Computation as Conversation

Johan van Benthem

Institute for Logic, Language & Computation (ILLC), University of Amsterdam, Amsterdam, Netherlands
johan@science.uva.nl
and
Department of Philosophy, Stanford University Stanford, USA
johan@csli.stanford.edu

Summary. Against the backdrop of current research into 'logical dynamics' of information, we discuss two-way connections between conversation and computation, leading to a broader perspective on both.

1 Information flow for children, and logical dynamics

The Amsterdam Science Museum *NEMO* organizes regular Kids' Lectures on Science. [1] Imagine 60 children aged around 8 sitting in a small amphitheatre – with parents present in the wings, but not allowed to speak. Last February, it was my pleasure to give one on Logic. While preparing for the event, I got more and more worried. How does one talk logic to such an audience, without boring or upsetting them? Was there *anything* in common between children that age and the abstractions that drive one's university career? How to even start? My first question was this:

> **The Restaurant** In a restaurant, your Father has ordered Fish, your Mother ordered Vegetarian, and you have Meat. Out of the kitchen comes some new person with the three plates. What will happen? The children got excited, many little hands were raised, and one said: "He asks who has the Meat". "Sure enough", I said: "He asks, hears the answer, and puts the plate. What happens next?" Children said "He asks who has the Fish!" Then I asked once more what happens next? And now one could see the Light of Reason start shining in those little eyes. One girl shouted: "He does not ask!" Now, *that* is logic ...

[1] See http://www.nemo-amsterdam.nl/.

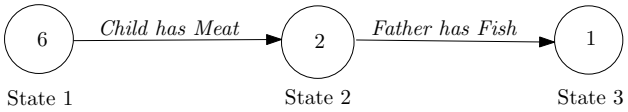
After that, we played a long string of scenarios, including card games, Master Mind, and Sudoku, and we discussed what best questions to ask and conclusions to draw. [2] In my view, the Restaurant is about the simplest realistic logical scenario. Several basic informational actions take place intertwined: questions, answers, and inferences, and the setting crucially involves more than one agent. Moreover, successive speech acts can be analyzed for their informational content once they have taken place, but they can also be planned beforehand: what best to ask, how best to answer? The program of 'Logical Dynamics' (van Benthem 1996) is about identifying and analyzing such scenarios, moving, in particular, the information-carrying events into the logical systems themselves. And once we take that view, we need a congenial account of *computation*. What happens during a conversation is that information states of children – singly, and in groups – change over time, in a systematic way triggered by various communicative events. In this universe of states and possible transitions between them, the long experience of computer scientists in modeling computation becomes relevant, from Turing's first 'single-minded' computers to dealing with the multi-agent Internet. Please note that this is not a matter of computational 'implementation', the subservient stance some computer scientists assume vis-a-vis other academic disciplines. We care rather about fundamental ideas, and the general cultural contribution of Informatics.

This paper is largely a discussion of known results and what they mean or suggest in a broader setting. Proofs and further details are found in the cited literature.

2 Multi-agent information models and epistemic logic

The first step in modeling conversation is a good notion of state, and hence the 'static component' of the total enterprise. For simple scenarios like the above, a logical apparatus exists, viz. *epistemic logic* (Hintikka 1962, Fagin et al. 1995). In the Restaurant scenario, the initial information state for the waiter from the kitchen had 6 possible arrangements for the three dishes over the three of us. As far as the new waiter is concerned, all are options, and he only 'knows' what is true in *all* of them. The new information that I have the Meat reduces this uncertainty to only 2: 'fish-vegetarian' or 'vegetarian-fish' for my father and mother. Either way, the waiter now knows that I have the Meat. Then hearing that father has the Fish reduces this to one single option: the waiter has complete information about the correct placement of the dishes, and does not need to ask any further question – even though he may still have to perform an inference to make this vivid to himself:

[2] The program included a Magic session with a card trick that failed to defy Logic in the end – plus a non-scheduled case of *crying*, a less common speech act in Academia. But that is another story.



Epistemic logic: language and models Here is some basic epistemic logic, as far as needed here. The syntax has a classical propositional base with added modal operators $K_i\phi$ ('i *knows that* ϕ') and $C_G\phi$ ('ϕ is *common knowledge* in group G'):

$$p \mid \neg\phi \mid \phi \vee \psi \mid K_i\phi \mid C_G\phi.$$

The states of our informational processes are *models* for this language, i.e., triples $\boldsymbol{M} = (W, \{\sim_i \mid i \in G\}, V)$ where W is a set of worlds, the $\sim_i$ are binary accessibility relations between worlds which agent i cannot distinguish as viable candidates for the real situation [3], and V is a propositional valuation. The fundamental epistemic truth condition for knowledge of an agent is then as follows:

$$\boldsymbol{M}, s \models K_i\phi \text{ iff for all } t \text{ with } s \sim_i t: \ \boldsymbol{M}, t \models \phi.$$

This language can define an existential dual of knowledge $\neg K_j \neg\phi$ (or $\langle j\rangle\phi$): agent j considers it *possible that* ϕ, plus other useful expressions such as $K_j\phi \vee K_j\neg\phi$: agent j knows *whether* ϕ. In particular, *multi-agent interaction* is a crucial feature. E.g., in asking a 'normal' question, a questioner $\boldsymbol{Q}$ conveys he does not know if ϕ: $\neg K_{\boldsymbol{Q}}\phi \wedge \neg K_{\boldsymbol{Q}}\neg\phi$. Moreover, usually he also thinks that the addressee $\boldsymbol{A}$ might know, which can be stated as an iterated two-agent assertion $\langle \boldsymbol{Q}\rangle(K_{\boldsymbol{A}}\phi \vee K_{\boldsymbol{A}}\neg\phi)$.

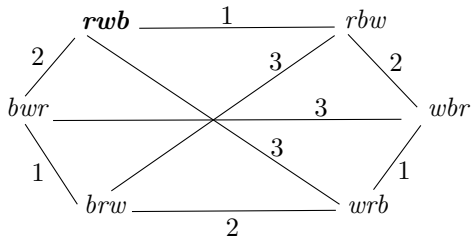
State transitions: information flow and model update Levels of knowledge about others occurred in the second scenario that was played with the children in *NEMO*:

> ***The Cards*** Three cards were given to three volunteers who stepped up: ***1*** got Red, ***2*** White, and ***3*** Blue. Each child could see its own card, but not those of the others (I was circling my little volunteers to make sure). Child ***2*** was allowed one question, and she asked ***1***: "Do you have the blue card". ***1*** answered truthfully: "No". Which child figured out what in this process?

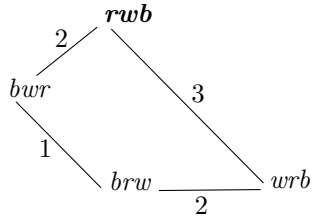
I asked beforehand, and all said they knew nothing. I asked again right after the question, and now Child ***1*** said he knew the cards. His reasoning, as whispered to me: "She would not have asked if she had the blue card herself. So, ***3*** has it." After the answer was given, child ***1*** and ***2*** said they knew the cards, and ***3*** still did not. But (with a little help) ***3*** did understand why the others knew the cards.

All this can be analyzed in words, but here is how things would look in an epistemic state transition framework. The initial situation again has 6 options, and the uncertainty lines indicate what players hold possible from where they are:

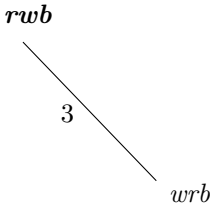
[3] One often takes these relations to be equivalence relations – but this is optional.



2's question, seen as informative [4], *eliminates* all worlds with second position 'b':



We see at once that, in the real world ***rwb***, ***1*** has no uncertainty line going out, and hence he knows the cards there. (We also see that ***3*** knows this, as it happens at both *rwb* and *wrb*.) Next, ***1***'s answer eliminates all worlds with first position 'b':



This reflects the final situation of the children.

Group knowledge Once again, multi-agent interaction is crucial. Indeed, the children even achieve a new level of knowledge that is sui generis, viz. *common knowledge*: in addition to what they know about the facts of the situation, they also know that the others know, and so on, up to any iteration. Common knowledge occurs in philosophy, linguistics, and economics as a prerequisite for coordinated action. Technically, this new notion is defined as follows over our models:

$$\begin{array}{ll} \boldsymbol{M}, s \models C_G\phi \ \text{ iff } & \text{for all } t \text{ that are reachable from } s \text{ by some} \\ & \text{finite sequence of } \sim_i \text{ steps } (i \in G)\text{: } \boldsymbol{M}, t \models \phi. \end{array}$$

This multi-agent view may seem far from standard logic and computation where single agents draw inferences or make calculation steps. But real argumentation is an interactive process, and even in the heartland of computation, very early on, Turing emphasized the crucial social character of using computers and learning. [5]

[4] Taking questions in this innocent way need not be sensible in the setting of *real games*!

[5] Cf. (Turing 1950). Wilfried Sieg explained to me how Turing emphasized social learning.

Belief and other attitudes of agents Knowledge is just one informational attitude of agents. One can also model *beliefs*, *probabilities*, and so on, using a broader variety of accessibility relations. A simple epistemic structure suffices for our aims, but we will mention less simplistic versions with agents' beliefs occasionally.

Summarizing then, our initial *NEMO* example is not 'child's play'. Conversational scenarios are a basic human ability involving sophisticated interactive knowledge that needs to be understood in depth. And thus, they provide a rich subject of study for Informatics, where logical and computational notions make good sense.

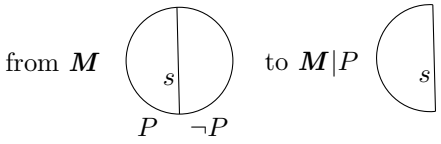
3 Conversation as computation: update actions

Communicative events range from simple public statements to complex private ones: recall my whispered conversation with child ***1***. And much more subtle scenarios exist in our lives. To move this inside our logic, we need an explicit account of relevant actions and their effects. Here a powerful metaphor comes into play:

Conversation is Computation!

Conversation is really an interactive form of computation, much as present-day computational systems have many agents engaged in a wide variety of tasks. Technically, then, conversational processes, and communication in general, may be modeled using existing systems from the computational tradition. In this paper, we will focus mainly on *dynamic logic*, originally developed as a logical account of programs and their effects (Pratt 1976), which has gradually evolved into a general theory of action. We start with the simplest mechanism of information flow.

Public announcement as world elimination Public announcements of true propositions P change the current situation as follows. For any model $\boldsymbol{M}$, world s, and formula P true at s, $(\boldsymbol{M} \mid P, s)$ (***M*** *relativized to* P *at* s) is the submodel of $\boldsymbol{M}$ whose domain is the set $\{t \in \boldsymbol{M} \mid \boldsymbol{M}, t \models P\}$. In a picture, one goes



Crucially, truth values of formulas may change in such an update step: most notably, because agents who did not know that P now do after the announcement. This truth value change can be quite subtle over time, including even cases where statements make themselves false. [6] One needs logics to keep this all straight.

Product update with event models Whispering is public announcement in a subgroup of a larger group, but it is only partially observable to the others. Hiding,

[6] Truly announce P ="You do not know that p, but it is really true" – and P becomes false.

secrets, and limited observation are ubiquitous in everyday communication. Consider your *email*. The epistemic-dynamic role of *cc* is public announcement. But the more sophisticated button *bcc* achieves *partial* announcement which can even mislead other participants. More complex scenarios arise in computer security, and in the arena of *games*, which are often designed to manipulate information flow. Partial observation of events may be analyzed as the following construction for changing models (Baltag et al. 1998). Scenarios where information flows in different ways for different agents can be represented in

$$\textit{Event models} \quad \boldsymbol{A} = (E, \{\sim_i \mid i \in G\}, \{PRE_e \mid e \in E\}).$$

Here E collects all relevant events. The uncertainty relations $\sim_i$ encode which events agents cannot distinguish. E.g., when the children checked their cards, the girl with the white card could not tell '***1***'s seeing red' from '***1***'s seeing blue'. Now, information flow occurs because events e have *preconditions* PRE_e for their occurrence (say, my having a red card, not knowing the answer to my question, etc.). When you observe an event, you learn that something must have been the case for this to happen.

The following *Update Rule* encodes the resulting mechanism of information flow:

> For any epistemic model $(\boldsymbol{M}, s)$ and event model $(\boldsymbol{A}, e)$, the *product model* $(\boldsymbol{M} \times \boldsymbol{A}, (s, e))$ has a distinguished new world (s, e), and then
>
> (a) a domain $\{(s, e) \mid s$ a world in $\boldsymbol{M}$, e an event in $\boldsymbol{A}$, $(\boldsymbol{M}, s) \models PRE_e\}$,
>
> (b) accessibility relations $(s, e) \sim_i (t, f)$ iff *both* $s \sim_i t$ *and* $e \sim_i f$,
>
> (c) the valuation for atomic formulas p at (s, e) is that for s in $\boldsymbol{M}$. [7]

Product update models a wide variety of information scenarios. And the universe of models with product update $\boldsymbol{M} \times \boldsymbol{A}$ has a rich logical and computational structure. [8]

Belief and other dynamic phenomena Knowledge was just one feature in information flow. If we also model agents' beliefs and expectations, product update can describe events affecting belief, including *misleading* actions, leading to false beliefs. Moreover, we need not just record *information update*. We can also model *belief revision*, a more agent-dependent phenomenon, which can depend on very different 'policies' for different types of agent, more conservative or more radical. [9]

[7] This stipulation of 'inertia' basically says that physical facts do not change under communication. This constraint can easily be lifted to let the system deal with genuine non-epistemic world change.

[8] Unlike with world elimination, epistemic product models can now get *larger* under update. But there is a counteracting force to this growth in complexity, as later models may be *bisimilar* with earlier ones, making the iterated epistemic long-term process cycle (van Benthem 2006C).

[9] Different policies even multiply when we define updates for further relevant phenomena in communication and interaction, such as changes in preferences or goals (van Benthem & Liu 2005).

4 Dynamic-epistemic logics of informative events

Given all these interesting actions that transform epistemic models, we want to study them explicitly. Now, keeping track of truth value changes for epistemic assertions can be as tricky as finding out what a particular program achieves over time. Thus, it is useful to keep track of both the statics and the dynamics in one logical calculus. Relevant frameworks from the computational literature include temporal logic, process algebra, or linear logic. Here, we choose *dynamic logic* (Kozen et al. 2000) with its two levels of expressions π for programs, and propositions ϕ describing successive states produced by these. The main operator of the language is

$[\pi]\phi$: "after any successful execution of π, ϕ holds in the resulting state".

This language stays close to that of modal logic, the lingua franca of much of computational logic, and it treats dynamic processes as being equivalent up to *bisimulation*, probably the most widely used notion of process equivalence. Still, this section is not meant as propaganda for any approach, but as a demonstration how computational logic of conversation and much more is entirely feasible.

Dynamic epistemic logic of public announcement The language of *public announcement logic PAL* is the epistemic language with added action expressions:

Formulas	$P:$	$p \mid \neg\phi \mid \phi \vee \psi \mid K_i\phi \mid C_G\phi \mid [A]\phi$
Action expressions	$A:$	$P!$

Here, treating announcements as actions, and having them explicitly inside modalities of the language comes from dynamic logic. The semantics is this:

$$\boldsymbol{M}, s \models [P!]\phi \quad \text{iff} \quad \textit{if } \boldsymbol{M}, s \models P, \textit{ then } \boldsymbol{M} \mid P, s \models \phi.$$

There is a complete calculus of information flow under public announcement – i.e., a complete logic of basic communication (Plaza 1989, Gerbrandy 1999):

Theorem. *PAL* without common knowledge is axiomatized completely by the usual laws of epistemic logic plus the following *reduction axioms*:

$$\begin{array}{ll} [P!]q \leftrightarrow (P \rightarrow q) & \text{for atomic facts } q \\ [P!]\neg\phi \leftrightarrow (P \rightarrow \neg[P!]\phi) & \\ [P!](\phi \wedge \psi) \leftrightarrow ([P!]\phi \wedge [P!]\psi) & \\ [P!]K_i\phi \leftrightarrow (P \rightarrow K_i[P!]\phi). & \end{array}$$

Methodology These axioms describe conversation in an elegant style, analyzing effects of assertions in a compositional way by recursion on the 'postconditions' behind the dynamic modalities. Thus, they reduce every formula of our dynamic-epistemic language eventually to a formula in the static epistemic language (cf. the

'regression procedure' of (Reiter 2001)). In terms of the logic, the reduction procedure shows that *PAL* is *decidable*, since the static epistemic base logic is decidable. [10]

This method of 'dynamification' applies to a wide range of informational events. First, choose a static language with models that represent information states for groups of agents. Next analyze the relevant informational events as update models changing the static ones. These updates are then described explicitly in a dynamic extension of the language, which can also state effects of events using propositions that hold after their occurrence. The resulting logics have a two-tier set-up:

At the static level, one gets a complete axiom system for one's chosen models. The computational analysis then adds a set of dynamic *reduction axioms* for effects of events. Thus every formula is equivalent to a static one – and hence, if the static base logic is decidable, so is its dynamic extension. In principle, this modular dynamic epistemic design is independent from specific properties of the static models. E.g., the *PAL* axioms do not depend on assumptions about epistemic accessibility relations. Its completeness theorem holds just as well if the static models are arbitrary, validating the minimal modal logic K as some minimal logic of *belief*.

Technical issues Sometimes, treating conversation as computation changes our ideas about an underlying static system. E.g., the completeness theorem for *PAL* omits *common knowledge* after announcements. To get a reduction axiom for formulas $[P!]C_G\phi$, one must enrich epistemic logic beyond its standard version, cf. (van Benthem et al. 2005). *Conditional common knowledge* $C_G(P, \phi)$ says that ϕ is true in all worlds reachable via some finite path of accessibilities running entirely through worlds with P. Then we get the valid reduction law: $[P!]C_G\phi \leftrightarrow C_G(P, [P!]\phi)$. Conditional common knowledge is not definable in the basic epistemic language – but it is bisimulation-invariant, and completeness proofs are easily generalized. [11] There is an analogy here with *conditional* assertions $\phi \Rightarrow \psi$ in belief revision, which state what we would believe were the antecedent to be considered (van Benthem 2006A). *PAL* has a modal bisimulation-based model theory, with many interesting issues of expressive power and computational complexity. [12]

General dynamic epistemic logic More general product update for communicative and observational scenarios can also be dealt with in this dynamic logic format. The language of *dynamic-epistemic logic* (*DEL*) has the following syntax:

$$p \mid \neg\phi \mid \phi \vee \psi \mid K_i\phi \mid C_G\phi \mid [\boldsymbol{A}, e]\phi :$$

[10] This reduction does not settle computational *complexity*: basic epistemic logic is ***Pspace***-complete, but translation via the axioms may increase the length of formulas exponentially. Cf. Section 6.

[11] Indeed, *PAL* with conditional common knowledge is axiomatized completely by adding just one more valid reduction law $[P!]C_G(\phi, \psi) \leftrightarrow C_G(P \wedge [P!]\phi, [P!]\psi)$.

[12] Cf. (van Benthem 2006D) for a survey of many open problems in this area.

with $(\boldsymbol{A}, e)$ any event model with actual event e. The semantic key clause is

$$\boldsymbol{M}, s \models [\boldsymbol{A}, e]\phi \qquad \text{iff} \qquad \boldsymbol{M} \times \boldsymbol{A}, (s, e) \models \phi.$$

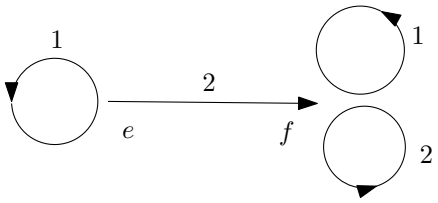
(Baltag et al. 1998) then showed completeness in this wider setting:

Theorem. *DEL* is effectively axiomatizable and decidable.

The key reduction axiom is the one extending that for public announcement:

$$[\boldsymbol{A}, e]K_i\phi \leftrightarrow PRE_e \rightarrow \wedge\{K_i[\boldsymbol{A}, f]\phi)) \mid f \sim_i e \text{ in } \boldsymbol{A}\}.$$

Further challenges Again consider *common knowledge* or *belief*. Just try to figure out what common beliefs hold in the following email scenario. Agent *1* sent message e, but in such a way that the other agent *2* believes that message f was sent:



(van Benthem et al. 2005) extends *DEL* to a logic *LCC* using ideas from dynamic logic and μ-calculus to get complete sets of axioms for such scenarios. [13]

Dynamic logics for belief revision and preference change The above format also provides complete logics for events of *belief revision*, and even more general *preference change*. These involve conditional beliefs, and compositional axioms for changes in them after 'hard facts' such as public announcements $P!$, or 'soft facts': weaker triggers for belief revision $\Uparrow P$ which may be overridden later on. [14]

Model change and other dynamic frameworks The general idea behind update mechanisms for knowledge or belief is *definable model change*. One selects or even creates new individual objects (the worlds) out of old ones, and then redefines the relevant relations between them. There are other systems than dynamic logic in the computational literature with a similar flavour. E.g., *process algebra* is a family of calculi for constructing new processes out of given ones. Indeed, our product update

[13] Another relevant issue is the 'view of agents' in product update. They satisfy *Perfect Memory*, and *No Miracles*: learning only occurs through observation of suitable events. (van Benthem & Liu 2004) shows that this is complete – (Liu 2006) looks at much greater *diversity* of epistemic agents.

[14] Just to show the format, here are two reduction axioms for new beliefs after hard and soft triggers: $[P!]B_i(\phi \mid \psi) \leftrightarrow P \rightarrow B_i([P!]\phi \mid P \wedge [P!]\psi)$, $[\Uparrow P]B(\phi \mid \psi) \leftrightarrow (\boldsymbol{E}(P \wedge [\Uparrow P]\psi) \wedge B([\Uparrow P]\phi \mid P \wedge [\Uparrow P]\psi)) \vee B([\Uparrow P]\phi \mid [\Uparrow P]\psi)$. Here $\boldsymbol{E}$ is the existential modality "in at least one world". For details, cf. (van Benthem 2006A, Baltag & Smets 2006).

$\boldsymbol{M} \times \boldsymbol{A}$ respects bisimulation in the standard process-algebraic sense. In our view, *DEL* is a nice calculus of model change intermediate between dynamic logic and process algebra, which combines an 'external language' for defining processes with an 'internal language' describing properties of states within these processes. Merging major computational process paradigms may be a good idea in general.

5 Program structures in conversation

Genuine computation involves control over long *sequences* of actions. Likewise, conversation involves many assertions governed by *program constructions*. When talking with our dean, we first praise the current state of the Faculty of Science, and then ask for funding. And what we say depends on his current state. We commiserate when he looks troubled, and joke when he looks happy. Finally, there is an iterative process of 'flattery'. We keep saying nice things until his brow clears, and the right moment for our funding request has come. Thus, conversation involves all the basic operations from sequential programming: (a) sequential composition ;, (b) guarded choice *IF* ... *THEN* ... *ELSE* ..., (c) guarded iterations *WHILE* ... *DO* ...

A much-quoted concrete example is the puzzle of the 'Muddy Children':

> *After playing outside, two of three children have mud on their foreheads. They all see the others, but not themselves, so they do not know their own status. Now their Father comes and says: "At least one of you is dirty". He then asks: "Does anyone know if he is dirty?" The children answer truthfully. As this question–answer episode repeats, what will happen?*

Nobody knows in the first round. Next, the muddy children argue as follows. "If I were clean, the one dirty child I see would have seen only clean kids, and so she would have known that she was dirty. But she did not. So I must be dirty, too!" Thus both know their status in the second round. The third child knows it is clean one round later. The puzzle easily extends to more clean and dirty children. [15]

Clearly, all three preceding program constructions occur here: sequential assertion, guarded action (children must respond differently depending on what they know), and iteration: the process repeats until common knowledge is achieved.

Adding full dynamic logic To analyze complex conversations, *PAL* or *DEL* must be extended with *propositional dynamic logic PDL*, which has a test operation $?\phi$ on propositions, plus the three regular operations of sequential composition ;, choice $\cup$, and iteration *. We display the major valid axioms here [16]:

---

[15] For a concrete update sequence describing this scenario, cf. (Fagin et al. 1995, van Benthem 2006C).

[16] The axioms for π^* say that a universal modality $[\pi^*]$ is a greatest fixed-point operator.

$$(a) \quad [\phi?]\psi \leftrightarrow (\phi \rightarrow \psi),$$
$$(b) \quad [\pi_1;\pi_2]\phi \leftrightarrow [\pi_1][\pi_2]\phi,$$
$$(c) \quad [\pi_1 \cup \pi_2]\phi \leftrightarrow ([\pi_1]\phi \wedge [\pi_2]\phi),$$
$$(d) \quad [\pi^*]\phi \leftrightarrow (\phi \wedge [\pi][\pi^*]\phi),$$

and

$$(e) \quad (\phi \wedge [\pi^*](\phi \rightarrow [\pi]\phi)) \rightarrow [\pi*]\phi.$$

These axioms work by recursion on the first argument of our modal statements $[\pi]\phi$, rather than the second. It is known that *PDL* as a system of arbitrary actions is completely axiomatized by these principles – and indeed, it is decidable. [17]

Further constructions But conversation also involves further program operations. It is crucial to the Muddy Children puzzle that the children answer *simultaneously*. This is parallel composition of individual actions, as in distributed computing and process algebra. *PAL* treats simultaneous speech as announcing a conjunction, and thus $(\phi \wedge \psi)!$ is a simple analogue of a parallel composition $\phi! \,||\, \psi!$. [18]

Temporal logic All this eventually embeds dynamic epistemic logics into broader *epistemic temporal logics* over branching trees of events (Fagin et al. 1995) and (Parikh & Ramanujam 2003). The latter links up with another process view in computer science, viz. temporal logics in the style of Pnueli, Clarke, and others. Cf. (van Benthem & Pacuit 2006) for connections with our current setting.

6 Complexity of logical tasks

Computation involves a balance between representation and processing of data, and so do logical systems. While dynamic epistemic logics provide a rich account of effects of events that carry information, their expressive power has a price in terms of *computational complexity*. Indeed, any logical system can be used for a variety of core tasks which all involve computational complexity.

Model checking We start with *model checking*, i.e., determining whether $\boldsymbol{M}, s \models \phi$ for a given model $\boldsymbol{M}$ and formula ϕ. For basic epistemic logic, this task is ***P-time*** in the size of formulas and models (Vardi 1997). In our conversational setting, model checking *DEL*-formulas corresponds to computing the effects of informational events in a given informational situation. (van Benthem et al. 2005) shows

[17] Combining *PDL* with epistemic logic into a richer version of *DEL* will involve recursions on both actions and postconditions. The precise nature of this joint approach remains to be understood.

[18] No explicit axiomatization is known yet for this parallel operator $||$ in *PAL* or *DEL*.

that model-checking complexity remains ***P-time*** for arbitrary formulas ϕ of *DEL*. [19] Thus, verifying the effects of a given conversational plan is an easy task.

Satisfiability But the more ambitious task is *conversation planning*: how to set up a setting in order to achieve certain desired effects? This can still be cast as a model checking problem when the epistemic 'space' is given beforehand (see below), but in general one asks for the existence of some information model satisfying some specified properties. This is the problem of *satisfiability* (*SAT*): when does a given formula have a model? The *SAT* problem for basic epistemic logic is ***Pspace***-complete. The axioms for *PAL* provide a *SAT* reduction to this system, but given the shape of the axioms, this might be exponential. (Lutz 2005) provides a better reduction which shows that *SAT* complexity for *PAL* remains ***Pspace***-complete. [20] While this might suggest that dynamifying a base logic does not affect complexity, further dynamic epistemic logics still have surprises in store. In particular, the above combination of *PAL* with the program operations of dynamic logic *PDL*, i.e., the combination of two systems each of which are decidable, leads to a surprise:

Theorem. (Miller & Moss 2005) *PAL* with *PDL* operations is *undecidable*. [21]

More concretely, designing puzzles like Muddy Children and solving conversation planning problems in them can be extremely hard!

Complexity of further tasks? Besides standard model checking and satisfiability, there may be other natural complexity issues for dynamic-epistemic logics. For instance, a set of admissible assertions, or a more general conversational protocol over some given initial model generates a model $\boldsymbol{M}$ with all possible trajectories for an informational process. In such a model, we can ask for conversational plans achieving intended effects; say in the form of *PDL* programs as above which are guaranteed to move from the initial state to some state satisfying some goal proposition ϕ. The resulting intermediate model-checking problems asks if there exists an executable *PDL* program π such that $[\pi]\phi$ holds at the current state s in the given model $\boldsymbol{M}$. This is not quite ordinary model-checking, but it is not full-fledged *SAT* either. Here is another variant issue. What actions are worth counting in our update setting? For instance, is there an analogue of the computational notion of *communication complexity* defined in (Yao 1979)? Finally, on the more empirical side, once partial observation of events is considered as in *DEL*, one expects intuitive complexity jumps from public to private announcement, or from speaking the truth to lying. But so far, not all relevant intuitions and empirical folk wisdom about such thresholds have been turned into precise mathematics yet.

Danger zones Many authors have explained (Halpern & Vardi 1989, Marx 2006, van Benthem & Blackburn 2006, van Benthem & Pacuit 2006) how modal logics practice the art of 'living dangerously' at the edge of undecidability. With expressive power *tree*-oriented, they are decidable guarded-quantifier formalisms. But when

[19] This complexity is *EXPTIME* for the full language with all *PDL* operations.
[20] The *SAT*-complexity for *DEL* probably remains the same.
[21] The proof uses *infinite* epistemic models: it is not known if it holds with just finite models.

dangerous patterns become definable, in particular two-dimensional *grids* with two confluent relations, they tend to become undecidable – and may even incur non-arithmetical complexity. In a dynamic epistemic setting, geometric confluence reflects commutativity laws for modalities (Halpern & Vardi 1989, van Benthem 2001) which may make logics undecidable – though the precise recipe for disaster is delicate. For knowledge and action, an equivalence between $K[e]\phi$ (knowing that an event e produces a certain result ϕ) and $[e]K\phi$ (an event e 's producing knowledge that ϕ) amounts to the semantic condition that agents have *perfect memory*. Thus, writing logics for well-endowed idealized agents can drive up complexity!

7 Reversing the direction: computation as conversation

We have amply shown by now how conversation can be viewed as computation, leading to interesting issues that can be studied by combining techniques from philosophical and computational logic. But this link also suggests an *inversion in perspective*. In particular, lower bound results concerning complexity often establish that some other problem of known complexity can be reduced to the current one. And though these reductions may be quite technical, usually, they convey a lot more useful information, often of a semantic nature – and hence they establish stronger analogous than mere 'equi-difficulty'. To see this more concretely, take our analysis of conversation as a form of computation. The simple point that we wish to make now is that complexity analysis, as available in known results, also allows us to view

Computation as Conversation!

Realizing computation as conversation High-complexity results are often taken to be bad news, as they say that some logical task is hard to perform. But the good news here is that, by the very same token, an interesting transfer happens: the logic manages to encode significant problems with mathematical content. For instance, consider the famous result that *SAT* in propositional logic is ***NP***-complete. Reversing the perspective, this result also means that solving just one basic logical task has *universal computational power* for a large class of problems encountered in practice. Moreover, the proof of ***NP***-completeness for propositional *SAT* even gives a simple *translation* from arbitrary computational tasks to logical ones. [22] The same reversal applies to other complexity classes. E.g., ***Pspace***-complete is the solution complexity for many natural games (Papadimitriou 1994, van Emde Boas 2002), and hence being able to solve *SAT* problems in our base logic, i.e., the ability to create consistent epistemic scenarios suffices for solving lots of games.

Now, in this same light, consider the above result from (Miller & Moss 2005). What they prove is essentially, that each *tiling problem* – and hence also, each significant

[22] A course in propositional logic is at the same time one in universal computation, *if* only you knew the key ...

problem about computability by Turing machines –can be effectively reduced to a *SAT* problem in *PAL* + *PDL*. I literally take this result to mean the following:

> *Conversation has Universal Computing Power* Any significant computational problem can be realized as one of conversation planning.

Even so, in this technical sense, 'computation as conversation' is mainly a metaphor. In what follows, I take one more step, which does not require us to 'take sides'.

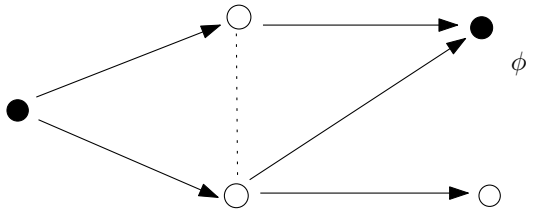
8 Merging computation and conversation

The real benefit of bringing together computation and conversation is not reduction of one to the other. It is creating a broader theory with interesting new questions. In particular, a theory of computation that absorbs ideas from conversation must incorporate the dynamics of *information flow* and *social interaction*. We will mainly discuss one way of doing this here. It starts from known algorithms, and then adds further structure. We proceed by a series of examples, as our aim is merely to show how many new questions can be asked at once in this setting, without established answers. At the end, we note a few more general trends.

Epistemizing algorithms Consider the basic computational issue of Graph Reachability (*GR*). Given a graph $\boldsymbol{G}$ with distinguished points x, y, is there a chain of directed arrows in $\boldsymbol{G}$ leading from x to y? This task can be solved in ***Ptime*** in the size of the graph: there are fast quadratic-time algorithms finding a path (Papadimitriou 1994). The same analysis holds for the task of reachability of some point in $\boldsymbol{G}$ satisfying a general goal condition ϕ. *GR* models search problems in general, and the solution algorithm performs two closely related tasks: determining whether a route exists at all, and giving us an actual *plan* to get from x to y. We consider various ways of introducing knowledge and information.

Knowing you have made it Suppose you are an agent trying to reach a goal region ϕ, but with only limited observation of the graph in which you are moving. In particular, you need not know, at any point x, at which precise location you are. Thus, the graph $\boldsymbol{G}$ is now a model $(G, R, \sim)$ with accessibility arrows, but also epistemic *uncertainty links* between nodes. A first epistemization of *GR* merely asks for the existence of some plan that will lead you to a point *which you know to be in the goal region* ϕ. (Brafman et al. 1993) analyzes a practical setting for this, with a robot whose sensors do not tell her exactly where she is standing. In this case, it seems reasonable to add a *test* to the task, inspecting current nodes to see if we are definitely in the goal region: $K\phi$. Given the ***P***-time complexity of model checking for modal-epistemic languages, the new search task remains ***P***-time.

Having a reliable plan In this setting, further issues arise. What about the plan itself? If we are to trust it, should not we require that we *know it to be successful*? Consider the following graph, with an agent at the root trying to reach a ϕ-point:



The dotted line indicates the agent cannot tell the two intermediate positions apart. A plan which achieves the goal is *Up ; Across*. But after following one part of this, the agent no longer knows where he is, and in particular, whether moving *Across* will reach the ϕ-point, or rather moving *Up*. Let us first formulate the requirement. Suppose for simplicity that a plan is just a finite sequence $\boldsymbol{a}$ of arrows. We may then require initial knowledge that this will work: $K[\boldsymbol{a}]K\phi$. But this is just at the start: we may also want to be sure at all intermediate stages that the remainder of the plan will work. This would require truth of all formulas

$$[\boldsymbol{a}_1]K[\boldsymbol{a}_2]K\phi, \qquad \text{where } \boldsymbol{a} = \boldsymbol{a}_1;\boldsymbol{a}_2$$ [23]

The existence of such a 'transparent' plan can still be checked in ***Ptime***, since the number and size of the relevant assertions only increases polynomially. But this quickly gets more complex with plans defined by more complex *PDL* programs. It is not obvious to how to even *define* the right notion of epistemic reliability, and we suspect that it may lead to new languages beyond *DEL* and *PDL*. [24]

Different types of agent But there is more to the epistemic setting in the preceding example. Note that the agent in the graph has *forgotten* her first action: otherwise, she could not be uncertain between the two nodes in the middle. Our earlier *DEL*-style agents with *Perfect Recall* would not be in this mess, as they can only have uncertainties about what other agents did. And the earlier-mentioned commutation law $K[a]\phi \rightarrow [a]K\phi$ which holds for them will automatically derive intermediate knowledge from initial knowledge $K[\boldsymbol{a}]K\phi$. But there are many kinds of epistemic agent: with perfect recall, with finite memory bounds, etc. [25] Thus, epistemized algorithms naturally go together with questions about what sorts of agents are to be running them – and the complexity of these tasks-for-agents can vary accordingly.

Epistemic plans But also, in an epistemic setting, the notion of a plan itself requires further thought. A plan is a sort of program which can react to circumstances, via conditional instructions such as *IF* α *THEN do* a *ELSE* b. The usual understanding of the test condition α is that one finds out if it holds, and then choose an action accordingly. But for this to work, the agent has to be able to *perform* that test! Say,

[23] If an agent has Perfect Recall, $K[a]\phi$ implies $[a]K\phi$, and the initial formula implies all the others. But for bounded agents, our distinction makes sense.

[24] As in (van Benthem 2001), some version of the epistemic μ-calculus may be needed, at least for reliable strategies of players in a game. These are related to the 'uniform strategies' of game theory.

[25] No standard taxonomy exists yet of this diversity: cf. (Liu 2006) for a first overview.

we ask a computer to check the current value of some variable, or a burglar to check whether the safe has a Yale lock or some inferior brand. But in the above graph, the plan "*IF* you went *Up*, *THEN* move *Across ELSE* move *Up*", though correct as an instruction for reaching the goal, is no use, as the agent has no way of deciding which alternative holds. There are two ways of dealing with this. One is to include knowledge into programs (Fagin et al. 1995). We make actions dependent on conditions like "the agent knows α" which can always be decided, provided agents have epistemic introspection. [26] Suitable epistemic programs are automatically transparent in the above sense (van Benthem 2001). The other option is to define a notion of '*executable plan*' in an epistemic model $\boldsymbol{M}$, making sure that agents can find out whether a test condition holds at any stage where this is needed. But so far, I have not found a definition for epistemic executability which satisfies me.

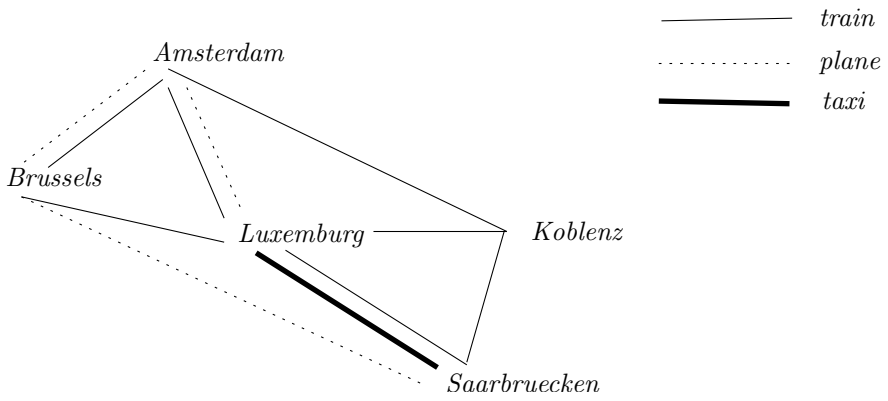
Dynamifying static logics: update actions Finding out if a proposition holds involves actions of communication or observation, and hence we move beyond epistemized static logics to dynamic ones. Then we could model the above test conditions α as explicit actions of *asking* whether α holds. This requires richer multi-agent models, though, where one can query other agents, or perhaps Nature, about certain things. We will not pursue this topic here, but the logic *DEL* in this paper is a show case of 'dynamification'. Thus, it should be well-suited for analyzing dynamified algorithms – and so are epistemic variants of *PDL* or the μ-calculus. [27]

Multi agent scenarios and interactive games Several epistemic scenarios in the preceding suggest adding more than one agent, moving from traditional lonely algorithmic tasks to more social ones. E.g., reaching a goal and knowing that you are there naturally comes with variants where *others* should *not* know where you are. Examples in the literature include the 'Moscow Puzzle' (van Ditmarsch 2002), where people have to tell each other the cards which they have without letting a third party present know the solution. Card games, or the earlier-mentioned use of email, provide many further examples. This social interactive perspective comes out even more in the setting of *games*, and interaction between different players. Indeed, games have been proposed as a very general model of computation (Abramsky 2006) and new logical questions about them abound (van Benthem 2005B).

Reachability and sabotage Turning algorithms into games involves the 'prying apart' of existing algorithms into games with different roles for different agents. Early examples are logic games in the style of Lorenzen, Ehrenfeucht, or Hintikka (cf. the survey in (van Benthem 1999)). A more algorithmic example is in (van Benthem 2005A). Consider again Graph Reachability. The following picture gives a travel network between two European capitals of logic and computation:

[26] Interestingly, some heuristic algorithms in (Gigerenzer & Todd 1999) have this flavour.

[27] An extreme case of this setting are pure information games, where all moves are actions of asking questions and giving answers, and players go for goals like 'being the first to know'.



It is easy to plan trips either way. But what if the transportation system breaks down, and a malevolent Demon starts canceling connections, anywhere in the network? At every stage of our trip, let the demon first take out one connection. Now we have a two-player *sabotage game*, and the question is who can win it where. Some simple reasoning will show that, from Saarbruecken, a German colleague still has a winning strategy. But the Dutch situation is less rosy: Demon has the winning strategy.

This example suggests a general transformation for any algorithmic task. It becomes a *sabotaged* one when cast as a game with obstructing players. This raises several new questions, e.g., about logical languages describing these games, and players' plans (*strategies*) in them. In particular: how does the *computational complexity* of the original task change when we need to solve the new game? For sabotaged Graph Reachability, it has been shown in (Rohde 2005) that this complexity moves up from low ***P***-time to ***Pspace***-completeness. That is, the problem now takes a polynomial amount of memory space, which makes it of the complexity of Go or Chess. [28]

Catch Me If You Can But there is no general rule predicting when a newly created game becomes more complex than its algorithmic ancestor. Again consider graphs, the setting par excellence for algorithmic tasks, but now with another game variant of GR. 'Obstruction' could also mean that some other player tries to catch me en route, making it impossible for me to continue. It is easy to cast this as a game, too:

> Starting from an initial position $(\boldsymbol{G}, x, y)$ with me located at x and you at y, I move first, then you, and so on. I win if I reach my goal region in some finite number of moves without meeting you. You win in all other cases. [29]

[28] Link cutting games also have other interesting interpretations. (van Benthem 2006B) has a variant dual to the above where a Teacher tries to trap a Student into reaching a certain state of knowledge.

[29] Thus, you win: if you catch me before I am in the goal region, if I get stuck, or if the game continues indefinitely. Other natural ways of casting these conditions would allow draws.

This game, too, is very natural, and it models a wide variety of realistic situations, such as warfare, or avoiding certain people at receptions. [30] But this time, the computational complexity stays lower. [31] Solving *Catch Me If You Can* still only takes ***Ptime*** in the size of the graph! This can be seen by the analysis of the analogous 'Cat & Mouse' game in (Greenlaw et al. 1991). [32] [33]

Adding knowledge and observation again In actual warfare, catching games naturally involves limited observation and partial knowledge. In such *games of imperfect information*, players need not be able to see where the others are, and solution complexity may go up to ***Pspace*** and beyond. (Sevenster 2006) is an extensive study of various epistemized algorithms in this setting, using connections with the '*IF* logic' of (Hintikka & Sandu 1997) to clarify their properties. In particular, he shows that the situation is delicate. E.g., consider that mild form of warfare called the game of 'Scotland Yard'. Here the invisible player who tries to avoid getting caught has to reveal her position after every k moves for some fixed k. But then, the game can be turned into one of perfect information by re-encoding players' moves making k-sequences of old moves into single steps. (van Benthem 2001, van Benthem 2005B) study many further aspects of merging *DEL* with game theory. [34]

Rephrasing the issues in game theory? From a genuine game-theoretic viewpoint, many further questions may become relevant, however. E.g., Sevenster's major complexity results are in the *IF* tradition of asking whether some player has a winning strategy even when hampered by lack of knowledge. But the most crucial feature of finite games of imperfect information, both mathematically and in practice, is the existence of something more delicate: *Nash equilibria in mixed strategies*, letting players choose moves with certain probabilities. Maybe it is the resulting *game values* that we should be after for gamified algorithms. Thus, gamification as generalized computation should also make us pause and think about most natural counterparts to the properties of algorithms when they were still pure.

This is just one of many issues when we take game structure seriously. Imperfect information games also invite explicit events of observation and communication (Osborne & Rubinstein 1994, van Benthem 1999, van Benthem 2001). Moreover, they fit naturally with the *parallel action* mentioned earlier, as much of game theory is about simultaneous choice of moves by players. And then: why two play-

[30] Fabius Maximus Cunctator tried to win a war by avoiding his enemy Hannibal throughout.

[31] The difference with the Sabotage game is that the graph remains fixed during the game.

[32] I owe this reference to Merlijn Sevenster, who also points out the finer complexity difference that Reachability is ***NL***-complete, while Cat & Mouse is ***Ptime***-complete.

[33] A direct argument is as follows. The game can be recast as a graph game over an extended graph with positions $(\boldsymbol{G}, x, y)$ counting players' moves as described while allowing you 'free moves' when I am caught or get stuck .Now we let you win if you can keep moving forever. It is known that graph games like this can be solved in ***Ptime***. One can see this as a modal model checking problem for formulas $<>^n T$ with n the graph size.

[34] (van Otterloo 2005, van Benthem 2007) study extensive games with explicit actions of announcing relevant facts or even players' intentions concerning their future moves.

ers, and not more? E.g., even inside the heartland of logic games, it has been proposed that argumentation, often cast as a tennis match, really needs a 'Proponent', an 'Opponent', and: a *Judge*. Thus, our view of algorithms in a social setting naturally merges computer science, logic, and game theory, with links running all across.

9 Toward a general theory: transformations and merges

Our discussion in the preceding section has been just a bunch of examples, trying to convey the pleasure of exploring an interactive epistemic viewpoint on computation. But it also suggests several more systematic topics.

Epistemizing logics One broad concern is the design of appropriate *logical languages* for these new structures. This might seem a simple matter of combining components like dynamic and epistemic logic, but it can be much more interesting. [35] Next, relevant tasks for these languages can fall into the cracks of the standard notions of complexity. E.g., natural planning problems seem intermediate between model checking and satisfiability. They ask, in a given model $\boldsymbol{M}$ with state s, whether some epistemic plan exists which takes us from s to the set of goal states. Thus, *epistemizing logics* is a non-trivial exercise, when done with a heart.

Epistemizing and gamifying algorithms Next, there is the issue of finding *general transformations* on algorithms behind the above examples. Instead of botany, one would want general results on what these do to the solution complexity of the original task. The dissertations (Rohde 2005, Sevenster 2006) were first steps.

A bit quixotically, what we are doing here can be seen as *dynamification* once more, but now at a meta-level. We have been using dynamic viewpoints to transform given problems in their original guise, and now, we are trying to make that process itself into an object of logical study. This is one way of seeing more unity in the diverse examples and logics that arise when 'computation and conversation' are mixed together. But there are other ways. In particular, promising *convergences* can be observed between various systems for describing 'computation and conversation', witness the comparison between dynamic epistemic logic, epistemic temporal logic, modal product logics, and other paradigms in (van Benthem & Pacuit 2006). More generally, one broad aim of theory construction in this arena is this:

Epistemized process theory Moving toward fundamental theories of computation, bringing in explicit considerations of observation and conversation suggests epistemic versions of existing process theories, such as Process Algebra. As the latter

[35] E.g., (van Benthem 1999) shows how even the issue of finding 'the epistemic version' of propositional dynamic logic is not at all simple, as *DEL* suggests a two-level approach, providing both *states* and *arrows* with uncertainty relations, giving us a range of options for matching logical languages.

includes an explicit account of 'communication channels', making the connection seems quite appropriate. [36] The same points apply to interaction and game semantics for computation. E.g., standard models for linear logic achieve non-determinacy by moving to infinite games. But non-determinacy reigns in simple finite games with imperfect information, suggesting epistemic versions of linear game semantics. Also, strategies in linear logic crucially involve switching across games, and using information about moves in one to make best moves in the other (Abramsky 2006): which is again well within our circle of ideas. [37] Of course, as we noted earlier, there is also the issue of how all this relates to existing *game theory*. Perhaps, the current contacts between logic, computer science, and game theory, may be viewed as preliminaries to a new theory with aspects of all three.

10 Conclusions

This paper fits in a broad current trend. Bringing together computation and broader information-based activities of conversation and communication is in the air, and it has been there for at least two decades. It may be seen with the epistemic analyses of communication protocols in (Fagin et al. 1995), with calculi of distributed computing like Milner's *CSP*, and of course, with modern theories of agents and intelligent information systems. We have tried to show here that this trend is more than a metaphor by pointing at concrete logics which deal with it, and at a sequence of interesting new issues which arise when we merge the two agendas systematically. To some, the resulting theory may look strange at first, as it combines hard-core computational logic with epistemic logics from the 'softer' philosophical tradition – something which may look even more outrageous when we add, not just knowledge, but also agents more ephemeral beliefs, and who knows, even their intentions and most intimate desires. Still, we think computation plus information update and belief revision is a perfectly viable marriage. It is rich in theory, and also, it fits very well with modern computation in societies of interacting agents. Indeed, recent research programs like 'social software' (Parikh 2002) even take this into activist mode, and propose, not just analyzing existing social procedures in this style, but even designing new better ones. In this, social software is like 'mechanism design' in game theory, but pursued by sophisticated computational techniques.

As a counter-point to such 'soft' social settings, it needs to be said that the Dynamic Turn advocated in this paper is also observable in hard-core areas like physics. Recent interfaces between computer science and quantum mechanics emphasize the dynamic interactions of observing agents with physical systems in operator-based Hilbert spaces. Accordingly, systems of dynamic logic and game semantics for linear

[36] (Dechesne & Wang) compares renderings of communication scenarios in *DEL* and Process Algebra.

[37] In recursion theory, a precursor is (Condon 1988) on Turing machines run by agents with limited observation – though for specialized complexity-theoretic purposes.

logics are crossing over from computation to the foundations of physics, as well as the practices of quantum computation. (Abramsky & Coecke 2004, Baltag & Smets 2004), and some entries in (Rahman et al. 2004) are samples of this trend. For what this might mean in a broader information theory, cf. (Abramsky 2006).

Another way of stating the main point of this paper is that computation is a pervasive and fundamental category across the sciences and humanities, provided we cast it in its proper generality, linking up with epistemic logics broadly construed. In one direction, our dynamic epistemic systems show how this introduces significant computational models into the study of what used to be thought of as preserves for linguists and philosophers. In the opposite direction, we can 'epistemize' and 'dynamify' existing logics and algorithms, to get interesting broader theories. Returning to our Introduction, it should be clear that this is much more than 'implementation' in an auxiliary sense, but rather a way of letting fundamental ideas from computer science play the central academic role which they so clearly deserve.

Despite all these grand perspectives, this paper was written by a logician, as biased – Heaven knows – as the next person. This may be a good place for a disclaimer. Despite the amount of space devoted to *dynamic epistemic logic* in this paper, we have used it mainly as a 'search-light system' for interesting phenomena, not as the final word on the structure and flow of information. Indeed, even from the viewpoint of the *NEMO* Restaurant, we have still missed crucial aspects of the children's activities! The waiter or the card players do not just update with new information, but also *infer* things already at their disposal. But valid conclusions from existing information do not change a current *DEL* information state. To describe this finer dynamics, further process structure is needed. [38] Likewise, our discussion of testing conditions for algorithms or games suggests that we have left out the dynamics of *evaluation* (van Benthem 1996). The logical core tasks of inference and model checking have their own dynamics, which goes beyond our framework here. Thus, even the logical foundations of information and computation remain wide open. [39]

Finally, what about *Computing in Europe*? I would like to believe that a broad stance on any subject matter, reflecting a certain erudition beyond one's immediate specialty is a crucial aspect of European culture. The view of computation offered in this paper qualifies. Also, pursuing theoretical interests without immediate practical gain seems a well-established European value – even though I admit it may be one of the old leisure class, rather than the hectic yuppies of to-day. But in this summer season, it is another image that intrigues me, based on just one passage in this paper. The undecidability of dynamic epistemic logic with iteration shows how the most difficult computational problems can be solved in *successful conversation*. Thus, what is

[38] This can be done, but no consensus exists. (van Benthem 1996, Chapter 11) dynamifies Herbrand models for Prolog to model such inferential steps, while (Abramsky 2006) presents a more general universe of abstract information states, where inference or computation steps do increase information.

[39] Cf. the forthcoming *Handbook of the Philosophy of Information*, P. Adriaans & J. van Benthem, eds., 2007.

going on at, say, all those Parisian terraces as I write these lines, on the last day of the Tour de France, is one gigantic parallel computer. 'Computing' usually evokes an image of boring machines, or even more boring nerds. Wouldn't it be great if Computing *in Europe* were the Art of Conversation in the Old World?

Acknowledgment. I would like to thank various audiences that have been exposed to these ideas, at the ESF Workshop 'Games and Verification' in Cambridge 2006, the ESSLLI Workshop on 'Knowledge and Rationality' in Malaga 2006, the Project 'Games, Action & Social Software' at NIAS Wassenaar 2006, and two Indian Logic Conferences held in Kolkata and Mumbai 2007. In particular, I would like to thank Merlijn Sevenster for helpful comments on complexity and games, and Andrea Sorbi for expert editorial help far beyond the call of duty.

References

[Abramsky 2006] Abramsky S. (2006), Information, Processes, and Games. To appear in Adriaans P., van Benthem J. (eds.), Handbook of the Philosophy of Information. Elsevier Science Publishers, Amsterdam

[Abramsky & Coecke 2004] Abramsky S., Coecke B. (2004) A Categorical Semantics of Quantum Protocols. In Proceedings of the 19th Annual IEEE Symposium on Logic in Computer Science (LiCS '04), IEEE Computer Science Press

[Baltag et al. 1998] Baltag A., Moss L., Solecki S. (1998) The Logic of Public Announcements, Common Knowledge and Private Suspicions. In: Proceedings TARK 1998, pages: 43–56, Morgan Kaufmann Publishers, Los Altos

[Baltag & Smets 2004] Baltag A., Smets, S. (2004) The Logic of Quantum Programs. In: Proceedings of the 2nd International Workshop on Quantum Programming Languages, TUCS General Publication No 33, Turku Center for Computer Science. Extended version: LQP: The Dynamic Logic of Quantum Information. Oxford Computing Lab & Philosophy, Free University Brussels

[Baltag & Smets 2006] Baltag A., Smets S. (2006) Dynamic Belief Revision over Multi-Agent Plausibility Models. In: Proceedings LOFT 2006, Department of Computing, University of Liverpool

[van Benthem 1996] van Benthem J. (1996) Exploring Logical Dynamics. CSLI Publications, Stanford

[van Benthem 1999] van Benthem J. (1999) Logic in Games. Lecture Notes, ILLC Amsterdam

[van Benthem 2001] van Benthem J. (2001) Games in Dynamic Epistemic Logic. In: Bonanno G., van der Hoek W. (eds.) Bulletin of Economic Research, 53:4, 219–248

[van Benthem 2005A] van Benthem J. (2005) An Essay on Sabotage and Obstruction. In D. Hutter (ed.), Mechanizing Mathematical Reasoning, Essays in Honor of Jörg Siekmann on the Occasion of his 69th Birthday. Springer, LNCS, 2605:268–276

[van Benthem 2005B] van Benthem J. (2005) Open Problems in Logic and Games. In: Artemov S., Barringer H., d'Avila Garcez A., Lamb L., J. Woods (eds.) Essays in Honour of Dov Gabbay. King's College Publications, London, 229–264

[van Benthem 2006A] van Benthem J. (2006) Dynamic Logic of Belief Revision. ILLC Tech Report, DARE electronic archive, University of Amsterdam. To appear in: J. Applied Non-Classical Logics

[van Benthem 2006B] van Benthem J. (2006) Living With Rational Animals. Invited Lecture at Workshop on Knowledge and Rationality, 18th ESSLLI Summer School, Malaga

[van Benthem 2006C] van Benthem J. (2006) One is a Lonely Number: on the logic of communication. In: Chatzidakis Z., Koepke P., Pohlers W. (eds.) Logic Colloquium '02. ASL and A.K. Peters, Wellesley MA, 96–129

[van Benthem 2006D] van Benthem J. (2006) Open Problems in Update Logic. In Gabbay D., Goncharov S., Zakharyashev M. (eds.) Mathematical Problems from Applied Logic I. Springer, New York, Novosibirsk, 137–192

[van Benthem 2007] Rationalizations and Promises in Games. Bejing Philosphical Review, Chinese Academy of Social Sciences

[van Benthem et al. 2005] van Benthem J., van Eijck J., Kooi B. (2005) A Logic for Communication and Change. ILLC & CWI Amsterdam & philosophy department, Groningen. First version in van der Meijden R. et al. (eds.) Proceedings TARK 2005, Singapore. Extended version in: Inform. and Comput. 2006

[van Benthem & Blackburn 2006] van Benthem J., Blackburn P. (2006) Modal Logic: a Semantic Perspective. Tech Report, ILLC, Amsterdam. In: Blackburn P., van Benthem J., F. Wolter & (eds.) Handbook of Modal Logic. Elsevier, Amsterdam, 2007

[van Benthem & Liu 2004] van Benthem J. , Liu F. (2004) Diversity of Logical Agents in Games. Philosophia Scientiae 8 (2): 163–178

[van Benthem & Liu 2005] van Benthem J., Liu F. (2005) Dynamic Logic of Preference Upgrade. ILLC Tech Report, DARE electronic archive, University of Amsterdam. To appear in: J. Applied Non-Classical Logics

[van Benthem & Pacuit 2006] van Benthem J., Pacuit E. (2006) The Tree of Knowledge in Action. Tech Report, ILLC Amsterdam. In: Proceedings AiML 2006, Melbourne

[Brafman et al. 1993] Brafman R., Latombe J-C, Shoham Y. (1993) Towards Knowledge-Level Analysis of Motion Planning. Proceedings AAAI 1993, pages: 670–675

[Condon 1988] Condon A. (1988) Computational Models of Games. PhD Thesis. Computer science Department, University of Washington

[Dechesne & Wang] Dechesne F., Wang Y. (2007) Dynamic Epistemic Verification of Security Protocols. In: van Benthem J., Ju S. Veltman F. (eds.) A Meeting of the Minds. Proceedings LORI, Bejing 2007. College Publications, London, 129–143

[van Ditmarsch 2000] van Ditmarsch H. (2000) Knowledge Games. Dissertation DS-2000-06. Institute for Logic, Language and Computation, University of Amsterdam

[van Ditmarsch 2002] van Ditmarsch H. (2002) Keeping Secrets with Public Communication. Department of Computer Science. University of Otago

[van Emde Boas 2002] van Emde Boas P. (2002) Models for Games and Complexity. Lecture Notes. ILLC, Amsterdam

[Fagin et al. 1995] Fagin R., Halpern J., Moses Y., Vardi M. (1995) Reasoning about Knowledge. MIT Press, Cambridge (Mass.)

[Gerbrandy 1999] Gerbrandy J. (1999) Bisimulations on Planet Kripke. Dissertation DS-1999-01. Institute for Logic, Language and Computation. University of Amsterdam

[Gigerenzer & Todd 1999] Gigerenzer G. , Todd P. M. , ABC Research Group (1999) Simple Heuristics That Make Us Smart. Oxford University Press

[Gochet] Gochet P. An Open Problem in the Logic of Knowing How. To appear in: Hintikka J. (ed.) Open Problems in Epistemology. IIP, Paris

[Greenlaw et al. 1991] Greenlaw R., Hoover H., Ruzzo W. (1991) A Compendium of Problems Complete for ***P***. University of Alberta, Computer Science Department, Technical Report 91-11

[Halpern & Vardi 1989] Halpern J., Vardi M. (1989) The Complexity of Reasoning about Knowledge and Time, I: lower bounds. J. Comput. System Sci.,38(1):195–237

[Hintikka 1962] Hintikka J. (1962) Knowledge and Belief. Cornell University Press, Ithaca

[Hintikka & Sandu 1997] Hintikka J., Sandu G. (1997) Game-Theoretical Semantics, in van Benthem J., ter Meulen A. (eds.) Handbook of Logic and Language, Elsevier, Amsterdam, 361–410.

[Kozen et al. 2000] Kozen D., Harel D., Tiuryn J. (2000) Dynamic Logic. MIT Press, Cambridge (Mass.)

[Liu 2006] Liu F. (2006) Diversity of Logical Agents. ILLC Research Report, University of Amsterdam. Presented at Workshop on Bounded Agents, ESSLLI Malaga 2006

[Lutz 2005] Lutz C. (2005) Complexity and Succinctness of Public Announcement Logic. LTCS Report 05-09, Technical University Dresden

[Marx 2006] Marx M. (2006) Complexity of Modal Logics. To appear in: Blackburn P., van Benthem J., WolterF. (eds.) Handbook of Modal Logic. Elsevier, Amsterdam

[Miller & Moss 2005] Miller J., Moss L. (2005) The undecidability of iterated modal relativization. Studia Logica, 79(3): 373–407

[Osborne & Rubinstein 1994] Osborne M., Rubinstein A. (1994) A Course in Game Theory. The MIT Press, Cambridge (Mass.)

[van Otterloo 2005] A Security Analysis of Multi-Agent Protocols. Dissertation. Department of Computing, University of Liverpool, & ILLC, University of Amsterdam, DS-2005-05

[Papadimitriou 1994] Papadimitriou C. (1994) Computational Complexity. Addison-Wesley

[Parikh 2002] Parikh R. (2002) Social Software. Synthese 132: 187–211

[Parikh & Ramanujam 2003] Parikh R., Ramanujam R. (2003) A Knowledge Based Semantics of Messages. CUNY New York & Chennai, India. In: van Benthem J., van Rooy R. (eds.) Special Issue on Information Theories, J. Logic Lang. Inform. 12(4): 453–467

[Plaza 1989] Plaza J. (1989) Logics of Public Announcements. In: Proceedings 4th International Symposium on Methodologies for Intelligent Systems

[Pratt 1976] Pratt V. (1976) Semantical Considerations on Floyd-Hoare Logic. In: Proceedings 17th Ann. IEEE Symposium on Foundations of Computer Science, 109–121

[Rahman et al. 2004] Rahman S., Gabbay D., Van Bendegem J-P , Symons J. (2004) Logic, Epistemology, and the Unity of Science, Vol. I. Kluwer, Dordrecht

[Reiter 2001] Reiter R. (2001) Knowledge in Action. The MIT Press, Cambridge (Mass.)

[Rohde 2005] Rohde Ph. (2005) On Games and Logics over Dynamically Changing Structures. Dissertation. Rheinisch-Westfälische Technische Hochschule Aachen

[Sevenster 2006] Sevenster M. (2006) Branches of imperfect information: logic, games, and computation. Dissertation DS-2006-06. ILLC Amsterdam

[Turing 1950] Turing A. M. (1950) Computing machinery and intelligence. Mind 59: 433–460

[Vardi 1997] Vardi M. (1997) Why is Modal Logic So Robustly Decidable?. In Immerman N., Kolaitis Ph. (eds.) Descriptive Complexity and Finite Models. American Mathematical Society.

[Yao 1979] Yao A. C. (1979) Some Complexity Questions Related to Distributed Computing. In: Proceedings of the 11th STOC, 209–213