Knowledge and Common Knowledge in a Distributed Environment*

Joseph Y. Halpern

IBM Almaden Research Center
San Jose, CA 95120

Yoram Moses

Department of Applied Mathematics
The Weizmann Institute of Science
Rehovot, 76100 ISRAEL

Abstract: Reasoning about knowledge seems to play a fundamental role in distributed systems. Indeed, such reasoning is a central part of the informal intuitive arguments used in the design of distributed protocols. Communication in a distributed system can be viewed as the act of transforming the system's state of knowledge. This paper presents a general framework for formalizing and reasoning about knowledge in distributed systems. We argue that states of knowledge of groups of processors are useful concepts for the design and analysis of distributed protocols. In particular, *distributed knowledge* corresponds to knowledge that is "distributed" among the members of the group, while *common knowledge* corresponds to a fact being "publicly known". The relationship between common knowledge and a variety of desirable actions in a distributed system is illustrated. Furthermore, it is shown that, formally speaking, in practical systems common knowledge cannot be attained. A number of weaker variants of common knowledge that are attainable in many cases of interest are introduced and investigated.

*This is a revised and expanded version of a paper with the same title that first appeared in the *Proceedings of the 3rd ACM Conference on Principles of Distributed Computing, 1984*. It is essentially identical to the version that appears in *Journal of the ACM* **37**:3, 1990, pp. 549–587. The work of the second author was supported in part by DARPA contract N00039-82-C-0250.

1 Introduction

Distributed systems of computers are rapidly gaining popularity in a wide variety of applications. However, the distributed nature of control and information in such systems makes the design and analysis of distributed protocols and plans a complex task. In fact, at the current time, these tasks are more an art than a science. Basic foundations, general techniques, and a clear methodology are needed to improve our understanding and ability to deal effectively with distributed systems.

While the tasks that distributed systems are required to perform are normally stated in terms of the global behavior of the system, the actions that a processor performs can depend only on its local information. Since the design of a distributed protocol involves determining the behavior and interaction between individual processors in the system, designers frequently find it useful to reason intuitively about processors' "states of knowledge" at various points in the execution of a protocol. For example, it is customary to argue that "... once the sender receives the acknowledgement, it *knows* that the current packet has been delivered; it can then safely discard the current packet, and send the next packet...". Ironically, however, formal descriptions of distributed protocols, as well as actual proofs of their correctness or impossibility, have traditionally avoided any explicit mention of knowledge. Rather, the intuitive arguments about the state of knowledge of components of the system are customarily buried in combinatorial proofs that are unintuitive and hard to follow.

The general concept of knowledge has received considerable attention in a variety of fields, ranging from Philosophy [Hin62] and Artificial Intelligence [MSHI79] and [Moo85], to Game Theory [Aum76] and Psychology [CM81]. The main purpose of this paper is to demonstrate the relevance of reasoning about knowledge to distributed systems as well. Our basic thesis is that explicitly reasoning about the states of knowledge of the components of a distributed system provides a more general and uniform setting that offers insight into the basic structure and limitations of protocols in a given system.

As mentioned above, agents can only base their actions on their local information. This knowledge, in turn, depends on the messages they receive and the events they observe. Thus, there is a close relationship between knowledge and action in a distributed environment. When we consider the task of performing coordinated actions among a number of agents in a distributed environment, it does not, in general, suffice to talk only about individual agents' knowledge. Rather, we need to look at states of knowledge of *groups* of agents (the group of all participating agents is often the most relevant one to consider). Attaining particular states of group knowledge is a prerequisite for performing coordinated actions of various kinds.

In this work we define a hierarchy of states of group knowledge. It is natural to think of communication in the system as the act of improving the state of knowledge, in the sense of "climbing up the hierarchy". The weakest state of knowledge we discuss is *distributed knowledge*, which corresponds to knowledge that is distributed among the members of the group, without any individual agent necessarily having it.[1] The strongest state of knowledge in the hierarchy is *common knowledge*, which roughly corresponds to "public knowledge". We show that the execution of simultaneous actions becomes common knowledge, and hence that such actions cannot be performed if common knowledge cannot be attained. Reaching agreement is an important example of a desirable simultaneous action in a distributed environment. A large part of the technical analysis in this paper is concerned with the ability and cost of attaining common knowledge in systems of various types. It turns out that attaining common knowledge in distributed environments is not a simple task. We show that when communication is not guaranteed it is impossible to attain common knowledge. This generalizes the impossibility of a solution to the well-known *coordinated attack* problem [Gra78]. A more careful analysis shows that common knowledge can only be attained in systems that support simultaneous coordinated actions. It can be shown that such actions cannot be guaranteed or detected in practical distributed systems. It follows that common knowledge cannot be attained in many cases of interest. We then consider states of knowledge that correspond to eventually coordinated actions and to coordinated actions that are guaranteed to be performed within a bounded amount of time. These are essentially weaker variants of common knowledge. However, whereas, strictly speaking, common knowledge may be difficult to attain in many practical cases, these weaker states of knowledge are attainable in cases of interest.

Another question that we consider is that of when it is safe to assume that certain facts are common knowledge, even when strictly speaking they are not. For this purpose, we introduce the concept of *internal knowledge consistency.* Roughly speaking, it is internally knowledge consistent to assume that a certain state of knowledge holds at a given point, if nothing the processors in the system will ever encounter will be inconsistent with this assumption.

The rest of the paper is organized as follows. In the next section we look at the "muddy children" puzzle, which illustrates some of the subtleties involved in reasoning about knowledge in the context of a group of agents. In Section 3 we introduce a hierarchy of states of knowledge in which a group may be. Section 4 focuses on the relationship between knowledge and communication by looking at the coordinated attack problem. In Section 5 we sketch a general definition of a distributed system, and in Section 6 we discuss how knowledge can be ascribed to processors in such systems so as to make statements such as "agent 1 *knows* φ" completely formal and precise. Section 7 relates common knowledge to the coordinated attack problem. In Section 8, we show that, strictly speaking, common knowledge cannot be attained in practical distributed systems. Section 9 considers the implications of this observation and in Section 10 we begin to reconsider the notion of common knowledge in the light of these implications. In Sections 11 and 12, we consider a number of variants of common knowledge that are

[1] In a previous version of this paper [HM90], what we are now calling distributed knowledge was called implicit knowledge. We have changed the name here to avoid conflict with the usage of the phrase "implicit knowledge" in papers such as [FH88, Lev84].

attainable in many cases of interest and discuss the relevance of these states of knowledge to the actions that can be performed in a distributed system. Section 13 discusses the notion of internal knowledge consistency, and Section 14 contains some concluding remarks.

2 The muddy children puzzle

A crucial aspect of distributed protocols is the fact that a number of different processors cooperate in order to achieve a particular goal. In such cases, since more than one agent is present, an agent may have knowledge about other agents' knowledge in addition to his knowledge about the physical world. This often requires care in distinguishing subtle differences between seemingly similar states of knowledge. A classical example of this phenomenon is the muddy children puzzle – a variant of the well known "wise men" or "cheating wives" puzzles. The version given here is taken from [Bar81]:

> Imagine n children playing together. The mother of these children has told them that if they get dirty there will be severe consequences. So, of course, each child wants to keep clean, but each would love to see the others get dirty. Now it happens during their play that some of the children, say k of them, get mud on their foreheads. Each can see the mud on others but not on his own forehead. So, of course, no one says a thing. Along comes the father, who says, "At least one of you has mud on your head," thus expressing a fact known to each of them before he spoke (if $k > 1$). The father then asks the following question, over and over: "Can any of you prove you have mud on your head?" Assuming that all the children are perceptive, intelligent, truthful, and that they answer simultaneously, what will happen?

The reader may want to think about the situation before reading the rest of Barwise's discussion:

> There is a "proof" that the first $k-1$ times he asks the question, they will all say "no" but then the kth time the dirty children will answer "yes."
>
> The "proof" is by induction on k. For $k = 1$ the result is obvious: the dirty child sees that no one else is muddy, so he must be the muddy one. Let us do $k = 2$. So there are just two dirty children, a and b. Each answers "no" the first time, because of the mud on the other. But, when b says "no," a realizes that he must be muddy, for otherwise b would have known the mud was on his head and answered "yes" the first time. Thus a answers "yes" the second time. But b goes through the same reasoning. Now suppose $k = 3$; so there are three dirty children, a, b, c. Child a argues as follows. Assume I don't have mud on my head. Then, by the $k = 2$ case, both b and c will answer

"yes" the second time. When they don't, he realizes that the assumption was false, that he is muddy, and so will answer "yes" on the third question. Similarly for b and c. [The general case is similar.]

Let us denote the fact "At least one child has a muddy forehead" by $\mathbf{m}$. Notice that if $k > 1$, i.e., more than one child has a muddy forehead, then every child can see at least one muddy forehead, and the children initially all know $\mathbf{m}$. Thus, it would seem, the father does not need to tell the children that $\mathbf{m}$ holds when $k > 1$. But this is false! In fact, had the father not announced $\mathbf{m}$, the muddy children would never have been able to conclude that their foreheads are muddy. We now sketch a proof of this fact.

First of all, given that the children are intelligent and truthful, a child with a clean forehead will never answer "yes" to any of the father's questions. Thus, if $k = 0$, all of the children answer all of the father's questions "no". Assume inductively that if there are exactly k muddy children and the father does not announce $\mathbf{m}$, then the children all answer "no" to all of the father's questions. Note that, in particular, when there are exactly k muddy foreheads, a child with a clean forehead initially sees k muddy foreheads and hears all of the father's questions answered "no". Now assume that there are exactly $k + 1$ muddy children. Let $q \geq 1$ and assume that all of the children answer "no" to the father's first $q - 1$ questions. We have argued above that a clean child will necessarily answer "no" to the father's q^{th} question. Next observe that before answering the father's q^{th} question, a muddy child has exactly the same information as a clean child has at the corresponding point in the case of k muddy foreheads. It follows that the muddy children must all answer "no" to the father's q^{th} question, and we are done. (A very similar proof shows that if there are k muddy children and the father does announce $\mathbf{m}$, his first $k - 1$ questions are answered "no".)

So, by announcing something that the children all know, the father somehow manages to give the children useful information! How can this be? Exactly what *was* the role of the father's statement? In order to answer this question, we need to take a closer look at knowledge in the presence of more than one knower; this is the subject of the next section.

3 A hierarchy of states of knowledge

In order to analyze the muddy children puzzle introduced in the previous section, we need to consider states of knowledge of groups of agents. As we shall see in the sequel, reasoning about such states of knowledge is crucial in the context of distributed systems as well. In Section 6 we shall carefully define what it means for an agent i to know a given fact φ (which we denote by $K_i\varphi$). For now, however, we need knowledge to satisfy only two properties. The first is that an agent's knowledge at a given time must depend only on its local history: the information that it started out with combined with the events it has observed since then. Secondly, we require that only true things be known, interpretation for R, and let $|G| \geq 2$. Then for all runs $r \in R$, times t, and formulas φ it is the case that $(\mathcal{I}, r, t) \models C_G\varphi$ iff $(\mathcal{I}, r, 0) \models C_G\varphi$.

Since practical systems turn out to have temporal imprecision, Theorem 8 implies that, strictly speaking, common knowledge cannot be attained in practical distributed systems! In such systems, we have the following situation: a fact φ can be known to a processor without being common knowledge, or it can be common knowledge (in which case that processor also knows φ), but due to (possibly negligible) imperfections in the system's state of synchronization and its communication medium, there is no way of getting from the first situation to the second! Note that if there is a global clock, then there cannot be any temporal imprecision. Thus, it is consistent with Theorem 8 that common knowledge is attainable in a system with a global clock.

Observe that we can now show that, formally speaking, even people cannot attain common knowledge of any new fact! Consider the father publicly announcing $\mathbf{m}$ to the children in the muddy children puzzle. Even if we assume that it is common knowledge that the children all hear whatever the father says and understand it, there remains some uncertainty as to exactly when each child comes to know (or comprehend) the father's statement. Thus, it is easy to see that the children do not immediately have common knowledge of the father's announcement. Furthermore, for similar reasons the father's statement can never become common knowledge.

9 A paradox?

There is a close correspondence between agreements, coordinated actions, and common knowledge. We have argued that in a precise sense, reaching agreements and coordinating actions in a distributed system requires attaining common knowledge of certain facts. However, in the previous section we showed that common knowledge *cannot be attained* in practical distributed systems! We are faced with a seemingly paradoxical situation on two accounts. First of all, these results are in contradiction with practical experience, in which operations such as reaching agreement and coordinating actions are routinely performed in many actual distributed systems. It certainly seems as if these actions are performed in such systems without the designers having to worry about common knowledge (and despite the fact that we have proved that common knowledge is unattainable!). Secondly, these results seem to contradict our intuitive feeling that common knowledge *is* attained in many actual situations; for example, by the children in the muddy children puzzle.

Where is the catch? How can we explain this apparent discrepancy between our formal treatment and practical experience? What is the right way to interpret our negative results from the previous section? Is there indeed a paradox here? Or perhaps we are using a wrong or useless definition of common knowledge?

We believe that we do have a useful and meaningful definition of common knowledge. However, a closer inspection of the situation is needed in order to understand the subtle

processor p_i has the same history at (r, t') and at $(r', t' + \delta')$. Since (r, t) and p_i were chosen arbitrarily, it thus follows that the system has temporal imprecision. ▮

References

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

[Bar81] J. Barwise. Scenes and other situations. *Journal of Philosophy*, 78(7):369–397, 1981.

[CASD85] F. Cristian, H. Aghili, H. R. Strong, and D. Dolev. Atomic broadcast: from simple diffusion to Byzantine agreement. In *Proc. 15th International Conf. on Fault-Tolerant Computing Systems*, 1985.

[CL85] K. M. Chandy and L. Lamport. Distributed snapshots: determining global states of distributed systems. *ACM Trans. on Computer Systems*, 3(1):63–75, 1985.

[CM81] H. H. Clark and C. R. Marshall. Definite reference and mutual knowledge. In A. K. Joshi, B. L. Webber, and I. A. Sag, editors, *Elements of discourse understanding*. Cambridge University Press, Cambridge, U.K., 1981.

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

[DHS86] D. Dolev, J. Y. Halpern, and H. R. Strong. On the possibility and impossibility of achieving clock synchronization. *Journal of Computer and System Sciences*, 32(2):230–250, 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.

[DRS90] D. Dolev, R. Reischuk, and H. R. Strong. Early stopping in Byzantine agreement. *Journal of the ACM*, 34(7):720–741, 1990.

[EC82] E. A. Emerson and E. M. Clarke. Using branching time temporal logic to synthesize synchronization skeletons. *Science of Computer Programming*, 2:241–266, 1982.

[FH88] R. Fagin and J. Y. Halpern. Belief, awareness, and limited reasoning. *Artificial Intelligence*, 34:39–76, 1988.

[FH94] R. Fagin and J. Y. Halpern. Reasoning about knowledge and probability. *Journal of the ACM*, 41(2):340–367, 1994.

[FHV92] R. Fagin, J. Y. Halpern, and M. Y. Vardi. What can machines know? On the properties of knowledge in distributed systems. *Journal of the ACM*, 39(2):328–376, 1992.

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

[FLP85] M. J. Fischer, N. A. Lynch, and M. S. Paterson. Impossibility of distributed consensus with one faulty processor. *Journal of the ACM*, 32(2):374–382, 1985.

[Gal79] R. G. Gallager. Seminar on computer communication networks, Office of Industrial Liason, MIT. 1979.

[Gra78] J. Gray. Notes on database operating systems. In R. Bayer, R. M. Graham, and G. Seegmuller, editors, *Operating Systems: An Advanced Course*, Lecture Notes in Computer Science, Vol. 66. Springer-Verlag, Berlin/New York, 1978. Also appears as IBM Research Report RJ 2188, 1978.

[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. F. Traub, B. J. Grosz, B. W. Lampson, and N. J. Nilsson, editors, *Annual Review of Computer Science, Vol. 2*, pages 37–68. Annual Reviews Inc., Palo Alto, Calif., 1987.

[HF85] J. Y. Halpern and R. Fagin. A formal model of knowledge, action, and communication in distributed systems: preliminary report. In *Proc. 4th ACM Symp. on Principles of Distributed Computing*, pages 224–236, 1985.

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

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

[HM92] J. Y. Halpern and Y. Moses. A guide to completeness and complexity for modal logics of knowledge and belief. *Artificial Intelligence*, 54:319–379, 1992.

[HMM85] J. Y. Halpern, N. Megiddo, and A. Munshi. Optimal precision in the presence of uncertainty. *Journal of Complexity*, 1:170–196, 1985.

[HZ92] J. Y. Halpern and L. D. Zuck. A little knowledge goes a long way: knowledge-based derivations and correctness proofs for a family of protocols. *Journal of the ACM*, 39(3):449–478, 1992.

[Koz83] D. Kozen. Results on the propositional μ-calculus. *Theoretical Computer Science*, 27(1):333–354, 1983.

[KT86] S. Katz and G. Taubenfeld. What processes know: definitions and proof methods. In *Proc. 5th ACM Symp. on Principles of Distributed Computing*, pages 249–262, 1986.

[Lev84] H. J. Levesque. A logic of implicit and explicit belief. In *Proc. National Conference on Artificial Intelligence (AAAI '84)*, pages 198–202, 1984.

[LR86] R. E. Ladner and J. H. Reif. The logic of distributed protocols (preliminary report). In J. Y. Halpern, editor, *Theoretical Aspects of Reasoning about Knowledge: Proc. 1986 Conference*, pages 207–222. Morgan Kaufmann, San Francisco, Calif., 1986.

[MDH86] Y. Moses, D. Dolev, and J. Y. Halpern. Cheating husbands and other stories: a case study of knowledge, action, and communication. *Distributed Computing*, 1(3):167–176, 1986.

[ML90] M. S. Mazer and F. H. Lochovsky. Analyzing distributed commitment by reasoning about knowledge. Technical Report CRL 90/10, DEC-CRL, 1990.

[Moo85] R. C. Moore. A formal theory of knowledge and action. In J. Hobbs and R. C. Moore, editors, *Formal Theories of the Commonsense World*, pages 319–358. Ablex Publishing Corp., Norwood, N.J., 1985.

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

[MSHI79] J. McCarthy, M. Sato, T. Hayashi, and S. Igarishi. On the model theory of knowledge. Technical Report STAN-CS-78-657, Stanford University, 1979.

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

[MW84] Z. Manna and P. L. Wolper. Synthesis of communicating processes from temporal logic specifications. *ACM Trans. on Programming Languages and Systems*, 6(1):68–93, 1984.

[Nei88] G. Neiger. Knowledge consistency: a useful suspension of disbelief. In M. Y. Vardi, editor, *Proc. Second Conference on Theoretical Aspects of Reasoning about Knowledge*, pages 295–308. Morgan Kaufmann, San Francisco, Calif., 1988.

[NT93] G. Neiger and S. Toueg. Simulating real-time clocks and common knowledge in distributed systems. *Journal of the ACM*, 40(2):334–367, 1993.

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

[PT92] P. Panangaden and S. Taylor. Concurrent common knowledge: defining agreement for asynchronous systems. *Distributed Computing*, 6(2):73–93, 1992.

[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: Proc. 1986 Conference*, pages 83–97. Morgan Kaufmann, San Francisco, Calif., 1986.

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

[Tar55] A. Tarski. A lattice-theoretic fixpoint theorem and its applications. *Pacific Journal of Mathematics*, 5:285–309, 1955.

[YC79] Y. Yemini and D. Cohen. Some issues in distributed processes communication. In *Proc. of the 1st International Conf. on Distributed Computing Systems*, pages 199–203, 1979.