Temporal Languages for Epistemic Programs

Joshua Sack*

October 8, 2007

Abstract

This paper adds temporal logic to public announcement logic (PAL) and dynamic epistemic logic (DEL). By adding a previous-time operator to PAL, we express in the language statements concerning *the muddy children puzzle* and *sum and product*. We also express a true statement that an agent's beliefs about another agent's knowledge flipped twice, and use a sound proof system to prove this statement. Adding a next-time operator to PAL, we provide formulas that express that belief revision does not take place in PAL. We also discuss relationships between announcements and the new knowledge agents thus acquire; such relationships are related to learning and to Fitch's paradox. We also show how inverse programs and hybrid logic each can be used to help determine whether or not an arbitrary structure represents the play of a game. We then add a past-time operator to DEL, and discuss the importance of adding yet another component to the language in order to prove completeness.

Keywords: Dynamic Epistemic Logic, Epistemic Logic, Games, Modal Logic, Public Announcement Logic, Temporal Logic,

1 Introduction

Suppose four players, Ann, Bob, Cathy, and David, have common knowledge that there are four cards. Two are indistinguishable ♠ (perhaps both 2♠ from two identical decks), one is ♢, and the third is ♣. The cards are distributed so that both Ann and Bob receive ♠, Cathy receives ♢, and David receives ♣. Then the following conversation takes place.

i. *Ann*: "I do not have ♢."

ii. *David*: "I do not have ♠."

*Department of Mathematics and Statistics, California State University Long Beach, Long Beach, CA 90840. email: jsack@csulb.edu. voice: +1-562-985-5452

iii. *Cathy*: "Before Ann's announcement, I of course knew that Bob did not know Ann's suit; but as a result of Ann's announcement, I no longer knew of Bob's uncertainty, and as a result of David's announcement, I know it again."

Note that before any announcement has been made, every agent has one of the three available suits, and thus each agent knows that the available suits for any given other agent are among at least the other two suits. As Cathy knows this about every agent, she knows in particular that Bob does not know what suit Ann has.

As Ann reveals that she has no ♢, Cathy, having the only available ♢ hears something she already knew, and learns nothing about what cards other agents have. But Cathy does however know this announcement was heard by the other agents. She knows whoever has the ♣ would know Ann's suit (and yes, thinks Cathy, if Ann herself were to have the ♣, she would of course know her own suit). Cathy also knows that whoever, other than Ann, has a ♠ would not know Ann's card. Cathy, not knowing whether Bob has the ♣ or a ♠, no longer knows whether Bob knows Ann's suit.

Finally, when David reveals that he does not have a ♠, Cathy now knows that David has the ♣ and both Ann and Bob have the ♠ cards. Cathy now knows what card everyone has, and she also knows that Bob still considers possible any of the three situations:

- Ann has ♣, Cathy ♠, and David ♢,
- Ann has ♠, Cathy ♢, and David ♣,
- Ann has ♠, Cathy ♣, and David ♢.

Thus Cathy now knows again that Bob does not know Ann's suit.

What stands out in this example is that Cathy states how her knowledge about Bob's ignorance flipped twice. A formal language that captures this should then be able to express current and past knowledge and also how knowledge changes. Epistemic logic uses modal logic to express current knowledge and belief of agents, and hence could serve as a foundation to such a formal language. Public announcement logic (PAL) lets us express what agents know as a result of public announcements. Public announcement logic is effectively a fragment of a logic, called dynamic epistemic logic (DEL), which allows for private announcements and other related actions as well as public announcements. Treatments of PAL and DEL can be found in [Baltag and Moss, 2004], [Baltag et al., 2003], [Plaza, 1989], and [Gerbrandy, 1998]. As much as we need of PAL and DEL will be given formal treatment in this paper. Since Cathy's knowledge flips twice as a result of public announcements, PAL is a natural choice of logic to use. However, PAL cannot express situations in the past, and hence cannot make reference to her past change in knowledge. By adding a previous-time operator to PAL, we can express Cathy's statement about how her knowledge flipped twice.

In section 2, we introduce a language formed by adding a previous-time operator to PAL. Other operators will be added in later sections, and we in general call a logic that adds temporal components to PAL *temporal public announcement logic* (TPAL). Section

3 provides applications to the previous-time operator. We first express a statement corresponding to the conversation involving Ann, Bob, Cathy, and David, and then introduce a sound proof system which we use to prove the statement. We next present an application to the sum and product puzzle given in [van Ditmarsch et al., 2007]. We then discuss an application of the previous-time operator given in [Yap, 2005] to the muddy children puzzle. Section 4 discusses belief change as a result of fact change rather than belief revision, and also the relationship between the content of announcements and the new knowledge agents thus obtain. We employ a next-time operator in our analysis. Section 5 is on game theory. We first present the way epistemic logic analyzes games, and then point out that moves can be treated as public announcements in PAL. The semantics used by TPAL involves structures called histories, which can record the history of a play of a game. Moreover, actions are treated similarly to programs in proposition dynamic logic. By adding to TPAL either inverse programs or hybrid logic components, we can characterize which histories represent some play of a given game. In section 6, we add the previous-time operator to DEL. We call a logic that adds temporal components to DEL *temporal dynamic epistemic logic* (TDEL). A number of examples of epistemic action are given, such as completely private and semi-private announcements. We also point out that adding another component to TDEL makes completeness easier to establish.

Absent from this paper are a complete proof system and a completeness proof itself. but there is a complete proof system and completeness proof for a class of TDEL languages given in [Sack, 2007a]. Among these complete proof systems is one for the TPAL language that adds the previous-time operator to PAL. Thus the sound proof system given in section 3 can be extended to a complete one.

2 Temporal public announcement language

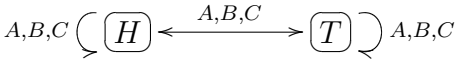
A foundation to PAL and TPAL is epistemic logic, and the structures of epistemic logic are state models (also known as epistemic models). The structures of TPAL are called histories and are constructed using state models.

Definition 2.1 (State model). Given a non-empty set Φ of proposition letters and a non-empty set $\mathcal{A}$ of agents, a *state model* is a triple $\mathbf{S} = (S, \xrightarrow{\mathcal{A}}_{\mathbf{S}}, \|\cdot\|_{\mathbf{S}})$, where

1. S is a set. Each element of S shall roughly represent a state (or possible world) of the world in consideration.
2. $\xrightarrow{\mathcal{A}}_{\mathbf{S}} = \{\xrightarrow{A}_{\mathbf{S}} \subseteq S^2 : A \in \mathcal{A}\}$. We write $s \xrightarrow{A}_{\mathbf{S}} t$ for $(s,t) \in \xrightarrow{A}_{\mathbf{S}}$, and read $s \xrightarrow{A}_{\mathbf{S}} t$ to be "in state s, A considers t possible". Such a relation is called an *epistemic relation*.
3. $\|\cdot\|_{\mathbf{S}}$ is a function mapping each atomic proposition in Φ to a subset of S. Such a function is called a *valuation*.

Typically, the subscript $\mathbf{S}$ will be absent when it is anticipated that there will be no confusion.

Example 2.2 (Concealed Coin). This example is similar to one in [Baltag and Moss, 2004]. There are three people in a room; a coin is flipped high into the air and lands in a box. No one can see the coin, and this is common knowledge. We let the set of agents be $\mathcal{A} = \{A, B, C\}$. We then define a state model where the set of states is $S = \{H, T\}$, and epistemic relations given by the labels in the following graph.



Note that the epistemic relations $\xrightarrow{A}, \xrightarrow{B}, \xrightarrow{C}$ are all equal to S^2, and they are reflected by the arrows in the diagram. In state H, all agents consider either state possible, and likewise for state T. Thus no agent can distinguish between states H and T. Let the set of atomic propositions be $\Phi = \{h, t\}$. We view the atomic proposition h as the natural language proposition "the coin landed with heads facing up", and we view t as "the coin landed with tails facing up". The valuation function shall be defined by $\|h\| = \{H\}$ and $\|t\| = \{T\}$. We say the valuation assigns an atomic proposition to a state if the state is an element of the valuation of that proposition. We think of propositions as having properties, such as "the coin landing heads up". For each atomic proposition assigned to a state, we view the state as having the properties given by that atomic proposition. Hence as $H \in \|h\|$, the proposition h is assigned to state H, and H represents a state in which the coin turns up heads. Similarly, as $T \in \|t\|$, T represents a state in which the coin turns up tails.

Now notice that in H and T, each agent considers H or T possible. The agents cannot distinguish between H and T, and they do not know whether the coin turned up heads or tails. But due to the reflexive nature of this state model, the following property $\mathfrak{P}$ holds: "no one can distinguish between H and T, and everyone knows $\mathfrak{P}$". By unraveling this circular property, we see that no one can distinguish between H and T, and everyone knows this, and everyone knows that everyone knows this, and so on. This implies that it is common knowledge that no one can distinguish between H and T.

Important to the definition of a history is the notion of a submodel of a state model.

Definition 2.3 (Submodel). Given a state model $\mathbf{S} = (S, \xrightarrow{A}_S, \|\cdot\|_S)$, a *submodel* is a state model $\mathbf{T} = (T, \xrightarrow{A}_T, \|\cdot\|_T)$ for which

1. $T \subseteq S$
2. $s \xrightarrow{A}_T t$ iff $s \xrightarrow{A}_S t$ for all $s, t \in T$.
3. $\|p\|_T = \|p\|_S \cap T$

A history will be defined effectively as a list of state models, where each subsequent model is a submodel of the previous. We simultaneously define histories with a function $\widehat{\mathbf{S}}$ that extracts the last (most recent) state model from a history.

Definition 2.4 (TPAL-History and function $\widehat{\mathbf{S}}$). We generate the set of TPAL-histories as follows:

- Any singleton list $(\mathbf{S})$ is a history, where $\mathbf{S}$ is a state model.
- If H is a history and $\mathbf{S}$ is a submodel of $\widehat{\mathbf{S}}(H)$, then $(H, \mathbf{S})$ is also a history,

We define $\widehat{\mathbf{S}}$ as a function from histories to state models that extracts the most recent state model from a history by $\widehat{\mathbf{S}}((\mathbf{S})) = \mathbf{S}$, where $\mathbf{S}$ is a state model, and $\widehat{\mathbf{S}}((H, \mathbf{S})) = \mathbf{S}$, where H is a history and $\mathbf{S}$ is a state model.

In addition to the function $\widehat{\mathbf{S}}$, we shall define two more functions on histories. The first is $\widehat{P}$, which shall return the previous history. The second is $\widehat{S}$ that returns the most carrier set of the most recent state model. As with the definition of a state model, boldfaced font represents the model, while the normal font represents the underlying set.

Definition 2.5 (Functions $\widehat{P}$ and $\widehat{S}$). We define functions $\widehat{P}$ as follows: If $H = (H', \mathbf{S})$ is a history, then $\widehat{P}(H) = H'$. If $H = (\mathbf{S})$ is a history with only one initial state model, then $\widehat{P}(H) = \emptyset$. We define $\widehat{S}$ such that $\widehat{S}(H)$ is the carrier (underlying) set of $\widehat{\mathbf{S}}(H)$. Finally, we define the function $\widehat{P}$ on the empty-set: $\widehat{P}(\emptyset) = \emptyset$.

Note on notation: We will usually write $s \in H$ for $s \in \widehat{S}(H)$.

We may iterate the function $\widehat{P}$. Thus $\widehat{P}^{n+1}(H) = \widehat{P}(\widehat{P}^n(H))$. Defining $\widehat{P}$ on $\emptyset$ allows us to apply $\widehat{P}$ to itself arbitrarily many times. It may be convenient when asking if a history H has depth less than n (has no more than n state models) that we can guarantee that $\widehat{P}^n(H)$ is defined.

2.1 Language

Like the update language in [Baltag and Moss, 2004], the languages in this paper are two-sorted, similar to PDL. We shall fix a set Φ of atomic propositions and a set $\mathcal{A}$ of agents.

Define the language $\mathcal{L}_Y^{\mathrm{PAL}}$ as consisting of sentences and programs given by

$$\varphi, \psi ::= \mathsf{true} \,|\, p \,|\, \neg\varphi \,|\, \varphi \wedge \psi \,|\, \Box_A \varphi \,|\, \Box^*_{\mathcal{B}} \varphi \,|\, [\pi]\varphi \,|\, Y_\Box \varphi$$

where $p \in \Phi$ is an atomic proposition, $A \in \mathcal{A}$ is an agent, $\mathcal{B} \subseteq \mathcal{A}$ is a set of agents, φ and ψ are sentences, and π is a program. The programs are defined by

$$\pi, \rho ::= \mathit{skip} \,|\, \mathit{crash} \,|\, \psi! \,|\, \pi \sqcup \rho \,|\, \pi; \rho \,|\, \pi^* \,|\, \pi^{-1}$$

where ψ is a sentence and both π and ρ are programs.

We define a *formula* to be a sentence, and may use the terms formula and sentence interchangeably. To emphasize the epistemic nature of the programs, we may call them epistemic programs. We call programs of the form $\psi!$ *basic (or atomic) programs.*

In TPAL and TDEL, sentences will be evaluated at coherent history/state pairs:

Definition 2.6. A *coherent history/state pair* is a pair (H, s) for which $s \in \widehat{S}(H)$.

Intuitive Meaning of Sentences and Programs

Sentences true, p, $\neg\varphi$, and $\varphi \wedge \psi$ have their usual propositional meaning, and are given a formal description below. Sentences of the form $\Box_A\varphi$ can be read as "A believes that φ". Sentences of the form $\Box^*_{\mathcal{B}}\varphi$ can be read as "It is common belief among the agents of $\mathcal{B}$ that φ". Sentences of the form $[\pi]\varphi$ can read "after each successful execution of epistemic program π, φ is true". Sentences of the form $Y_\Box\varphi$ can be read as "previously, φ was true, if there was a previous stage" (here Y stands for "yesterday").

Programs *skip*, *crash*, $\pi \sqcup \rho$, $\pi;\rho$, and π^* shall have their usual reading from proposition dynamic logic, and will be given a formal description below. The basic program $\psi!$ shall express that ψ was announced or revealed to everyone.

2.2 Semantics

The semantics will be defined by simultaneous recursion on programs and sentences. Programs will represent relations over coherent history/state pairs, and sentences will represent sets of coherent history/state pairs. We use a function $[\![\cdot]\!]$ to map each program and sentence to its meaning. Like the syntax, the semantics will be defined by mutual recursion. The semantics of programs will depend on the semantics of sentences and vice versa.

2.2.1 Programs as Relations

Given a program π, let $[\![\pi]\!]$ be a relation over coherent history/state pairs as follows. We define $(H,s)[\![\pi]\!](H',s')$ iff one of the following holds:

- π is *skip*, and $H' = H$ and $s' = s$.
- π is $\psi!$ and $H' = (H, \mathbf{S})$, such that ψ is a sentence and both $\widehat{\mathbf{S}}(H) = (S_H, \xrightarrow{A}_H, \|\cdot\|_H)$ and $\mathbf{S} = (S, \xrightarrow{A}, \|\cdot\|)$, where
  $$S = \{s \in S_H : (H,s) \in [\![\psi]\!]\},$$
  $$\xrightarrow{A} = \xrightarrow{A}_H \cap (S \times S), \text{ and}$$
  $$\|p\| = \|p\|_H \cap S.$$
- $\pi = \pi_1 \sqcup \pi_2$ and $(H,s)[\![\pi_1]\!](H',s')$ or $(H,s)[\![\pi_2]\!](H',s')$.
- $\pi = \pi_1;\pi_2$ and there exists (H'',s'') such that both $(H,s)[\![\pi_1]\!](H'',s'')$ and $(H'',s'')[\![\pi_2]\!](H',s')$.

Note that as $\pi =$ *crash* is not one of the conditions, $[\![$ *crash*$]\!]$ is the empty relation. Note also that $[\![$*skip*$]\!]$ is the reflexive relation, $[\![\pi \sqcup \rho]\!] = [\![\pi]\!] \cup [\![\rho]\!]$, and $[\![\pi;\rho]\!] = [\![\pi]\!][\![\rho]\!]$ is relation composition.

2.2.2 Sentences

The semantics of sentences is characterized by sets of coherent history/state pairs. Our notation will involve a function $[\![\cdot]\!]$ from sentences to sets of coherent history/state pairs.

1. $[\![\,\mathsf{true}\,]\!]$ is the set of all coherent history/state pairs (H, s).
2. $[\![p]\!] = \{(H, s) : s \in \|p\|_{\widehat{\mathbf{S}}(H)}\}$, for each atomic sentence $p \in \Phi$.
3. $[\![\neg\varphi]\!] = [\![\,\mathsf{true}\,]\!] - [\![\varphi]\!] = \{(H, s) \in [\![\,\mathsf{true}\,]\!] : (H, s) \notin [\![\varphi]\!]\}$.
4. $[\![\varphi_1 \wedge \varphi_2]\!] = [\![\varphi_1]\!] \cap [\![\varphi_2]\!]$.
5. $[\![\Box_A\varphi]\!] = \{(H, s) : (H', s') \in [\![\varphi]\!]$ whenever $H = H'$ and $s \xrightarrow[\widehat{\mathbf{S}}(H)]{A} s'\}$, where $A \in \mathcal{A}$.
6. $[\![\Box^*_{\mathcal{B}}\varphi]\!] = \{(H, s) : (H', s') \in [\![\varphi]\!]$ whenever $H = H'$ and $s \xrightarrow[\widehat{\mathbf{S}}(H)]{\mathcal{B}^*} s'\}$, where $s \xrightarrow{\mathcal{B}^*} s'$ iff there is a sequence
$$s = u_0 \xrightarrow{A_1} u_1 \xrightarrow{A_2} \cdots \xrightarrow{A_n} u_{n+1} = s'$$
with $A_1, \ldots, A_n \in \mathcal{B}$. We allow $n = 0$, in which case $\xrightarrow{\mathcal{B}^*}$ is the identity relation.
7. $[\![[\pi]\varphi]\!] = \{(H, s) : (H', s') \in [\![\varphi]\!]$ whenever $(H, s)[\![\pi]\!](H', s')\}$, where π is a program.
8. $[\![Y_\Box\varphi]\!] = \{(H, s) :$ if $\widehat{P}(H) \neq \emptyset$, then $(\widehat{P}(H), s) \in [\![\varphi]\!]\}$.

2.3 Abbreviations and Basic Relationships

We have our usual abbreviations $\langle\pi\rangle\varphi$ for $\neg[\pi]\neg\varphi$ and similarly for other modalities. Also, for a relation R, we define for each $n \geq 1$ $R^{n+1} = RR^n$, where RR^n represents the usual composition of the relation R with R^n. For each modality $\Box$, we define $\Box^0\varphi = \varphi$ and for $n \geq 0$, $\Box^{n+1}\varphi = \Box\Box^n\varphi$.

3 Applications of $\mathcal{L}_Y^{\mathrm{PAL}}$

3.1 Flipping of Knowledge

We return to the loss and recovery of knowledge example that was given in the introduction. Our goal here is to present a provable statement in the language $\mathcal{L}_Y^{\mathrm{PAL}}$ that both expresses initial assumptions agents have about the deal as well as asserts that Cathy's statement is true.

The set of agents is $\mathcal{A} = \{A, B, C, D\}$, where we represent each agent by the first letter of his or her name. Let the set of atomic propositions Φ be the set of all deals. We

represent a deal using a tuple such as $(\spadesuit, \spadesuit, \diamondsuit, \clubsuit)$, where deal gives Ann and Bob $\spadesuit$, Cathy $\diamondsuit$, and David $\clubsuit$. Given an atomic proposition p, let $p(A)$ be the first coordinate of p, $p(B)$ be the second coordinate of p, $p(C)$ the third, and $p(D)$ the forth. Given an agent i and a suit s, let

- $(is) \equiv \bigvee\{p : p(i) = s\}$

abbreviate the (non-atomic) proposition "i has suit s". Then define the following abbreviations in our language.

- $\mathsf{knowl}(p) \equiv \bigwedge\{\Box_i(i, p(i)) : i \in \mathcal{A}\}$
- $\mathsf{pos}(p) \equiv \bigwedge\{\Diamond_i q : i \in \mathcal{A}, p(i) = q(i)\}$.
- $\mathsf{knowl} \equiv \bigwedge\{p \to \mathsf{knowl}(p)\}$.
- $\mathsf{pos} \equiv \bigwedge\{p \to \mathsf{pos}(p)\}$,
- $\mathsf{dist} \equiv \bigwedge\{p \to \neg q : p \neq q\}$
- $\mathsf{deal} \equiv \bigvee\{p : p \in \Phi\}$.

We read $\mathsf{knowl}(p)$ as "everyone knows his/her own suit, given the deal is p", and knowl as "everyone knows his/her own suit". We read $\mathsf{pos}(p)$ as "given the deal is actually p, every agent i considers possible any deal in which i's suit is $p(i)$", and pos as "every agent i considers possible any deal in which i has his/her own suit". We read dist as "cards were dealt out at most once", and that the atomic propositions represent "distinct" deals. We read deal as "cards were dealt out" Finally, we shall represent the conditions at the beginning of the discussion in the opening example by

- $\mathsf{ant} \equiv (\spadesuit, \spadesuit, \diamondsuit, \clubsuit) \wedge \Box^*_{\mathcal{A}}(\mathsf{knowl} \wedge \mathsf{pos} \wedge \mathsf{dist} \wedge \mathsf{deal})$

As Cathy is concerned about whether Bob knows Ann's suit, we shall represent "Bob knows Ann's suit" by

- $\mathsf{BkAs} \equiv \Box_B(A\spadesuit) \vee \Box_B(A\clubsuit) \vee \Box_B(A\diamondsuit)$.

The following sentence reflects the situation given in the opening example:

$$\begin{aligned} \vdash \mathsf{ant} \to & [\neg(A\diamondsuit)!][\neg(D\spadesuit)!]Y_\diamondsuit Y_\diamondsuit \\ & (\Box_C \neg \mathsf{BkAs} \wedge [\neg(A\diamondsuit)!](\neg\Box_C \neg \mathsf{BkAs} \wedge [\neg(D\spadesuit)!](\Box_C \neg \mathsf{BkAs}))) \end{aligned} \tag{1}$$

3.1.1 Sound proof system

Let us define a proof system $\mathcal{PS}$ given by the following axiom schema and rules.

- $[\pi](\varphi \to \psi) \to ([\pi]\varphi \to [\pi]\psi)$
- $\Box_A(\varphi \to \psi) \to (\Box_A\varphi \to \Box_A\psi)$
- $\Box^*_{\mathcal{B}}(\varphi \to \psi) \to (\Box^*_{\mathcal{B}}\varphi \to \Box^*_{\mathcal{B}}\psi)$
- $Y_\Box(\varphi \to \psi) \to (Y_\Box\varphi \to Y_\Box\psi)$

Next, we shall have axioms regarding permanence of atomic propositions.

- $[\psi!]p \leftrightarrow (\psi \to p)$
- $Y_\Box p \leftrightarrow (Y_\Diamond \mathsf{true} \to p)$

Then there are partial-functionality axioms:

- $[\psi!]\neg\chi \leftrightarrow (\psi \to \neg[\psi!]\chi)$
- $Y_\Diamond\varphi \leftrightarrow (\neg Y_\Box \mathsf{false} \wedge Y_\Box\varphi)$

Here is an axiom relating basic programs to yesterday:

- $[\psi!]Y_\Box\varphi \leftrightarrow (\psi \to \varphi)$

The next set of axioms involve belief, and some are expressed with conjunctions over a set of formulas. In the case that a conjunction (as in the epistemic action axiom) turns out to be over $\emptyset$ (that is, it has no conjuncts), replace it with the formula true.

- $[\psi!]\Box_A\varphi \leftrightarrow (\psi \to \Box_A[\psi!]\varphi)$ (Epistemic action).
- $\Box^*_{\mathcal{B}}\varphi \to \varphi \wedge \bigwedge\{\Box_A\Box^*_{\mathcal{B}}\varphi : A \in \mathcal{B}\}$ (Epistemic-mix).
- $Y_\Diamond \mathsf{true} \to \Box_A Y_\Diamond \mathsf{true}$ (non-initial-time)
- $Y_\Box \mathsf{false} \to \Box_A Y_\Box \mathsf{false}$ (initial-time)
- $Y_\Box\Box_A\varphi \to \Box_A Y_\Box\varphi$ (epistemic-yesterday)

We have axioms familiar to PDL

- $[\pi;\rho]\varphi \leftrightarrow [\pi][\rho]\varphi$
- $[\pi \sqcup \rho]\varphi \leftrightarrow [\pi]\varphi \wedge [\rho]\varphi$

We have rules:

- (modus ponens) From $\vdash \varphi$ and $\vdash \varphi \to \psi$, infer $\vdash \psi$

- ($[\pi]$-necessitation) From $\vdash \varphi$, infer $\vdash [\pi]\varphi$
- ($\Box_A$-necessitation) From $\vdash \varphi$, infer $\vdash \Box_A\varphi$
- ($\Box_{\mathcal{B}}$-necessitation) From $\vdash \varphi$, infer $\vdash \Box^*_{\mathcal{B}}\varphi$
- ($Y_\Box$-necessitation) From $\vdash \varphi$, infer $\vdash Y_\Box\varphi$

The proof system $\mathcal{PS}$ is sound but not complete; a key missing rule would have required additional notation and definitions not included in this paper. A complete system is given in [Sack, 2007a][1]. But a subset of $\mathcal{PS}$ is complete with respect to a sublanguage of $\mathcal{L}^{PAL}_Y$. Let $\mathcal{L}^{\mathrm{PAL}}_{Y-C}$ be the result of removing all instances of the common knowledge operators $\Box^*_{\mathcal{B}}$ from the $\mathcal{L}^{\mathrm{PAL}}_Y$. Let $\mathcal{PS}_{-C}$ be the result of removing from $\mathcal{PS}$ any axiom or rule involving common knowledge operators.

Theorem 3.1. *The proof system $\mathcal{PS}_{-C}$ is weakly complete with respect to $\mathcal{L}^{\mathrm{PAL}}_{Y-C}$.*

This theorem follows from a minor adaptation to the completeness proof in [Sack, 2007a].

We now employ $\mathcal{PS}$ to prove the formula (1).

Proof of formula (1): Let

$$\psi \equiv (\Box_C\neg\mathsf{BkAs} \wedge [\neg(A\diamondsuit)!](\neg\Box_C\neg\mathsf{BkAs} \wedge [\neg(D\spadesuit)!](\Box_C\neg\mathsf{BkAs}))).$$

The following hold:

1. $\mathsf{ant} \rightarrow (\spadesuit, \spadesuit, \diamondsuit, \clubsuit)$
2. $\mathsf{ant} \rightarrow (\neg(A\diamondsuit) \wedge \neg(D\spadesuit))$ from 1
3. $\mathsf{ant} \rightarrow \mathsf{knowl}$
4. $\mathsf{ant} \rightarrow \mathsf{pos}$
5. $\mathsf{ant} \rightarrow \Box_C\Box^*_{\mathcal{A}}(\mathsf{knowl} \wedge \mathsf{pos} \wedge \mathsf{dist} \wedge \mathsf{deal})$
6. $\mathsf{ant} \rightarrow \Box_C\,\mathsf{knowl}$ from 5
7. $\mathsf{ant} \rightarrow \Box_C\,\mathsf{pos}$ from 5
8. $\mathsf{ant} \rightarrow \Box_C[(\spadesuit, \spadesuit, \diamondsuit, \clubsuit) \vee (\spadesuit, \clubsuit, \diamondsuit, \spadesuit) \vee (\clubsuit, \spadesuit, \diamondsuit, \spadesuit)]$ from 3
9. $\mathsf{ant} \rightarrow \Box_C[((\spadesuit, \spadesuit, \diamondsuit, \clubsuit) \rightarrow \bigwedge\{\diamondsuit_B p : p(B) = \spadesuit\})$
   $\wedge\, ((\spadesuit, \clubsuit, \diamondsuit, \spadesuit) \rightarrow \bigwedge\{\diamondsuit_B p : p(B) = \clubsuit\})$
   $\wedge\, ((\clubsuit, \spadesuit, \diamondsuit, \spadesuit) \rightarrow \bigwedge\{\diamondsuit_B p : p(B) = \spadesuit\})]$ from 7

[1] This complete system is for a class of temporal dynamic epistemic languages. Adjusting for notation, the TPAL introduced in this paper is a particular temporal dynamic epistemic language.

10. $\mathsf{ant} \to \Box_C \neg \mathsf{BkAs}$ from 8,9

11. $\mathsf{ant} \to \Diamond_C(\spadesuit, \clubsuit, \Diamond, \spadesuit)$ from 4

12. $\mathsf{ant} \to \Box_C[(\spadesuit, \clubsuit, \Diamond, \spadesuit) \to \Box_B(B\clubsuit)]$ from 6

13. $(B\clubsuit) \to (\neg(A\Diamond) \to [(\spadesuit, \clubsuit, \Diamond, \spadesuit) \vee (\spadesuit, \clubsuit, \spadesuit, \Diamond)])$

14. $\mathsf{ant} \to \Box_C[(\spadesuit, \clubsuit, \Diamond, \spadesuit) \to \Box_B(\neg(A\Diamond) \to [(\spadesuit, \clubsuit, \Diamond, \spadesuit) \vee (\spadesuit, \clubsuit, \spadesuit, \Diamond)])]$ from 12,13

15. $\mathsf{ant} \to \Diamond_C[(\spadesuit, \clubsuit, \Diamond, \spadesuit) \wedge \Box_B(\neg(A\Diamond) \to [(\spadesuit, \clubsuit, \Diamond, \spadesuit) \vee (\spadesuit, \clubsuit, \spadesuit, \Diamond)])]$ from 11,14

16. $\mathsf{ant} \to \Diamond_C[\neg(A\Diamond) \wedge \Box_B(\neg(A\Diamond) \to (A\spadesuit))]$ from 15

17. $\mathsf{ant} \to \neg\Box_C\neg[\neg(A\Diamond) \wedge (\Box_B[\neg(A\Diamond)!](A\spadesuit) \vee \Box_B[\neg(A\Diamond)!](A\clubsuit) \vee \Box_B[\neg(A\Diamond)!](A\Diamond))]$ from 16

18. $\mathsf{ant} \to \neg\Box_C[\neg(A\Diamond) \to (\neg[\neg(A\Diamond)!]\Box_B(A\spadesuit) \wedge \neg[\neg(A\Diamond)!]\Box_B(A\clubsuit) \wedge \neg[\neg(A\Diamond)!]\Box_B(A\Diamond))]$ from 17

19. $\mathsf{ant} \to \neg\Box_C([\neg(A\Diamond)!]\neg\Box_B(A\spadesuit) \wedge [\neg(A\Diamond)!]\neg\Box_B(A\clubsuit) \wedge [\neg(A\Diamond)!]\neg\Box_B(A\Diamond))$ from 18

20. $\mathsf{ant} \to \neg\Box_C[\neg(A\Diamond)!]\neg\mathsf{BkAs}$ from 19

21. $\mathsf{ant} \to [\neg(A\Diamond)!]\neg\Box_C\neg\mathsf{BkAs}$ from 2,20

22. $\mathsf{ant} \to \Box_C(\neg(A\Diamond) \wedge \neg(D\spadesuit) \to (\spadesuit, \spadesuit, \Diamond, \clubsuit))$ from 3

23. $\mathsf{ant} \to \Box_C[(\spadesuit, \spadesuit, \Diamond, \clubsuit) \to (\Diamond_B(\clubsuit, \spadesuit, \spadesuit, \Diamond) \wedge \Diamond_B(\spadesuit, \spadesuit, \clubsuit, \Diamond))]$ from 7

24. $\mathsf{ant} \to \Box_C[(\spadesuit, \spadesuit, \Diamond, \clubsuit) \to [\Diamond_B((\neg(A\Diamond) \wedge \neg(D\spadesuit)) \wedge (\clubsuit, \spadesuit, \spadesuit, \Diamond)) \wedge \Diamond_B((\neg(A\Diamond) \wedge \neg(D\spadesuit)) \wedge (\spadesuit, \spadesuit, \clubsuit, \Diamond))]]$ from 23

25. $\mathsf{ant} \to \Box_C[(\neg(A\Diamond) \wedge \neg(D\spadesuit)) \to [\Diamond_B((\neg(A\Diamond) \wedge \neg(D\spadesuit)) \wedge \neg(A\spadesuit)) \wedge \Diamond_B((\neg(A\Diamond) \wedge \neg(D\spadesuit)) \wedge \neg(A\clubsuit)) \wedge \Diamond_B((\neg(A\Diamond) \wedge \neg(D\spadesuit)) \wedge \neg(A\Diamond))]]$ from 22,24

26. $\mathsf{ant} \to \Box_C\neg[\neg(A\Diamond) \wedge \neg(D\spadesuit) \wedge [\Box_B((\neg(A\Diamond) \wedge \neg(D\spadesuit)) \to (A\spadesuit)) \vee \Box_B((\neg(A\Diamond) \wedge \neg(D\spadesuit)) \to (A\clubsuit)) \vee \Box_B((\neg(A\Diamond) \wedge \neg(D\spadesuit)) \to (A\Diamond))]]$ from 25

27. $\mathsf{ant} \to \Box_C\neg[\neg(A\Diamond) \wedge [\neg(A\Diamond)!]\neg(D\spadesuit) \wedge [\Box_B[\neg(A\Diamond)!][\neg(D\spadesuit)!](A\spadesuit) \vee \Box_B[\neg(A\Diamond)!][\neg(D\spadesuit)!](A\clubsuit) \vee \Box_B[\neg(A\Diamond)!][\neg(D\spadesuit)!](A\Diamond)]]$ from 26

28. $\mathsf{ant} \to \Box_C\neg[\neg(A\diamondsuit) \wedge ([\neg(A\diamondsuit)!]\neg(D\spadesuit)$
$\wedge ([\neg(A\diamondsuit)!]\Box_B[\neg(D\spadesuit)!](A\spadesuit)$
$\vee [\neg(A\diamondsuit)!]\Box_B[\neg(D\spadesuit)!](A\clubsuit)$
$\vee [\neg(A\diamondsuit)!]\Box_B[\neg(D\spadesuit)!](A\diamondsuit))))]$ from 27

29. $\mathsf{ant} \to \Box_C(\neg(A\diamondsuit) \to [\neg[\neg(A\diamondsuit)!](\neg(D\spadesuit) \wedge \Box_B[\neg(D\spadesuit)!](A\spadesuit))$
$\wedge \neg[\neg(A\diamondsuit)!](\neg(D\spadesuit) \wedge \Box_B[\neg(D\spadesuit)!](A\clubsuit))$
$\wedge \neg[\neg(A\diamondsuit)!](\neg(D\spadesuit) \wedge \Box_B[\neg(D\spadesuit)!](A\diamondsuit))])$ from 28

30. $\mathsf{ant} \to \Box_C(\neg(A\diamondsuit) \to (\neg[\neg(A\diamondsuit)!](\neg(D\spadesuit) \wedge [\neg(D\spadesuit)!]\Box_B(A\spadesuit))$
$\wedge \neg[\neg(A\diamondsuit)!](\neg(D\spadesuit) \wedge [\neg(D\spadesuit)!]\Box_B(A\clubsuit))$
$\wedge \neg[\neg(A\diamondsuit)!](\neg(D\spadesuit) \wedge [\neg(D\spadesuit)!]\Box_B(A\diamondsuit)))$ from 29

31. $\mathsf{ant} \to \Box_C([\neg(A\diamondsuit)!](\neg(D\spadesuit) \to \neg[\neg(D\spadesuit)!]\Box_B(A\spadesuit))$
$\wedge [\neg(A\diamondsuit)!](\neg(D\spadesuit) \to \neg[\neg(D\spadesuit)!]\Box_B(A\clubsuit))$
$\wedge [\neg(A\diamondsuit)!](\neg(D\spadesuit) \to \neg[\neg(D\spadesuit)!]\Box_B(A\diamondsuit)))$ from 30

32. $\mathsf{ant} \to \Box_C([\neg(A\diamondsuit)!][\neg(D\spadesuit)!]\neg\Box_B(A\spadesuit)$
$\wedge [\neg(A\diamondsuit)!][\neg(D\spadesuit)!]\neg\Box_B(A\clubsuit)$
$\wedge [\neg(A\diamondsuit)!][\neg(D\spadesuit)!]\neg\Box_B(A\diamondsuit))$ from 31

33. $\mathsf{ant} \to \Box_C[\neg(A\diamondsuit)!][\neg(D\spadesuit)!]\neg\mathsf{BkAs}$ from 32

34. $\mathsf{ant} \to [\neg(A\diamondsuit)!]\Box_C[\neg(D\spadesuit)!]\neg\mathsf{BkAs}$ from 2,33

35. $\mathsf{ant} \to [\neg(A\diamondsuit)!](\neg(D\spadesuit) \to \Box_C[\neg(D\spadesuit)!]\neg\mathsf{BkAs})$ from 34

36. $\mathsf{ant} \to [\neg(A\diamondsuit)!][\neg(D\spadesuit)!]\Box_C\neg\mathsf{BkAs}$ from 35

37. $\mathsf{ant} \to (\Box_C\neg\mathsf{BkAs} \wedge [\neg(A\diamondsuit)!](\neg\Box_C\neg\mathsf{BkAs} \wedge [\neg(D\spadesuit)!](\Box_C\neg\mathsf{BkAs})))$ from 10,21,36

38. $\mathsf{ant} \to [\neg(A\diamondsuit)!]Y_\diamondsuit\psi$ from 37

39. $\mathsf{ant} \to [\neg(A\diamondsuit)!](\neg(D\spadesuit) \to Y_\diamondsuit\psi)$ from 38

40. $\mathsf{ant} \to [\neg(A\diamondsuit)!][\neg(D\spadesuit)!]Y_\diamondsuit Y_\diamondsuit\psi$ from 39

$\dashv$

3.2 Sum and Product

The sum and product riddle has a number of forms. Different forms are discussed in [van Ditmarsch et al., 2007], and they focus on the following version.

> A says to S and P: I have chosen two integers x, y such that $1 \leq x \leq y$ and $x + y \leq 100$. In a moment, I will inform S only of $s = x + y$, and P only of $p = xy$. These announcements remain private. You are required to determine the pair (x, y).
> He acts as said. The following conversation now takes place:
>
> i. P says: "I do not know it."
>
> ii. S says: "I knew you didn't."
>
> iii. P says: "I now know it."
>
> iv. S says: "I now also know it."
>
> Determine the pair (x, y).

In [van Ditmarsch et al., 2007], they tried to express this conversation in public announcement logic[2], though they faced an immediate problem with the second line, which refers to the past. They thus found a slightly different conversation that does not refer to the past, but should result in the same (x, y) pair:

a. S says "I know you don't know it."

b. P says "Now I know it."

c. S says "Now I also know it."

While only the second conversation can be expressed using dynamic epistemic logic, both conversations can be expressed using the language $\mathcal{L}_Y^{\mathrm{PAL}}$, with the public announcement action signature from example 6.6. The set of atomic propositions will be the same as those in [van Ditmarsch et al., 2007]: propositions of the form $x = n$ or $y = n$ for $1 \leq n \leq 100$. We abbreviate $x = n$ by x_n and $y = n$ by y_n. We finally abbreviate $\bigvee_{1 \leq i,j \leq 100} \Box_A(x_i \wedge y_j)$ by $K_A\mathrm{pair}$, which can be read as "A knows the pair (x, y)". Although $\Box_A$ and K_A have similar English readings, we use the different notation to emphasize that K_A is part of an abbreviation of a more complicated formula. The first conversation can be expressed as

i. $\neg K_P\mathrm{pair}$

ii. $Y_\Diamond \Box_S \neg K_P\mathrm{pair}$

iii. $K_P\mathrm{pair}$

iv. $K_S\mathrm{pair}$,

while the second is expressed by

[2]They call the logic dynamic epistemic logic, though they only use public announcements

a. $\Box_S \neg K_P \text{pair}$

b. $K_P \text{pair}$

c. $K_S \text{pair}$.

3.3 Muddy Children

In [Yap, 2005], Yap showed how statements about the previous stage can be relevant in a "muddy children problem" involving a sequence of individual announcements by the children rather than simultaneous announcements. She showed how reasoning by agents can depend on the answer to the following type of question. If agent A later in the sequence claims to know φ, did A acquire this knowledge from the last action (announcement), or did A know φ before? This question can be answered if A can announce, not only that he knows φ, but also that he did not at the previous stage. This is expressed by $\Box_A \varphi \wedge Y_\Diamond \neg \Box_A \varphi$.

4 Future

We obtain a new language $\mathcal{L}^{\text{PAL}}_{YT}$ by adding a next-time operator $T_\Box$ to $\mathcal{L}^{\text{PAL}}_{Y}$. The syntax for $T_\Box$ is the same as for $Y_\Box$, that is, if φ is a sentence, then so is $T_\Box \varphi$. Abbreviations similar to those used for $Y_\Box$ apply. The semantics of $\mathcal{L}^{\text{PAL}}_{YT}$ depends on the language of epistemic logic $\mathcal{L}^{\text{EL}}$, which contains proposition letters, boolean connectives $\neg$ and $\wedge$ as well as epistemic operators $\Box_A$ for each agent. We define the semantics of the next-time operator by

- $(H, s) \in [\![T_\Box \varphi]\!]$ iff for all ψ in $\mathcal{L}^{\text{EL}}$, $(H, s) \in [\![[\psi]\varphi]\!]$.

By removing the previous-time operator $Y_\Box$ and common knowledge operators $\Box^*_{\mathcal{B}}$ from $\mathcal{L}^{\text{PAL}}_{YT}$ we obtain the language $\mathcal{L}^{\text{PAL}}_{T-C}$. A complete proof system for $\mathcal{L}^{\text{PAL}}_{T-C}$ restricted to the case where all epistemic relations are equivalence relations is given in [Balbiani et al., 2007].

4.1 Changeability of Facts and Causes of Belief Change

In the language $\mathcal{L}^{\text{PAL}}_{YT}$, for each n, the formula $\Box_A \varphi \rightarrow T^n_\Box \Box_A Y^n_\Diamond \varphi$ is valid. In other words, if A believes φ, then for ever after, she will believe φ was the case. The validity of these formulas reflects the fact that belief revision does not take place in PAL. However, agents can believe certain facts can change, such as what other agents know. It is beliefs about a particular point in time that never change.

There are some φ that when true, are always true, that is for which $\varphi \rightarrow T_\Box \varphi$ is valid. The simplest example is an atomic propositions p, which has a truth value that does not change under submodels. Once an agent believes such a formula, the agent will always believe the formula. Thus the set of formulas for which $\varphi \rightarrow T_\Box \varphi$ is valid

is closed under epistemic operators $\Box_A$. It turns out that such formulas are also closed under conjunction and common knowledge $\Box^*_{\mathcal{B}}$.

We now explore facts that can become false. The opening example in the introduction demonstrates how beliefs about the present (not the past) may be retracted in anticipation of facts changing. At first Cathy knew Bob did not know what suit Ann had. But after some information was revealed, Cathy was aware of the possibility that the fact that Bob did not know could have changed. She thus retracted her belief. But note that, as in the previous example, she still believes that Bob did not know Ann's suit before the announcement was made. So we should not interpret the word "retract" too strongly. We simply mean that Cathy is no longer sure that Bob's uncertainty is still true.

At this point, before the second announcement, and after the first, among the formulas that are true are the following three:

$T_\Diamond \neg \Box_C \Box_B$Ann's

$T_\Diamond \Box_C \Box_B$Ann's

$T_\Diamond \Box_C \neg \Box_B$Ann's.

The first states the possibility that no significant information is revealed by the next stage, and Cathy's uncertainty remains. The second states the possibility of the situation given in the opening example, where more information provides Cathy with certainty that Bob does not know Ann's suit. The third states the possibility that Cathy knows Bob does know Ann's suit. This may seem at odds with the second, especially when we notice that in this particular example $\Box_C \psi$ means Cathy knows (correctly) that ψ is true. But we realize that $T_\Diamond \Box_B$Ann's and $T_\Diamond \neg \Box_B$Ann's are both true. Ann could announce her suit to everyone and thus $\Box_B$Ann's would be true after her announcement, and Cathy would know Bob's certainty too.

Suppose that indeed Ann announces her suit to everyone. Now everyone knows Ann's suit (including Bob), and no true statement given in the future can change this knowledge. In particular, Cathy will always know that Bob knows Ann's suit, and we will express this as follows. For every n, $T^n_\Box \Box_C \Box_B$Ann's.

4.1.1 Memory and Learning

We have seen that once an agent knows (or believes) φ, then that agent will always know (or believe) that φ was true. This tells us that in some sense, an agent never retracts or forgets what she knew or believed about a feature of a particular time. In the opening example, Cathy retracting her belief about Bob's uncertainty does not mean that she no longer believes Bob was uncertain at the first stage, but rather that she considers it possible that Bob's uncertainty may have changed. When after the second announcement, Cathy believes again that Bob is uncertain, she is not recalling a fact she knew before, but learning that it applies to the current situation. In this example, we can view each change in epistemic state a result of learning, and never forgetting.

4.2 Fitch's Paradox

An application of $\mathcal{L}^{\mathrm{PAL}}_{T-C}$ is Fitch's paradox, and has been discussed in the papers [van Benthem, 2007] and [Balbiani et al., 2007]. Fitch addressed the question: if φ is true now, can it be known later that φ is true at that later time? Fitch's paradox [Brogaard and Salerno, 2002] showed how this is not necessarily true. If φ were $p \wedge \neg \Box_A p$, agent A could never know φ to be true, for this would imply that A knows p but also knows she doesn't know p, which by Moore's paradox cannot be true.

Using the operator $T_\Box$, we can ask, for what φ is $\varphi \rightarrow T_\Diamond \Box_A \varphi$ satisfiable? A well known case, where φ cannot become known by an agent A, is where $\Box_A \varphi$ is inconsistent. Examples of this involve Moore's paradox, where φ may have the form $p \wedge \neg \Box_A p$. But there are also cases where $\Box_A \varphi$ is consistent, φ is true, but $\Box_A \varphi$ cannot become true. An example of this using dynamic epistemic logic is given in [van Benthem, 2007] where the formula is $\varphi = (p \wedge \Diamond_A \neg p) \vee \Box_A \neg p$, and the model has two points, one with p true the other with $\neg p$ true, and with A not being able to distinguish between these two points. In the state where p is true, we can check that φ is true and that there is no submodel in which $\Box_A \varphi$ is true. Hence there is no public announcement that can result in A knowing φ. But a more positive result is shown in [van Benthem, 2007] giving us that such learnability is decidable given that there is one agent, whose epistemic relation is an equivalence relation.

5 Games

5.1 Games and Epistemic Logic

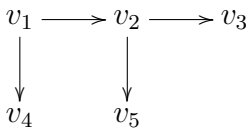
We first define a game as follows:

Definition 5.1 (Game). A game is defined as a tuple: $(V, R, \mathcal{A}, \iota, \{u_A : A \in \mathcal{A}\})$, where

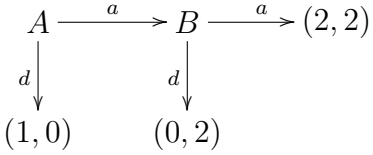
1. V is a set (of elements called vertices).
2. R is a relation on V satisfying the following properties that make it a tree in the following way:
   (a) There is a unique element $r \in V$ (called the root) such that if R^* is the reflexive transitive closure of R, then for all $v \in V$, rR^*v.
   (b) For every $v \in V - \{r\}$, there is a unique $v' \in V$, such that $v'Rv$.
   (c) If R^+ is the transitive closure of R, then for all v, it is not the case that vR^+v.
3. ι is a function mapping every not-terminal vertex (a v such that there exists a w for which vRw) to a player in $\mathcal{A}$. At vertex v, we consider it $\iota(v)$'s turn.
4. u_A is an injective function (called a utility function) mapping every terminal vertex (a v for which there is no w with vRw) to a real number.

We shall be particularly concerned with finite games (games for which the set V is finite).

Example 5.2. Suppose V consists of 5 vertices v_1, v_2, v_3, v_4, and v_5. The following diagram depicts the relation R.



As v_1 and v_2 are non-terminal vertices, ι will assign each to a player. We will involve two players: A for Ann and B for Bob. Similarly v_3, v_4, and v_5 will each be mapped to a payoff number for each player. We obtain the next diagram by replacing the vertex v_1 by $\iota(v_1)$ and v_2 similarly, and by replacing v_3 by $(u_A(v_3), u_B(v_3))$, and v_4 and v_5 similarly. We also label the relational connections by a specific action a for across, and d for down. These actions are not in our definition of a game, but may make it easier for us to analyze this game.[3]



We next define a strategy profile, which shall be a function from every non-terminal vertex to a successor vertex. Let D be the set of non-terminal vertices in V.

Definition 5.3 (Strategy Profile). A function $p : D \to V$ is a strategy profile if for all $v \in D$, $vRp(v)$.

In example 5.2, there are 4 strategy profiles:

- $v_1 \mapsto v_2, v_2 \mapsto v_3$, which we shall abbreviate aa
- $v_1 \mapsto v_2, v_2 \mapsto v_5$, which we shall abbreviate ad
- $v_1 \mapsto v_4, v_2 \mapsto v_3$, which we shall abbreviate da
- $v_1 \mapsto v_4, v_2 \mapsto v_5$, which we shall abbreviate dd

Let D be a subset of the set of vertices V consisting of vertices v for which there exists a w such that vRw. In other words D is the set of non-terminal vertices. of V. We can partition the set of non-terminal vertices D into a set D_A for each player such that $D_A = \iota^{-1}(\{A\})$ is the pre-image of $\{A\}$ under ι. Given a strategy profile p we define A's strategy to be $p_A = p|D_A$, the restriction of p to the domain D_A. Also given a strategy profile p and a decision vertex $v \in D$, we define the action to be $p_v = p|\{v\}$. We can represent a strategy s_A and action s_v in epistemic logic (as well as PAL or any language containing epistemic logic) using the abbreviations

[3]Our definition of a game can represent actions by specifying the next vertex to be reached. We use a and d instead in this example, for we hope that they will be more clear.

- $\mathsf{strat}[p, A] \equiv \bigvee\{t \in S : t | D_A = s_A\}$
- $\mathsf{action}[p, v] \equiv \bigvee\{t \in S : t | \{v\} = s_v\}$

Throughout a game, we may often assume that each player knows his/her own strategy, and does not know anyone else's strategy. State models have been used to capture the understanding players have of the strategy profile at the beginning of a game, and at different stages throughout the game. Aumann, Stalnaker, and Halpern are among those who have used state models as such. While state models focus on strategy profiles, they ignore the payoffs the agents will receive. Given a game, we can place restrictions on a state model, so that it can more realistically capture the situation of a game. Here are some restrictions we may want to place on a state model.

Definition 5.4 (Game model). Given a game Γ, a state model $\mathbf{S}$ is a *game model* if the following conditions are satisfied.

1. Each epistemic relation is serial, that is, for each A and x there is a z such that $x \xrightarrow{A} z$
2. Each epistemic relation is transitive, that is, for each A, x, y, and z, if $x \xrightarrow{A} y$ and $y \xrightarrow{A} z$, then $x \xrightarrow{A} z$.
3. Each epistemic relation is Euclidean, that is, for each A, x, y, and z, if $x \xrightarrow{A} y$, and $x \xrightarrow{A} z$, then $y \xrightarrow{A} z$.
4. For every $s \in S$, there is exactly one $p \in \Phi$ for which $s \in \|p\|_{\mathbf{S}}$.
5. If $s, t \in S_0$, $s \in \|p\|$, $t \in \|q\|$, and $s \xrightarrow{A} t$, then $p_A = q_A$ (that is A uses the same strategy in both s and t).

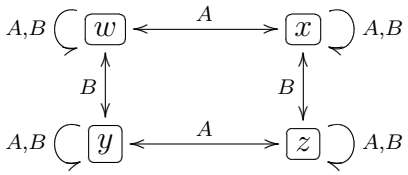
Conditions 1, 2, and 3 help capture properties many find desirable when describing belief. Seriality ensures that an agent considers something possible, and hence does not believe everything including $p \wedge \neg p$. Conditions 2 and 3 correspond to positive and negative introspection: if an agent believes something, she knows she believes it, and if an agent does not believe something, she then knows she does not believe it. Condition 4 ensures that at each possible world, there is exactly one strategy profile. Condition 5 ensures that each player knows his/her own strategy.

Given the game in example 5.2, the following state model satisfies the all 5 conditions.

Example 5.5. We let the set of atomic propositions Φ be the set of all four strategy profiles. There shall be four states, each assigned exactly one strategy profile. Let $\mathbf{S}_0 = (S, \{\xrightarrow{A}, \xrightarrow{B}\}, \|\cdot\|)$, defined by

1. $S = \{w, x, y, z\}$

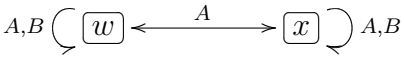
2. $\xrightarrow{A}$ and $\xrightarrow{B}$ are equivalence relations given by the diagram below



3. $\|\cdot\|$ maps $aa \mapsto \{w\}$, $ad \mapsto \{x\}$, $da \mapsto \{y\}$, and $dd \mapsto \{z\}$.

5.2 Games and Temporal Public Announcement Logic

Let $\mathbf{S}_0$ be the state model given in example 5.5. Suppose that in the game, A decides to go across, and that her action is revealed to everyone (that is to B as well as herself). We can treat this as a public announcement that the actual state is either w or x, and hence the updated model $\mathbf{S}_1$ shall have the following relational structure:



In this way, public announcements can capture the play of an extensive game. We can record the stages of the game through the first move in the tpal-history $((\mathbf{S}_0), \mathbf{S}_1)$.

Given a set of coherent history/state pairs and a history H, let $H \otimes X \equiv (H, \mathbf{S})$, where $\mathbf{S}$ is the restriction of $\widehat{\mathbf{S}}(\widehat{P}(H))$ to the set X. Now given a history, can we determine whether it is the history of a given game Γ? Let Γ be a game.

Definition 5.6 (Game History for Γ). Let r be the root of the game Γ, H be a tpal-history, let n be such that $\widehat{P}^n(H) \neq \emptyset$ but $\widehat{P}^{n+1}(H) = \emptyset$. Let $\mathbf{S}_0 = \widehat{\mathbf{S}}(\widehat{P}^n(H))$ be the initial state model in the history. Then H is a *game history* iff it satisfies the following properties.

(a) $\mathbf{S}_0$ is a game model.

(b) There is a strategy profile $p \in \Phi$ such that $\|p\|_{\mathbf{S}_0} \neq \emptyset$ and for each $0 < k \leq n$, $\widehat{P}^{k-1}(H) = \widehat{P}^k(H) \otimes [\![\mathsf{action}[p, p^{n-k}(r)]]\!]$.

We have defined validity of a formula φ by $[\![\varphi]\!] = [\![\,\mathsf{true}]\!]$, the set of all coherent history/state pairs. We now define the notion of validity within a history. We will distinguish between two kinds: weak and strong.

Definition 5.7 (Strong and Weak validity). Let H be a history, and let n be its age ($\widehat{P}^n(H) \neq \emptyset$ but $\widehat{P}^{n+1}(H) = \emptyset$). Then φ is *weakly valid* in H iff $(H, s) \in [\![\varphi]\!]$ for each $s \in H$ (recall our convention that $s \in H$ means $s \in \widehat{\mathbf{S}}(H)$). φ is *strongly valid* in H iff for each $k \leq n$ and each $s \in \widehat{P}^k(H)$, $(\widehat{P}^n(H), s) \in [\![\varphi]\!]$.

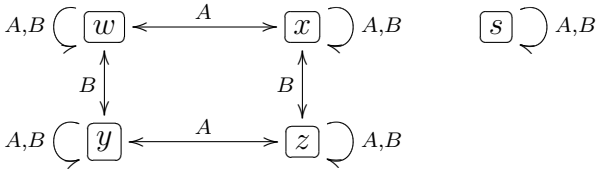
Now given a game history H, the following are strongly valid:

1. $Y_{\Box}\,\mathsf{false} \to \Diamond_A\,\mathsf{true}$
2. $\Box_A\varphi \to \Box_A\Box_A\varphi$
3. $\Diamond_A\varphi \to \Box_A\Diamond_A\varphi$
4. $\bigvee \Phi$
5. $p \to \neg q$ for all $p, q \in \Phi$ and $p \neq q$
6. $\mathsf{strat}[p, A] \to \Box_A\mathsf{strat}[p, A]$ for each $A \in \mathcal{A}$ and $p \in \Phi$.

Y $p \wedge Y^k_{\Diamond}\,\mathsf{true} \wedge Y^{k+1}_{\Box}\,\mathsf{false} \to (\varphi \leftrightarrow Y_{\Box}[\mathsf{action}[p, p^k(r)]!]\varphi)$ for each formula φ, $k \geq 1$, where k is less than the depth of V and $p \in \Phi$ (and with $r \in \Phi$ the root of the game tree).

The first collection of formulas corresponds to the initial model being serial. Strong validity results in the fact that the initial state model is a game model. It is well known in epistemic logic that 2 and 3 are valid in transitive and Euclidean models respectively. Public announcements preserve transitive and Euclidean models, and hence 2 and 3 are strongly valid. Both 4 and 5 correspond to the fact that a game model assigns exactly one atomic proposition (viewed as a strategy profile) to each state. Public announcements preserve the atomic proposition assignment, and hence 4 and 5 are strongly valid in a game history. The collection of formulas relating to 6 corresponds to the fact that each player knows his/her own strategy. This also is preserved through public announcements, and hence 6 is strongly valid. Finally the formulas of Y are strongly valid, because every public announcement in a game history is an announcement of an atomic proposition (strategy profile).

But it is not the case that a history for which 1,2,3,4,5,6, and Y are strongly valid is a game history. There could be history, for example, where the first update is a public announcement of $\mathsf{action}[p, v] \wedge \psi$, and the only states not satisfying ψ are unreachable from states satisfying ψ. Suppose Γ were the game given in example 5.2 and $\mathbf{S}_0$ were given by the following relational structure:



where $\|\cdot\|$ maps $aa \mapsto \{w, s\}$, $ad \mapsto \{x\}$, $da \mapsto \{y\}$, and $dd \mapsto \{z\}$. Suppose that $\mathbf{S}_1 = \mathbf{S}_0 \otimes [\![(aa \vee ab) \wedge \Diamond_A ab]\!]$. Then $((\mathbf{S}_0), \mathbf{S}_1)$ does not satisfy condition (b) of a game history. But 1,2,3,4,5,6, and Y are strongly valid.

This last counterexample could have been prevented if we strengthen the definition of a game history to require that in the initial state model, every agent considers possible every strategy profile for which she uses her strategy. It is yet unknown whether there

could be another counterexample. But we may find this too much of a restriction. For example, there may be a game in which a certain strategy for B would consistently yield a lower payoff than any other strategy regardless of the strategies of other players. Then A might from the very start not consider such a strategy a possibility.

5.3 Inverse programs and Hybrid Logic

So far TPAL only allows us to express actions in relationship with their consequences, and cannot assert that an action actually occurred. In order to characterize a game history, it would help to be able to assert that a certain action took place. In this subsection, we explore two extensions of $\mathcal{L}_Y$ that can help us achieve this ability. The first is to add inverse programs, and the second is to add nominals. Completeness and decidability of these two extensions have not yet been explored.

5.3.1 Inverse programs

Let us add to $\mathcal{L}_Y$ programs of the form π^{-1} for every program π. For example $\langle (p!)^{-1} \rangle \mathsf{true}$ is a well-formed formula with the program $(p!)^{-1}$, and asserts that either p or an equivalent expression was just announced. The semantics of π^{-1} shall be as follows:

- $[\![\pi^{-1}]\!] = [\![\pi]\!]^{-1} = \{(b,a) : (a,b) \in [\![\pi]\!]\}$ the inverse of the relation $[\![\pi]\!]$.

Let us replace the set of formulas Y that are strongly valid in all game histories for a given Γ with the following

(I) $p \wedge Y^k_\Diamond \mathsf{true} \wedge Y^{k+1}_\Box \mathsf{false} \rightarrow [(\mathsf{action}[p, p^{k+1}(r)]!)^{-1}]$ for each $k \geq 0$ where k is less than the depth of V, and $p \in \Phi$ (and with $r \in \Phi$ the root of the game tree.)

Then 1,2,3,4,5,6, and I are strongly valid in all game histories for Γ.

Given a history H for which $\widehat{\mathbf{S}}(H)$ is non-empty, if 1,2,3,4,5,6, and I are strongly valid in H, then H is a game history. The sets of formulas 1,2,3 ensure that the initial state model is serial, transitive, and Euclidean. The formulas 4 and 5 ensure that exactly one strategy profile is assigned to each state. Condition 6 ensures that each agent knows his/her own strategy in the initial state model. Note that transitivity, the Euclidean property, the assignment of atomic propositions, as well as the fact that every agent knows his/her own strategy is preserved under any public announcement. Finally the formulas of I ensure that every update is the result of announcing an action. To this, pick any state $s \in H$ (without loss of generality assume s is in the most recent state model). Let p be the strategy profile true in s. The claim is that p qualifies as such a strategy profile (and any other strategy profile true in the most recent state model shall have the same initial play of the game). We can see this by observing that $[(\mathsf{action}[p, p^{j-1}(r)]!)^{-1}]$ is true at $\widehat{p}^k(s)$, where $j+k$ is the number of models in the history. This means that the action at $p^{j-1}(r)$ was revealed in getting from the $j-1^{\text{st}}$ state model to the j^{th} state model.

5.3.2 Hybrid logic

As an alternative to inverse programs, we may use Hybrid logic. Hybrid logic allows us to make reference to a particular state, either in the current history or in a past or future history. It also allows us to maintain the local perspective of modal logic. Several hybrid languages are introduces in [Blackburn and Seligman, 1995]. These languages each involve a set with elements called nominals, where each nominal names a particular state in a given Kripke model. Nominals in TPAL name coherent history/state pairs. The reason for making the histories variable as well as states is that TDEL semantics is dynamic: the truth of a formula involving past or future at a state in a history depends on the truth of another formula at a state in a different history. Let Ω be a set of nominals, and let ν be a function assigning nominals to history/state pairs. For a given language, the set Ω will be fixed as will the set Φ of proposition letters. The semantics of languages involving nominals specifies truth of formulas in coherent history/state/assignment triples.

Let us add to $\mathcal{L}_Y$ sentences of the form i, where $i \in \Omega$, and for each sentence φ and nominal i, sentences of the form $\downarrow i.\varphi$. The nominals i syntactically behave the same way as atomic propositions do. Sentences of the form i shall assert that the current state (represented by the state in the triple) is named i (determined by the assignment in the triple). Sentences of the form $\downarrow i.\varphi$ assert that φ would be true if the current state were named i.

The semantics of programs involve a function $[\![\cdot]\!]_n$ that maps programs to relations over coherent history/state/assignment triples. We then define $(H, s, \nu)[\![\pi]\!]_n(H', s', \nu')$ the same way we did for $[\![\pi]\!]$, except here we add the condition that $\nu = \nu'$. We make sure actions preserve the nominal assignment since nominals are assigned to coherent history/state pairs, and hence what history the nominal refers to is also important.

The semantics of sentences involve a function $[\![\cdot]\!]_n$ that maps sentences to sets of coherent history/state/assignment triples. We define the semantics of some key sentences:

- $[\![\,\mathsf{true}]\!]_n$ is the set of all coherent history/state/assignoment triples.
- $[\![p]\!]_n = \{(H, s, \nu) : s \in \|p\|_{\widehat{\mathbf{S}}(H)}\}$, for each atomic sentence $p \in \Phi$.
- $[\![i]\!]_n = \{(H, s, \nu) : s = \nu(i)\}$, for each nominal $i \in \Omega$.
- $[\![\downarrow i.\varphi]\!]_n = \{(H, s, \nu) : (H, s, \nu') \in [\![\varphi]\!]_n$ whenever $\nu'(j) = \nu(j)$ for all $j \neq i$ and $\nu'(i) = (H, s)\}$.
- $[\![\Box_A\varphi]\!]_n = \{(H, s, \nu) : (H', s', \nu') \in [\![\varphi]\!]_n$ whenever $H = H'$, $s \xrightarrow{A}_{\widehat{\mathbf{S}}(H)} s'$ and $\nu = \nu'\}$,
- $[\![[\pi]\varphi]\!]_n = \{(H, s, \nu) : (H', s', \nu') \in [\![\varphi]\!]_n$ whenever $(H, s, \nu)[\![\pi]\!]_n(H', s', \nu')\}$,
- $[\![Y_\Box\varphi]\!]_n = \{(H, s, \nu) :$ if $\widehat{P}(H) \neq \emptyset$, then $(\widehat{P}(H), \widehat{p}(s), \nu) \in [\![\varphi]\!]_n\}$.

The remaining cases are defined similarly to the way the semantics for $\mathcal{L}_Y$ is defined.

We could have replaced formulas I with the following.

(N) $p \wedge Y_{\Diamond}^{k} \mathsf{true} \wedge Y_{\Box}^{k+1} \mathsf{false} \rightarrow\downarrow i.Y_{\Box}[\mathsf{action}[p, p^{k+1}(r)]!]i$ for each $k \geq 0$ where k is less than the depth of V, and $p \in \Phi$ (and with $r \in \Phi$ the root of the game tree).

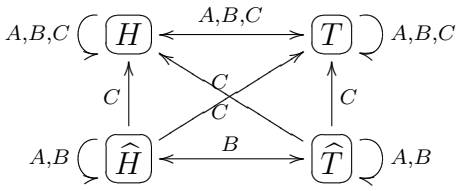
6 Temporal Dynamic Epistemic Logic

This section explores the addition of temporal operators to dynamic epistemic logic. Dynamic epistemic logic (DEL) allows us to reason about more complicated epistemic actions than just public announcements. Such actions include private announcements. We will discuss the basic definitions behind DEL, and then introduce TDEL histories. We will give many examples of actions and TDEL-histories.

6.1 Dynamic Epistemic Logic Structures

Example 2.2 presented a state model depicting the situation of three agents, A, B and C, who have common knowledge that none of them knows whether a coin landed with heads facing up or tails facing up. Let us continue this example with the following situation:

Example 6.1 (Peeking). Now suppose A peeks at the coin, B does not see the coin, but B notices A peeking, and C does not think anything happened at all. Neither A nor B suspect C of observing A peek, but A knows B saw her peek. To address the result of this situation, we define a state model with states $\{H, T, \widehat{H}, \widehat{T}\}$. Because C does not think anything happened, she must imagine the situation of the last example, where it was common knowledge that no one knew whether the coin was heads or tails. The states H and T shall represent the states in the previous example, which C now imagines to actually be the case. The actual situation, which C is not aware of, will be either the state $\widehat{H}$ or $\widehat{T}$ (depending on what was actually flipped). There shall be no C-arrow to either of these new states, in order to reflect C's misconception. The epistemic relations will be given according to the following diagram:



The valuation is defined by $\|h\| = \{H, \widehat{H}\}$ and $\|t\| = \{T, \widehat{T}\}$. From $\widehat{H}$, agent A considers only the state $\widehat{H}$ possible, reflecting that A knows the entire situation without any uncertainty. From $\widehat{T}$, the situation is similar. From $\widehat{H}$, as B knows A peeked, B considers only $\widehat{H}$ or $\widehat{T}$ possible, but does not know whether the coin turned up heads or tails.

The transition between the models in examples 2.2 and 6.1 is precisely the type of dynamic situation DEL is concerned with. Dynamic epistemic logic expresses epistemic consequences of epistemic actions. Epistemic consequences represent new beliefs by the

agents regarding properties of the world (among which are the atomic propositions in Φ) and the beliefs other agents have about these, and are represented as state models as in example 6.1. The transition from one state model to the next induced by some epistemic action is formalized by a function called the update product, which gives a new state model upon two inputs: a (previous) state model and an epistemic action. We shall first give more explanation to what an epistemic action is and then define the update product. Epistemic actions are actions that involve the receipt of information, such as public and private announcements. In example 6.1, the simultaneous events of A peeking, B observing the peeking but not knowing the content of what was learned, and C not observing anything at all, can all be considered one single epistemic action, which together with the original concealed coin model induces the model in the second example. We do not assume that an epistemic action can always occur. For example, suppose Ann draws a card from a deck. A truthful announcement that Ann has the 3♡ is an epistemic action. In the situation that Ann does not have the 3♡, this action cannot occur. We call conditions for epistemic action to be made "preconditions". In epistemic logic and DEL, truth of statements such as "Ann has the 3♡" are evaluated in coherent model/state pairs:

Definition 6.2 (Coherent model/state pair). A *coherent model/state pair* is a pair $(\mathcal{S}, s)$ for which s is a state of model $\mathbf{S}$.

We formally treat preconditions as a specification as to where an epistemic action can take place, that is where a certain statement or condition is true.

Definition 6.3 (Precondition). A *precondition* is a collection of coherent model/state pairs.

In [Baltag and Moss, 2004] and [Baltag et al., 2003], an epistemic action is defined as a structure called an action model. Each action model consists of

1. a finite set whose elements are called action types[4].

2. for each agent, a binary (epistemic) relation over the set of action types.

3. a precondition X_σ associated with each action type $\sigma \in \Sigma$.

In [Baltag and Moss, 2004] and [Baltag et al., 2003], the update product produces a new state model given an old state model and an action model. Any DEL or TDEL language

[4]In the absence of preconditions, these action types can be viewed as types of actions. Such a type of action could be a public announcement of something. The precondition will provide the content of the announcement. In [Baltag and Moss, 2004] these elements are called "actions" when preconditions are present and "action types" when preconditions are absent. In this paper, we use the term "action type" in both situations, so that the term we use to describe these states does not depend on context. We will use the term "action" more informally, and will typically use it to refer to all components of an action model not understood by context.

and semantics will assume the first two of the three components of an action model to be fixed. In addition, these first two parts will be fixed for a particular proof system, and hence we call this underlying relational structure an *action signature*.[5] We now give the action signature a more formal treatment.

Definition 6.4 (Action signature). Given a set $\mathcal{A}$ of agents, an *action signature* is a pair $\mathbf{\Sigma} = (\Sigma, \xrightarrow{\mathcal{A}}_{\mathbf{\Sigma}})$, such that

1. $\Sigma = \{\sigma_1, \ldots, \sigma_n\}$ is a finite set, where each σ_i is called an *action type*. It may be helpful to assume a fixed enumeration of Σ.

2. $\xrightarrow{\mathcal{A}}_{\mathbf{\Sigma}}$ is a set of relations $\xrightarrow{A}_{\mathbf{\Sigma}}$ over Σ for each $A \in \mathcal{A}$. When $(\sigma, \tau) \in \xrightarrow{A}_{\mathbf{\Sigma}}$, we write $\sigma \xrightarrow{A}_{\mathbf{\Sigma}} \tau$, which can be read as "$A$ considers τ a possible occurrence if σ actually happens". Due to the epistemic nature of this relation, this relation can also be called an *epistemic relation*.

It may be helpful to compare the reading of $\xrightarrow{A}_{\mathbf{\Sigma}}$ in an action signature and $\xrightarrow{A}_{\mathbf{S}}$ in a state model. They are both epistemic relations. If we change in the reading of $\xrightarrow{A}_{\mathbf{\Sigma}}$ the word "occurrence" to "world" and the word "happens" to "is the case", we would obtain a suitable reading for $\xrightarrow{A}_{\mathbf{S}}$. In general, notation and context will make it clear which relation we mean, and we often omit the subscripts $\mathbf{\Sigma}$ and $\mathbf{S}$.

Note that if $s \xrightarrow{A} t$, A might not consider t possible. Similarly, $\sigma \xrightarrow{A} \tau$, A might not consider an occurrence of σ. If we wanted it to be the case that A always considers possible the actual state or the action type that actually occurred, we would make A's epistemic relation reflexive in the state model or action model respectively. In general when A's epistemic relation is reflexive, A can only believe something if it is true, and hence we call the belief knowledge.

It will be significant in this paper that the set Σ be finite, as we will be considering finitary formulas that include every $\sigma \in \Sigma$. The name "action type" reflects the fact that the action signature together with a specified element only provides the kind of action involved, rather than the specific action; a specific action would also require preconditions.

As we will see, the formal definition of the update product will use all three components of an action model. In this paper, unlike in [Baltag et al., 2003], the action signature shall be encoded in the definition of the update product, for the list of preconditions is the only variable component of the action model when a specific semantics is given. Doing so will slightly simplify notation. When mentioning the inputs of the update product, we generally say "a list of preconditions" instead of "an epistemic action".

Definition 6.5 (Update Product). Given an action signature $\mathbf{\Sigma} = (\Sigma, \xrightarrow{\mathcal{A}}_{\mathbf{\Sigma}})$, where $\Sigma = \{\sigma_1, \ldots, \sigma_n\}$ is given a fixed enumeration, we define the update product as a function

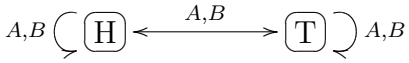
[5] This is exactly how an action signature is defined in [Baltag et al., 2003], but action signatures in [Baltag and Moss, 2004] include an additional component.

$\otimes$ that takes a state model $\mathbf{S} = (S, \xrightarrow{\mathcal{A}}_{\mathbf{S}}, \|\cdot\|_{\mathbf{S}})$ and a sequence $\vec{X} = (X_1, \ldots, X_n)$ of preconditions, and produces a new state model $\mathbf{S} \otimes \vec{X} = (S \otimes \vec{X}, \xrightarrow{\mathcal{A}}_{\mathbf{S}\otimes\vec{X}}, \|\cdot\|_{\mathbf{S}\otimes\vec{X}})$, where

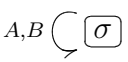
1. $S \otimes \vec{X} = \{(s, \sigma_i) \in S \times \Sigma : (\mathbf{S}, s) \in X_i\}$
2. $(s, \sigma_i) \xrightarrow{A}_{\mathbf{S}\otimes\vec{X}} (t, \sigma_j)$ iff both $s \xrightarrow{A}_{\mathbf{S}} t$ and $\sigma_i \xrightarrow{A}_{\mathbf{\Sigma}} \sigma_j$ (Note that $(s, \sigma_i), (t, \sigma_j) \in S \otimes \vec{X}$.)
3. $\|p\|_{\mathbf{S}\otimes\vec{X}} = \{(s, \sigma_i) \in S \otimes \vec{X} : s \in \|p\|_{\mathbf{S}}\}$

We now see that public announcements can be captured by an action signature

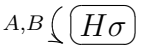
Example 6.6 (Common Discovery using Public Annonucement Action Signature). Suppose the set of agents is $\mathcal{A} = \{A, B\}$. The state model $\mathbf{S}$ being updated will correspond to the following situation. Suppose two agents enter a room and know a coin is in a box with either heads or tails facing up, but do not know which is the case. We model this by using a two agent version of the state model given in example 2.2, and we depict it by:



We consider an action signature with $\Sigma = \{\sigma\}$, and reflexive relations for both A and B as shown in the following graph.



Suppose both agents look into the box and find that coin is actually heads. We would expect the updated model to represent common knowledge of everyone's certainty that the coin had heads facing up. We shall represent this discovery of heads using the precondition X_σ consisting of the set of all pairs $(\mathbf{S}, s)$ for which $s \in \|h\|$. The updated model is represented by



Here $H\sigma$ corresponds to the pair (H, σ).

The action in the example was a common discovery, which has a similarity to a truthful public announcement in that the content of the announcement is a common discovery by the agents. A key difference between a discovery and an announcement is the source of information. But dynamic epistemic logic does not specify the source of information in any epistemic action, though it can reveal an agent's knowledge of the information, thus suggesting that the agent could have been the source of the information. For this reason we blur this distinction between a discovery and an announcement. We shall see in examples 6.10 and 6.11 that there are action signatures for which discoveries may be common among only a subset of the agents (or viewed as announcements, it is possible that announcements can be restricted to a smaller set of recipients).

Our view of the action type in the public announcement action signature can change when we are provided with a precondition. In general, we would interpret the precondition as the content of the announcement. But if the precondition were the empty set or consisted of all coherent model state pairs we may have a different interpretation. If the precondition were the empty set, the action cannot be made from any state, and the updated model would be the empty model. We might view this as a truthful announcement of a contradiction, something that cannot be done. But if the precondition were the set of all pairs, we might view the action as an announcement of a tautology. Epistemic logicians may see significance in the discovery of a tautology, but DEL already assumes the agents have logical omniscience. Any state in the former model satisfies the precondition, and hence the updated model is structurally identical to the former. It is as if nothing happened at all. More complicated action signatures, may contain a state whose generated submodel is the public announcement action signature. It will often be helpful to view such an action type together with a precondition consisting of all coherent pairs as just an uneventful passage of time.

With the public announcement action signature, all successful announcements are truthful, and each announcement results in common knowledge among the agents that the content of the announcement was true just before the announcement was made. Any state in the updated model must have a first coordinate (previous state) that satisfied the precondition, and thus the precondition was true in this previous state. For each agent, all epistemic arrows point to a state for which the precondition had previously been true, and hence everyone believes the content of this announcement was true. Also all epistemic arrows point to a state for which all agents believe that the precondition was true. Iterating this reasoning, we see that it is common knowledge that the content of the announcement was true. Although all announcements are truthful using the public announcement action signature, there are other action signatures, for which this need not be so, and we will see in example 6.9 an action signature for which information might not be truthful.

Update products using the public announcement action signature have a particular simplicity, in that the updated model is effectively a submodel of the state model being updated. Only the names of the states prevent the updated model from literally being a submodel, as each state in the updated model is a pair consisting of the previous state and the action type. For this reason, those who want to define a DEL semantics that only accommodates the public announcement action signature, may define the update product as simply taking the submodel for which the precondition is true. Such logic is often called Public Announcement Logic (PAL). The opening example only uses public announcements, and hence we will first consider adding yesterday to PAL, resulting in a simple form of TDEL that we shall call temporal public announcement logic (TPAL).

6.2 Histories

The primary structure used in TDEL is a history. We define TDEL-histories effectively as a list of state models similarly to the way we defined TPAL-histories. Rather than each subsequent state model being a submodel of the previous, each subsequent state model in a TDEL-history is some update product of the previous state model together with a list of preconditions. We simultaneously define histories with the function $\widehat{\mathbf{S}}$ on the new domain of TDEL-histories. The function $\widehat{\mathbf{S}}$ is defined as before, except here it extracts the most recent state model from a TDEL-history, rather than from a TPAL-history.

Definition 6.7 (TDEL-History and function $\widehat{\mathbf{S}}$). Given an action signature $\mathbf{\Sigma}$ and a set Φ of atomic propositions, we generate the set of histories $\mathfrak{H}$ recursively as follows

- Any singleton list $(\mathbf{S})$ is a history, where $\mathbf{S}$ is a state model.
- If H is a history, and $\vec{X}$ is a sequence of preconditions, then $(H, \widehat{\mathbf{S}}(H) \otimes \vec{X})$ is also a history.

We define $\widehat{\mathbf{S}}$ as a function mapping histories to state models by $\widehat{\mathbf{S}}((\mathbf{S})) = \mathbf{S}$ and $\widehat{\mathbf{S}}((H, \mathbf{S})) = \mathbf{S}$, where H is a history, and $\mathbf{S}$ is a state model.

As formulas in DEL are evaluated in coherent model/state pairs, formulas in TDEL will typically be evaluated in coherent history/state pairs. Similarly to the coherent model/state pairs, a history/state pair is coherent if the state is contained in the last (most recent) model in the history.

This shift from model to history should also be reflected in the notion of the precondition; our new precondition is a collection of coherent history/state pairs. We can distinguish between this kind of precondition and the one consisting of model/state pairs by calling this a *history precondition* and the former a *model precondition*. But usually this distinction will be apparent from context, and the initial distinguishing word may be omitted.

We can also extend the domain of the update product to determine a history from both an old history and a tuple of history preconditions as follows.

Definition 6.8 (Update product for histories). Fix an action signature $\mathbf{\Sigma}$ and a set Φ of atomic propositions. Given a history H and a vector $\vec{X}$ of history preconditions, we define $H \otimes \vec{X}$ to be the history $(H, \widehat{\mathbf{S}}(H) \otimes \vec{Y})$, where $Y_i = \{(\widehat{\mathbf{S}}(H), s) : (H, s) \in X_i\}$ for $1 \leq i \leq n$.

6.3 Examples

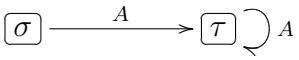
We present three examples of histories, each using different actions signatures. To help us interpret these examples, notice that in general, given an action signature, an action type σ and an agent A, if the only A-arrow from the action type σ is the reflexive arrow to σ, then σ is an announcement to A (though σ may reflect more than just the announcement,

such as what agents believe about what other agents are learning). If an action type σ is such that for every agent A the only A-arrow from the action type σ is the reflexive arrow to σ, then σ is a public announcement to A. After all, the subframe generated by σ is the public announcement action signature.

We saw earlier that a public announcement is always truthful. Here is an example that reflects information that need not be truthful.

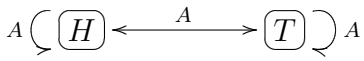
Example 6.9 (Lying or Misconception). As DEL does not provide sources of information, it is easy to blur the distinction between lying (false information from someone else) and misconception (possibly caused by no one else but oneself).

Action signature A simple action signature that can capture lying and misconception (but does not distinguish these) is depicted as follows:



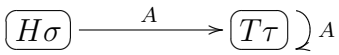
Here $\mathcal{A} = \{A\}$. Note that the action type τ is a public announcement. The action type σ corresponds to a belief by A that there is a public announcement of the precondition of τ. One place where misconception can arise is at a model/state pair that is in the precondition of σ but not in the precondition of τ. If σ occurs, then A, believing τ is the action type that occurred, would believe the precondition for τ. As the original coherent model/state pair was not in the precondition of τ, A would believe this pair not to be the actual pair, even though it really was.

Sequence of Histories We will construct a history by starting with an initial state model and applying updates to it. Let us start with the state model, S_0, representing the concealed coin scenario involving one agent. Agent A enters a room and knows a coin is in a box with either heads or tails facing up, but does not know which is the case. We represent this model by



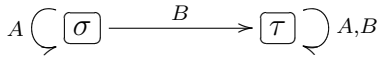
We let H and T be state names, but we also have a set of atomic propositions $\Phi = \{h, t\}$, where h and t mean the coin has heads facing up and tails facing up respectively. The valuation is as we would expect, mapping h to $\{H\}$ and t to $\{T\}$. This state model determines the history $H_0 = (S_0)$, where no time has passed.

Suppose that A begins to look in the box, but is distracted the moment she gets a glimpse of the coin. She is rather sure she saw tails, though actually she is mistaken. The coin is really heads. To perform the appropriate update product, we assign preconditions as follows: $X_\sigma = [\![h]\!]$ and $X_\tau = [\![t]\!]$, where $[\![h]\!]$ is the set of all coherent model/state pairs for which h is assigned, and similarly for $[\![t]\!]$. Let S_1 be the updated state model, and depict it by



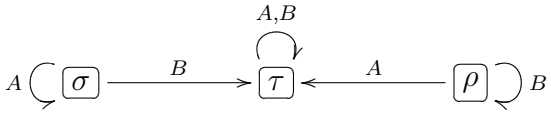
The state labeled $H\sigma$ represents the state (H, σ) and similarly for $T\tau$. The state $H\sigma$ represents the situation described above, where the coin is actually heads, and A is mistaken. The state $H\tau$ represents a situation where A still believes the coin is tails, but is not mistaken about it. We can define another history by $H_1 = (H_0, S_1)$.

Example 6.10 (Completely Private announcement). Let $\mathcal{A} = \{A, B\}$. A completely private announcement to A involves information revealed to A in such a way that B thinks nothing happened at all. We consider the action signature



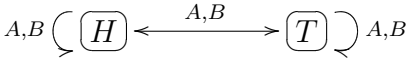
Let X_σ be the precondition of σ, and X_τ be the precondition of τ. Here τ is a public announcement of (a property characterizing) X_τ, and σ is an announcement to A of X_σ that occurs in such a way that B believes there is a public announcement of X_τ. To make σ a completely private announcement, B must believe nothing happened at all. To ensure that B believes nothing happened at all, we make the public announcement of X_τ trivial. We do this by making X_τ all coherent model/state pairs (where in each pair, the state is in the model).

Action Signature for History Suppose $\mathcal{A} = \{A, B\}$ is the set of agents, and the action signature looks like



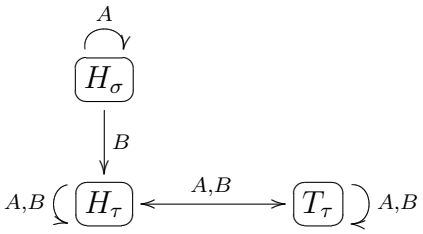
Note that τ is a public announcement. Let X_1, X_2, and X_3 be the preconditions for σ, τ, and ρ respectively. First note that τ is a public announcement of X_2. If X_2 all coherent model/state pairs, τ can be viewed as a trivial announcement or no announcement at all. Let us represent this by $X_2 = [\![\mathsf{true}]\!]$, the set of coherent model/state pairs where the formula true is true. Then σ becomes a completely private announcement to A of X_1, and ρ becomes a completely private announcement to B of X_3.

Sequence of completely private announcements recorded in histories Let us start with the state model, S_0, representing the concealed coin scenario involving two agents. Two agents enter a room and know a coin is in a box with either heads or tails facing up, but do not know which is the case. We represent this model by



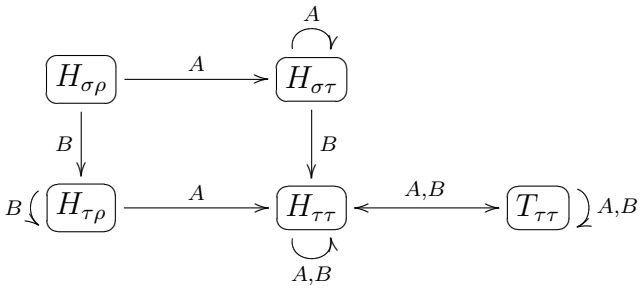
We let H and T be state names, but we also have a set of atomic propositions $\Phi = \{h, t\}$, where h and t mean the coin has heads facing up and tails facing up respectively. The valuation is as we would expect, mapping h to $\{H\}$ and t to $\{T\}$.

The first action[6] shall be represented by the precondition list $[\![h]\!], [\![\,\mathsf{true}\,]\!], \emptyset$, where $[\![h]\!]$ is the set of all coherent model/state pairs where h is assigned. This list provides a precondition for each action type. We shall view σ as a completely private announcement to A of the status of the coin. The result of this action is the update product of S_0 with this precondition list. This product shall be represented by the state model S_1:



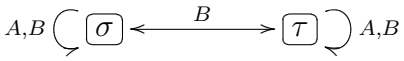
In the names of each state, the Latin letter corresponds to the original state (the first coordinate), and the Greek letter corresponds to the action type that led to that state (the second coordinate).

The next action is represented by the list $\emptyset, [\![h]\!], [\![\,\mathsf{true}\,]\!]$. We can view ρ as a completely-private announcement to B of the status of the coin. The update product between S_1 and this precondition list is S_2:



where the state labeled $H_{\sigma\rho}$ represents the state $((\mathrm{H}, \sigma), \rho)$, and other states follow a similar pattern.

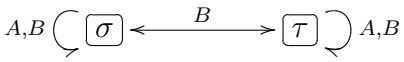
Example 6.11 (Semi-Private announcement). A semi-private announcement to A involves revealing information to A in such a way that B knows that A learned something, yet does not know the content of what is learned. Let $\mathcal{A} = \{A, B\}$.

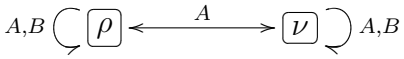


Let X_σ be the precondition of σ, and X_τ the precondition for τ. If $X_\sigma = X_\tau$, then both σ and τ are effectively public announcements, in the sense that the precondition X_σ (equal to X_τ) becomes common knowledge, and there is no uncertainty about what was announced. If $X_\sigma \neq X_\tau$, then σ and τ are semi-private announcements to A of X_σ and X_τ respectively.

[6]We use the term "action" informally. We use the word here because although this term may more appropriately denote an action type in the presence of a precondition, the only part of this action needed to perform the update product is the precondition itself (no specific action type need be specified).

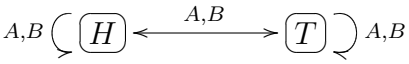
Action signature for history Suppose $\mathcal{A} = \{A, B\}$ is the set of agents, and the action signature looks like





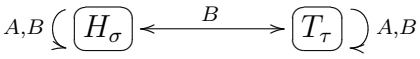
Action types σ, τ, ρ, and ν will induce specific actions when provided with sets X_1, X_2, X_3, and X_4 used in the update product. These sets are interpreted as preconditions for the action types. Focusing on action types σ and τ, and assuming X_1 and X_2 to be distinct, agent A will know which precondition is correct, and B will not. But B will be aware that A knows. This makes both types σ and τ induce semi-private announcements to A. Similarly, types ρ and ν induce semi-private announcements to B.

Sequence of semi-private announcements recorded in histories Let us start with the state model, S_0, representing the concealed coin scenario with two agents. Two agents enter a room and know a coin is in a box with either heads or tails facing up, but do not know which is the case. We represent this model by



We let H and T be state names, but we also have a set of atomic propositions $\Phi = \{h, t\}$, where h and t mean the coin has heads facing up and tails facing up respectively. The valuation is as we would expect, mapping h to $\{H\}$ and t to $\{T\}$.

The first action shall be represented by $\{H\}, \{T\}, \emptyset, \emptyset$. We can view this action as a semi-private announcement to A of the status of the coin. The update product between S_0 and this precondition list is the state model S_1:



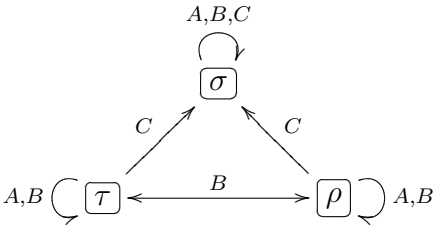
In the names of each state, the Latin letter corresponds to the original state (the first coordinate), and the Greek letter corresponds to the action type that lead to that state (the second coordinate). Hence the state labeled H_σ is the state (H, σ) and state labeled T_τ is the state (T, τ).

The next action is represented by the list $\emptyset, \emptyset, \{H\}, \{T\}$. We can view this action as a semi-private announcement to B of the status of the coin. The update product between S_1 and this precondition list results in the state model S_2:

The state labeled $H_{\sigma\rho}$ is the state $((H, \sigma), \rho)$ according to the update products, and the state labeled $T_{\tau\nu}$ is the state $((T, \tau), \nu)$.

Example 6.12 (Action signature for continuing concealed coin example and record in history). If S_0 were the state model in the concealed coin example 2.2, then $H_0 = (S_0)$ is clearly a history. Let S_1 be the state model in example 6.1. We see that $H_1 = (H_0, S_1)$ is also a history by defining an appropriate action signature and preconditions for the update from S_0 to S_1.

As in examples 2.2 and 6.1, $\mathcal{A} = \{A, B, C\}$. The action signature involved in the action which led to the model in example 6.1 is as follows.



The action type corresponding to the situation described by example 6.1 was either τ or ρ and the preconditions were as follows:

- X_σ shall be the set of all coherent model/state pairs (that is a trivial announcement), so that C thinks nothing happened.
- X_τ shall be the set of all coherent model/state pairs for which h is true.
- X_ρ shall be the set of all coherent model/state pairs for which t is true.

Note that τ and ρ can each be considered to be an announcement to A that is completely private from C and semi-private from B.

6.4 Temporal Dynamic Epistemic Languages

Fix an action signature $\mathbf{\Sigma}$. Based on this action signature, we create a new language $\mathcal{L}_Y^{\mathrm{DEL}}$ by changing the basic programs in $\mathcal{L}_Y^{\mathrm{PAL}}$. In $\mathcal{L}_Y^{\mathrm{PAL}}$, basic programs have the form $\psi!$, where ψ is a sentence. For $\mathcal{L}_Y^{\mathrm{DEL}}$, basic programs have the form $\sigma, \psi_1, \ldots, \psi_n$, where n is the number of action types in Σ. The semantics is defined in much the same way as for $\mathcal{L}_Y^{\mathrm{PAL}}$, but with the following key case for the basic program.

- $(H, s)[\![\sigma\varphi_1, \ldots, \varphi_n]\!](H', s')$ iff $[\![\varphi_i]\!]$ is the set of coherent history/state pairs making φ_i true, $H' = H \otimes ([\![\varphi_1]\!], \ldots, [\![\varphi_n]\!])$, and $s' = (s, \sigma)$.

If $\mathbf{\Sigma}$ is the public announcement action signature given by example 6.6, then there is just one action type σ. The languages $\mathcal{L}_Y^{\mathrm{DEL}}$ and $\mathcal{L}_Y^{\mathrm{PAL}}$ differ only in superficial notation: the action type σ appears in every $\mathcal{L}_Y^{DEL}$ basic program, while the symbol ! appears in every $\mathcal{L}_Y^{PAL}$ basic programs.

6.4.1 Action Types and Completeness

We obtain a new language $\mathcal{L}^{\mathrm{DEL}}_{YA}$ by adding to $\mathcal{L}^{\mathrm{DEL}}_{Y}$ the sentences of the form σ, where $\sigma \in \Sigma$. Syntactically, σ can appear in a formula anywhere that an atomic proposition $p \in \Phi$ can appear. The sentence σ tells us that the last action type that occurred was σ. Many action types, however, might not have nice English interpretations such as public announcement or private announcement (and we have even seen that the action types in examples 6.10 and 6.11 cannot always be considered completely-private or semi-private announcements), and we may prefer a language that helps us express the previous action more completely. But the language $\mathcal{L}^{\mathrm{DEL}}_{YA}$ allows us to express everything we can in $\mathcal{L}^{\mathrm{DEL}}_{Y}$, and has the additional benefit of allowing us to express axioms that are helpful in proving completeness.

A completeness proof using an $\mathcal{L}^{\mathrm{DEL}}_{YA}$ proof system for each action signature is given in [Sack, 2007b]. By removing from this proof system every axiom that involves an action type, we obtain an $\mathcal{L}^{\mathrm{DEL}}_{Y}$ proof system in which completeness depends on the action signature. If the action signature includes an action type that is reflexive for every agent, then a slight modification of the completeness proof that uses the $\mathcal{L}^{\mathrm{DEL}}_{YA}$ proof system can be used to prove completeness using the corresponding $\mathcal{L}^{\mathrm{DEL}}_{Y}$ proof system. Although this falls short of the ideal provided by the completeness of the $\mathcal{L}^{\mathrm{DEL}}_{YA}$ system, many important action signatures, including those in all examples presented in this paper, do have an action type that is reflexive for every agent, and hence with these, there is a sound and complete $\mathcal{L}^{\mathrm{DEL}}_{Y}$ proof system. For some action signatures that do not have an action type that is reflexive for every agent, we know that such a proof system is not complete, though there still may be other axioms that can be added that might result in a complete system. For example, if the action signature contains no A-relational connections, then the formula $Y_{\Diamond}\,\mathsf{true} \rightarrow \Box_A\,\mathsf{false}$ is valid. Note that this formula is not valid if the action signature contains at least one A-relational connection, and hence should not be provable in either the $\mathcal{L}^{\mathrm{DEL}}_{YA}$ proof system or the corresponding $\mathcal{L}^{\mathrm{DEL}}_{Y}$ proof system. Different action signatures may require that different approaches be used to establish a complete $\mathcal{L}^{\mathrm{DEL}}_{Y}$ proof system.

The most important axiom scheme involving action types may be $\sigma \rightarrow \Box_A \neg\tau$, for each $A \in \mathcal{A}$ and each $\sigma, \tau \in \Sigma$ for which it is not the case that $\sigma \xrightarrow{A}_{\mathbf{\Sigma}} \tau$. Each axiom of the form $\sigma \rightarrow \Box_A \neg\tau$ asserts that if σ were the last action type to have occurred, then A would believe τ to have been impossible, which is also the English reading we get by interpreting the absence of an A-relational link from σ to τ in the action signature. You may notice that no relational connections are absent in the public announcement action signature, and hence no such axioms would be needed. Thus it should not be surprising that, with the public announcement action signature, the completeness proof using the $\mathcal{L}^{\mathrm{DEL}}_{YA}$ proof system mentioned above can be modified very slightly to be a completeness proof using the corresponding $\mathcal{L}^{\mathrm{DEL}}_{Y}$ proof system.[7]

[7]Note that these proof systems are not the same, for even though the axioms of the form $\sigma \rightarrow \Box_A\tau$ are absent from this $\mathcal{L}^{\mathrm{DEL}}_{YA}$ proof system, there must be other axioms involving action types, among

For either proof system, when constructing a history for a given consistent formula, we start by constructing a filtration whose states are equivalence classes of maximally consistent sets.[8] When using the $\mathcal{L}_{YA}^{\mathrm{DEL}}$ proof system, the syntactic structure of each state determines what action type if any must be true at that state. When using the $\mathcal{L}_{Y}^{\mathrm{DEL}}$ proof system, we decide ourselves what action type if any shall be true at what state. We may consider using the same action type to be true at each state that we consider to be a non-initial state (a state for which $Y_{\Diamond}\mathsf{true}$ holds). If, however, the selected action type is not reflexive in the action signature for some agent A, then we see by investigating the definition of the update product that the action type cannot be true at any two states for which one is A-related to the other. When using the $\mathcal{L}_{YA}^{\mathrm{DEL}}$ proof system, the axiom $\sigma \to \Box_A \neg\sigma$ would ensure that this difficulty never arises. But when using $\mathcal{L}_{Y}^{\mathrm{DEL}}$, we can avoid this difficulty if we can pick a single action type that is reflexive for every agent. If the action signature has an action type that is reflexive for every agent, this strategy can be used in proving completeness.

6.4.2 Future Operator in TDEL

We obtain a new language $\mathcal{L}_{YT}^{\mathrm{DEL}}$ by adding a next-time operator $T_{\Box}$ to $\mathcal{L}_{Y}^{\mathrm{DEL}}$. Here the semantics is

- $(H, s) \in [\![T_{\Box}\varphi]\!]$ iff $(H', s') \in [\![\varphi]\!]$ for every basic program π and (H', s') for which $(H, s)[\![\pi]\!](H', s')$.

Equivalently, $(H, s) \in T_{\Box}\varphi$ iff $(H, s) \in [\![[\pi]\varphi]\!]$ for every basic program π. By removing $Y_{\Box}$ from $\mathcal{L}_{YT}^{\mathrm{DEL}}$, we obtain the language $\mathcal{L}_{T}^{\mathrm{DEL}}$, which is just DEL together with the operator $T_{\Box}$.

Much of the discussion on future in section 4 transfers here too. The formulas $\Box_A\varphi \to T_{\Box}^n \Box_A Y_{\Diamond}^n \varphi$ are valid for each n regardless of action signature, reflecting the fact that belief revision does not take place in DEL. Also formulas φ for which $\varphi \to T_{\Box}\varphi$ are valid are closed under epistemic operators, conjunction, and common knowledge operators regardless of action signature. We can ask the same questions about knowability and Fitch's paradox here too: for what action signature and sentences φ is $\varphi \to T_{\Diamond}\Box_A\varphi$ true?

7 Conclusion

We have seen how adding a last-time operator to PAL allows us to express how an agent's beliefs flipped twice. By adding a next-time operator to this, we can express that this change of belief is not due to belief revision. Agents always maintain their beliefs about features of a particular stage in time. But agents may change their beliefs about facts concerning the present when they anticipate that these facts themselves may have

which are axioms that assert that σ cannot be true in an initial state (a state for which $Y_{\Box}\mathsf{false}$ is true).

[8] This filtration will most likely not be a history, and involves different semantics than that given in this paper. A number of modifications will transform it into a structure that is bisimilar to a history.

changed. This remains true in DEL too: agents only change their beliefs when they anticipate the facts may have changed, regardless of action signature. Adding both last-time and next-time operators to DEL, we can express formulas that reflect that belief revision does not take place in DEL too.

We have seen how moves in a game can be made by public announcements, and that either inverse programs or hybrid logic can help us characterize which TPAL-histories correspond to the play of a given game. The relationship between PAL and games has not been well emphasized before, and this may be largely due to the fact that PAL does not undergo belief revision. But this relationship does capture the fact that epistemic updates are fundamental to games, as perfect information extensive game involve updates.

Some TPAL languages have complete proof systems. There is a complete proof system for the language that adds the last-time operator to PAL. Also a proof system for the language that adds a next-time operator has been established with respect to models for which every epistemic relation is an equivalence relation. No proof system has been established for TPAL languages that have inverse programs or hybrid logic.

A complete proof system for the language that adds the last-time operator to DEL has been determined only for action signatures that contain some action type that is reflexive for every agent. It is not known whether there are complete proof systems with other action signatures Adding action types to the language allows us to have a complete proof system for any action signature. Proof systems for languages that add next-time operators to DEL (with action signatures other than the public announcement one) have not been determined yet.

Not explored in this paper are always-in-the-past and always-in-the-future operators. These would be useful operators to have, though no completeness or decidability results have been determined.

Acknowledgements

I would like to thank my advisor Lawrence Moss for his helpful and encouraging advice. I also thank Alexandru Baltag, Johan van Benthem, Esfandiar Haghverdi, Chung-min Lee, Daniel Leivant, and Eric Pacuit for their comments.

References

[Alchourron et al., 1985] Alchourron, C.E., Gardenfors, P. & Makinson, D. (1985). On the Logic of Theory Change: Partial Meet Contraction and Revision Functions. *The Journal of Symbolic Logic*, 50, No 2, 510–530.

[Balbiani et al., 2007] Balbiani, P., Baltag, A., van Ditmarsch, H., Herzig, A., Hoshi, T. & de Lima, T. (2007). What can we achieve by arbitrary announcements? A dynamic

take on Fitch's knowability. *Proceedings of Theoretical Aspects of Rationality and Knowledge.*

[Baltag and Moss, 2004] Baltag, A. & Moss, L. (2004). Logics for Epistemic Programs. *Synthese 139*: 165–224.

[Baltag et al., 2003] Baltag, A., Moss, L. & Solecki, S. (2003). Logics for Epistemic Actions: Completeness, Decidability, Expressivity. *ms. Indiana University.*

[Baltag and Smets, 2006] Baltag, A. & Smets, S. (2006). Conditional Doxastic Models: A Qualitative Approach to Dynamic Belief Revision. *www.elsevier.nl/locate/entcs (Electronic Notes in Theoretical Computer Science.)*

[van Benthem, 2007] van Benthem, J. (2007). What One May Come to Know. *ms. Universiteit van Amsterdam.*

[van Benthem and Liu, 2004] van Benthem, J.& Liu, F. (2004). Diversity of logical agents in games. *Philosophica Scientiae*, 8(2)163-178.

[van Benthem and Pacuit, 2006] van Benthem, J.& Pacuit, E. (2006). The Tree of Knowledge in Action: Towards a Common Perspective. *Proceedings of Advances in Modal Logic.*

[Blackburn and Seligman, 1995] Blackburn, P. & Seligman, J. (1995). Hybrid Logic. *Journal of Logic, Language and Information.* 4, 251-272.

[Brogaard and Salerno, 2002] Brogaard, B. & Salerno, J. (2002). Fitch's Paradox of Knowability. *Stanford Electronic Encyclopedia of Philosophy, http://plato.stanford.edu/entries/fitch-paradox/.*

[Bull and Segerberg, 2001] Bull, R. & Segerberg, K. (2001). Basic Modal Logic. (In D. Gabbay, F. Guenthner (Eds.), *Handbook of Philosophical Logic, 2nd Edition vol 3.* Kluwer Academic Publisher, Boston.)

[van Ditmarsch et al., 2007] van Ditmarsch, H.P., Ruan, J. & Verbrugge, L.C. (2007). Sum and Product in Dynamic Epistemic Logic. *Journal of Logic and Computation*, to appear.

[Fagin et al., 1995] Fagin, R., Halpern, J., Moses, Y. & Vardi M. (1995). *Reasoning About Knowledge.* The MIT Press, Boston.

[Frech et al., 2005] French, T., van der Meyden, R. & Reynolds, M. (2005). Axioms for Logics of Knowledge and Past Time: Synchrony and Unique Initial States. *Proceedings of Advances in Modal Logic.*

[Gabbay et al., 1994] Gabbay, D., Hodkinson, I. & Reynolds, M. (1994). *Temporal Logic: mathematical foundations and computational aspects vol 1.* Oxford University Press.

[Gerbrandy, 1998] Gerbrandy, D. (1998). *Bisimulations on Planet Kripke.* PhD thesis, ILLC, Universiteit van Amsterdam.

[Meyer and van der Hoek, 1995] Meyer, J.J. Ch. & van der Hoek, W. (1995). *Epistemic Logic for AI and Computer Science.* Cambridge Tracts in Theoretical Computer Science 41.

[Miller and Moss, 2005] Miller, J. & Moss, L. (2005). Undecidability of Iterated Modal Relativization. *Studia Logica*, 79(3).

[Moore, 1963] Moore, G.E. *The Commonplace Book 1919–1953.* Casimir Lewy (Ed.), London: Allen & Unwin. 1963.

[Nute, 1980] Nute, D. (1980). *Topics in Conditional Logic.* D. Reidel Publishing Company, Boston.

[Parikh and Ramanujam, 2003] Parikh, R. & Ramanujam, R. (2003). A knowledge based semantics of messages. *Journal of Logic, Language, and Information*, 12:453–467.

[Plaza, 1989] Plaza, J. (1989). Logics of Public Communications. *Proceedings of 4th International Symposium on Methodologies for Intelligent Systems.*

[Sack, 2007a] Sack, J. (2007). *Adding Temporal Logic to Dynamic Epistemic Logic.* PhD thesis, Indiana University.

[Sack, 2007b] Sack, J. (2007). Logic for Update Products and Steps into the Past. *ms. California State University Long Beach.*

[Yap, 2005] Yap, A. (2005). Product Update and Looking Backward. *ms. Stanford University. Also at www.illc.uva.nl/lgc/papers/bms-temporal.pdf*