Two Ways to Scare a Gruffalo

Shikha Singh[1], Kamal Lodaya[2], and Deepak Khemani[1]

[1] Dept of Computer Science and Engineering, IIT Madras, Chennai 600036, India
cs16d008@smail.iitm.ac.in, khemani@cse.iitm.ac.in
[2] Bengaluru 560064

Abstract. This paper applies and extends the results from [22] on *agent-update frames* and their logic. Several interesting examples of actions for forgery and deception, agent-upgrade and agent-downgrade are considered. Going on from the earlier paper, a second interesting children's story is modelled using these ideas. A dynamic epistemic logic is defined with all these actions and provided with a complete axiomatization. Decision procedures for satisfiability and model checking follow. A planning-oriented approach is also discussed.

Keywords: Agent-update · Deception · Forgery · Completeness and Decidability · Epistemic planning

1 Introduction

In [22], the authors modelled Julia Donaldson's children's story *The Gruffalo* [14] in dynamic epistemic logic [28]. The technical enhancement required extending the update modality, specified by an action frame $U = (E, O_i, pre, post)$ [3], with a product update operation providing the updated Kripke model. We extended the semantics with agent-update frames $U = (E, O_i, O_i^+, O_i^-, pre, post)$, which allow adding and deleting agents at states of the Kripke model. The different viewpoints of agents provided in an action frame are used to model deception. An extended sum-product update operation underlies this development.

Our work showed us the flexibility and extendibility provided by action frames. Our extensions were extremely general, and dealing with how to ascribe beliefs to the updated agents was a challenge. The axioms we came up with seemed quite ad hoc.

In *The Gruffalo*, a mouse runs into a fox, owl and snake, all intent on eating it. It deceives them by claiming it is friends with a terrible gruffalo, and they all run away. In a magical twist, the gruffalo appears and wants to eat the mouse. The mouse claims everyone is scared of it and takes the gruffalo to fox, owl and snake, each of whom run away on seeing them. The gruffalo is convinced this must be because of the mouse, and it too runs away.

Remark 1 (Historical). The story by Donaldson [14] is close to one in the *Arabian Nights* [7]. 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 [13].

In this paper we model a sequel: Julia Donaldson's *The Gruffalo's Child* [15]. Here is the story in brief. There are several agents in the story. We begin with four: the gruffalo (child) g, mouse m, fox f and owl o, and all four believe in each others' agency. For simplicity, we do not consider a snake agent from the book. The gruffalo has been brought up to believe that the mouse is big and bad, which it is skeptical about. In the context of the story, "big and bad" means that it eats gruffalos. The gruffalo first meets the fox (instead of the snake in the story) and then the owl, both of whom reiterate this belief. The gruffalo remains skeptical. It then meets the mouse who is not big at all, and it sees that it can eat the mouse. The mouse utilizes the rising Moon to project a big shadow of itself. This reverses the gruffalo's belief and it runs away.

Remark 2 (Historical). A similar idea (a solar eclipse) was used by Tintin in *Prisoners of the Sun* [16].

To model the story we introduce agent-upgrade and agent-downgrade operations: the mouse is downgraded in the gruffalo's eyes, and then the mouse upgrades itself. We see them as variants of agent-addition and agent-deletion. There is a different way to present belief upgrade and downgrade without changing the set of agents [24, 4, 25] using ordered Kripke models. Although we do not study this, our work suggests that the two approaches may be inter-translatable.

Our paper [22] and the present paper began with a problem in AI planning. This is seen as model checking from a logic perspective [10, 11], and has been studied in DEL [18, 17]. We sketch how formulating it in logic suggests ways to tackle it.

Remark 3 (On stories). Amarel [1] suggested using the folk problem of missionaries and cannibals to study planning problems in artificial intelligence. Smullyan's books, starting from [23], are masterpieces of logic puzzles of various kinds. The book of [27] is an inspiring account of modelling epistemic puzzles as stories. Woods's books on fiction [31, 32] explore 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.

Here is an outline of the paper.

Section 2 gives some basic definitions of models, as well as the agent-update semantics of [22]. In Section 3 we define a few kinds of agent-updates and a logic with which we can use them. In particular we show how forgery can be modelled in addition to deception, as well as new operations of agent upgrade and downgrade. These updates are used to model the story from *The Gruffalo's Child* [15]. In Section 4 we prove the usual theoretical results: completeness of the proof system, algorithms for satisfiability and model checking. Section 5 has a discussion suggesting a more planning-oriented approach.

We want to thank Hans van Ditmarsch, Anantha Padmanabha, R. Ramanujam and Yanjing Wang for discussions on the earlier paper [22] which led to the writing of this paper.

2 Models and logic

We begin with Kripke structures.

Definition 1 (Kripke model). *$M = (S, \{R_i \mid i \in A\}, V)$, where model M consists of a set of possible worlds S and accessibility relations $R_i \subseteq S \times S$ for every agent $i \in A$, and a valuation function $V : Prop \to 2^S$ assigns states to a proposition. sR_it abbreviates $\langle s, t\rangle \in R_i$ and it means that at a world s, agent i believes possible that the world may be t. When an agent relation is reflexive, symmetric and transitive the worlds are said to be indistinguishable by the agent. 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_it and an undirected line between two worlds, say s and t, labelled with i, represents arrows for sR_it and tR_is.

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 only linear in the number of states. Thus the input is of size $O(|A||S|^2)$.

2.1 Updating Kripke models with actions

We present our agent-updates in the style of Baltag, Moss and Solecki's *action frames* [3], further developed in [26].

We formally define *Agent Update frames* on a countable set of potential agents $\mathcal{A}$ and a finite $A \subseteq \mathcal{A}$ of agents in a model [22]. The logic EL will be defined in Definition 4.

Definition 2 (Agent-update frame on $A \subseteq \mathcal{A}$). *An* agent-update frame *is a finite structure $U = (E, \{O_i \mid i \in A\}, \{O_i^+ \mid i \in \mathcal{A}\}, \{O_i^- \mid i \in A\}, pre)$ with a finite set of events E,* observability *relations for each agent: $O_i, O_i^+, O_i^- \subseteq E \times E$, the former two being transitive, together with function $pre : E \to EL$ which assigns a* precondition *for each event. uO_iv means that agent i perceives event u as event v. 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.*

A pointed 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 $pre(u)$ holds. See Definition 3 below.

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 *del* arrows, dotted, for O_i^- on A in the agent-update frames. Where required, the precondition of an event is shown alongside. Otherwise the precondition at an event u can be taken as $pre(u) = \top$.

We use letters a, b, g, h, 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. We will use R_X for a subset of agents X to abbreviate the transitive closure of $\bigcup\{R_j \mid j \in X\}$.

A *skip* event, represented as an event with $\top$ precondition and self-loops for all agents A, denotes no change. It will be frequently seen in agent frames.

The updated model after an action is formalized as a product of a Kripke model with an action frame [3].

We defined sum-product update [22] to describe belief update for the existing agents and to ascribe beliefs to the newly added agents, and drop beliefs of the deleted agents. During model transformation, for an existing agent a, the possible worlds for an agent in the updated model are inherited from the possible worlds it considered 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 due to an agent adding event u ($i \in Add(u)$), 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 $\mathcal{A}$ is applied to it, and we call it *sum-product update*. This is product update for agents in A, along with sum and difference for agents in $Add()$ and $Del()$ respectively. An agent's deletion takes priority over its addition.

Definition 3 (Sum-product update). *Given a pointed Kripke model (M, s) on agents A and a pointed agent-update frame (U, e) with $U = (E, O, O^+, O^-, pre)$ on agents $\mathcal{A}$, 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:*

1. $S' = \{(s, u) \mid M, s \models pre(u)\} \cap (S \times E)$
2. $V'(p) = \{(s, u) \in S' \mid s \in V(p)\}$
3. R'_a *is the transitive closure of* $(Q_a^{unf} \cup Q_a^{asc} \cup Q_a^{inh})$, *where:*
   unforgotten: $(s, u)Q_a^{unf}(t, v) \iff sR_at$ *and* uO_av *and not* uO_a^-v
   ascribed: $(s, u)Q_a^{asc}(s, v) \iff uO_a^+v$, *for* $a \in (Add(u) \setminus Del(u))$
   inherited: $(s, u)Q_a^{inh}(t, u) \iff sR_{Obs(u)}t$, *for* $a \in (Add(u) \setminus Del(u))$

2.2 Logic

We define our agent-update logic using the BNF below. Let $p \in Prop$ be a proposition, Y, X, H be disjoint subsets of $\mathcal{A}$ and i an element. We add specific agent-changing operators U given in the BNF below to obtain the language

AUL. The sublanguage without these operators is called EL. The book of Van Ditmarsch, Van der Hoek and Kooi [28] presents various dynamic epistemic logics.

Definition 4 (Formulas of updates and language AUL).

$$U ::= skip \mid p\ for\ X \mid p\ dcv\ X \mid +Y\ for\ X \mid -Y\ for\ X \mid \Uparrow Y\ for\ X \mid \Downarrow Y\ for\ X \mid$$
$$H : +Y\ dcv\ X \mid H : -Y\ dcv\ X \mid H : \Uparrow Y\ dcv\ X \mid H : \Downarrow Y\ dcv\ X$$
$$\phi ::= p \mid \neg\phi \mid (\phi \wedge \phi) \mid P_i\phi \mid \langle U \rangle \phi$$

The modality $P_i\phi$ is read as "agent i possibly believes ϕ". The dual modality $B_i\phi = \neg P_i \neg \phi$ is read as "agent i believes ϕ". The other modalities are action modalities, $\langle U \rangle \phi$ is read as "after possible update U, ϕ holds". The dual modality is $[U]\phi = \neg\langle U \rangle \neg \phi$. The updates will be explained through examples in Section 3.

Each action operator U is provided a specific action frame $F(U)$. More specifically, given these fixed frames (defined in Section 3), the semantics of AUL can be defined as follows, using Definition 3 for sum-product update. We use u for the designated event of the update.

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

- $M, s \models \top$ *(always),*
- $M, s \models p \Leftrightarrow s \in V(p)$,
- $M, s \models \neg\phi \Leftrightarrow$ *not* $(M, s) \models \phi$,
- $M, s \models (\phi \wedge \psi) \Leftrightarrow (M, s) \models \phi$ *and* $(M, s) \models \psi$, *and*
- $M, s \models P_i\phi \Leftrightarrow$ *for some* t, sR_it *and* $(M, t) \models \phi$
- $M, s \models \langle U \rangle \phi$ *iff* $(M * F(U), (s, u)) \models \phi$

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

We work only with *transitive* relations R_i, hence $B_i\phi \implies B_iB_i\phi$ is a valid formula. It says that positive belief is introspective. In our models, $\neg B_i\phi \implies B_i\neg B_i\phi$ is not a valid formula. It says that negative belief is introspective. Chellas has a textbook treatment of modal logic [9] which describes such correspondences of valid formulas with properties of Kripke frames.

Independently of Wang *et al* [30] which has the same idea, we model existence of agents at a world using presence of that agent's accessibility at the world. We sometimes use the "agency" of an agent i, by which we mean: 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, $B_jP_i\top$.

2.3 Proof system

The proof system gives axioms and inference rules to prove valid formulas. There are 8 axioms below and 2 standard inference rules. Several axioms for the update operators will be presented in Section 3.

1. all instances of propositional tautologies
2. $B_a(\phi \implies \psi) \implies (B_a\phi \implies B_a\psi)$
3. $B_a\phi \implies B_aB_a\phi$
4. $[U](\phi \implies \psi) \implies ([U]\phi \implies [U]\psi)$
5. $[U]p \Leftrightarrow (pre(u) \implies p)$
6. $[U]\neg\phi \Leftrightarrow (pre(u) \implies \neg[U]\phi)$
7. $[U](\phi \wedge \psi) \Leftrightarrow ([U]\phi \wedge [U]\psi)$
8. $\langle skip\rangle\phi \iff \phi$
9. From ϕ and $\phi \implies \psi$, infer ψ
10. From ϕ, infer $B_a\phi$

3 Agent-update actions and their logic

In this section, we will examine different kinds of agent-update actions.

We first identify a set of agents whose beliefs remain unchanged at an event in an agent-update frame.

Definition 6 (Observer). *The set of* observers $Obs(u)$ *at an event* u *in an agent-update frame is those* j *with agency at* u *such that* $uO_jv \iff v = u$.

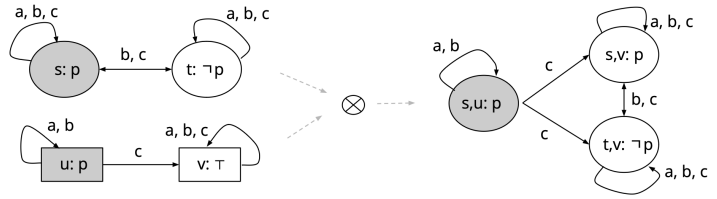
A subset of these are deceivers. In brief, the deceived come to believe the situation depicted at event v observable from u. But at u, the deceivers' beliefs are unchanged.

Definition 7 (Deceiver). *In an agent-update frame if event* v *is observable at* u *by* X *(*uO_Xv*), the set of* deceivers $Dcvr(u, v)$ *is observers at* u, $Dcvr(u, v) \subseteq Obs(u)$, *whereas the* deceived $Dcvd(u, v)$ *are those* $D \subseteq X$ *disjoint from* $Dcvr(u, v)$ *such that* $D \cup Dcvr(u, v)$ *are observers at* v.

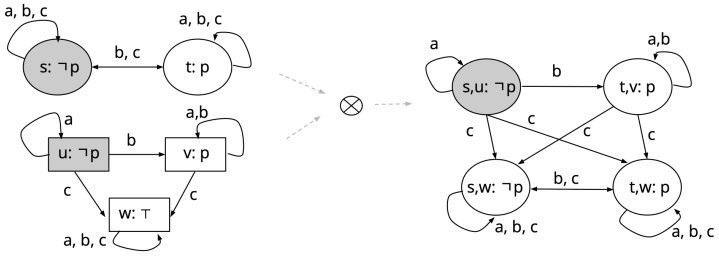
Agents from A which are added, deleted, observed at u or deceived at v, or to whom information is communicated participate in an action. We call other agents *remote*.

The following axioms are validities. The first axiom says that no beliefs change for remote agents. The next axiom is the epistemic action axiom (we call it belief-action) which is common in the literature [3, 2, 28]. It says that for agents which are observers at the designated event u, beliefs after the update can be reduced to beliefs before the update.

11. $\langle U\rangle P_k\phi \iff P_k\langle skip\rangle\phi$, for $k \in A \setminus (Add() \cup Del() \cup Obs(u) \cup Dcvd(u, v))$
12. $\langle U\rangle P_j\phi \iff P_j\langle U\rangle\phi$, for $j \in Obs(u)$



(a)



(b)

Fig. 1: (a)Alice informing Bob, $F(p\ for\ \{a,b\})$; (b)Alice lying to Bob, $F(a : p\ dcv\ b)$

3.1 Private update and lying

Example 1. Suppose Alice knows the truth value of proposition p and Bob does not know. The actions of Alice telling Bob the truth value of p ($p\ for\ \{a,b\}$) and Alice lying to Bob about the truth value of p ($a : p\ dcv\ b$) are depicted in Figure 1 [29].

In Figure 1a, top left is a Kripke model, bottom left is an action frame with $Obs(u) = \{a,b\}$ and on the right is the product Kripke model. Alice telling Bob that p is true is modelled with a single event with precondition p, such that both Alice and Bob believe p after the update. Another agent c is unaffected.

Whereas in Figure 1b above left is a Kripke model, below left is an action frame with $Obs(u) = \{a\}, Dcvr(u,v) = \{a\}$ and $Dcvd(u,v) = \{b\}$, on the right is the product Kripke model. Alice lying to Bob that p is true is modelled using v with precondition p, representing perception of Bob, while event u with precondition $\neg p$ and with an outgoing Bob-arrow to event v is the perception of Alice. Agent c remains unaffected.

The next axiom expresses a validity about information communicated during an update. The remote agent axiom covers the other agents.

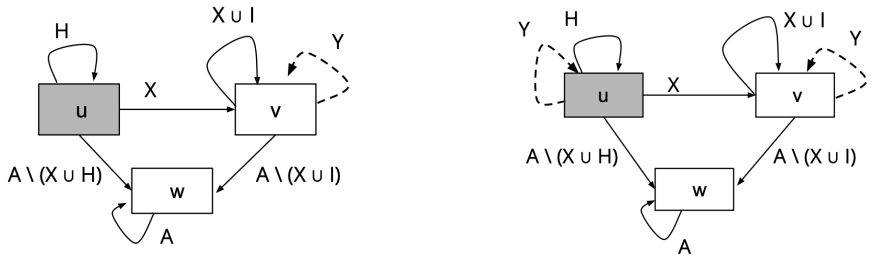
13. $\langle p\ for\ X\rangle P_j\phi \iff (p \wedge P_j\langle p\ for\ X\rangle\phi)$, for $j \in X = Obs(u)$

Next we have the axioms for lying. The first reduces to the previous truthful update axioms. This is a pattern which we will repeatedly see in deception. The belief-action axiom covers agents in $H = Dcvr(u) = Obs(u)$ and the remote agent axiom covers the rest.

14. $\langle H : p\ dcv\ X\rangle P_j\phi \iff (\neg p \wedge P_j\langle p\ for\ X \cup H\rangle\phi)$, for $j \in X = Dcvd(u, v)$

3.2 Forgery with deception and without

The first agent-update operator we consider is agent-addition $H : +Y\ dcv\ X$ which is deceptive [20, 21]. For our example we consider a generalization where the deceivers H do not reveal themselves but use a forged message pretending to be from I.



(a) $F(H\ forge\ I : +Y\ dcv\ X)$ (b) $H\ forge\ I : +Y\ for\ X$

Fig. 2: Forgery

Example 2. Figure 2a is from *The final problem* of Sherlock Holmes [12] with $H = \{Moriarty\},\ X = \{Dr.Watson\},\ and\ I = \{Innkeeper\}$.

Moriarty wants to deceive Dr Watson away from Holmes by saying that there is a lady who is ill at their inn and needs his help. Such an attempt would not succeed because Watson would not believe Moriarty. So Moriarty forges a letter from the innkeeper, and Watson gets deceived. Thus Watson believes the innkeeper knows of the existence of the lady, whereas Moriarty knows that the innkeeper knows nothing.

We consider lying $H : +Y\ dcv\ X$ plausible when Y are new agents, thus the deceived are credulous. The set of agents is now $A' = A \cup Y$. When all the Y are new fictitious agents (we restrict to $Y \cap A = \emptyset$), the next axiom is valid. For observers and remote agents, we do not repeat the axioms.

15. $[H\ forge\ I\ :\ +Y\ dcv\ X]B_l\bot$, for $l \in Y = Add(v)$ (so $\langle H\ forge\ I\ :\ +Y\ dcv\ X\rangle P_l\phi \iff \bot$)
16. $\langle H\ forge\ I : +Y\ dcv\ X\rangle P_j\phi \Leftrightarrow P_j\langle +Y\ for\ X \cup I\rangle\phi$, for $j \in X = Dcvd(u, v)$

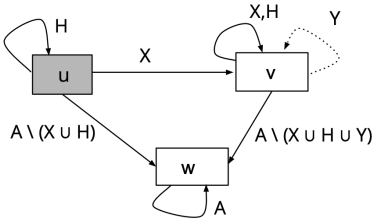
The second axiom above is a belief-action which reduces the deceptive agent-addition operator to a private agent-addition operator which is described next.

Example 3. Figure 2b illustrates a variant of the Sherlock Holmes story if such a lady did exist. For example, Moriarty could send a lady agent to the inn who could then have pretended to be ill. The forged message from the innkeeper would say a lady is arriving at the inn and has requested a physician's help. The innkeeper is not aware of the existence of the lady.

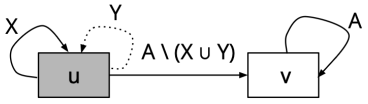
17. $\langle H\ forge\ I : +Y\ for\ X\rangle P_l\phi \Leftrightarrow \phi \vee \bigvee_{h\in H} P_h\langle H\ forge\ I : +Y\ for\ X\rangle\phi$, for $l \in Y = Add(u) = Add(v)$ (so $\langle H\ forge\ I : +Y\ for\ X\rangle P_l\top$ is valid)
18. $\langle H\ forge\ I : +Y\ for\ X\rangle P_j\phi \Leftrightarrow P_j\langle +Y\ for\ X \cup I\rangle\phi$, for $j \in X = Obs(v)$

3.3 Agent-deletion with deception and without

In modelling the gruffalo story we use commonsense conditions from AI, which include *actors*: an action is carried out by an actor. Initially we have the agent set $A = \{m, f, o\}$. Associated with this is a commonsense order $Co = \{f > m, o > m\}$ reflecting that foxes and owls eat mice. Co does not have any agent $a > f$ or $a > o$. For example, the action of fox eating mouse has precondition $P_m\top \wedge P_f\top$ and postcondition $\neg P_m\top$. Candel Bormann points out [8] that this order underlies the story. In the action language of Baral *et al* [5], additional predicates $present(m)$ and $present(f)$ are used to denote that these agents are at the initial location in order to being part of the set A. In our modelling locations play no role so we dispense with these conditions.



(a) $(F(H : -Y\ dcv\ X), u)$ for *deceptive agent-deletion* update



(b) $(F(-Y\ \ for\ \ X), u)$ for *private agent-deletion* update

Fig. 3: Agent-deletion

In Figure 3a, deceivers H, whose beliefs about agency of Y (which are neither deceiver nor deceived) are unchanged at u, deceive X into believing that agents $Y \subseteq A \setminus (X \cup H)$ have been privately deleted at v for themselves and for H.

In a private deletion, agents Y are selectively deleted for observers in $X \subseteq A$ ($Y \subseteq A \setminus X$), at event u in an agent update frame. The remaining agents are oblivious at v.

Example 4. In Figure 3b, the dotted self-loop for Y at u could stand for the mouse $i \in Y$ being eaten by the fox f, observed by others $X(f \in X)$, the rest of

the animals being oblivious of the meal. The beliefs of the rest of the animals in $A \setminus X$ are unchanged, v is a skip. In particular, the animals in $A \setminus X$ believe in the agency of i at v.

Based on the commonsense order, we have the actions $Act = \{f : -m\ for\ X, o : -m\ for\ X\}$, for subsets $X \subseteq A$. We will make up the action syntax as we go along, it is copied from the AUL update modalities and only meant to informally refer to the actions. The actors are f and o respectively (which do the eating). The actions $-f\ for\ X$ and $-o\ for\ X$ of fox or owl being eaten are not in Act since they do not respect the commonsense order: there is no agent eating them.

Because no i-arrows remain after i-deletion, the next axiom is valid. The next two axioms follow from belief-action axiom.

The next three axioms for deceptive agent-deletion follow from belief-action axiom.

19. $\langle H : -Y\ dcv\ X\rangle P_i\phi \Leftrightarrow P_i\langle skip\rangle\phi$, for $i \in Y = Del(v) \cup A \setminus (Del(v) \cup Obs(u) \cup Dcvd(u, v))$
20. $\langle H : -Y\ dcv\ X\rangle P_h\phi \Leftrightarrow P_h\langle H : -Y\ dcv\ X\rangle\phi$, for $h \in H = Dcvr(u, v) = Obs(u)$
21. $\langle H : -Y\ dcv\ X\rangle P_j\phi \Leftrightarrow P_j\langle -Y\ for\ (X \cup H)\rangle\phi$, for $j \in X = Dcvd(u, v)$

Here is the key axiom for private agent-deletion.

22. $[-Y\ for\ X]B_i\bot$, for $i \in Y = Del(u)$. So $\langle -Y\ for\ X\rangle P_i\phi \iff \bot$.

Fox tries to convince gruffalo The initial situation in the *Gruffalo's child* story is modelled with M_0 with a designated world s as is shown in Figure 4. $(M_0, s) \models P_g\top \wedge P_f\top \wedge P_o\top \wedge P_m\top \wedge \neg P_m p$. By the proposition is meant $p \iff \neg P_g\top$, that is, a "big bad mouse" is one which eats gruffalos.

Example 5. In *The Gruffalo's child*, the first move is fox telling the gruffalo g of a mouse which likes to eat fox. This move is modelled as a combination of the addition and deletion actions. We write it as a $f : (-g)\ for\ g$ action. At $v1$, the fox believes the mouse believes in eating gruffalos, which we represent as a deletion of gruffalo at $x1$. At $u1$, the gruffalo does not buy the belief. Another agent, the owl, is oblivious of this interaction at $w1$.

Owl tries to convince gruffalo

Example 6. The mouse runs into the owl after deceiving fox and makes a deceptive move again, $o : (-g)\ for\ g$, as before a combination of an addition and a deletion in Figure 5. At $v2$ the owl believes that the mouse believes in eating gruffalos. At $u2$ the gruffalo does not accept believing this, with fox being oblivious of the interaction.

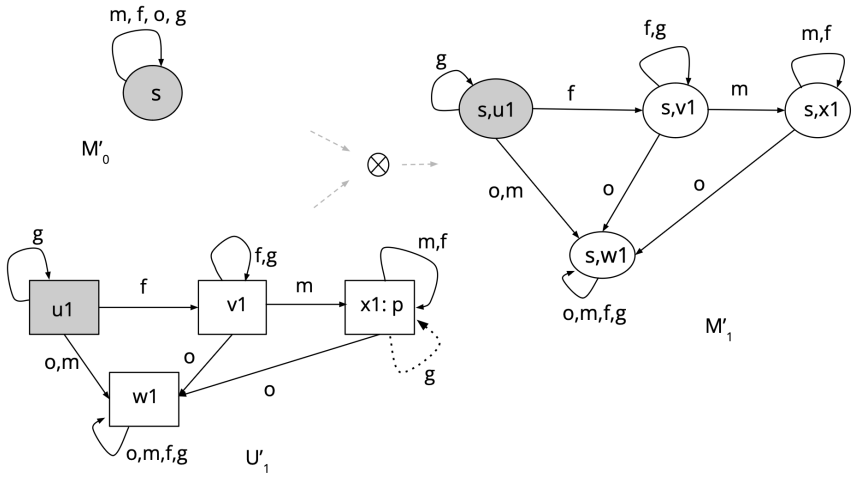


Fig. 4: Fox tries to convince gruffalo of a gruffalo-eating mouse

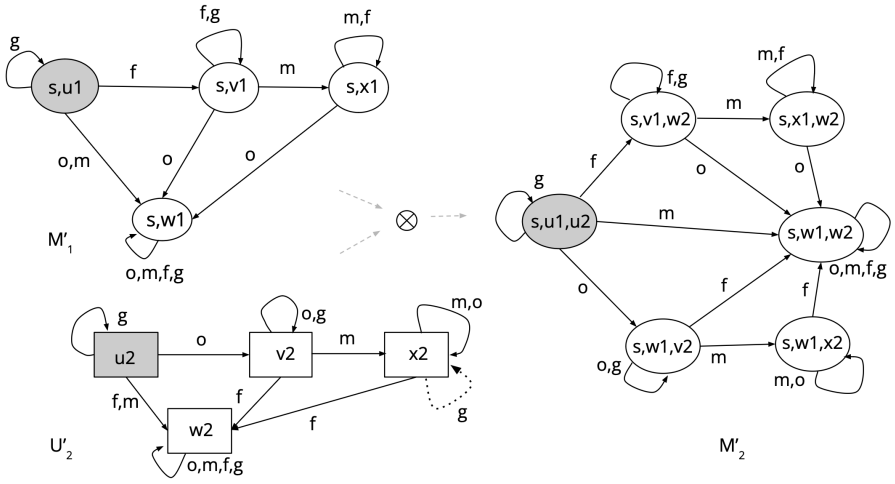


Fig. 5: Owl tries to convince gruffalo that there is a gruffalo-eating mouse

3.4 Private agent-addition: +Y for X

In Figure 6, agents Y are selectively added at event u for observers $X \subseteq A$ ($Y \cap X = \emptyset$) in an agent frame. Agents in Y can be outside A. Event v is a skip event that does not change anything for anyone. At u, agents in $A \setminus X$ believe that event v occurs; they consider that all agents in A are observers at v.

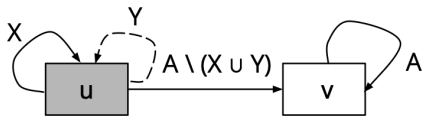


Fig. 6: $(F(+Y\ for\ X), u)$ for *private agent-addition* update

Example 7. In Figure 6, let a new agent owl $i \in Y$ appear in the action $+Y\ for\ X$ as indicated by the dashed arrow. The actor is Y, so we could write it as $Y : +Y\ for\ X$. The mouse m is present on the scene at u, it constitutes $X(m \in X)$. Other animals such as the fox f in $A \setminus X$ are unaware of the agency of i at this moment. They believe that nothing happens at v (a skip).

The agent set $A = \{m, f, g\}$ is expanded to $A' = \{m, f, g, i\}$. The commonsense order Co is unchanged, it has $g > m$ from the earlier introduction of the gruffalo by fox. An action $a : +i\ for\ X$ with $i > a$ does not respect the commonsense order, how would a commandeer such a performance?

When all the Y are new agents (so we restrict to $Y \cap A = \emptyset$), the next axiom is a valid equivalence. The next two axioms follow from the belief-action axiom.

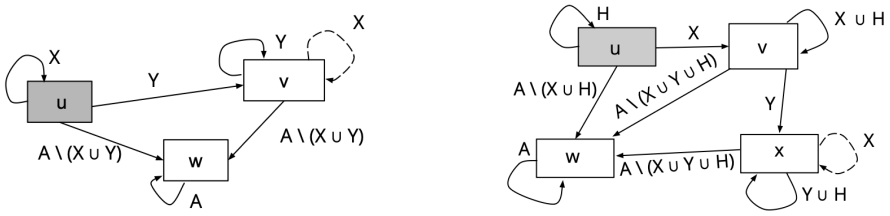
23. $\langle +Y\ for\ X\rangle P_i\phi \Leftrightarrow \phi \vee \bigvee_{j\in X} P_j\langle +Y\ for\ X\rangle\phi$, for $i \in Y = Add(u)$ (so $\langle +Y\ for\ X\rangle\top$ is valid)

3.5 Downgrade and deceptive downgrade

Figure 7a shows an agent-downgrade action. In the literature with ordered Kripke models [24, 4], such updates typically refer to a proposition. For example an action $\Downarrow p$ would place worlds satisfying p below worlds that do not satisfy p. Our interest in [22] was in existence of agents, where we introduced agent-addition and agent-deletion operations. In this paper, we attempt integrating these ideas into commonsense situations which appear in AI modelling, which are represented by the order Co. Hence agent-downgrade (and agent-upgrade) actions will affect propositional values related to the commonsense order.

Example 8. The agent-downgrade action is motivated by our story *The Gruffalo's child*. Here $g \in X$ at event u downgrades the mouse $m \in Y$ at event v, which it had heard of from fox and owl as eating gruffalos, to one which does not eat gruffalos. That is, the commonsense order Co is updated to remove $m > g$. This has the postcondition $P_m\top$, f continues to be present but without its desire to eat m the mouse remains safe. However, notice that at event v, a self-loop for agent g is added. That is, if the high-grade mouse considered gruffalos as vermin which it had eaten up, the low-grade mouse allows gruffalos to peacefully co-exist.

The explanation above serves to reduce agent-downgrade to agent-addition, which provides a simple axiom.



(a) F(⇓ Y *for* X) (b) F(H : ⇓ Y *dcv* X)

Fig. 7: Agent-downgrade

24. $\langle \Downarrow Y \ for \ X \rangle P_m \phi \iff P_m \langle +X \ for \ Y \rangle \phi$, for $m \in Y = Obs(v), X = Add(v)$

The deceptive downgrade $\langle H :\Downarrow Y \ dcv \ X \rangle \phi$ removes j-deletion arrows ($j \in X$), however H does not believe that j is not capable of eating Y, as shown in Figure 7b.

25. $\langle H :\Downarrow Y \ dcv \ X \rangle P_g \phi \iff P_g \langle \Downarrow Y \ for \ (X \cup H) \rangle \phi$, for $g \in X = Add(x)$

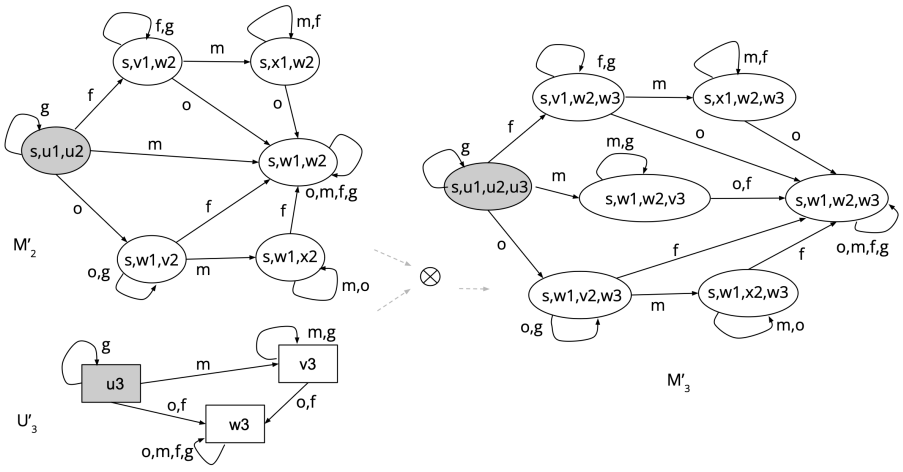


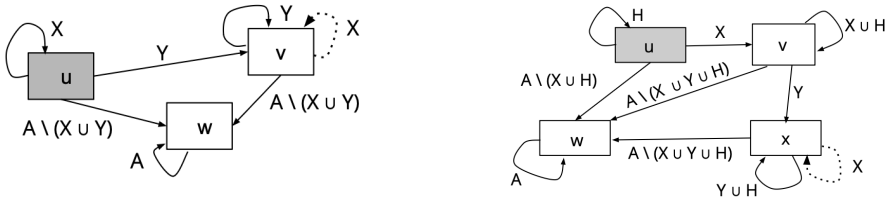
Fig. 8: Mouse appears for gruffalo

The mouse appears

Example 9. Further in *The Gruffalo's child*, the gruffalo runs into a mouse which is not big and bad. We model this as a downgrade ⇓ $m \ for \ g$ about m appearing for gruffalo. m doesn't have any g-deletion arrow. See update $U3$ illustrated in Figure 8.

3.6 Upgrade and deceptive upgrade

An i-upgrade for j is one which adds the possibility of j-deletion as shown in Figure 9a.



(a) F($\Uparrow Y$ *for* X) (b) F($H : \Uparrow Y$ *dcv* X)

Fig. 9: Agent-upgrade

The deceptive upgrade also adds j-deletion arrows for X. Beliefs of $A \setminus X$ as well as H about Y's capabilities will be unaffected as shown in Figure 9b.

26. $\langle \Uparrow Y \ for \ X \rangle P_m \phi \iff P_m \langle -X \ for \ Y \rangle \phi$, for $m \in Y = Obs(v), X = Del(v)$

As usual, the axioms for deceptive upgrade use those for upgrade.

27. $\langle H :\Uparrow Y \ dcv \ X \rangle P_g \phi \iff P_g \langle \Uparrow Y \ for \ (X \cup H) \rangle \phi$, for $g \in X = Del(x)$

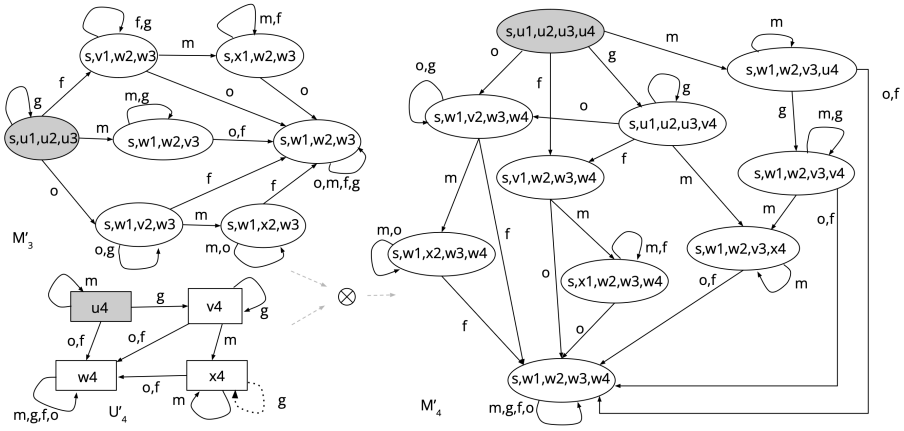


Fig. 10: Mouse deceives gruffalo that it is the big bad mouse, $F(m : \Uparrow m \ dcv \ g)$

Mouse deceives gruffalo

Example 10. Further in the story, the mouse deceives the gruffalo $m :\Uparrow m\ dcv\ g$ by showing m capable of eating g. This makes the upgraded m have a $-g$ arrow, although m itself does not believe in its upgraded capability. Owl is oblivious at $w4$. This is illustrated in Figure 10.

Mouse uses the Moon to implement this projection action, $m\ uses\ Moon : \Uparrow m\ dcv\ g$. Modelling these ideas require several location properties in the planning domain, which are ignored in our simple setup.

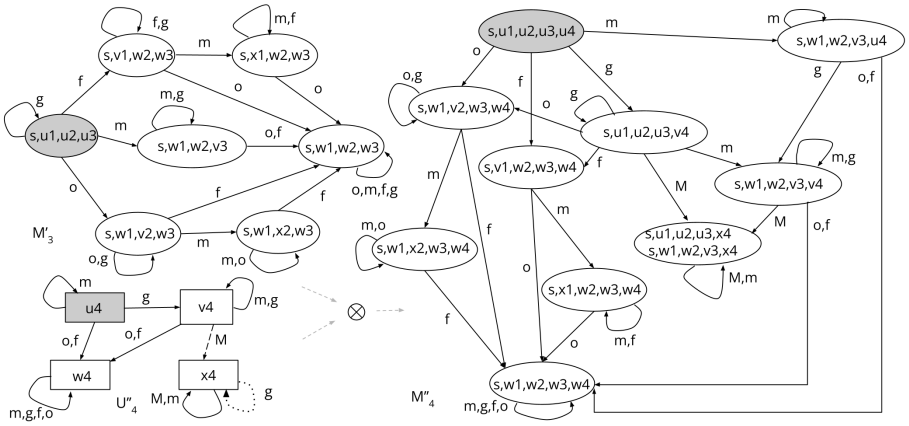


Fig. 11: Mouse deceives gruffalo that there is a big bad mouse

Mouse deceives gruffalo another way

Example 11. We present an alternate modelling. This may be what the author intended in *The Gruffalo's child*, since the mouse talks about a friend, although there is some ambiguity in the book. In this model the mouse deceives the gruffalo that there is a *different* big bad mouse M with another combination of agent-addition and agent-deletion $m : (+M : -g)\ dcv\ g$ action, as shown in Figure 11. Mouse is an observer of event $u4$ at which g observes $+m : p$-addition at $v4$. Owl is oblivious at $w4$. In any case, the gruffalo runs away and the story has a happy ending.

4 Some results for Agent Update Logic

Theorem 1 (Completeness). *The proof system of Sections 2.3 and 3 is sound and complete over transitive Kripke models.*

Proof. For the proof we define the lexicographic size of a formula, following the DEL book [28, Definition 7.38]. For all formulas, this is as expected, for example $\ell(P_a\phi) = 1 + \ell(\phi)$; except only the update formula: $\ell(\langle U\rangle\phi)$ is defined

as $(4 + \ell(U))\ell(\phi)$. This means that the lexicographic size of the left hand side of every bi-implication in the proof system is greater than the lexicographic size of its right hand side. For example, for axiom 23: $\langle +Y \ for \ X\rangle P_j\phi = \phi \vee \bigvee_{j\in X} P_j\langle +Y \ for \ X\rangle\phi$, it can be shown that $(4+n+k)(1+m) > (1+n)+(4+n+k)m$ where n is $|X|$, k is $|Y|$ and m is $\ell(\phi)$. Thus a reduction algorithm can apply these equivalences to go from an AUL formula to an equivalent EL formula. EL is complete over transitive Kripke models [9]. □

The reader may ask what is achieved by having a proof system with 2 general axioms and 15 specific axioms for 10 update operations (excluding *skip*), compared to the couple of axioms for a single general update operation in [22]. We will discuss this in the context of AI planning in Section 5.

Next we provide decision procedures mentioned in [22]. The proof of the first theorem follows from the completeness argument.

Theorem 2 (Satisfiability). *There is a polynomial space algorithm to check satisfiability of an AUL formula.*

Theorem 3 (Model checking). *Given a transitive Kripke model, checking whether an AUL formula holds at a designated state can be done in polynomial time.*

Proof. A labelling algorithm can be implemented by saving the action updates as one proceeds inwards in the formula (without performing the updates). On evaluating a belief modality which requires an agent relation in the updated model, the relation after the updates is calculated by using the saved updates. An extra multiplication by the length of the formula is needed for the number of modalities this has to be done for. This gives a polynomial time algorithm for model checking on transitive models. □

5 Planning

The planning community has traditionally worked with a fixed set of actions Act, and a planning problem is defined as a triple $\langle S, Act, G\rangle$ where S is the start state that is completely specified, and G is the set of goal predicates that are desired to be true. The goal predicates G may be true in many states, and any one of them may be acceptable to the planner. For example, in the *Gruffalo's child* story both the mouse running away and the gruffalo running away would satisfy the goal of the mouse being alive.

In the real world the set of actions available to an agent may virtually be unlimited, limited only by the agent's imagination. For a planner, considering a much larger set of actions may be intractable. In the real world agents normally pick a familiar sequence of actions that have been known to work in the past. For example a traveller may choose between going by bus to the airport or hiring a taxi based on time and money constraints. But what if there were to be a taxi strike and time is running out? In that scenario the traveller may think of the

option of calling up a friend to drop him to the airport, an action that one would not normally consider.

We propose that when the set of operators may be potentially unlimited, one can prescribe a graded set of partitions that are accessible to the planner, in a lazy evaluation manner. Thus the set Act may be partitioned into an ordered set of subsets $Act_1 \subset Act_2 \subset \ldots$. The planner can now operate with an iterative broadening algorithm in which it begins searching for a plan with the minimal set Act_1 and under certain conditions broadens it to Act_2, and so on. This broadening could be when a plan cannot be found within a reasonable time with only the set Act_1, but there could be other conditions too involving sub-goal interaction.

The above scenario is exemplified in the stories that we are considering where desperate agents seek desperate solutions, often in life threatening situations. For example, the default plan that a mouse may have is to flee in the presence of a hungry predator, but spatial proximity may prohibit that, prompting it to think of other options. In the original story by Donaldson [14], the mouse invents the Gruffalo, with fingers crossed that the predator will swallow the story. And when the unexpected happens and the relieved mouse next encounters the Gruffalo in flesh and blood, it is compelled to spin yet another yarn.

The approach that we are advocating is to not limit a planner to a fixed set of actions, but have access to graded sets of actions when a plan with fewer actions cannot be found. The actions in the extended sets may be computationally more demanding, or may have a lesser chance of success.

The goal for the planner is $P_m\top$ after one step. Informally speaking we have a set of actions Act_1 available to model check the transitive closure $\langle Act_1^+\rangle P_m\top$ (this is outside the logic AUL) at the initial state $[11, 17]$. Since $[Act_1]\neg P_m\top$, the goal is unreachable after 1 step, the base of the transitive closure. Thus from this Act_1, the planner moves to a larger set of actions Act_2.

To reach the goal we restrict ourselves to actions that add agents from Section 3, that only alter matters related to agent existence. Since the possible agents form an infinite set, for a practical solution we will have to use some rules about how to go about adding agents.

First rule We use the following commonsense inference rule. Suppose $s \models P_f\top \wedge P_m\top$. If $s \models \bigwedge_{a>m, a\neq f\in A} \neg P_a\top$, for $Co \supseteq \{f > m\}$, then add fresh $g \notin A$ to get $A' = A \cup \{g\}$ with $Co' = Co \cup \{g > f\}$. The word "fresh" indicates that the agents outside A form one equivalence class; an arbitrary g is chosen from them, thus dividing the equivalence class into $\{g\}$ which gets added to A' and another equivalence class of the agents outside A'.

Let Act be the current set of actions. Consider $Act \cup \{g : -f\}$. The new action is not applicable since $P_g\top$ is false in the initial state.

So one generates a new action, $+g\ for\ f$ which has the postcondition $P_g\top$ for the sub-goal. This action does not identify an actor. Try $m : +g\ for\ f$ using agent m as actor. ($g : +g\ for\ f$ is useless because precondition $P_g\top$ is not

met.) But the commonsense order $m < f < g$ says mouse cannot commandeer a gruffalo to appear for fox.

So one generates a new action $m : +g\ dcv\ f$. This action has an actor present, respects commonsense (assuming a credulous fox) and achieves the desired subgoal. The fact that gruffalo g is fictitious helps in plausibility. So the action set expands to $Act' = Act \cup \{g : -f, m : +g\ dcv\ f\}$.

This process can be repeated for the owl. The goal $P_m\top$ is reached.

Second rule Here is another inference rule. If $P_m\top \wedge P_f\top \wedge \bigwedge_{f>a>m\in A} \neg P_a\top$, then add fresh $h \notin A$ with the new commonsense order $Co' = Co \cup \{f > h, h > m\}$ on the expanded agent set $A' = A \cup \{h\}$.

By the reasoning process we saw above, this will eventually lead to an action $m : +h\ dcv\ f$ being added to Act. For example, mouse leads the fox to believe there is a hen which is more delicious than itself. The mouse has to still find a way to escape, but for the moment the action is plausible as it preserves $P_m\top$. It requires a planning domain where the story will move to a location where mouse can escape. This is basically the action in Book 4 of the *Panchatantra* stories [6, 19] where the monkey who foolishly asked a crocodile to ferry it across the river on its back, only to find itself being considered a meal, tells the crocodile it has left its most delicious heart on the shore, and exhorts the crocodile to swim to the river bank so that the heart can be recovered. This requires a planning domain where river and its bank are modellable.

Remark 4 (Historical). The *Panchatantra* stories are dated to the 3rd century. One of the stories appears in sculpture at a Nalanda temple (7th century).

Third rule Here is another inference rule. In a commonsense order with $m > g \in Co$, remove $m > g$ to get $Co' = Co \setminus \{m > g\}$. This is the essential idea behind the action of agent-downgrade. In the *Gruffalo's child*, the downgrade leads to $Co' = (Co \setminus \{m > g\}) \cup \{g > m\}$.

These are only suggestions towards a planning-oriented view of the agent-update logic.

Conclusion

Van Ditmarsch, Van der Hoek and Kooi's book on DEL [28, Section 6.1] has a discussion on action frames as syntax and semantics for a logic. In this paper, we suggested using an explicit syntax for our agent-update modalities. The bulk of the paper is a discussion on what kind of syntax works to model stories in an AI planning setting. The usual theoretical results of completeness and algorithms for satisfiability and model checking were obtained. Our syntactic view suggests an approach to synthesis which can be used in planning. A collaboration between people working in logic and AI can lead to fruitful results in this area.

References

1. Saul Amarel. On representation of problems of reasoning about action. In Donald Michie, editor, *Machine Intelligence 3*, pages 131–171. Edinburgh Univ press, 1971.
2. Alexandru Baltag and Lawrence S Moss. Logics for epistemic programs. *Synthese*, 139(2):165–224, 2004.
3. Alexandru Baltag, Lawrence S Moss, and Slawomir Solecki. The logic of common knowledge, public announcements, and private suspicions. In Itzhak Gilboa, editor, *Proc. 7th TARK, Evanston*, pages 43–56. Morgan Kaufmann, 1998.
4. Alexandru Baltag and Sonja Smets. A qualitative theory of dynamic interactive belief revision. In Giacomo Bonnano, Wiebe van der Hoek, and Michael Wooldridge, editors, *Logic and the foundations of game and decision theory (LOFT 7)*, pages 9–58. Amsterdam Univ press, 2008.
5. Chitta Baral, Gregory Gelfond, Enrico Pontelli, and Tran Cao Son. An action language for multi-agent domains. *Artif. Intell.*, 302(103601), 2022.
6. Georg Bühler. *Panchatantra (5 volumes)*. Bombay, 1891.
7. Richard F. Burton. *The book of the thousand nights and one night*. Kama Shashtra Society, 1888.
8. Daniel Candel Bormann. Moving possible workd theory from logic to value. *Poetics today*, 34(1–2), 2013.
9. Brian F Chellas. *Modal logic: an introduction*. Cambridge University Press, 1980.
10. Alessandro Cimatti, Marco Pistore, Marco Roveri, and Paolo Traverso. Weak, strong, and strong cyclic planning via symbolic model checking. *Artif. Intell.*, 147(1-2):35–84, 2003.
11. Alessandro Cimatti, Marco Pistore, and Paolo Traverso. Automated planning. In *Handbook of Knowledge Representation*, pages 841–867. Elsevier, 2008.
12. Arthur Conan Doyle. *The memoirs of Sherlock Holmes*. G. Newnes Ltd., 1894.
13. E.B. Cowell and R.A. Neil. *The Jatakas or stories of the Buddha's former births*. Cambridge University Press, 1907.
14. Julia Donaldson. *The Gruffalo*. Pan Macmillan, 1999.
15. Julia Donaldson. *The Gruffalo's child*. Pan Macmillan, 2004.
16. Hergé. *Prisoners of the sun*. Casterman, 1949.
17. Yanjun Li, Quan Yu, and Yanjing Wang. More for free: a dynamic epistemic framework for conformant planning over transition systems. *J. Log. Comput.*, 27(8):2383–2410, 2017.
18. Benedikt Löwe, Eric Pacuit, and Andreas Witzel. Del planning and some tractable cases. In *International Workshop on Logic, Rationality and Interaction*, pages 179–192. Springer, 2011.
19. Arthur W Ryder. *The Panchatantra*. University of Chicago Press, 1925.
20. Chiaki Sakama. A formal account of deception. In *Deceptive and counter-deceptive machines, AAAI Fall symposia, Arlington*, pages 34–41, 2015.
21. Stefan Sarkadi, Alison Panisso, Rafael Bordini, Peter McBurney, Simon Parsons, and Martin Chapman. Modelling deception using theory of mind in multi-agent systems. *AI Commun.*, 32(4):287–302, 2019.
22. Shikha Singh, Kamal Lodaya, and Deepak Khemani. Agent-update models. *arXiv preprint arXiv:2211.02452*, 2022.
23. Raymond M. Smullyan. *What is the name of this book?* Prentice-Hall, 1978.
24. Johan Van Benthem. Dynamic logic for belief revision. *J. Appl. Nonclass. Logic*, 17(2):129–155, 2007.
25. Johan Van Benthem. *Modal logic for open minds*. CSLI, 2010.

26. Johan Van Benthem, Jan van Eijck, and Barteld Kooi. Logics of communication and change. *Inform. Comput.*, 204(11):1620–1662, 2006.
27. Hans Van Ditmarsch and Barteld Kooi. *One hundred prisoners and a light bulb*. Springer, 2015.
28. Hans Van Ditmarsch, Wiebe van Der Hoek, and Barteld Kooi. *Dynamic epistemic logic*, volume 337 of *Synthese library*. Springer Science & Business Media, 2008.
29. Hans Van Ditmarsch, Jan Van Eijck, Floor Sietsma, and Yang Wang. On the logic of lying. In Jan Van Eijck and Rineke Verbrugge, editors, *Games, actions and social software*, volume 7010 of *LNCS*, pages 41–72. Springer, 2012.
30. Yanjing Wang, Yu Wei, and Jeremy Seligman. Quantifier-free epistemic term-modal logic with assignment operator. *Ann. Pure Appl. Log.*, 173(103071), 2022.
31. John Woods. *The logic of fiction: A philosophical sounding of deviant logic*. Mouton, 1974.
32. John Woods. *Truth in fiction: Rethinking its logic*. Springer, 2018.