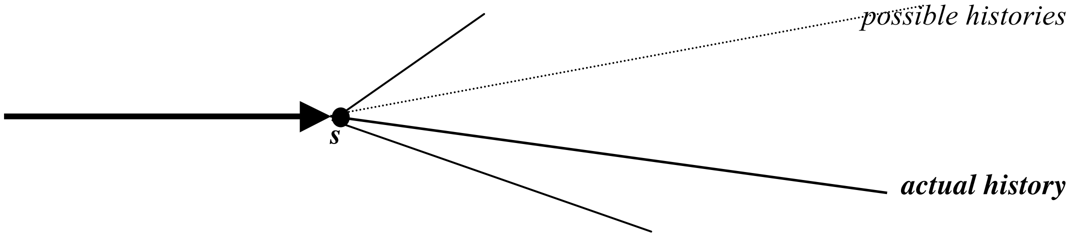
Chapter 11 PROCESSES OVER TIME

The preceding chapters took our study of rational agency from single update steps to mid-term activities like finite games that mix agent's actions, beliefs and preferences. In the limit, this leads to long-term behaviour over possibly infinite time, that has many features of its own. In particular, in addition to information about facts, agents can now have procedural information about the process they are in. This chapter makes a junction between dynamic epistemic logic and temporal logics of discrete events, occurring in philosophy, computer science, and other disciplines. We prove semantic representation theorems, and show how dynamic-epistemic languages are fragments of temporal ones for the evolution of knowledge and belief. Amongst other things, this gives a better understanding of the balance between expressive power and computational complexity for agent logics. We also show how these links, once found, lead to merges of ideas between frameworks, proposing new systems of *PAL* or *DEL* with informational protocols.

11.1 Dynamic epistemic logic meets temporal logics

The Grand Stage The following global view has surfaced at various places in Chapters 4, 10, in branching tree-like pictures for agents over time:



Branching temporal models are a Grand Stage view of agency, with histories as complete runs of some information-driven process, described by languages with epistemic and temporal operators. [246] The Grand Stage is a natural habitat for the local dynamics of *DEL*, and this chapter brings the two views together. Temporal trees can be created through constructive unfolding of an initial epistemic model ***M*** by successive product updates ***M x E*** with event models ***E*** (cf. the 'update evolution' of Chapter 4), and we will determine which trees arise in this way. Thus, *DEL* adds fine-structure to temporal models. We will use this to connect facts about epistemic-temporal logics and our findings about *DEL*.

---

[246] This view underlies Interpreted Systems (Fagin et al. 1995), Epistemic-Temporal Logic (Parikh & Ramanujam 2001), *STIT* (Belnap et al. 2001), or Game Semantics (Abramsky 2008).

Protocols Linking frameworks leads to flow of ideas. Our key example will be *protocols,* constraints on possible histories of an agent process. Message-passing systems may demand that only true information is passed, or each request is answered. In conversation, some things cannot be said, and there are rules like 'do not repeat yourself', 'let others speak in turn'. Restricting the legitimate sequences of announcements affects our logics:

Example Contracting consecutive assertions.

A *PAL*-validity in Chapter 3 stated that the effect of two consecutive announcements $!P$, $!Q$ is the same as that of the single announcement $!(P \wedge [!P]Q)$. This equivalence may fail in protocol-based models, as the latter trick assertion may not be an admissible one. ■

Protocols occur in puzzles (the Muddy children made only epistemic statements), games, and learning. Physical experiments, too, obey protocols, in line with our broader view of *PAL* and *DEL* as logics of observation. Finally, knowing a protocol is a new form of *procedural information* (cf. Chapter 5) beyond information about facts and other agents.

11.2 Basics of epistemic temporal logic

Temporal logics come in flavours (cf. Hodkinson & Reynolds 2006, van Benthem & Pacuit 2006). Chapter 9 used complete branches (perhaps infinite) for actual histories plus finite stages on them. In this chapter, we use only finite histories as indices of evaluation, living in a modalized future of possible histories extending the current one.

Models and language Take sets $\mathbb{A}$ of agents and $\mathbb{E}$ of events (usually finite). A *history* is a finite sequence of events, and $\mathbb{E}^*$ is the set of all histories. Here he is history h followed by event e, representing the unique history after e has happened in h. We write $h \leq h'$ if h is a prefix of h', and $h \leq_e h'$ if $h' = he$. Our first semantic notion represents protocols. [247]

Definition *ETL* Frames.

A *protocol* is a set of histories $\mathbb{H} \subseteq \mathbb{E}^*$ closed under prefixes. An *ETL frame* is a tuple $(\mathbb{E}, \mathbb{H}, \{\sim_i\}_{i\in\mathbb{A}})$ with a protocol $\mathbb{H}$, and accessibility relations $\sim_i$. An *ETL-model* is an *ETL*-frame plus a valuation map V sending proposition letters to sets of histories in $\mathbb{H}$. ■

An *ETL* frame describes how knowledge evolves over time in some informational process. The relations $\sim_i$ represent uncertainty of agents about how the current history has evolved,

[247] In what follows, a protocol is a family of finite histories. A more general setting would allow for *infinite histories*, where protocols need not reduce to such finitely presented ones.

due to their limited powers of observation or memory. Thus, $h \sim_i h'$ means that from agent i's point of view, the history h' looks the same as the history h.

An *epistemic temporal language* L_{ETL} for these structures extends *EL* from Chapter 2 with event modalities. It is generated from a set of atomic propositions *At* by this syntax:

$$p \mid \neg\varphi \mid \varphi\wedge\psi \mid [i]\varphi \mid \langle e\rangle\varphi \qquad \text{where } i\in\mathbb{A},\ e\in\mathbb{E}, \text{ and } p\in At.$$

Here $[i]\varphi$ stands for $K_i\varphi$. Booleans, and dual modalities $\langle i\rangle$, $[e]$ are as usual. [248]

Definition Truth of L_{ETL} formulas.

Let $\boldsymbol{M} = (\mathbb{E}, \mathbb{H}, \{\sim_i\}_{i\in\mathbb{A}}, V)$ be an *ETL* model. The truth of a formula φ at a history $h\in\mathbb{H}$, denoted $\boldsymbol{M}, h \models \varphi$, is defined inductively as usual, with the following key clauses:

(a) $\boldsymbol{M}, h \models [i]\varphi$ iff for each $h'\in\mathbb{H}$, if $h \sim_i h'$, then $\boldsymbol{M}, h' \models \varphi$

(b) $\boldsymbol{M}, h \models \langle e\rangle\varphi$ iff there exists $h'=he\in\mathbb{H}$ with $\boldsymbol{M}, h' \models \varphi$. ■

Agent properties Further constraints on models reflect special features of agents, or of the informational process of the model. [249] These come as conditions on epistemic and action accessibility, or as epistemic-temporal axioms matched by modal frame correspondences. Here are some examples from earlier chapters (we suppress indices for convenience):

Fact The axiom $K[e]\varphi \rightarrow [e]K\varphi$ corresponds to *Perfect Recall:*

if $he \sim k$, then there is a history h' with $k = h'e$ and $h \sim h'$. [250]

This says that agents' current uncertainties can only come from previous uncertainties: a strong form of perfect memory. An induction on distance from the root then derives:

Synchronicity: uncertainties $h \sim k$ only occur between h, k at the same tree level.

Weaker forms of Perfect Recall in game theory lack Synchronicity, allowing uncertainty links that cross between tree levels. Note that the axiom presupposes perfect observation of the current event e: in *DEL*, it would not hold, as uncertainty can also be created by the current observation, when some event f is indistinguishable from e for the agent.

---

[248] We can add group operators of distributed or common knowledge, as earlier chapters. In temporal logic, such extensions can have dramatic effects on the complexity of validity: see below.

[249] The border-line can be vague: am I clever as an agent, or thanks to the process I am in?

[250] The elementary proof uses a simple modal substitution argument. Details simplify by assuming, as in our tree models, that transition relations for events e are *partial functions*.

This point will return in our analysis below. In a similar fashion, we have a dual fact:

Fact The axiom $[e]K\varphi \rightarrow K[e]\varphi$ corresponds to *No Miracles:*
for all ke with $h \sim k$, we also have $he \sim ke$. [251]

This principle is sometimes called 'No Learning', but its content is rather that learning can take place, but only by observing events to resolve current uncertainties.

Epistemic-temporal languages also describe other agents. Take a *memory-free* automaton that only remembers the last-observed event, making any two histories *he, ke* ending in the same event epistemically accessible. Then, with finitely many events, knowledge of the automaton can be defined in the temporal part of the language. Using backward modalities P_e plus a universal modality U over all histories, we have the equivalence

$$K\varphi \leftrightarrow \vee_e (\langle e^{\cup}\rangle T \wedge U((\langle e^{\cup}\rangle T \rightarrow \varphi)))$$

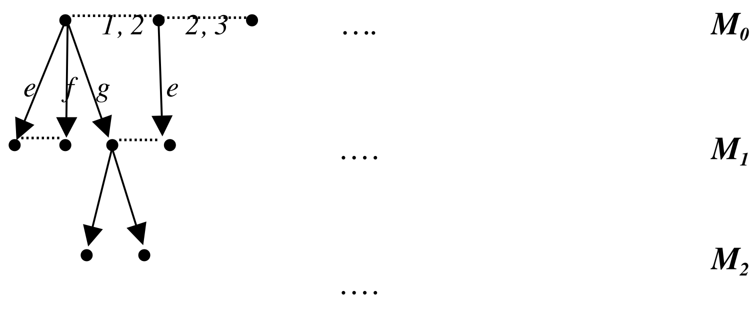
Similar ideas work for bounded memory in general (Halpern & Vardi 1989, Liu 2008). Thus, properties of processes and agents meet with epistemic-temporal languages. [252]

11.3 A basic representation theorem

Now we can state how *DEL* and *ETL* are related. Recall the scenario of *Update Evolution* in Chapter 4: some initial epistemic model $\boldsymbol{M}$ is given, and it then gets transformed by the gradual application of event models $\boldsymbol{E_1}, \boldsymbol{E_2}, \ldots$ to form a sequence

$$\boldsymbol{M_0 = M,\ \ M_1 = M_0 \times E_1,\ \ M_2 = M_1 \times E_2,\ \ldots} \qquad [253]$$

It helps to visualize this in trees, or rather forest pictures like the following:



[251] When we write h, he, etc., we always assume that these histories occur in the protocol.

[252] Further questions arise with common knowledge. For instance, if we assume Perfect Recall for all individual agents separately, it also hold for the group: $C_G[e]\varphi \rightarrow [e]C_G\varphi$ becomes valid.

[253] An important special case had one event model $\boldsymbol{E}$ throughout. Van Benthem & Liu 2004 suggest that the latter simple format suffices for mimicking the more general approach here.

where stages are horizontal, while worlds may extend downward via 0, 1, or more event successors. Through product update, worlds in these models arise from successive pair formation, forming finite sequences starting with one world in the initial epistemic model ***M*** followed by a finite sequence of events that were executable when their turn came. But that means that these worlds are just histories in the sense of the above *ETL* models.

Definition Induced *ETL* forests.

Given a model ***M*** and a finite or countable sequence of event models $\mathbb{E}$, the *induced ETL-model Forest(**M**,* $\mathbb{E}$*)* has as its histories all finite sequences $(w, e_1, .., e_k)$ produced by successive product update, with accessibility relations and valuation as in *DEL*. ■

Drawing pictures shows how this works. In particular, induced *ETL*-models have a simple protocol $\mathbb{H}$ given by the finite sequences that pass the local requirements of the update rule. Accordingly, they have three striking properties making them stand out:

Fact *ETL*-models ***H*** of the form *Forest(**M**,* $\mathbb{E}$*)* satisfy the following three principles, where quantified variables *h, h', k, …* range only over histories present in ***M***:

(a) If $he \sim k$, then there is some f with $k = h'f$ and $h \sim h'$ *Perfect Recall*

(b) If $h \sim k$, and $h'e \sim k'f$, then $he \sim kf$ *Uniform No Miracles*

(c) The domain of any event e is definable in the epistemic base language. *Definable Executability*

Now, the crucial observation is that this can be converted to prove a representation result for *DEL* inside *ETL* (van Benthem 2001, van Benthem & Liu 2004):

Theorem For *ETL* models ***H***, the following two conditions are equivalent:

(a) ***H*** is isomorphic to some model *Forest(**M**,* $\mathbb{E}$*)*,

(b) ***H*** satisfies Perfect Recall, Uniform No Miracles, and Definable Executability.

Proof The direction from (a) to (b) is the preceding Fact. Conversely, consider any *ETL*-model ***H*** satisfying the three conditions. We define an update sequence as follows:

(i) ***M*** is the set of histories in ***H*** of length *1*, copying their given epistemic accessibilities and valuation,

(ii) $\boldsymbol{E}_k$ is the set of events occurring at tree level $k+1$ in ***H***, setting $e \sim f$ if there exist histories h, k of length k with $he \sim kf$ in ***H***. Definability of preconditions is the Definable Executability.

We prove by induction that the tree levels $\boldsymbol{H_k}$ at depth k of the *ETL* model $\boldsymbol{H}$ are isomorphic to the epistemic models $\boldsymbol{M_k = M \; x \; E_1 \; x \; \ldots \; x \; E_{k-1}}$. The crucial fact is this, using our definition and the first two given properties (here (s, e) is the same history as *'se'*):

$$(s, e) \sim_{Hk} (t, f) \quad \text{iff} \quad (s, e) \sim_{Mk} (t, f)$$

From left to right. By Perfect Recall, $s \sim t$ in $\boldsymbol{H}_{k-1}$, and so by the inductive hypothesis, $s \sim t$ in $\boldsymbol{M}_{k-1}$. Also, by our definition, $e \sim f$ in $\boldsymbol{E_k}$. Then by the forward half of the Product Update rule, $(s, e) \sim_{Mk} (t, f)$. *From right to left*. By the other half of Product Update, $s \sim t$ in $\boldsymbol{M}_{k-1}$, and by the inductive hypothesis, $s \sim t$ in $\boldsymbol{H}_{k-1}$. Next, since $e \sim f$, by our definition, there are histories i, j with $ie \sim jf$ in $\boldsymbol{H_k}$. By Uniform No Miracles then, $se \sim tf$ holds in $\boldsymbol{H}$. ■

This result stipulates definability for preconditions of events e, i.e., the domains of the matching partial functions in the tree $\boldsymbol{H}$. Here is a purely structural version:

Theorem The preceding theorem still holds when we replace Definable Executability by *Bisimulation Invariance*: that is, closure of event domains under all purely epistemic bisimulations of the *ETL*-model $\boldsymbol{H}$.

The proof follows from two facts in Chapter 2: (a) epistemically definable sets of worlds are invariant for epistemic bisimulations, and (b) each invariant set has an explicit definition in the *infinitary* version of the epistemic language. [254] Our two results tell us how special *DEL* update is as a mechanism generating epistemic-temporal models. It is about idealized agents with perfect memory and driven by observation only, while their informational protocols involve only local epistemic conditions on executability.

Variations This is just a starting point. In particular, a mild relaxation of the definability requirement for events would allow more general preconditions referring to the epistemic *past* beyond local truth. Think of conversation with no repeated assertions: this needs a memory of what was said that need not be encoded in a local state. Also, other styles of representations might make sense, representing *ETL*-models only up to epistemic-temporal *bisimulation*. Finally, our proof method can also characterize effects of other update rules, such as the earlier $(s, e) \sim (t, f)$ *iff* $e \sim f$ for memory-free agents.

[254] While this only guarantees finite epistemic definitions for preconditions on finite models, we feel that further tightening of conditions has no real value in understanding the situation.

11.4 Temporal languages: expressive power and complexity

DEL as a temporal language The conditions in our representation for *DEL* evolution as *ETL* models suggest definability in matching epistemic-temporal languages. We saw how Perfect Recall corresponds to a syntactic operator switch between knowledge and action modalities. [255] Indeed, what is the language of *DEL* in Chapter 4 and following when viewed as an epistemic-temporal formalism? This is quite easy to answer:

> *DEL* is a static knowledge language for individual and collective epistemic agents, plus a *one-step future operator* saying what holds after some specified next event.

This is less than what can be said in general epistemic-temporal logics. The latter also have *past* operators, as we saw with event preconditions for extended protocols. [256] And typically also, a temporal language can talk about the whole *future*, and effects of arbitrary finite sequences of evens from now on. This is crucial in specifying how an information process is to behave, in terms of 'safety' and 'liveness' properties.

A complete repertoire of relevant temporal modalities will depend on how one views th relevant processe model, and the agents living inside it. For instance, with the earlier Synchronicity, it also makes sense to have a modality $<=>\varphi$ of *simultaneity* saying that φ is true at some history of the same length.

The balance with complexity But then we meet the Balance discussed in Chapter 2. Increases in expressive power may lead to upward jumps in computational *complexity* of combined logics of knowledge and time. The first investigation of these phenomena was made in Halpern & Vardi 1989. Here is a Table with a few observations from their work showing where the dangerous thresholds lie for the complexity of validity:

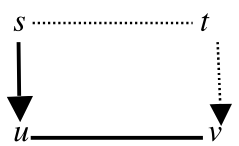
	K, P, F	K, C_G, F_e	$K, C_G,, F_e, P_e$	K, C_G, F
All *ETL* models	*decidable*	*decidable*	*decidable*	*RE*
Perfect Recall	*RE*	*RE*	*RE*	Π^1_1-complete
No Miracles	*RE*	*RE*	*RE*	Π^1_1-complete

[255] Facts like these suggest a general modal correspondence theory on *ETL*-frames.

[256] A one-step past modality Y also occurred in Chapter 3, where a public announcement $!\varphi$ achieved common knowledge that φ was true just before the event: $[!\varphi]C_GY\varphi$.

Here complexities run from decidable through axiomatizable (*RE*) to Π^1_1-complete, which is the complexity of truth for universal second-order statements in arithmetic. [257] The latter complexity is often a worst case for modal logics, witness Chapter 2. Van Benthem & Pacuit 2006 is a survey of expressive power and complexity in connection with *DEL,* citing much relevant background in work on tree logics and products of modal logics.

Dangerous agent properties As we just saw, epistemic-temporal logic over all *ETL*-models is simple even with rich vocabularies, but things change with special assumptions on agents such as Perfect Recall. The technical explanation is *grid encoding* (Chapter 2). Essentially, Perfect Recall [258] makes epistemic accessibility and future moves in time behave like a grid model of type *IN x IN*, with cells enforced by a confluence property that we have seen already (Chapters 3, 10), as pictured in the uncertainty-action diagram



Satisfiability in the language can then express the Recurrent Tiling Problem that is known to have Σ^1_1-complete complexity. But to really encode the tiling argument, the language needs sufficient expressive power, in particular, a *universal quantifier* ranging over all points in the grid. This can be supplied by combining an unbounded future modality in the tree, plus a common knowledge modality accessing all reachable points at the same horizontal tree level. If one of these resources is not available, say we have common knowledge but no unbounded future, complexity may drop, as shown in the table.

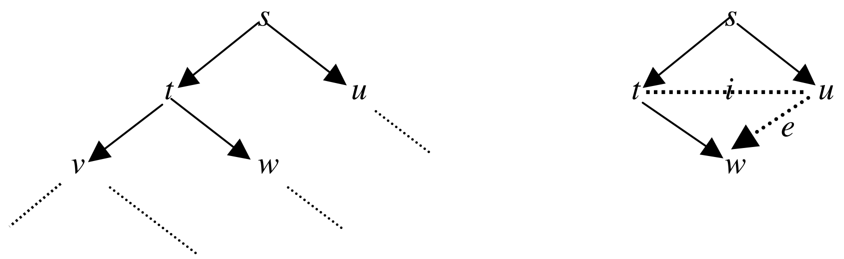
Technical points The balance in these results is subtle, partly because of a tension between two pictures in Chapter 2. As trees, our models should have simple logics, by Rabin's Theorem on the decidability of the monadic second-order logic of trees with the relation of initial segment and partial successor functions. Indeed, pure process theories can be simple in temporal logic. But the process theory of epistemic agents that handle information adds

---

[257] There is a gap between the complexities *RE* and Π^1_1*-complete*, few epistemic-temporal logics fall in between. This also occurs with extensions of first-order logic, where being able to *define a copy of the natural numbers IN* is a watershed. If you cannot, like first-order logic, complexity stays low: if you can, like first-order fixed-point logics or second-order logic, complexity jumps.

[258] Similar observations to all that follows hold for the converse principle of No Miracles.

a second relation, of epistemic accessibility, and then a tree may carry grid structure after all. Van Benthem & Pacuit 2006 explain how Perfect Recall supports tiling proofs even though it just requires basic cell structure downward in a tree. [259] Also, there is a difference between our *forest* models, where the first level may have many starting points (the worlds in the initial epistemic model ***M***), and *trees* with just one root. To get grids in trees with single roots and perhaps finite horizontal levels, we must create cells by a trick:



Here we reach the key bottom corner of a cell by an epistemic move plus an event move, as shown to the right. To make use of the latter, the language needs to mix epistemic and temporal steps in patterns $(\sim_i ; e)^*$. This requires a *propositional dynamic logic* PDL_{et} with both epistemic accessibility and temporal event moves as basic transition relations:

*Theorem (*van Benthem & Pacuit 2006) The validity problem for PDL_{et} is Π^1_1-complete.

This is one of the many points in this book where *PDL* program structure makes sense.

DEL as an ETL-logic Against this background, we can now place *DEL* and understand its behaviour in this book. Its language is the K, C_G, F_e slot in the earlier Table, over models satisfying Perfect Recall and No Miracles. Thus, there is grid structure, but the expressive resources of the language do not exploit it to the full, using only one-step future operators $<!P>$ or $<\boldsymbol{E}, e>$. If we add unbounded future, the same complexity arises as for *ETL*. Indeed, Miller & Moss 2004 show that the logic of just public announcement with common knowledge and Kleene iteration of assertions $!P$ is Π^1_1-complete. [260]

But more interesting is a language extension that came up earlier. It seems obvious that adding one-step past does not endanger decidability. But even beyond, we have this

---

[259] Models with Perfect Recall suffice for tiling *arbitrary finite sub-models* of *N x N* – and, by Koenig's Lemma, the latter suffices for the existence of a complete tiling. Also, some tricks are needed placing literals $p, \neg p$ to ensure existence of a sufficient number of branches.

[260] The Miller & Moss result leaves a loop-hole. It is not known what happens exactly to the logic over families of finite models and their sub-models reachable by public announcements.

Open problem Does *DEL* stay decidable over *ETL*-models when we add an *unbounded past* operator that can only go back finitely many steps to the root?

Discussion: complexity of agents How do complexity results for logics relate to agency? They tell us delicate things about richness of behaviour. Take the following paradox(ette). How can the logic of ideal agents with perfect memory be so highly complex, while the logic of arbitrary agents is simple, witness the first line in the above Table? A moment's reflection dissolves the paradox. The general logic describes what is true for all agents: a simple story. [261] For some special kinds of agent, that logic stays simple, as we saw with bounded memory, whose epistemic-temporal logic was embedded in the pure temporal logic of our models. But agents with perfect memory are so regular that an *ETL*-record of their activities shows grid patterns that encode arithmetical computation – and the Π^1_1-completeness says that understanding this behaviour requires substantial mathematics.

To us, this computational perspective on agency is more than a formal tool. In Chapter 14, we discuss real analogies between computation and conversation. For some remaining worries on the computational complexity of agency: see the end of this chapter.

11.5 Adding protocols to dynamic epistemic logic

Now that we have things in one setting, we can go further and create merges, transferring ideas from one framework to another. The preceding sections showed how *DEL* adds fine-structure to *ETL*. Product update is a mechanism for creating temporal models, and *DEL*-imports some of that model structure into the object language, where it becomes subject to explicit manipulation. In doing so, *DEL*-type languages suggest new fragments of *ETL* and other process languages, providing concrete new systems for investigation.

Protocols In the opposite direction, a notion missing in *DEL* is that of a temporal *protocol* defining or constraining the informational process agents are in. Clearly, this procedural information (cf. Chapters 3, 13) is crucial to agency. Now *DEL* does have *preconditions* constraining which events are executable where, cutting down on possible histories. Van Benthem & Liu 2004 suggest that this device can represent most natural protocols, especially, if we go a bit beyond preconditions defined in a pure epistemic base language. But this approach only works for protocols that are locally defined restrictions on events.

---

[261] What we say here no longer holds if we can explicitly define *agent types* inside the language.

Chapter 8 did a bit more, shifting protocol information into the *definition of the events*, using 'thick events' of forms like *Liar says P* instead of bare public announcements *!P*. But now we will take a more radical explicit approach to these matters.

DEL protocol models Much more can be learnt by also having *DEL* protocols in a straightforward *ETL* style, an idea first proposed in Gerbrandy 1999A. What follows is based on the results in van Benthem, Gerbrandy, Hoshi & Pacuit 2009:

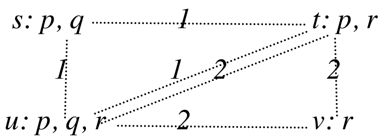
Definition *DEL* protocols.

Let $\mathbb{E}$ be the class of all pointed event models. A *DEL protocol* is a set $\boldsymbol{P} \subseteq \mathbb{E}^*$ closed under taking initial segments. Let ***M*** be any epistemic model. A *state-dependent DEL protocol* is a map ***P*** sending worlds in ***M*** to *DEL* protocols. If the protocol assigned is the same for all worlds, the state-dependent *DEL* protocol is called *uniform*. ■

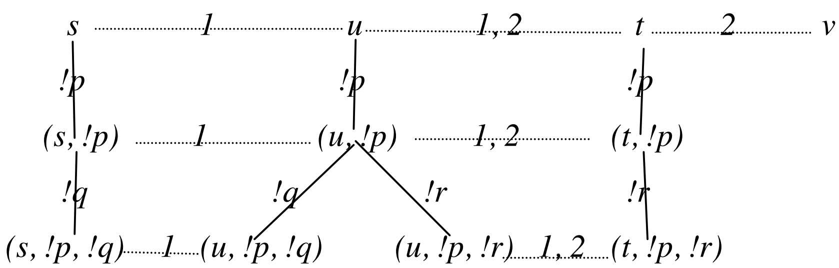
In Chapters 3, 4, the dynamic modalities of *PAL* and *DEL* were interpreted in the total universe of all possible epistemic models, representing total freedom of information flow. But now, protocols in the epistemic-temporal sense restrict the range of reachable models. Though details of this require some technical care, the idea is very simple. We extend the earlier notion of update unfolding as in the following illustration:

Example *ETL* model generated by a uniform *PAL* protocol.

We use a public announcement protocol for graphical simplicity. Consider the following epistemic model ***M*** with four worlds and agents *1, 2*:



with a protocol ***P*** = *{<!p>, <!p, !q>, <!p, !r>}* of available sequences of announcements or observations. The *ETL* forest model in the following diagram is the obvious update evolution of ***M*** in the earlier sense, with histories restricted by the event sequences in ***P***. Note how some worlds drop out, while others 'multiply':



■

Remark This example also shows the syntactic nature of our protocols. Different formulas *P, Q* lead to different events, even when semantically equivalent. This syntactic flavour, used in Hoshi 2008 for analyzing inferential information (cf. Chapter 5), is a disadvantage in some settings – but we leave more semantic definitions as a desideratum.

Now comes the general notion behind this, suppressing some technicalities: [262]

Definition Generated *ETL* models from *DEL* protocols.
Let ***M*** be an epistemic model and ***P*** a state-dependent *DEL* protocol on ***M***. The *induced ETL* (forest) *model Forest(**M**, **P**)* is defined as follows. Starting from ***M*** as the first layer, one computes the finite update evolutions of ***M*** containing only sequences $(w, e_1, .., e_k)$ where preconditions are satisfied as earlier, while now also the event sequence $(e_1, .., e_k)$ must be in the local protocol assigned by ***P*** to *w*. The complete epistemic tree or forest model is then generated as a straightforward union of these finite stages. [263] ■

State-dependent protocols are very flexible. Uniform protocols are the same in every world, making them *common knowledge*: a usual assumption in epistemic-temporal logic. [264] But when procedural information is agent dependent, there are finer distinctions:

Example A protocol that is not common knowledge.
Let the model ***M*** have two worlds *s, t* as depicted here:

s: p .... *t: p, q*

The state-dependent protocol ***P*** assigns $\{<!p>\}$ to *s* and $\{<!q>\}$ to *t*. In the induced model the formula $<!p>T$ is true at *s*, meaning that the information that *p* holds can be revealed by the protocol. But the agent does not know this procedural fact, since $<!p>T$ is false at *t*, and hence the epistemic-dynamic formula $K<!p>T$ is false at *s*. By contrast, with uniform protocols, a true formula $<!p>T$ is common knowledge: $<!p>T \rightarrow C_G<!p>T$ is valid. ■

Representation revisited Which *ETL*-models are induced by epistemic models with a state-dependent protocol? For the universal protocol of *all possible* finite sequences of event models, we get trees of all models reachable by update from some fixed model ***M***. In Section 11.3, we characterized the induced class for uniform protocols, with one

---

[262] The definitions in van Benthem, Gerbrandy, Hoshi & Pacuit 2009 run through two pages.

[263] One might also allow protocol constraints at later stages in the update evolution.

[264] This formulation makes more sense when our language can *define* protocols.

sequence of event models without branchings, using Perfect Recall, Uniform No Miracles, and Bisimulation Invariance for epistemic bisimulations. Van Benthem, Gerbrandy, Hoshi & Pacuit 2008 extend this analysis to possibly branching state-dependent protocols. [265]

Translation Finally, our current perspective may also be viewed as a syntactic translation from *DEL* to *ETL* in the following sense. Suppose we are working with the full protocol $\boldsymbol{Prot}_{DEL}$ of all finite sequences of event models. Then we have this equivalence:

Fact For any epistemic model $\boldsymbol{M}$, world w, and any formula φ in *DEL*,

$$\boldsymbol{M}, w \models \varphi \text{ iff } Forest(M, \boldsymbol{Prot}_{DEL}), \langle w \rangle \models \varphi.$$

11.6 Determining the logic of *PAL* protocols

Adding *DEL* protocols raises new questions for dynamic epistemic logic itself.

PAL protocols A telling example is information flow by public announcement. The earlier definitions specialize to protocols for conversation, or experiments where only few things can be measured, in certain orders. What is the logic of epistemic models plus models reachable by some announcement protocol? Note that *PAL* itself no longer qualifies. It was the logic of arbitrary models subjected to the universal protocol of all announcement sequences. But when the latter are constrained, two axioms from Chapter 3 will fail.

Note In the rest of this chapter, for convenience, protocols only involve pure epistemic formulas without dynamic announcement modalities. This restriction was lifted in Hoshi 2009, and most assertions below can be extended to the full language of *PAL*. We will use existential action modalities for their greater vividness in a procedural setting:

Example Failures of *PAL* validities.

PAL had a valid axiom $\langle !P \rangle q \leftrightarrow P \wedge q$. As a special case, this implies

$$\langle !P \rangle T \leftrightarrow P$$

From left to right, this is valid with any protocol: an announcement $!P$ can only be executed when P holds. But the direction from right to left is no longer valid: P may be true at the current world, but the protocol need not allow public announcement of this fact at this stage. Next, consider the crucial knowledge recursion law, in its existential version

$$\langle !P \rangle \langle i \rangle \varphi \leftrightarrow (P \wedge \langle i \rangle \langle !P \rangle \varphi)$$

---

[265] Local versions of the earlier conditions work, linking histories only to histories reachable by epistemic accessibility paths. Extracting event models from *ETL* forests takes some extra care.

This, too, fails in general from right to left. Even when P is true right now, and the agent thinks it possible that P can be announced to make φ true, she need not know the protocol – and a state-dependent protocol need not allow action $!P$ in the actual world. ■

The point is this: assertions $<!P>T$ now come to express genuine *procedural information* about the informative process agents are in, and hence, they no longer reduce to basic epistemic statements. Stated more critically, *PAL* only expressed factual and epistemic information, but left no room for genuine procedural information. We now remedy this:

Definition The logic *TPAL*.

The logic *TPAL* of arbitrary announcement protocols has the same language as *PAL*, and its axioms consist of (a) the chosen static epistemic base logic, (b) the minimal modal logic for each announcement modality, and (c) the following modified recursion axioms:

$$<!P>q \quad \leftrightarrow \quad <!P>T \wedge q \qquad \textit{for atomic facts } q$$

$$<!P>(\varphi \vee \psi) \quad \leftrightarrow \quad (<[!P>\varphi \ \vee <!P>\psi)$$

$$<!P>\neg\varphi \quad \leftrightarrow \quad <!P>T \wedge \ \neg<!P>\varphi$$ [266]

$$<!P>K_i\varphi \quad \leftrightarrow \quad <!P>T \wedge \ K_i(<!P>T \rightarrow <!P>\varphi)$$ ■

It is easy to verify that these modified recursion laws hold generally:

Fact The axioms of *TPAL* are sound on all *PAL* protocol models.

Proof We explain the validity of the crucial axiom $<!P>K_i\varphi \leftrightarrow <!P>T \wedge K_i(<!P>T \rightarrow <!P>\varphi)$. From left to right, if $<!P>K_i\varphi$ is true at world s in a model $\boldsymbol{M}$, then $!P$ is executable, i.e. $<!P>T$ holds at s. Moreover, $K_i\varphi$ holds at $(s, !P)$ in the updated model $\boldsymbol{M}|P$. Next, for each $\sim_i$-successor t of s where $<!P>T$ holds, the world $(t, !P)$ makes it into $\boldsymbol{M}|P$ as a $\sim_i$-successor of $(s, !P)$. But then φ holds at $(t, !P)$, by the truth of $K_i\varphi$ at $(s, !P)$, and $<!P>\varphi$ holds at t. From right to left, let world s in the model $\boldsymbol{M}$ satisfy $<!P>T \wedge K_i(<!P>T \rightarrow <!P>\varphi)$. By the executability of $!P$, the world $(s, !P)$ is in $\boldsymbol{M}|P$. Now consider any of its $\sim_i$-successors there: it must be of the form $(t, !P)$ with $s \sim_i t$. But then, $\boldsymbol{M}, t$ satisfies $<!P>\varphi$, and therefore $(t, !P)$ satisfies φ in the updated model $\boldsymbol{M}|P$. ■

Note that our original method of finding *recursion axioms* for effects of informational events still works. But now, it has been decoupled from the more drastic reduction to pure

[266] A useful effect of the negation axiom is this: $[!P]\varphi \leftrightarrow \neg<!P>\neg\varphi \leftrightarrow (<!P>T \rightarrow <!P>\varphi)$.

epistemic form that held for *PAL* in its original version. Also, earlier schematic validities are typically going to fail now, such as the statement composition law

$$<!P><!Q>\varphi \leftrightarrow <!(P \wedge <!P>Q)>\varphi$$

The protocol need not allow any compound statements $!(P \wedge <!P>Q)$ at all. [267]

In between It is also of interest to compare the above *TPAL* axiom for knowledge $<!P>K_i\varphi \leftrightarrow (<!P>T \wedge K_i(<!P>T \rightarrow <!P>\varphi))$ with the stronger variant

$$<!P>K_i\varphi \quad \leftrightarrow \quad <!P>T \wedge K_i(P \rightarrow <!P>\varphi)$$

Setting $\varphi = T$, this implies $<!P>T \rightarrow K_i(P \rightarrow <!P>T)$, saying that agents know which statements are currently available for announcement. This is an intermediate requirement on protocols, in between the most general setting and the full protocol.

Theorem The logic *TPAL* is complete for the class of all *PAL* protocol models.

Proof Here is just a brief outline, to show the difference in labour required with our earlier fast completeness proofs: details are in van Benthem, Gerbrandy, Hoshi & Pacuit 2008. No reduction argument of the earlier kind works. Instead, we need to do a variant of a standard modal Henkin construction (cf. Blackburn, de Rijke & Venema 2000):

(a) Start from the canonical model of all *TPAL* maximally consistent sets, with epistemic accessibility defined as usual (that is, $w \sim_i v$ if for all $K_i\varphi \in w$: $\varphi \in v$) as the initial level of worlds w. [268]

Then create further levels through finite sequences of announcements with maximally consistent sets at each stage. Suppose that we have reached stage Σ in such a sequence:

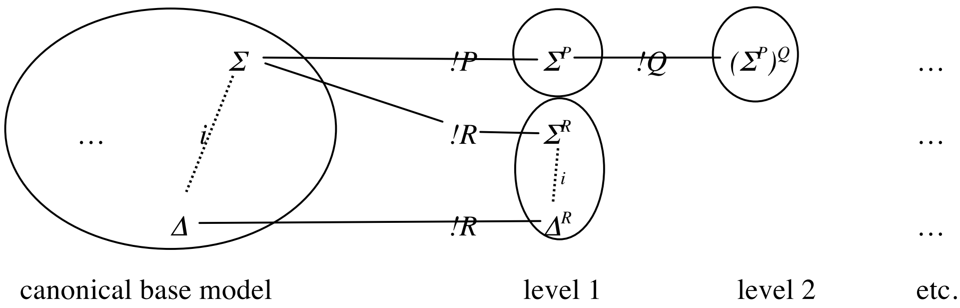
(b) The successors to the sequence are created as follows. Take any formula $<!P>T$ in Σ, and note that in the canonical model, by the Boolean axioms of *TPAL*, the set $\Sigma^P = \{\alpha \mid <!P>\alpha \in \Sigma\}$ is itself maximally consistent. Now, add a $!P$ move to the current sequence, and place the latter set right after it:

---

[267] *TPAL* still allows for a sort of normal form, since every formula is equivalent to a purely epistemic combination of proposition letters p plus procedural atoms $<!P>T$.

[268] As we shall see, this level already pre-encodes the further protocol through its true assertions involving (stacked) procedural formulas of the form $<!P>T$.

The successive levels contain epistemic models derived in this way from a previous stage by the same announcement action. Their ordering for worlds is copied from that of the preceding level. The following picture may help visualize this:



This construction yields a matching semantic forest model whose worlds are initial worlds followed by sequences of announcement events, where levels reflect construction stages. As usual, the heart of the matter is harmony between syntax and semantics:

Truth Lemma A formula φ belongs to the last set on a finite sequence h iff φ is true at that sequence h viewed as a history in the model matching the construction.

The proof reduces reasoning about formulas φ in worlds w at finite levels k to reasoning about stacked dynamic formulas $<!P_1>\dots<!P_k>\varphi$ in the initial canonical model, where P_1, ..., P_k reflects the unique construction of w. Repeatedly applying the recursion axiom $<!P>K_i\varphi \leftrightarrow (<!P>T \wedge K_i(<!P>T \rightarrow <!P>\varphi))$ then shows that the epistemic relations at the final level are inherited from the model at the initial stage, as they should.

The final stage of the construction is a routine verification that the ad-hoc model matching the construction is indeed a protocol model of the sort we have been considering. ■

Uniform protocols The construction gets slightly simpler with uniform protocols assigning the same set of sequences to each initial world. Their axiomatization works best with a universal modality U over all worlds, stipulating axioms for recursion over announcement actions, plus a strong form of common knowledge of the protocol:

(a) $<!P>U\varphi \leftrightarrow (<!P>T \wedge U(<!P>T \rightarrow <!P>\varphi))$

(b) $<!P>T \rightarrow U(P \rightarrow <!P>T)$

Decidability and complexity Analyzing the completeness argument in more detail shows

Theorem Validity in *TPAL* is decidable.

But the additional expressive power has a price:

Open Problem What is the precise computational complexity of *TPAL?*

There is no obvious reduction of *TPAL* to *PAL*, whose complexity was *Pspace*-complete (Chapter 3). So, how do *TPAL* and *PAL* compare as logics? Here is the only result so far:

Theorem There exists a faithful polynomial-time embedding for validity from *PAL* into the logic *TPAL* extended with the universal modality.

Again, the proof is in van Benthem, Gerbrandy, Hoshi & Pacuit 2008. The results of this section have been extended from *PAL* to *DEL* in Hoshi 2009 with syntax-induced event models and a construction that carefully tracks preconditions Pre_e.

11.7 Language extensions

The preceding results were for the epistemic base language, occasionally with a universal modality. Realistic scenarios for agency with protocols will bring in other operators:

Group modalities We have no complete logic for *TPAL* yet that includes *common knowledge* $C_G\varphi$. Also, several scenarios in Chapters 3, 12, 15 turn implicit factual knowledge of groups into common knowledge through iterated public announcement of what agents know to be true. This calls for languages with the *distributed knowledge* modality $D_G\varphi$. It seems quite plausible, that like *PAL* itself, *TPAL* can be extended to deal with such extensions, by adding principles like

$$<!P>C_G{}^{\psi}\varphi \leftrightarrow (<!P>T \wedge C_G{}^{<!P>\psi}<!P>\varphi),$$
$$<!P>D_G\varphi \leftrightarrow (<!P>T \wedge D_G[!P]\varphi).$$

Logics of protocols Concrete protocols for communication or experiment are a natural complement to our earlier local analysis of update steps in agency. The best examples so far are from epistemic temporal logics, cf. Fagin et al. 1995. One key result is the problem of the Byzantine Generals who must coordinate their attacks: if a communication channel is not known to be reliable, no new common knowledge arises from communication. [269] Such special protocols validate axioms beyond *TPAL*. For instance, consider the epistemic-temporal assertion that implicit knowledge eventually turns into common knowledge:

$$D_G\varphi \rightarrow F\, C_G\varphi$$

---

[269] Gerbrandy 1999A analyzes such arguments explicitly in dynamic-epistemic logics. Cf. also Roelofsen 2006 and van Eijck, Sietsma & Wang 2009 on channels and protocols in *DEL*.

This is not a general law of epistemic-temporal logic: its truth depends on the type of statement φ and the communication available in the channel. Chapters 3, 12 show that $D_G\varphi \rightarrow F\ C_G\varphi$ holds with simultaneous announcement, and even in some sequential settings. Gerbrandy 2008, Hoshi 2008 study laws for special protocols such as the Muddy Children. Chapter 15 in this book adds one more example: logics for protocols with iterated statements about knowledge and belief that arise in solution procedures for games.

11.8 Beliefs over time

One test for the epistemic temporal analysis in this chapter is how it generalizes to other attitudes that drive rational agency, such as belief. Epistemic-doxastic-temporal *DETL models* are branching event trees as before, with nodes in epistemic equivalence classes now also ordered by *plausibility relations* for agents (connected orders, for convenience). These tree or forest models interpret belief modalities at histories, in the style of Chapter 7. But they are very general, and as with knowledge, we ask which of them arise as traces of some systematic update scenario.

For this purpose, we take epistemic-doxastic models ***M*** and plausibility event models ***E*** to create products ***M x E*** whose plausibility relation obeys (cf. Baltag & Smets 2006):

Priority Rule $(s, e) \leq (t, f)$ iff $(s \leq t \wedge e \leq f) \vee e < f$

Van Benthem & Dégrémont 2008 extend the representation theorems of Sections 11.3, 11.5 to link belief updates to temporal forests. Let update evolution take place from some initial model along a sequence of plausibility event models in some uniform protocol. [270] Here are the relevant properties:

Fact The histories h, h', j, j' arising from iterated Priority Update satisfy the following two principles for any events e, f:

(a) whenever $je \leq j'f$, then $he \geq h'f$ implies $h \geq h'$ *Plausibility Revelation*

(b) whenever $je \leq j'f$, then $h \leq h'$ implies $he \leq h'f$ *Plausibility Propagation*

Representation Together, these express the revision policy in the Priority Rule: its bias toward the last-observed event, but also its conservativity with respect to previous worlds whenever possible given the former priority. Here is the key result:

[270] The cited reference also analyzes the case of pre-orders, and of state-dependent protocols.

Theorem A *DETL* model can be represented as the update evolution of an epistemic-doxastic model under a sequence of epistemic-plausibility updates iff it satisfies the structural conditions of Section 11.3, with Bisimulation Invariance now for epistemic-doxastic bisimulations, plus Plausibility Revelation and Propagation.

Proof The idea of the proof is as before. Given a *DETL*-model $\boldsymbol{H}$, we say that

$e \leq f$ in the epistemic plausibility model $\boldsymbol{E_k}$ if the events e, f occur at the same tree level k, and there are histories h, h' (of length $k\text{-}1$) with $he \leq_{\boldsymbol{H}} h'f$.

One can then check inductively, making crucial use of Priority Update plus Plausibility Revelation and Propagation in $\boldsymbol{H}$, that the given plausibility order in $\boldsymbol{H}$ matches the one computed by sequences of events in the update evolution stages

$$\boldsymbol{M_H \; x \; E_1 \; x \; \ldots \; x \; E_k}$$

starting from the epistemic plausibility model $\boldsymbol{M_H}$ at the bottom level of the tree. ■

Languages, logics, and long-term information scenarios Next, one can introduce doxastic temporal languages over our tree models, extending dynamic doxastic logic to a temporal setting. Van Benthem & Dégrémont 2008 uses the safe belief modality of Chapter 7 to state correspondences with agent properties. Dégrémont 2010 proves completeness for the logics, comparing them with the postulational analysis of Bonanno 2007. He also adds doxastic protocols, linking up with game theory and learning theory. Further doxastic protocols occur in Baltag & Smets 2009 on long-term belief update in groups via iterated soft announcements of the form $\Uparrow\varphi$ of Chapter 7. See Chapter 15 for some examples.

11.9 Conclusion

We have linked the dynamic logics of agency in earlier chapters to long-term temporal logics of knowledge and belief, in a precise technical sense. We found that this is a natural combination, where *DEL* describes fine-structure of widely used *ETL*-style branching tree models, and so we made a contribution to framework convergence in a crowded area. In the process, we also saw how ideas flow across frameworks, resulting in an interesting new version of *PAL* and *DEL* with protocols, incorporating genuine procedural information, and shedding the ‘fast reduction’ ideology of our earlier chapters where needed.

11.10 Further directions and open problems

This chapter has made a first connection, but it leaves many further desiderata:

Groups and preferences Given the social nature of agency (and interactive protocols), we need extensions to *group notions* like common and distributed knowledge and belief. Also, since evaluation works over time as well, we need temporal versions of the *preference logics* of Chapter 9. But our major concerns at the end of this chapter are the following:

Logics with explicit protocols Like *ETL*, our logics leave protocols implicit in models, with only muted impact in the language via assertions $<!P>T$ or $<\boldsymbol{E}, e>T$. The latter are just one-step local events for *PAL* or *DEL*, whereas we also want to bring out agents' long-term behaviour explicitly, like we did with strategies in Chapter 10. [271] We want to define this behaviour explicitly, and talk and reason about it. Van Benthem & Pacuit 2007 propose a version of epistemic *PDL* for this purpose, akin to the knowledge programs of Fagin et al. 1995. Van Eijck, Sietsma & Wang 2009 use *PDL*-definable protocols to explore concrete communication scenarios. Explicitness becomes particularly pressing when we realize that a protocol itself can be subject to dynamic change, just as games could change in Chapter 9. How to best model *protocol change*?

Joining forces with learning theory Temporal information mechanisms and long-term convergence to knowledge drive Formal Learning Theory (Kelly 1996, Hendricks 2003). What is the connection with our logics? *DEL* describes local learning of facts, but not long-term identification of the total history one is on. The latter suggests the branching temporal models of this chapter that are also the habitat of learning theory. One can learn long-term properties then, say about strategies of other players, depending on what one observes, and what one knows about the protocol. A logical perspective can distinguishes a whole range of learning goals, and our languages for epistemic-temporal models do just that:

> $FK\varphi$ or modalized variants express for suitable φ that there comes a stage where the agent will know that φ. Stronger variants are $FGK\varphi$ or $F(GK\varphi \vee GK\neg\varphi)$.

Even more distinctions arise by adding *beliefs* that fit the ambient hypotheses in learning scenarios. What such assertions say depends on one's epistemic-doxastic temporal models: versions with and without explicit histories both make sense. In fact, our epistemic-doxastic-temporal languages can express success principles for learning such as

[271] Strategies and protocols are not the same intuitively, but they are similar as logical objects.

$F(B\psi \rightarrow \psi)$, or $F(B\psi \rightarrow K\psi)$ saying that my beliefs will eventually be true, or even, that they will turn into knowledge. [272]

Chapters 3, 10, 12, 15 have examples of learning by the special mechanism of iterated announcement for the same assertion. The dynamics here can be updates with either hard or soft information (cf. Chapter 7, and Baltag & Smets 2009). Gierasimczuk 2009, Dégrémont 2010, de Jongh & Gierasimczuk 2009 find specific links between Formal Learning Theory, *PAL* and *DEL*. But much more remains to be done. Kelly 2002 suggests that learning theory is a natural extension of belief revision, separating bad policies from optimal ones. This seems attractive, also from a logical point of view.

Complexity of logics: further sources Perfect Recall created grid patterns in trees, and hence the logic of agents with perfect memory turned out to be complex. What about complexity effects of the doxastic agent properties in this chapter? Structurally, Plausibility Revelation for belief seems to act like Perfect Recall. And what about still more complex entangled properties with preference, such as Rationality of agents in games?

Complexity of agents versus complexity of logics Here is a major worry, resuming an earlier discussion in this chapter. Does the high complexity of epistemic-temporal logics really mean that the tasks performed by, say, agents with Perfect Recall are complex? Logic complexity lives at a meta-level, while task complexity lives at an object-level. The two differ: the theory of a simple activity can be complex. Object-level complexity of tasks for agents performing them might call for a major reworking of our logical analysis. [273]

Agents and automata It has been noted many times that *DEL* or *ETL* do not provide an explicit account of *agents* by themselves, their abstract models only show the 'epistemic traces' of agents in action. Though this seems a well-chosen abstraction level, ignoring details of implementation, Ramanujam 2008 makes a strong plea for linking *DEL* with *Automata Theory*, where agents are automata with concrete state spaces and memory structure. This seems a congenial and correct idea, but: it has to be carried out.

Outreach: agency, process algebra, and dynamical systems We conclude by repeating a few lines of outreach that started in Chapter 4. Our systems interface with temporal logics

[272] Learning theory distinguishes finite identification of a history and identification in the limit.

[273] Incidentally, high complexity for agent tasks may also be a good thing in rational agency, say, when a chairman finds it too hard to manipulate the agenda for her own purposes.

in AI such as the Situation Calculus (cf. van Benthem 2009B, van Ditmarsch, Herzig & de Lima 2007, Lakemeyer 2009), or *STIT*-based formalisms for agency and games as studied in Toulouse and Utrecht (cf. J. Broersen, A. Herzig & N. Troquard 2006, Balbiani, Herzig & Troquard 2007, Herzig & Lorini 2010). Another relevant line are epistemic versions of Alternating Temporal Logic (Van der Hoek & Wooldridge 2003, Van Otterloo 2005, Ågotnes, Goranko & Jamroga 2007). Computer science, too, has elegant calculi of process construction, such as Process Algebra (Bergstra, Ponse & Smolka eds. 2001) and Game Semantics (Abramsky & Jagadeesan 1992). With an explicit calculus of event models, *DEL* and *ETL* link such systems to modal languages describing properties of internal system states as a process unfolds. Is there a useful merge? An upcoming issue of the *Journal of Logic, Language and Information* (J. van Benthem & E. Pacuit, eds., spring 2010) on temporal logics of agency brings some of these systems together. [274]

11.11 Literature

Two key references on epistemic temporal logics in different guises are Fagin, Halpern, Moses & Vardi 1995, and Parikh & Ramanujam 2003, both reporting work going back to the 1980s. Other temporal frameworks for agency are *STIT* (Belnap, Perloff & Xu 2001), and the Situation Calculus (McCarthy 1963, Reiter 2001). Van Benthem & Pacuit 2006 give many further references, also to the computational tradition in temporal logic going back to Rabin's theorem (cf. Thomas 1992). For natural extensions of *DEL* to temporal languages, see Sack 2008, Hoshi & Yap 2009. The representation theorems in this chapter are from van Benthem 2001, van Benthem & Liu 2004, van Benthem & Dégrémont 2008. They are extended to partial observation, belief, and questions in Hoshi 2009, Dégrémont 2010, and van Benthem & Minica 2009. Gerbrandy 1999A introduces *DEL* protocols, van Benthem, Gerbrandy, Hoshi & Pacuit 2009 has the results reported here.

[274] And to repeat an issue from earlier chapters, now that we have a temporal logic in place, what is the connection between our discrete framework and continuous ones like evolutionary game theory or the mathematical theory of dynamical systems?