In dynamic epistemic logic (Van Ditmarsch et al., 2008) it is customary to use an action frame (Baltag and Moss, 2004; Baltag et al., 1998) to describe different views of a single action. In this article, action frames are extended to add or remove agents, we call these agent-update frames. This can be done selectively so that only some specified agents get information of the update, which can be used to model several interesting examples such as private update and deception, studied earlier by Baltag and Moss (2004); Sakama (2015); Van Ditmarsch et al. (2012). The product update of a Kripke model by an action frame is an abbreviated way of describing the transformed Kripke model which is the result of performing the action. This is substantially extended to a sum-product update of a Kripke model by an agent-update frame in the new setting. These ideas are applied to an AI problem of modelling a story. We show that dynamic epistemic logics, with update modalities now based on agent-update frames, continue to have sound and complete proof systems. Decision procedures for model checking and satisfiability have expected complexity. A sublanguage is shown to have polynomial space algorithms.

O, what a tangled web we weave when first we practise to deceive! Scott (1808)

1 Introduction

Epistemic and doxastic logics have been used to model many information and communication situations in multi-agent domains (Fagin et al., 2004; Meyer and Van Der Hock, 2004). The language is defined on a fixed set of propositional and agent variables and its semantics is given in terms of Kripke models, also known as possible worlds semantics (Hintikka, 1962; Kripke, 1963). Dynamic epistemic logic is used to model knowledge acquisition and belief revision of agents (Van Ditmarsch et al., 2008). Updates in an action framework can represent private as well as public announcements (Plaza, 1989). In artificial intelligence, DEL has been used to model forward progression in epistemic planning (Baral et al., 2022; Bolander, 2017). Extensions of DEL were proposed to formalise false-belief tasks (Bolander, 2018).

In this paper we consider an AI application (Singh and Khemani, 2019) of modelling a story in which agents can be added and deleted. The agent updates can be done selectively, leading to an informational update for only a subset of agents. The set of propositions remains fixed.

With nested modalities it is natural for an agent to have beliefs about other agents, including their existence. A key question is where do the beliefs about a new agent come from. Things get really exciting when fictional agents come into the picture, where their creators themselves do not believe in their agency, but other agents swallow the fiction, like Captain William Martin in Operation Mincemeat conjured up by the Allies in World War II, as in the film of Madden (2001). In this military situation, the beliefs were planted by the Allies. But in a more routine situation, such as when new members join a group on social media or follow a person’s page, what beliefs can be ascribed to them?
Agent-update Models

Baltag et al. (1998) introduced a product update which takes a model before the update, executes an action, and gives a new model after the update. They gave several examples of its use and an axiomatization reducing DEL to epistemic logic EL (without updates). Satisfiability was also reduced to the simpler logic. The complexity of satisfiability was determined by Aucher and Schwarzentruber (2013).

In this paper, the product update operation undergoes a substantial enhancement. It incorporates a sum to accommodate beliefs of new agents and a difference to remove beliefs of deleted agents. Axiomatization and completeness follow DEL techniques. Complexity of satisfiability is preserved. A sublanguage with better complexity is identified.

We show the application of the new framework to modelling a children’s story, The Gruffalo by Donaldson (1999).

Here is an outline of the paper.

Section 2 gives some basic definitions of dynamic epistemic logics, as well as the action frames which our work extends. Section 2.1 discusses related work. In Section 3, we describe our new agent-update frames and show agent-updates with several examples. These show the expressiveness available with our general mechanism of sum-product update. Section 4 applies our definitions to follow the children’s story, which we found explores complex ideas that required ingenuity to deal with technically. We view the ability to model this story as a contribution of the logical formalism. Section 5 generalises the logics of Section 2 to our new updates, and has more traditional theoretical results. We discuss a proof system to derive valid formulas. We prove its soundness and completeness, and argue that the complexity of model checking the logic as well as deciding validity of formulas does not exceed that of dynamic epistemic logic.

2 Background and literature

A Kripke model (Kripke, 1963) can be defined as follows.

**Definition 1** (Kripke model). \( M = (S, \{R_i \mid i \in A\}, V) \), where \( S \) is a set of possible worlds, \( R_i \subseteq S \times S \) are accessibility relations for every agent \( i \in A \), and \( V : Prop \rightarrow 2^S \) is a valuation function assigning states to a proposition. \( sR_i t \) abbreviates \((s, t) \in R_i\) and it means that at a world \( s \), agent \( i \) believes possible that the world may be \( t \).

A pointed Kripke model is written as \((M, s)\) where \( s \in S \) is a designated state.

In the figures, a directed arrow labelled with \( i \) from world \( s \) to world \( t \) depicts \( sR_i t \) and a bidirectional arrow between two worlds, say \( s \) and \( t \), labelled with \( i \), represents arrows for \( sR_i t \) and \( tR_i s \).

We will assume a fixed set of propositions \( Prop \) throughout this article. When used as an input to an algorithm, the size of a Kripke model is the sum of the number of states \( |S| \), the number of agents \( |A| \), the sizes of the accessibility relation \( |R_i| \) of every agent \( i \) and the size of the valuation, presented in some convenient manner such as a bitvector of states for every proposition. The asymptotically dominant component will be the sizes of the relations, which can be quadratic in the number of states. The size of the valuation is linear in the number of states. Thus the input of size \( O(|A||S|^2) \).

The multi-agent scenarios that motivate us will not have all the agents believe in the agency of all the agents at all possible worlds. We consider private and deceptive updates which bring about an asymmetric information change (Section 3). We build our agent-updates into a more general framework of action models as defined by Baltag et al. (1998) and developed by Van Benthem et al. (2006); Van Ditmarsch and Kooi (2008). The definition of postcondition in these papers is a little more general, which is unnecessary for our purposes.

**Definition 2** (Action frame). Let EL be a logical language defined on a finite set of agents \( A \) and a finite set of propositions \( Prop \). We will define this language below in Definition 4.

An action frame \( U = (E, \{O_i \mid i \in A\}, \text{pre}, \text{post}) \) consists of a finite set of events \( E \) and observability relations for each agent: \( O_i \subseteq E \times E \), together with functions \( \text{pre} : E \rightarrow EL \) and \( \text{post} : E \rightarrow \{\text{Prop} \rightarrow \{\top, \bot, \text{no}\}\} \), which assign a precondition and a postcondition for each event, the latter setting, resetting or not changing propositions.

An action frame is the representation of an action as seen by different agents. A pointed action frame \((U, u)\) with \( u \in E \) specifies the semantics of an action which updates a Kripke model, applied at event \( u \) where the precondition \( \text{pre}(u) \) holds.

The size of an action frame \( U \) includes the sizes of the sets \( A \) and \( E \), the observability relations, and the preconditions and postconditions at every event. It is asymptotically dominated by the sizes of its observability relations.

In figures of action frames, a directed arrow labelled with \( i \) from an event \( u \) to an event \( v \) depicts \( uO_i v \), and it means that when event \( u \) occurs agent \( i \) considers event \( v \) occurring. Where required, the precondition and postcondition of an event are shown alongside. Otherwise the event \( u \) is a skip, its precondition can be taken as \( \text{pre}(u) = \top \) and the postcondition can be taken as no change, \( \text{post}(u)(p) = \text{no} \) for all \( p \in \text{Prop} \).
Agent-update Models

The updated model after an action is formalised as a product of a Kripke model with an action frame (Baltag et. al., 1998).

**Definition 3** (Product update). Given a pointed Kripke model \((M, s)\) and a pointed action model \((U, u)\), the resulting pointed Kripke model after the execution of the action, \(M \otimes U\) is defined as \((M', (s, u))\) with \(M' = (S', V', R')\), where

- \(S' = \{(s, u) \mid (M, s) \models \text{pre}(u)\} \cap (S \times E)\)
- \(V'(p) = \{(s, u) \in S' \mid s \in V(p) \text{ and post}(u)(p) = \text{no}\}, \text{or post}(u)(p) = \top\)
- \(R'_a = \{((s, u), (t, v)) \mid sR_{at} u, uO_au\}, \text{for } a \text{ in } A\)

The *ontic* effect of the action is to *set to true* those propositions where the postcondition \(\text{post}(u)(p)\) is \(\top\), *set to false* those propositions where the postcondition \(\text{post}(u)(p)\) is \(\bot\) and *leave unchanged* those where it is \(\text{no}\).

The number of states in the updated Kripke model is included in \(\text{NEXPTIME}\). The inclusion is not known to be proper.

The number of states in the updated Kripke model is the product \(|S||E|\) of the number of states in the starting Kripke model and the number of events in the action model. Its size will be asymptotically dominated by the size of its relations.

Now we define our logical languages.

**Definition 4** (Formulas of languages \(EL, DEL^-\), \(DEL\)). Starting with a set of propositions \(\text{Prop}\) and agents \(A\), the formulas of the logical language \(DEL\) are constructed using the following BNF.

\[
\phi ::= \top \mid p \in \text{Prop} \mid \neg \phi \mid (\phi \land \psi) \mid P_i \phi, \ i \in A \mid (U, u)\phi \mid ((U_1, u_1) \cup \cdots \cup (U_k, u_k))\phi
\]

The sublanguage \(DEL^-\) does not have the union update formulas \(((U_1, u_1) \cup \cdots \cup (U_k, u_k))\phi\). \(EL\) is a sublanguage of \(DEL^-\) which does not have update formulas \((U, u)\phi\).

The modality \(P_i\) is read as ‘agent \(i\) possibly believes \(\phi\)’. The dual modality \(B_i\phi = \neg P_i\neg\phi\) is read as ‘agent \(i\) believes \(\phi\)’. \((U, u)\phi\) is read as *possibly after update* \(U, \phi\) holds. The dual modality is \([U, u]\phi\). Note that \((U, u)\) is a representation of a pointed action model, inside a formula. The size of that action frame is included in the length of the formula.

**Definition 5** (Truth at a world in a model). Given a formula \(\phi\) in language \(DEL\), and a pointed Kripke model \((M, s)\), the assertion that formula \(\phi\) is true at world \(s\) in model \(M\) is abbreviated as \(M, s \models \phi\) and recursively defined as:

- \((M, s) \models \top\),
- \((M, s) \models p \iff s \in V(p)\),
- \((M, s) \models \neg \phi \iff \text{not } (M, s) \models \phi\),
- \((M, s) \models (\phi \land \psi) \iff (M, s) \models \phi \text{ and } (M, s) \models \psi\), and
- \((M, s) \models P_i \phi \iff \text{for some } t, sR_{it} t \text{ and } (M, t) \models \phi\)
- \((M, s) \models [U, u]\phi \text{ iff } (M, s) \models \text{pre}(u) \text{ and } (M \otimes U, (s, u)) \models \phi\).
- \((M, s) \models ((U_1, u_1) \cup \cdots \cup (U_k, u_k))\phi \text{ iff } (M, s) \models \text{pre}(u_i) \text{ and } (M \otimes U_i, (s, u_i)) \models \phi, \text{ for some } i, 1 \leq i \leq k\).

A formula is *valid* if it is true in all models at all worlds. A formula is *satisfiable* if it is true in some model at some world.

With the semantics of the update modalities given using product update, the logics for the languages considered in Definition 4 are called \(EL\), \(DEL_\emptyset\) and \(DEL_\emptyset\), respectively.

For a logic \(\mathcal{L}\), the problem of checking satisfiability of its formulas is called \(\text{SAT}(\mathcal{L})\). We will also consider the *model checking* problem \(MC(\mathcal{L})\), which checks, given a pointed Kripke model \((M, s)\) and formula \(\phi\) of \(\mathcal{L}\), whether \(\phi\) is true at world \(s\) in \(M\).

Figure 1 shows the upper bounds known about the computational complexity of these problems. The syntax and semantics of the logic at the source of an edge are available in the logic at the target of the arrow. The class \(\text{PSPACE}\) contains problems which are solvable by algorithms using space bounded by a polynomial in the size of the input \(n\). The class \(\text{NEXPTIME}\) contains problems which are solvable by nondeterministic algorithms using time bounded by an exponential \(2^{\text{poly}(n)}\), where \(\text{poly}(n)\) is a polynomial in the size of the input \(n\). It is well known that \(\text{PSPACE}\) is included in \(\text{NEXPTIME}\). The inclusion is not known to be proper.
We sometimes talk of ‘agency’ by which we mean that an agent $i$ exists at a world $s$ in model $M$ (iff $(M, s) \models P_i \top$). An agent $i$ exists for another agent $j$ at a world $s$ if $i$’s agency holds at all the worlds $t$ reachable by $j$ from $s$. Formally, $B_j P_i \top$.

We work only with transitive relations $R_i$, hence $B_i \phi \Rightarrow B_j B_i \phi$ is a valid formula. It says that positive belief is introspective. In our models, $\neg B_i \phi \Rightarrow B_i \neg B_i \phi$ is not a valid formula. It says that negative belief is introspective.

Such correspondences of valid formulas with properties of Kripke frames are described in the textbook of Chellas (1980). Van Benthem (1984) is a far-reaching survey.

Soundness and completeness of DEL with respect to an axiom system are proved in Baltag and Moss (2004); Van Benthe-mer et al. (2006); Van Ditmarsch and Kooi (2008). Decidability of the logic follows. The complexity of checking validity is nondeterministic exponential time, established by Aucher and Schwarzentruber (2013).

The product update framework does not allow one to define actions that can introduce new agents or remove agents from the Kripke models. How would one model the following?

Consider a simple dormitory setting in which the arrival of a new warden is announced to the residents. The agents in the dormitory model are the existing set of residents and staff.

To take cognizance of the warden, a new agent after this announcement, demands a new kind of update, which is introduced in Section 3.

2.1 Literature

Approaches to model epistemic actions as changes on Kripke models broadly fall under two categories: state elimination and arrow elimination. For instance, Plaza (1989) models public announcements by eliminating states from the input model where the announced formula is false. On the other hand, Gerbrandy and Groeneveld (1997) model public announcements by eliminating arrows from the model such that only those worlds are reachable by the agents where the announced formula is true.

The action framework of Baltag and Moss (2004); Baltag et al. (1998) is used to model not only epistemic actions but also ontic effects. We present our agent updates in the style of action frames. The book by Van Ditmarsch et al. (2008) gives various dynamic epistemic logics. Our language $DEL_{\otimes}$ is the language $\mathcal{L}_{K_{\otimes}}$, as can be derived from (Van Ditmarsch et al., 2008, Theorem 8.54). The complexity of a related dynamic logic avoiding nondeterministic choice was established in Halpern and Reif (1983). The complexity of model checking and satisfiability of $DEL_{\otimes}$ with nondeterministic choice (the logic $\mathcal{L}_{K_{\otimes}}$ of Van Ditmarsch et al., 2008) was established by Aucher and Schwarzentruber (2013).

Because our updates can add and delete agents, we can model interesting examples involving deception about agents. The formal account of Sakama (2015) gives a classification of deceptions, of which we consider only a simple form in this paper, which is called intentional deception by lying. In our case the lying is more general, about existence of agents, and our semantics has additional features. Modelling lies has been studied in dynamic epistemic logic (Sakama et al., 2010; Van Ditmarsch 2014; Van Ditmarsch et al., 2012).
Sarkadi et al. (2019) model machine deception using ideas from communication theory and implement it in BDIA (Belief, Desires, Intentions and Action) agent architecture. They require the precondition that the deceived person is aware of the ascribed formula, which we do not require; and that the deceiver has a ‘theory of mind’ about the deceived person’s beliefs, desires, intentions, which we do not model.

Awareness of propositions has been studied in many papers, with agents becoming aware or unaware of them (Fagin and Halpern, 1988; Van Bentheim and Velázquez-Quesada, 2010). The papers (Van Ditmarsch and French, 2008) specifically mention awareness of agents, using bisimulation quantifiers to model introduction of new propositions (which includes agent propositions). Our agent-update approach is completely different, it works at the level of semantic operations on action models. A simpler form of quantification is used by Padmanabha et al. (2018) to obtain polynomial space algorithms.

Independently of Wang et al. (2022) which has the same idea, we model existence of agents at a world using presence of that agent’s accessibility at the world. Our results in Section 5 show that complexity is unaffected.

The recent paper Van Ditmarsch (2021) also allows removal of agents. This paper explores other interesting directions such as concurrency.

Amarel (1971) suggested using the folk problem of missionaries and cannibals to study planning problems in artificial intelligence. Smullyan’s books, starting from Smullyan (1978), are masterpieces of logic puzzles of various kinds. The book of Van Ditmarsch and Kooi (2015) is an inspiring account of modelling epistemic puzzles as stories. The book of Woods (2018) explores the paradox that Sherlock Holmes lived in 221B Baker Street in the 19th century, and that he didn’t since he didn’t exist then.

We make extensive use of The Gruffalo (Donaldson, 1999) from children’s fiction to illustrate deception. We invite the reader to explore the book and its sequel (Donaldson, 2004) for interesting ideas. Bormann (2013) points out that these stories (and others) have not been subject to much analysis in possible worlds theory.

Remark 1 (Historical). The story by Donaldson (1999) is based on an Eastern folktale. It is close to one in Burton (1888), a retelling of a 10th century Persian story in the collection One thousand and one nights (also called Arabian nights). Here a cat on the ground, after eating several hens, tells a cock on a tree, perched beyond its reach, that the King of Beasts has declared that all of them should love each other and proposes that they go for a stroll. The cock says it can see that the hounds of the King of Beasts are coming to make others aware of the declaration. The cat hurriedly leaves, on being asked why, tells the cock that it is not sure that the hounds know of the declaration. The essence of the story is the same, this time with the cock as deceiver and cat as deceived. Modelling it would require a planning domain (McDermott, 2000), showing the asymmetric location where the cock’s visibility is higher than that of the cat. Donaldson’s story is simpler and easier to model in pure doxastic logic.

It appears this Arabian nights story is derived from a simpler one in the Buddhist Jataka tales, which go at least as far back as the 3rd century (Cowell and Neil, 1907). In the Kukkuta jataka, engraved on a Buddhist stupa at Bharhut in Madhya Pradesh, India, a cat declares its love for a cock and proposes to it. The cock declares (at a higher plane) that there cannot be friendship or love between predator and prey. There is no deception here. Modelling the story would require quantified logic.

3 Agent-update models

We formally define agent-update frames on a countably infinite set of agents $A$ and a finite $A \subseteq A$. Recall the language $EL$ from Section 2.

Definition 6 (Agent-update frame on $A \subseteq A$). An agent-update frame is a finite structure $U = \langle E, \{O_i \mid i \in A\}, \{O^+_i \mid i \in A\}, \{O^-_i \mid i \in A\}\rangle$ with relations for agents $i$ as indicated, the former two being transitive, defined on a finite set of events $E$, together with the precondition and postcondition functions $pre : E \rightarrow EL$ and $post : E \rightarrow \{Prop \rightarrow \{\top, \bot, no\}\}$ as in Definition 2. $uO^+_i v$ means that event $u$ adds agent $i$, we collect such added agents $i$ in the set $Add(u)$. $uO^-_i v$ means that event $u$ deletes agent $i$, and $Del(u)$ is the collection of such deleted agents.

A pointed agent-update frame is written as $(U, u)$ where $u \in E$ is a designated event.

The size of an agent-update frame is as before the sum of the sizes of its components, with $3|AllE|^2$ for the three kinds of relations.

In pictures, in addition to the traditional (solid) arrows (here denoted as $O_i$) in an action frame on $A$, we have two other types of arrows: sum arrows, dashed, for $O^+_i$, which can range over new agents outside $A$, and difference arrows, dotted, for $O^-_i$ on $A$ in the agent-update frames.
Agent-update Models

We use letters $a, b, i, j, k$ to denote agents, $s, t$ to denote worlds in Kripke frames, and $u, v, w, x$ to denote events in the agent-update frames throughout the paper.

3.1 Examples

We start with examples from a dormitory and from a story by Donaldson (1999).

**Example 1.** Consider the agent-update frame illustrated in Figure 2 in which event $u$ adds a warden agent $i$, this is shown using a dashed arrow. The other (solid) arrows define observability of existing agents as usual. Let $a$ be some agent, say a security guard, and consider agents in set $R$ as residents of the dorm. All the agents in $R \cup \{a\}$ observe the event $u$ that adds $i$ in the frame.

![Figure 2: i-addition for $R \cup \{a\}$](image)

Suppose the initial situation in the dorm is depicted by the Kripke model in Figure 3, where at state $s$, the guard $a$ believes the proposition $p$, but the residents $R$ do not. What beliefs should $i$ be ascribed with after event $u$ is executed in state $s$?

Our approach is to make the warden inherit the beliefs commonly held by the residents and the guard, except where their beliefs differ from the warden’s. One reason is that group beliefs about the dorm, such as the existence of its residents, are automatically communicated to the warden. Think of all these beliefs being listed on the group’s webpage.

The issue of beliefs implicitly held by a group is more tricky, we do not consider it.

The effect of the agent-update in Example 1 is depicted in Figure 3. At $(s, u)$, agent $i$ inherited arrows of $R$ and $a$ from world $s$ to $t_1$ and $t_2$, respectively. Therefore, the warden does not inherit any belief about $p$. 

6
Agent-update Models

**Example 2.** Suppose a new resident, say *Tom*, joins a hall *h* in the dorm. The residents of the hall $R_h$ witness his arrival, as do the warden $i$ and the security guard $a$, as indicated by the dashed arrow. Other residents $R \setminus R_h$ are unaware of the agency of *Tom* at this moment. Such a private agent update is illustrated in Figure 4. Agents in $R_h$, $i$ and $a$ see *Tom* being added at $u$. Agents in $R \setminus R_h$ are oblivious, they believe that beliefs of $R_h$, $i$ and $a$ are unchanged.

In Example 2 only warden $i$, guard $a$ and residents $R_h$ become aware of *Tom* joining the dorm (event $u$), therefore *Tom* inherits the possible worlds from $i$, $a$ and $R_h$ from the input model into the updated model as shown in Figure 5. Beliefs of $R \setminus R_h$ remain static (do not change after the update), while beliefs of $R_h \cup \{i, a\}$ remain static except for the beliefs about the agent being added: *Tom*.

**Example 3.** Similarly a private deletion of an agent happens when one of the residents, say *John*, leaves the dorm, and the departure is observed only by a subset of residents $R_h$ and $\{i, a\}$, the rest being oblivious of the departure. In Figure 6 the dotted self-loop for *John* at $u$ denotes that resident *John* is to be deleted. The agents in $R_h \cup \{i, a\}$ observe this event at $u$. Beliefs of rest of the residents in $R \setminus R_h$ do not change. In particular, they continue to believe that $R \cup \{i, a\}$ believe in the agency of *John* at $v$. Model update is shown in Figure 7.
3.2 A children’s story

Now we look at the story from *The Gruffalo* [Donaldson (1999)]. This children’s story has a clever mouse who invents a fictitious creature called gruffalo, in order to scare some predators away. The fox, the owl and the snake in turn are duly convinced into fleeing as this creature likes to eat the others. But then the story takes a curious turn when the mouse runs into the gruffalo in flesh and blood. The clever mouse finds a way to its safety in a different manner this time. It boasts to the gruffalo that the fox, the owl and the snake are scared of it as the mouse is the most dangerous creature in the jungle. It takes the gruffalo to the fox, which flees on seeing the gruffalo in flesh and blood. The gruffalo understands this to be because the fox is afraid of the dreaded mouse. After this episode is repeated with the owl and the snake too, the gruffalo itself flees.

**Example 4.** Two events in the agent-update frame shown in Figure 8 depict the mouse’s action of misleading the fox into believing a fictitious gruffalo: \( u \) as perceived by the mouse \( m \) and \( v \) as perceived by the fox \( f \). The presence of a dashed self-loop on event \( v \) means that agent \( g \) (the gruffalo) is added by \( v \).

In Example 4 the fox \( f \) is deceived into believing in the agency of gruffalo \( g \) after mouse \( m \)’s announcement, while \( m \) itself does not believe in \( g \)’s agency, as shown in Figure 9. In the fox’s imagination, gruffalo believes \( egf \) (gruffalo eats fox), a belief induced by \( m \). So a new belief is ascribed to gruffalo which is not inherited.

Similarly, we can have a deceptive deletion of an agent, which we leave to the reader. Section 4 looks at the story in more detail, showing how beliefs are ascribed to new agents in more complex scenarios.

In our doxastic models, agency at a state \( s \) is about the belief that an agent exists. Thus whether an existing agent *Tom* arrives at a scene, or a fictitious agent *g* ‘arrives’ in the imagination of an agent (at a state in the model representing beliefs of the agent), both are treated in the same way. The distinction is the following: *Tom*’s arrival is witnessed by the agents present, and their beliefs regarding his presence change; the residents in the other hall do not witness his arrival, and their beliefs regarding his presence do not change. Belief in the gruffalo \( g \) enters the mind of the fox \( f \), and there is the mouse \( m \) who ‘witnesses’ this change of belief, yet its belief about the non-existence of \( g \) does not change.
We identify a set of agents whose beliefs remain unchanged at an event in an agent-update frame.

**Definition 7** (Observer). The set of observers \( \text{Obs}(u) \) at an event \( u \) in an agent-update frame is those \( j \) with agency at \( u \) such that \( uO_j v \iff v = u \).

Deceivers are certain observers who remain unmoved by changes in beliefs of observed agents. We do not formalise this here since we do not use it in the paper.

### 3.3 Observing and simultaneous deception

#### 3.3.1 Observing deception

Consider a wise owl \( o \) observing deceiver \( m \) (oblivious of the owl’s presence) deceiving \( f \) into believing in the addition of agent \( g \), as illustrated in the agent update frame illustrated in Figure 10. Event \( e \) encodes the observation of deception by \( o \). At \( e \), \( m \) as well as \( f \) consider \( o \) to be away and therefore \( o \) is considered observing a *skip* by both. \( m \) considers \( u \) where it tries to deceive \( f \).
3.3.2 Deception down the line

In the agent update frame illustrated in Figure 11 while \( f \) is being deceived by \( m \) into believing in the addition of agent \( g_1 \), an owl \( o \) is deceived by \( f \) and \( m \) into believing in the addition of agent \( g_2 \).

We can also make \( m \) oblivious of the deceit carried out by \( f \).

By now the idea should be clear: arbitrary levels of observation and deception can be modelled. More work is needed to better formalise this observation.

Next we put together all that we have seen so far into a single operation generalising product update. The specific updates which we saw above contributed significantly to the formulation of the next definition.

3.4 Sum-product update

We define sum-product update to describe belief update for existing agents, to ascribe beliefs to newly added agents, and to drop beliefs of deleted agents. During model transformation, for an existing agent \( a \), possible worlds for an agent in the updated model include the (unforgotten) worlds it considered possible earlier. In world \( (s, u) \) (after execution of event \( u \) in world \( s \)) of the product model, another world \( (t, v) \) is possible if and only if \( t \) is possible from \( s \), and \( v \) is possible from \( u \). For the agent \( i \) being added by an agent-adding event \( u \) (\( i \in \text{Add}(u) \)), the worlds that \( i \) considers possible at \( (s, u) \) are observer-dependent.

The beliefs of the existing agents are determined by product, the beliefs of the newly added/deleted agents are determined by sum/difference. We describe the transformation of a model on \( A \) when an agent-update frame on \( A \) is applied to it, and we call it sum-product update. An agent's deletion takes priority over its addition.

Definition 8 (Sum-product update). Given a pointed Kripke model \((M, s)\) on agents \( A \) and a pointed agent-update frame \((U, e)\) with \( U = (E, O, O^+, O^-, \text{pre}) \) on agents \( A \), the updated pointed Kripke model \((M^* U, (s, e))\), is defined as: \((S', \{R'_a \mid a \in A'\}, V')\) on the updated set of agents \( A' \) (those \( a \) such that \( R'_a \) is nonempty), where:

- \( S' = \{(s, u) \mid M, s \models \text{pre}(u)\} \cap (S \times E) \)
- \( V'(p) = \{(s, u) \in S' \mid (s \in V(p) \text{ and } \text{post}(u)(p) = no) \text{, or } \text{post}(u)(p) = \top\} \)
- \( R'_a \) is the transitive closure of \((Q^\text{unf}_a \cup Q^\text{asc}_a \cup Q^\text{inh}_a)\), where:
  - unforgotten: \((s, u)Q^\text{unf}_a(t, v) \iff sR_at \text{ and } uO_au \text{ and not } uO_a^\top v\)
  - ascribed: \((s, u)Q^\text{asc}_a(s, v) \iff uO_a^\top v, \text{ for } a \in \text{Add}(u) \setminus \text{Del}(u)\)
  - inherited: \((s, u)Q^\text{inh}_a(t, u) \iff sR_{\text{Obs}(u)}t, \text{ for } a \in \text{Add}(u) \setminus \text{Del}(u)\)

The accessibility relation for an agent, say \( a \) in the transformed model, is defined such that after execution of an event \( u \) in the world \( s \), it considers a world \((t, v)\) possible at world \((s, u)\) if and only if:

1. If \( a \) is neither being added nor deleted, this is the product of arrows of Definition 3. We call the beliefs accessible in this way unforgotten.
2. If \(a\) is being deleted (which takes priority over addition), then the beliefs accessible by undeleted \(a\)-arrows are also called *unforgotten*.

3. If \(u\) is an \(a\)-adding event and not simultaneously an \(a\)-deleting event, there are three cases.
   
   (a) If \(a\)-arrows were already present, they are handled using product. These are again *unforgotten* beliefs.
   
   (b) An explicit \(a\)-adding arrow is from event \(u\) to event \(v\), both executable at \(s\). Any new beliefs here are *ascribed* by the updating agent.
   
   (c) Worlds \(t\) considered possible from world \(s\) by an observer at \(u\), provided \(\text{pre}(u)\) holds in \(s\) are *inherited* from the observer.

The number of states in the updated Kripke model is the product \(|\mathcal{S}| |\mathcal{E}|\) of the number of states in the starting Kripke model and the number of events in the update frame. Its size will be asymptotically dominated by the size of its relations \(3|A'||\mathcal{S}|^2|\mathcal{E}|^2\).

4. **The Gruffalo formalised**

We have seen many one-step agent-update examples. In this section, we model the story in Section 3.2 to formally show agency-creation of *gruffalo*, first for the *fox* and *owl*, then for the *mouse*, following the course of events presented in the story. The story is a little more complex than the scope of our models. It uses magical realism which is difficult to model within a propositional modal framework like ours, which does not distinguish between appearance and reality. Fortunately we are able to use a trick to show how the gruffalo deceives itself.

Action frames are used to model updates and not reasoning. Following the book (which is aimed at children), there is no suggestion that the mouse understands predator-prey theory, nor that this is a precondition for deception. A theory of what kind of *causes* ([Lin, 1995] studies these formally) could lead to actions such as agent addition and deletion, is beyond the scope of this paper. Action frames do not even specify which agent(s) perform the action.

We do not use a planning domain ([McDermott, 2000]), there is no location or time in an action frame, so we do not attempt to adequately model actions such as ‘fox runs away’, leaving them to the imagination of the reader.

4.1 The fox is deceived

The initial situation in the story is modelled with \(M_0\) with a designated world \(s\) as shown in Figure 12. There are several agents in the story. We begin with three: mouse \(m\), fox \(f\) and owl \(o\), and all three believe in each others’ agency.

![Figure 12: Mouse deceives fox that there is a fox-eating gruffalo](image)

**Example 5.** The first move of \(m\) is a deceptive agent-addition action which introduces the agency of a gruffalo \(g\) which, \(m\) announces, likes to eat fox. In Figure 12, this move is modelled as a combination of an addition and a deletion event,
Agent-update Models

at $u_1$ the mouse deceives the fox into believing in the agency of the gruffalo; at $v_1$, the fox believes the gruffalo believes in eating foxes, which we represent as a deletion of fox at $x_1$. The owl at $w_1$ is oblivious of this interaction.

Figure 13: Mouse deceives owl that there is an owl-eating gruffalo

4.2 The owl is deceived

Example 6. Next when the mouse runs into the owl, it makes a similar deceptive move again, at $u_2$ the owl is deceived into believing in the agency of the gruffalo with fox being oblivious of the interaction; at $v_2$ the owl believes that the gruffalo believes in eating owls. The model update is shown in Figure 13.

Note that the owl starts believing in the agency of gruffalo but it does not know of the beliefs of the fox. Likewise the fox believes in the agency of gruffalo and it believes that the owl does not.

In the book by Donaldson (1999), there is next the deception of a snake on similar lines, which we do not get into, as we have not introduced the snake as an agent at all.

4.3 The gruffalo appears

Further in *The Gruffalo*, Donaldson inventively mixes in magical realism.

Example 7. The mouse runs into a ‘real’ gruffalo, illustrated in Figure 14. We model this as a private $g$-addition-update about $g$’s agency for mouse while fox and owl remain oblivious. To save space, worlds $(s, u_1, u_2), (s, v_1, w_2), (s, x_1, w_2), (s, w_1, v_2), (s, w_1, w_2)$ and $(s, w_1, x_2)$ in model $M_2$ (Figure 13) are renamed as $s_1, s_2, s_3, s_4, s_5$ and $s_6$ in Figure 14.

4.4 Gruffalo meets fox

Example 8. Further in the story, the mouse introduces the gruffalo to the fox. Mouse is an observer of event $u_4$ at which $g$ observes $f$-addition at $v_4$. Owl is oblivious at $w_4$. Fox’s belief that it will be eaten away is unforgotten at $x_4$ in the agent-update frame at the bottom of Figure 15. Hence the fox runs away as soon as it sees them. Gruffalo’s belief is that the fox, appearing at $v_4$, ran away because it encountered them, these are the $m, g$-addition arrows at $v_4$. Gruffalo chooses to believe $P_f P_m - P_f ^\perp$ at $(s_1, v_3, v_4)$. This is represented as proposition $p$. 
4.5 Gruffalo meets owl

Example 9. Next in the story, the mouse introduces the gruffalo to the owl as shown in Figure 16. To save space, worlds \((s_1, u_3, u_4), (s_2, w_3, w_4), (s_3, w_3, w_4), (s_4, w_3, w_4), (s_5, w_3, w_4), (s_6, w_3, w_4), (s_1, v_3, v_4),\) and \((s_1, v_3, x_4)\) are renamed as \(t_1, t_2, t_3, t_4, t_5, t_6, t_7\) and \(t_8\), respectively. The owl runs away as soon as it sees them, since its belief that gruffalo eats owl is unforgotten. Gruffalo’s belief \(P_o P_m \neg P_o T\) can be seen in the updated model at \((t_7, v_5)\). This is
representated as postcondition \( q \). The earlier formula \( p \) representing its belief \( P_f P_m \neg P_f \top \) also holds at \((t7, v5)\). Thus gruffalo believes \( p \land q \), that both fox and owl are afraid of mouse, and it also runs away.

5 Agent-update logic

The logic DEL is now defined with agent-update frames being used as update actions. The truth of formulas with update modalities is interpreted using sum-product update:

\[
(M,s) \models \langle U,u \rangle \phi \iff (M,s) \models \text{pre}(u) \land (M \ast U,(s,u)) \models \phi
\]

The sublanguages of Definition 4 define the logics EL, DEL∗ and DEL†, respectively, with this semantics.

We revisit the mouse-gruffalo example one last time.

**Example 10.** Initially \( M_0 \) is defined such that \((M_0,s) \models P_m \top \land P_f \top \land P_o \top \land \neg P_g \top \). In Figure [12] when the mouse deceives the fox about existence of a gruffalo, keeping the owl oblivious, \( M_0 \ast U_1,(s,u1) \models \neg P_g \top \land P_f \top \land B_m B_f P_g \top \land \neg P_g P_g \top \). When the mouse runs into the owl, it deceives the owl too but keeping the fox oblivious this time, as shown in Figure [13], and therefore in the updated model none of them know that the other is aware of gruffalo’s agency: \( M_0 \ast U_1 \ast U_2,(s,u1,u2) \models \neg P_g \top \land P_f \top \land P_o \top \land \neg P_f P_g \top \land \neg P_o P_g \top \).

Eventually the gruffalo appears in front of the mouse leading to the update of the mouse’s beliefs in the existence of gruffalo: \( M_0 \ast U_1 \ast U_2 \ast U_3,(s,u1,u2,u3) \models P_m P_g \top \). The cunning mouse contrives a situation where the gruffalo believes that fox is eaten away (as illustrated in Figure [15]), then according to the the information state of gruffalo: \( M_0 \ast U_1 \ast U_2 \ast U_3 \ast U_4,(s,u1,u2,u3,u4) \models P_g (p \land \neg P_f \top) \). But it can be seen from the model (Figure [15]) that: \( M_0 \ast U_1 \ast U_2 \ast U_3 \ast U_4,(s,u1,u2,u3,u4) \models P_f \top \land B_m P_f \top \). Similarly, after the Gruffalo is taken to the owl, its information state is updated such that: \( M_0 \ast U_1 \ast U_2 \ast U_3 \ast U_4 \ast U_5,(s,u1,u2,u3,u4,u5) \models P_g (p \land q \land \neg P_f \top \land \neg P_o \top) \).

**Theorem 1** (Model checking). Given a transitive pointed Kripke model, whether a DEL∗ formula holds (with agent-update semantics) can be checked in polynomial space.

It is well known (see the book by Blackburn et al. [2001]) there is a lower bound of polynomial space for model checking transitive Kripke models and the modal logic \( K4 \).
5.1 Proof system

The proof system for the logic $DEL^-_a$ with agent-update semantics gives axioms and inference rules to prove valid formulas. There are 9 axioms and 3 standard inference rules below. In the expression $(U, u) P_a \phi$, $(U, u)$ stands for an agent-update, and $a$ is any agent, which may or may not be involved in the update.

1. all instantiations of propositional tautologies
2. $B_a (\phi \implies \psi) \implies (B_a \phi \implies B_a \psi)$
3. $B_a \phi \implies B_a B_a \phi$
4. $[U, u] (\phi \implies \psi) \implies ([U, u] \phi \implies [U, u] \psi)$
5. $(U, u) p \iff (\text{pre}(u) \land ((p \land (\text{post}(u)(p) = no)) \lor (\text{post}(u)(p) = \top)))$
6. $(U, u) \neg \phi \iff (\text{pre}(u) \land \neg (U, u) \phi)$
7. $(U, u) (\phi \land \psi) \iff ((U, u) \phi \land (U, u) \psi)$
8. $(U, u) P_a \phi \iff (\text{pre}(u) \land \bigvee_{v : aO_v \neg aO_v} P_a (U, v) \phi)$, for $a \not\in (\text{Add}(u) \setminus \text{Del}(u))$
9. $(U, u) P_a \phi \iff (\text{pre}(u) \land \bigvee_{v : aO_v} P_a (U, v) \phi \lor \bigvee_{v : uO_v} P_a (U, v) \phi \lor \bigvee_{k : \neg \text{Obs}(a)} P_b (U, u) \phi)$, for $a \in (\text{Add}(u) \setminus \text{Del}(u))$
10. From $\phi$ and $\phi \implies \psi$, infer $\psi$
11. From $\phi$, infer $B_a \phi$
12. From $\phi$, infer $[U, u] \phi$

It follows from [Van Ditmarsch et al., 2008] Theorem 8.54) that the logic $DEL_\ast$ can be axiomatized with the additional axiom:

13. $((U_1, u_1) \sqcup \cdots \sqcup (U_k, u_k)) \phi \iff ((U_1, u_1) \phi \lor \cdots \lor (U_k, u_k) \phi)$

Theorem 2 (Soundness and Completeness). The proof system given in Section 5.1 is sound and complete for transitive Kripke models.

5.2 Decidability

The logic $DEL^-_a$ is defined in the book of [Van Ditmarsch et al., 2008], its complexity is not pinned down there. A lower bound of polynomial space for the sublanguage EL (the logic K4 of Chellas (1980)) is found in the book of [Blackburn et al., 2001]. The minus in the superscript specifies that unions of update frames are not available in the syntax. This enables a better upper bound than found in the literature.

Theorem 3 (Satisfiability). There is a polynomial space algorithm to check satisfiability of a $DEL^-_a$ formula. There is a nondeterministic exponential algorithm to check satisfiability of a $DEL_\ast$ formula.

Proof. For the complexity of the algorithm, note that an update formula $(U, u) \phi$ contains within it a description of the agent-update frame. The number of agents, the number of events in it, the observability relations are all included in the size of the frame, defined in Section 2. In the length of an update formula we include the size of the update frame.

Consider a representative world where the reduction equivalence is used as a rewrite rule going from left to right. The number of disjuncts on the right hand side of an axiom is linear in the number of events in the agent update model, as Definition 7 defines observers in terms of events. Thus the right hand side is bounded by a polynomial in the representation of $(U, u)$ and at most twice the modal depth of the formula, allowing for a rewrite step to go from an update modality over a belief modality over a belief modality over an update modality, before doing a further rewrite step.

Thus a satisfiability algorithm can run on a tree with depth linear in the modal depth of the formula; for example, going from a parent node with the left hand side of a reduction equivalence to a child node with one of the disjuncts on the right hand side. The tree has number of nodes exponential in the length of the formula $\phi$.

Checking whether an agent relation holds for a node can be done in time polynomial in the formulas which appear along the path from the node to the root of the tree. Agent relations are recomputed every time they are required, so the algorithm can take exponential time, but working on one path uses at most polynomial space.

Such a (satisfying) path can be guessed and verified by a nondeterministic algorithm which takes space polynomial in the length of the formula. This gives an algorithm for satisfiability of $DEL^-_a$. 

15
Agent-update Models

For $DEL$, this does not suffice since the $\langle (U_1, u_1) \cup \cdots \cup (U_k, u_k) \rangle$ modality can refer to different branches of the satisfying tree (Aucher and Schwarzentruber, 2013). The whole satisfying tree has to be guessed and verified, which takes a nondeterministic algorithm taking time exponential in the length of the formula.

We illustrate the status of the model checking and satisfiability problems for the different logics in the picture. The upper bound for $SAT(DEL_{\rightarrow})$ has been improved from the earlier picture. All bounds shown are tight, the upper bounds match the lower bounds, apart from the possibility that $PSPACE=NEXPTIME$.

![Figure 17: Upper bounds for $MC(\mathcal{L})$ and $SAT(\mathcal{L})$, revised](image)

6 Discussion

In this paper we explored the idea of agent-addition and deletion in a group. We see our work as a starting point. We can broadly classify the different kinds of agent-updates that are possible under nested levels of modalities, as hinted at in Section 3.3. More refined ways of ascribing beliefs and nesting updates remain to be found.

Wang et al. (2022) has interesting axioms for properties, such as weaker forms of reflexivity, which hold in models of epistemic term-modal logic. Since our logic is doxastic, this particular property is not valid for our models. Finding validities which express meaningful properties would be of interest.

Action frames in dynamic epistemic logic have traditionally been associated with ‘hard’ update operations. We have extended them with agent-update operations. ‘Soft’ operations such as upgrades have also been studied for belief revision (see the book of Van Benthem (2010)). Interesting agent-update operations in such settings remain open.

References

Amarel, S. (1971). On representation of problems of reasoning about action. In Michie, D., editor, Machine Intelligence 3, pages 131–171. Edinburgh Univ press.

Aucher, G. and Schwarzentruber, F. (2013). On the complexity of dynamic epistemic logic. In Schipper, B. C., editor, Proc. 13th TARK, Chennai, pages 19–28.

Baltag, A. and Moss, L. S. (2004). Logics for epistemic programs. Synthese, 139(2):165–224.

Baltag, A., Moss, L. S., and Solecki, S. (1998). The logic of common knowledge, public announcements, and private suspicions. In Gilboa, I., editor, Proc. 7th TARK, Evanston, pages 43–56. Morgan Kaufmann.

Baral, C., Gelfond, G., Pontelli, E., and Son, T. C. (2022). An action language for multi-agent domains. Artif. Intell., 302(103601).

Blackburn, P., De Rijke, M., and Venema, Y. (2001). Modal logic. Cambridge University Press.

Bolander, T. (2017). A gentle introduction to epistemic planning: The DEL approach. In Proc. 9th M4M, Kanpur, volume 243, pages 1–22. EPTCS.

Bolander, T. (2018). Seeing is believing: Formalising false-belief tasks in dynamic epistemic logic. In Van Ditmarsch, H. and Sandu, G., editors, Jaakko Hintikka on Knowledge and Game-Theoretical Semantics, pages 207–236. Springer.
Agent-update Models

Borman, D. C. (2013). Moving possible world theory from logic to value. Poetics today, 34(1–2).

Burton, R. F. (1888). The book of the thousand nights and one night. Kama Shashtra Society.

Chellas, B. F. (1980). Modal logic: an introduction. Cambridge University Press.

Cowell, E. and Neil, R. (1907). The Jatakas or stories of the Buddha’s former births. Cambridge University Press.

Donaldson, J. (1999). The Gruffalo. Pan Macmillan.

Donaldson, J. (2004). The Gruffalo’s child. Pan Macmillan.

Fagin, R. and Halpern, J. Y. (1988). Belief, awareness, and limited reasoning. Artificial intelligence, 34(1):39–76.

Fagin, R., Halpern, J. Y., Moses, Y., and Vardi, M. (2004). Reasoning about knowledge. MIT press.

Gerbrandy, J. and Groeneveld, W. (1997). Reasoning about information change. J. Log. Lang. Inform., 6(2):147–169.

Halpern, J. Y. and Reif, J. H. (1983). The propositional dynamic logic of deterministic, well-structured programs.

Theoret. Comp. Sci., 27(1-2):127–165.

Hintikka, J. (1962). Knowledge and belief: An introduction to the logic of the two notions. Cornell Univ. press.

Kripke, S. (1963). Semantical analysis of modal logic i: Normal modal propositional calculi. Zeitschr. math. Logik Grundl. math., 9:67–96.

Lin, F. (1995). Embracing causality in specifying the indirect effects of actions. In Proc. 14th IJCAI, Montreal, pages 1985—1993. Morgan Kaufmann, San Mateo, CA.

Madden, J. (2001). Operation Mincemeat. Film Nation Entertainment.

McDermott, D. M. (2000). The 1998 ai planning systems competition. AI magazine, 21(2):35–35.

Meyer, J.-J. C. and Van Der Hoek, W. (2004). Epistemic logic for AI and computer science. Number 41 in Tracts in Theoret. Comp. Sci. Cambridge University Press.

Padmanabha, A., Ramanujam, R., and Wang, Y. (2018). Bundled fragments of first-order modal logic: (un)decidability.

In Ganguly, S. and Pandya, P. K., editors, Proc. 38th FSTTCS, volume 122 of LIPIcs, pages 43:1–20. Schloss Dagstuhl.

Plaza, J. (1989). Logics of public communications. In Emrich, M., Pfeifer, M., Hadzikadic, M., and Ras, Z., editors, Proc. 4th Meth. Intell. Syst., New York, pages 201–216. Academic press.

Sakama, C. (2015). A formal account of deception. In Deceptive and counter-deceptive machines, AAAI Fall Symp., Arlington, pages 34–41.

Sakama, C., Caminada, M., and Herzig, A. (2010). A logical account of lying. In Janhunen, T. and Niemela, I., editors, Proc. 12th IELIA, Helsinki, volume 6341 of LNCS, pages 286–299. Springer.

Sarkadi, S., Panisso, A., Bordini, R., McBurney, P., Parsons, S., and Chapman, M. (2019). Modelling deception using theory of mind in multi-agent systems. AI Commun., 32(4):287–302.

Scott, W. (1808). Marmion: a tale of Flodden Field. Cambridge University Press.

Singh, S. and Khemani, D. (2019). Planning to deceive in a multi-agent scenario. In Cox, M. T., editor, Proc. 7th Adv. Cog. Syst., pages 473–491. Tech. Rep. COLAB2-TR-4. Dayton, OH: WSU, Collaboration and Cognition Lab.

Smullyan, R. M. (1978). What is the name of this book? Prentice-Hall.

Van Benthem, J. (1984). Correspondence theory. In Guenthner, F. and Gabbay, D. M., editors, Handbook of philosophical logic, volume II, pages 167–247. D. Reidel.

Van Benthem, J. (2010). Modal logic for open minds. CSLI, Stanford University.

Van Benthem, J., Van Eijck, J., and Kooi, B. (2006). Logics of communication and change. Inform. Comput., 204(11):1620–1662.

Van Benthem, J. and Velázquez-Quesada, F. R. (2010). The dynamics of awareness. Synthese, 177(1):5–27.

Van Ditmarsch, H. (2014). Dynamics of lying. Synthese, 191(5):745–777.

Van Ditmarsch, H. (2021). Wanted dead or alive : Epistemic logic for impure simplicial complexes. In Silva, A., Wassermann, R., and de Queiroz, R. J., editors, Proc. 27th Log. Lang. Inf. Comput., volume 1308 of LNCS, pages 31–46. Springer.

Van Ditmarsch, H. and French, T. (2009). Awareness and forgetting of facts and agents. In 2nd WLIAMAS, Proc. 9th WI-IAT workshops, Milan, volume 3, pages 478–483. IEEE.

Van Ditmarsch, H. and French, T. (2014). Semantics for knowledge and change of awareness. J. Log. Lang. Inform., 23(2):169–195.

Van Ditmarsch, H. and Kooi, B. (2008). Semantic results for epistemic and ontic change. In Bonnano, G., Van Der Hoek, W., and Wooldridge, M., editors, Proc. 7th LOFT, pages 87–117. Amsterdam Univ press.

Van Ditmarsch, H. and Kooi, B. (2015). One hundred prisoners and a light bulb. Springer.

Van Ditmarsch, H., Van Der Hoek, W., and Kooi, B. (2008). Dynamic epistemic logic, volume 337 of Synthese library. Springer Science & Business Media.

Van Ditmarsch, H., Van Eijck, J., Sietsma, F., and Wang, Y. (2012). On the logic of lying. In Van Eijck, J. and Verbrugge, R., editors, Games, actions and social software, volume 7010 of LNCS, pages 41–72. Springer.

Wang, Y., Wei, Y., and Seligman, J. (2022). Quantifier-free epistemic term-modal logic with assignment operator. Ann. Pure Appl. Logic, 173(103071).

Woods, J. (2018). Truth in fiction: Rethinking its logic. Springer.