A combination of explicit and deductive knowledge with branching time: completeness and decidability results*

Alessio Lomuscio and Bożena Woźna

Department of Computer Science, University College London
Gower Street, London WC1E 6BT, United Kingdom
email: {A.Lomuscio,B.Wozna}@cs.ucl.ac.uk

Abstract. Logics for knowledge and time comprise logic combinations between epistemic logic $S5_n$ for n agents and temporal logic. In this paper we examine a logic combination of Computational Tree Logic and an epistemic logic augmented to include an additional epistemic operator representing explicit knowledge. We show the resulting system enjoys the finite model property, decidability and is finitely axiomatisable. It is further shown that the expressivity of the resulting system enables us to represent a non-standard notion of deductive knowledge which seems promising for applications.

1 Introduction

The use of modal logic has a long tradition in the area of epistemic logic. In its simplest case (dating back to Hintikka [16]) one considers a system of n agents and associates an S5 modality $\mathcal{K}_i$ for every agent i in the system, thereby obtaining the system $S5_n$. In this system, all agents can be said to be logically omniscient and enjoy positive and negative introspection with respect to their knowledge (which will always be true in the real world). While the system $S5_n$ can already be seen as a (trivial) combination, or fusion [20], of $S5$ with itself n times, more interesting combinations have been considered. For example one of the systems presented in [17] is a fusion between systems $S5_n$ for knowledge and system $KD45_n$ for belief plus interaction axioms regulating the relationship for knowledge and belief. The system $S5WD_n$ [22] is an extension of $S5_n$ obtained by adding the "interaction axiom" $\bigwedge_{i=1}^{n-1} \Diamond_i \Box_{i+1} \alpha \Rightarrow \Box_n \Diamond_1 \alpha$. Other examples are discussed in the literature, including [23, 2]. The completeness proofs in these works typically are based on some reasoning on the canonical model [1].

Because of the importance in applications, and in particular in verification, there has been recent growing interest in combining of temporal with epistemic logic. This allows for the representation of concepts such as knowledge of one agent about a changing world, the temporal evolution of the knowledge of agents

* The authors acknowledge support from the EPSRC (grant GR/S49353) and the Nuffield Foundation (grant NAL/690/G).

about the knowledge of others, and other various epistemic properties. A first approach for a logic of knowledge and time was given by Sato in [31]. Subsequently, other logics have been proposed [11, 21, 32, 33]; as it is reported in [14] there are 96 logics of knowledge and time: 48 are based on the linear time logic (LTL) and 48 involve branching time. In particular, a variety of semantical classes (interpreted systems with perfect recall, synchronicity, asynchronicity, no learning, etc.) have been defined and their axiomatisations shown with respect to a temporal and epistemic language [25, 26, 14]. This is of particular relevance for verification of multi-agent systems via model checking, an area that has recently received some attention [13, 18, 27, 30, 34–36].

While these results as a whole seem to constitute a rather mature area of investigation, the underlying assumption there is that an $S5$ modality is an *adequate* operator for knowledge. This is indeed the case in a variety of scenarios (typically in communication protocols) when the properties of interests are best captured by means of an information-theoretic concept. Of interest in these cases is not what an agent explicitly knows but what the specifier of the system can ascribe to the agents given the information they have at their disposal. In other instances $S5$ is not a useful modality to consider, at least on its own, and weaker forms of knowledge are called for.

A variety of weaker variants of the epistemic logic $S5_n$ (most of them inspired by solving what is normally referred to as the "problem of logical omniscience") have been developed [19, 10, 8, 15] over the years. The most relevant for this paper are awareness and explicit knowledge presented in [8, 9]. In this work two new operators: $\mathcal{A}_i$ and $\mathcal{X}_i$ are introduced. The former represents the information an agent has at its disposal; its semantics is not given as in standard Kripke semantics by considering the accessible points on the basis of some accessibility relation, but simply by checking whether the formula of which an agent is aware of is present in its local database, i.e. whether the formula ϕ is $i-$local in the state in question. The latter represents the information an agent explicitly knows, this being interpreted as standard knowledge and awareness of that fact.

The aim of the present paper is two-fold. First, we aim to axiomatise the concept of explicit knowledge when combined with branching time CTL on a standard multi-agent systems semantics, and show the decidability of the resulting system. Second, we argue that the combination of explicit knowledge with branching time not only give rise to interesting axiomatisation problems, but also allows to express some subtle epistemic concepts needed in applications. In particular, it allows to characterise the notion of "deductive knowledge"[1] formalised below.

The rest of the paper is organised as follows. In Section 2 we briefly present the basics of the underlying syntactical and semantical assumptions used in the paper. Section 3 is devoted to the construction of the underlying machinery to prove completeness and decidability, viz Hintikka structures and related concepts. Sections 4 and 5 present the main results of this paper: a decidability

[1] Our use of the term "deductive knowledge" is inspired by [29], although the focus in this paper is different.

result and a completeness proof for the logic. We conclude in Section 6 with some observations on alternative definitions.

2 Temporal Deductive Logic

Logics for time and knowledge provide a fundamental framework for a formal modelling of properties of multi-agent systems (MASs) [9]. In particular, epistemic logics are designed to reason about the knowledge of agents in distributed and multi-agent systems. They are not taken as describing what an agent actually knows, but only what is implicitly represented in his information state, i.e., what logically follows from his actual knowledge. Temporal logics are used to model how a state of agents' knowledge changes over times. Most of them enable to express such temporal concepts as "eventually", "always", "nexttime", "until", "release", etc.

In this section, we present Temporal Deductive Logic (TDL), a combination of the temporal logic CTL [3, 6, 5] with different epistemic notions. In fact, TDL extends standard combinations of branching time epistemic languages by introducing three further epistemic modalities: awareness, explicit knowledge, and deductive knowledge. These new modalities can be combined with Boolean connectives, temporal and standard epistemic operators or nested arbitrarily.

2.1 Syntax

Assume a set of propositional variables $\mathcal{PV}$ also containing the symbol $\top$ standing for *true*, and a set of agents $\mathcal{AG} = \{1, \ldots, n\}$, where $n \in \{1, 2, 3, \ldots\}$. The set $\mathcal{WF}$ of well-formed TDL formulae is defined by the following grammar:

$$\varphi := p \mid \neg\varphi \mid \varphi \vee \varphi \mid \mathrm{E}\bigcirc\varphi \mid \mathrm{E}(\varphi\mathcal{U}\varphi) \mid \mathrm{A}(\varphi\mathcal{U}\varphi) \mid \mathcal{K}_i\varphi \mid \mathcal{A}_i\varphi \mid \mathcal{X}_i\varphi,$$

where $p \in \mathcal{PV}$ and $i \in \mathcal{AG}$.

In the above syntax, a path quantifier, either A ("for all the computation paths") or E ("for some computation paths") is immediately followed by the single one of the usual linear-time operators $\bigcirc$ ("next time") and $\mathcal{U}$ ("until"). This implies that the above grammar defines only one type of temporal formulae, namely *state formulae*; this is a traditional name for the temporal formulae that are interpreted over states only rather than over paths. In fact, the above syntax extends CTL with standard epistemic modality $\mathcal{K}_i$ as well as operators for explicit knowledge ($\mathcal{X}_i$) and awareness ($\mathcal{A}_i$) as in [9]. The formula $\mathcal{X}_i\varphi$ is read as "agent i knows explicitly that φ", the formula $\mathcal{A}_i\varphi$ is read as "agent i is aware of φ", and $\mathcal{K}_i\varphi$ (the standard epistemic modality) is read as "agent i knows (implicitly) that φ" . We shall further use the shortcut $\mathcal{D}_i\varphi$ to represent $\mathrm{E}(\mathcal{K}_i\alpha\mathcal{U}\mathcal{X}_i\alpha)$. The formula $\mathcal{D}_i\varphi$ is read as "agent i may deduce φ (by some computational process)".

The remaining operators can be introduced as abbreviations in the usual way, i.e. $\alpha \wedge \beta \stackrel{def}{=} \neg(\neg\alpha \vee \neg\beta)$, $\alpha \Rightarrow \beta \stackrel{def}{=} \neg\alpha \vee \beta$, $\alpha \Leftrightarrow \beta \stackrel{def}{=} (\alpha \Rightarrow \beta) \wedge (\beta \Rightarrow \alpha)$, $\mathrm{A}\bigcirc\alpha \stackrel{def}{=} \neg\mathrm{E}\bigcirc\neg\alpha$, $\mathrm{E}\Diamond\alpha \stackrel{def}{=} \mathrm{E}(\top\mathcal{U}\alpha)$, $\mathrm{A}\Diamond\alpha \stackrel{def}{=} \mathrm{A}(\top\mathcal{U}\alpha)$, $\mathrm{E}\Box\alpha \stackrel{def}{=} \neg\mathrm{A}\Diamond\neg\alpha$, $\mathrm{A}\Box\alpha \stackrel{def}{=} \neg\mathrm{E}\Diamond\neg\alpha$, $\mathrm{A}(\alpha\mathcal{W}\beta) \stackrel{def}{=} \neg\mathrm{E}(\neg\alpha\mathcal{U}\neg\beta)$, $\mathrm{E}(\alpha\mathcal{W}\beta) \stackrel{def}{=} \neg\mathrm{A}(\neg\alpha\mathcal{U}\neg\beta)$, $\overline{\mathcal{K}}_i\alpha \stackrel{def}{=} \neg\mathcal{K}_i(\neg\alpha)$.

We conclude this section with some essential definitions, used later on in all the proofs.

Let φ and ψ be TDL formulae. ψ is a *sub-formula* of φ if either (a) $\psi = \varphi$; or (b) φ is of the form $\neg\alpha$, $\mathrm{E}\bigcirc\alpha$, $\mathcal{K}_i\alpha$, $\mathcal{X}_i\alpha$, $\mathcal{A}_i\alpha$, or $\mathcal{D}_i\alpha$, and ψ is a sub-formula of α; or (c) φ is of the form $\alpha \vee \beta$, $\mathrm{E}(\alpha\mathcal{U}\beta)$,or $\mathrm{A}(\alpha\mathcal{U}\beta)$ and ψ is a sub-formula of either α or β.

The *length* of φ (denoted by $|\varphi|$) is defined inductively as follows:

- If $\varphi \in \mathcal{PV}$, then $|\varphi| = 1$,
- If φ is of the form $\neg\alpha$, $\mathcal{K}_i\alpha$, $\mathcal{X}_i\alpha$, or $\mathcal{A}_i\alpha$, then $|\varphi| = |\alpha| + 1$,
- If φ is of the form $\mathrm{E}\bigcirc\alpha$, then $|\varphi| = |\alpha| + 2$,
- If φ is of the form $\alpha \vee \beta$ then $|\varphi| = |\alpha| + |\beta| + 1$.
- If φ is of the form $\mathrm{A}(\alpha\mathcal{U}\beta)$ or $\mathrm{E}(\alpha\mathcal{U}\beta)$, then $|\varphi| = |\alpha| + |\beta| + 2$.

2.2 Semantics

Traditionally, the semantics of temporal logics with epistemic operators is defined on interpreted systems, defined in the following way [9].

Definition 1 (Interpreted Systems). *Assume that each agent $i \in \mathcal{AG}$ is associated with a set of local states L_i, and the environment is associated with a set of local states L_e. An* interpreted system *is a tuple $IS = (S, T, \sim_1, \ldots, \sim_n, \mathcal{V})$, where $S \subseteq \prod_{i=1}^{n} L_i \times L_e$ is a set of global states; $T \subseteq S \times S$ is a (serial) temporal relation on S; $\sim_i \subseteq S \times S$ is an (equivalence) epistemic relation for each agent $i \in \mathcal{AG}$ defined by: $s \sim_i s'$ iff $l_i(s') = l_i(s)$, where $l_i : S \mapsto L_i$ is a function returning the local state of agent i from a global state; $\mathcal{V} : S \mapsto 2^{\mathcal{PV}}$ is a valuation function. $\mathcal{V}$ assigns to each state a set of propositional variables that are assumed to be true at that state.*

For more details and further explanations of the notation we refer to [9].

In order to give a semantics to TDL we extend the above definition by means of local awareness functions, used to indicate the facts that agents are aware of. As in [9], we do not attach any fixed interpretation to the notion of awareness, i.e. to be aware may mean "to be able to figure out the truth", "to be able to compute the truth within time T", etc.

Definition 2 (Model). *Given a finite set of agents $\mathcal{AG} = \{1, \ldots, n\}$, a* model *is a tuple $M = (S, T, \sim_1, \ldots, \sim_n, \mathcal{V}, \mathbb{A}_1, \ldots, \mathbb{A}_n)$, where S, T, $\sim_i$, and $\mathcal{V}$ are defined as in the interpreted system above, and $\mathbb{A}_i : L_i \mapsto 2^{\mathcal{WF}}$ is an awareness function assigning a set of formulae to each state, for each $i \in \mathcal{AG}$.*

Intuitively, $\mathbb{A}_i(l_i(s))$ is a set of formulae that the agent i is aware of at state s, i.e. the set of formulae for which the agent can assign a truth value to (unconnected with the global valuation), but he does not necessarily know them. Note that the set of formulae that the agent is aware of can be arbitrary and may not be closed under sub-formulae. Note also that the definition of a model is an extension of the awareness structure, introduced in [9], by a temporal relation. Moreover, it restricts the standard awareness function to be defined over local states only. A review of other restrictions, which can be placed on the set of formulae that an agent may be aware of, and their consequences is given in Section 6.

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 $i \in \mathbb{N}$. For a path $\pi = (s_0, s_1, \ldots)$, we take $\pi(k) = s_k$. 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, β TDL formulae. 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* $p \in \mathcal{V}(s)$, $(M, s) \models \alpha \wedge \beta$ *iff* $(M, s) \models \alpha$ *and* $(M, s) \models \beta$,
$(M, s) \models \neg\alpha$ *iff* $(M, s) \not\models \alpha$, $(M, s) \models \mathrm{E}\bigcirc\alpha$ *iff* $(\exists \pi \in \Pi(s))(M, \pi(1)) \models \alpha$,
$(M, s) \models \mathrm{E}(\alpha \mathcal{U} \beta)$ *iff* $(\exists \pi \in \Pi(s))(\exists m \geq 0)[(M, \pi(m)) \models \beta$ *and* $(\forall j < m)(M, \pi(j)) \models \alpha]$,
$(M, s) \models \mathrm{A}(\alpha \mathcal{U} \beta)$ *iff* $(\forall \pi \in \Pi(s))(\exists m \geq 0)[(M, \pi(m)) \models \beta$ *and* $(\forall j < m)(M, \pi(j)) \models \alpha]$,
$(M, s) \models \mathcal{K}_i\alpha$ *iff* $(\forall s' \in S)$ $(s \sim_i s'$ *implies* $(M, s') \models \alpha)$,
$(M, s) \models \mathcal{A}_i\alpha$ *iff* $\alpha \in \mathbb{A}_i(l_i(s))$,
$(M, s) \models \mathcal{X}_i\alpha$ *iff* $(M, s) \models \mathcal{K}_i\alpha$ *and* $(M, s) \models \mathcal{A}_i\alpha$.

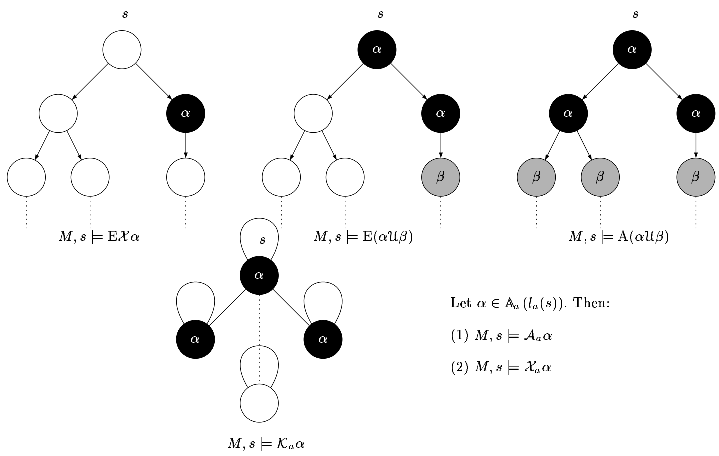

Fig. 1. Examples of TDL formulae which hold in the state s of the model M.

Note that since $\mathcal{D}_i\alpha$ is a shortcut for $\mathrm{E}(\mathcal{K}_i\alpha \mathcal{U} \mathcal{X}_i\alpha)$, as defined on page 3, we have that $(M, s) \models \mathcal{D}_i\alpha$ iff $(M, s) \models \mathrm{E}(\mathcal{K}_i\alpha \mathcal{U} \mathcal{X}_i\alpha)$. Note also that satisfaction for $\mathcal{X}_i$ can be defined simply on $\mathcal{K}_i$ and $\mathbb{A}_i$, but we will find it convenient in the axiomatisation to have a dedicated operator for $\mathbb{A}_i$. This is in line with [9]. Satisfaction for the Boolean and temporal operators as well as the epistemic modalities $\mathcal{K}_i$, $\mathcal{X}_i$, $\mathcal{A}_i$ is standard (see Figure 1 for some examples of TDL formulae holding in a state of a given model). The formula $\mathcal{D}_i\alpha$ holds at state s in a model M if $\mathcal{K}_i\alpha$ holds at s and there exists a path starting at state s such that $\mathcal{X}_i\alpha$ holds in some state on that path and always earlier $\mathcal{K}_i\alpha$ holds. The meaning captured here is the one of potential deduction by the agent: the agent is able to participate in a run (path) of the system under consideration, which leads him to the state, where he knows explicitly the fact in question. Moreover, from an external observer point of view, the agent had enough information from the beginning of such run to deduce the fact. The computation along the path represents, in abstract terms, the deduction performed by the agent to turn implicit into explicit knowledge. Note that the operator $\mathcal{D}_i$ is introduced to account for the process of deduction; other processes resulting in explicit knowledge (discovery, communication, ...) are possible but are not modelled by it. Alternative definitions of deductive knowledge are possible, and we discuss few of them in Section 6.

We conclude this section with some other essential definitions.

Definition 4 (Validity and Satisfiability in a Model). *Let M be a model. A* TDL *formula φ is* valid in M *(written $M \models \varphi$), if $M, s \models \varphi$ for all states $s \in S$. A* TDL *formula φ is* satisfiable in M*, if $M, s \models \varphi$ for some state $s \in S$.*

Definition 5 (Validity and Satisfiability). *A* TDL *formula φ is* valid *(written $\models \varphi$), if φ is valid in all the models M. A* TDL *formula φ is* satisfiable *if it is satisfiable in some model M. In the latter case M is said to be a model for φ.*

3 Finite Model Property for TDL

In this section we prove that the TDL language has the *finite model property* (FMP); a logic has the FMP if any satisfiable formula is also satisfiable in a finite model. The standard way of proving such a result for modal logics is to collapse, according to an equivalence relation of finite index, a possibly infinite model to a finite one (so called quotient structure or filtration), and then to show that the resulting finite structure is still a model for the formula in question. The technique, for example, has been used in [12] to prove that PDL has FMP.

For some logic (in particular for TDL) 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. Therefore, to prove FMP for TDL, we follow [7], where a combination of the filtration and unwinding technique has been applied to prove FMP for CTL[2]. We begin with providing definitions of two auxiliary structures: a *Hintikka structure* for a given TDL formula, and the *quotient structure* for a given model, also called - in the classical modal logic - *filtration.*

Definition 6 (Hintikka structure). *Let φ be a* TDL *formula, and $\mathcal{AG} = \{1, \ldots, n\}$ a set of agents. A* Hintikka structure *for φ is a tuple $HS = (S, T, \sim_1, \ldots, \sim_n, \mathbb{L}, \mathbb{A}_1, \ldots, \mathbb{A}_n)$ such that the elements S, T, $\sim_i$, and $\mathbb{A}_i$, for $i \in \mathcal{AG}$, are defined as in Definition 2, and $\mathbb{L} : S \mapsto 2^{\mathcal{WF}}$ 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:*

[2] Here, we would like to emphasise that despite the fact that in [7] a logic CTL* is introduced as a underlying formalism, the decidability and completeness results are given for CTL only.

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* $(\alpha \vee \beta) \in \mathbb{L}(s)$, *then* $\alpha \in \mathbb{L}(s)$ *or* $\beta \in \mathbb{L}(s)$
H.4. *if* $\neg(\alpha \vee \beta) \in \mathbb{L}(s)$, *then* $\neg\alpha \in \mathbb{L}(s)$ *and* $\neg\beta \in \mathbb{L}(s)$
H.5. *if* $\mathrm{E}(\alpha\mathcal{U}\beta) \in \mathbb{L}(s)$, *then* $\beta \in \mathbb{L}(s)$ *or* $\alpha \wedge \mathrm{E}\bigcirc\mathrm{E}(\alpha\mathcal{U}\beta) \in \mathbb{L}(s)$
H.6. *if* $\neg\mathrm{E}(\alpha\mathcal{U}\beta) \in \mathbb{L}(s)$, *then* $\neg\beta \wedge \neg\alpha \in \mathbb{L}(s)$ *or* $\neg\beta \wedge \neg\mathrm{E}\bigcirc\mathrm{E}(\alpha\mathcal{U}\beta) \in \mathbb{L}(s)$
H.7. *if* $\mathrm{A}(\alpha\mathcal{U}\beta) \in \mathbb{L}(s)$, *then* $\beta \in \mathbb{L}(s)$ *or* $\alpha \wedge \neg\mathrm{E}\bigcirc(\neg\mathrm{A}(\alpha\mathcal{U}\beta)) \in \mathbb{L}(s)$
H.8. *if* $\neg\mathrm{A}(\alpha\mathcal{U}\beta) \in \mathbb{L}(s)$, *then* $\neg\beta \wedge \neg\alpha \in \mathbb{L}(s)$ *or* $\neg\beta \wedge \mathrm{E}\bigcirc(\neg\mathrm{A}(\alpha\mathcal{U}\beta)) \in \mathbb{L}(s)$
H.9. *if* $\mathrm{E}\bigcirc\alpha \in \mathbb{L}(s)$, *then* $(\exists t \in S)((s,t) \in T$ *and* $\alpha \in \mathbb{L}(t))$
H.10. *if* $\neg\mathrm{E}\bigcirc\alpha \in \mathbb{L}(s)$, *then* $(\forall t \in S)((s,t) \in T$ *implies* $\neg\alpha \in \mathbb{L}(t))$
H.11. *if* $\mathrm{E}(\alpha\mathcal{U}\beta) \in \mathbb{L}(s)$, *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* $\mathrm{A}(\alpha\mathcal{U}\beta) \in \mathbb{L}(s)$, *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* $\mathcal{K}_i\alpha \in \mathbb{L}(s)$, *then* $\alpha \in \mathbb{L}(s)$
H.14. *if* $\mathcal{K}_i\alpha \in \mathbb{L}(s)$, *then* $(\forall t \in S)(s \sim_i t$ *implies* $\alpha \in \mathbb{L}(t))$
H.15. *if* $\neg\mathcal{K}_i\alpha \in \mathbb{L}(s)$, *then* $(\exists t \in S)(s \sim_i t$ *and* $\neg\alpha \in \mathbb{L}(t))$
H.16. *if* $\mathcal{X}_i\alpha \in \mathbb{L}(s)$, *then* $\mathcal{K}_i\alpha \in \mathbb{L}(s)$ *and* $\mathcal{A}_i(\alpha) \in \mathbb{L}(s)$
H.17. *if* $\neg\mathcal{X}_i\alpha \in \mathbb{L}(s)$, *then* $\neg\mathcal{K}_i\alpha \in \mathbb{L}(s)$ *or* $\neg\mathcal{A}_i\alpha \in \mathbb{L}(s)$
H.18. *if* $s \sim_i t$ *and* $s \sim_i u$ *and* $\mathcal{K}_i\alpha \in \mathbb{L}(t)$, *then* $\alpha \in \mathbb{L}(u)$
H.19. *if* $\mathcal{A}_i\alpha \in \mathbb{L}(s)$, *then* $\alpha \in \mathbb{A}_i(l_i(s))$
H.20. *if* $\neg\mathcal{A}_i\alpha \in \mathbb{L}(s)$, *then* $\alpha \notin \mathbb{A}_i(l_i(s))$

Note that the Hintikka structure differs from a model in that the assignment $\mathbb{L}$ is not restricted to propositional variables, nor it is required to always contain p or $\neg p$ for every $p \in \mathcal{PV}$. Further, the labelling rules are of the form "if" and not "if and only if". They provide the requirements that must be satisfied by a valid labelling, but they do not require that the formulae belonging to $\mathbb{L}(s)$ form a maximal set of formulae, for any $s \in S$. This means that there are formulae that are satisfied in a given state but they are not included in the label of that state. As usual, we call the rules $H1$-$H8$, $H13$, and $H16$ *propositional consistency rules*, the rules $H9$, $H10$, $H14$, $H15$, $H17$, and $H18 - H20$ *local consistency rules*, and the rules $H11$ and $H12$ the *eventuality rules*.

A consequence of such a definition of the Hintikka structure is the following:

Lemma 1 (Hintikka's Lemma for TDL). *A* TDL *formula* φ *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 any model $M = (S, T, \sim_1, \ldots, \sim_n, \mathcal{V}, \mathbb{A}_1, \ldots, \mathbb{A}_n)$ for φ is a Hintikka structure for φ, when we extend $\mathcal{V}$ to cover all formulae which are true in a state, i.e., in M we replace $\mathcal{V}$ by $\mathbb{L}$ that is defined as: $\alpha \in \mathbb{L}(s)$ if $(M, s) \models \alpha$, for all $s \in S$.

Conversely, any Hintikka structure $HS = (S, T, \sim_1, \ldots, \sim_n, \mathbb{L}, \mathbb{A}_1, \ldots, \mathbb{A}_n)$ for φ can be restricted to form a model for φ. Namely, it is enough to restrict $\mathbb{L}$ to propositional variables only, and require that for every propositional variable p appearing in φ and for all $s \in S$ either $p \in \mathbb{L}(s)$ or $\neg p \in \mathbb{L}(s)$. □

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 will define it with respect to the *Fischer-Ladner closure* of a TDL formula φ (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 $\neg\alpha \in CL(\varphi)$, then $\alpha \in CL(\varphi)$,
(b). if $\alpha \vee \beta \in CL(\varphi)$, then $\alpha, \beta \in CL(\varphi)$,
(c). if $\mathrm{E}(\alpha \mathcal{U} \beta) \in CL(\varphi)$, then $\alpha, \beta, \mathrm{E}\bigcirc\mathrm{E}(\alpha\mathcal{U}\beta) \in CL(\varphi)$,
(d). if $\mathrm{A}(\alpha \mathcal{U} \beta) \in CL(\varphi)$, then $\alpha, \beta, \mathrm{A}\bigcirc\mathrm{A}(\alpha\mathcal{U}\beta) \in CL(\varphi)$,
(e). if $\mathrm{E}\bigcirc\alpha \in CL(\varphi)$, then $\alpha \in CL(\varphi)$,
(f). if $\mathcal{K}_i\alpha \in CL(\varphi)$, then $\alpha \in CL(\varphi)$,
(g). if $\mathcal{A}_i\alpha \in CL(\varphi)$, then $\alpha \in CL(\varphi)$,
(h). if $\mathcal{X}_i\alpha \in CL(\varphi)$, then $\mathcal{K}_i\alpha \in CL(\varphi)$ and $\mathcal{A}_i\alpha \in CL(\varphi)$,

Note that for a given TDL formula φ, $FL(\varphi)$ is a finite set of formulae, as the following lemma shows; hereafter, the size of a set A (denoted by $Card(A)$) is the cardinality of A.

Lemma 2. *Let φ be a* TDL *formula. Then,* $Card(FL(\varphi)) \leq 2(|\varphi| + 3)$.
Proof. Straightforward by induction on the length of φ. □

Definition 7 (Fischer-Ladner's equivalence relation). *Let φ be a* TDL *formula and $M = (S, T, \sim_1, \ldots, \sim_n, \mathcal{V}, \mathbb{A}_1, \ldots, \mathbb{A}_n)$ a model for φ. The relation $\leftrightarrow_{FL(\varphi)}$ on a set of states S is defined as follows:*

$$s \leftrightarrow_{FL(\varphi)} s' \text{ if } (\forall\alpha \in FL(\varphi))((M, s) \models \alpha \text{ iff } (M, s') \models \alpha)$$

By $[s]$ we denote the set $\{w \in S \mid w \leftrightarrow_{FL(\varphi)} s\}$.

Observe that $\leftrightarrow_{FL(\varphi)}$ is indeed an equivalence relation, so using it we can define the quotient structure for a given TDL model.

Definition 8 (Quotient structure). *Let φ be a* TDL *formula, $M = (S, T, \sim_1, \ldots, \sim_n, \mathcal{V}, \mathbb{A}_1, \ldots, \mathbb{A}_n)$ a model for φ, and $\leftrightarrow_{FL(\varphi)}$ a Fischer-Ladner's equivalence relation. The* quotient structure of M by $\leftrightarrow_{FL(\varphi)}$ *is the structure $M_{\leftrightarrow_{FL(\varphi)}} = (S', T', \sim'_1, \ldots, \sim'_n, \mathbb{L}', \mathbb{A}'_1, \ldots, \mathbb{A}'_n)$, 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.* $(w, w') \in T\}$,
- *for each agent i,* $\sim'_i = \{([s], [s']) \in S' \times S' \mid (\exists w \in [s])(\exists w' \in [s'])$ *s.t.* $(w, w') \in \sim_i\}$,
- $\mathbb{L}' : S' \mapsto 2^{FL(\varphi)}$ *is a function defined by:* $\mathbb{L}'([s]) = \{\alpha \in FL(\varphi) \mid (M, s) \models \alpha\}$,
- $\mathbb{A}'_i : l'_i(S') \mapsto 2^{FL(\varphi)}$ *is a function defined by:* $\mathbb{A}'_i(l'_i([s])) = \bigcup_{t \in [s]} \mathbb{A}_i(l_i(t))$, *for each agent i. $l'_i : S' \mapsto 2^{L_i}$ is a function defined by:* $l'_i([s]) = \bigcup_{t \in [s]} l_i(t)$. *It returns a set of local states for agent $i \in \mathcal{AG}$ for a given set of global states.*

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 $\mathbb{A}'_i$ is well defined.

The quotient structure cannot be directly used to show that TDL has FMP. This is because the resulting quotient structure may not be a model, as the following lemma shows.

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

Proof.[Idea] Consider the following model $M = (S, T, \sim, \mathcal{V}, \mathbb{A})$ for $\mathrm{A}(\top\mathcal{U}p)$, where $S = \{s_0, s_1, \dots, \}$, $T = \{(s_0, s_0)\} \cup \{(s_i, s_{i-1}) \mid i > 0\}$, $\sim = \{(s_i, s_i) \mid i \geq 0\}$, $p \in \mathcal{V}(s_0)$ and $p \notin \mathcal{V}(s_i)$ for all $i > 0$, and $\mathbb{A}(s_i) = \emptyset$ for all $i \geq 0$. Observe that in the quotient structure of M two distinct states s_i and s_j for $i, j > 0$ will be identified, resulting in a cycle in the quotient structure, along which p is always false. Hence $\mathrm{A}(\top\mathcal{U}p)$ does not hold along the cycle. □

Although the quotient structure of a given model M by $\leftrightarrow_{FL(\varphi)}$ may not be a model, it satisfies another important property, which allows us to view it as a *pseudo-model*, a definition of which is defined later on; it can be unwound into a proper model that can be used to show that the TDL language 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)[3] 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* $M = (S', T', \sim'_1, \dots, \sim'_n, \mathbb{L}', \mathbb{A}'_1, \dots, \mathbb{A}'_n)$ of a Hintikka structure $HS = (S, T, \sim_1, \dots, \sim_n, \mathbb{L}, \mathbb{A}_1, \dots, \mathbb{A}_n)$ is a structure such that (S', T') generates a finite DAG, in which the interior nodes satisfy $H1$-$H10$ and $H13$-$H20$, and the frontier nodes satisfy $H1$-$H8$, and $H13$, $H16$-$H20$. Given $M = (S, T, \sim_1, \dots, \sim_n, \mathbb{L}, \mathbb{A}_1, \dots, \mathbb{A}_n)$ and $M' = (S', T', \sim'_1, \dots, \sim'_n, \mathbb{L}', \mathbb{A}'_1, \dots, \mathbb{A}'_n)$, we say that M is *contained* in M', and write $M \subseteq M'$, if $S \subseteq S'$, $T = T' \cap (S \times S)$, $\sim_i = \sim'_i \cap (S \times S)$, $\mathbb{L} = \mathbb{L}'|S$, $\mathbb{A}_i = \mathbb{A}'_i|L_i$.

Definition 9 (Pseudo-model). *Let* φ *be a* TDL *formula. A* pseudo-model $M = (S, T, \sim_1, \dots, \sim_n, \mathbb{L}, \mathbb{A}_1, \dots, \mathbb{A}_n)$ *for* φ *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* $H'12$*:* $(\forall s \in S)$ *if* $\mathrm{A}(\alpha\mathcal{U}\beta) \in \mathbb{L}(s)$*, then there is a fragment* $(S', T', \sim'_1, \dots, \sim'_n, \mathbb{L}', \mathbb{A}'_1, \dots, \mathbb{A}'_n) \subseteq M$ *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* $u \in S'$*,* $\alpha \in \mathbb{L}'(u)$*.*

Now we can prove the main claim of the section, i.e., the fact that TDL has the finite model property.

Theorem 1 (FMP for TDL**).** *Let* φ *be a* TDL *formula. 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

[3] 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, presented below. 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 [7] for the proof of Theorem 4.1. (3) $\Rightarrow$ (1) follows from Lemma 1. $\square$

Lemma 4. *Let φ be a* TDL *formula, $FL(\varphi)$ the Fischer-Ladner closure of φ, $M = (S, T, \sim_1, \dots, \sim_n, \mathcal{V}, \mathbb{A}_1, \dots, \mathbb{A}_n)$ a model for φ, and $M_{\leftrightarrow_{FL(\varphi)}} = (S', T', \sim'_1, \dots, \sim'_n, \mathbb{L}, \mathbb{A}'_1, \dots, \mathbb{A}'_n)$ the quotient structure of M by $\leftrightarrow_{FL(\varphi)}$. Then, $M_{\leftrightarrow_{FL(\varphi)}}$ is a pseudo-model for φ.*

Proof. The proof for the temporal part of TDL follows immediately from Lemma 3.8 in [7]. Consider now φ to be of the following forms: $\neg\mathcal{K}_i\alpha$, $\mathcal{X}_i\alpha$, and $\mathcal{A}_i\alpha$. The other cases can be proven in a similar way.

1. $\varphi = \neg\mathcal{K}_i\alpha$. Let $(M, s) \models \neg\mathcal{K}_i\alpha$, and $\neg\mathcal{K}_i\alpha \in \mathbb{L}([s])$. By the definition of $\models$, we have that $(\exists t \in S)$ such that $s \sim_i t$ and $(M, t) \models \neg\alpha$. Thus, by the definitions of $\leftrightarrow_{FL(\varphi)}$ and $\mathbb{L}$, we have that $\neg\alpha \in \mathbb{L}([t])$. Therefore, by the definition of $\sim'_i$ we conclude that $\exists[t] \in S'$ such that $[s] \sim'_i [t]$ and $\neg\alpha \in \mathbb{L}([t])$. So, condition $H15$ is fulfilled.
2. $\varphi = \mathcal{X}_i\alpha$. Let $(M, s) \models \mathcal{X}_i\alpha$, and $\mathcal{X}_i\alpha \in \mathbb{L}([s])$. By the definition of $\models$, we have that $(M, s) \models \mathcal{K}_i\alpha$ and $(M, s) \models \mathcal{A}_i\alpha$. By the definition of $\leftrightarrow_{FL(\varphi)}$ and $\mathbb{L}$, we have that $\mathcal{K}_i\alpha \in \mathbb{L}([s])$ and $\mathcal{A}_i\alpha \in \mathbb{L}([s])$. So, condition $H16$ is fulfilled.
3. $\varphi = \mathcal{A}_i\alpha$. Let $(M, s) \models \mathcal{A}_i\alpha$, and $\mathcal{A}_i\alpha \in \mathbb{L}([s])$. By the definition of $\models$, we have that $\alpha \in \mathbb{A}_i(l_i(s))$. Since $\mathbb{A}_i(l_i(s)) \subseteq \mathbb{A}'_i(l'_i([s]))$, we have that $\alpha \in \mathbb{A}'_i(l'_i([s]))$. So, condition $H19$ is fulfilled.

$\square$

In the subsequent sections we will present an algorithm for deciding satisfiability and an axiomatic system for proving all valid formulae in TDL.

4 Decidability for TDL

Let φ be a TDL formula, and $FL(\varphi)$ the Fischer-Ladner closure of φ. We define $\Delta \subseteq FL(\varphi)$ to be *maximal* if for every formula $\alpha \in FL(\varphi)$, either $\alpha \in \Delta$ or $\neg\alpha \in \Delta$.

Theorem 2. *There is an algorithm for deciding whether any* TDL *formula is satisfiable.*

Proof. Given a TDL formula φ, we construct a finite pseudo-model for φ. We proceed as follows.

1. Build a structure $M^0 = (S^0, T^0, \sim^0_i, \dots, \sim^0_n, \mathbb{L}^0, \mathbb{A}^0_1, \dots, \mathbb{A}^0_n)$ for φ in the following way:
   - $S^0 = \{\Delta \subseteq FL(\varphi) \mid \Delta$ maximal and satisfying all the propositional consistency rules$\}$;
   - $T^0 \subseteq S^0 \times S^0$ is the relation such that $(\Delta_1, \Delta_2) \in T^0$ iff $\neg\mathrm{E}\bigcirc\alpha \in \Delta_1$ implies that $\neg\alpha \in \Delta_2$;
   - for each agent $i \in \mathcal{AG}$, $\sim^0_i \subseteq S^0 \times S^0$ is the relation such that $(\Delta_1, \Delta_2) \in \sim_i$ iff $\{\alpha \mid \mathcal{K}_i\alpha \in \Delta_1\} \subseteq \Delta_2$;

- $\mathbb{L}^0(\Delta) = \Delta$;
- assume that for each agent $i \in \mathcal{AG}$ the set of local states L_i is equal to S^0. Then, $\mathbb{A}_i^0(\Delta) = \{\alpha \mid \mathcal{A}_i\alpha \in \Delta\}$ for each agent $i \in \mathcal{AG}$.

It is easy to observe that M^0, as constructed above, satisfies all the propositional consistency properties; property $H10$ (because of the definition of T^0), property $H14$ (because of the definition of $\sim_i^0$), and properties $H19$ and $H20$ (because of the definition of $\mathbb{A}_i^0$).
2. Test the above structure M^0 for fulfilment of the properties $H9$, $H11$, $H'12$, $H15$, $H17$, and $H18$ by repeatedly applying the following deletion rules until no more states in the pseudo-model can be deleted.
   (a) Delete any state which has no T^0-successors.
   (b) Delete any state $\Delta_1 \in S^0$ such that $\mathrm{E}(\alpha\mathcal{U}\beta) \in \Delta_1$ (respectively $\mathrm{A}(\alpha\mathcal{U}\beta) \in \Delta_1$) 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 $\Delta_3 \in S''$, $\alpha \in \Delta_3$.
   (c) Delete any state $\Delta_1 \in S^0$ such that $\neg\mathcal{K}_i\alpha \in \Delta_1$, and Δ_1 does not have any $\sim_i^0$ successor $\Delta_2 \in S^0$ with $\neg\alpha \in \Delta_2$.
   (d) Delete any state $\Delta \in S^0$ such that $\neg\mathcal{X}_i\alpha \in \Delta$ and $\mathcal{K}_i\alpha \in \Delta$ and $\alpha \in \mathbb{A}_i^0(\Delta)$.
   (e) 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 $\mathcal{K}_i\neg\alpha \in \Delta_3$

We call the algorithm above the *decidability algorithm for* TDL.

Claim (1). The decidability algorithm for TDL terminates.
Proof. The termination is obvious given that the initial set S^0 is finite.

Claim (2). Let $M = (S, T, \sim_1, \ldots, \sim_n, \mathbb{L}, \mathbb{A}_1, \ldots, \mathbb{A}_n)$ be the resulting structure of the algorithm. The formula $\varphi \in \mathcal{WF}$ is satisfiable iff $\varphi \in s$, for some $s \in S$.
Proof. In order to show the part right-to-left of the satisfaction 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). Any pseudo-model for φ can be extended to a model for φ (see the proof of Theorem 1).

Conversely, if φ is satisfiable, then there exists a model M^* such that $M^* \models \varphi$. Let $M_{\leftrightarrow_{FL(\varphi)}} = M' = (S', T', \sim'_1, \ldots, \sim'_n, \mathbb{L}', \mathbb{A}'_1, \ldots, \mathbb{A}'_n)$ be the quotient structure of M^* by $\leftrightarrow_{FL(\varphi)}$. By Theorem 1 we have that M' is a pseudo-model for φ. So, $\mathbb{L}'$ satisfies all the propositional consistency rules, the local consistency rules, and properties $H11$ and $H'12$. Moreover, 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 $M'' = (S'', T'', \sim''_1, \ldots, \sim''_n, \mathbb{L}'', \mathbb{A}''_1, \ldots, \mathbb{A}''_n)$ be a 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 conditions hold:
1. if $(s, t) \in T'$, then $(f(s), f(t)) \in T''$;
   Proof (via contradiction): Let $(s, t) \in T'$ and $(f(s), f(t)) \notin T''$. Then, by the definition of T'' we have that $\neg\mathrm{E}\bigcirc\alpha \in f(s)$ and $\alpha \in f(t)$. By the definition of f, we have that $\neg\mathrm{E}\bigcirc\alpha \in \mathbb{L}'(s)$ and $\alpha \in \mathbb{L}'(t)$. So, by the definition of $\mathbb{L}'$ in the quotient structure we have that $M^*, s \models \neg\mathrm{E}\bigcirc\alpha$ and $M^*, t \models \alpha$, which contradict the fact that $(s, t) \in T'$.

2. if $(s,t) \in \sim_i'$, then $(f(s), f(t)) \in \sim_i''$;
   Proof (via contradiction): Let $(s,t) \in \sim_i'$ and $(f(s), f(t)) \notin \sim_i''$. Then, by the definition of $\sim_i''$ we have that $\mathcal{K}_i\alpha \in f(s)$ and $\alpha \notin f(t)$. By the definition of f, we have that $\mathcal{K}_i\alpha \in \mathbb{L}'(s)$ and $\alpha \notin \mathbb{L}'(t)$. So, by the definition of $\mathbb{L}'$ in the quotient structure we have that $M^*, s \models \mathcal{K}_i\alpha$ and $M^*, t \models \neg\alpha$, which contradict the fact that $(s,t) \in \sim_i'$.

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 $\mathrm{A}(\alpha\mathcal{U}\beta) \in f(s)$. By the definition of f, we have that $\mathrm{A}(\alpha\mathcal{U}\beta) \in \mathbb{L}'(s)$. Now, since M' is a pseudo-model, by Definition 9 we have that there exists a fragment rooted at s that is contained in M' and it satisfies property $H'12$. Thus, since f preserves the above condition (1), we have that there exists a fragment rooted at $f(s)$ that is contained in M'' and it satisfies property $H'12$. This implies that $f(s) \in S''$ will not be eliminated in step 2(b) of the decidability algorithm. Other cases can be proven similarly. Therefore, it follows that for some $s \in S$ we have $\varphi \in \mathbb{L}(s)$.

□

5 A Complete Axiomatic System for TDL

Recall, 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, i.e. formulae that are universally valid. 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 $i \in \{1, \ldots, n\}$. Consider system TDL as defined below:

PC. All substitution instances of classical tautologies.
T1. $\mathrm{E}\bigcirc\top$
T2. $\mathrm{E}\bigcirc(\alpha \vee \beta) \Leftrightarrow \mathrm{E}\bigcirc\alpha \vee \mathrm{E}\bigcirc\beta$
T3. $\mathrm{E}(\alpha\mathcal{U}\beta) \Leftrightarrow \beta \vee (\alpha \wedge \mathrm{E}\bigcirc\mathrm{E}(\alpha\mathcal{U}\beta))$
T4. $\mathrm{A}(\alpha\mathcal{U}\beta) \Leftrightarrow \beta \vee (\alpha \wedge \mathrm{A}\bigcirc\mathrm{A}(\alpha\mathcal{U}\beta))$
K1. $(\mathcal{K}_i\alpha \wedge \mathcal{K}_i(\alpha \Rightarrow \beta)) \Rightarrow \mathcal{K}_i\beta$
K2. $\mathcal{K}_i\alpha \Rightarrow \alpha$
K3. $\neg\mathcal{K}_i\alpha \Rightarrow \mathcal{K}_i\neg\mathcal{K}_i\alpha$
X1. $\mathcal{X}_i\alpha \Leftrightarrow \mathcal{K}_i\alpha \wedge \mathcal{A}_i\alpha$
A1. $\mathcal{A}_i\alpha \Rightarrow \mathcal{K}_i\mathcal{A}_i\alpha$
A2. $\neg\mathcal{A}_i\alpha \Rightarrow \mathcal{K}_i\neg\mathcal{A}_i\alpha$
R1. From α and $\alpha \Rightarrow \beta$ infer β (Modus Ponens)
R2. From α infer $\mathcal{K}_i\alpha$, $i = 1, \ldots, n$ (Knowledge Generalisation)
R3. From $\alpha \Rightarrow \beta$ infer $\mathrm{E}\bigcirc\alpha \Rightarrow \mathrm{E}\bigcirc\beta$
R4. From $\gamma \Rightarrow (\neg\beta \wedge \mathrm{E}\bigcirc\gamma)$ infer $\gamma \Rightarrow \neg\mathrm{A}(\alpha\mathcal{U}\beta)$
R5. From $\gamma \Rightarrow (\neg\beta \wedge \mathrm{A}\bigcirc(\gamma \vee \neg\mathrm{E}(\alpha\mathcal{U}\beta)))$ infer $\gamma \Rightarrow \neg\mathrm{E}(\alpha\mathcal{U}\beta)$

Theorem 3. *The system* TDL *is sound and complete with respect to the class of models of Definition 1, i.e.* $\models \varphi$ *iff* $\vdash \varphi$*, for any formula* $\varphi \in \mathcal{WF}$.
Proof. Soundness can be checked inductively as standard. For completeness, it is sufficient to show that any consistent formula is satisfiable. To do this, we first construct the structure $M = (S, T, \sim_1, \ldots, \sim_n, \mathbb{L}, \mathbb{A}_1, \ldots, \mathbb{A}_n)$ for φ just as in step 1 of the decidability algorithm for TDL. 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 $\vdash \alpha \Leftrightarrow \bigvee_{\{s \mid \alpha \in s \text{ and } \psi_s \text{ is consistent}\}} \psi_s$. Thus, in particular, we have $\vdash \varphi \Leftrightarrow \bigvee_{\{s \mid \varphi \in s \text{ and } \psi_s \text{ is consistent}\}} \psi_s$. Thus, if φ is consistent, then some ψ_s is consistent as well for some $s \in S$. It follows by Claim 2 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 (3). 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 (4). Let $s, t \in S$, both of them be maximal and propositionally consistent, and $s \sim_i t$. If $\alpha \in t$, then $\neg K_i \neg \alpha \in s$.
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. □

Claim (5). Let $s \in S$ be a maximal and consistent set of formulas 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 ψ_s where $\psi_s \in s$. 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 TDL, that if a state $s \in S$ is eliminated, then $\vdash \neg\psi_s$.

Claim (6). If ψ_s is consistent, then s is not eliminated at step 2 of the decidability algorithm for TDL.
Proof.
(a). Let $\mathrm{E}\bigcirc\alpha \in s$ and ψ_s be consistent. By the same reasoning as in the proof of Claim 4(a) in [7], we conclude that s satisfies $H9$. So s is not eliminated.
(b). Let $\mathrm{E}(\alpha \mathcal{U} \beta) \in s$ (respectively, $\mathrm{A}(\alpha \mathcal{U} \beta) \in s$) and suppose s is eliminated at step 2 because $H11$ (resp. $H'12$) 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 [7].

(c). Let $\neg\mathcal{K}_i\alpha \in s$ and ψ_s be consistent. Consider the set $S_{\neg\alpha} = \{\neg\alpha\}\cup\{\beta \mid \mathcal{K}_i\beta \in s\}$. 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 $\beta_j \in \{\beta \mid \mathcal{K}_i\beta \in s\}$ for $j \in \{1,\ldots,m\}$. By rule $R2$ we have $\vdash \mathcal{K}_i((\beta_1 \wedge \ldots \wedge \beta_m) \Rightarrow \alpha)$. By axioms $K1$ and PC we have $\vdash (\mathcal{K}_i\beta_1 \wedge \ldots \wedge \mathcal{K}_i\beta_m) \Rightarrow \mathcal{K}_i\alpha$. Since each $\mathcal{K}_i\beta_j \in s$ for $j \in \{1,\ldots,m\}$ and s is maximal and propositionally consistent, we have $\mathcal{K}_i\alpha \in s$. This contradicts the fact that ψ_s is consistent. So, $S_{\neg\alpha}$ is consistent. Now, since each set of formulae can be extended to a maximal one, we have that $S_{\neg\alpha}$ is contained in some maximal set t. Thus $\neg\alpha \in t$, and moreover, by the definition of $\sim_i$ in M and the definition of $S_{\neg\alpha}$ we have that $s \sim_i t$. Thus s cannot be eliminated at step 2(c) of the decidability algorithm.

(d). (By contradiction) Let $\neg\mathcal{X}_i\alpha \in s$ and s be eliminated at step 2(d) (because $H17$ is not satisfied). We will show that ψ_s is inconsistent. Since $\neg\mathcal{X}_i\alpha \in s$, by Claim 3 we have that $\vdash \psi_s \Rightarrow \neg\mathcal{X}_i\alpha$. Since $H17$ fails, by Claim 3 we have that $\vdash \psi_s \Rightarrow \mathcal{K}_i\alpha \wedge \mathcal{A}_i\alpha$. So, by axiom $X1$ we have $\vdash \psi_s \Rightarrow \mathcal{X}_i\alpha$. Therefore, we have that $\vdash \psi_s \Rightarrow \neg\mathcal{X}_i\alpha$ and $\vdash \psi_s \Rightarrow \mathcal{X}_i\alpha$. This implies that $\vdash \psi_s \Rightarrow \bot$. Thus, ψ_s is inconsistent.

(e). Suppose that s is consistent and it is eliminated at step 2(e)(because $H18$ is not satisfied) of the decidability algorithm. Thus, we have that $s \sim_i t$, $s \sim_i u$, $\alpha \in t$, and $\mathcal{K}_i\neg\alpha \in u$. 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 K3 and Claim 5, we have that $\neg\mathcal{K}_i\neg\alpha \Rightarrow \mathcal{K}_i\neg\mathcal{K}_i\neg\alpha \in s$. Therefore, we have that $\mathcal{K}_i\neg\mathcal{K}_i\neg\alpha \in s$. Thus, since $s \sim_i u$, we have that $\neg\mathcal{K}_i\neg\alpha \in u$. But this is a contradictions given that $\mathcal{K}_i\neg\alpha \in u$ an u is propositionally consistent. So s is inconsistent. Therefore s cannot be eliminated at step 2(e) of the decidability algorithm.

□

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

6 Discussion

In the paper we have shown that the logic TDL is decidable, and can be axiomatised. TDL permits to express different concepts of knowledge as well as time. In the following we briefly discuss alternative definitions of the notions defined in TDL.

Let us first note that the semantics of explicit knowledge in TDL is defined as in [9], with the difference that we assume the awareness function to be defined on local states (as opposed to global states as in [9]). In other words we have that: if $s \sim_i t$, then $\mathbb{A}_i(l_i(s)) = \mathbb{A}_i(l_i(t))$. Although this is a special case of the definition used in [9], we find this natural for the tasks we have in mind (communication, fault-tolerance, security [4], ...), given that all the information the agents will have in these cases can be represented in their local states.

[4] Using the TDL framework, it is easy to capture the capabilities of the $Dolev-Yao$ adversary [4]; we specify how the adversary can extract an intercepted message by defining an adequate awareness function. Moreover, using TDL we can perform a se-

Next, observe that the considered notion of explicit knowledge is sound, i.e., the following axiom is valid on TDL models: $\mathcal{X}_i\alpha \Rightarrow K_i\alpha$, but it is not complete i.e., $\not\models K_i\alpha \Rightarrow \mathcal{X}_i\alpha$ holds. Note also that defining awareness on local states forces the following two axiom schemas to be valid on TDL models: $\mathcal{A}_i\alpha \Rightarrow \mathcal{K}_i\mathcal{A}_i\alpha$ and $\neg\mathcal{A}_i\alpha \Rightarrow \mathcal{K}_i\neg\mathcal{A}_i\alpha$. These do not seem counterintuitive.

Further restrictions can be imposed on the awareness function. One consists in insisting that the function $\mathbb{A}_i$ maps consistent sets. If this is the case, the formula $\mathcal{A}_i\alpha \Rightarrow \neg\mathcal{A}_i(\neg\alpha)$ becomes valid on TDL models. While this is a perfectly sound assumption in some applications (for instance in the case $\mathbb{A}_i$ models a consistent database), for the aims of our work it seems more natural not to insist on this condition.

An even more crucial point is whether the local awareness functions should be consistent among one another, whether a "hierarchy of awareness" should be modelled, and whether they should at least agree with the global valuation function. In this paper we have made no assumption about the power of different agents; insisting this is the case is again reasonable in some scenarios but not considered here. It should be noted that forcing consistency between any $\mathbb{A}_i$ and $\mathcal{V}$ would make awareness and explicit knowledge collapse to the same modality. Further, knowledge about negative facts would be impaired given that $\mathbb{A}_i$ would only return propositions. The interested reader should refer to [9] for more details.

We have found that the decidability and completeness proofs presented here can be adapted to account for different choices on the awareness function, provided that appropriate conditions are included in the construction of Hintikka structures.

The notion presented here of *deductive knowledge* is directly inspired by the notion of algorithmic knowledge of [15, 29]. Typically, formalisms for algorithmic knowledge capture the notion of which the derivation algorithm is used to obtain a formula, and whether these derivations are correct and complete. The work presented here, on the other hand, focuses on the meta-logical properties of these notions, something not normally discussed, to our knowledge, in the literature.

It should be pointed out that alternative definitions of deductive knowledge can be considered. For example one can consider: $(M, s) \models \mathcal{D}'_i\alpha$ iff $(M, s) \models \mathrm{A}(\mathcal{K}_i\alpha\mathcal{U}\mathcal{X}_i\alpha)$, or $(M, s) \models \mathcal{D}''_i\alpha$ iff $(M, s) \models \mathcal{K}_i\alpha \wedge \mathrm{E}(\top\mathcal{U}\mathcal{X}_i\alpha)$. Both of them enjoy the same logical properties as the one proposed here. The first one describes a notion of "inevitability" in the deductions carried out by the agent. This does not seem as appropriate as the one we used here, as typically one intends to model the capability, not the certainty, of deducing some information. The second definition does not insist on implicit knowledge remaining true over the run while the deduction is taking place. In this case any explicit knowledge deduced could well be unsound (in the sense of [29]), something that cannot happen in the formalism of this paper.

---

mantical analysis of the *timed efficient stream loss-tolerant authentication* (TESLA) protocol [28]. The analysis allows us to reason about properties that so far could not be expressed in any other formalism. For more details we refer to [24].

We stress that all logics discussed above remain decidable. This allows us to explore model checking methods for them. We leave this for further work.

References

1. P. Blackburn, M. de Rijke, and Y. Venema. *Modal Logic*, volume 53 of *Cambridge Tracts in Theoretical Computer Science*. Cambridge University Press, 2001.
2. L. Catach. Normal multimodal logics. In *Proceedings of the 7th National Conference on Artificial Intelligence (AAAI'88)*, pp. 491–495, 1988. Morgan Kaufmann.
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*, pp. 52–71. Springer-Verlag, 1981.
4. D. Dolev and A. Yao. On the security of public key protocols. *IEEE Trans. Inf. Theory*, 29(2):198–208, 1983.
5. E. A. Emerson. Temporal and modal logic. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science*, chapter 16, pp. 996–1071. Elsevier Science Publishers, 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. 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.
8. R. Fagin and J. Y. Halpern. Belief, awareness, and limited reasoning. *Artificial Intelligence*, 34(1):39–76, 1988.
9. R. Fagin, J. Y. Halpern, Y. Moses, and M. Y. Vardi. *Reasoning about Knowledge*. MIT Press, Cambridge, 1995.
10. R. Fagin, J. Y. Halpern, and M. Vardi. A nonstandard approach to the logical omniscience problem. *Artificial Intelligence*, 79, 1995.
11. R. Fagin, J. Y. Halpern, and M. Y. Vardi. A model-theoretic analysis of knowledge. *Journal of ACM*, 91:382–428, 1991.
12. M. J. Fischer and R. E. Ladner. Propositional dynamic logic of regular programs. *Journal of Computer and System Sciences*, 18(2):194–211, 1979.
13. P. Gammie and R. van der Meyden. MCK: Model checking the logic of knowledge. In *Proceedings of 16th International Conference on Computer Aided Verification (CAV'04)*, volume 3114 of *LNCS*, pp. 479–483. Springer-Verlag, 2004.
14. J. Halpern, R. van der Meyden, and M. Y. Vardi. Complete axiomatisations for reasoning about knowledge and time. *SIAM Journal on Computing*, 33(3):674–703, 2003.
15. J. Y. Halpern, Y. Moses, and M. Y. Vardi. Algorithmic knowledge. In *Theoretical Aspects of Reasoning About Knowledge. Proceedings of the 5th Conference (TARK 1994)*, pp. 255–66. Morgan Kaufmann Publishers, 1994.
16. J. Hintikka. *Knowledge and Belief, An Introduction to the Logic of the Two Notions*. Cornell University Press, Ithaca (NY) and London, 1962.
17. W. van der Hoek. Systems for knowledge and belief. *Journal of Logic and Computation*, 3(2):173–195, 1993.
18. W. van der Hoek and M. Wooldridge. Model checking knowledge and time. In *Proceedings of the 9th International SPIN Workshop on Model Checking of Software*, 2002.
19. K. Konolige. *A Deduction Model of Belief*. Brown University Press, 1986.

20. M. Kracht and F. Wolter. Properties of independently axiomatizable bimodal logics. *Journal of Symbolic Logic*, 56(4):1469–1485, 1991.
21. D. Lehman. Knowledge, common knowledge, and related puzzles. In *Proceedings of the 3rd ACM Symposium on Principles of Distributed Computing*, pp. 62–67, 1984.
22. A. Lomuscio, R. van der Meyden, and M. Ryan. Knowledge in multi-agent systems: Initial configurations and broadcast. *ACM Transactions of Computational Logic*, 1(2), 2000.
23. A. Lomuscio and M. Sergot. Deontic interpreted systems. *Studia Logica*, 75(1):63–92, 2003.
24. A. Lomuscio and B. Woźna. A complete and decidable security-specialised logic and its application to the tesla protocol. In *Proceedings of the 5th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS'06)*. ACM Press, 2006. To appear.
25. R. van der Meyden. Axioms for knowledge and time in distributed systems with perfect recall. In *Proceedings, 9th Annual IEEE Symposium on Logic in Computer Science*, pp. 448–457. IEEE Computer Society Press, 1994.
26. R. van der Meyden and K. Wong. Complete axiomatizations for reasoning about knowledge and branching time. *Studia Logica*, 75(1):93–123, 2003.
27. W. Penczek and A. Lomuscio. Verifying epistemic properties of multi-agent systems via bounded model checking. *Fundamenta Informaticae*, 55(2):167–185, 2003.
28. A. Perrig, R. Canetti, J. D. Tygar, and Dawn X. Song. Efficient authentication and signing of multicast streams over lossy channels. In *IEEE Symposium on Security and Privacy*, pp. 56–73, May 2000.
29. R. Pucella. Deductive Algorithmic Knowledge. In *Proceedings of the 8th International Symposium on Artificial Intelligence and Mathematics (SAIM'04)*, Online Proceedings: AI&M 22-2004, 2004.
30. F. Raimondi and A. Lomuscio. Verification of multiagent systems via ordered binary decision diagrams: an algorithm and its implementation. In *Proceedings of the 3rd International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS'04)*, volume II. ACM, July 2004.
31. J. M. Sato. A study of Kripke style methods for some modal logic by Gentzen's sequential method. Technical report, Publication Research Institute for Mathematical Science, 1977.
32. E. Spann. Nextime is not necessary. In *Proceedings of the 3rd Conference on Theoretical Aspects of Reasoning about Knowledge*, pp. 241–256, 1990.
33. W. van der Hoek and M. Wooldridge. Cooperation, knowledge, and time: Alternating-time temporal epistemic logic and its applications. *Studia Logica*, 75(1):125–157, 2003.
34. R. van der Meyden and H. Shilov. Model checking knowledge and time in systems with perfect recall. In *Proceedings of the 19th Conference on Foundations of Software Technology and Theoretical Computer Science (FST&TCS'99)*, volume 1738 of *LNCS*, pp. 432–445. Springer-Verlag, 1999.
35. R. van der Meyden and Kaile Su. Symbolic model checking the knowledge of the dining cryptographers. In *Proceedings of the 17th IEEE Computer Security Foundations Workshop (CSFW'04)*, pp. 280–291. IEEE Computer Society, 2004.
36. B. Woźna, A. Lomuscio, and W. Penczek. Bounded model checking for knowledge over real time. In *Proceedings of the 4st International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS'05)*, volume I, pp. 165–172. ACM Press, 2005.