (s_0, s_1, \ldots)$ of states such that $(s_i, s_{i+1}) \in T$ for each $i \in \{0, 1, \ldots\}$. For a path $\pi = (s_0, s_1, \ldots)$, we take $\pi(i) = s_i$. By $\Pi(s)$ we denote the set of all the paths starting at $s \in S$.

Definition 3 (Satisfaction). *Let M be a model, s a state, and α, $\beta \in \mathcal{L}$. The satisfaction relation $\models$, indicating truth of a formula in model M at state s, is defined inductively as follows:*

$(M, s) \models p$ *iff* $p \in \mathcal{V}(s)$,
$(M, s) \models \neg\alpha$ *iff* $(M, s) \not\models \alpha$,
$(M, s) \models \alpha \wedge \beta$ *iff* $(M, s) \models \alpha$ *and* $(M, s) \models \beta$,
$(M, s) \models \mathrm{EX}\alpha$ *iff* $(\exists \pi \in \Pi(s))(M, \pi(1)) \models \alpha$,
$(M, s) \models \mathrm{E}(\alpha \mathrm{U} \beta)$ *iff* $(\exists \pi \in \Pi(s))(\exists_{m\geq 0})[(M, \pi(m)) \models \beta$ *and* $(\forall_{0\leq j<m})(M, \pi(j)) \models \alpha]$,
$(M, s) \models \mathrm{A}(\alpha \mathrm{U} \beta)$ *iff* $(\forall \pi \in \Pi(s))(\exists_{m\geq 0})[(M, \pi(m)) \models \beta$ *and* $(\forall_{0\leq j<m})(M, \pi(j)) \models \alpha]$,
$(M, s) \models \mathrm{K}_i\alpha$ *iff* $(\forall s' \in S)$ $(s \sim_i s'$ *implies* $(M, s') \models \alpha)$,
$(M, s) \models \mathrm{R}_i\alpha$ *iff* $(\forall s' \in S)$ $(s \sqcap_i s'$ *implies* $(M, s') \models \alpha)$.

Satisfaction for the Boolean and temporal operators as well as the epistemic modality K_i is standard. The formula $\mathrm{R}_i\alpha$ holds at state s in a model M if α holds in all the states that are reachable from s via temporal relation T and they are in the $i - th$ epistemic relation with s. In other words, $(M, s) \models \mathrm{R}_i\alpha$ means that in the state s agent i knows α under assumption that he does not consider as possible states that do not belong to the future of s.

We also say that φ is *valid* in M (written $M \models \varphi$), if $M, s \models \varphi$ for all states $s \in S$, and that φ is *satisfiable* in M, if $M, s \models \varphi$ for some state $s \in S$. Further, we say that φ is *valid* (written $\models \varphi$), if φ is valid in all the models M, and that φ is *satisfiable* if it is satisfiable in some model M. In the latter case M is said to be a model for φ.

The models as defined here can also be used to interpret formulae on interpreted systems (Definition 1) directly. To do so we need to associate a model to every interpreted system.

Definition 4 (Generated models). *Given an interpreted system $IS = (S, \tau, \mathcal{V})$ a model M_{IS} for IS is a tuple $M = (S, T, \{\sim_i\}_{i\in\mathcal{A}g}, \mathcal{V}, \{\sqcap_i\}_{i\in\mathcal{A}g})$ such that S is the set of global states; T is defined by sTs' iff there exist actions $act_1 \in ACT_1, \ldots, act_n \in ACT_n, act_e \in ACT_e$ such that $\tau(s, act_1, \ldots, act_n, act_e) = s'$; for any $i \in \mathcal{A}g$ $s \sim_i s'$ iff $l_i(s) = l_i(s')$, where $l_i : S \to L_i$ is the function returning the local state of agent i given a global state s; $\mathcal{V}$ is an interpretation for the proposition variables; for any $i \in \mathcal{A}g$ $\sqcap_i = \sim_i \cap T^*$.*

We say that a formula ϕ is true (respectively, valid) on an interpreted system IS if ϕ is true (respectively, valid) on the generated model.

Definition 5. *A formula ϕ is valid on the class of interpreted systems $\mathcal{IS}$ if ϕ is valid on any model generated from any $IS \in \mathcal{IS}$.*

It is of note that we can show the following which will be useful for the completeness proof.

LEMMA 1. *For any model M there exists an interpreted system IS such that, the generated model of IS is isomorphic to M.*

Proof. The proof follows the same construction given in [15, 16].

3. THE BIT TRANSMISSION PROBLEM

In this section we present an example regarding the use of the operator R_i defined in this paper. In particular we analyse a variant of the bit transmission problem (BTP), a well-known example in reasoning about knowledge [6].

Imagine we have two processes, a *sender* $\mathfrak{S}$ and a *receiver* $\mathfrak{R}$, which communicate over a possibly faulty communication line. $\mathfrak{S}$ wants to send a finite stream of bits to $\mathfrak{R}$. They follow this protocol: $\mathfrak{S}$ immediately starts sending the bit to $\mathfrak{R}$, and continues to do so until it receives an acknowledgement from $\mathfrak{R}$. $\mathfrak{R}$ does nothing until it receives the bit; from then on it sends acknowledgements of receipt to $\mathfrak{S}$. When $\mathfrak{S}$ receives an acknowledgement, it stops sending the "old" bit to $\mathfrak{R}$, and performs a reset operation, thereby giving a sign to $\mathfrak{R}$ that a new bit will be sent[1]. Then, $\mathfrak{S}$ starts sending the new bit and the cycle repeats. We apply the $\mathcal{L}$ formalism to model and reason about the above scenario.

Let us begin by building a model $M = (S, T, \sim_{\mathfrak{S}}, \sim_{\mathfrak{R}}, \mathcal{V}, \sqcap_{\mathfrak{S}}, \sqcap_{\mathfrak{R}})$ for the BTP. There are three active components in the scenario: agents $\mathfrak{S}$ and $\mathfrak{R}$, and a communication channel represented by an environment $\mathfrak{E}$. Each of these can be modelled by considering their respective local states. For $\mathfrak{S}$, it is enough to consider five possible local states. They represent the value of the bit that $\mathfrak{S}$ is attempting to transmit, and whether or not $\mathfrak{S}$ has received an acknowledgement or an end signal from $\mathfrak{R}$. We thus have: $L_{\mathfrak{S}} = \{0,\ 1,\ 0\text{-}ack,\ 1\text{-}ack, end\}$. Let $n > 0$ be the maximal length of the stream of bits to be sent. We consider $L_{\mathfrak{R}} = \{x^j y \mid x \in \{0\text{-}ack, 1\text{-}ack\},\ y \in \{0,1\}, 0 < j \leq n\} \cup \{x^j \mid x \in \{0\text{-}ack, 1\text{-}ack\}, 0 < j \leq n\} \cup \{\epsilon\}$. $\mathfrak{R}$'s local state is ϵ if $\mathfrak{R}$ has received no bits from $\mathfrak{S}$. $\mathfrak{R}$'s local state is 0 (respectively 1) if the received bit is 0 (respectively 1). $\mathfrak{R}$'s local state is $k_1\text{-}ack \ldots k_j\text{-}ack$ (respectively $k_1\text{-}ack \ldots k_j\text{-}ack\ k$) for $k_1, \ldots, k_j, k \in \{0,1\}$ and $j \leq n$, if the stream of bits he received is $k_1 \ldots k_j$ (respectively $k_1 \ldots k_j k$) and $\mathfrak{S}$ has performed j reset actions. For $\mathfrak{E}$ it is enough to consider a singleton: $L_{\mathfrak{E}} = \{\cdot\}$.

The following sets of actions are available to the agents: $Act_{\mathfrak{S}} = \{sendbit, reset, \lambda\}$; $Act_{\mathfrak{R}} = \{sendack, sendend, \lambda\}$, where λ stands for no action. The actions $Act_{\mathfrak{E}}$ for the environment correspond to the transmission of messages between $\mathfrak{S}$ and $\mathfrak{R}$ on the unreliable communication channel. We will assume that the communication channel can transmit messages in both directions simultaneously and independently. The set of actions for $\mathfrak{E}$ is $Act_{\mathfrak{E}} = \{\leftrightarrow, \rightarrow, \leftarrow, -\}$, where $\leftrightarrow$ represents the action in which the channel transmits any message successfully in both directions, $\rightarrow$ that it transmits successfully from $\mathfrak{S}$ to $\mathfrak{R}$ but loses any message from $\mathfrak{R}$ to $\mathfrak{S}$, $\leftarrow$ that it transmits successfully from $\mathfrak{R}$ to $\mathfrak{S}$ but loses any message from $\mathfrak{S}$ to $\mathfrak{R}$, and $-$ that it loses any messages sent in either direction.

The protocols the agents are running are defined as follows: $P_{\mathfrak{S}}(0) = P_{\mathfrak{S}}(1) = \{sendbit\}$, $P_{\mathfrak{S}}(0\text{-}ack) = P_{\mathfrak{S}}(1\text{-}ack) = \{reset\}$, $P_{\mathfrak{S}}(end) = \{\lambda\}$, $P_{\mathfrak{R}}(\epsilon) = P_{\mathfrak{R}}(k_1\text{-}ack \ldots k_j\text{-}ack) = \{\lambda\}$, $P_{\mathfrak{R}}(k_1\text{-}ack \ldots k_{j-1}\text{-}ack\ k_j) = \{sendack\}$, $P_{\mathfrak{R}}(k_1\text{-}ack \ldots k_{n-1}\text{-}ack\ k_n) = \{sendend\}$, $P_{\mathfrak{E}}(\cdot) = \{\leftrightarrow, \rightarrow, \leftarrow, -\}$, where $k_1, \ldots, k_j, k_n \in \{0,1\}$ and $j < n$.

The evolution of the BTP is defined by means of an evolution function $t : (L_{\mathfrak{S}} \times L_{\mathfrak{R}} \times L_{\mathfrak{E}}) \times Act \to 2^{L_{\mathfrak{S}} \times L_{\mathfrak{R}} \times L_{\mathfrak{E}}}$, where Act is a subset of $Act_{\mathfrak{S}} \times Act_{\mathfrak{R}} \times Act_{\mathfrak{E}}$. It is straightforward to infer a definition of this function from the informal description of the scenario we considered above together with the local states and protocols defined above; the function t not only determines the set of reachable global states $S \subseteq L_{\mathfrak{S}} \times L_{\mathfrak{R}} \times L_{\mathfrak{E}}$, but it also gives us the transition relation T. Namely, for all states $s,\ s' \in S$, $(s, s') \in T$ iff there exists $act \in Act$ such that $t(s, act) = s'$.

To complete the description of M for BTP, we introduce the set of propositional variables: $\mathcal{PV} = \{\mathbf{ack}\} \cup \{\mathbf{jBit = 0}, \mathbf{jBit = 1}, \mathbf{reset_j} \mid 0 < j \leq n\}$, and we define the valuation function $\mathcal{V} : S \mapsto 2^{\mathcal{PV}}$ as:

- $\mathbf{ack} \in \mathcal{V}(s)$ if $l_{\mathfrak{S}}(s) = 0\text{-}ack$ or $l_{\mathfrak{S}}(s) = 1\text{-}ack$,
- $\mathbf{jBit = 0} \in \mathcal{V}(s)$ if $l_{\mathfrak{R}}(s) = (k_1 \ldots k_j \ldots k_k)$ and $(k_j = 0$ or $k_j = 0\text{-}ack)$ for $0 < j \leq k \leq n$,
- $\mathbf{jBit = 1} \in \mathcal{V}(s)$ if $l_{\mathfrak{R}}(s) = (k_1 \ldots k_j \ldots k_k)$ and $(k_j = 1$ or $k_j = 1\text{-}ack)$ for $0 < j \leq k \leq n$,
- $\mathbf{reset_j} \in \mathcal{V}(s)$ if $l_{\mathfrak{R}}(s) = (k_1 \ldots k_j \ldots k_k)$ and $k_k \notin \{0,1\}$ for $0 < j \leq k \leq n$.

Let us consider the following property ($\star$): "*whenever a fresh bit has been sent, $\mathfrak{S}$ knows that whenever he receives an acknowledgement, $\mathfrak{R}$ knows the value of that bit*". One can try to express this property with the following CTLK formula, for $0 < j \leq n$:

$$\mathrm{AG}[\mathbf{reset_j} \Rightarrow \mathrm{AGK}_{\mathfrak{S}}(\mathbf{ack} \Rightarrow (\mathrm{K}_{\mathfrak{R}}(\mathbf{jBit = 0}) \vee \mathrm{K}_{\mathfrak{R}}(\mathbf{jBit = 1})))]$$

But one can check that the above formula is not valid in M. The problem is that a bit received before a reset may account for the receiver's knowledge about the current bit. What we need to do is to express explicitly that past states should not be considered in $\mathfrak{S}$'s accessible states, following the reset operation. The epistemic modality R_i enables us not to include past states, and can be used to capture this intuition. Indeed, property ($\star$) can be formalised by the following $\mathcal{L}$ formula:

$$\mathrm{AG}[\mathbf{reset_j} \Rightarrow \mathrm{AGR}_{\mathfrak{S}}(\mathbf{ack} \Rightarrow (\mathrm{K}_{\mathfrak{R}}(\mathbf{jBit = 0}) \vee \mathrm{K}_{\mathfrak{R}}(\mathbf{jBit = 1})))]$$

It can be checked manually that this formula is valid in M.

4. FINITE MODEL PROPERTY

We now focus on proving that the language $\mathcal{L}$ has the *finite model property* (FMP); recall a logic has the FMP if any satisfiable formula is also satisfiable in a finite model. To prove this normally a quotient (or "filtration") is constructed, and then it is shown that the resulting finite structure is still a model for the formula in question. This technique, for example, has been used in [8, 17] to prove that PDL and a temporal deontic logic have FMP, respectively.

For some logics (in particular for $\mathcal{L}$) the quotient construction yields a quotient structure that is not a model; however, it still contains enough information to be unwound into a genuine model. In this specific case; to prove FMP for $\mathcal{L}$, we follow [5], where a combination of the filtration and unwinding technique has been applied to prove the FMP for CTL. We begin with providing definitions of two auxiliary structures: a *Hintikka structure* for a given $\mathcal{L}$ formula,

[1]For simplicity we assume that resets are communicated with no faults; an acknowledgement-based protocol could be introduced for resets without violating the properties we show below.

and the *quotient structure* for a given model. We follow the constructions and the scheme of the proofs presented in [10].

DEFINITION 6 (HINTIKKA STRUCTURE). *Let $\varphi \in \mathcal{L}$, and $\mathcal{A}g$ be a set of agents. A* Hintikka structure *for φ is a tuple $H = (S, T, \{\sim_i\}_{i\in\mathcal{A}g}, \mathbb{L}, \{\sqcap_i\}_{i\in\mathcal{A}g})$ such that S is a set of states, T is a serial binary relations on S, $\sim_i$ and $\sqcap_i$ are binary relations on S, and $\mathbb{L} : S \mapsto 2^{\mathcal{L}}$ is a labelling function assigning a set of formulae to each state such that $\varphi \in \mathbb{L}(s)$ for some $s \in S$. Moreover, $\mathbb{L}$ satisfies the following conditions:*

H.1. *if* $\neg\alpha \in \mathbb{L}(s)$, *then* $\alpha \notin \mathbb{L}(s)$

H.2. *if* $\neg\neg\alpha \in \mathbb{L}(s)$, *then* $\alpha \in \mathbb{L}(s)$

H.3. *if* $(\alpha \wedge \beta) \in \mathbb{L}(s)$, *then* $\alpha \in \mathbb{L}(s)$ *and* $\beta \in \mathbb{L}(s)$

H.4. *if* $\neg(\alpha \wedge \beta) \in \mathbb{L}(s)$, *then* $\neg\alpha \in \mathbb{L}(s)$ *or* $\neg\beta \in \mathbb{L}(s)$

H.5. *if* $\mathrm{E}(\alpha \mathrm{U} \beta) \in \mathbb{L}(s)$, *then* $\beta \in \mathbb{L}(s)$ *or* $\alpha \wedge \mathrm{EXE}(\alpha \mathrm{U} \beta) \in \mathbb{L}(s)$

H.6. *if* $\neg\mathrm{E}(\alpha \mathrm{U} \beta) \in \mathbb{L}(s)$, *then* $\neg\beta \wedge \neg\alpha \in \mathbb{L}(s)$ *or* $\neg\beta \wedge \neg\mathrm{EXE}(\alpha \mathrm{U} \beta) \in \mathbb{L}(s)$

H.7. *if* $\mathrm{A}(\alpha \mathrm{U} \beta) \in \mathbb{L}(s)$, *then* $\beta \in \mathbb{L}(s)$ *or* $\alpha \wedge \neg\mathrm{EX}(\neg\mathrm{A}(\alpha \mathrm{U} \beta)) \in \mathbb{L}(s)$

H.8. *if* $\neg\mathrm{A}(\alpha \mathrm{U} \beta) \in \mathbb{L}(s)$, *then* $\neg\beta \wedge \neg\alpha \in \mathbb{L}(s)$ *or* $\neg\beta \wedge \mathrm{EX}(\neg\mathrm{A}(\alpha \mathrm{U} \beta)) \in \mathbb{L}(s)$

H.9. *if* $\mathrm{EX}\alpha \in \mathbb{L}(s)$, *then* $(\exists t \in S)((s,t) \in T$ *and* $\alpha \in \mathbb{L}(t))$

H.10. *if* $\neg\mathrm{EX}\alpha \in \mathbb{L}(s)$, *then* $(\forall t \in S)((s,t) \in T$ *implies* $\neg\alpha \in \mathbb{L}(t))$

H.11. *if* $\mathrm{E}(\alpha \mathrm{U} \beta) \in \mathbb{L}(s)$, *then* $(\exists_{\pi\in\Pi(s)})(\exists_{n\geq 0})(\beta \in \mathbb{L}(\pi(n))$ *and* $(\forall_{j<n})\alpha \in \mathbb{L}(\pi(j)))$

H.12. *if* $\mathrm{A}(\alpha \mathrm{U} \beta) \in \mathbb{L}(s)$, *then* $(\forall_{\pi\in\Pi(s)})(\exists_{n\geq 0})(\beta \in \mathbb{L}(\pi(n))$ *and* $(\forall_{j<n})\alpha \in \mathbb{L}(\pi(j)))$

H.13. *if* $\mathrm{K}_i\alpha \in \mathbb{L}(s)$ *and* $s \sim_i t$, *then* $\alpha \in \mathbb{L}(t)$

H.14. *if* $\neg\mathrm{K}_i\alpha \in \mathbb{L}(s)$, *then* $(\exists t \in S)(s \sim_i t$ *and* $\neg\alpha \in \mathbb{L}(t))$

H.15. *if* $\mathrm{K}_i\alpha \in \mathbb{L}(s)$, *then* $\alpha \in \mathbb{L}(s)$

H.16. *if* $s \sim_i t$ *then* $\mathrm{K}_i\alpha \in \mathbb{L}(s)$ *iff* $\mathrm{K}_i\alpha \in \mathbb{L}(t)$

H.17. *if* $s \sim_i t$ *and* $s \sim_i u$ *and* $\mathrm{K}_i\alpha \in \mathbb{L}(t)$, *then both* $\alpha \in \mathbb{L}(u)$ *and* $\mathrm{K}_i\alpha \in \mathbb{L}(u)$

H.18. *if* $\mathrm{R}_i\alpha \in \mathbb{L}(s)$ *and* $(s \sqcap_i t)$, *then* $\alpha \in \mathbb{L}(t)$

H.19. *if* $\neg\mathrm{R}_i\alpha \in \mathbb{L}(s)$, *then* $(\exists t \in S)(s \sqcap_i t$ *and* $\neg\alpha \in \mathbb{L}(t))$

H.20. *if* $\mathrm{R}_i\alpha \in \mathbb{L}(s)$, *then* $\alpha \in \mathbb{L}(s)$

H.21. *if* $\mathrm{R}_i\alpha \in \mathbb{L}(s)$ *and* $(s \sqcap_i t)$, *then* $\mathrm{R}_i\alpha \in \mathbb{L}(t)$

H.22. *if* $\mathrm{K}_i\alpha \in \mathbb{L}(s)$, *then* $\mathrm{R}_i\alpha \in \mathbb{L}(s)$

H.23. *if* $\mathrm{AG}\alpha \in \mathbb{L}(s)$, *then* $\mathrm{R}_i\alpha \in \mathbb{L}(s)$

Note that, intuitively, rule H16 corresponds to transitivity and symmetry for the relations associated with the epistemic modality, while rules H14, H15, and H17 correspond, respectively, to seriality, reflexivity and Euclideaness. Rules H19, H20 and H21 correspond, respectively, to seriality, reflexivity, and transitivity for the relation of the "reset" operator. A consequence of such a definition of the Hintikka structure is the following:

LEMMA 2 (HINTIKKA'S LEMMA FOR $\mathcal{L}$). *$\varphi \in \mathcal{L}$ is satisfiable (i.e., φ has a model) if and only if there is a Hintikka structure for φ.*

Proof. It is easy to check that if $M = (S, T, \{\sim_i\}_{i\in\mathcal{A}g}, \mathcal{V}, \{\sqcap_i\}_{i\in\mathcal{A}g})$ is a model for φ then the tuple $H = (S, T, \{\sim_i\}_{i\in\mathcal{A}g}, \mathbb{L}, \{\sqcap_i\}_{i\in\mathcal{A}g})$ with $\mathbb{L}$ defined by $\alpha \in \mathbb{L}(s)$ if $(M, s) \models \alpha$, for all $s \in S$, is a Hintikka structure for φ.

Conversely, suppose that $H = (S, T, \{\sim_i\}_{i\in\mathcal{A}g}, \mathbb{L}, \{\sqcap_i\}_{i\in\mathcal{A}g})$ is a Hintikka structure for φ. Let $M = (S, T, \{\sim'_i\}_{i\in\mathcal{A}g}, \mathcal{V}, \{\sqcap'_i\}_{i\in\mathcal{A}g})$, where each $\sim'_i$ is the reflexive, symmetric and transitive closure of $\sim_i$; each $\sqcap'_i$ is reflexive, serial, and transitive closure of $\sqcap_i$; $\mathcal{V} : S \to 2^{\mathcal{P}\mathcal{V}}$ is defined by $\mathcal{V}(s) = \{p \mid p \in \mathbb{L}(s)\}$. We now show by induction on the structure of formulae that if $\psi \in Sub(\varphi)$, then $\psi \in \mathbb{L}(s)$ implies $M, s \models \psi$ and $\neg\psi \in \mathbb{L}(s)$ implies $M, s \models \neg\psi$.

1. The propositional case as well as all the temporal ones can be shown in similar way to [10].
2. ψ is of the form $\mathrm{K}_i\beta$. Suppose that $\mathrm{K}_i\beta \in \mathbb{L}(s)$. We have to show that $(M, s) \models \mathrm{K}_i\beta$. It suffices to show that $(M, t) \models \beta$ for all states t such that $s \sim'_i t$. But since $\sim'_i$ is the reflexive, symmetric and transitive closure of $\sim_i$, if $s \sim'_i t$ then there exist $n \geq 0$ and a sequence $s_0, \ldots, s_n$ of states such that $s = s_0$, $t = s_n$ and either $s_j \sim_i s_{j+1}$ or $s_{j+1} \sim_i s_j$ for all $j \in \{0, \ldots, n-1\}$. Thus, since $\mathrm{K}_i\beta \in \mathbb{L}(s)$ and $s = s_0$, we have that $\mathrm{K}_i\beta \in \mathbb{L}(s_0)$. Now, since either $s_0 \sim_i s_1$ or $s_1 \sim_i s_0$ by rule $H16$, we have that $\mathrm{K}_i\beta \in \mathbb{L}(s_1)$. So, by the simple induction on $j = 0, \ldots, n-1$ and rule $H16$, we have that $\mathrm{K}_i\beta \in \mathbb{L}(s_n)$. Since $s_n = t$, we have that $\mathrm{K}_i\beta \in \mathbb{L}(t)$. Thus by rule $H15$ we have that $\beta \in \mathbb{L}(t)$. So, by induction we have that $(M, t) \models \beta$. Since the above holds for an arbitrary t such that $s \sim'_i t$ we can conclude that $(M, s) \models \mathrm{K}_i\beta$.
   Now, suppose that $\neg K_i\beta \in \mathbb{L}(s)$. By rule $H14$, there exists a state t such that $s \sim_i t$ and $\neg\beta \in \mathbb{L}(t)$. Since $\sim_i \subseteq \sim'_i$, and given that by induction we have that $(M, t) \models \neg\beta$, we must have $(M, s) \models \neg\mathrm{K}_i\beta$.
3. ψ is of the form $\mathrm{R}_i\beta$. This can be shown similarly to the case above by using $H19 - H21$.

Now we proceed to define the quotient structure for a given model M. The quotient construction depends on an equivalence relation on states of M of a finite index, therefore we first have to provide such a relation. We define it with respect to the *Fischer-Ladner closure* of a formula $\varphi \in \mathcal{L}$ (denoted by $FL(\varphi)$) that is defined by: $FL(\varphi) = CL(\varphi) \cup \{\neg\alpha \mid \alpha \in CL(\varphi)\}$, where $CL(\varphi)$ is the smallest set of formulae that contains φ and satisfies the following conditions: (a) if $\neg\alpha$ or $\mathrm{EX}\alpha$ or $\mathrm{K}_i\alpha$ or $\mathrm{R}_i\alpha \in CL(\varphi)$, then $\alpha \in CL(\varphi)$; (b) if $\alpha \wedge \beta \in CL(\varphi)$, then $\alpha, \beta \in CL(\varphi)$; (c) if $\mathrm{E}(\alpha \mathrm{U} \beta) \in CL(\varphi)$, then $\alpha, \beta, \mathrm{EXE}(\alpha \mathrm{U} \beta) \in CL(\varphi)$; (d) if $\mathrm{A}(\alpha \mathrm{U} \beta) \in CL(\varphi)$, then $\alpha, \beta, \neg\mathrm{EX}(\neg\mathrm{A}(\alpha \mathrm{U} \beta)) \in CL(\varphi)$.

Denote the size of a set A by $Card(A)$. Then, note that for a given formula $\varphi \in \mathcal{L}$, $FL(\varphi)$ is a finite set of formulae. In particular, it can be shown by induction on the length of φ that $Card(FL(\varphi)) \leq 2 \cdot |\varphi|$.

DEFINITION 7 (QUOTIENT STRUCTURE). *Let $\varphi \in \mathcal{L}$, $M = (S, T, \{\sim_i\}_{i\in\mathcal{A}g}, \mathcal{V}, \{\sqcap_i\}_{i\in\mathcal{A}g})$ be a model for φ, and $\leftrightarrow_{FL(\varphi)}$ a binary relation on S defined by $s \leftrightarrow_{FL(\varphi)} s'$ if $(\forall \alpha \in FL(\varphi))((M, s) \models \alpha$ iff $(M, s') \models \alpha)$. Moreover, let $[s] = \{w \in S \mid w \leftrightarrow_{FL(\varphi)} s\}$. The* quotient structure of M by $\leftrightarrow_{FL(\varphi)}$ *is defined as $M_{\leftrightarrow_{FL(\varphi)}} = (S', T', \{\sim'_i\}_{i\in\mathcal{A}g}, \mathbb{L}', \{\sqcap'_i\}_{i\in\mathcal{A}g})$, where $S' = \{[s] \mid s \in S\}$, $T' = \{([s],[s']) \in S' \times S' \mid (\exists w \in [s])(\exists w' \in [s'])$ s.t. $(w, w') \in T\}$, $\sim'_i$ is a transitive closure of $\{([s], [s']) \in S' \times S' \mid (\exists w \in [s])(\exists w' \in [s'])$ s.t. $(w, w') \in \sim_i\}$, $\mathbb{L}' : S' \mapsto 2^{FL(\varphi)}$ is defined by: $\mathbb{L}'([s]) = \{\alpha \in FL(\varphi) \mid$*

$(M,s) \models \alpha\}$, *and* $\sqcap'_i$ *is a transitive closure of* $\{([s],[s']) \in S' \times S' \mid (\exists w \in [s])(\exists w' \in [s'])$ *s.t.* $(w,w') \in \sqcap_i\}$.

Note that the set S' is finite as it is the result of collapsing states satisfying formulae that belong to the finite set $FL(\varphi)$. In fact we have $Card(S') \leq 2^{Card(FL(\varphi))}$. Note also that the relation T' is serial, $\sim'_i$ is reflexive, symmetric and transitive (i.e., it is an equivalence relation), $\sqcap'_i$ is serial, reflexive, and transitive. Further, since $\mathcal{L}$ is an extension of CTL, the resulting quotient structure may not be a model. In particular, similarly to Theorem 3.6 [5], the following lemma holds:

LEMMA 3. *The quotient construction does not preserve satisfiability of formulae of the form* $\mathrm{A}(\alpha \mathrm{U} \beta)$, *where* $\alpha, \beta \in \mathcal{L}$. *In particular, there is a model* M *for* $\mathrm{A}(\top \mathrm{U} p)$ *with* $p \in \mathcal{PV}$ *such that* $M_{\leftrightarrow_{FL(\mathrm{A}(\top \mathrm{U} p))}}$ *is not a model for* $\mathrm{A}(\top \mathrm{U} p)$.

Although $M_{\leftrightarrow_{FL(\varphi)}}$ may not be a model, it satisfies another important property, which allows us to view it as a *pseudo-model*; it can be unwound into a proper model that can be used to show that $\mathcal{L}$ has the FMP property. To make this idea precise, we introduce the following auxiliary definitions.

An *interior* (respectively *frontier*) node of a *directed acyclic graph* (DAG)[2] is one which has (respectively does not have) a successor. The *root* of a DAG is the node (if it exists) from which all other nodes are reachable. A *fragment* $M' = (S', T', \{\sim'_i\}_{i \in \mathcal{A}g}, \mathbb{L}', \{\sqcap'_i\}_{i \in \mathcal{A}g})$ of a Hintikka structure $H = (S, T, \{\sim_i\}_{i \in \mathcal{A}g}, \mathbb{L}, \{\sqcap_i\}_{i \in \mathcal{A}g})$ is a structure such that (S', T') generates a finite DAG, in which the interior nodes satisfy $H1$-$H10$ and $H13$-$H23$, and the frontier nodes satisfy $H1$-$H8$, and $H13$-$H23$. Given $M = (S, T, \{\sim_i\}_{i \in \mathcal{A}g}, \mathbb{L}, \{\sqcap_i\}_{i \in \mathcal{A}g})$ and $M' = (S', T', \{\sim'_i\}_{i \in \mathcal{A}g}, \mathbb{L}', \{\sqcap'_i\}_{i \in \mathcal{A}g})$, we say that M is *contained* in M', and write $M \subseteq M'$, if $S \subseteq S'$, $T = T' \cap (S \times S)$, $\sim_i = \sim'_i \cap (S \times S)$, $\mathbb{L} = \mathbb{L}'|S$, $\sqcap_i = \sqcap'_i \cap (S \times S)$.

DEFINITION 8 (PSEUDO-MODEL). *A* pseudo-model $M = (S, T, \{\sim_i\}_{i \in \mathcal{A}g}, \mathbb{L}, \{\sqcap_i\}_{i \in \mathcal{A}g})$ *for* $\varphi \in \mathcal{L}$ *is defined in the same manner as a Hintikka structure for* φ *in Definition 6, except that condition* $H12$ *is replaced by the following condition* $H12'$: *for all* $s \in S$, *if* $\mathrm{A}(\alpha \mathrm{U} \beta) \in \mathbb{L}(s)$, *then there is a fragment* $(S', T', \{\sim'_i\}_{i \in \mathcal{A}g}, \mathbb{L}', \{\sqcap'_i\}_{i \in \mathcal{A}g}) \subseteq M$ *such that: (a)* (S', T') *generates a DAG with root* s; *(b) for all frontier nodes* $t \in S'$, $\beta \in \mathbb{L}'(t)$; *(c) for all interior nodes* $u \in S'$, $\alpha \in \mathbb{L}'(u)$.

It can be proven that the following lemma holds.

LEMMA 4. *Let* $\varphi \in \mathcal{L}$, $FL(\varphi)$ *be the Fischer-Ladner closure of* φ, $M = (S, T, \{\sim_i\}_{i \in \mathcal{A}g}, \mathcal{V}, \{\sqcap_i\}_{i \in \mathcal{A}g})$ *a model for* φ, *and* $M_{\leftrightarrow_{FL(\varphi)}} = (S', T', \{\sim'_i\}_{i \in \mathcal{A}g}, \mathbb{L}, \{\sqcap'_i\}_{i \in \mathcal{A}g})$ *the quotient structure of* M *by* $\leftrightarrow_{FL(\varphi)}$. *Then,* $M_{\leftrightarrow_{FL(\varphi)}}$ *is a pseudo-model for* φ.

Now we can prove the main claim of the section, i.e., the fact that $\mathcal{L}$ has the finite model property (FMP).

THEOREM 1. *Let* $\varphi \in \mathcal{L}$. *Then the following are equivalent: (1)* φ *is satisfiable; (2) There is a finite pseudo-model for* φ; *(3) There is a Hintikka structure for* φ.

[2]Recall that a directed acyclic graph is a directed graph such that for any node v, there is no nonempty directed path starting and ending on v.

Proof. (1) $\Rightarrow$ (2) follows from Lemma 4. To prove (2) $\Rightarrow$ (3) it is enough to construct a Hintikka structure for φ by "unwinding" the pseudo-model for φ. This can be done in the same way as is described in [5] for the proof of Theorem 4.1. (3) $\Rightarrow$ (1) follows from Lemma 2.

5. DECIDABILITY

Let $FL(\varphi)$ be the Fischer-Ladner closure of $\varphi \in \mathcal{L}$. We define $\Delta \subseteq FL(\varphi)$ to be *maximal* if for every formula $\alpha \in FL(\varphi)$, either $\alpha \in \Delta$ or $\neg\alpha \in \Delta$. Note that maximal sets may be inconsistent.

THEOREM 2. *There is an algorithm for deciding whether any formula of* $\mathcal{L}$ *is satisfiable.*

Proof. Given a formula $\varphi \in \mathcal{L}$, we will construct a finite pseudo-model for φ. We proceed as follows.

1. Build a structure $M^0 = (S^0, T^0, \{\sim_i^0\}_{i \in Ag}, \mathbb{L}^0, \{\sqcap_i^0\}_{i \in Ag})$ for φ as follows: $S^0 = \{\Delta \mid \Delta \subseteq FL(\varphi)$ with Δ maximal and satisfying the rules $H1$-$H8$, $H15$, $H20\}$; $T^0 \subseteq S^0 \times S^0$ is a relation such that $(\Delta_1, \Delta_2) \in T^0$ iff $\neg\mathrm{EX}\alpha \in \Delta_1$ implies that $\neg\alpha \in \Delta_2$ for any $\alpha \in \mathcal{L}$; for each agent $i \in \mathcal{A}g$, $\sim_i^0 \subseteq S^0 \times S^0$ is a relation such that $(\Delta_1, \Delta_2) \in \sim_i^0$ iff $\{\alpha \mid \mathrm{K}_i\alpha \in \Delta_1\} \subseteq \Delta_2$; for all $\Delta \in S^0$, $\mathbb{L}^0(\Delta) = \Delta$; for each agent $i \in \mathcal{A}g$, $\sqcap_i^0 \subseteq S^0 \times S^0$ is the relation such that $(\Delta_1, \Delta_2) \in \sqcap_i^0$ iff $\{\alpha \mid \mathrm{R}_i\alpha \in \Delta_1\} \subseteq \Delta_2$.
   Note that M^0 satisfies conditions $H1$-$H8$, $H15$, $H20$, by construction. It can also be checked that M^0 satisfies $H10$, $H13$, and $H18$ (because of the definition of T^0, $\sim_i^0$ and $\sqcap_i^0$ respectively).
2. Test the structure M^0 for fulfilment of the properties $H9$, $H11$, $H12'$, $H14$, $H16$, $H17$, $H19$, $H21$, $H22$, and $H23$ by repeatedly applying the following deletion rules until no more states in M^0 can be deleted.
   - H9 Delete any state which has no T^0-successors.
   - H11-H12' Delete any state $\Delta_1 \in S^0$ such that either $\mathrm{E}(\alpha \mathrm{U} \beta) \in \Delta_1$ or $\mathrm{A}(\alpha \mathrm{U} \beta) \in \Delta_1$ and there does not exist a fragment $M'' \subseteq M^0$ such that: (i) (S'', T'') is a DAG with root Δ_1; (ii) for all frontier nodes $\Delta_2 \in S''$, $\beta \in \Delta_2$; (iii) for all interior nodes $\Delta_3 \in S''$, $\alpha \in \Delta_3$.
   - H14 Delete any state $\Delta_1 \in S^0$ such that $\neg\mathrm{K}_i\alpha \in \Delta_1$, and Δ_1 does not have any $\sim_i^0$ successor $\Delta_2 \in S^0$ with $\neg\alpha \in \Delta_2$.
   - H16 Delete any state $\Delta_1 \in S^0$ such that $\Delta_1 \sim_i^0 \Delta_2$ and $\mathrm{K}_i\alpha \in \Delta_1$ and $\neg\mathrm{K}_i\alpha \in \Delta_2$.
   - H17 Delete any state $\Delta_1 \in S^0$ such that $\Delta_1 \sim_i^0 \Delta_2$ and $\Delta_1 \sim_i^0 \Delta_3$ and $\alpha \in \Delta_2$ and $\mathrm{K}_i\neg\alpha \in \Delta_3$
   - H19 Delete any state $\Delta_1 \in S^0$ such that $\neg\mathrm{R}_i\alpha \in \Delta_1$, and Δ_1 does not have any $\sqcap_i^0$ successor $\Delta_2 \in S^0$ with $\neg\alpha \in \Delta_2$.
   - H21 Delete any state $\Delta_1 \in S^0$ such that $\Delta_1 \sqcap_i^O \Delta_2$ and $\mathrm{R}_i\alpha \in \Delta_1$ and $\neg\mathrm{R}_i\alpha \in \Delta_2$.
   - H22 Delete any state $\Delta \in S^0$ such that $\mathrm{K}_i\alpha \in \Delta$ and $\mathrm{R}_i\alpha \in \Delta$.
   - H23 Delete any state $\Delta \in S^0$ such that $\mathrm{AG}\alpha \in \Delta$ and $\mathrm{R}_i\alpha \in \Delta$.

   We call the above the *decidability algorithm for* $\mathcal{L}$.
   Note that the algorithm terminates since S^0 is finite.

CLAIM 1. *Let* $M = (S, T, \{\sim_i\}_{i \in Ag}, \mathbb{L}, \{\sqcap_i\}_{i \in Ag})$ *be the resulting structure of the algorithm. The formula* $\varphi \in \mathcal{L}$ *is satisfiable iff* $\varphi \in s$, *for some* $s \in S$.

Proof. In order to show the right-to-left implication of the above property, note that either the resulting structure is a pseudo-model for φ, or $S = \emptyset$ (this can be shown inductively on the structure of the algorithm). So, if $\varphi \in s$ for some $s \in S$, φ is satisfiable by Theorem 1.

Conversely, if φ is satisfiable, then there exists a model M^* such that $M^* \models \varphi$. Let $M^*_{\leftrightarrow_{FL(\varphi)}} = M' = (S', T', \{\sim'_i\}_{i\in Ag}, \mathbb{L}', \{\sqcap'_i\}_{i\in Ag})$ be the quotient structure of M^* by $\leftrightarrow_{FL(\varphi)}$. By Lemma 4 we have that M' is a pseudo-model for φ. By the definition of $\mathbb{L}'$ in the quotient structure, $\mathbb{L}'(s)$ is maximal with respect to $FL(\varphi)$ for all $s \in S'$. Now, let $M'' = (S'', T'', \{\sim''_i\}_{i\in Ag}, \mathbb{L}'', \{\sqcap''_i\}_{i\in Ag})$ be the structure defined by step 1 of the decidability algorithm, and $f : S' \mapsto S''$ a function defined by $f(s) = \mathbb{L}'(s)$. The following hold:

(a) If $(s,t) \in T'$, then $(f(s), f(t)) \in T''$;
Proof (via contradiction): Let $(s,t) \in T'$ and $(f(s), f(t)) \notin T''$. By the definition of T'' we have that $\neg \mathrm{EX}\alpha \in f(s)$ and $\alpha \in f(t)$. Then, by the definition of f, we have that $\neg \mathrm{EX}\alpha \in \mathbb{L}'(s)$ and $\alpha \in \mathbb{L}'(t)$. So, by the definition of $\mathbb{L}'$ in the quotient structure we have that $M^*, s' \models \neg \mathrm{EX}\alpha$ for any M^* state s' in the class s and $M^*, t' \models \alpha$ a for any M^* state t' in the class t. Thus (s', t') is not in T (where T is the relation in M^*) for any such s', t', and thus (s,t), is not in T'.

(b) If $(s,t) \in \sim'_i$, then $(f(s), f(t)) \in \sim''_i$;
Proof (via contradiction): Let $(s,t) \in \sim'_i$ and $(f(s), f(t)) \notin \sim''_i$. By the definition of $\sim''_i$ we have that $\mathrm{K}_i\alpha \in f(s)$ and $\alpha \notin f(t)$. Then, by the definition of f, we have that $\mathrm{K}_i\alpha \in \mathbb{L}'(s)$ and $\alpha \notin \mathbb{L}'(t)$.
So, by the definition of $\mathbb{L}'$ in the quotient structure we have that $M^*, s' \models \mathrm{K}_i\alpha$ for any M^* state s' in the class s and $M^*, t' \models \neg\alpha$ a for any M^* state t' in the class t. Thus (s', t') is not in $\sim_i$ (where $\sim_i$ is the relation in M^*) for any such s', t', and thus (s,t), is not in $\sim'_i$.

(c) if $(s,t) \in \sqcap'_i$, then $(f(s), f(t)) \in \sqcap''_i$;
Proof (via contradiction): Let $(s,t) \in \sqcap'_i$ and $(f(s), f(t)) \notin \sqcap''_i$. Then, by the definition of $\sqcap''_i$ we have that $\mathrm{R}_i\alpha \in f(s)$ and $\alpha \notin f(t)$. By the definition of f, we have that $\mathrm{R}_i\alpha \in \mathbb{L}'(s)$ and $\alpha \notin \mathbb{L}'(t)$. So, by the definition of $\mathbb{L}'$ in the quotient structure we have that $M^*, s' \models \mathrm{R}_i\alpha$ for any M^* state s' in the class s and $M^*, t' \models \neg\alpha$ a for any M^* state t' in the class t. Thus (s', t') is not in $\sqcap_i$ (where $\sqcap_i$ is the relation in M^*) for any such s', t', and thus (s,t), is not in $\sqcap'_i$.

Thus, the image of M' under f is contained in M'', i.e., $M' \subseteq M''$. It remains to show that if $s \in S'$, then $f(s) \in S''$ will not be eliminated in step 2 of the decidability algorithm. This can be checked by induction on the order in which states of S'' are eliminated. For instance, assume that $s \in S'$, and $\mathrm{A}(\alpha \mathrm{U} \beta) \in f(s)$. By the definition of f, we have that $\mathrm{A}(\alpha \mathrm{U} \beta) \in \mathbb{L}'(s)$. Now, since M' is a pseudo-model, by Definition 8 we have that there exists a fragment rooted at s that is contained in M' and it satisfies property $H12'$. Thus, since f preserves the above condition (a), we have that there exists a fragment rooted at $f(s)$ that is contained in M'' and it satisfies property $H12'$. This implies that $f(s) \in S''$ will not be eliminated in $H12'$ step of the decidability algorithm. Other cases can be proven similarly. Therefore, it follows that for some $s \in S$ we have $\varphi \in \mathbb{L}(s)$.

6. A COMPLETE AXIOMATIC SYSTEM

An *axiomatic system* consists of a collection of *axioms schemes* and *inference rules*. An axiom scheme is a rule for generating an infinite number of axioms. An inference rule has the form "from formulae $\varphi_1, \ldots, \varphi_m$ infer formula φ". We say that φ is *provable* (written $\vdash \varphi$) if there is a sequence of formulae ending with φ, such that each formula is either an instance of an axiom, or follows from other provable formulae by applying an inference rule. We say that a formula φ is *consistent* if $\neg\varphi$ is not provable. A finite set $\{\varphi_1, \ldots, \varphi_m\}$ of formulae is *consistent* exactly if and only if the conjunction $\varphi_1 \wedge \ldots \wedge \varphi_m$ of its members is consistent. A set F of formulae is a *maximally consistent set* if it is consistent and for all $\varphi \notin F$, the set $F \cup \{\varphi\}$ is inconsistent. An axiom system is *sound* (respectively *complete*) with respect to the class of models, if $\vdash \varphi$ implies $\models \varphi$ (respectively if $\models \varphi$ implies $\vdash \varphi$).

Let $i \in \mathcal{A}g$. We define the following axiomatic system:

PC. All substitution instances of classical tautologies.

T1. $\mathrm{EX}\top$ T2. $\mathrm{EX}(\alpha \vee \beta) \Leftrightarrow \mathrm{EX}\alpha \vee \mathrm{EX}\beta$

T3. $\mathrm{E}(\alpha \mathrm{U} \beta) \Leftrightarrow \beta \vee (\alpha \wedge \mathrm{EXE}(\alpha \mathrm{U} \beta))$

T4. $\mathrm{A}(\alpha \mathrm{U} \beta) \Leftrightarrow \beta \vee (\alpha \wedge \mathrm{AXA}(\alpha \mathrm{U} \beta))$

$\mathbf{K}_{\mathrm{K_i}}$. $(\mathrm{K}_i\alpha \wedge \mathrm{K}_i(\alpha \Rightarrow \beta)) \Rightarrow \mathrm{K}_i\beta$ $\mathbf{5}_{\mathrm{K_i}}$. $\neg \mathrm{K}_i\alpha \Rightarrow \mathrm{K}_i \neg \mathrm{K}_i\alpha$

$\mathbf{K}_{\mathrm{R_i}}$. $(\mathrm{R}_i\alpha \wedge \mathrm{R}_i(\alpha \Rightarrow \beta)) \Rightarrow \mathrm{R}_i\beta$ $\mathbf{4}_{\mathrm{K_i}}$. $\mathrm{K}_i\alpha \Rightarrow \mathrm{K}_i\mathrm{K}_i\alpha$

$\mathbf{T}_{\mathrm{R_i}}$. $\mathrm{R}_i\alpha \Rightarrow \alpha$ $\mathbf{4}_{\mathrm{R_i}}$. $\mathrm{R}_i\alpha \Rightarrow \mathrm{R}_i\mathrm{R}_i\alpha$

$\mathbf{I_1}$. $\mathrm{K}_i\alpha \Rightarrow \mathrm{R}_i\alpha$ $\mathbf{I_2}$. $\mathrm{AG}\alpha \Rightarrow \mathrm{R}_i\alpha$

R1. From α and $\alpha \Rightarrow \beta$ infer β R2. From α infer $\mathrm{K}_i\alpha$

R3. From $\alpha \Rightarrow \beta$ infer $\mathrm{EX}\alpha \Rightarrow \mathrm{EX}\beta$

R4. From $\gamma \Rightarrow (\neg\beta \wedge \mathrm{EX}\gamma)$ infer $\gamma \Rightarrow \neg \mathrm{A}(\alpha \mathrm{U} \beta)$

R5. From $\gamma \Rightarrow (\neg\beta \wedge \mathrm{AX}(\gamma \vee \neg \mathrm{E}(\alpha \mathrm{U} \beta)))$ infer $\gamma \Rightarrow \neg \mathrm{E}(\alpha \mathrm{U} \beta)$

THEOREM 3. *The axiomatic system above is sound and complete with respect to the class of models of Definition 2, i.e., $\models \varphi$ iff $\vdash \varphi$, for any formula $\varphi \in \mathcal{L}$.*

Proof. This proof is an in [10], we simply extend the claims to cover all cases here.

Soundness can be checked inductively as standard. For completeness, we show that any consistent formula φ is satisfiable. To do this, we first consider the structure $M = (S, T, \{\sim_i\}_{i\in\mathcal{A}g}, \mathbb{L}, \{\sqcap_i\}_{i\in\mathcal{A}g})$ for φ as defined in step 1 of the decidability algorithm. We then execute step 2 of the algorithm, obtaining a pseudo-model for φ. Crucially we show below that if a state $s \in S$ is eliminated at step 2 of the algorithm, then the formula $\psi_s = \bigwedge_{\alpha\in s} \alpha$ is inconsistent. Observe now that for any $\alpha \in FL(\varphi)$ we have $\vdash \alpha \Leftrightarrow \bigvee_{\substack{\{s \mid \alpha \in s \text{ and} \\ \psi_s \text{ is consistent}\}}} \psi_s$. In particular, $\vdash \varphi \Leftrightarrow \bigvee_{\substack{\{s \mid \varphi \in s \text{ and} \\ \psi_s \text{ is consistent}\}}} \psi_s$. Thus, if φ is consistent, then ψ_s is consistent as well for some $s \in S$. It follows by Claim 1 of Theorem 2 that this particular s is present in the pseudo-model resulting from the execution of the algorithm. So, by Theorem 1, φ is satisfiable. Note that pseudo-models share the structural properties of models, i.e., their underlying frames have the same properties.

It remains to show that if a state $s \in S$ is eliminated at step 2 of the algorithm then the formula ψ_s is inconsistent. Before we do it, we need some auxiliary claims.

CLAIM 2. *Let $s \in S$ and $\alpha \in FL(\varphi)$. Then, $\alpha \in s$ iff $\vdash \psi_s \Rightarrow \alpha$.*

Proof. ('if'). Let $\alpha \in s$. By the definition of S, we have that any s in S is maximal. Thus, $\neg\alpha \notin s$. So, $\vdash \psi_s \Rightarrow \alpha$.

('only if'). Let $\vdash \psi_s \Rightarrow \alpha$. So, since s is maximal we have that $\alpha \in s$.

CLAIM 3. *Let $s, t \in S$, both of them be maximal and propositionally consistent, and $s \sim_i t$ (respectively $s \sqcap_i t$). If $\alpha \in t$, then $\neg K_i \neg\alpha \in s$ (respectively $\neg \mathrm{R}_i \neg\alpha \in s$).*

Proof. [By contraposition] Let $\alpha \in t$ and $\neg K_i \neg\alpha \notin s$. Then, since s is maximal we have that $K_i \neg\alpha \in s$. Thus, since $s \sim_i t$, we have that $\neg\alpha \in t$. This contradicts the fact that $\alpha \in t$, since t is propositionally consistent; The same proof applies to R_i.

CLAIM 4. *Let $s \in S$ be a maximal and consistent set of formulae and α such that $\vdash \alpha$. Then $\alpha \in s$.*

Proof. Suppose $\alpha \notin s$ and $\vdash \alpha$. Since s is maximal then $\neg\alpha \in s$. So $\neg\alpha \wedge \psi_s$ is consistent where $\psi_s = \bigwedge_{\beta \in s} \beta$. So by definition of consistency we have that $\nvdash \neg(\neg\alpha \wedge \psi_s)$, so $\nvdash \alpha \vee \neg\psi_s$. But we have $\vdash \alpha \vee \psi_s$, so this is a contradiction.

We now show, by induction on the structure of the decidability algorithm for $\mathcal{L}$, that if a state $s \in S$ is eliminated at step 2 of the decidability algorithm, then $\vdash \neg\psi_s$.

CLAIM 5. *If ψ_s is consistent, then s is not eliminated at step 2 of the decidability algorithm for $\mathcal{L}$.*

Proof.

H9 Let $\mathrm{EX}\alpha \in s$ and ψ_s be consistent. By the same reasoning as in the proof of Claim 4(a) in [5], we conclude that s satisfies $H9$. So s is not eliminated.

H11-H12' Let $\mathrm{E}(\alpha \mathrm{U} \beta) \in s$ (respectively $\mathrm{A}(\alpha \mathrm{U} \beta) \in s$) and suppose s is eliminated at step 2 because $H11$ (respectively $H12'$) is not satisfied. Then ψ_s is inconsistent. The proof showing that fact is the same as the proof of Claim 4(c) (respectively Claim 4(d)) in [5].

H14 Let $\neg \mathrm{K}_i \alpha \in s$ and ψ_s be consistent. Consider the set $S_{\neg\alpha} = \{\neg\alpha\} \cup \{\beta \mid \mathrm{K}_i \beta \in s\}$. We will show that $S_{\neg\alpha}$ is consistent. Suppose that $S_{\neg\alpha}$ is inconsistent. Then, $\vdash \beta_1 \wedge \ldots \wedge \beta_m \Rightarrow \alpha$, where $\beta_j \in \{\beta \mid \mathrm{K}_i \beta \in s\}$ for $j \in \{1, \ldots, m\}$. By rule $R2$ we have $\vdash \mathrm{K}_i((\beta_1 \wedge \ldots \wedge \beta_m) \Rightarrow \alpha)$. By axioms $\mathbf{K}_{\mathrm{K_i}}$ and PC we have $\vdash (\mathrm{K}_i\beta_1 \wedge \ldots \wedge \mathrm{K}_i\beta_m) \Rightarrow \mathrm{K}_i\alpha$. Thus, since each $\mathrm{K}_i\beta_j \in s$ for $j \in \{1, \ldots, m\}$ and s is maximal and propositionally consistent, we have $\mathrm{K}_i\alpha \in s$. This contradicts the fact that ψ_s is consistent. So, $S_{\neg\alpha}$ is consistent.
Now, we have to show that $S_{\neg\alpha}$ can be extended to a maximal set that is in S^0. To show this, as standard, we first construct a sequence $t_0, t_1, \ldots$ of $\mathcal{L}$ - consistent sets as follows. Because $\mathcal{L}$ is a countable language, let $\psi_1, \psi_2, \ldots$ be an enumeration of the formulae in $\mathcal{L}$. Let $t_0 = S_{\neg\alpha} \cup \{CL(\beta) \mid \beta \in S_{\neg\alpha}\}$, and inductively construct the rest of the sequence by taking $t_{i+1} = t_i \cup \{\psi_{i+1}\}$ if this set is $\mathcal{L}$ - consistent and satisfies rules: $H1 - H8$, $H10$, $H13$ $H15$, $H18$, and $H20$, and otherwise by taking $t_{i+1} = t_i$. It is easy to see that each set in the sequence $t_0, t_1, \ldots$ is $\mathcal{L}$ - consistent and satisfies rules: $H1 - H8$, $H10$, $H13$ $H15$, $H18$, and $H20$, Let $t = \bigcup_{i=0}^{\infty} t_i$. Each finite subset of t must be contained in t_j for some j, and thus must be $\mathcal{L}$ - consistent and satisfies the adequate rules. It follows that t itself is $\mathcal{L}$ - consistent and satisfies the adequate rules, so t is in S^0. We claim that t is maximal. For suppose $\psi \in \mathcal{L}$ and $\psi \notin t$. Since ψ is a formula in $\mathcal{L}$, it must appear in our enumeration, say as ψ_k. If $t_k \cup \{\psi_k\}$ were $\mathcal{L}$ - consistent, then our construction would not guarantee that $\psi_k \in t_{k+1}$, and hence $\psi_k \in t$. Because $\psi_k = \psi \notin t$, it follows that $t_k \cup \{\psi\}$ is not $\mathcal{L}$ - consistent. Hence $t \cup \{\psi\}$ is also not $\mathcal{L}$ - consistent. It follows that t is a maximal $\mathcal{L}$ - consistent set.
It follows that $S_{\neg\alpha}$ is contained in some maximal set t that is in S^0. So by the definition of $S_{\neg\alpha}$ we have that $\neg\alpha \in t$, and by the definition of $\sim_i$ in M we have that $s \sim_i t$. Thus, s satisfies $H14$, and it is not eliminated by step H14 of the decidability algorithm.

H16 Suppose that ψ_s is consistent and s is eliminated at step (H16) of the decidability algorithm. Then, we have that $s \sim_i t$, $\mathrm{K}_i\alpha \in s$ and $\neg \mathrm{K}_i\alpha \in t$. Thus, since s and t are maximal and propositionally consistent, by Claim 4 we have that $\neg \mathrm{K}_i \mathrm{K}_i \alpha \in s$. By axiom 4_{K_i} and Claim 5 we have that $\mathrm{K}_i\alpha \Rightarrow \mathrm{K}_i\mathrm{K}_i\alpha \in s$. So, since $\mathrm{K}_i\alpha \in s$ we have that $\mathrm{K}_i\mathrm{K}_i\alpha \in s$. So s is inconsistent. Therefore s cannot be eliminated at step (H16) of the decidability algorithm.

H17 Suppose that s is consistent and it is eliminated at step (H17) of the decidability algorithm. Thus, we have that $s \sim_i t$, $s \sim_i u$, $\alpha \in t$, and $\mathrm{K}_i \neg\alpha \in u$. So, since $s \sim_i t$, $\alpha \in t$, s and t are maximal and propositionally consistent, by Claim 4 we have that $\neg K_i \neg\alpha \in s$. Since s is maximal and consistent, by axiom 5_{K_i} and Claim 5, we have that $\neg \mathrm{K}_i \neg\alpha \Rightarrow \mathrm{K}_i \neg \mathrm{K}_i \neg\alpha \in s$. Therefore, we have that $\mathrm{K}_i \neg \mathrm{K}_i \neg\alpha \in s$. Thus, since $s \sim_i u$, we have that $\neg \mathrm{K}_i \neg\alpha \in u$. But this is a contradiction given that $\mathrm{K}_i \neg\alpha \in u$ and u is propositionally consistent. So s is inconsistent. Therefore s cannot be eliminated at step (H17) of the decidability algorithm.

H19 Let $\neg \mathrm{R}_i\alpha \in s$ and ψ_s be consistent. Consider the set $S_{\neg\alpha} = \{\neg\alpha\} \cup \{\beta \mid \mathrm{R}_i\beta \in s\}$. We will show that $S_{\neg\alpha}$ is consistent. Suppose that $S_{\neg\alpha}$ is inconsistent. Then, $\vdash \beta_1 \wedge \ldots \wedge \beta_m \Rightarrow \alpha$, where $\beta_j \in \{\beta \mid \mathrm{R}_i\beta \in s\}$ for $j \in \{1, \ldots, m\}$. By rule $R2$ we have $\vdash \mathrm{K}_i((\beta_1 \wedge \ldots \wedge \beta_m) \Rightarrow \alpha)$. By axioms $\mathbf{K}_{\mathrm{K_i}}$ and PC we have $\vdash (\mathrm{K}_i\beta_1 \wedge \ldots \wedge \mathrm{K}_i\beta_m) \Rightarrow \mathrm{K}_i\alpha$. By axiom $\mathbf{I_1}$ we have that $\vdash (\mathrm{R}_i\beta_1 \wedge \ldots \wedge \mathrm{R}_i\beta_m) \Rightarrow \mathrm{R}_i\alpha$. Thus, since each $\mathrm{R}_i\beta_j \in s$ for $j \in \{1, \ldots, m\}$ and s is maximal and propositionally consistent, we have $\mathrm{R}_i\alpha \in s$. This contradicts the fact that ψ_s is consistent. So, $S_{\neg\alpha}$ is consistent. Now, we have to show that $S_{\neg\alpha}$ can be extended to a maximal set that is in S^0. This can be done in the same way as in $H14$ case. So, we can now say that $S_{\neg\alpha}$ is contained in some maximal set t and thus $\neg\alpha \in t$. By the definitions of $\sqcap_i$ in M and $S_{\neg\alpha}$ we have that $s \sqcap_i t$. Thus, s satisfies $H19$, and it is not eliminated by step H19 of the decidability algorithm.

H21 Suppose that ψ_s is consistent and s is eliminated at step (H21) of the decidability algorithm. Then, we have that $s \sqcap_i t$, $\mathrm{R}_i\alpha \in s$ and $\neg \mathrm{R}_i\alpha \in t$. Thus, since s and t are maximal and propositionally consistent, by Claim 4 we have that $\neg \mathrm{R}_i \mathrm{R}_i \alpha \in s$. By axiom 4_{R_i} and Claim 5 we have that $\mathrm{R}_i\alpha \Rightarrow \mathrm{R}_i\mathrm{R}_i\alpha \in s$. So, since $\mathrm{R}_i\alpha \in s$ we have that $\mathrm{R}_i\mathrm{R}_i\alpha \in s$. So ψ_s is inconsistent. Therefore s cannot be eliminated at step (H21) of the decidability algorithm.

H22 Suppose that ψ_s is consistent and s is eliminated at step (H22) of the decidability algorithm. Then, we have $\mathrm{K}_i\alpha \in s$ and $\neg \mathrm{R}_i\alpha \in s$. By axiom $\mathbf{I}_1$, since $\mathrm{K}_i\alpha \in s$ we have that $\mathrm{R}_i\alpha \in s$. So ψ_s is inconsistent. Therefore s cannot be eliminated at step (H22) of the decidability algorithm.

H23 Suppose that ψ_s is consistent and s is eliminated at step (H23) of the decidability algorithm. Then, we have $\mathrm{AG}\alpha \in s$ and $\neg \mathrm{R}_i\alpha \in s$. By axiom $\mathbf{I}_2$, since $\mathrm{AG}\alpha \in s$ we have that $\mathrm{R}_i\alpha \in s$. So ψ_s is inconsistent. Therefore s cannot be eliminated at step (H23) of the decidability algorithm.

We have now shown that only states s with ψ_s inconsistent are eliminated. This ends the completeness proof.

Given the above we have the following.

COROLLARY 1. *The axiomatic system above is sound and complete with respect to the class of interpreted systems.*

Proof. We need to show that if ϕ is valid on the class of all interpreted systems it is provable in the axiom system. Suppose ϕ is not provable, then $\neg\phi$ is consistent in the axiom system. By Theorem 3 $\neg\phi$ has a model M. But then by Lemma 1 there is an interpreted system IS whose generated model is isomorphic to M. But then by definition IS does not satisfy $\neg\phi$ and so $\neg\phi$ is not satisfied in the whole class of interpreted systems.

7. CONCLUSIONS

In this paper we have shown completeness and decidability of a logic which includes CTL, epistemic operators and a further modality, defined on the intersection between the temporal and the epistemic relation, expressing knowledge under the assumption of being able to distinguish the future of a time instant.

As it is known, the standard canonical model technique cannot normally be used for providing axiomatisations for modalities that are defined on intersections of relations. Still intersections can be axiomatised by ad-hoc techniques; for instance see distributed knowledge [7, 12] and dynamic knowledge [1]. We were unable to replicate the use of these techniques for the logic in hand and used, instead, the basic schema of [5] used there for axiomatising plain CTL.

Our motivation for studying the reset operator is also rather practical. When studying temporal/epistemic properties of multi-agent systems scenario by using model checking [20, 19] it is often the case that the agents are able to distinguish the past from the present. In such cases, because of efficiency to be gained in model checking optimisations, it is efficient to have epistemic operators that "forget about the past" such as the one presented in this paper. Given we now understand the behaviour of such operator we plan to investigate possible extensions of an open source model checker for knowledge and time such as [14, 9] to evaluate experimentally the performance speed-up it offers.

Acknowledgements: The authors are grateful to one anonymous AAMAS07 reviewer for invaluable technical feedback and suggestions.

8. REFERENCES

[1] P. Balbiani and L. Fariñas del Cerro. Complete axiomatization of a relative modal logic with composition and intersection. *Journal of Applied Non-Classical Logics*, (8):325–335, 1998.

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

[3] E. Clarke and E. Emerson. Design and synthesis of synchronization skeletons for branching-time temporal logic. In *Proceedings of Workshop on Logic of Programs*, volume 131 of *LNCS*, pages 52–71. Springer-Verlag, 1981.

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

[5] E. A. Emerson and J. Y. Halpern. Decision procedures and expressiveness in the temporal logic of branching time. *Journal of Computer and System Sciences*, 30(1):1–24, 1985.

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

[7] 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, 1992.

[8] M. J. Fischer and R. E. Ladner. Propositional dynamic logic of regular programs. *Journal of Computer and System Sciences*, 18(2):194–211, 1979.

[9] P. Gammie and R. van der Meyden. MCK: Model checking the logic of knowledge. In *Proceedings of CAV'04*, volume 3114 of *LNCS*, pages 479–483. Springer-Verlag, 2004.

[10] J. Halpern and Y. Moses. A guide to completeness and complexity for modal logics of knowledge and belief. *Artificial Intelligence*, 54:319–379, 1992.

[11] J. Hintikka. *Knowledge and Belief, An Introduction to the Logic of the Two Notions*. Cornell University Press, Ithaca (NY) and London, 1962.

[12] W. van der Hoek. Systems for knowledge and belief. *Journal of Logic and Computation*, 3(2):173–195, 1993.

[13] W. van der Hoek and M. Wooldridge. Model checking knowledge and time. In *Proceedings of SPIN'02*. 2002.

[14] A. Lomuscio and F. Raimondi. MCMAS: A model checker for multi-agent systems. In *Proceedings of TACAS'06*, volume 3920, pages 450–454. Springer Verlag, 2006.

[15] A. Lomuscio and M. Ryan. On the relation between interpreted systems and Kripke models. In Volume 1441 of *LNAI*. Springer Verlag, 1998.

[16] A. Lomuscio and M. Sergot. Deontic interpreted systems. *Studia Logica*, 75(1):63–92, 2003.

[17] A. Lomuscio and B. Woźna. A complete and decidable axiomatisation for deontic interpreted systems. In *Proceedings of DEON'06*, volume 4048, pages 238 – 254. Springer Berlin / Heidelberg, 2006.

[18] R. van der Meyden and K. Wong. Complete axiomatizations for reasoning about knowledge and branching time. *Studia Logica*, 75(1):93–123, 2003.

[19] W. Penczek and A. Lomuscio. Verifying epistemic properties of multi-agent systems via bounded model checking. *Fundamenta Informaticae*, 55(2):167–185, 2003.

[20] F. Raimondi and A. Lomuscio. Automatic verification of multi-agent systems by model checking via OBDDs. *Journal of Applied Logic*, 2007. To appear.

[21] B. Woźna, A. Lomuscio, and W. Penczek. Bounded model checking for knowledge over real time. In *Proceedings of AAMAS'05*, volume I, pages 165–172. ACM Press, 2005.