COMPUTATION AS SOCIAL AGENCY: WHAT AND HOW

Johan van Benthem

University of Amsterdam & Stanford University, http://staff.fnwi.uva.nl/j.vanbenthem

Abstract The Turing machine paradigm of the 1930s abstracted away from computing by human agents, in a fruitful manner that focused on what is computable in principle. But the agents have struck back. Modern computation may be seen as agency in social networks with communication links and a wide range of available actions. This means that the traditional partnership of mathematical logic and computation, though still flourishing in its own right, gets enriched in several ways, with influences coming from fields as far apart as philosophy and economic game theory.

In this example-based discussion and survey paper, we identify two major channels for this influence. One is the rise of 'epistemization' where conditions on or specifications of computational tasks refer explicitly to knowledge or beliefs of the agents performing these tasks. The other channel is the rise of games as a paradigm for interactive computation, leading to 'gamification' of computing tasks and of logical notions themselves. Thus, knowledge, beliefs, goals, preferences, intentions, and other features of agents traditionally studied in philosophical logic enter models of computation. We describe these phenomena with a special focus on games, and discuss some challenges, some with a narrower focus, some across disciplines, that emerge out of these. They all have to do with the shift from what is computed to how it is computed, or in other terms, a shift from output to behavior.

Finally, we discuss what all this means for a general understanding of computation, especially when we realize that the true impact of modern computers is not in replacing humans, but in creating new mixed societies where humans and machines interact. Thus, computer science meets cognitive science of human agents, and the sum of our themes becomes an integral part of general culture.

1 Views of computing, from 'what' to 'how'

The Turing Year 2012 saw lively debates about the nature of computing. Are we on the threshold of new styles of computation that transcend the limitations of established paradigms? Let us briefly recall three classical themes from the golden age when Turing and his generation made computation a subject for mathematical inquiry, and hand in hand with that, for practical development. First of all, by analyzing the bare basics of mechanical computing, Turing defined a *Universal Machine* that can compute the result of any algorithm on any matching input, when both are presented in suitably encoded form. This notion then supported the subsequent development of Recursion Theory, bringing to light both the basic structures and powers of effective computation, but also its limitations as exemplified in the undecidability of the Halting Problem. Second, on the basis of this and equivalent models proposed independently at the time by Church, Gödel, Post and others, *Church's Thesis* then claimed that all effectively computable functions over the natural numbers, a canonical domain that can mimic non-numerical computation by various encodings going back to Gödel and others, coincide with the 'recursive functions', that can be computed on Turing machines. Finally, as the power of this paradigm became clear, it was suggested in the famous *Turing Test* that computation might well emulate human cognition, to the extent that in conversation, humans would not be able to tell whether they are interacting with another human or a machine.

Now, 80 years later, computer science and information technology have totally transformed the world of both machines and humans in sometimes wholly unpredictable ways. Given the experience obtained over this period, can we now spring the bounds of the classical age, and compute or solve a larger class of functions/problems after all? There are lively debates on this theme, with proposals like using infinite machines or letting the physical universe do computing in its own ways (Cooper et al., eds., 2008, Cooper 2011). Still, this is a debate I would like to sidestep. It seems important to make a distinction between two issues:

(a) *What* can we compute,
(b) *How* can we compute it?

Stated somewhat apodictically, I see no evidence in current debates that we can compute more than before, forcing us to extend the calibration class of recursive functions. But then, this What question is not of great interest to me. Of much greater interest is a How question, not addressed by Church's Thesis, namely: what are natural styles of computing? Or if you insist on 'what' questions after all: do not ask what is computable, but what is *computing,* seen as a kind of process producing characteristic forms of *behavior* (cf. Abramsky 2008A).

Right from its start, the history of computer science has shown an outburst of ideas on these themes, and this paper is about one of these: computation as social agency. My discussion will have a logical slant, reflecting my own work in this area (van Benthem 2008, 2011, 2014), and I am not claiming to represent public opinion in either computer science or logic.

2 Computer science as a hot spring of ideas

Before I start with my main theme, however, here is some very quick historical background that may be familiar to computer scientists, but that is often not on the radar of many of my fellow logicians and philosophers interested in the foundations of computing.

Logic and fine-structuring views of computing Turing machines have opaque programs telling the machine in complete detail what to do in machine code, making heavy use of that old enemy of perspicuity called 'go to' instructions (Dijkstra 1968). Real computer science took off when higher programming languages were invented, that represent higher-level perspectives on the sort of computation taking place. One can think of programs in such languages as 'algorithms' that describe the essence of some computational task at some more suitable abstraction level. Different programming languages have given a wealth of views on this, often drawing on traditions in logic. For instance, imperative programs in Algol or C^+ can be viewed as a 'dynamified' version of logical formulas in standard formalisms like predicate logic, telling the machine what sort of results to achieve given certain preconditions. Such systems lend themselves to model-theoretic semantics in the usual logical style (first-order, modal, or otherwise), witness the Hoare Calculus or Dynamic Logic (cf. Harel, Kozen & Tiuryn 2000). On the other hand, functional programming languages such as LISP or Haskell, akin to systems of lambda calculus and type theory, are closer to proof-theoretic and category-theoretic traditions in logic. And of course, there are many other styles that do not fall simply into this dichotomy, including object-oriented programs, logic programs, and so on. The semantics for this large family of programming languages have provided a wealth of matching process models that offer many insights into how we compute and even how we act.

Distributed computation and process theory A major new development around 1980 was a theoretical reflection on the practice of distributed computing emerging at the time. One major line here, moving up the abstraction level beyond programming languages, was the invention of Process Algebra by Milner, Bergstra, Klop and others (Bergstra, Ponse & Smolka, eds., 2001), an abstract view of processes as graphs modulo some suitable notion of structural behavioral invariance, often some variant of bisimulation. While it is true that no consensus has emerged on one canonical model of distributed computation, comparable in breadth of allegiance to Turing machines, a deep process theory did emerge with many features that have no counterpart in the classical theory of sequential computing (van Emde Boas 1990). Abstract process theories are still emerging in the foundations of computation. A noticeable new development has been the birth of *co-algebra*, a theory of computing on infinite streams, tied to fixed-point logics and category-theoretic methods (Venema 2006, 2012). My point here is very modest: thinking about the foundational hows of computation is a productive line of thought that shows no signs of abating yet. For instance, in the last 15 years, a striking take on multi-agent distributed computing has been the introduction of *game models* (Abramsky 2008, and in another paradigm Grädel, Thomas & Wilke, eds., 2002), leading to new encounters between computer science, logic, and game theory (van Benthem 2014).

Artificial intelligence To be added to this mix is the emergence of *Artificial Intelligence* since the 1950s. While to some, this has been the soft side of computer science, AI has added a lot of major themes to our understanding of computation, such as automated deduction or learning, that I will not even begin to enumerate here. But I do note one striking inversion. While a major thrill in the area has been the goal of replacing humans by machines, the reality of history is that AI has done more than any other discipline to revitalize and speed up research in philosophical logic (cf. Gabbay, Maibaum & Shepherdson, eds., 1995), teaching us a lot of new things about human behavior in the process. Those who seek to replace us often study us the best. This leads me to my next topic, and indeed the main theme of this paper.

3 Computation and social agency

It is often said that Turing took the human out of the term 'computer', extracting only the abstract procedures behind their pencil-and-paper activity. In that light, the Turing Test then added insult to injury, since the computer thus defined might then even dispel the mystery of our other intelligent tasks just as well. And even without such grand reductionist programs, it is undeniably true that exact computational models have proved of immense value in studying what might be considered typical human activities, such as conversation as a form of computed information flow driven by natural language functioning as a programming language (van Benthem 1996). I will mention more examples of this influence below.

But there is an opposite stream as well. Much of the recent history of computer science can be seen as a case of the 'humans striking back'. Here is a first, still relatively mild instance of this phenomenon. Around 1980, Halpern and others started the TARK tradition of enlisting the delicate yet powerful understanding that we have of human agents with knowledge and social activity in support of modeling complex distributed protocols: how they function, what they do and do not achieve, and what might go wrong with them (Fagin, Halpern, Moses & Vardi 1995). Now this may be just a metaphor, but the resulting revival of epistemic logic, broadly conceived, has had widespread repercussions in several disciplines.

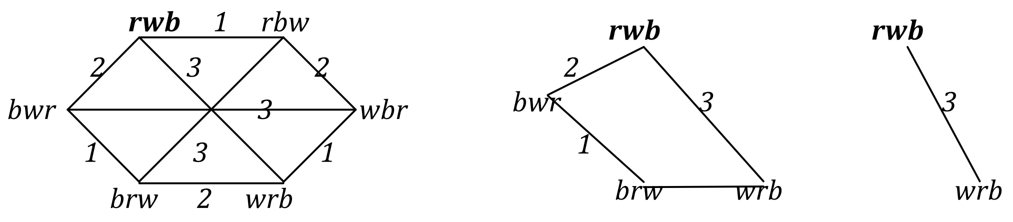
Likewise, as we said already, even much earlier, the rise of Artificial Intelligence, perceived by some as a reductionist exercise, in fact made computer scientists and others aware of the amazing subtlety of human behavior and skills in problem solving, knowledge acquisition, and social interaction. A wealth of logical systems arose out of this that started influencing other areas far beyond computer science, including linguistics and philosophy. And finally, just consider the realities of computing today. What we see all around us are complex societies of humans and machines engaged in new interactive styles of behavior, nowadays even driven by design of 'gaming' as ICT-facilitated human behavior. Thus, it may even be that the real challenge today is understanding what makes this reality tick, rather than abstruse discussions about computing at infinity or in the Milky Way.

4 Conquering daily life: conversation as computation

Computational models are widely used in the study of human agency today – so much so, that the area of multi-agent systems combines features of computer science, game theory, epistemology, and cognitive science (Leyton-Brown & Shoham 2009, Wooldridge 2009). Of the vast literature in this area merging themes and techniques from logic, computer science and other disciplines, I mainly mention one current strand as an illustration: viz. 'dynamic-epistemic logics' (cf. Baltag, Moss & Solecki 1998, van Benthem, van Eijck & Kooi 2006, van Ditmarsch, van der Hoek & Kooi 2007, van Benthem 2011). This is by no means the only game in town, but in this paper, it will merely serve as a pilot illustration for collaboration and confluences between the fields I am interested in.

Conversation and information update Simple games are a good setting for studying communication. Here is a standard example. Three cards "red", *w* "white", and "blue" are given to three children: *1* gets red*, 2* white, and *3* blue. Each child sees his own card, not the others. Now *2* asks *1:* "Do you have the blue card?", and the truthful answer comes: "No". Who knows what now? Here is what seems the correct reasoning. If the question is genuine, agent *1* will know the cards after it was asked. After the answer, agent *2* knows, too, while *3* still does not. But there is also knowledge about others involved. At the end, all players know that *1* and *2*, but not *3*, have learnt the cards, and this is even 'common knowledge' between them. [1]

The Cards scenario involves a computational process of state change, whose basic actions are updates shrinking a current range. In the diagrams below, indexed lines indicate an uncertainty for the relevant agents. Informational events then shrink this range stepwise:



The first step is for the presupposition of the question, the second for the answer. In the final model to the right, both players *1* and *2* know the cards, but *3* does not, even though he can see that, in both of his remaining eventualities, *1, 2* have no uncertainties left. ■

The geometry of the diagram encodes both knowledge about the facts and knowledge about others: such as *3*'s knowing that the others know the cards. The latter kind of 'higher' knowledge is crucial to social scenarios, holding behavior in place. Indeed, at the end of the scenario, everything described has become *common knowledge* in the group *{1, 2, 3}*. [2]

[1] This way of understanding the scenario presupposes that the questions and answers are sincere – as is reasonable with children. But dynamic-epistemic methods also cover possibly insincere scenarios.

[2] Cf. Fagin, Halpern, Moses & Vardi 1995 for all these notions in games and computation.

Dynamic logics of communication The cards problem can be described in the standard epistemic logic that we know from philosophy. What agents know about the facts, or each other, at any state in the above process is described by a standard language of *epistemic logic*:

$$p \mid \neg\varphi \mid \varphi\wedge y \mid K_i\varphi$$

while the corresponding epistemic models, matching our diagrams, are tuples

$$\boldsymbol{M} = (W, \{\sim_i \mid i\in G\}, V)$$

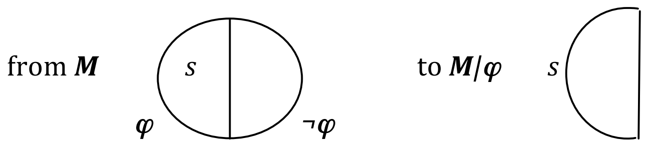
with a set of relevant worlds W, accessibility relations $\sim_I$ for each agent i, and a propositional valuation V for atomic facts. Knowledge is then defined as having semantic information: [3]

$$\boldsymbol{M}, s \models K_i\varphi \quad \text{iff for all worlds } t \sim_i s\text{: } \boldsymbol{M}, t \models \varphi$$

Common knowledge $\boldsymbol{M}, s \models C_G\varphi$ is defined as φ's being true for all t reachable from s by finite sequences of $\sim_i$ steps. If necessary, we distinguish an *actual world* in the model.

But we can go one step further. 'Dynamic epistemic' logics that also describe the information *flow* in the preceding scenario, i.e., the changes in what agents know from state to state, have been found on the analogy of program logics in computer science.

Update as model change The key idea is now that informational action is model change. The simplest relevant action or event here is a *public announcement* $!\varphi$ of hard information: learning with total reliability that φ is the case eliminates all current worlds with P false:



We call this *hard information* for its irrevocable character: counter-examples are removed. [4]

This dynamics typically involves truth value change for complex formulas. While an atom p stays true after update (the physical base facts do not change under communication, though current dynamic-epistemic logics can also handle factual change in the world if needed), complex epistemic assertions may change their truth values: before the update $!p$, I did not know that p, afterwards I do. As with imperative programs, this may result in order dependence. A sequence $!\neg Kp; !p$ makes sense, but the permuted $!p ; !\neg Kp$ is contradictory.

Public announcement logic The dynamic logic *PAL* arises by extending the epistemic language with a dynamic modality for public announcements of true facts, interpreted as follows:

$$\boldsymbol{M}, s \models [!\varphi]\psi \quad \text{iff} \quad \textit{if } \boldsymbol{M}, s \models \varphi, \textit{ then } \boldsymbol{M}|\varphi, s \models \psi$$

Note how several cultures meet in the logical formulas available here. Consider $[!\varphi]K\psi$, which says that an agent knows ψ that after reliable information φ has been received. The K-modality here comes from philosophy, the explicit treatment of $!\varphi$ comes from linguistic speech act theory, while the *[]*-modality comes from dynamic logic of programs in computer science. [5]

[3] Epistemic models encode 'semantic information' as a current range of options for the real world, a widespread notion – but other logical views of information exist (van Benthem & Martinez 2008).

[4] Note: This information need not come from anything being said: it could also be a public observation.

[5] Logical artifacts are often little pieces of interdisciplinary art, just as so major artifacts in our daily lives testify to meetings and cooperation between major world cultures.

The resulting system of `public announcement logic' *PAL* can be axiomatized completely by combining a standard logic for the static epistemic base plus a *recursion law* for knowledge that holds after update, the basic 'recursion equation' of the system:

$$[!\varphi]K_i\psi \;\leftrightarrow\; (\varphi \rightarrow K_i(\varphi \rightarrow [!\varphi]\psi))$$

Dynamics of other agent features Similar logics exist for updating agents' *beliefs*, defined as truth in the *most plausible* epistemically accessible worlds. Here variety of dynamic events increases. Beliefs can change under hard information, but also under *soft* information, where $\neg\varphi$-worlds are not eliminated, but made less plausible than φ–worlds (Baltag & Smets 2008). Similar methods work for modifying *preferences* (Liu 2011), and a wide range of other events that change key features of agents (van Benthem 2011), or social networks of agents (Girard, Liu & Seligman 2014, Christoff 2016). However, we will not need technical details of such further update mechanisms for the points to be made in what follows.

Time and program structure But there is more to information flow in agency. Single events are just triggers for local atomic actions, but usually, they such actions congregate over time to form meaningful larger scenarios. Again, computational ideas are essential here. Action and communication involve complex programs with operations of sequential composition ***;***, guarded choice *IF THEN ELSE*, and iteration *WHILE DO*. Even parallel composition *||* occurs when people act or speak simultaneously: parallel action is crucial to social settings.

Here is a well-known concrete illustration, that we will mainly use to identify a few more general themes, without any attempt at deeper technical coverage in this paper:

The Muddy Children "After playing outside, two of a group of three children have mud on their foreheads. They can only see the others, and do not know their own status. Now the Father says: "At least one of you is dirty". He then asks: "Does anyone know if he is dirty?" The Children always answer truthfully. What will happen?"

No one knows in the first round. But in the next round, each muddy child reasons as follows: "If I were clean, the one dirty child I see would have seen only clean children, and would have known that she was dirty at once. But she did not. So I am dirty, too." This scenario falls within the above update setting, even with arbitrary iterations, but we forgo details. ■

Clearly, there is a program here involving sequence, guarded choice and iteration:

!"At least one of you is dirty" ; WHILE not know your status

DO (IF not know THEN "say don't know" ELSE "say know")

Temporal limit behavior Another interesting feature is the limit behavior in the puzzle, leading to a stable endpoint where updates have no further effect and agents' knowledge is in equilibrium. In particular, the children have common knowledge of their status in the limit model $\#(\boldsymbol{M}, \varphi)$ reached by the iterated updates $!\varphi$ of their ignorance assertion. So in the end, this statement 'refutes itself'. In other scenarios, like game solution procedures viewed as deliberation scenarios for agents intending to play the game (van Benthem 2007, 2014), the statement announced is rather 'self-fulfilling', becoming common knowledge in the limit. [6]

[6] There is no general theory of what happens when. Also, things get yet more complex with belief revision over time, a mechanism in formal learning theory: Gierasimczuk 2010. With iterated plausibility change, fixed-points need not always be reached: oscillations and even divergence may occur.

Limit features of computation over time can be studied in sophisticated fixed-point logics, but one simple case is just propositional dynamic logic *PDL* of basic imperative programs. Even this natural setting has vast computational power (Miller & Moss 2005): the logic *PAL* with finite iteration of updates is Π^1_1-complete. Van Benthem 2008 has a positive interpretation of high complexity results for logics like these in the realm of agency, namely that

conversation has universal computing power:

That is, any significant computational problem can be realized as conversation planning. However, while this looks attractive as an observation about conversation as a paradigm for computation in general, there is a catch. The high complexity resides in the logic of reasoning about conversation, but as discussed in van Benthem 2011, conversational algorithms themselves might have low complexity as far as computational procedures go.

Digression Logic often offers different resources for looking at agency. The above scenarios can also be described in `epistemic temporal logics' (cf. Fagin, Halpern, Moses & Vardi 1995, van Benthem & Pacuit 2006), an alternative, though related line that we will ignore here. [7]

Our examples and their glimpses of wider implications may have shown how computational notions and techniques arise all the way in a basic human activity like conversation. For many further examples of 'communication as computation', we refer to the cited literature.

5 Daily life strikes back: computation as conversation

The preceding section has illustrated how computation can inform the study of social agency. But our view in Section 3 had two aspects: it thrives just as well on a reverse perspective. Human-style agency may also be seen as an essential aspect of computation. The work in the TARK community mentioned earlier has shown how human metaphors of knowledge and social interaction, made precise in logical terms, are a powerful tool for specifying and proving properties of complex protocols for multi-agent systems. But the borderline between a metaphor and the real thing may be thin. One can also entertain a viable view that computing *itself* is a form of social behavior, mixing action and information much as humans do. After all, there does not seem to be preferred direction of reduction in the mixed information society of today. Correspondingly, the same formal objects that act as programs for machines might also be viewed, with a mental Gestalt switch, as 'plans' or strategies for humans.

Two essential human features then enter our understanding of computation. One is the *knowledge* of agents (and also other attitudes, such as their beliefs and preferences), and the other their social *interaction*. That this mixture is stable from a logical point of view has been shown in the preceding section. We now step back, and look at broader developments that support this trend. We will take up the two themes separately, showing how they enter our view of computing in several natural ways. Our major tool for highlighting this influence are transformations from standard algorithms to knowledge-based social procedures.

6 Epistemizing computational tasks

This section is about what we call the phenomenon of *epistemization*, the introduction of agents' knowledge at various parts in basic computational tasks.

[7] This complexity may go down on extended *protocol models* for epistemic-temporal logics (van Benthem, Gerbrandy, Hoshi & Pacuit 2009) that constrain available histories, or sequences of updates.

Epistemizing algorithmic tasks Consider the basic planning task of Graph Reachability (*GR*). Given a graph ***G*** with points *x, y*, is there a chain of arrows from *x* to *y*? *GR* can be solved in *Ptime* in the size of ***G***: a quadratic-time algorithm will find a path (cf. Papadimitriou 1994). The same holds for reachability of a point in ***G*** satisfying a goal condition φ. In fact, the solution algorithm performs two related tasks: determining if a path exists at all, and giving a concrete way or plan for getting from *x* to *y*. We will now consider various natural ways of introducing knowledge and information in this setting that come from the literature.

Knowing you made it Suppose an agent is trying to reach a goal region defined by φ, with only limited observation of the terrain. The graph ***G*** is now a model *(G, R, ~)* with accessibility arrows, but also the earlier epistemic uncertainty links between nodes. It is natural to ask for a plan that will lead you to a point *that you know to be in the goal region* φ. Brafman, Latombe & Shoham 1993 analyze a robot whose sensors do not tell her exactly where she is. They then add a *knowledge test* to the task specification, inspecting current nodes to see if we are definitely in the goal region φ, by requiring that $K\varphi$ be true. Given the well-known *P-time* complexity of model checking for epistemic logic, the new search task remains *P-time*. [8]

Epistemizing social tasks Many algorithmic tasks themselves come from social scenarios, witness the area of computational social choice (Endriss & Lang, eds., 2006). Here, too, epistemization makes sense. Think of the basic computational task of *merging orderings*. In social choice theory, preferences of individual agents are to be merged into a preference order for the group as a whole. This way of phrasing started with Arrow's Theorem stating that no social choice procedure exists that satisfies some basic postulates of unanimity, monotonicity, context independence, and especially, absence of a 'dictator', an individual whose preference ranking always coincides with that of the group. These famous specifications are completely non-epistemic, which is somewhat surprising, since much of what we consider essential about democratic decision making has to do with privacy, and what agents may know or not know. But, there is even a mismatch between the usual base conditions in social choice theory and how they are interpreted intuitively in terms of agency. The existence of a dictator is problematic if we think of an individual who can abuse her powers: but for that, she should *know* that she is a dictator – and perhaps, others should also know (or not know) this. Thus, epistemic rethinking of the scenario of social choice seems in order, and it is not yet clear what a knowledge-based version of the basic theory would look like.

Other algorithmic tasks where similar points can be made occur in *game theory*. Indeed, the move from games of perfect information to games with imperfect information (Osborne & Rubinstein 1994) may be considered a case of epistemization in our sense.

Two aspects of epistemization Our examples show two different aspects of introducing knowledge. One is that the *specifications* of what an algorithmic task is to achieve may come to involve knowledge, like saying we must know we are at the goal, This does not necessarily mean that the algorithm itself has to be epistemic. Many social algorithms are purely physical, such as folding ballot slips, although they do have epistemic effects. We will return to this distinction later on in our discussion of interfaces with game theory.

[8] Here is one more illustration of the entanglement of ideas from computation and general agency. A general model for epistemic robots relying on possibly limited or defective sensors is proposed in Su et al. 2005. This approach has led to new 'evidence models' for human agency (van Benthem & Pacuit 2011) that are more fine-grained than the standard epistemic models of this paper.

The second step, then, makes the *algorithms themselves* contain knowledge aspects. One obvious place where this happens is test conditions for conditional action. We find it obvious that a computer 'checks' in its registers whether, say, $x = 1$, before performing an *IF THEN* task: truth and knowledge are easily confused here. But for more complex conditions, such as 'the battery is functioning', we can only perform *IF THEN* instructions if we *know* which condition holds. And there may be yet more subtle aspects of knowledge involved in basic computing. Turing 1937 says a machine should know which symbol it is reading, and even which state it is in: this seems a rather human form of introspection. [9]

Epistemic programs Algorithms with conditions that we know to be true or false look like human plans. One format for epistemizing standard algorithms is the *knowledge programs* of Fagin, Halpern, Moses & Vardi 1995, making actions dependent on conditions like "the agent knows φ" that can always be decided given epistemic introspection. [10] The language of the programs now also explicitly contains our earlier epistemic operators. Knowledge programs make sense in epistemic planning (Bolander & Andersen 2011), and also as definitions for uniform strategies in imperfect information games (van Benthem 2014). [11]

A related way of epistemizing programs is offered by the earlier dynamic epistemic logics. Public announcements are closely related to the widely used dynamic 'test actions' that cut all epistemic uncertainty links between φ–worlds and $\neg\varphi$–worlds. The behavior of link-cutting test actions, and that of many related ubiquitous informational actions, such as asking questions and giving answers, can be described in exactly the same logical style as before.

Further aspects of epistemization But once we entangle algorithms and knowledge, many further issues emerge, going beyond just opening a 'knowledge parameter' here and there. For a start, epistemic specifications or programs essentially refer to some agent performing the task – and then, the nature of those agents becomes an explicit concern.

Different types of agent Epistemic algorithms may work for one type of agent but not for another. The literature on dynamic-epistemic logic has mainly focused on agents with *Perfect Recall* who remember everything they knew at earlier stages of the process, and who also learn from their observations only (van Benthem 2011). But equally important are agents with bounded memory and attention span, such as finite automata.

Making different assumptions about agents will be reflected in the epistemic logic of action. For instance, Perfect Recall holds for an agent iff the following apparently innocent and attractive-looking commutation law for action and knowledge governs its behavior:

$$K[a]\varphi \to [a]K\varphi$$

This says that, if we know beforehand what an action is going to achieve, we will know its effect afterwards. This is crucial to consciously following a plan, though it can fail in other circumstances. For a counter-example, I may know that entering this enticing nightclub will

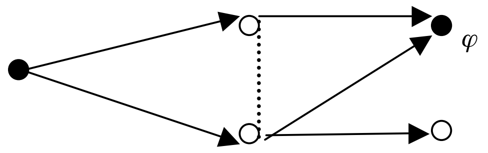
[9] Turing himself thinks that these epistemic properties are guaranteed by having only *finite* sets of symbols and states. This is not the notion of knowledge used in this paper, since it seems to refer more to perceptual discrimination (cf. Williamson 2000 on the latter notion in epistemology).
[10] Some of the surprising cognitive algorithms in Gigerenzer et al. 1999 have this flavor.
[11] Independently from this paper, a program of epistemization has been proposed in Moses 2015.

lead to my swift moral downfall, but once inside, all such thoughts may have left my mind. [12] Different axioms that can be written in a similar logical form govern the behavior of finite automata. Clearly, such assumptions about agents influence what we can expect an epistemic algorithm to achieve – but I am not aware of any general theory yet.

Know-how, and knowing a program So far we followed the mainstream of epistemic logic in letting knowledge apply to propositions. But our setting suggests a richer view. In addition to propositional *knowing-that*, computation involves a study of *know-how,* embodied in algorithms, plans and procedures. In fact, knowing how is the subject of much of our learning, perhaps more than knowing that (Stanley 2011). Much of what we have discussed already tends in this direction, since we treated actions and propositions on a par.

Moreover, know-how is related to an important notion in our natural language, that of knowing an *object*. We can know numbers, people, answers, and so on. In our setting, one obvious instance is what it means to 'know a program'. There seems to be no unique answer to this question, but here is a tie with propositional knowledge that seems relevant. If we have an epistemic program, knowing it seems to involve at least some grasp of its execution, not just being lucky. Should the agent know a plan to be successful: beforehand, and at all stages of its execution? There are at least two aspects to this mastery (van Benthem 2014). Suppose that the agent has an epistemic plan: does it follow that she knows its effects? It is easy to see that this is not always so, and hence we might use this as a stronger requirement on epistemized algorithms than we have imposed so far. But there is also another natural aspect to knowing a plan. Suppose that the agent knows now what the plan will achieve, will this knowledge persist over time as the plan is being followed?

Example Knowing one's plan. For an illustration, recall the earlier problem of epistemized graph reachability. Let the agent at the root of the following graph trying to reach a φ-point:



The dotted line says that the agent cannot tell the two intermediate positions apart. A plan that reaches the goal is *Up; Across*. But after the first action, the agent no longer knows where she is, and whether moving *Across* or *Up* will reach the φ-point. ■

Much more can be said about when intermediate knowledge of effects does hold, but we merely cite one result discussed in van Benthem 2014: agents with Perfect Recall have intermediate knowledge of effects for all knowledge programs in the earlier sense.

In the setting of this paper, the point of this discussion is just to show that epistemization may be a more subtle and wider-ranging enterprise than the reader might have thought. We reinforce this point with a small digression taking us even into contemporary philosophy.

[12] Attractive as this principle may look, in the presence of further simple expressive devices such as common knowledge or a universal modality, it induces high complexity in logics of agency. This is due to the confluent `grid structure' of two relations in imposes on models that makes it impossible to encode undecidable tiling problems in the logic, cf. Halpern & Vardi 1989, van Benthem & Pacuit 2006.

From knowing to understanding Even more stringent intuitive requirements come up with knowing a program, such as *understanding* what one is doing. [13] In addition to propositional knowledge of effects of a plan, or parts of it in the real world, one key feature then seems a form of modal 'robustness': namely, counterfactually still knowing the effects of a plan under changed circumstances, or at least having an ability to modify a plan as needed. [14] And there are yet other tests of understanding a subject, such as 'talent for zoom': being able to describe a plan at different levels of detail, moving up or down between grain levels as needed.

Epistemization in general We will not explore these issues further here, except to note that epistemizing algorithms seems to open up a rich and interesting area of investigation. Perhaps the first issue on the agenda here should be to *define epistemization* as a general transformation, or a family of transformations, on traditional algorithms and specifications, whose properties can then be studied as such. The next general issue would be what happens when we systematically epistemize major existing process theories of computation, such as process algebra or game semantics (Bergstra et al., eds., 2001, Abramsky 2008B). [15] There are bits and pieces in the literature, but I am not aware of general results in this spirit. The earlier dynamic-epistemic logics seem relevant to this enterprise, and so does the literature on computational complexity of epistemic-temporal action logics (Halpern & Vardi 1989).

Finally, it should be pointed out that epistemization is a more general phenomenon than just adding standard epistemic logic to the world of algorithms. Epistemic logic is one way of modeling knowledge – based, as we saw, on the notion of semantic information. However, various other views of information make sense for logic and computation, including more fine-grained syntactic accounts of information structure as code (van Benthem & Martinez 2008). The issues that we have raised in this section would still make sense then.

7 Interaction and gamification

The second essential feature of social agency that we mentioned earlier was multi-agent interaction. The typical paradigm for multi-agent action with many-mind knowledge are games, and what we will do know is look at a trend of 'social transformation' of algorithmic tasks that might be called *gamification* (van Benthem 2008).

Multi agent scenarios and knowledge games Reaching a goal and knowing you are there naturally comes with social variants where, say, *others* should not know where you are. In the 'Moscow Puzzle' (van Ditmarsch 2002), two players must inform each other about the cards they have without letting a third party know the solution. More general *knowledge games* of this sort have been studied in Ågotnes & van Ditmarsch 2011, Minica 2010. [16] One

[13] Similar issues arise in analyzing what it means for someone to understand a formal *proof*, and hence useful intuitions might also be drawn from our experience with mathematical practice.

[14] Counterfactual robustness under a natural range of deviant circumstances is also well-known in the philosophical literature on definitions of knowledge: see Nozick 1981, Holliday 2012. In that literature, knowledge gets tied to policies for *belief revision* – and the intriguing thought that real understanding of a plan, program or algorithm might also have to do with the relevant agents' *beliefs* about it.

[15] Adding epistemic action to process algebra would fit its emphasis on communication channels. Adding an explicit epistemic component would also makes sense given the intuitive motivations underlying game semantics of programming languages (Abramsky 2008B).

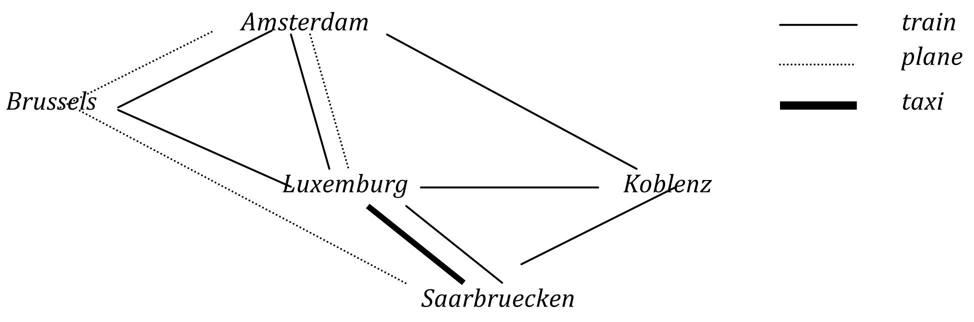
[16] Games for inquiry even play an important role in modern epistemology, Lehrer 1990, Fiutek 2013.

can think of these games as extended semantic explorations of a given epistemic model, assigning different roles to different parties to model more interesting features of inquiry.

Logic games Turning algorithms into games involves prying things apart assigning different roles for different agents. Early examples of this gamification have long occurred in logic, witness so-called *logic games* in the style of Lorenzen, Ehrenfeucht, or Hintikka, where basic logical notions are gamified in terms of a split between a player for truth (proof, analogy, ...) versus a player for falsity (existence of a counter-model, seeing a difference, ...). The strategic game-theoretic powers of players in such games then provide a new fine-structured analysis of many classical logical notions. For instance, truth of a formula will correspond to the existence of a winning strategy for the Verifier (in our earlier terms, the 'what' aspect), while that strategy itself is a game-theoretic procedure, much like a generalized algorithm, for establishing the truth (the 'how', in our earlier terms). Similar results hold for other logic games. Van Benthem 2014 has an extensive survey of logic games for evaluation of formulas, comparing between models, constructing models, finding proofs, and general argumentation.

Gamification has a long history in logic, but it can also be used creatively for computation.

Reachability and the sabotage game For a more purely algorithmic example, consider again the earlier Graph Reachability, now in a different scenario with two agents. The following picture gives a travel network between two European capitals of logic and computation:



It is easy to plan trips either way. But what if transportation breaks down, and a malevolent Demon can cancel connections, anywhere in the network? At each stage of our trip, let the Demon first take out one connection, while Traveler then follows a remaining link. This turns a one-agent planning problem into a two-player *sabotage game*. Simple game-theoretic reasoning shows that, from Saarbruecken, a German Traveler still has a winning strategy, but in Amsterdam, Demon has the winning strategy against the Dutch Traveler. [17]

Sabotage, logic, and complexity The above suggests a transformation for any algorithmic task to a *sabotage game* with obstructing players. This raises general questions. First, there is an issue of logic (van Benthem 2005). One can design languages for these games and players' strategies in terms of 'sabotage modalities' on models with accessibility relations *R*:

$$\boldsymbol{M}, s \models \langle - \rangle \varphi \text{ iff there is a link } (s, t) \text{ in } R \text{ such that } \boldsymbol{M}[R := R - \{(s, t)\}], s \models \varphi$$

[17] These games also have interesting interpretations in terms of *learning*, where a Teacher tries to trap a Student into a certain state of knowledge by blocking all escape routes (Gierasimczuk 2010). Recently, it has also been suggested that they can be used to model changes in dynamic communication networks such as being 'unfriended', Christoff 2016, Aucher, van Benthem & Grossi 2015.

In these unusual modal logics, models change in the process of evaluation, and indeed, one can show that sabotage modal logic, though axiomatizable, is *undecidable*: somehow the computational content of the logic has increased from standard modal logic. [18] Next, consider other tasks. For sabotaged Graph Reachability, the solution complexity of the game jumps from *P-time* for modal model checking to *Pspace*-completeness. This takes a polynomial amount of memory space, like Go or Chess. For all these results, see Rohde 2005. [19]

Thus, gamification leads to interesting new perspectives in complexity, and in logic, where models may now change in a process of evaluation. Still, it all depends on how we do it. Game versions of algorithmic tasks need not always be more complex than the original task.

Catch me if you can Consider another game variant of *GR*. Obstruction could also mean that someone tries to stop me en route: "Starting from an initial position ***(G**, x, y)* with me 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." This game, too, models realistic situations, such as avoiding obnoxious people at receptions. One important difference with the sabotage game is that, this time, the graph remains fixed during the course of play. Sevenster 2006 proves that now the computational complexity stays in *P-time*.

Adding knowledge and observation again Clearly, it also makes sense to combine the above games with our earlier epistemization. For instance, real sabotage as practiced in warfare involves limited observation and partial knowledge. If we turn algorithms into games of *imperfect information*, solution complexity may increase even further. Jones 1978 gives an exponential complexity jump for a simple search task. Sevenster 2006 studies a broader array of epistemized gamified algorithms, linked with the '*IF* logic' of Hintikka & Sandu 1997.

Gamification in general The general program behind the preceding examples would be a theory of gamifying algorithmic tasks, and the study of their strategic properties as related to their earlier process properties. We mentioned knowledge games and sabotage games as specific instances – but as we have said, many further examples of successful gamification exist in logic and computer science. A general understanding of this phenomenon might profit from current contacts between logic, computer science, and game theory. [20]

What and how, logic and game equivalence Fundamental issues about gamification are close to central topics of interfaces between logic and games, as studied in van Benthem 2014. One concrete technical example is the general transition from logics of programs, possibly epistemized, to *logics of strategies* in games. [21] But perhaps most generally, an earlier issue raised at the beginning of this paper returns. A fundamental conceptual and mathematical question in the logical foundations of game theory is this:

When are two games the same?

[18] Our earlier dynamic-epistemic logics could also change relations and cut links, but their transformations were done in a *definable* manner allowing us to write recursion laws, and keep logics decidable.

[19] Ron van der Meyden, p.c., has pointed out that, while the sabotage game gamifies the original reachability *task*, there is still an additional issue of how the game solution procedure gamifies the original *algorithm* solving the task. I am not aware of systematic results yet at this second level.

[20] One should also mention the practical uses of gamification in the world of computer games, which have developed very similar aims independently. Cf. Loewe, Pacuit & Witzel 2011.

[21] Cf. also Ghosh, van Benthem & Verbrugge, eds., 2016, for a broad collection of recent papers on formal modeling of strategies coming from computer science, logic, game theory, and philosophy.

Different answers embody different views of a game as an interactive process, and hence, views of social computation. One such view is that we only need to look at the 'control' that players have over outcomes of the game, a notion close to output views of processes, and the equivalence view behind Church's Thesis. But, connecting with the preceding line, another persistent intuition concerning game equivalence has been something more intensional: viz. the possibility of simulating strategies in one game in the other game. Sometimes this can even be a brutal but very effective form of merely copying other players' moves, something that occurs both in computer science (Abramsky 2008B) and in game theory (Axelrod 1984).

But once again, one can look at all these issues from both sides, computation or human agency. Computational ideas concerning game equivalence are at the same time intuitions about the fundamental issue of what is the glue of actual social behavior. [22]

Epistemics, games, and theory of play While the above issues have pure action-oriented versions, eventually, we want to look at epistemized scenarios. In the setting of games, this brings us to the theory of imperfect information games (Osborne & Rubinstein 1994), as well as the even broader arena of epistemic game theory (Perea 2012, Brandenburger 2014, Pacuit & Roy 2015). This framework for epistemized gamified action meshes well with the dynamic-epistemic logics mentioned earlier, as these enrich game theory with an explicit analysis of the informational actions taking place during the game. Indeed, combining logic and game theory itself leads to an agency-inspired research program called a *"Theory of Play"* in van Benthem 2014, where the fine-structure is investigated of the agents playing games, including their different abilities for getting information, their different propensities for changing beliefs, and the possibly quite intricate structure of their goals. [23]

Deliberate versus automated We conclude by pointing at one further aspect of the game-theoretic connection that also applies to social agency in general. Our discussion so far may have suggested that theories of agency will get ever more elaborate, endowing agents with an ever larger repertoire of epistemic attitudes and potential for deliberation, choices, and actions. This is the main line of classical philosophical logic, and also of classical game theory. But there is also an opposite trend toward making agents as simple as possible. For instance, in evolutionary game theory (Hofbauer & Sigmund 1998, Skyrms 1990) agents are often extremely simple devices, such as finite automata, and elaborate social behavior gets largely explained as an emergent higher-level phenomenon out of very simple automated contact patterns. The border line between these two views of agency is a topic of lively current discussion, partly, because the difference between the two paradigms of explaining behavior may have sweeping implications for philosophical issues such as rationality or morality. Yet, here, we merely note one caveat that follows for our study of computation as social agency.

[22] Maybe game-theoretic notions of equivalence also have to depend on the *types of agent* playing the games, with their ways of reasoning based on combining belief and preference.

[23] An important aspect left out in our presentation of the game theory connection is *probability*. Imperfect information games have bona fide solutions in terms of Nash equilibria *in mixed strategies*, letting players play moves with certain probabilities. Also, probabilities can serve as more fine-grained descriptions of agents' knowledge and beliefs. Thus, eventually, epistemization and gamification may need foundations in terms of mixtures of logic and probability theory.

The patterns that we seek in computation as agency may require complex sophisticated deliberating agents, but they may also represent aggregate behavior where the complexity resides in how the group behaves over time, not in the individual agents. [24]

8 Conclusion

There is a famous Dutch story about the 'tide changer' who holds a ceremony at every turn of the tide to make sure it happens. There are also some doubts whether all this is necessary, since the tides might just change by themselves, because of deeper laws yet unknown, but the tide changer just does not want to take this risk. Likewise, this survey and discussion paper has identified some ongoing trends leading to a view of computation as social agency, and vice versa, that seem to be happening anyway, whether we advocate them or not.

However this may be, we hope to have added an awareness that the above process can be discerned better if we think of the recent history of computation in terms of epistemization and gamification. We also emphasized a symmetric view of computation and agency as a natural and interesting one, where both sides of the coin are attractive. It is illuminating to think of computational algorithms in terms of human agency, but just as well, it can be surprising to think of social activities in a different light than their prima facie appearance, as a process of computing. [25] We have seen how this mix has serious theoretical underpinnings raising interesting foundational questions, if we use logic as a matchmaker. In going this way, we found ourselves led to the current interface of logic, computation, game theory, and occasionally even philosophy, which may well prove a stable interdisciplinary alignment.

One theme that we have underplayed in this paper, but which, too, seems to fit naturally, is the empirical and practical dimension of all this. Computer science is changing the world, including forms of social behavior by humans. Thus, we are all part of some vast experiment seeing what happens when human behavior is affected by new devices, and the human mind by the new concepts that these devices bring along. Many of the topics discussed before lend themselves well to systematic cognitive experiments, where research into knowledge and games of the sort described above is taking off: cf. Isaac, Szymanik & Verbrugge 2014 on 'marble drop games', and Blokpoel, van Kesteren, Stolk, Haselager, Toni & van Rooij 2012 on the 'tacit communication game'. This research, too, can influence theories again. [26] [27]

[24] There are connections here with current studies of interfaces between dynamic-epistemic logics and logics for *dynamical systems*, and also with logics for analyzing evolutionary game theory, van Benthem 2015. Also relevant are major results in the theory of computation starting Büchi & Landweber 1969, which states broad circumstances under which finite automata can optimally solve classes of games with complex specifications, cf. Grädel, Thomas & Wilke, eds., 2002, Venema 2012.

[25] As just one example, cf. van Benthem & Liu 2016 on analogies between multi-agent social networks and cellular automata, and the use of social hierarchy to find new fine-structures inside computation.

[26] As an example, just put the following two publications side by side: the short-sight games' of Grossi & Turrini 2012 presented at a typical agency conference in computer science, and the 'marble drop games' with eye-tracking experiments for testing Theory of Mind and Backward Induction reasoning designed in Meijering, Taatgen, Van Rijn & Verbrugge et al. 2012.

[27] Some colleagues even believe that theoretical insight is going to *improve* society in some non-profit-seeking sense, an ambition found with some founding fathers of computer science such as Dijkstra in the 1950w, but also in recent general publications on logic and agency (Hendricks & Hansen 2014).

To close the circle of our story, however, we return to the three main themes from the Golden Age of the foundations of logic and computation in the 1930s, as stated in Section 1. What does a modern social agency perspective on computation tell us about the original grand questions about computation? We will go in reverse order, and just state the conclusions that we would draw from the material presented in this paper.

As for the *Turing Test,* the old AI program of mimicking, or even replacing, humans by machines seems tedious and, despite some unholy attractions, ultimately uninteresting. Given how the world of computation has developed in reality, the real challenge today is understanding the diverse *mixed societies of computers and humans* that have sprung up all around us, and that have vastly increased the behavioral repertoire of humans (and machines).

More tenable today is the original *Church's Thesis*. Given the close entanglement of social computation and our use of classical techniques of analysis in logic and complexity theory, we see no need to doubt its 'What' answer: the recursive functions seem fine as the extensional view of what can be computed. But this may be the less interesting question eventually, if one's aim is to understand computation as agency. As we said before, what we really want to understand is the 'How" question of what constitutes computational behavior. And if we take the social perspective of information and interaction outlined here seriously, then we have seen how some very fundamental questions are on the table: when are two social processes the same, and how do we factor in the essential role of the agents involved in them? What we really need is a convincing foundational theory of social behavior, and maybe the focus on computation of this paper will be a good way of making progress here. [28]

Finally, let us return to Turing's original contribution. The *Universal Machine* was, and remains, a crucial device for making our thinking about computation sharp, and allowing, for the first time in history, precise mathematical results on the power and limitations of what is computable. But there is a further question. Can there be a similar universal format for the behavior produced by social computation in the sense of this paper? We still lack a 'new Turing', but my guess is that the answer will come in the form of an abstract conceptual analysis of what it really means to be a *game* – beyond the details of current game theory.

9 Acknowledgements

This discussion paper goes back to an invited lecture in Manchester at the ASL Logic Colloquium in the Turing Year 2012, and follow-up versions presented at ESSLLI Opole 2012, JAIST Kanazawa 2012, the Oxford Strachey lecture 2015, and other venues. As for written sources elaborating on our presentation, the reader may consult van Benthem 2008, 2011, 2014. We thank the many audiences of colleagues and students that provided feedback. [29]

10 References

S. Abramsky, 2008A, 'Information, Processes and Games', in P. Adriaans and J. van Benthem, eds., 483–549.

[28] Admittedly, the lack of convergence to a unique view even in the restricted area of *concurrency* may be a source of some worries here. But see Abramsky 2012 for some positive answers.

[29] The present text will appear in a special issue of *Information and Computation,* edited by Julian Gutierrez and Michael Wooldridge.

S. Abramsky, 2008B, *Tutorial on Game Semantics*. ERC LINT Workshop, ILLC Amsterdam, Department of Computing, Oxford University.

S. Abramsky, 2012, 'Foundations of Interactive Computation', Invited Lecture, ASL Colloquium/Alan Turing Year, University of Manchester.

P. Adriaans & J. van Benthem, eds., *2008*, *Handbook of the Philosophy of Information*, Elsevier Science, Amsterdam.

T. Ågotnes & H. van Ditmarsch, 2011, 'What Will They Say? Public Announcement Games', *Synthese (KRA)* 179 (Suppl.1), 57–85.

G. Aucher, J. van Benthem & D. Grossi, 2015, 'Sabotage Modal Logic: Some Model- and Proof-Theoretic Aspects', in W. Holliday, W. van der Hoek & W-f Wang, eds., *Proceedings LORI 2015 Taipei*, Springer Lecture Notes in Computer Science, Heidelberg.

R. Axelrod, 1984, *The Evolution of Cooperation*, Basic Books, New York.

A. Baltag & S. Smets, 2008, 'A Qualitative Theory of Dynamic Interactive Belief Revision', in G. Bonanno, W. van der Hoek & M. Wooldridge, eds., *Logic and the Foundations of Game and Decision Theory (LOFT-7)*, Amsterdam University Press, 13–60.

J. van Benthem, 1996, *Exploring Logical Dynamics*, CSLI Publications, Stanford.

J. van Benthem, 2005, 'An Essay on Sabotage and Obstruction', in D. Hutter & W. Stephan, eds., *Mechanizing Mathematical Reasoning*, Volume 2605 of Lecture Notes in Computer Science, Springer Verlag, Berlin, 268–276.

J. van Benthem, 2007, 'Rational Dynamics and Epistemic Logic in Games', *International Game Theory Review* 9:1, 2007, 13–45. Erratum reprint, Volume 9:2, 377–409.

J. van Benthem, 2008, 'Computation as Conversation', in S. Cooper et al., eds., 35–58.

J. van Benthem, 2011, *Logical Dynamics of Information and Interaction*, Cambridge University Press, Cambridge UK.

J. van Benthem, 2013, 'Reasoning about Strategies', in B. Coecke, L. Ong, and P. Panangaden, eds., *Computation, Logic, Games, and Quantum Foundations*, Volume 7860 of Lecture Notes in Computer Science, Springer, Berlin, 336–347.

J. van Benthem, 2014, *Logic in Games*, The MIT Press, Cambridge MA.

J. van Benthem, J. van Eijck & B. Kooi, 2006, 'Logics of Communication and Change', *Information and Computation* 204, 1620–1662.

J. van Benthem, J. Gerbrandy, T. Hoshi & E. Pacuit, 2009, 'Merging Frameworks for Interaction', *Journal of Philosophical Logic 38,* 2009, 491–526.

J. van Benthem & M. Martinez, 2008, 'The Stories of Logic and Information', in P. Adriaans & J. van Benthem, eds., 217–280.

J. van Benthem & E. Pacuit, 2006, 'The Tree of Knowledge in Action', *Proceedings Advances in Modal Logic,* ANU Melbourne, 87–106.

J. van Benthem & E. Pacuit, 2011, 'Dynamic Logic of Evidence-Based Beliefs', *Studia Logica* 99:1, 61–92.

M. Blokpoel, M. van Kesteren, A. Stolk, P. Haselager, I. Toni & I. van Rooij, 2012, 'Recipient Design in Human Communication: Simple Heuristics or Perspective Taking?', *Frontiers in Human Neuroscience* 6:253, 1–13.

T. Bolander & M. B. Andersen, 2011, 'Epistemic Planning for Single and Multi-Agent Systems', *Journal of Applied Non-Classical Logics* 21: 1, 9–34.

J. A. Bergstra, A. Ponse & S. A. Smolka, eds., 2001, *Handbook of Process Algebra*, Elsevier Science, Amsterdam.

R. Brafman, J-C Latombe, and Y. Shoham, 1993, Towards Knowledge-Level Analysis of Motion Planning, *Proceedings AAAI 1993*, 670-675.

A. Brandenburger, 2014, *The Language of Game Theory: Putting Epistemics into the Mathematics of Games*, World Scientific, Singapore.

J. Büchi & L. Landweber, 1969, 'Definability in the Monadic Second-Order Theory of Successor', Journal Symbolic Logic 34:2, 166–170.

Z. Christoff, 2016, *Dynamic Logics of Networks: Information Flow and Spread of Opinion*, Dissertation, Institute for Logic, Language and Computation, University of Amsterdam.

S. B. Cooper, B. Löwe & A. Sorbi., eds., 2008, *New Computational Paradigms, Changing Conceptions of What is Computable*, Springer, New York.

S. B. Cooper, 2011, 'Computability Theory', in J. van Benthem & A. Gupta, ed., *Logic and Philosophy Today*, Volume 1, College Publications, London, 197–218.

H. van Ditmarsch, 2002, 'The Russian Cards Problem: a case study in cryptography with public announcements', *Proceedings of AWCL 2002* (Australasian Workshop on Computational Logic), Canberra, 47–67.

H. van Ditmarsch, W. van der Hoek & B. Kooi, 2007, *Dynamic Epistemic Logic*, Springer Publishers, Dordrecht.

E. W. Dijkstra, 1968, 'Goto Statement Considered Harmful', *Communications of the ACM*, Vol. 11, No. 3, 147–148.

P. van Emde Boas, 1990, 'Machine Models and Simulations', in J. van Leeuwen, ed., *Handbook of Theoretical Computer Science*, vol. A, Algorithms and Complexity, Elsevier Science, Amsterdam, 3–66.

U. Endriss & J. Lang, eds., 2006, *Proceedings of the 1st International Workshop on Computational Social Choice (COMSOC-2006),* ILLC, University of Amsterdam.

R. Fagin, J. Halpern, M. Vardi & Y. Moses, 1995, *Reasoning about Knowledge*, The MIT Press, Cambridge MA.

V. Fiutek, 2013, *Playing with Knowledge and Belief*, Dissertation DS-2-13-02, ILLC, University of Amsterdam.

D. Gabbay, T. Maibaum & J. Shepherdson, eds., 1995, *Handbook of Logic in Artificial Intelligence and Logic Programming*, Oxford University Press, Oxford.

S. Ghosh, J. van Benthem & R. Verbrugge, eds., 2016, *Models of Strategic Reasoning: Logics, Games, and Communities*, FoLLI Lecture Notes, Springer, Heidelberg.

N. Gierasimczuk, *Knowing One's Limits*, *Logical Analysis of Inductive Inference*, Dissertation DS-2010-11, Institute for Logic, Language and Computation, University of Amsterdam.

G. Gigerenzer, Peter M. Todd, and the ABC Research Group, 1999, *Simple Heuristics That Make Us Smart*, Oxford University Press.

P. Girard, F. Liu & J. Seligman, 2014, 'Logical Dynamics of Belief Change in the Community', *Synthese* 191:11, 2402–2431.

E. Grädel, W. Thomas, and Th. Wilke, eds., 2002, *Automata, Logics, and Infinite Games*, Lecture Notes in Computer Science, Springer Verlag, Heidelberg.

D. Grossi & P. Turrini, 2012, 'Short-Sight in Extensive Games', in V. Conitzer & M. Winikoff, Eds., *Proceedings of the 11th International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS),* 805–812.

J. Halpern & M. Vardi, 1989, 'The Complexity of Reasoning about Knowledge and Time, I. Lower Bounds, *Journal of Computer and System Sciences* 38(1), 195–237.

D. Harel, D. Kozen & J. Tiuryn, 2000, *Dynamic Logic*, The MIT Press, Cambridge MA.
V. Hendricks & P. Hansen, 2014, *Infostorms*, Copernicus Books & Springer, New York.
J. Hintikka & G. Sandu, 1997, 'Game-Theoretical Semantics', in J. van Benthem & A. ter Meulen, eds., *Handbook of Logic and Language*, Elsevier, Amsterdam, 361–410.
J. Hofbauer & K. Sigmund, 1998, *Evolutionary Games and Population Dynamics*, Cambridge University Press, Cambridge UK.
W. Holliday, 2012, *Knowing What Follows: Epistemic Closure and Epistemic Logic*, Ph.D. thesis, Department of philosophy, Stanford University. Appeared in ILLC Dissertation Series DS-2012-09, University of Amsterdam.
A. Isaac, A., J. Szymanik & R. Verbrugge, 2014, 'Logic and Complexity in Cognitive Science', In A. Baltag & S. Smets, eds., *Johan van Benthem on Logic and Information Dynamics*, Springer, Dordrecht, 787–833.
N. D. Jones, 1978, 'Blindfold Games are Harder than Games with Perfect Information', *Bulletin of the EATCS* 6, 4–7.
K. Lehrer, 1990, *Theory of Knowledge*, Westview Press, Boulder CO.
K. Leyton-Brown & Y. Shoham, 2009, *Multiagent Systems: Algorithmic, Game Theoretic and Logical Foundations*, Cambridge University Press, Cambridge UK.
F. Liu, 2011, *Dynamic Logic of Preference Change*, Springer Publishers, Dordrecht.
B. Loewe, E. Pacuit & A. Witzel, 2011, 'DEL Planning and some Tractable Cases', in H. van Ditmarsch, J. Lang & S. Ju, eds., *Proceedings LORI Guangzhou 2011*, Springer, Heidelberg, 179–192.
B. Meijering, H. van Rijn, N. Taatgen & R. Verbrugge, 2012, 'What Eye Movements Can Tell about Theory of Mind in a Strategic Game', *PLoS ONE 7(9):* e45961. doi:10.1371/journal.pone.0045961
J. Miller & L. Moss, 2005, 'The Undecidability of Iterated Modal Relativization', *Studia Logica* 97, 373–407.
S. Minica, 2011, *Dynamic Logic of Questions*, Dissertation DS-2011-08, Institute for Logic, Language and Computation, University of Amsterdam, Amsterdam, The Netherlands.
Y. Moses, 2015, 'Relating Knowledge and Coordinated Action: The Knowledge of Preconditions Principle', *Proceedings TARK 2015*, CMU Pittsburgh.
R. Nozick, 1981, *Philosophical Explanations*, Harvard University Press, Cambridge (Mass.).
M. Osborne & A. Rubinstein, 1994, *A Course in Game Theory*, The MIT Press, Cambridge MA.
E. Pacuit & O. Roy, 2015, *Interactive Rationality*, Lecture Notes, Department of Philosophy, University of Maryland and Institute for Philosophy, University of Bayreuth.
C. Papadimitriou, 1994, *Computational Complexity*, Addison-Wesley, Reading.
A. Perea, 2012, *Epistemic Game Theory: Reasoning and Choice*, Cambridge University Press, Cambridge UK.
P. Rohde, 2005, *On Games and Logics over Dynamically Changing Structures*, Ph. D. thesis, Rheinisch-Westfälische Technische Hochschule, Aachen.
M. Sevenster, 2006, *Branches of Imperfect Information: Logic, Games, and Computation*, Ph. D. thesis, Institute for Logic, Language and Computation, University of Amsterdam. ILLC Dissertation series DS-2006-06.

B. Skyrms, 1990, *The Dynamics of Rational Deliberation*, Harvard University Press, Cambridge MA.

J. Stanley, 2011, *Know How*, Oxford University Press, Oxford.

K. Su, A. Sattar, G. Governatori & Q. Chen, 2005, 'A Computationally Grounded Logic of Knowledge, Belief and Certainty', *Proceedings Fourth International Joint Conference on Autonomous Agents and Multiagent Systems*, AAMAS '05, 149–156.

A. Turing, 1937, 'On Computable Numbers, with an Application to the *Entscheidungs-problem*, *Proceedings of the London Mathematical Society*, series 2, 42, p. 230–65.

Y. Venema, 2006, 'Algebras and Co-Algebras', in P. Blackburn, J. van Benthem & F. Wolter, eds., *Handbook of Modal Logic*, Elsevier Science, Amsterdam, 331–426.

Y. Venema, 2012, *Lectures on the Modal μ–Calculus*, Institute for Logic, Language and Computation, University of Amsterdam.

T. Williamson, 2000, *Knowledge and its Limits*, Oxford University Press, Oxford.

M. Wooldridge, 2009, *An Introduction to Multi-Agent Systems*, John Wiley Sons, New York.