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

Bożena Woźna and Alessio Lomuscio

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

Technical report: RN/05/18
August 2005

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 allows 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 [9]) 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 [14], of $S5$ with itself n times, more interesting extensions have been considered. For example one of the systems presented in [10] 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 $S5WD_n$ [15] 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 [16, 1]. The completeness proofs in these works typically are based on some reasoning on the canonical model [12, 17].

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

Because of the importance in applications, and in particular in verification, there has been recent growing interests in combinations 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 about the knowledge of others, and other various epistemic properties typically of interest in applications. Combinations of the epistemic system $S5_n$ with linear logic, and branching time logic have been explored. 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 [18, 19]. This is of particular relevance for verification of multi-agent systems via model checking, an area that has received some attention recently [7, 11, 20, 22–25].

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 [13, 6, 4, 8] over the years. The most relevant for this paper is the logic for implicit knowledge (i.e. modelled by $S5$), awareness, and explicit knowledge presented in [4, 5]. In this work two new operators: $\mathcal{A}_i$ and $\mathcal{X}_i$ are introduced. The former represents the information an agent has at his 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 his 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 try and show that combinations of explicit knowledge with branching time not only give rise to interesting axiomatisation problems, but seem to allow to express other rather subtle epistemic concepts needed in applications, one of them being the one of "deductive knowledge"[1] formalised below.

The rest of the paper is organised as follows. In Section 2 we present briefly the basics of the underlying syntactical and semantical assumptions used in the

[1] Our use of the term "deductive knowledge" is inspired by [21], although the focus in this paper is different.

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

2 Temporal Deductive Logic

In this section, we present syntax, semantics and some properties of Temporal Deductive Logic (TDL), a language for branching time, and different epistemic notions. TDL extends standard combinations of branching time epistemic languages by introducing three further epistemic modalities: awareness, explicit knowledge, and deductive knowledge.

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 $\mathcal{WF}(\mathrm{TDL})$ of well-formed TDL formulas 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 $i \in \mathcal{AG}$.

The above syntax extends CTL [2] with standard epistemic modality $\mathcal{K}_i$ as well as operators for explicit knowledge ($\mathcal{X}_i$) and awareness ($\mathcal{A}_i$) as in [5]. 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. $\alpha \wedge \beta \stackrel{def}{=} \neg(\neg\alpha \vee \neg\beta)$, $\alpha \Rightarrow \beta \stackrel{def}{=} \neg\alpha \vee \beta$, $\alpha \Leftrightarrow \beta \stackrel{def}{=} (\alpha \Rightarrow \beta) \wedge (\beta \Rightarrow \alpha)$, $\mathrm{A}\bigcirc\alpha \stackrel{def}{=} \neg\mathrm{E}\bigcirc\neg\alpha$, $\mathrm{E}\Diamond\alpha \stackrel{def}{=} \mathrm{E}(\top\mathcal{U}\alpha)$, $\mathrm{A}\Diamond\alpha \stackrel{def}{=} \mathrm{A}(\top\mathcal{U}\alpha)$, $\mathrm{E}\Box\alpha \stackrel{def}{=} \neg\mathrm{A}\Diamond\neg\alpha$, $\mathrm{A}\Box\alpha \stackrel{def}{=} \neg\mathrm{E}\Diamond\neg\alpha$, $\mathrm{A}(\alpha\mathcal{W}\beta) \stackrel{def}{=} \neg\mathrm{E}(\neg\alpha\mathcal{U}\neg\beta)$, $\mathrm{E}(\alpha\mathcal{W}\beta) \stackrel{def}{=} \neg\mathrm{A}(\neg\alpha\mathcal{U}\neg\beta)$, $\overline{\mathcal{K}}_i\alpha \stackrel{def}{=} \neg\mathcal{K}_i(\neg\alpha)$.

Let φ and ψ be TDL formulas. We say that ψ is a *sub-formula* of φ if either (a) $\psi = \varphi$; or (b) φ is of the form $\neg\alpha$, $\mathrm{E}\bigcirc\alpha$, $\mathcal{K}_i\alpha$, $\mathcal{X}_i\alpha$, $\mathcal{A}\alpha$, 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 a TDL formula φ (denoted by $|\varphi|$) is equal to the number of symbols appearing in φ.

2.2 Semantics

Traditionally, the semantics of temporal logics with epistemic operators is defined on interpreted systems, defined in the following way [5]. Each agent $i \in \mathcal{AG}$ is associated with a set of local states L_i; 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 $l_i : S \to L_i$ is a function which returns the local state of agent i from a global state; $\mathcal{V} : S \longrightarrow 2^{\mathcal{PV}}$ is a valuation function such that $(\forall s \in S)\ \top \in \mathcal{V}(s)$. $\mathcal{V}$ assigns to each state a set of proposition variables that are assumed to be true at that state. For more details and further explanations of the notation we refer to [5].

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 [5], we do not attach any fixed interpretation to the notion of awareness, i.e. to be aware can mean "to be able to figure out the truth", "to be able to compute the truth within time T", etc.

Definition 1 (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 $\mathbb{A}_i : L_i \longrightarrow 2^{\mathcal{WF}(\mathrm{TDL})}$ is an awareness function assigning a set of formulas to each state, for each $i \in \mathcal{AG}$.*

Intuitively, $\mathbb{A}_i(l_i(s))$ is a set of formulas that the agent i is aware of at state s, i.e. the set of formulas 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 formulas that the agent is aware of can be arbitrary and may not be closed under sub-formulas. Note also that the definition of a model is an extension of the awareness structure, introduced in [5], 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 formulas 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 2 (Satisfaction). *Let M be a model, s a state, and α, β* TDL *formulas. 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* $p \in \mathcal{V}(s)$, $\quad (M, s) \models \alpha \wedge \beta$ *iff* $(M, s) \models \alpha$ *and* $(M, s) \models \beta$,
$(M, s) \models \neg\alpha$ *iff* $(M, s) \not\models \alpha$, $\quad (M, s) \models \mathrm{E}\bigcirc\alpha$ *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* $(M, s') \models \alpha)$,
$(M, s) \models \mathcal{X}_i\alpha$ *iff* $(M, s) \models \mathcal{K}_i\alpha$ *and* $(M, s) \models \mathcal{A}_i\alpha$,
$(M, s) \models \mathcal{A}_i\alpha$ *iff* $\alpha \in \mathbb{A}_i(l_i(s))$.

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 [5]. Satisfaction for the Boolean and temporal operators as well as the epistemic modalities $\mathcal{K}_i$, $\mathcal{X}_i$, $\mathcal{A}_i$ is standard. 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, i.e. he had implicit knowledge of it. 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.

Let M be a model. We say that a TDL formula φ is *valid* in M (written $M \models \varphi$), if $M, s \models \varphi$ for all states $s \in S$, and a TDL formula φ is *satisfiable* in M, if $M, s \models \varphi$ for some state $s \in S$. We say that a TDL formula φ is *not valid* in M (written $M \not\models \varphi$), if $\neg\varphi$ is satisfiable in M. We say that a TDL formula 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. We say that a TDL formula φ is *not valid* (written $\not\models \varphi$), if there is a model M such that $M \not\models \varphi$.

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.

In order to establish the FMP for TDL, we follow the construction presented in [3]. Therefore we begin with providing definitions of two auxiliary structures: a *Hintikka structure* for a given TDL formula, and the *quotient construction* for a given model.

Definition 3 (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 1, and $\mathbb{L} : S \to 2^{\mathcal{WF}(\mathrm{TDL})}$ is a labelling function assigning a set of formulas 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 $(\alpha \vee \beta) \in \mathbb{L}(s)$, 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* $\mathcal{K}_i\alpha \in \mathbb{L}(u)$ *and* $\alpha \in \mathbb{L}(u)$
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 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 (i.e. consistent with semantics rules), but they do not require that the formulas belonging to $\mathbb{L}(s)$ form a maximal set of formulas, for any $s \in S$. This means that there are formulas that are satisfied in a given state but they are not included in the label of that state. As usually, 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 properties*.

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. In order to define the quotient construction for M, an indexed equivalence on states of the model M is defined first. Equivalent states are then identified thereby generating the quotient structure (a finite model). The equivalence relation is defined with respect to the *Fischer-Ladner closure* of 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 formulas 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 $\mathcal{A}_i\alpha \in CL(\varphi)$,
Observation. Note that for a given TDL formula φ, $FL(\varphi)$ is the set of formulas that are essential to establish the truth of φ in a model. Moreover, this set is finite since the following lemma holds.

Lemma 1. *Given a* TDL *formula* φ, $\overline{\overline{FL(\varphi)}} \leq 2(|\varphi| + 3)$*, where* $\overline{\overline{FL(\varphi)}}$ *denotes the size of the set* $FL(\varphi)$*.*
Proof. Straightforward by induction on the length of φ. □

Lemma 2. *Let* φ *be a* TDL *formula. Then the following holds:*
(a). If $M = (S, T, \sim_1, \ldots, \sim_n, \mathcal{V}, \mathbb{A}_1, \ldots, \mathbb{A}_n)$ *is a model for* φ*, then* $HS = (S, T, \sim_1, \ldots, \sim_n, \mathbb{L}, \mathbb{A}_1, \ldots, \mathbb{A}_n)$ *with* $\mathbb{L}$ *defined by:* $\alpha \in \mathbb{L}(s)$ *if* $\alpha \in FL(\varphi)$ *and* $(M, s) \models \alpha$*, for all* $s \in S$*, is a Hintikka structure for* φ*.*

(b). If $HS = (S, T, \sim_1, \ldots, \sim_n, \mathbb{L}, \mathbb{A}_1, \ldots, \mathbb{A}_n)$ is a Hintikka structure for φ, then $M = (S, T, \sim_1, \ldots, \sim_n, \mathcal{V}, \mathbb{A}_1, \ldots, \mathbb{A}_n)$ with $\mathcal{V}$ defined by: $\mathcal{V}(s) = \mathbb{L}(s) \cap \mathcal{PV}$, for all $s \in S$, is a model for φ.

Proof. Straightforward by induction on the length of φ. $\square$

The equivalence relation $\leftrightarrow_{FL(\varphi)}$ on a set of states S in the model M 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 $\{w \in S \mid w \leftrightarrow_{FL(\varphi)} s\}$.

Definition 4 (Quotient structure). *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* 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. $(w, w') \in T\}$, $\sim'_i = \{([s], [s']) \in S' \times S' \mid (\exists w \in [s])(\exists w' \in [s'])$ s.t. $(w, w') \in \sim_i\}$, $\mathbb{L}' : S' \to 2^{FL(\varphi)}$ is a function defined by: $\mathbb{L}'([s]) = \{\alpha \in FL(\varphi) \mid (M, s) \models \alpha\}$, $\mathbb{A}'_i : l'_i(S') \to 2^{\mathcal{WF}(\mathrm{TDL})}$ 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. $l'_i : S' \to 2^{L_i}$ is a function defined by: $l'_i([s]) = \bigcup_{t \in [s]} l_i(t)$, for $i \in \mathcal{AG}$, returning a set of local states for agent i for a given set of global states.*

Observation. Note that the set S' is finite as it is the result of collapsing states satisfying formulas that belong to the finite set $FL(\varphi)$. In fact we have $\overline{\overline{S'}} \leq 2^{\overline{\overline{FL(\varphi)}}}$. Note also that $\mathbb{A}'_i$ is well defined. Moreover, it is easy to see that as in the CTL case, the resulting quotient structure may not be a model. In particular, the following lemma holds.

Lemma 3. *The quotient construction does not preserve satisfiability of formulas of the form* $\mathrm{A}(\alpha \mathcal{U} \beta)$*, where $\alpha, \beta \in \mathcal{WF}(\mathrm{TDL})$. 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* $\mathrm{A}(\top \mathcal{U} p)$*.*

The proof of Theorem 3.6 in [3] can easily be extended to the case of TDL.

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*; 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.

A *directed acyclic graph* is a pair $DAG = (S, T)$, where S is a set of states (nodes) and $T \subseteq S \times S$ is a set of edges (a transition relation). An *interior* (respectively *frontier*) node of a DAG is one which has (respectively does not have) a T-successor. The *root* of a DAG is the node (if it exists) from which all other nodes are reachable. A *fragment* $M = (S', T', \sim'_1, \ldots, \sim'_n, \mathbb{L}', \mathbb{A}'_1, \ldots, \mathbb{A}'_n)$ of a Hintikka structure $HS = (S, T, \sim_1, \ldots, \sim_n, \mathbb{L}, \mathbb{A}_1, \ldots, \mathbb{A}_n)$ is a structure such that (S', T') is 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 $M = (S, T, \sim_1, \ldots, \sim_n, \mathbb{L}, \mathbb{A}_1, \ldots, \mathbb{A}_n)$ and $M' = (S', T', \sim'_1, \ldots, \sim'_n, \mathbb{L}', \mathbb{A}'_1, \ldots, \mathbb{A}'_n)$, we say that M is *contained* in M', and write $M \subseteq M'$, if $S \subseteq S'$, $T = T' \cap (S \times S)$, $\sim_i = \sim_i' \cap (S \times S)$, $\mathbb{L} = \mathbb{L}'|S$, $\mathbb{A}_i = \mathbb{A}_i'|L_i$.

It can be checked that the proof of Lemma 3.8 in [3] can be extended to establish the following.

Lemma 4. *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 $M' = (S', T', \sim_1', \ldots \sim_n', \mathbb{L}', \mathbb{A}_1', \ldots, \mathbb{A}_n')$ the quotient structure of M by $\leftrightarrow_{FL(\varphi)}$. Suppose $\mathrm{A}(\alpha \mathcal{U} \beta) \in \mathbb{L}'([s])$ for some $[s] \in S'$. Then there is a fragment $(S'', T'', \sim_1'', \ldots, \sim_n'', \mathbb{L}'', \mathbb{A}_1'', \ldots, \mathbb{A}_n'') \subseteq M'$ such that: (a) (S'', T'') is a DAG with root $[s]$; (b) for all the frontier nodes $[t] \in S''$, $\beta \in \mathbb{L}''([t])$; (c) for all the interior nodes $[u] \in S''$, $\alpha \in \mathbb{L}''([u])$.*

Definition 5 (Pseudo-model). *Let φ be a* TDL *formula. A* pseudo-model *$M = (S, T, \sim_1, \ldots, \sim_n, \mathbb{L}, \mathbb{A}_1, \ldots, V_n)$ for φ is defined in the same manner as a Hintikka structure for φ in Definition 3, 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 $(S', T', \sim_1', \ldots, \sim_n', \mathbb{L}', \mathbb{A}_1', \ldots, \mathbb{A}_n') \subseteq M$ such that: (a) (S', T') is a DAG with root s; (b) for all frontier nodes $t \in S'$, $\beta \in \mathbb{L}'(t)$; (c) for all interior nodes $u \in S'$, $\alpha \in \mathbb{L}'(u)$.*

Lemma 5. *Let φ be a* TDL *formulas, $FL(\varphi)$ the Fischer-Ladner closure of φ, $M = (S, T, \sim_1, \ldots, \sim_n, \mathcal{V}, \mathbb{A}_1, \ldots, \mathbb{A}_n)$ a model for φ, and $M_{\leftrightarrow_{FL(\varphi)}} = (S', T', \sim_1', \ldots, \sim_n', \mathbb{L}, \mathbb{A}_1', \ldots, \mathbb{A}_n')$ the quotient structure of M by $\leftrightarrow_{FL(\varphi)}$. Then, $M_{\leftrightarrow_{FL(\varphi)}}$ is a pseudo-model for φ.*

Proof. We will consider φ to be of the forms $\mathrm{A}(\alpha \mathcal{U} \beta)$, $\neg \mathrm{E}\bigcirc\alpha$, $\neg \mathcal{K}_i \alpha$, $\mathcal{X}_i \alpha$, and $\mathcal{A}_i \alpha$. The other cases can be proven in the similar way.

1. $\varphi = \mathrm{A}(\alpha \mathcal{U} \beta)$. Let $(M, s) \models \mathrm{A}(\alpha \mathcal{U} \beta)$, and $\mathrm{A}(\alpha \mathcal{U} \beta) \in \mathbb{L}([s])$. By the definition of $\models$, we have that $(\forall \pi \in \Pi(s))(\exists n \geq 0)[(M, \pi(n)) \models \beta$ and $(\forall j < n)$ $(M, \pi(j)) \models \alpha]$. This implies that either $(M, s) \models \beta$ or $(\forall \pi \in \Pi(s))(\exists n > 0)$ $[(M, \pi(n)) \models \beta$ and $(\forall j < n)(M, \pi(j)) \models \alpha]$. Thus, $(M, s) \models \beta$ or $(M, s) \models \alpha \wedge \mathrm{A}\bigcirc\mathrm{A}(\alpha \mathcal{U} \beta)$, which is equivalent to the fact that $(M, s) \models \beta$ or $(M, s) \models \alpha \wedge \neg \mathrm{E}\bigcirc(\neg \mathrm{A}(\alpha \mathcal{U} \beta))$. Therefore, by the definitions of $\leftrightarrow_{FL(\varphi)}$ and $\mathbb{L}$, we have that $\beta \in \mathbb{L}([s])$ or $\alpha \wedge \neg \mathrm{E}\bigcirc(\neg \mathrm{A}(\alpha \mathcal{U} \beta)) \in \mathbb{L}([s])$. So, condition $H7$ is fulfilled.
2. $\varphi = \neg \mathrm{E}\bigcirc\alpha$. Let $(M, s) \models \neg \mathrm{E}\bigcirc\alpha$, and $\neg \mathrm{E}\bigcirc\alpha \in \mathbb{L}([s])$. By the definition of $\models$, we have that $(\forall t \in S)$ if $(s, t) \in T$ then $(M, t) \models \neg\alpha$. Thus, by the definitions of $\leftrightarrow_{FL(\varphi)}$ and $\mathbb{L}$, we have that $\neg\alpha \in \mathbb{L}([t])$ for all $t \in S$ such that $(s, t) \in T$. Since $(s, t) \in T$, by the definition of T', we have that $([s], [t]) \in T'$. Therefore, we conclude that $(\forall [t] \in S')$ if $([s], [t]) \in T'$ then $\neg\alpha \in \mathbb{L}([t])$. So, condition $H10$ is fulfilled.
3. $\varphi = \mathrm{A}(\alpha \mathcal{U} \beta)$. Let $(M, s) \models \mathrm{A}(\alpha \mathcal{U} \beta)$, and $\mathrm{A}(\alpha \mathcal{U} \beta) \in \mathbb{L}([s])$. By Lemma 4, we have that there is a fragment $(S'', T'', \sim_1'', \ldots, \sim_n'', \mathbb{L}'', V_1'', \ldots, V_n'') \subseteq M_{\leftrightarrow_{FL(\varphi)}}$ such that: (a) (S'', T'') is a DAG with root $[s]$; (b) for all the frontier nodes $[t] \in S''$, $\beta \in \mathbb{L}''([t])$; (c) for all the interior nodes $[u] \in S''$, $\alpha \in \mathbb{L}''([u])$. So, condition $H'12$ is fulfilled.
4. $\varphi = \neg \mathcal{K}_i \alpha$. 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.
5. $\varphi = \mathcal{X}_i\alpha$. 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.
6. $\varphi = \mathcal{A}_i\alpha$. 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.

$\square$

Theorem 1. TDL *has the finite model property.*

Proof. [sketch] To prove the theorem it is sufficient to show that for a given TDL formula φ the following conditions are equivalent: (1) φ is satisfiable; (2) there is a finite pseudo-model for φ; (3) there is a Hintikka structure for φ.

(3) $\Rightarrow$ (1) follows from Lemma 2. (1) $\Rightarrow$ (2) follows from Lemma 5. 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 [3] for the proof of Theorem 4.1. $\square$

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 pseudo-model for φ. We proceed as follows.

1. Build an initial pseudo-model $M^0 = (S^0, T^0, \sim_i^0, \ldots, \sim_n^0, \mathbb{L}^0, \mathbb{A}_1^0, \ldots, \mathbb{A}_n^0)$ for φ with the following constraints: $S^0 = \{\Delta \mid \Delta \subseteq FL(\varphi)$ and Δ is maximal and satisfies 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 $i \in \mathcal{AG}$, $\sim_i^0 \subseteq S^0 \times S^0$ is the relation such that $(\Delta_1, \Delta_2) \in \sim_i$ iff $\{\alpha \mid \mathcal{K}_i\alpha \in \Delta_1\} \subseteq \Delta_2$; $\mathbb{L}^0(\Delta) = \Delta$. Further, we assume that for each agent $i \in \mathcal{AG}$ the set of local states L_i is equal to S^0. So, $\mathbb{A}_i^0(\Delta) = \{\alpha \mid \mathcal{A}_i\alpha \in \Delta\}$ for each agent $i \in \mathcal{AG}$.
Note that the initial pseudo-model 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 initial pseudo-model 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 $\mathrm{E}(\alpha \mathcal{U} \beta) \in \Delta_1$ (resp. $\mathrm{A}(\alpha \mathcal{U} \beta) \in \Delta_1$) 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 $\sim_i$ 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 three states $\Delta_1, \Delta_2, \Delta_3 \in S^O$ such that $\Delta_1 \sim_i \Delta_2$ and $\Delta_1 \sim_i \Delta_3$ and $\mathcal{K}_i \alpha \in \Delta_2$ and ($\neg \mathcal{K}_i \alpha \in \Delta_3$ or $\neg \alpha \in \Delta_3$).

Note that this part of the algorithm must terminate, since there are only a finite number of states in the pseudo-model.

3. Let $M^f = (S^f, T^f, \sim_1^f, \ldots, \sim_n^f, \mathbb{L}^f, \mathbb{A}_1^f, \ldots, \mathbb{A}_n^f)$ be the final pseudo-model. If there exists a state $s \in S^f$ such that $\varphi \in \mathbb{L}^f(s)$, then φ is satisfiable. If not, then φ is not satisfiable.

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

Lemma 6. *The decidability algorithm for* TDL *terminates. Let* $M^f = (S^f, T^f, \sim_1^f, \ldots, \sim_n^f, \mathbb{L}^f, \mathbb{A}_1^f, \ldots, \mathbb{A}_n^f)$ *be the resulting structure of the algorithm. The* TDL *formula* φ *is satisfiable iff* $\varphi \in s$*, for some* $s \in S^f$*.*

Proof. [sketch] Termination is obvious given that the initial set is finite. 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 $S^f = \emptyset$ (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 $M_{\leftrightarrow_{FL(\varphi)}} = (S', T', \sim'_1, \ldots, \sim'_n, \mathbb{L}', \mathbb{A}'_1, \ldots, \mathbb{A}'_n)$ be the quotient structure of M by $\leftrightarrow_{FL(\varphi)}$. $M_{\leftrightarrow_{FL(\varphi)}}$ is a pseudo-model for φ (see the proof of Theorem 1). 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 $s \in S'$.

Let us consider the following function $f : S' \to S^0$ that is defined by $f(s) = \mathbb{L}'(s)$. It is easy to check that for T^0, and $\sim_i^0$, defined as in step 1 of the decidability algorithm, the following conditions hold:

1. if $(s,t) \in T'$, then $(f(s), f(t)) \in T^0$;
Proof (via contradiction): Let $(s,t) \in T'$ and $(f(s), f(t)) \notin T^0$. Then, by the definition of T^0 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 $(f(s), f(t)) \in \sim_i^0$;
Proof (via contradiction): Let $(s,t) \in \sim'_i$ and $(f(s), f(t)) \notin \sim_i^0$. Then, by the definition of $\sim_i^0$ 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 $M_{\leftrightarrow_{FL(\varphi)}}$ under f is contained in M^f, i.e. $M_{\leftrightarrow_{FL(\varphi)}} \subseteq M^f$. It also can be checked that if $s \in S'$, then $f(s) \in S^0$ will not be eliminated via the step 2 of the decidability algorithm. So, in fact, $f(s) \in S^f$. This can be checked by induction on the order in which states of S^0 are eliminated. Therefore, it follows that for some $s \in S^f$ we have $\varphi \in \mathbb{L}^f(s)$.

□

5 A Complete Axiomatic System for TDL

An *axiomatic system* consists of a collection of *axioms* and *inference rules*. An axiom is a formula, and an inference rule has the form "from formulas $\varphi_1, \ldots, \varphi_m$ infer formula φ". We say that φ is *provable* (written $\vdash \varphi$) if there is a sequence of formulas ending with φ, such that each formula is either an instance of an axiom, or follows from other provable formulas 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 formulas is *consistent* exactly if and only if the conjunction $\varphi_1 \wedge \ldots \wedge \varphi_m$ of its members is consistent, and an infinite set of formulas is *consistent* exactly if all of its finite subsets are consistent. A set F of formulas is a *maximal consistent set* if it is consistent and for all $\varphi \notin F$, the set $F \cup \{\varphi\}$ is inconsistent. An axiom system is said to be *sound*, if $\vdash \varphi$ then $\models \varphi$. An axiom system is said to be *complete*, if $\models \varphi$ then $\vdash \varphi$.

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, i.e.* $\models \varphi$ *iff* $\vdash \varphi$*, for any formula* $\varphi \in \mathcal{WF}(\mathrm{TDL})$.

Proof. Soundness can be checked inductively as standard. For completeness, it is sufficient to show that any consistent formula is satisfiable. To do this, first we construct a pseudo-model $M = (S^0, T^0, \sim_1^0, \ldots, \sim_n^0, \mathbb{L}^0, \mathbb{A}_1^0, \ldots, \mathbb{A}_n^0)$ for φ just as in the decidability algorithm for TDL, and for each $s \in S^0$ we define the formula ψ_s as the conjunction of the formulas in s, i.e. $\psi_s = \bigwedge_{\alpha \in s} \alpha$. Next, we show that if a state $s \in S^0$ is eliminated at step 2 of the decidability algorithm for TDL, then ψ_s is inconsistent. Once we have shown this, we proceed as follows.

It can be checked by propositional reasoning 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 some ψ_s is consistent as well. This particular s will not be eliminated at step 2 of the decidability algorithm for TDL. Therefore, a pseudo-model for φ is obtained. So, by Theorem 1, φ is satisfiable.

Claim (1). Let $s \in S^0$ 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^0, we have that any s in S^0 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 $\alpha \in s$. $\square$

Claim (2). Let $i \in \mathcal{AG}$. If $(s,t) \not\in \sim_i$ as constructed in step 1 of the decidability algorithm for TDL, then $\psi_t \wedge \psi_s$ is inconsistent.

Proof. Let $(s,t) \not\in \sim_i$. Then, by the definition of $\sim_i$, we have that $\mathcal{K}_i\alpha \in s$ and $\alpha \notin t$, for some α. Thus, by maximality $\neg\alpha \in t$. So, we have $\vdash \psi_s \Rightarrow \mathcal{K}_i\alpha$ and $\vdash \psi_t \Rightarrow \neg\alpha$. By axiom $K2$ $\vdash \psi_s \Rightarrow \alpha$. Therefore, $\vdash (\psi_t \wedge \psi_s) \Rightarrow \neg\alpha \wedge \alpha$. Hence, $\psi_t \wedge \psi_s$ is inconsistent. $\square$

Claim (3). If $(s,t) \notin T$ as constructed at step 1 of the decidability algorithm for TDL, then $\psi_s \wedge \mathrm{E}\bigcirc\psi_t$ is inconsistent.

Proof. Let $(s,t) \notin T$. By the definition of T we have that $\neg\mathrm{E}\bigcirc\alpha \in s$ and $\alpha \in t$. Therefore, we have $\vdash \psi_s \Rightarrow \neg\mathrm{E}\bigcirc\alpha$ and $\vdash \psi_t \Rightarrow \alpha$. By $R3$ we have $\vdash \mathrm{E}\bigcirc\psi_t \Rightarrow \mathrm{E}\bigcirc\alpha$. This implies that $\vdash (\psi_s \wedge \mathrm{E}\bigcirc\psi_t) \Rightarrow (\neg\mathrm{E}\bigcirc\alpha \wedge \mathrm{E}\bigcirc\alpha)$. Thus $\vdash (\psi_s \wedge \mathrm{E}\bigcirc\psi_t) \Rightarrow \bot$, which means that $\psi_s \wedge \mathrm{E}\bigcirc\psi_t$ is inconsistent. $\square$

We now show, by induction on the structure of the decidability algorithm for TDL, that if a state $s \in S^0$ is eliminated, then $\vdash \neg\psi_s$.

Claim (4). 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 [3], we conclude that s satisfies $H9$. So s is not eliminated.

(b). Let $\mathrm{E}(\alpha\mathcal{U}\beta) \in s$ (resp. $\mathrm{A}(\alpha\mathcal{U}\beta) \in s$) 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) (resp. Claim 4(d)) in [3].

(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 $j \in \{1, \ldots, m\}$, 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 formulas 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 $\sim_i^0$ in M and the definition of $S_{\neg\alpha}$ we have that $s \sim_i^0 t$. Thus, s satisfies $H15$.

(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 1 we have that $\vdash \psi_s \Rightarrow \neg\mathcal{X}_i\alpha$. Since $H17$ fails, again by Claim 1 we have $\vdash \psi_s \Rightarrow \mathcal{K}_i\alpha \wedge \mathcal{A}_i\alpha$. 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 $\vdash \psi_s \Rightarrow \perp$. Thus, ψ_s is inconsistent.

$\square$

Claim (5). If $\psi_s \wedge \psi_t \wedge \psi_u$ is consistent, then s, t and u are not eliminated at step 2(e) of the decidability algorithm for TDL.

Proof.[By contraposition] We show that if s, t, and u are eliminated at step 2.(e) (because $H20$ is not satisfied), then $\psi_s \wedge \psi_t \wedge \psi_u$ is inconsistent. Let s, t, and u be eliminated at step 2.(e). Then, we have that $s \sim_i t$ and $s \sim_i u$ and $\mathcal{K}_i\alpha \in t$ and either $\neg\mathcal{K}_i\alpha \in u$, or $\neg\alpha \in u$. Let first assume that $s \sim_i t$ and $s \sim_i u$ and $\mathcal{K}_i\alpha \in t$ and $\neg\mathcal{K}_i\alpha \in u$. By Claim 1 we have that $\vdash \psi_t \Rightarrow \mathcal{K}_i\alpha$ and $\vdash \psi_u \Rightarrow \neg\mathcal{K}_i\alpha$. It follows that $\vdash \psi_t \wedge \psi_u \Rightarrow \mathcal{K}_i\alpha \wedge \neg\mathcal{K}_i\alpha$ holds. This implies that $\vdash \psi_t \wedge \psi_u \Rightarrow \perp$. So, $\vdash \psi_s \wedge \psi_t \wedge \psi_u \Rightarrow \perp$ as well. Therefore $\psi_s \wedge \psi_t \wedge \psi_u$ is inconsistent. The case that $s \sim_i t$ and $s \sim_i u$ and $\mathcal{K}_i\alpha \in t$ and $\neg\alpha \in u$ can be proven similarly.
$\square$
We have now shown that only states s with ψ_s inconsistent are eliminated. This ends the completeness proof. $\square$

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 [5], with the difference that we assume the awareness function to be defined on local states (as opposed to global states as in [5]). 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 [5] we find this natural for the tasks we have in mind (communication, security, fault-tolerance, ...), given that all the information of the agents will in these cases be represented in their local states. Note 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 $\neg\mathcal{A}_i\alpha \Rightarrow \mathcal{K}_i\neg\mathcal{A}_i\alpha$.

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 [5] 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 [8, 21]. Typically, formalisms for algorithmic knowledge are interested in the notion of the derivation algorithm 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 discussed, to our knowledge, in the literature.

It should be pointed out that alternative definitions of deductive knowledge can be considered. For example: $(M, s) \models \mathcal{D}'_i\alpha$ iff $(M, s) \models \mathrm{A}(\mathcal{K}_i\alpha \mathfrak{U} \mathcal{X}_i\alpha)$, and $(M, s) \models \mathcal{D}''_i\alpha$ iff $(M, s) \models \mathcal{K}_i\alpha$ and $(M, s) \models \mathrm{E}(\top \mathfrak{U} \mathcal{X}_i\alpha)$.

Both of them enjoy the same logical properties as the one proposed here. However, 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, 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. Obviously in this case any explicit knowledge deduced could well be unsound (in the sense of [21]), something that cannot happen in the formalism of this paper.

We are keen to 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. L. Catach. Normal multimodal logics. In T. M. Smith and G. R. Mitchell, editor, *Proc. of AAAI'88*, pages 491–495, St. Paul, MN, 1988. Morgan Kaufmann.
2. 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.
3. 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.
4. R. Fagin and J. Y. Halpern. Belief, awareness, and limited reasoning. *Artificial Intelligence*, 34(1):39–76, 1988.
5. R. Fagin, J. Y. Halpern, Y. Moses, and M. Y. Vardi. *Reasoning about Knowledge.* MIT Press, Cambridge, 1995.

6. R. Fagin, J. Y. Halpern, and M. Vardi. A nonstandard approach to the logical omniscience problem. *Artificial Intelligence*, 79, 1995.
7. P. Gammie and R. van der Meyden. Mck: Model checking the logic of knowledge. In *Proc. of CAV'04*, volume 3114 of *LNCS*, pages 479–483. Springer-Verlag, 2004.
8. J. Y. Halpern, Y. Moses, and M. Y. Vardi. Algorithmic knowledge. In *Proc. of TARK'94*, pages 255–66. Morgan Kaufmann Publishers, 1994.
9. J. Hintikka. *Knowledge and Belief, An Introduction to the Logic of the Two Notions.* Cornell University Press, Ithaca (NY) and London, 1962.
10. W. van der Hoek. Sytems for knowledge and belief. *Journal of Logic and Computation*, 3(2):173–195, 1993.
11. W. van der Hoek and M. Wooldridge. Model checking knowledge and time. In *Proc. of SPIN'02*, 2002.
12. D. Kaplan. Review of "A semantical analysis of modal logic I: normal modal propositional calculi". *Journal of Symbolic Logic*, 31:120–122, 1966.
13. K. Konolige. *A Deduction Model of Belief.* Brown University Press, 1986.
14. M. Kracht and F. Wolter. Properties of independently axiomatizable bimodal logics. *Journal of Symbolic Logic*, 56(4):1469–1485, 1991.
15. 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.
16. A. Lomuscio and M. Sergot. Deontic interpreted systems. *Studia Logica*, 75(1):63–92, 2003.
17. D. Makinson. On some completeness theorems in modal logic. *Zeitschrift für Mathematische Logik und Grundlagen der Mathematik*, 12:379–384, 1966.
18. R. van der Meyden. Axioms for knowledge and time in distributed systems with perfect recall. In *Proc. of LICS*, pages 448–457, 1994. IEEE Computer Society Press.
19. R. van der Meyden and K. Wong. Complete axiomatizations for reasoning about knowledge and branching time. *Studia Logica*, 75(1):93–123, 2003.
20. W. Penczek and A. Lomuscio. Verifying epistemic properties of multi-agent systems via bounded model checking. *Fundamenta Informaticae*, 55(2):167–185, 2003.
21. R. Pucella. Deductive Algorithmic Knowledge. In *Proc. of SAIM'04*, Online Proceedings: AI&M 22-2004, 2004.
22. F. Raimondi and A. Lomuscio. Verification of multiagent systems via ordered binary decision diagrams: an algorithm and its implementation. In *Proc. of AAMAS'04*, volume II. ACM, July 2004.
23. R. van der Meyden and H. Shilov. Model checking knowledge and time in systems with perfect recall. In *Proc. of FST&TCS'99*, volume 1738 of *LNCS*, pages 432–445. Springer-Verlag, 1999.
24. R. van der Meyden and Kaile Su. Symbolic model checking the knowledge of the dining cryptographers. In *Proc. of CSFW '04*, pages 280 – 291. IEEE Computer Society, 2004.
25. B. Woźna, A. Lomuscio, and W. Penczek. Bounded model checking for knowledge over real time. In *Proc. of AAMAS'05*, volume I, pages 165-172. ACM, July 2005.