Knowledge and Probability in Distributed Systems: Abstract

Joseph Y. Halpern

IBM Almaden Research Center
San Jose, CA 95120
email: halpern@ibm.com

Epistemology, the study of knowledge, has a long and honorable tradition in philosophy, starting with the Greeks on through the formal analysis of Hintikka [Hin62] and his successors in the 1960's and 70's. It will not come as a great surprise that epistemlogy has also been of interest to researchers in Artificial Intelligence. After all, being able to represent and reason about knowledge is a key attribute of an intelligent system. Perhaps it is more surprising that reasoning about knowledge should have relevance to distributed computing. The goals of my talk are (a) to convince you that it does indeed have great relevance, (b) to show how distributed systems can be analyzed in terms of knowledge, and (c) to show how our tools for analyzing protocols in terms of knowledge can be extended to handle probability. The talk is essentially a summary of material from two papers: [HM90] and [HT89]. In this abstract, I will (very briefly!) sketch a few of the details, and provide references to other work in applying tools of epistemic logic to distributed computing.

Philosophers have used the *possible worlds* framework to analyze knowledge, belief, and other modaliities. Roughly speaking, the idea is that each agent imagines a number of *possible worlds*, that is, ways the world could be. An agent knows a fact φ if φ is true in all the worlds an agent considers possible. This is formalized by means of a *Kripke structure*. (The reader unfamiliar with Kripke structures can consult references such as [Che80, HC78, HM85] for details.)

Work on applying epistemic logic to distributed computing started roughly seven years ago, with [HM90]. In the context of distributed systems, what matters is not just the knowledge of one agent about the state of nature, but also the knowledge an agent has about other agents' knowledge, and the state of the group's knowledge. In particular, it was shown in [HM90] that the state of knowledge known as *common knowledge* (where φ is said to be common knowledge if everyone knows φ, everyone knows that everyone knows φ, etc.) was critical for reaching agreement. It was also demonstrated that common knowledge was not attainable under some minimal assumptions, and attainable variants of common knowledge were identified. The themes of this paper have been pursued in numerous papers (for example, [CM86, DM90, FI86, Had87, HF89, HMW90, HZ87, Maz88, MT88, Nei88a, NT87, PT88, PR85]; see [Hal87] for an overview) and Ph.D. theses [Maz89, Mic89, Mos86, Nei88b, Tut89].

The model of knowledge used in most papers that consider formalizing knowledge in distributed systems [CM86, FI86, HF89, HM90, PR85] is quite straightforward. We assume that we have a system of n processes, communicating with each other. We further assume that if we look at the system at any point in time, each of the processes is in some *local state*. The tuple consisting of each process' local state at a given time is called the *global state* of the system. (Occasionally it is also useful to add a component describing the state of the *environment*, which includes all information about the system not contained in the state of the processes, such as information about messages in transit.) A *run* is a complete description of the system's behavior over time in one possible execution of the system. Formally, it is a function from time to global states. A system is defined to be a set of runs; intuitively, this set describes all the possible executions of the system. We call

a pair (r, m) consisting of a time r and a time m a *point*. Thus, at a point (r, m), the system is in global state $r(m)$. We can view the points of the system as the possible worlds. At a point (world) (r, m), we say that process i considers the point (r', m') possible if process i is in the same local state at the global states $r(m)$ and $r'(m')$. Intuitively, process i cannot distinguish the points (r, m) and (r', m') because it has the same information at both points. It is easy to see that process i's possibility relation as defined above is an equivalence relation. We can now apply the possible-worlds definition of knowledge in a straightforward way: a process i knows a fact φ at a point (r, m) if φ holds at all points (r', m') that i considers possible at (r, m), namely, all points where i has the same local state as at (r, m). This definition of knowledge in distributed systems can easily be shown to satisfy the axioms of the well-known modal logic S5 [Che80]. More importantly, it corresponds to one important way that the word knowledge has been informally among systems designers. When a system designer says "we cannot terminate the protocol at this point because process 2 does not know that process 3 received the value of x", the word "know" is being used in a way completely consistent with this definition.

We remark that although this definition seems geared to distributed systems, it easily translates to other contexts. The processes can be agents in a bargaining session, robots trying to carry out some task, or even the wires in a digital circuit; the definitions go through perfectly well. Perhaps not surprisingly, definitions related to the one above have appeared in other disciplines. Rosenschein and his collaborators [Ros85, RK86] adopted it for analyzing digital machines. In the context of game theory, knowledge of what other agents know is clearly crucial. *Game trees* [Ras89] can be viewed as essentially defining a system: each path in the game tree roughly corresponds to a run. Game theorists talk about *information sets*: these are precisely the sets of nodes in the game tree that a given agent cannot distinguish. They use these to define notions of knowledge that are essentially identical to those defined above. These notions of knowledge (including common knowledge) have been the subject of a great deal of research in the economics community, starting with the publication of Aumann's seminal paper [Aum76].

Philosophers have presented criticisms against S5 as an appropriate model for knowledge (see [Len78] for details and an overview of the philosophers' work in the 1960's and 70's). While these are legitimate criticisms, I would argue that the notion of knowledge being used in distributed systems is a useful one, and does reflect one way the word is in practice. Moreover, as the papers cited above show, it is a *useful* notion; it can be used to help us better understand, analyze, and design distributed algorithms.

Nevertheless, it is clear that this notion of knowledge is not adequate for all applications. One extension that has been considered, which I shall not discuss further here, is trying to define a notion of *computable* knowledge, that takes the complexity of computing knowledge into account in a reasonable way; see [Mos88, HMT88] for further details along these lines. Another extension, which I shall briefly discuss below, is that of incorporating probability into the framework.

Given the importance of randomized algorithms and, more generally, the need to reason about probability in many of the applications involving reasoning about knowledge, trying to combine reasoning about knowledge and about probability was an obvious step to take. It seems that it should be relatively straightforward to combine knowledge and probability in a possible-worlds framework. We simply view the set of worlds that an agent considers possible as a probability space. However, there are some subtleties, particularly those involving the interaction of *probabilistic* and *non-probabilistic* events. For example, when analyzing a randomized algorithm, the probabilistic events are the outcomes of the coin tosses. For these, it makes perfect sense to assign a probability (for example, if a fair coin is tossed, it seems reasonable to assign probability 1/2 to the set of worlds where the coin lands heads). Events such as input values, for which we are typically not given a probability, can be viewed as non-probabilistic. For such events it is inappropriate (at least, at this level of analysis) to assume probabilities. When proving that a randomized algorithm such as Rabin's primality-testing algorithm [Rab80] is correct, we do not want to assume that there is a probability that the input will be 127,531. We want to prove the algorithm gives the correct answer with high probability (taken over the coin tosses) independent of the probability on the inputs.

Arguments can be made that the best (and arguably most natural) way of doing this is to split up the state space (in this case, the set of points) into a number of separate probability spaces, essentially one corresponding to each possible input [FZ88, HT89]. While this approach may seem somewhat *ad hoc*, as shown in [HT89], it can be given a natural interpretation. It corresponds to playing different adversaries, with different knowledge. This point is perhaps best clarified by considering the following situation. Suppose agent 1 tosses a fair coin. He is observed by agents 2 and 3, who do not know the outcome of the toss, but do know that the coin is fair. What probability should agent 2 assign to the event "heads" after the coin has landed? There are two standard answers to this question. One is that the probability is 1/2; after all, it was 1/2 before the coin was tossed, and agent 2 has not learned anything that would cause him to change his opinion regarding the probability. The other answer is that since the coin has landed, the event has already occurred. It doesn't make sense to say it has probability 1/2; the probability is either 0 or 1, but agent 2 does not know which.

Another way of thinking about this is that the probability of heads is not well defined until we specify which adversary agent 2 is playing against. If agent 2 is playing against agent 3, since both 2 and 3 have no idea of the outcome of the coin toss, the appropriate probability for agent 2 to assign is 1/2; on the other hand, when playing agent 1, who presumably knows the outcome of the coin toss, then the best agent 2 can say is that he knows the probability is either 0 or 1, but does not know which. The connection to splitting up a probability space into subspaces is relatively straightforward in this case. The answer 1/2 clearly comes from considering a probability space consisting of two points (one corresponding to the events heads and one corresponding to the event tails). If we split up this probability space into two subspaces, the event "heads" has probability 1 in one of them (the one consisting of just the point corresponding to heads) and 0 in the other. All these points are formalized in [HT89].

The advantage of thinking in terms of adversaries is that it closely resembles the way researchers in cryptography, game theory, and distributed systems already analyze many situations. The analysis in [HT89] is only a first step; much more work remains to be done in terms of considering specific classes of adversaries. In addition, much more work needs to be done to apply these ideas to analyzing more protocols, both probabilistic and non-probabilistic.

References

[Aum76] R. J. Aumann. Agreeing to disagree. *Annals of Statistics*, 4(6):1236–1239, 1976.

[Che80] B. F. Chellas. *Modal Logic.* Cambridge University Press, 1980.

[CM86] K. M. Chandy and J. Misra. How processes learn. *Distributed Computing*, 1(1):40–52, 1986.

[DM90] C. Dwork and Y. Moses. Knowledge and common knowledge in a Byzantine environment: crash failures. *Information and Computation*, 88(2):156–186, 1990.

[FH88] R. Fagin and J. Y. Halpern. Reasoning about knowledge and probability: preliminary report. In M. Y. Vardi, editor, *Proceedings of the Second Conference on Theoretical Aspects of Reasoning about Knowledge*, pages 277–293. Morgan Kaufmann, 1988.

[FI86] M. J. Fischer and N. Immerman. Foundations of knowledge for distributed systems. In J. Y. Halpern, editor, *Theoretical Aspects of Reasoning about Knowledge: Proceedings of the 1986 Conference*, pages 171–186. Morgan Kaufmann, 1986.

[FZ88] M. J. Fischer and L. D. Zuck. Reasoning about uncertainty in fault-tolerant distributed systems. Technical Report YALEU/DCS/TR-643, Yale University, 1988.

[GMR89] S. Goldwasser, S. Micali, and C. Rackoff. The knowledge complexity of interactive proof systems. *SIAM Journal on Computing*, 18(1):186–208, February 1989.

[Had87] V. Hadzilacos. A knowledge-theoretic analysis of atomic commitment protocols. In *Proc. 6th ACM Symp. on Principles of Database Systems*, pages 129–134, 1987. A revised version has been submitted for publication.

[Hal87] J. Y. Halpern. Using reasoning about knowledge to analyze distributed systems. In J. Traub et al., editors, *Annual Review of Computer Science, Vol. 2*, pages 37–68. Annual Reviews Inc., 1987.

[HC78] G. E. Hughes and M. J. Cresswell. *An Introduction to Modal Logic.* Methuen, 1978.

[HF89] J. Y. Halpern and R. Fagin. Modelling knowledge and action in distributed systems. *Distributed Computing*, 3(4):159–179, 1989.

[Hin62] J. Hintikka. *Knowledge and Belief.* Cornell University Press, 1962.

[HM85] J. Y. Halpern and Y. Moses. A guide to the modal logics of knowledge and belief. In *Ninth International Joint Conference on Artificial Intelligence (IJCAI-85)*, pages 480–490, 1985.

[HM90] J. Y. Halpern and Y. Moses. Knowledge and common knowledge in a distributed environment. *Journal of the ACM*, 37(3):549–587, 1990. An early version appeared in *Proceedings of the 3rd ACM Symposium on Principles of Distributed Computing*, 1984.

[HMT88] J. Y. Halpern, Y. Moses, and M. R. Tuttle. A knowledge-based analysis of zero knowledge. In *Proc. 20th ACM Symp. on Theory of Computing*, pages 132–147, 1988.

[HMW90] J. Y. Halpern, Y. Moses, and O. Waarts. A characterization of eventual Byzantine agreement. In *Proc. 9th ACM Symp. on Principles of Distributed Computing*, pages 333–346, 1990.

[HT89] J. Y. Halpern and M. R. Tuttle. Knowledge, probability, and adversaries. In *Proc. 8th ACM Symp. on Principles of Distributed Computing*, pages 103–118, 1989.

[HZ87] J. Y. Halpern and L. D. Zuck. A little knowledge goes a long way: Simple knowledge-based derivations and correctness proofs for a family of protocols. In *Proc. 6th ACM Symp. on Principles of Distributed Computing*, pages 269–280, 1987. A revised and expanded version appears as IBM Research Report RJ 5857, 1987 and will appear in *Journal of the ACM.*

[Len78] W. Lenzen. Recent work in epistemic logic. *Acta Philosophica Fennica*, 30:1–219, 1978.

[Maz88] M. S. Mazer. A knowledge theoretic account of recovery in distributed systems: the case of negotiated commitment. In M. Y. Vardi, editor, *Proceedings of the Second Conference on Theoretical Aspects of Reasoning about Knowledge*, pages 309–324. Morgan Kaufmann, 1988.

[Maz89] M. S. Mazer. *A knowledge-theoretic account of negotiated commitment.* PhD thesis, University of Toronto, 1989.

[Mic89] R. Michel. *Knowledge in distributed Byzantine environments.* PhD thesis, Yale University, 1989.

[Mos86] Y. Moses. *Knowledge in a distributed environment.* PhD thesis, Stanford University, 1986.

[Mos88] Y. Moses. Resource-bounded knowledge. In M. Y. Vardi, editor, *Proceedings of the Second Conference on Theoretical Aspects of Reasoning about Knowledge*, pages 261–276. Morgan Kaufmann, 1988.

[MT88] Y. Moses and M. R. Tuttle. Programming simultaneous actions using common knowledge. *Algorithmica*, 3:121–169, 1988.

[Nei88a] G. Neiger. Knowledge consistency: a useful suspension of disbelief. In M. Y. Vardi, editor, *Proceedings of the Second Conference on Theoretical Aspects of Reasoning about Knowledge*, pages 295–308. Morgan Kaufmann, 1988.

[Nei88b] Gil Neiger. *Techniques for Simplifying the Design of Distributed Systems.* PhD thesis, Cornell University, August 1988. Department of Computer Science Technical Report 88-933.

[NT87] G. Neiger and S. Toueg. Substituting for real time and common knowledge in asynchronous distributed systems. In *Proc. 6th ACM Symp. on Principles of Distributed Computing*, pages 281–293, 1987. To appear, *Journal of the ACM.*

[PR85] R. Parikh and R. Ramanujam. Distributed processing and the logic of knowledge. In R. Parikh, editor, *Proc. of the Workshop on Logics of Programs*, pages 256–268, 1985.

[PT88] P. Panangaden and S. Taylor. Concurrent common knowledge: A new definition of agreement for asynchronous systems. In *Proc. 7th ACM Symp. on Principles of Distributed Computing*, pages 197–209, 1988.

[Rab80] M. O. Rabin. Probabilistic algorithm for testing primality. *Journal of Number Theory*, 12:128–138, 1980.

[Ras89] E. Rasmusen. *Games and Information: An Introduction to Game Theory.* Basil Blackwell, 1989.

[RK86] S. J. Rosenschein and L. P. Kaelbling. The synthesis of digital machines with provable epistemic properties. In J. Y. Halpern, editor, *Theoretical Aspects of Reasoning about Knowledge: Proceedings of the 1986 Conference*, pages 83–97. Morgan Kaufmann, 1986.

[Ros85] S. J. Rosenschein. Formal theories of AI in knowledge and robotics. *New Generation Computing*, 3:345–357, 1985.

[Tut89] M. R. Tuttle. *Knowledge and distributed computation.* PhD thesis, MIT, 1989.