Diversity of Logical Agents in Games

Article *in* Philosophia Scientiae · July 2004

DOI: 10.4000/philosophiascientiae.571 · Source: CiteSeer

Diversity of Logical Agents in Games

Johan van Benthem[*], *Fenrong Liu*[†]
University of Amsterdam

Résumé :

Abstract: Epistemic agents may have different powers of observation and reasoning, and we show how this diversity fits into dynamic update logics.

[*]University of Amsterdam & Stanford University
[†]University of Amsterdam & Chinese Academy of Social Sciences

Philosophia Scientiæ, 8 (2), 2004, 165–181.

1 Varieties of imperfection

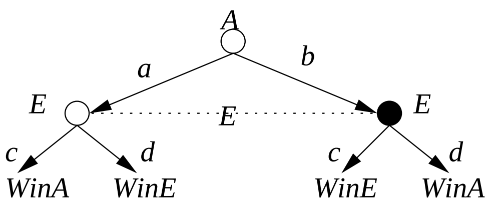
Logical agents are usually taken to be epistemically perfect. But in reality, imperfections are inevitable. Even the most logical reasoners may have limited powers of observation of relevant events, generating uncertainty as time proceeds. In addition, agents can have processing bounds on their knowledge states, say, because of finite memory. This note explores how different types of agents can be defined, and even co-exist inside the same logical system. Our motivating interest are games with imperfect information, but our technical results concern imperfect agents in current logics for information update and belief revision. For extended versions of results and discussions, cf. van Benthem 2001, van Benthem and Liu 2004, and Liu 2004.

2 Imperfect information games as models for dynamic-epistemic logic

Dynamic-epistemic language Games in extensive form are trees (S, $\{R_a\}_{a \in A}$), with nodes for successive states of play, and players' moves represented as binary transition relations between nodes. Imperfect information is encoded by equivalence relations $\sim_i$ between nodes that model the uncertainty of player i. Nodes in these structures are naturally described in a combined *modal-epistemic language*. An action modality $[a]\phi$ is true at a node x when ϕ holds after every successful execution of move a at x, and a knowledge modality $K_i\phi$ is true at x when ϕ holds at every node $y \sim_i x$. As usual, we write $\langle a \rangle$, $\langle i \rangle$ for the existential duals of these modalities. Such a language can describe many common scenarios.

Example Not knowing one's winning move

Here is a game where player ***E*** does not know the initial move played by ***A***:



The modal formula $[a]\langle d\rangle Win_E \wedge [b]\langle c\rangle Win_E$ expresses the fact that ***E*** has a winning strategy in this game, and at the root, she knows both conjuncts. After ***A*** plays move b from the root, however, in the black intermediate node, ***E*** knows merely 'de dicto' that playing either c or d is a winning move: $K_E(\langle c\rangle Win_E \vee \langle d\rangle Win_E)$. But she does not know 'de re' of any specific move that it guarantees a win: $\neg K_E\langle c\rangle Win_E \wedge \neg K_E\langle d\rangle Win_E$ also holds. In contrast, given the absence of dotted lines for ***A***, whatever is true at any stage of this game is known to ***A***. In particular, at the black intermediate node, ***A*** does know that c is a winning move for ***E***. ■

Sometimes, converse relations $a^{\cup}$ for moves a are needed, looking back up the game tree. Such an extended *temporal-epistemic language* can describe play so far, as well as what might have happened, instead of what did.

Strategies, plans, and programs More global behaviour than just single moves can be formulated in a *dynamic-epistemic language*. A *strategy* for player i is a function from i's turns x in the game to possible moves at x, while we think of a *plan* as any relation constraining these choices. Such binary relations and functions can be described using (i) single moves a, (ii) tests $(\phi)?$ on the truth of some formula ϕ, combined using operations of (iii) union $\cup$, relational composition ;, and iteration *. In particular, these operations define the usual program constructs *IF THEN ELSE* and *WHILE DO*. In this setting, it only makes sense to use test conditions $K_i\phi$ which an agent *knows to be true or false*, as in the 'knowledge programs' of Fagin, Halpern, Moses & Vardi 1995. In finite imperfect information games, the strategies defined by knowledge programs are precisely the game-theoretic *uniform strategies* (van Benthem 2001).

Valid laws of reasoning about agents and plans Our models validate the minimal modal or dynamic logic, plus the epistemic logic matching the uncertainty relations – in our case, multi-$S5$. But what about players' changing knowledge as a game proceeds? Is the following interchange principle for knowledge and action valid?

$$K_i[a]p \rightarrow [a]K_ip$$

The answer is "No". I know that I am boring after drinking – but after drinking, I need not know that I am boring. General dynamic-epistemic logic has no significant interaction axioms for knowledge and action. If such axioms hold, this is due to special features of agents.

Axioms for perfect agents In a standard modal correspondence style, the above interchange law really describes a special type of agent.

Fact $K_i[a]p \rightarrow [a]K_i p$ corresponds to the relational frame condition that *for all s, t, u, if* $sR_a t$ & $t \sim_i u$, *then there is a v with* $s \sim_i v$ & $vR_a u$.

This says that new uncertainties for an agent are always grounded in earlier ones. Van Benthem 2001 takes this as defining players' *Perfect Recall* in a game-theoretic sense: they know their own moves and also remember their past uncertainties at each stage. (More precisely, one must distinguish nodes which are turns for the relevant player from turns of others.) Other versions of Perfect Recall allow players uncertainty about the number of moves played so far. Bonanno 2004 has an account of this in our correspondence style in a temporal-epistemic language. A similar analysis works for other dynamic-epistemic axioms, such as the converse $[a]K_i p \rightarrow K_i[a]p$, whose frame truth demands a converse frame condition of 'No Learning' (cf. Fagin, Halpern, Moses & Vardi 1995).

Agents with Perfect Recall also show special behavior with respect to their knowledge about complex plans, including their own strategies.

Fact Agents with Perfect Recall validate all dynamic-epistemic formulas of the form $K_i[\sigma]p \rightarrow [\sigma]K_i p$, where σ is a knowledge program.

Proof A simple induction on programs works. For knowledge tests $(K_i\varphi)?$, we have $K_i[(K_i\varphi)?]p \leftrightarrow K_i(K_i\varphi \rightarrow p)$ in dynamic logic, and then $K_i(K_i\varphi \rightarrow p) \leftrightarrow (K_i\varphi \rightarrow K_i p)$ in epistemic S5, and $(K_i\varphi \rightarrow K_i p)$ $\leftrightarrow [(K_i\varphi)?]K_i p$ in dynamic logic. For choice and composition, the inductive steps are obvious, and program iteration is repeated composition. ∎

Thus, an agent with Perfect Recall who knows what a plan will achieve also knows these effects halfway, when only part of his strategy has been played.

Axioms for imperfect agents At the opposite extreme of Perfect Recall, agents with bounded memory only remember a fixed number of previous events. Such 'bounded rationality' is modelled in game theory by strategies defined by some finite automaton (Osborne & Rubinstein 1994). Van Benthem 2001 considers the most drastic restriction, to just the last event observed. In modal-epistemic terms, these *memory-free* agents satisfy

$$\langle a\rangle p \rightarrow U[a]\langle i\rangle p \qquad MF$$

Here the *universal modality* $U\varphi$ states that φ holds in all worlds.

Claim The axiom *MF* corresponds to the structural frame condition that, *if* $sR_a t$ & $uR_a v$, *then* $v \sim_i t$.

Thus, nodes where the same action has been performed are indistinguishable. Reformulated in terms of knowledge, the axiom becomes $\langle a\rangle K_i p \rightarrow U[a]p$. This says that the agent can only know things after an action which are true wherever the action has been performed. Either way, memory-free agents know very little indeed! We will study their behavior further in Section 4.

3 Update for perfect agents

Imperfect information trees are just a static record of players' uncertainties at the stages of a game. Not provided is a plausible *scenario* explaining these uncertainties. A mechanism for this purpose comes from *update logics* for actions with epistemic import (Baltag, Moss & Solecki 1998).

Product update A general update step has two components:

(a) an *epistemic model* $\boldsymbol{M}$ of all relevant possible worlds with agents' uncertainty relations indicated,

(b) an *action model* $\boldsymbol{A}$ of all relevant actions, again with agents' uncertainty relations between them.

Action models can have any pattern of uncertainty relations, just as epistemic models. This reflects agents' limited powers of observation. E.g., in a card game, $\boldsymbol{M}$ might be the initial situation after the cards have been dealt, while $\boldsymbol{A}$ contains all legal moves. Some actions are public, like throwing a card on the table. Others, like drawing a new card from the stock, are only transparent to the player who draws, while others cannot distinguish draws of different cards. But there is still one more element. E.g., I can only draw the Ace of Hearts if it is still on the table. Such restrictions are encoded by

(c) *preconditions* PRE_a for actions a,

which are common knowledge. In the simplest case, these are defined in the pure epistemic language of facts and agents' (mutual) information about them. Now, the next epistemic model $\boldsymbol{M}\times\boldsymbol{A}$ is computed as follows:

The domain is $\{(s,\ a) \mid s$ a world in $\boldsymbol{M}$, a an action in $\boldsymbol{A}$, $(\boldsymbol{M},\ s) \models PRE_a\}$.
The new uncertainties satisfy $(s,\ a) \sim_i (t,\ b)$ iff both $s \sim_i t$ and $a \sim_i b$.
A world $(s,\ a)$ satisfies a propositional atom p iff s already did in $\boldsymbol{M}$.

In particular, the *actual world* of the new model is the pair consisting of the actual world in $\boldsymbol{M}$ and the actual action in $\boldsymbol{A}$. The product rule says that uncertainty among new states can only come from existing uncertainty via indistinguishable actions. This simple mechanism covers surprisingly many forms of epistemic update. Baltag, Moss & Solecki 1998, van Benthem 2003, and many other recent publications provide introductions to update logics and the many open questions one can ask about them.

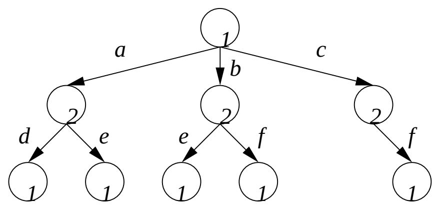
The same perspective may now be applied to imperfect information games, where successive levels correspond to successive repetitions of the sequence

$\boldsymbol{M}$, $\boldsymbol{M}\times\boldsymbol{A}$, $(\boldsymbol{M}\times\boldsymbol{A})\times\boldsymbol{A}$, ...

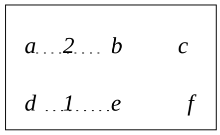
The result is an obvious tree-like model *Tree*$(\boldsymbol{M},\ \boldsymbol{A})$, which may be infinite.

Example Propagating uncertainty along a game
The following illustration is from van Benthem 2001. Suppose we are given a game tree with admissible moves (preconditions will be clear immediately). Let the moves come with epistemic uncertainties encoded in an action model:

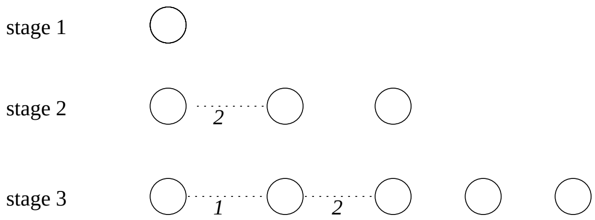
Game Tree



Action Model



Then the imperfect information game can be computed with levels as follows:



■

Now enrich the modal-epistemic language with a dynamic operator

$$\boldsymbol{M},\ s \models \langle \boldsymbol{A},\ a\rangle\varphi \quad \text{iff} \quad (\boldsymbol{M},\ s) \times (\boldsymbol{A},\ a) \models \varphi$$

Then valid principles express how knowledge is related before and after an action. In particular, we have this key *reduction axiom*:

$$\langle \boldsymbol{A},\ a\rangle\langle \mathrm{i}\rangle\varphi \leftrightarrow (PRE_a \wedge \bigvee\{\langle \mathrm{i}\rangle\langle \boldsymbol{A},\ b\rangle\varphi\colon\ a \sim_i b \textit{ for some } b \textit{ in } \boldsymbol{A}\})$$

From left to right, the axiom states Perfect Recall, adapted to our setting with indistinguishable actions. The converse implication, too, is an earlier principle. No Learning is possible for agents among indistinguishable situations by actions that they cannot distinguish.

Thus, product update is geared toward special agents. E.g., perfect memory is built in, as the clauses for $(s,\ a) \sim_i (t,\ b)$ give equal weight to

(a) $s \sim_i t$: past states representing the 'memory component',

(b) $a \sim_i b$: options for the newly observed event.

Changes in this mechanism will produce other 'product agents' by assigning different weights to these two factors (see Section 5). But first, we determine the essence of product update from the general perspective of Section 2.

Abstract characterization of product update Consider a tree structure $\mathcal{E}$ whose nodes are finite sequences X, Y, ... of events. (This allows for multiple root nodes.) Nodes can have uncertainty relations among them, and they can also interpret atomic propositions p, q, ... We think of $\mathcal{E}$ as the possible evolutions of some process – for instance, a game. A particular case is the above model $\mathit{Tree}(\boldsymbol{M}, \boldsymbol{A})$ starting from an initial epistemic model $\boldsymbol{M}$ and an action model $\boldsymbol{A}$, and repeating product updates forever. Now, the preceding discussion has shown that the following two principles hold in $\mathit{Tree}(\boldsymbol{M}, \boldsymbol{A})$. Stated as general properties of a tree $\mathcal{E}$, they express Perfect Recall and 'Uniform No Learning' (with $^\cap$ for concatenation):

PR *If* $X^\cap(a) \sim_i Y$, *then* $\exists b\, \exists Z$: $Y = Z^\cap(b)$ & $X \sim_i Z$.

UNL *If* $X^\cap(a) \sim_i Y^\cap(b)$, *then* $\forall U, V$: *if* $U \sim_i V$, *then* $U^\cap(a) \sim_i V^\cap(b)$, provided that $U^\cap(a)$, $V^\cap(b)$ both occur in the tree $\mathcal{E}$.

Also, the special preconditions in product update, definable inside the current epistemic model, validate one more abstract constraint on $\mathcal{E}$:

BIS-INV The set $\{X \mid X^\cap(a) \in \mathcal{E}\}$ of nodes where action a can be performed is closed *under purely epistemic bisimulations* of nodes.

Now we have all we need to prove a converse representation result.

Theorem For any tree $\mathcal{E}$, the following are equivalent:

(a) $\mathcal{E} \cong \mathit{Tree}(\boldsymbol{M}, \boldsymbol{A})$ for some $\boldsymbol{M}$, $\boldsymbol{A}$

(b) $\mathcal{E}$ satisfies *PR*, *UNL*, *BIS-INV*

Proof From (a) to (b) is the above observation. Next, from (b) to (a), define an epistemic model $\boldsymbol{M}$ as all initial points in $\mathcal{E}$ with their relations $\sim_i$. The action model $\boldsymbol{A}$ contains all actions occurring in $\mathcal{E}$, with:

$$a \sim_i b \quad \text{iff} \quad \exists X\, \exists Y\!: X^\cap(a) \sim_i Y^\cap(b)$$

Finally, the *preconditions* PRE_a for actions a are definable by the well-known fact that in any epistemic model, any set of worlds closed under epistemic bisimulations has a purely epistemic definition – perhaps using infinite conjunctions and disjunctions.

Now, the obvious identity map F sends nodes X of $\mathcal{E}$ to corresponding states in the model *Tree* ($\boldsymbol{M}$, $\boldsymbol{A}$). First, we observe the following fact about $\mathcal{E}$ itself:

Lemma If $X \sim_i Y$, then *length(X)* = *length(Y).*

Proof All initial points X, Y in $\mathcal{E}$, have length 0. Next, let X have length $n+1$. By *PR*, X's initial segment of length n stands in the relation $\sim_i$ to a proper initial segment of Y whose length is that of Y minus 1. Repeating this peels off both sequences to initial points after the same number of steps.

Claim $X \sim_i Y$ holds in $\mathcal{E}$ iff $F(X) \sim_i F(Y)$ holds in *Tree*($\boldsymbol{M}$, $\boldsymbol{A}$).

The proof is by induction on the common length of the two sequences X, Y. The case of initial points is clear by the definition of $\boldsymbol{M}$. As for the inductive steps, consider first the direction $\Rightarrow$. If $U^\frown(a) \sim_i V$, then by *PR*, $\exists b\ \exists Z$: $V = Z^\frown(b)$ & $U \sim_i Z$. By the inductive hypothesis, we have $F(U) \sim_i F(Z)$. We also have $a \sim_i b$ by the definition of $\boldsymbol{A}$. Moreover, given that the sequences $U^\frown(a)$, $Z^\frown(b)$ both belong to $\mathcal{E}$, their preconditions as listed in $\boldsymbol{A}$ are satisfied. Therefore, in *Tree*($\boldsymbol{M}$, $\boldsymbol{A}$), by the definition of product update, $(F(U), a) \sim_i (F(Z), b)$, i.e. $F(U^\frown(a)) \sim_i F(Z^\frown(b))$.

As for the direction $\Leftarrow$, suppose that in *Tree*($\boldsymbol{M}$, $\boldsymbol{A}$) we have $(F(U), a) \sim_i (F(Z), b)$. Then by the definition of product update, $F(U) \sim_i F(Z)$ and $a \sim_i b$. By the inductive hypothesis, from $F(U) \sim_i F(Z)$ we get $U \sim_i Z$ in $\mathcal{E}$(*). Also, by the given definition of $a \sim_i b$ in the action model $\boldsymbol{A}$, we have $\exists X\ \exists Y$: $X^\frown(a) \sim_i Y^\frown(b)$(**). Combining (*) and (**), by *UNL* we get $U^\frown(a) \sim_i Z^\frown(b)$, provided that $U^\frown(a)$, $V^\frown(b) \in \mathcal{E}$. But this is so since the preconditions PRE_a, PRE_b of the actions a, b were satisfied at $F(U)$, $F(Z)$. This means these epistemic formulas were also true at U, V – so, given what PRE_a, PRE_b defined, $U^\frown(a)$, $V^\frown(b)$ exist in the tree $\mathcal{E}$. ■

This result is only one of a kind. In many game scenarios, preconditions for actions are not purely epistemic, but rather depend on what happens over time. E.g., a game may have initial factual announcements – like the Father's saying that at least one child is dirty in the puzzle of the Muddy Children. These are not repeated, even though their preconditions still hold at later stages. This requires preconditions PRE_a that refer to the temporal structure of the tree $\mathcal{E}$, and then the above invariance for purely epistemic bisimulations would fail. Another strong assumption is our use of a single action model $\boldsymbol{A}$ that gets repeated all the time in levels $\boldsymbol{M}$, $(\boldsymbol{M}\times\boldsymbol{A})$, $(\boldsymbol{M}\times\boldsymbol{A})\times\boldsymbol{A}$, ... to produce the structure *Tree*$(\boldsymbol{M}, \boldsymbol{A})$. A more local perspective would allow different action models $\boldsymbol{A}_1, \boldsymbol{A}_2$, ... in stepping from one tree level to another. And an even more finely-grained view would arise if single moves in a game themselves can be complex action models.

4 Update logic for bounded agents

Information-processing capacity of agents may be bounded in various ways. One is 'external': agents may have restricted powers of observation. This feature is built into the above action models – and the product update mechanism reflected this. Another restriction is 'internal': agents may have bounded memory. Perfect Recall agents had limited powers of observation but perfect memory. At the opposite extreme memory-free agents can only observe the last event, without maintaining any record of their past.

Characterizing types of agents In the above, agents with Perfect Recall have been described in various ways. Our general setting was the tree $\mathcal{E}$ of event sequences, where different types of agents i correspond to different types of uncertainty relation $\sim_i$. One approach was via *structural conditions* on such relations, such as *PR*, *UNL*, and *BIS-INV*. Essentially, these three constraints say that

$X \sim_i Y$ iff *length(X)* = *length(Y)* and *X(s)* $\sim_i$ *Y(s)* for all positions s

Next, these conditions validated corresponding *axioms in the dynamic-epistemic language* that govern typical reasoning about the relevant type of agent. But we can also think of agents as a sort of *processing mechanism.* An agent with Perfect Recall is a push-down store automaton maintaining a stack of all past events and adding new observations.

Bounded memory Another broad class of agents arises by assuming bounded memory up to some fixed finite number k of positions. In general trees $\mathcal{E}$, this makes two event sequences X, Y $\sim_i$-equivalent for such agents i iff their last k positions are $\sim_i$-equivalent. In this section we only consider the most extreme case of this, viz. *memory-free agents* i:

$$X \sim_i Y \quad \text{iff} \quad \textit{last(X)} \sim_i \textit{last(Y)} \text{ or } X = Y = \textit{the empty sequence} \qquad \$$$

These agents only respond to the last-observed event. Their uncertainty relations can now cross different levels of a game tree. Examples are *Tit-for-Tat* in iterated Prisoner's Dilemma which merely repeats its opponent's last move (Axelrod 1984), or *Copy-Cat* in games for linear logic which wins 'parallel disjunctions' of games $G \vee G^d$ (Abramsky 1996).

Theorem An equivalence relation $\sim_i$ on $\mathcal{E}$ is memory-free in the sense of \$ if and only if the following two conditions are satisfied:

PR^- *If* $X^\frown(a) \sim_i Y$, *then* $\exists b \sim_i a\ \exists Z$: $Y = Z^\frown(b)$.

UNL^+ *If* $X^\frown(a) \sim_i Y^\frown(b)$, *then* $\forall U, V$: $U^\frown(a) \sim_i V^\frown(b)$, provided that $U^\frown(a)$, $V^\frown(b)$ both occur in the tree $\mathcal{E}$.

Proof If an agent i is memory-free, its relation $\sim_i$ evidently satisfies PR^- and UNL^+. Conversely, suppose that these conditions hold. If $X \sim_i Y$, then either X, Y are both the empty sequence, and we are done, or, say, $X = Z^\frown(a)$. Then by PR^-, $Y = U(b)$ for some $b \sim_i a$, and so *last(X)* $\sim_i$ *last(Y)*. Conversely, the reflexivity of $\sim_i$ plus UNL^+ imply that, if the right-hand side of the equivalence \$ holds, then $X \sim_i Y$.■

There is a characteristic modal-epistemic axiom for this case. First, set

$$a \sim_i b \quad \text{iff} \quad \exists X\ \exists Y : X^\frown(a) \sim_i Y^\frown(b)$$

Fact The following equivalence is valid for memory-free agents:

$$\langle a \rangle \langle i \rangle \varphi \leftrightarrow (PRE_a \ \&\ E \textstyle\bigvee_{b \sim_i a} \langle b \rangle \varphi)$$

Here the *existential modality* $E\varphi$ says that φ holds in at least one node. This implies axiom *MF* from Section 2. To restore the harmony of the total update logic, we also need a reduction axiom for the new device:

$$\langle \boldsymbol{A},\ a\rangle E\varphi \leftrightarrow (PRE_a \wedge E \bigvee \langle \boldsymbol{A},\ b\rangle\varphi \textit{ for some } b \textit{ in } \boldsymbol{A})$$

The process mechanism: finite automata The processor of memory-free agents is a very simple *finite automaton* creating their correct $\sim_i$ links:

States of the automaton: all equivalence classes $X^{\sim_i}$
Transitions for actions a: $X^{\sim_i}$ goes to $(X^\frown(a))^{\sim_i}$

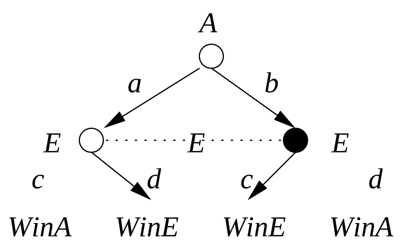
When the automaton is fed an event sequence X, it ends in state $X^{\sim_i}$. Moreover, UNL^+ and PR^- tell us that special *rigid* automata suffice:

All transitions a end in the same state (as $X^\frown(a) \sim_i Y^\frown(a)$ for all X, Y), and by PR^-, no transition ends in the initial state.

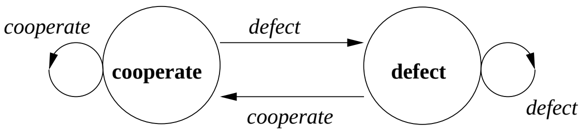
Fact Memory-free agents are exactly those whose uncertainty relation is generated by a rigid finite-state automaton.

Links with Automata Theory are in van Benthem & ten Cate 2003 (Nerode representation), Harel, Kozen & Tiuryn 2000 (action-test automata).

Strategies and automata Our automata for bounded agents are reaction devices to incoming events. But in game theory, automata define *strategies*. E.g., player $\boldsymbol{E}$'s winning strategy in the game of Section 2 is



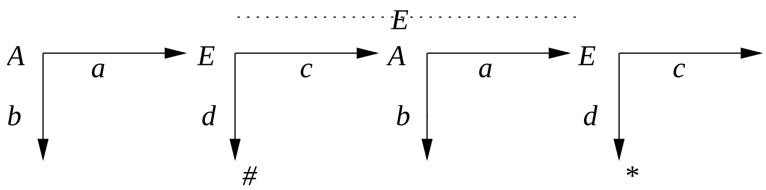
A finite automaton for this only reacts to moves by one's opponent. E.g., the one for *Tit-for-Tat* encodes the agent's actions as *states*, while those of the opponent are the observed events:



We do not undertake an integration of the two views of finite automata here.

What different agents know Memory-free agents i know less than agents with Perfect Recall, as their equivalence classes for $\sim_i$ tend to be larger. E.g., *Tit-for-Tat* only knows she is in two of the four possible matrix squares (*cooperate, cooperate*) or (*defect, defect*). But she does not know the accumulated current score. So, can memory-free agents only run very simplistic strategies? This is not quite right, as any knowledge program makes sense for all agents. The point is rather that knowledge conditions may evaluate differently. E.g., a Perfect Recall agent may be able to act on conditions like "action a has occurred twice so far", which a memory-free agent can never know to be true. Thus the difference is rather in the successful behavior guaranteed by the available uniform strategies.

Example How memory-free agents may suffer
Consider the following game tree for an agent $\boldsymbol{A}$ with perfect information, and a memory-free agent $\boldsymbol{E}$ who only observes the last move.



Suppose that outcome # is a bad thing, and $*$ a good thing for $\boldsymbol{E}$. Then the desirable strategy "play d only after you have seen two a's" is unavailable to $\boldsymbol{E}$ – while it is available to a player with Perfect Recall. ∎

Memory-free agents also know less about their *strategies*. Agent with Perfect Recall satisfied the implication $K_i[\sigma]p \to [\sigma]K_i p$ for every complex knowledge program σ. By contrast, the *MF* Memory Axiom $\langle a\rangle p \to U[a]\langle i\rangle p$ does not lift to all knowledge programs. Just look at choice actions $a\cup b$.

Memory and time So far, we had purely epistemic preconditions and forward action modalities for moves in a game tree. This misses intuitive distinctions. E.g., let there be one initial world s and an identity action Id:

$$s \qquad (s, Id) \qquad ((s, Id), Id) \qquad ...$$

Thus, each horizontal level contains just one world. In this model, the uncertainties of Perfect Recall agents and memory-free ones differ. The latter see all worlds ending in *Id* as indistinguishable, whereas product update for the former makes all worlds different. Still, all agents know the same purely epistemic statements, as all worlds are epistemically bisimilar. But levels do become distinguishable in the *temporal-epistemic language* of Section 2 with backward-looking modalities. This language is more true, then, to intuitive distinctions between players. Moreover, it can express more complex preconditions for actions, and hence a much broader range of strategies (Rodenhauser 2001). This again raises new issues of *backward-looking update* and matching reduction axioms for *postconditions* rather than preconditions of epistemic actions. We cannot purse these fascinating implications here.

5 Exploring Diversity of Agents

In between Perfect Recall and memory-free agents, there is a lot of mixed behavior. This final section suggest some general questions – elaborated in van Benthem and Liu 2004, Liu 2004. First, there is a spectrum of options in defining epistemic update rules.

Finite memory The finite automata of Section 4 can define update for progressively better informed k-bit agents having k memory cells, creating much greater diversity in behavior. And even memory-free agents ($k = 1$) have variations. E.g., 'forgetful updaters' compute uncertainty lines for worlds (w, a) without the product update clause for the world w, using only that for the action a. All these agents can be described with dynamic-epistemic reduction axioms (Liu 2004, Snyder

2004).

Probabilistic weights Agents can also give different *weights* to memory of past worlds and observation of current events in computing a new information state – as happens in inductive logic and Bayesian statistics. This requires *probabilistic* product update, defined in van Benthem 2003.

Belief revision and plausibility update Another source of variation arises in the setting of *belief revision*. Clearly, agents may have different policies, more conservative or more radical, for incorporating new information. Aucher 2003 proposes a doxastic logic whose models assigns *plausibility values* to both states and actions. Then, degrees of belief in a proposition show up as truth in all worlds up to a certain plausibility:

$$\boldsymbol{M}, s \models B_i^{\alpha}\varphi \text{ iff } \boldsymbol{M}, t \models \varphi \text{ for all worlds } t \sim_i s \text{ with } \kappa(t) \leq \alpha.$$

The update rule for models $\boldsymbol{M}\times\boldsymbol{A}$ computes new κ-values as follows:

$$\kappa'_j(w,a) = Cut_M(\kappa_j(w) + \kappa^*_j(a) - \kappa^w_j(PRE_a))$$

Here *Cut* is a 'rescaling' device, and the correction factor $\kappa^w_j(PRE_a)$ is the smallest κ-value in $\boldsymbol{M}$ among all worlds $v \sim_i w$ satisfying PRE_a.

Aucher's rule makes an agent 'eager': the last action a weighs as much as the previous state w, even though w might encode a long history of earlier beliefs. Diversity in belief revision arises with weights λ and μ:

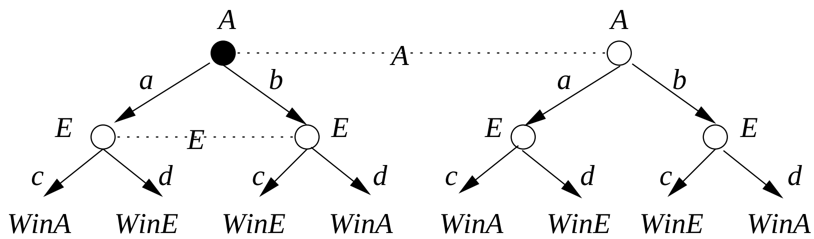
$$\kappa'_j(w,a) = Cut_M(\boldsymbol{\lambda}\kappa_j(w) + \boldsymbol{\mu}\kappa^*_j(a) - \kappa^w_j(PRE_a))$$

Liu 2004 explores many agents and policies in this λ, μ-spectrum.

So far, we charted diverse behavior for single agents. But equally important is a social aspect. Epistemic agents of different kinds can *interact*!

Mixing different types of agents Humans occasionally meet Turing machines – like their computers, or finite automata, like very stupid devices, or persons. What makes groups of agents most interesting is that they *interact*. Then, questions abound. For a start, do different types of agents *know each other's type*? They do in our models so far, where the dynamic-epistemic axioms for Perfect Recall or memory-freedom are common knowledge. Ignorance of types requires more complex models (Hötte 2003), with disjoint unions of game trees and uncertainty links across.

Example Ignorance of the opponent type
The following game is a simple variant of the example in Section 2.



At the start, agent $\boldsymbol{A}$ does not know if $\boldsymbol{E}$ has limited powers of observation. Thus, the valid law $\langle \boldsymbol{A} \rangle p \to \langle (M \cup M^{\cup})^* \rangle p$ for imperfect information games fails here. The 'second root' toward the right is an epistemic alternative for $\boldsymbol{A}$, but it is not reachable by any sequence of moves. ∎

Agents can even take advantage of knowing another's type. In the recent movie "Memento", the protagonist has lost his long-term memory and is exploited by unscrupulous cops and women. But *must* a memory-free agent do badly against a more sophisticated one? Memory-free *Tit-for-Tat* won against much more sophisticated computer programs (Axelrod 1984)...

Learning and revision of expectations over time In practice, one may have to *learn* the types of other agents. Learning mechanisms are a further source of epistemic diversity (Hendricks 2003). In general, a learning method need not reveal the type of an opponent – and agents make do with hypotheses about each other that can be refuted over time. Many issues need to be straightened out in such scenarios, including

(a) representing beliefs in addition to knowledge,

(b) counterfactual assertions about what might have happened,

(c) updating local facts about the current situation versus revising global expectations about the future.

This happens in van Benthem and Liu 2004, van Benthem 2004a, 2004b.

Merging update logic and temporal logic The preceding issues all involve *time*, just as in computational and philosophical studies of agency and planning. Branching-time models extending dynamic epistemic logic are found in Fagin, Halpern, Moses & Vardi 1995, Parikh & Ramanujam 2003. Evidently, our tree structures $\mathcal{E}$ support such a richer language.

6 Conclusion

Diversity of logical agents is a fact of life. Technically, we have characterized different kinds of epistemic agent in update logics. Next, we defined many more types of agents than the usual suspects, especially with belief revision added to the scenario. Finally, we considered issues that arise when different types of agents interact. Our results suggest many further questions, such as mathematical characterizations of agent types in settings with belief revision, and development of integrated temporal-update logics. But mainly, we hope to have shown that interaction of diverse agents is an important topic with intriguing logical repercussions.