Model checking dynamic epistemics in branching time

H.P. van Ditmarsch[1], J. Ruan[2], and W. van der Hoek[2]

[1] University of Otago, New Zealand & IRIT, Toulouse, hans@cs.otago.ac.nz
[2] University of Liverpool, United Kingdom, {jruan,wiebe}@csc.liv.ac.uk

Abstract. We give a relation between a logic of knowledge and change, with a semantics on Kripke models, and a logic of knowledge and time, with a semantics on interpreted systems. More in particular, given an *epistemic state* (pointed multi-agent Kripke model where all accessibility relations are equivalence relations) and a formula in *action model logic* (a logic describing the consequences of epistemic events), we construct an *interpreted system* relative to that epistemic state and that formula that satisfies the translation of the formula into *branching temporal epistemic logic*. The construction involves that the *protocol* that is implicit in the dynamic epistemic formula, i.e. the set of sequences of events being executed to evaluate the formula, is made explicit. For presentation reasons, we focus on the logic of knowledge and change that is known as public announcement logic, which can be seen as a specific action model logic. The interpreted system that is constructed in the process is minimal in the sense that it precisely contains all the event structure present in the dynamic epistemic formula. Different approaches to this correspondence can be considered syntactic or semantic sugar. That observation brings applications of our insights closer, because some such approaches originate in the model checking community.

Keywords: modal logic, interpreted system, model checking, action logic, protocol, temporal logic

1 Introduction

Epistemic logic is the formalisation of knowledge. Seminal work in this area is Hintikka's [10], from 1962, and since then many philosophers have been interested in further developing the notions of knowledge and belief using a possible world semantics. In the late 1980s these approaches were picked up and further developed by computer scientists, cf. [9, 5]. This development was originally motivated by the need to reason about communication protocols. One is typically interested in what different parties to a protocol know before, during and after a run (an execution sequence) of the protocol. This interest in change of knowledge over time is already eminent in this area for twenty years. Fagin, Halpern, Moses and Vardi's seminal *Reasoning about Knowledge* [5] is a culmination of several earlier papers in this area, and also incorporates Halpern and Vardi's 1986 paper [9] *The Complexity of Reasoning about Knowledge and Time*.

The central notion in the work of Fagin *et al.* [5] is that of an *interpreted system*. When compared to Kripke (possible worlds) models, interpreted systems have at least two appealing features: a natural and appealing accessibility relation between domain objects, that can be summarised as 'each agent knows its own state', and an equally natural notion of dynamics, modelled by *runs*. The accessibility relation as we know it from the possible worlds model is in this case *grounded*; it has a direct and natural interpretation, as follows. In an interpreted system, the role of possible worlds is performed by global states, which are constituted by the agents' local states and the state of the environment. Each agent knows exactly its own local state: two global states are indistinguishable for an agent if his local compartment is the same. Secondly, an interpreted system defines a number of *runs* through such global states (i.e., a sequence of global states). Each run corresponds to a possible computation allowed by a protocol. In an object language with temporal and epistemic operators one can then express temporal properties such as *liveness* and temporal epistemic properties such as *perfect recall*.

The interpreted systems approach has proven its value far beyond the scope of communication protocols, and temporal epistemic logics that describe them have been studied and applied extensively. Rather than linear time, one may consider branching time logic, and apart from synchrony (roughly, the agents know what the time is) and perfect recall, one may consider properties with or without assuming a unique *initial state*, and with or without the principle of *no learning*. Varying only these parameters already yield 96 different logics: for a comprehensive overview of the linear case we refer to [8], and for the branching time case, to [24]. Moreover, apart from the interpreted systems stance there have been several other and related approaches to knowledge and time, like the distributed processes approach of [15]. The recent paper [23] provides a picture of different logics for knowledge and time from the point of view of decidability and undecidability results for several logics. Apart from computer science, there is much interest in the temporal dynamics of knowledge and belief in areas as diverse as artificial intelligence [13], multiagent systems [18], philosophy [1] and game theory [2].

Dynamic epistemic logic studies what kinds of events are responsible for change of knowledge in a multiagent setting. A quizmaster may publicly announce the winning lot, or whisper it in the ear of his assistant. Both result in a change of knowledge for everybody present, although the change is different in either case. Where belief revision [1] is interested in describing the effect of expansion, contraction and revision of a belief set of one agent, dynamic epistemic logic treats all of knowledge, higher-order knowledge, and its dynamics on the same level, and it gives a fine-tuned analysis of the *way* the revision is brought about, ranging from a private insight by one agent to a public announcement in a group. Unlike temporal epistemic logics, where the meaning of a temporal shift only appears from the underlying model, in dynamic epistemic logic this change is specified 'directly' in the dynamic operators. Starting with a few somewhat isolated contributions in the late 1980s [16, 20], the area strongly developed from the late 1990s onward [7, 3, 25]. A general theory only now and partially emerges. We will base our treatment of dynamic epistemic logic on [25].

The presented frameworks interact both on the level of logical languages and on the level of semantic objects—and it is precisely this interaction that is the subject of the underlying investigation. Various results have already been achieved. The relation between Kripke models and interpreted systems has been investigated by Lomuscio and Ryan in [12]. Their approach suits Kripke models where all states have different valuations, which is not generally the case. A recent study by Pacuit [14] compares the history-based approach by Parikh and Ramanujam [15] to interpreted systems, with runs. This addresses the relation between Kripke models *with histories consisting of event sequences* (in our case this primitive is derived and called a tree model) and interpreted systems. Pacuit handles *partial* observability of agents, when they perceive only some but not all of a sequence of events. He does not address in [14] the partial observability common in dynamic epistemics, where only an aspect of an event is observable. Other recent work by van Benthem, Gerbrandy and Pacuit [23], rooted in older work [22, 21], gives a precise relation between temporal epistemics and dynamic epistemics. In their approach, each event (M, w) (an action model) corresponds to a unique *labelled* modality $\bigcirc_{(\mathsf{M},\mathsf{w})}$, interpreted in a linear temporal logic, such that a dynamic epistemic formula of the form $[\mathsf{M}, \mathsf{w}]\varphi$ ('after execution of event '(M, w)' it holds that φ') is true in a Kripke model with merely epistemic accessibility relations, if and only if a temporal epistemic formula $\bigcirc_{(\mathsf{M},\mathsf{w})}\varphi$ is true in an 'enlarged' Kripke model that is constructed using two copies of the former and an accessibility relation for $\bigcirc_{(\mathsf{M},\mathsf{w})}$-execution that connects them. This is a *forest* that we will introduce later. We have straightforwardly applied their elegant approach. Unlike them, we do not assume a protocol but compute it based on the structure of a given formula.

Much recent work in model checking is based on temporal epistemics describing interpreted systems (MCMAS [17], MCK [6], and see also [19]), although some recent, very promising, work is based on dynamic epistemics describing Kripke models (DEMO, [27]). In a previous study we addressed the relation between these logics. We encoded public announcement logic into temporal epistemics [26] by way of explicitly introducing boolean state variables for each announcement with values corresponding to unknown (i.e., before the announcement is made), and true (after a truthful announcement). Our current investigation can be seen as a continuation of that effort, that however remains rooted in such practical applications.

Section 2 contains short notes on logical languages and structures involved. Section 3 presents our main results for a specific event logic, namely public announcement logic. We translate all formulas with announcements into a fragment of the logic ETL, and prove a theorem that identifies the truth of such a dynamic epistemic formula in a world of a Kripke model, with the truth of a temporal epistemic formula in a global state of an interpreted system. In Section 4 we provide precise technical details on the generalization of that result to arbitrary event execution.

2 Logical preliminaries

We introduce *three* structural primitives and *two* languages. The structures are: *epistemic models* (plus a somewhat derived primitive called an *event model*), i.e., Kripke models with for each agent an accessibility relation representing his knowledge, *tree models*, that fall under the branch of *epistemic PDL-models* which are Kripke models with not just accessibility relations representing agent knowledge but also accessibility relations for event execution (thus closer to models/graphs used for PDL), and *interpreted systems*. The languages are those of *dynamic epistemic logic* and a fragment of a (branching) *temporal epistemic logic* (that one could think of as 'epistemic next-time logic'). The former can be given meaning both on (purely) epistemic models but also on epistemic PDL-models; the latter both on epistemic PDL-models and on interpreted systems. As global parameters to both the languages and the structures we have a set A of n agents, and a (countable) set P of atoms p. Details on action models and on common knowledge are omitted (see the references in the previous section).

2.1 Languages

The logical language $\mathcal{L}_{\mathsf{del}}$ of action model logic is inductively defined as

$$\varphi ::= p \mid \neg\varphi \mid (\varphi \wedge \varphi) \mid K_i\varphi \mid C_B\varphi \mid [\mathsf{M},\mathsf{w}]\varphi$$

where $p \in P$, $i \in A$, $B \subseteq A$, and (M,w) a pointed action model with the usual constraints. Without loss of generality (and for the convenience of the comparison with PDL-like abstract action names) we assume that all points of all action models are differently named, so that we can associated a particular w with the pointed model (M,w) whenever convenient. For the special case of singleton event models with reflexive access for all agents, i.e. public announcements, we write $[\varphi]\psi$ where φ is the precondition (the announced formula).

The language $\mathcal{L}_{\mathsf{etl}}$ is inductively defined as

$$\varphi ::= p \mid \neg\varphi \mid (\varphi \wedge \varphi) \mid K_i\varphi \mid C_B\varphi \mid \bigcirc_{\mathsf{w}}\varphi$$

where, again, $p \in P$, $i \in A$, $B \subseteq A$, and w the (uniquely named, see above) point of pointed action model (M,w).

2.2 Structures

An *epistemic model* is a structure $\langle W, \sim_1, \dots, \sim_n, \pi\rangle$ where W is a domain of worlds, for each agent i, $\sim_i$ is the binary accessibility relation between worlds, that is an equivalence relation, expressing the worlds that are indistinguishable from each other for that agent, and where $\pi : W \to P \to \{0, 1\}$ is a valuation (or interpretation) that determines for each world which atoms are true or false in that world.

An *epistemic PDL-model* is a structure $\langle W, \sim_1, \ldots, \sim_n, \{\rightarrow_{\mathsf{w}}\}, \pi\rangle$ that is like an epistemic model but *plus* a set of accessibility relations $\rightarrow_{\mathsf{w}}$ expressing the execution of actions w (we assume a background parameter set of events W—each event corresponds to an action model (M, w)). We will in fact use a specific epistemic PDL-model where all worlds have the form $(w, \mathsf{w}^1, \ldots, \mathsf{w}^m)$ where $\mathsf{w}^1, \ldots, \mathsf{w}^m$ is a sequence of executed events.

To formally define an *interpreted system* $\mathcal{I}$ for n agents we first give the notion of a global state. A global state s is a tuple $s = (s_e, s_1, \ldots, s_n)$ where s_e is the state of the environment and where for $i = 1 \ldots n$, s_i is the local state of agent i. The set of global states of interest will be denoted $\mathcal{G}$. A *run* over $\mathcal{G}$ is a sequence of states, or, rather, a function r from time $\mathbb{N}$ to global states. The pair (r, m) consisting of a run and a time point is also referred to as a point. Let $r(m) = s$ be the global state at time m in run r, then with $r_i(m)$ we mean local state s_i. An *interpreted system* $\mathcal{I} = (\mathcal{R}, \pi)$ over $\mathcal{G}$ is a system $\mathcal{R}$ of runs over a set $\mathcal{G}$ of global states with an interpretation π which decides for each point (r, m) and atom $p \in P$, whether p is true in (r, m) or not. Two points (r, m) and (r', m') are indistinguishable for i, written $(r, m) \sim_i (r', m')$, if $r_i(m) = r'_i(m')$.

2.3 Semantics

The most pregnant clauses of the semantics are for knowledge, event execution (for which we only give the example of announcement execution), and the temporal connective. Formula $K_i\varphi$ means 'agent i knows φ'. On epistemic models, and on epistemic PDL-models, its semantics is

$$M, w \models K_i\varphi \quad \text{iff} \quad \text{for all } v \in W : w \sim_i v \text{ implies } M, v \models \varphi$$

This expresses that given a model M and a state w, agent i knows that φ, if φ is the case in every state the agent cannot distinguish from w.

A singleton event model with universal access for all agents represents a public announcement. Formula $[\varphi]\psi$ stands for 'after public announcement of φ it holds that ψ'. Its semantics with respect to an epistemic model M is:

$$M, w \models [\varphi]\psi \quad \text{iff} \quad M, w \models \varphi \text{ implies } M|\varphi, w \models \psi$$

where $M|\varphi$ is the restriction of M to the worlds where φ is true. On the other hand, the semantics of $[\varphi]\psi$ with respect to an epistemic PDL-model M is as follows. Let w be the name for $[\varphi]$ (in other words, w is the point of the singleton action model corresponding to the announcement), and suppose that in the model M all worlds where φ is true have a $\rightarrow_{\mathsf{w}}$ transition (this condition is satisfied in the tree models to be introduced in the next section). Then:

$$M, w \models [\varphi]\psi \quad \text{iff} \quad M, w \models \varphi \text{ implies } \exists v \text{ s.t. } w \rightarrow_{\mathsf{w}} v \text{ and } M, v \models \psi$$

For the temporal operator $\bigcirc_{\mathsf{w}}$ the semantics on epistemic PDL-models is:

$$M, w \models \bigcirc_{\mathsf{w}}\varphi \text{ iff there is a state } v \text{ such that } w \rightarrow_{\mathsf{w}} v \text{ and } M, v \models \varphi$$

whereas the semantics for this expression on interpreted systems is formulated in terms of runs through the system. Let $\mathcal{I} = (\mathcal{R}, \pi)$ be an interpreted system over a set $\mathcal{G}$ of global states. "Runs r and r' are equivalent to time m" means that the initial segments of r and r' are the same from 0 to m, i.e., $r(0) = r'(0)$ up to $r(m) = r'(m)$. Choosing the bundle semantics as in [24] we now define that:

$(\mathcal{I}, r, m) \models \bigcirc_{\mathsf{w}}\varphi$ iff there is a run r' that is equivalent to r to time m and $r'(m+1)$ is an w-successor of $r(m)$ such that: $(\mathcal{I}, r', m+1) \models \varphi$.

The semantics of knowledge of interpreted systems expresses that global states are indistinguishable if they are at the same point in time (we assume synchronicity) and if the local states of that agent are the same:

$(\mathcal{I}, r, m) \models K_i\varphi$ iff for all runs r' such that $r'_i(m) = r_i(m)$: $(\mathcal{I}, r', m) \models \varphi$.

3 Public announcements and interpreted systems

Given a formula φ in the language of public announcement logic, and a multi-agent epistemic model (M, w), we want to simulate checking the truth of φ in (M, w) by checking the truth of a corresponding temporal epistemic formula in a corresponding interpreted system. The interpreted system is based on (M, w) but should also encode the dynamics that is implicitly present in φ in the form of public announcement operators. It is therefore relative to both φ and (M, w). In other words, we are looking for a *semantic transformation* F and a *syntactic translation* TRS such that: $M, w \models \varphi$ iff $\mathrm{F}((M, w), \varphi) \models \mathrm{TRS}(\varphi)$. The image of actual world w under F (a global state s_w) is entirely determined by the role of w in M. It is therefore sufficient to determine $\mathrm{F}(M, \varphi)$:

$$M, w \models \varphi \quad \text{iff} \quad \mathrm{F}(M, \varphi), s_w \models \mathrm{TRS}(\varphi)$$

3.1 Syntactic translation

The dynamic epistemic formula translates to a temporal epistemic one in a simple way. All static features remain the same. For the dynamic feature, consider a formula $[\varphi]\psi$ containing the announcement of φ. Each announcement should correspond precisely one temporal transition, and that is expressed in a next temporal operator labelled by an event corresponding to that announcement.

Definition 1 (Dynamic epistemic to temporal epistemic). *All clauses are trivial except the case for public announcement. Suppose that the event* w

corresponds to announcement modality $[\varphi]$*:*

$$\begin{array}{ll} \textsc{trs}(p) & \equiv p \\ \textsc{trs}(\varphi \wedge \psi) & \equiv \textsc{trs}(\varphi) \wedge \textsc{trs}(\psi) \\ \textsc{trs}(\neg\varphi) & \equiv \neg\textsc{trs}(\varphi) \\ \textsc{trs}(K_i\varphi) & \equiv K_i\textsc{trs}(\varphi) \\ \textsc{trs}(C_B\varphi) & \equiv C_B\textsc{trs}(\varphi) \\ \textsc{trs}([\varphi]\psi) & \equiv \textsc{trs}(\varphi) \rightarrow \bigcirc_{\mathsf{w}}\textsc{trs}(\psi) \end{array}$$

The translation is to a language fragment of ETL.

Example 1. Given a public announcement formula $[p][q](p\wedge q)\wedge[q]K_1q$. Suppose w' corresponds to $[p]$, w'' to first $[q]$, and w''' to second $[q]$. This formula is then translated as $(p \rightarrow \bigcirc_{\mathsf{w}'}(q \rightarrow \bigcirc_{\mathsf{w}''}(p \wedge q))) \wedge (q \rightarrow \bigcirc_{\mathsf{w}'''}K_1q)$.

Protocol The dynamics implicitly present in φ, that plays a role in $\textsc{f}((M,w),\varphi)$, can be *identified* with the set of all sequences of public announcements that may need to be evaluated in order to determine the truth of φ. As this is known as a protocol [14], we call this the *protocol of a formula* φ. It can be determined from φ and is therefore another syntactic feature that we can address before applying it in the semantic transformation $\textsc{f}((M,w),\varphi)$.

To determine the protocol, we must be able to distinguish an announcement in a formula from all other announcements occurring in that formula. For example, in $[p][q](p\wedge q)\wedge[q]K_1q$ there are two announcements of q, but they have a different dynamic effect. We distinguish the announcements in a formula from one another by giving them different names. We name the n announcements occurring in a formula $\mathsf{w}^1,\ldots,\mathsf{w}^n$ in the order of occurrence of their left '[' bracket, when reading the formula from left to right. Note that this is unambiguous. The names will also serve as (components of) local state values in the interpreted system to be constructed later.

Example 2. The three announcements in the formula $[p][q](p\wedge q)\wedge[q]K_1q$ correspond to three state variables $\mathsf{w}',\mathsf{w}'',\mathsf{w}'''$. For a less trivial example, the three announcements in the formula $[p\wedge[q]K_2q]C_{12}p\wedge[\top]\neg K_1p$ are associated with three announcement state variables as follows:

$$\begin{array}{ll} \mathsf{v} & [p\wedge[q]K_2q] \\ \mathsf{v}' & [q] \\ \mathsf{v}'' & [\top] \end{array}$$

The left bracket '[' in '$[p\wedge\ldots$' comes first when reading from left to right. Then comes the left bracket of the announcement $[q]$ that is a subformula of $[p\wedge[q]K_2q]$. Finally we reach the announcement $[\top]$ in the right-hand side of the conjunction.

Definition 2 (Protocol of a formula). *The* protocol of a formula *is defined by induction on formula structure. In the last clause,* w *is the name for* *the announcement of* φ *in* $[\varphi]\psi$*, and* $\text{wPROT}(\psi) = \{\mathsf{w}\mathsf{w}^1 \dots \mathsf{w}^m \mid \mathsf{w}^1 \dots \mathsf{w}^m \in \text{PROT}(\psi)\}$*, i.e. the concatenation of* w *to all sequences in the set of* $\text{PROT}(\psi)$*.*

$$
\begin{aligned}
\text{PROT}(p) &\equiv \emptyset \\
\text{PROT}(\neg\varphi) &\equiv \text{PROT}(\varphi) \\
\text{PROT}(\varphi \wedge \psi) &\equiv \text{PROT}(\varphi) \cup \text{PROT}(\psi) \\
\text{PROT}(K_i\varphi) &\equiv \text{PROT}(\varphi) \\
\text{PROT}(C_B\varphi) &\equiv \text{PROT}(\varphi) \\
\text{PROT}([\varphi]\psi) &\equiv \text{PROT}(\varphi) \cup \text{wPROT}(\psi)
\end{aligned}
$$

This notion of protocol is similar to in [14]. The difference is that in our case $\text{PROT}(\psi)$ is not subsequence closed. The protocol of a formula would be subsequence closed if the last clause is changed to $\text{PROT}([\varphi]\psi) \equiv \text{PROT}(\varphi) \cup \text{wPROT}(\psi) \cup \{\mathsf{w}\}$. For a protocol variable we use $\mathcal{T}$.

Example 3. We have that $\text{PROT}([p][q](p \wedge q) \wedge [q]K_1q) = \{\mathsf{w}'\mathsf{w}'', \mathsf{w}'''\}$, and that $\text{PROT}([p \wedge [q]K_2q]C_{12}p \wedge [\top]\neg K_1p) = \{\mathsf{v}, \mathsf{v}', \mathsf{v}''\}$. (See previous example.)

3.2 Semantic transformation

The required semantic transformation F in $\text{F}(M, \varphi)$ is determined in two steps. (As said, the F-image of actual world w is simply its image as a world in the domain of M.) First, we construct the *forest*, a.k.a. *tree model*, $\text{FRST}(M, \text{PROT}(\varphi))$ from the epistemic model M and the protocol $\text{PROT}(\varphi)$ of the dynamic epistemic formula φ [21, 23]. The tree model is an epistemic PDL-model $\langle W, \sim_1, \dots, \sim_n, \{\rightarrow_{\mathsf{w}}\}, \pi\rangle$ (where $\{\rightarrow_{\mathsf{w}}\}$ is a set of accessibility relations, one for each event occurring in φ, such as an announcement). Then we can determine an interpreted system $\text{IS}(M')$ corresponding to a tree model M'. We then simply define $\text{F}(M, \varphi)$ as $\text{IS}(\text{FRST}(M, \text{PROT}(\varphi)))$.

Definition 3 (Epistemic model to tree model). *Given epistemic model* $M = \langle W, \sim_1, ..., \sim_n, \pi\rangle$*,* $w \in W$*, and a protocol* $\mathcal{T}$*. Let* φ *be the announcement named* w*. The tree model* $\text{FRST}(M, \mathcal{T})$ *is defined by*

$$
\begin{aligned}
\text{FRST}(M, \emptyset) &\equiv M \\
\text{FRST}(M, \mathcal{T} \cup \mathcal{T}') &\equiv \text{FRST}(M, \mathcal{T}) \cup \text{FRST}(M, \mathcal{T}') \\
\text{FRST}(M, \{\mathsf{w}^1 \dots \mathsf{w}^m\mathsf{w}\}) &\equiv \textit{from } \text{FRST}(M, \{\mathsf{w}^1 \dots \mathsf{w}^m\}) \textit{ and } \mathsf{w} \textit{ as below}
\end{aligned}
$$

For the iteration in the last clause we show the first step. Tree model $\text{FRST}(M, \{\mathsf{w}\}) = M'$ *with* $M' = \langle W', \sim_1, \dots, \sim_n, \{\rightarrow_{\mathsf{w}}\}, \pi\rangle$ *is defined as*

$$
\begin{aligned}
&W' = W \cup \{(w, \mathsf{w}) \mid M, w \models \varphi\} \\
&w \sim'_i w' \textit{ iff } w \sim_i w' \textit{ and } (w, \mathsf{w}) \sim'_i (w', \mathsf{w}) \textit{ iff } w \sim_i w' \\
&w \rightarrow_{\mathsf{w}} (w, \mathsf{w}) \\
&\pi'(w)(p) = \pi(w)(p) \textit{ and } \pi'((w, \mathsf{w}))(p) = \pi(w)(p)
\end{aligned}
$$

The construction can be seen as repeatedly merging a model and the restricted modal product of that model and a singleton 'action model' corresponding to the announcement [3].

Next, from such a tree model we can determine an interpreted system. This is based on a fairly simple idea (that however we have not encountered in this form). For each *world* in a Kripke model we need to find a corresponding *global state*. This can be achieved keeping that world as the value of the *environmental state* and for each agent the set of indistinguishable worlds as the value of that agent's *local state*. The valuation π remains as it was. Now for a world w in an epistemic model $M = \langle W, \sim_1, \ldots, \sim_n \rangle$ this recipe delivers a corresponding global state $s = (w, \widetilde{w}_1, \ldots, \widetilde{w}_n)$. (Partly for its pretty layout we write $\widetilde{w}_i$ for the i-equivalence class containing w, i.e. $\{w' \in W \mid w' \sim_i w\}$. Alternatively, we write $w^{\sim_i}$.) The same recipe applies, in principle, to worlds $(w, \mathsf{w}^1, \ldots, \mathsf{w}^m)$ in the tree model $\textsc{frst}(M, \textsc{prot}(\varphi))$, but here we can somewhat simplify matters by observing that (i) the environment is fully determined by the w in $(w, \mathsf{w}^1, \ldots, \mathsf{w}^m)$ because all events (such as announcements) are defined relative to their combined effect on the agents *only*, and by observing that (ii) public announcements are fully observable by all agents so we can represent them as global parameters. In the following we use $(w, \widetilde{w}_1, \ldots, \widetilde{w}_n, \mathsf{w}^1, \ldots, \mathsf{w}^m)$ to denote the global state $((w, \mathsf{w}^1, \ldots, \mathsf{w}^m), (w, \mathsf{w}^1, \ldots, \mathsf{w}^m)^{\sim_1}, \ldots, (w, \mathsf{w}^1, \ldots, \mathsf{w}^m)^{\sim_n})$.

Given that each global state has some world as its environmental state, and given that the environment remains constant throughout a run, we tend to label global states and runs with that world: write s_w and r_w, respectively, for global and runs with w as environmental state. Incidentally, note that we cannot do away with the environmental state altogether: consider an epistemic model with universal access for all agents but where facts are true in some worlds but false in other worlds. Global states corresponding to such worlds have identical local states for all agents *but* are only different in their environmental state value (that represents the value of facts).

Definition 4 (Tree model to interpreted system). *To a given tree model* $M = \langle W, \sim_1, ..., \sim_n, \{\rightarrow_{\mathsf{w}}\}, \pi \rangle$ *corresponds to the interpreted system* $\mathcal{I} = (\mathcal{R}, \pi)$ *over* $\mathcal{G}$ *as follows. Every world* $(w, \mathsf{w}^1, \ldots, \mathsf{w}^m)$ *in the tree model corresponds to a global state* $(w, \widetilde{w}_1, \ldots, \widetilde{w}_n, \mathsf{w}^1, \ldots, \mathsf{w}^m)$ *in* $\mathcal{G}$*, where the local state of agent* i *is* $(\widetilde{w}_i, \mathsf{w}^1, \ldots, \mathsf{w}^m)$*. The set* $\mathcal{R}$ *consists of runs* r_w *with*

- $r_w(0) = (w, \widetilde{w}_1, \ldots, \widetilde{w}_n)$;
- $r_w(i{+}1) = (w, \widetilde{w}_1, \ldots, \widetilde{w}_n, \mathsf{w}^1, \ldots, \mathsf{w}^{i+1})$ *iff* $(w, \widetilde{w}_1, \ldots, \widetilde{w}_n, \mathsf{w}^1, \ldots, \mathsf{w}^i) \rightarrow_{\mathsf{w}^{i+1}} (w, \widetilde{w}_1, \ldots, \widetilde{w}_n, \mathsf{w}^1, \ldots, \mathsf{w}^i, \mathsf{w}^{i+1})$ *and* $r_w(i) = (w, \widetilde{w}_1, \ldots, \widetilde{w}_n, \mathsf{w}^1, \ldots, \mathsf{w}^i)$;
- $r_w(i + 1) = r_w(i)$*, otherwise.*

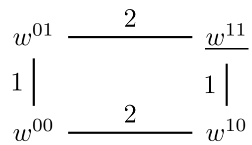
The valuations correspond: for arbitrary r*, world* w*, and time* i*,* $\pi(r_w(i))(p) = \pi(w)(p)$*.*

We now define the interpretation of DEL formula over interpreted system more precisely:

$(\mathcal{I}, r, m) \models \bigcirc_{\mathsf{w}}\varphi$ iff there is a run r' that is equivalent to r to time m and $r'(m) \rightarrow_{\mathsf{w}} r'(m+1)$ such that: $(\mathcal{I}, r', m+1) \models \varphi$.

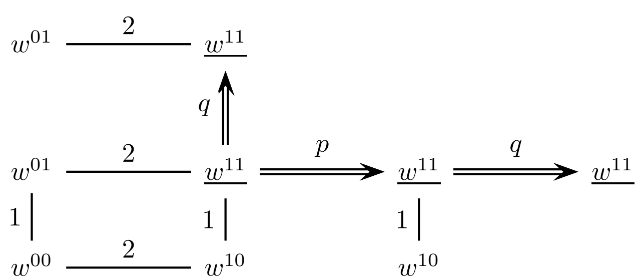
3.3 Example

Consider two agents 1 and 2 and two facts p and q. Agent 1 knows whether p but is uncertain about the truth of q, whereas agent 2 knows whether q but is uncertain about the truth of p. The agents are commonly aware of each other's factual knowledge and ignorance. In fact, both p and q are true. This is modelled by

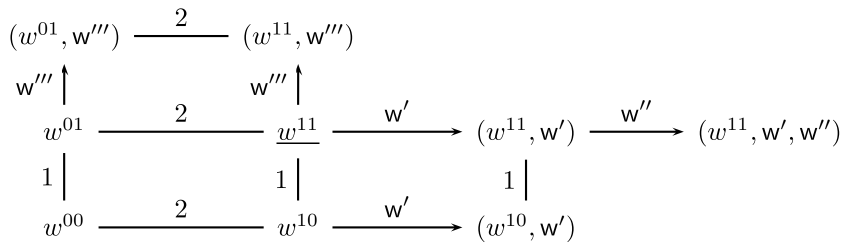


We have named the four worlds of the model w^{00}, w^{10}, w^{01}, and w^{11}, where the index reveals the valuation of atoms p and q (in that order). Agent 1 cannot distinguish worlds w^{00}, w^{01} from one another, therefore they are linked and the link is labelled with a 1. Etc. The actual world is underlined.

We now want to check the truth of formula $[p][q](p \wedge q) \wedge [q]K_1 q$ in state w^{11} of the model. First, consider p. An announcement of p results in a model restriction where the $\neg p$-worlds w^{00} and w^{01} are eliminated. In the resulting model, agent 1 is still uncertain about q, but agent 2 now knows that p. We can visualize such a state transition by an $\Rightarrow$-arrow between the epistemic state before the announcement and the epistemic state resulting from the announcement. To distinguish this from other announcements we label $\Rightarrow$ with that announcement. After the announcement of p, given $[p][q](p \wedge q)$, atom q is subsequently announced, resulting in another state transition. But q could also initially have been announced, namely in the right-hand part $[q]K_1 q$ of the initial formula. Depicting all three announcements at the same time, we get



The tree model equivalent of this model plus announcements is as follows. The epistemic state transitions induced by announcements are now accessibility relations within a *single* epistemic PDL-model, where we have to take into account that other transitions are indistinguishable from the above three. Please recall that the event variables $\mathsf{w}', \mathsf{w}''$, and w''' represent the three different announcements.

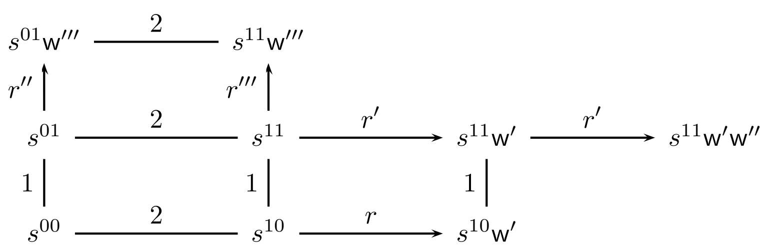


This forest (tree model) consists of four trees: the one with root w^{11} corresponds essentially to the previous picture. The other trees have roots w^{00},w^{01}, and w^{10}. Somewhat arbitrarily we have chosen the root w^{11} to be the 'actual world' of the model; this represents the state of the system where none of the announcements have yet been made.

The interpreted system corresponding to the initial epistemic state and the formula to be evaluated is of course utterly similar to the tree model just given. The world w^{10} corresponds to the global state $(w^{10}, \{w^{10}, w^{11}\}, \{w^{00}, w^{10}\})$, and the world (w^{10}, w') corresponds to the global state $(w^{10}, \{w^{10}, w^{11}\}, \{w^{10}\}, \mathsf{w}')$, (in other words: $(w^{10}, \{(w^{10}, \mathsf{w}'), (w^{11}, \mathsf{w}')\}, \{(w^{10}, \mathsf{w}')\}))$, etc. Write s^{10} for the former global state and $s^{10}\mathsf{w}'$ for the latter. The accessibility relations for agent 1 and 2 remain the same. Instead of event-labelled transitions we now have runs connecting the global states. There are four non-trivial runs, (arbitrarily) named as

$$\begin{array}{ll} r & (s^{10}, s^{10}\mathsf{w}') \\ r' & (s^{11}, s^{11}\mathsf{w}', s^{11}\mathsf{w}'\mathsf{w}'') \\ r'' & (s^{01}, s^{01}\mathsf{w}''') \\ r''' & (s^{11}, s^{11}\mathsf{w}''') \end{array}$$

The interpreted system can now be depicted as



The translation TRS of the formula $[p][q](p \wedge q) \wedge [q]K_1 q$, was, as we have already seen, $(p \rightarrow \bigcirc_{\mathsf{w}'}(q \rightarrow \bigcirc_{\mathsf{w}''}(p \wedge q))) \wedge (q \rightarrow \bigcirc_{\mathsf{w}'''} K_1 q)$, which is clearly true in state s^{11} of the interpreted system.

3.4 Theoretical results

We now show the correspondence $M, w \models \varphi$ iff $\text{F}(M, \varphi), s_w \models \text{TRS}(\varphi)$ by way of a few intermediate results. Proof details are fairly elementary and omitted.

Proposition 1 (Baltag et al., Venema, van Benthem, Pacuit). *Let M be an epistemic model and $\varphi \in \mathcal{L}_{\mathsf{del}}$.*

$$M, w \models \varphi \quad \textit{iff} \quad \textsc{frst}(M, \textsc{prot}(\varphi)), w \models \varphi$$

This is a known result, expressing that we can either evaluate dynamic modal operators by pointed Kripke model transformation, or alternatively construct a 'supermodel' that already contains all future dynamic developments as submodels. Our formulation, relative to a formula φ to be evaluated, is slightly different from the standard purely semantic form. For a description of the technique see Venema's chapter 'Dynamic Models in their Logical Surroundings' in [22, page 122], or [21, 23].

The next result says that on tree models that have the dynamic transitions in the right place (namely $\rightarrow_{\mathsf{w}}$ transitions exactly when φ is true, where w is the name for φ) a dynamic epistemic formula φ is equivalent to its translation $\textsc{trs}(\varphi)$.

Proposition 2.

$$\textsc{frst}(M, \textsc{prot}(\varphi)), w \models \varphi \leftrightarrow \textsc{trs}(\varphi)$$

The proof is by induction on φ. The only logical connectives affecting the protocol of a formula are conjunction and announcement, and for all but announcement the clauses of $\textsc{trs}$ are trivial. We therefore only have to consider the cases conjunction and announcement. An essential proof detail for the clause conjunction $\varphi \wedge \psi$ is, that it can be shown that $\textsc{frst}(M, \textsc{prot}(\varphi)), w \models \varphi$ iff (for all ψ : $\textsc{frst}(M, \textsc{prot}(\varphi \wedge \psi)), w \models \varphi$). The proof uses that tree model $\textsc{frst}(M, \textsc{prot}(\varphi))$ is a submodel of tree model $\textsc{frst}(M, \textsc{prot}(\varphi \wedge \psi))$.

Proposition 3. *For every* executable $\varphi \in \mathcal{L}_{\mathsf{etl}}$ *(i.e., a formula of the form* $\textsc{trs}(\psi)$*), and tree model M:*

$$M, w \models \varphi \quad \textit{iff} \quad \textsc{is}(M), (w, \widetilde{w}_1, \ldots, \widetilde{w}_n) \models \varphi$$

We emphasize that Proposition 3 does not hold for arbitrary formulas in our temporal epistemic fragment, because of the observed slight but essential difference between tree models, where event chains are finite, and corresponding interpreted systems, with infinite runs. More precisely: in case $\varphi \rightarrow \bigcirc_{\mathsf{w}}\psi$ of the proof of Proposition 3 the precondition φ is essential! Worlds in tree models do not necessarily have an event successor, so that in such worlds all formulas of form $\bigcirc_{\mathsf{w}}\psi$ are false, whereas runs in interpreted systems keep looping after a finite meaningful prefix, such that (e.g.) $\bigcirc_{\mathsf{w}}p$ will always remain true if p is true.

We now have the main result in our pocket, directly from Propositions 1, 2, and 3—note that $\textsc{f}(M, \varphi)$ is by definition $\textsc{is}(\textsc{frst}(M, \textsc{prot}(\varphi)))$.

Theorem 1.

$$M, w \models \varphi \quad \textit{iff} \quad \textsc{f}(M, \varphi), s_w \models \textsc{trs}(\varphi)$$

Finally, we come to the minimality of our construction. Let 'M is minimal' mean that it is a bisimulation contraction, a.k.a. strongly extensional [4]. Informally, a model is minimal if it cannot contain fewer worlds without changing its information content.

Proposition 4. *Assume* $M, w \models \varphi$*. If* M *is minimal, then* $\textsc{frst}(M, \textsc{prot}(\varphi))$ *is minimal. If we remove (at least) a pair from a dynamic relation* $\rightarrow_{\mathsf{w}}$ *in* $\textsc{frst}(M, \textsc{prot}(\varphi))$*, then* $\textsc{frst}(M, \textsc{prot}(\varphi)), w \not\models \textsc{trs}(\varphi)$*.*

Obviously, a similar result holds for M and $\textsc{f}(M, \varphi)$.

3.5 Notational variants and model checking

Theorem 1 provides a way to model check dynamic epistemic formulas in temporal epistemic model checkers 'directly'. The temporal epistemic checkers mentioned [17, 6, 19] have all in common that: their semantics is on interpreted systems, and that: they optimize search by implementing binary decision diagrams (BDD, see [11] for an introduction). This requires *boolean* local state variables. Our presentation so far was on a rather 'abstract' level where local states took structured forms such as $(w, \mathsf{w}, \mathsf{v})$, thus providing a convenient representation for the comparison with dynamic logics. Now there are simple and standard ways to convert ('compile') any such multi-valued formalism into a binary one, unrelated to the accidental structure of that formalism, and hidden from the modeller. But there is also a particular way that uses the structure rather well, keeps the result intelligable to the modeller / (human) model checker, and is more or less enforced, in our opinion, by these currently available temporal epistemic model checkers. We introduce it informally.

In a typical initial epistemic model to undergo dynamic developments, all worlds are factually distinct. (Typical is initial uncertainty for all agents between all worlds. A domain of worlds with the universal relation between them for all agents is bisimilar to that domain where all worlds having the same valuation are identified.) In that case a world can be represented a binary array representing which facts are true and false in that world.

Now assume a world w and a world (w, w) expressing the result of announcing some φ with name w in that world. Alternatively, we can see w as a binary state variable with value 0 (for 'unknown') in the initial state of the system and that gets value 1 (for 'known and true') in the global state corresponding to (w, w). In other words: all announcement variable names are present from the initial state of the system onwards.

We apply these basic simplifications to the run $r = (s^{10}, s^{10}\mathsf{w}')$ described in the example of subsection 3.3. Formally, s^{10} was $(w^{10}, \{w^{10}, w^{11}\}, \{w^{10}, w^{00}\})$ and $s^{10}\mathsf{w}'$ was $(w^{10}, \{w^{10}, w^{11}\}, \{w^{10}, w^{00}\})$. Instead, we now get $r = (10.000, 10.100)$. The part '10.' contains the values of atoms p and q, the part (e.g.) '100' contains the values of the announcement variables; note that w' was the first of three state variables, the others remain irrelevant in this particular run. This is the approach applied in [26].

4 Action models and interpreted systems

The approach in the previous section is straightforwardly generalized from public announcements as in $[\varphi]\psi$ to action models as in $[\mathsf{M},\mathsf{w}]\psi$. To translate formulas in action model logic, replace clause $\text{TRS}([\varphi]\psi) \equiv \text{TRS}(\varphi) \rightarrow \bigcirc_{\mathsf{w}}\text{TRS}(\psi)$ in Definition 1 on page 6 by

$$\text{TRS}([\mathsf{M},\mathsf{w}]\psi) \equiv \text{TRS}(\mathsf{pre}(\mathsf{w})) \rightarrow \bigcirc_{\mathsf{w}}\text{TRS}(\psi)$$

The clause for public announcement is now a special case, as the precondition of a singleton action model with universal access is the announcement formula. The state variables that we associate with announcements can now simply be the event names — where we assume w.l.o.g. that in a given formula all names are different. The clause $\text{PROT}([\varphi]\psi) \equiv \text{PROT}(\varphi) \cup \mathsf{w}\text{PROT}(\psi)$ in Definition 2 on page 7 of the protocol of a formula now becomes

$$\text{PROT}([\mathsf{M},\mathsf{w}]\psi) \quad \equiv \quad \bigcup_{\mathsf{v}\in\mathcal{D}(\mathsf{M})} (\text{PROT}(\mathsf{pre}(\mathsf{v}) \cup \mathsf{v}\text{PROT}(\psi))$$

where $\mathcal{D}(\mathsf{M})$ is the domain of the action model, which includes the point w. There are no changes at all in the construction of the tree model from a given protocol.

The interpreted system is somewhat different, as the events are no longer accessible to all agents. In the case of sequences of events corresponding to public announcements, every world $(w, \mathsf{w}^1, \dots, \mathsf{w}^m)$ in the tree model corresponds to a global state $(w, \widetilde{w}_1, \dots, \widetilde{w}_n, \mathsf{w}^1, \dots, \mathsf{w}^m)$ in $\mathcal{G}$, where the local state of agent i is $(\widetilde{w}_i, \mathsf{w}^1, \dots, \mathsf{w}^m)$. Now, instead we have that every world $(w, \mathsf{w}^1, \dots, \mathsf{w}^m)$ in the tree model corresponds to a global state

$$(w, \widetilde{w}_1, \dots, \widetilde{w}_n, \widetilde{\mathsf{w}^1}_1, \dots, \widetilde{\mathsf{w}^1}_n, \dots, \widetilde{\mathsf{w}^m}_1, \dots, \widetilde{\mathsf{w}^m}_n)$$

in $\mathcal{G}$, where the local state of agent i is $(\widetilde{w}_i, \widetilde{\mathsf{w}^1}_i, \dots, \widetilde{\mathsf{w}^m}_i)$.

The main Theorem 1 still holds for the generalization—nowhere in the proof it was essential that events corresponded to public announcements. (And the minimality results also hold unchanged.) So, again:

$$M, w \models \varphi \quad \text{iff} \quad \text{F}(M, \varphi), s_w \models \text{TRS}(\varphi)$$

5 Conclusions

Given an epistemic state and a formula in action model logic, we constructed an interpreted system relative to that epistemic state and that formula that satisfied the translation of the formula into temporal epistemic logic, and vice versa. The construction involved the protocol implicitly present in the dynamic epistemic formula, i.e. the set of sequences of events being executed to evaluate the formula. The interpreted system that is constructed in the process is minimal in the sense that it precisely contains all the event structure present in the dynamic epistemic formula. We hope our results are useful for the model checking community.

References

1. C.E. Alchourrón, P. Gärdenfors, and D. Makinson. On the logic of theory change: partial meet contraction and revision functions. *Journal of Symbolic Logic*, 50:510–530, 1985.
2. R.J. Aumann and A. Brandenburger. Epistemic conditions for nash equilibrium. *Econometrica*, 63:1161–1180, 1995.
3. A. Baltag, L.S. Moss, and S. Solecki. The logic of public announcements, common knowledge, and private suspicions. In I. Gilboa, editor, *Proceedings of the 7th Conference on Theoretical Aspects of Rationality and Knowledge (TARK 98)*, pages 43–56, 1998.
4. P. Blackburn, M. de Rijke, and Y. Venema. *Modal Logic*. Cambridge University Press, Cambridge, 2001. Cambridge Tracts in Theoretical Computer Science 53.
5. R. Fagin, J.Y. Halpern, Y. Moses, and M.Y. Vardi. *Reasoning about Knowledge*. MIT Press, Cambridge MA, 1995.
6. P. Gammie and R. van der Meyden. MCK: Model checking the logic of knowledge. In R. Alur and D. Peled, editors, *Proceedings of the 16th International Conference on Computer Aided Verification (CAV 2004)*, pages 479–483. Springer, 2004.
7. J.D. Gerbrandy and W. Groeneveld. Reasoning about information change. *Journal of Logic, Language, and Information*, 6:147–169, 1997.
8. J.Y. Halpern, R. van der Meyden, and M.Y. Vardi. Complete axiomatizations for reasoning about knowledge and time. *SIAM Journal on Computing*, 33(3):674–703, 2004.
9. J.Y. Halpern and M.Y. Vardi. The complexity of reasoning about knowledge and time. In *Proceedings 18th ACM Symposium on Theory of Computing*, pages 304–315, 1986.
10. J. Hintikka. *Knowledge and Belief*. Cornell University Press, Ithaca, NY, 1962.
11. M.R.A. Huth and M.D. Ryan. *Logic in Computer Science – Modelling and Reasoning about Systems*. Cambridge University Press, Cambridge, UK, 2000.
12. A.R. Lomuscio and M.D. Ryan. On the relation between interpreted systems and kripke models. In C. Zhang M. Pagnucco, W.R. Wobcke, editor, *Proceedings of the AI97 workshop on the theoretical and practical foundations of intelligent agents and agent-oriented systems*, pages 46–59, Berlin, 1998. Springer. Lecture Notes in Artificial Intelligence 1441.
13. R.C. Moore. A formal theory of knowledge and action. In J.F. Allen J., Hendler, and A. Tate, editors, *Readings in Planning*, pages 480–519. Morgan Kaufmann Publishers, San Mateo, CA, 1990.
14. E. Pacuit. Some comments on history-based structures. To appear, 2007.
15. R. Parikh and R. Ramanujam. Distributed processing and the logic of knowledge. In *Logic of Programs*, volume 193 of *Lecture Notes in Computer Science*, pages 256–268. Springer, 1985. A newer version appeared in *Journal of Logic, Language and Information*, vol. 12, 2003, pp. 453–467.
16. J.A. Plaza. Logics of public communications. In M.L. Emrich, M.S. Pfeifer, M. Hadzikadic, and Z.W. Ras, editors, *Proceedings of the 4th International Symposium on Methodologies for Intelligent Systems: Poster Session Program*, pages 201–216. Oak Ridge National Laboratory, 1989. ORNL/DSRD-24.
17. F. Raimondi and A.R. Lomuscio. Verification of multiagent systems via ordered binary decision diagrams: An algorithm and its implementation. In *Proceedings of the Third International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS 04)*, pages 630–637. IEEE Computer Society, 2004.

18. A.S. Rao and M.P. Georgeff. Modeling rational agents within a BDI-architecture. In James Allen, Richard Fikes, and Erik Sandewall, editors, *Proceedings of the 2nd International Conference on Principles of Knowledge Representation and Reasoning (KR'91)*, pages 473–484, San Mateo (CA), 1991. Morgan Kaufmann.
19. K. Su. Model checking temporal logics of knowledge in distributed systems. In D.L. McGuinness and G. Ferguson, editors, *Proceedings of the Nineteenth National Conference on Artificial Intelligence (AAAI 04)*, pages 98–103. AAAI Press / The MIT Press, 2004.
20. J.F.A.K. van Benthem. Semantic parallels in natural language and computation. In *Logic Colloquium '87*, Amsterdam, 1989. North-Holland.
21. J.F.A.K. van Benthem. Games in dynamic epistemic logic. *Bulletin of Economic Research*, 53(4):219–248, 2001.
22. J.F.A.K. van Benthem, P. Dekker, J. van Eijck, M. de Rijke, and Y. Venema. *Logic in Action*. ILLC, Amsterdam, 2001.
23. J.F.A.K. van Benthem, J.D. Gerbrandy, and E. Pacuit. Merging frameworks for interaction: Del and etl. In D. Samet, editor, *Proceedings of TARK 2007*, 2007.
24. R. van der Meyden and K. Wong. Complete axiomatisations for reasoning about knowledge and branching time. *Studia Logica*, 75(1):93–123, 2003.
25. H.P. van Ditmarsch, W. van der Hoek, and B.P. Kooi. *Dynamic Epistemic Logic*, volume 337 of *Synthese Library*. Springer, 2007.
26. H.P. van Ditmarsch, W. van der Hoek, R. van der Meyden, and J. Ruan. Model checking russian cards. *Electronic Notes in Theoretical Computer Science*, 149:105–123, 2006. Presented at MoChArt 05 (Model Checking in Artificial Intelligence).
27. J. van Eijck. Dynamic epistemic modelling. Technical report, Centrum voor Wiskunde en Informatica, Amsterdam, 2004. CWI Report SEN-E0424.