Reasoning About Knowledge: A Survey*

Joseph Y. Halpern

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

Abstract: In this survey, I attempt to identify and describe some of the common threads that tie together work in reasoning about knowledge in such diverse fields as philosophy, economics, linguistics, artificial intelligence, and theoretical computer science, with particular emphasis on work of the past five years, particularly in computer science.

*This articule is essentially the same as one that appears in *Handbook of of Logic in Artificial Intelligence and Logic Programming*, Vol. 4, D. Gabbay, C. J. Hogger, and J. A. Robinson, eds., Oxford University Press, 1995, pp. 1–34. It is a revised and updated version of a paper entitled "Reasoning about knowledge: a survey circa 1991", which appears in the *Encyclopedia of Computer Science and Technology*, Vol. 27, Supplement 12 (ed. A. Kent and J. G. Williams), Marcel Dekker, 1993, pp. 275–296. That article, in turn is a revision of an article entitled "Reasoning About Knowledge: An Overview" that appears in *Theoretical Aspects of Reasoning About Knowledge: Proceedings of the 1986 Conference*, Morgan Kaufmann, 1986 (J. Y. Halpern, ed.). Portions of this article are taken from the book *Reasoning About Knowledge* by R. Fagin, J. Y. Halpern, Y. Moses, and M. Y. Vardi, MIT Press, 1994.

1 Introduction

Although *epistemology*, the study of knowledge, has a long and honorable tradition in philosophy, starting with the Greeks, the idea of a formal logical analysis of reasoning about knowledge is somewhat more recent, going back to at least von Wright [Wright 1951]. The first book-length treatment of epistemic logic is Hintikka's seminal work, *Knowledge and Belief* [Hintikka 1962]. The 1960's saw a flourishing of interest in this area in the philosophy community. Axioms for knowledge were suggested, attacked, and defended. Models for the various axiomatizations were proposed, mainly in terms of possible-worlds semantics, and then again attacked and defended (see, for example, [Gettier 1963; Lenzen 1978; Barwise and Perry 1983]).

More recently, reasoning about knowledge has found applications in such diverse fields as economics, linguistics, artificial intelligence, and computer science. While researchers in these areas have tended to look to philosophy for their initial inspiration, it has also been the case that their more pragmatic concerns, which often centered around more computational issues such as the difficulty of computing knowledge, have not been treated in the philosophical literature. The commonality of concerns of researchers in all these areas has been quite remarkable, as has been attested by the recent series of interdisciplinary conferences on the subject [Halpern 1986b; Vardi 1988; Parikh 1990; Moses 1992; Fagin 1994].

In this survey, I attempt to identify and describe some of the common threads that tie together research in reasoning about knowledge in all the areas mentioned above. I also briefly discuss some of the more recent work, particularly in computer science, and suggest some lines for future research. This should by no means be viewed as a comprehensive survey. The topics covered clearly reflect my own biases.

2 The "classical" model

We begin by reviewing the "classical" model for knowledge and belief (now almost 40 years old!), the so-called *possible-worlds* model. The intuitive idea here is that besides the true state of affairs, there are a number of other possible states of affairs, or possible worlds. Some of these possible worlds may be indistinguishable to an agent from the true world. An agent is then said to *know* a fact φ if φ is true in all the worlds he thinks possible. For example, an agent may think that two states of the world are possible: in one it is sunny in London, while in the other it is raining in London. However, in both these states it is sunny in San Francisco. Thus, this agent knows that it is sunny in San Francisco, but does not know whether it is sunny in London.

The philosophical literature has tended to concentrate on the one-agent case, in order to emphasize the properties of knowledge. However, many applications of interest involve multiple agents. Then it becomes important to consider not only what an agent knows about "nature", but also what he knows about what the other agents know and don't know. It should be clear that this kind of reasoning is crucial in bargaining and economic decision making. As we shall see, it is also relevant in analyzing protocols in distributed computing systems (in this context, the "agents" are the processes in the system). Such reasoning can get very complicated. Most people quickly lose the thread of such nested sentences as "Dean doesn't know whether Nixon knows that Dean knows that Nixon knows that McCord burgled O'Brien's office at Watergate". (Clark and Marshall [1981] discuss the difficulties people have dealing with such statements.) But this is precisely the type of reasoning that goes on in a number of applications involving many agents.

To formalize this type of reasoning, we first need a language. The language I'll consider here is a propositional modal logic for n agents; this is a slight generalization of the logic described in Fitting's chapter in Volume 1 of this *Handbook*. Starting with a set Φ of primitive propositions (usually denoted by the letters p, q and r), complicated formulas are formed by closing off under negation, conjunction, and the modal operators K_1, ..., K_n. Thus, if φ and ψ are formulas, then so are $\neg\varphi$, $\varphi \wedge \psi$, and $K_i\varphi$, $i = 1, ..., n$. As usual, we take $\varphi \vee \psi$ to be an abbreviation for $\neg(\neg\varphi \wedge \neg\psi)$ and $\varphi \Rightarrow \psi$ to be an abbreviation for $\neg\varphi \vee \psi$.

The formula $K_i\varphi$ is read "agent i knows φ". The K_i's are called modal operators; hence the name modal logic. We could also consider a first-order modal logic that allows quantification along the lines discussed in Fitting's chapter, but the propositional case is somewhat simpler and has all the ingredients we need for our discussion.

We can express quite complicated statements in a straightforward way using this language. For example, the formula

$$K_1K_2p \wedge \neg K_2K_1K_2p$$

says that agent 1 knows that agent 2 knows p, but agent 2 doesn't know that agent 1 knows that agent 2 knows p. We view possibility as the dual of knowledge. Thus, agent 1 considers φ possible exactly if he doesn't know $\neg\varphi$. This situation can be described by the formula $\neg K_1\neg\varphi$. A statement like "Dean doesn't know whether φ" says that Dean considers both φ and $\neg\varphi$ possible. With these observations, we can deal with the sentence above, "Dean doesn't know whether Nixon knows that Dean knows that Nixon knows that McCord burgled O'Brien's office at Watergate." If we take Dean to be agent 1, Nixon to be agent 2, and p to be the statement "McCord burgled O'Brien's office at Watergate", then this sentence can be expressed in the logic as

$$\neg K_1\neg(K_2K_1K_2p) \wedge \neg K_1\neg(\neg K_2K_1K_2p).$$

When reasoning about the knowledge of a group, it becomes useful to reason not just about an individual agent's state of knowledge, but also about the knowledge of the group. For example, we might want to make statements such as "everyone in group G knows φ". It turns out to be useful to be able to make even more complicated statements such as "everyone in G knows that everyone in G knows φ", and "φ is common knowledge among the agents in G", where *common knowledge* is, informally, the infinite conjunction of the statements "everyone knows, and everyone knows that everyone knows, and everyone knows that everyone knows that everyone knows, ...".

Common knowledge was first studied by Lewis [1969] in the context of conventions. He points out that in order for something to be a convention, it must be common knowledge among the members of the group.

Common knowledge also arises in discourse understanding. If Ann asks Bob "Have you ever seen the movie playing at the Roxy tonight?", then in order for this question to be interpreted appropriately, not only must Ann and Bob know what movie is playing tonight, but Ann must know that Bob knows, Bob must know that Ann knows that Bob knows, etc. (This is discussed by Clark and Marshall [1981]; Perrault and Cohen [1981] offer a slightly dissenting view.)

Interest in common knowledge in the economics community was inspired by Aumann's seminal result [1976]. Aumann showed that if two people have the same prior probability for an event and their posterior probability for the event (that is, the probability they place on the event after getting some possibly different pieces of information) are common knowledge, then these posterior probabilities must be equal. This result says that people with the same prior probabilities *cannot agree to disagree*. Since then, common knowledge has received a great deal of attention in the economics literature; the issues examined include the number of rounds of communication information required before the posteriors for an event become common knowledge [Geanaokoplos and Polemarchakis 1982; Parikh and Krasucki 1990] and whether it is reasonable for rationality to be common knowledge (see [Brandenburger 1989] for a survey).

In order to express these notions, we augment the language with modal operators E_G ("everyone in the group G knows") and C_G ("it is common knowledge among the agents in G"), for every nonempty subset G of $\{1, \ldots, n\}$. This, we can make statements such as $E_G p \wedge \neg C_G p$: everyone in G knows p, but p is not common knowledge.

As discussed in Fitting's chapter, we can give semantics to this logic using the idea of *possible worlds* and *Kripke structures* [Kripke 1963]. Formally, a Kripke structure M is a tuple $(S, \pi, \mathcal{K}_1, ..., \mathcal{K}_n)$, where S is a set of *states* or *possible worlds*, π is an *interpretation* which associates with each state in S a truth assignment to the primitive propositions (i.e., $\pi(s)(p) \in \{\mathbf{true}, \mathbf{false}\}$ for each state $s \in S$ and each primitive proposition p), and $\mathcal{K}_i$ is an *equivalence relation* on S (recall that an equivalence relation is a binary relation which is reflexive, symmetric, and transitive). $\mathcal{K}_i$ is agent i's *possibility relation*. Intuitively, $(s, t) \in \mathcal{K}_i$ if agent i cannot distinguish state s from state t (so that if s is the actual state of the world, agent i would consider t a possible state of the world). We take $\mathcal{K}_i$ to be an equivalence relation, since it corresponds to the situation where, in state s, agent i considers t possible if it has the same information in both s and t. This type of situation arises frequently in distributed systems and economics applications. However, it is also possible to consider possibility relations with other properties (for example, reflexive and transitive, but not symmetric); most of the discussion goes through with very few changes if we change the nature of the possibility relation.

We now define a relation $\models$, where $(M, s) \models \varphi$ is read "φ is true, or *satisfied*, in state s of structure M".

$(M, s) \models p$ for a primitive proposition p if $\pi(s)(p) = \mathbf{true}$

$(M, s) \models \neg\varphi$ if $(M, s) \not\models \varphi$

$(M, s) \models \varphi \wedge \psi$ if $(M, s) \models \varphi$ and $(M, s) \models \psi$

$(M, s) \models K_i\varphi$ if $(M, t) \models \varphi$ for all t such that $(s, t) \in \mathcal{K}_i$

$(M, s) \models E_G\varphi$ if $(M, s) \models K_i\varphi$ for all $i \in G$

$(M, s) \models C_G\varphi$ if $(M, s) \models E_G^k\varphi$ for $k = 1, 2, \ldots$, where $E_G^1\varphi =_{\mathrm{def}} E_G\varphi$ and $E_G^{k+1}\varphi =_{\mathrm{def}} E_G E_G^k\varphi$.

The first clause shows how we use the π to define the semantics of the primitive propositions. The next two clauses, which define the semantics of $\neg$ and $\wedge$, are the standard clauses from propositional logic. The fourth clause is designed to capture the intuition that agent i knows φ exactly if φ is true in all the worlds that i thinks are possible. The fifth clause defines the semantics of $E_G\varphi$ in the most obvious way: $E_G\varphi$ holds if each agent in G knows φ, i.e., if $K_i\varphi$ holds for all $i \in G$. Finally, the last clause captures the intuitive definition of common knowledge discussed above.

These ideas are perhaps best illustrated by an example. One of the advantages of a Kripke structure is that it can be viewed as a labeled graph, that is, a set of labeled nodes connected by directed, labeled edges. The nodes are the states of S; each node is labeled by the primitive propositions true and false there, and there is an edge from s to t labeled i exactly if $(s, t) \in \mathcal{K}_i$. For example, suppose $\Phi = \{p\}$ and $n = 2$, so that our language only has one primitive proposition p and there are only two agents. Further suppose that $M = (S, \pi, \mathcal{K}_1, \mathcal{K}_2)$, where $S = \{s, t, u\}$, p is true at states s and u, but false at t (so that $\pi(s)(p) = \pi(u)(p) = \mathbf{true}$ and $\pi(t)(p) = \mathbf{false}$), agent 1 cannot tell s and t apart (so that $\mathcal{K}_1 = \{(s, s), (s, t), (t, s), (t, t), (u, u)\}$), and agent 2 cannot tell s and u apart (so that $\mathcal{K}_2 = \{(s, s), (s, u), (t, t), (u, s), (u, u)\}$). This situation can be captured by the graph in Figure 1.

If we view p as standing for "it is sunny in San Francisco", then in state s it is sunny in San Francisco but agent 1 doesn't know it (since he considers both s and t possible). On the other hand, agent 2 does know that it is sunny in state s, since in both worlds that agent 2 considers possible at s (namely, s and u), the formula p is true. Agent 2 also knows the true situation at state t, namely, that it is not sunny. It follows that in state s agent 1 knows that agent 2 knows whether or not it is sunny in San Francisco (since in both worlds agent 1 considers possible in state s, agent 2 knows what the weather in San Francisco is). Thus, although agent 1 does not know the true situation at s, he does know that agent 2 knows the true situation. By way of contrast, although in state s agent 2 knows that it is sunny in San Francisco, he doesn't know that agent 1 doesn't know

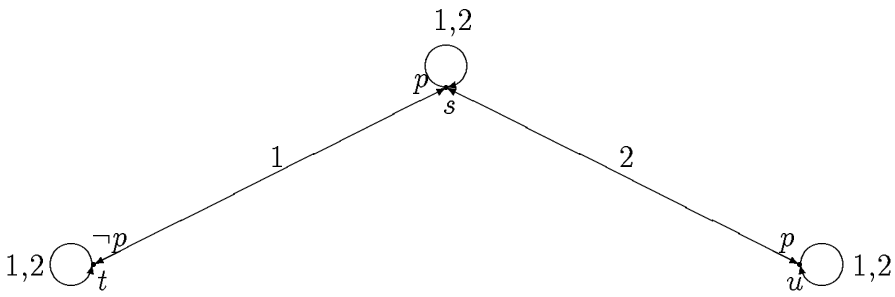

Figure 1: A simple Kripke structure

this fact. (In one world that agent 2 considers possible, namely u, agent 1 does know that it is sunny, while in another world agent 2 considers possible, s, agent 1 does not know this fact.) All of this relatively complicated English discussion can be summarized in one mathematical statement:

$$(M, s) \models p \land \neg K_1 p \land K_2 p \land K_1(K_2 p \lor K_2 \neg p) \land \neg K_2 \neg K_1 p.$$

What about common knowledge? It is not hard to check that the formula $\psi = K_2 p \lor K_2 \neg p$ is true at all three states s, t, and u in M. Taking $G = \{1, 2\}$, an easy induction on k now shows that in fact $E_G^k \psi$ is true at all three states, for all k. Thus, $(M, s) \models C_G \psi$.

Note that in both s and u, the primitive proposition p (the only primitive proposition in our language) gets the same truth value. One might think, therefore, that s and u are the same, and that perhaps one of them can be eliminated. This is not true! A state is not completely characterized by the truth values that the primitive propositions get there. The possibility relation is also crucial. For example, in world s, agent 1 considers t possible, while in u he doesn't. As a consequence, agent 1 doesn't know p in s, while in u he does.

How reasonable is this notion of knowledge? What are its properties? One way of investigating this issue is to try to find a complete characterization of the *valid* formulas, that is, those formulas that are true in every state in every structure.

If we ignore the operators E_G and C_G for the moment, the valid formulas in the language with only K_i can be completely characterized by the following *sound* and *complete* axiom system, due to Hintikka [Hintikka 1962]; i.e., all the axioms are valid and every valid formula can be proved from these axioms.

A1. All instances of propositional tautologies.

A2. $K_i \varphi \land K_i(\varphi \Rightarrow \psi) \Rightarrow K_i \psi$

A3. $K_i\varphi \Rightarrow \varphi$

A4. $K_i\varphi \Rightarrow K_iK_i\varphi$

A5. $\neg K_i\varphi \Rightarrow K_i\neg K_i\varphi$

R1. From φ and $\varphi \Rightarrow \psi$ infer ψ (modus ponens)

R2. From φ infer $K_i\varphi$

A1 and R1, of course, are holdovers from propositional logic. A2 says that an agent's knowledge is closed under implication. A3 says that an agent knows only things that are true. This is the axiom that is usually taken to distinguish *knowledge* from *belief*. You cannot know a fact that is false, although you may believe it. A4 and A5 are axioms of introspection. Intuitively, they say that an agent is introspective: he can look at his knowledge base and will know what he knows and doesn't know. There are numerous papers in the philosophical literature discussing the appropriateness of these axioms (see [Lenzen 1978] for an overview). Philosophers have tended to reject both of the introspection axioms for various reasons.

The validity of A3, A4, and A5 is due to the fact that we have taken the $\mathcal{K}_i$'s to be equivalence relations. In a precise sense, A3 follows from the fact that $\mathcal{K}_i$ is reflexive, A4 from the fact that it is transitive, and A5 from the fact that it is symmetric and transitive. By modifying the properties of the $\mathcal{K}_i$ relations, we can get notions of knowledge that satisfy different axioms. For example, by taking $\mathcal{K}_i$ to be reflexive and transitive, but not necessarily symmetric, we retain A3 and A4, but lose A5; similar modifications give us a notion that corresponds to belief, and does not satisfy A3. (See [Halpern and Moses 1992] for a survey of these issues, as well as a review of the standard techniques of modal logic which give completeness proofs in all these cases.)

However, the possible-worlds approach seems to commit us to A2 and R2. This suggests a view of our agents as "ideal knowers", ones that know all valid formulas as well as all logical consequences of their knowledge. This certainly doesn't seem to be a realistic model for human agents (although it might perhaps be acceptable as a first approximation). Nor does it seem to even be an adequate model for a knowledge base which is bounded in terms of the computation time and space in memory that it can use. We'll discuss some approaches to this problem of *logical omniscience* in Section 4 below.

Once we include the operators E_G and C_G in the language, we get further properties. These are completely characterized by the following additional axioms:

C1. $E_G\varphi \Leftrightarrow \bigwedge_{i\in G} K_i\varphi$

C2. $C_G\varphi \Leftrightarrow E_G(\varphi \wedge C_G\varphi)$ (fixed point axiom)

RC1. From $\varphi \Rightarrow E_G(\varphi \wedge \psi)$ infer $\varphi \Rightarrow C_G\psi$ (induction rule)

The fixed point axiom says that common knowledge of φ holds exactly when the group G is in a particular situation where everyone in G knows that φ holds and that common knowledge of φ holds. It turns out that this is the key property of common knowledge that makes it a prerequisite for agreement and coordination. The induction rule gives us a technique to verify that common knowledge holds in a certain situation. The reason for its name is that once we know that $\varphi \Rightarrow E_G(\varphi \wedge \psi)$ is valid, then we can show by induction on k that $\varphi \Rightarrow E_G^k(\varphi \wedge \psi)$ is valid for all k, from which we can conclude that $\varphi \Rightarrow C_G\psi$ is valid.

How hard is it to tell if a given formula defines a valid property of knowledge? We can give an answer in terms of complexity theory. (See [Hopcroft and Ullman 1979] for an introduction to complexity-theoretic notions mentioned below such as co-NP-completeness.) It can be shown that if a formula φ is valid iff it is true at every state in every structure with at most 2^n states, where n is the length of φ viewed as a string of symbols. From this result, it follows that validity is decidable: there is an algorithm that, given a formula φ, can tell whether or not it is valid. However, deciding validity is not easy. If we consider systems with just one agent, then it is co-NP-complete, just as it is for propositional logic [Ladner 1977]. But once we consider systems with two or more agents, any algorithm that decides validity requires space polynomial in the size of the input formula, even if we do not include common knowledge in the language. Once we include common knowledge, the complexity goes up to exponential time [Halpern and Moses 1992]. We'll return to the implication of these complexity results in Section 4.

3 A concrete interpretation: multi-agent systems

We want to use knowledge as a tool for analyzing multi-agent systems. For our purposes, we can view any collection of interacting agents as a multi-agent system. This includes the players in a poker game, processes in a computer network, or robots on an assembly line.

To model such a system formally, we assume it consists of n agents, each of which is in some *local state* at a given point in time. We assume that an agent's local state encapsulates all the information to which the agent has access. In a distributed system, the local state of a process might include some initial readings, the list of messages it has sent and received, and perhaps the reading of a clock. In a poker game, a player's local state might consist of the cards he currently holds, the bets made by other players, any other cards he has seen, and any information he may have about the strategies of the other players (for example, Bob may know that Alice likes to bluff, while Charlie tends to bet conservatively). We make no assumptions here about the precise nature of the local state.

We can then view the whole system as being in some *global state*, which is a tuple consisting of each process' local state, together with the state of the *environment*, where the environment consists of everything that is relevant to the system that is not contained in the state of the processes. Thus, a global state has the form $(s_e, s_1, \ldots, s_n)$, where s_e is the state of the environment and s_i is agent i's state, for $i = 1, \ldots, n$. The actual form of the agents' local states and the environment's state depends on the application being modeled. If we are studying a message-passing system consisting of communicating agents, the environment's state may include the status of the communication line (whether it is up or down, or whether there are any messages in transit on the line), while an agent's local state may include the sequences of messages she has sent and received. If we consider a system of sensors observing some terrain, a sensor's local state may just consist of its last (or last few) observations, while the environment's state may include features of the terrain not contained in the state of any of the sensors.

A system is not a static entity. To capture its dynamic aspects, we define a *run* to be a function from time to global states. Intuitively, a run is a complete description of what happens over time in one possible execution of the system. A *point* is a pair (r, m) consisting of a run r and a time m. For simplicity, we take time to range over the natural numbers in the remainder of this discussion. (In particular, this means that time steps are discrete and that time is infinite.) At a point (r, m), the system is in some global state $r(m)$. If $r(m) = (s_e, s_1, \ldots, s_n)$, then we take $r_i(m)$ to be s_i, agent i's local state at the point (r, m).

We formally define a *system* to consist of a set of runs. Notice how this definition abstracts our intuitive view of a system as a collection of interacting agents. Instead of trying to model the system directly, our definition models the possible *behaviors* of the system. For example, in a poker game, the runs could describe all the possible deals and betting sequences.

As we shall see, a system can be viewed as a Kripke structure except that we have no function π telling us how to assign truth values to the primitive propositions. (In the terminology of Fitting's chapter, a system can be viewed as a *frame*.) To view a system as a Kripke structure, we assume that we have a set Φ of primitive propositions, which we can think of as describing basic facts about the system. In the context of distributed systems, these might be such facts as "the value of the variable x is 0", "process 1's initial input was 17", "process 3 sends the message $\mathbf{m}$ in round 5 of this run", or "the system is deadlocked". An *interpreted system* $\mathcal{I}$ consists of a pair $(\mathcal{R}, \pi)$, where $\mathcal{R}$ is a system and π is an interpretation for the propositions in Φ which assigns truth values to the primitive propositions at the global states. Thus, for every $p \in \Phi$ and global state s that arises in $\mathcal{R}$, we have $\pi(s)(p) \in \{\mathbf{true}, \mathbf{false}\}$. Of course, π induces also an interpretation over the points of $\mathcal{R}$; simply take $\pi(r, m)$ to be $\pi(r(m))$. We refer to the points of the system $\mathcal{R}$ as points of the interpreted system $\mathcal{I}$. That is, we say that the point (r, m) is in the interpreted system $\mathcal{I} = (\mathcal{R}, \pi)$ if $r \in \mathcal{R}$.

We can associate with an interpreted system $\mathcal{I} = (\mathcal{R}, \pi)$ a Kripke structure $M_{\mathcal{I}} = (S, \pi, \mathcal{K}_1, \ldots, \mathcal{K}_n)$ in a straightforward way: We take S to consist of the points in $\mathcal{I}$. We define $\mathcal{K}_i$ so that $((r, m), (r', m')) \in \mathcal{K}_i$ if $r_i(m) = r'_i(m')$. Clearly $\mathcal{K}_i$ is an equivalence relation on points. Intuitively, agent i considers a point (r', m') possible at a point (r, m) if i has the same local state at both points. Thus, the agents' knowledge is completely determined by their local states.

We can now define what it means for a formula φ to be true at a point (r, m) in an interpreted system $\mathcal{I}$, written $(\mathcal{I}, r, m) \models \varphi$, by applying our earlier definitions:

$$(\mathcal{I}, r, m) \models \varphi \text{ iff } (M_{\mathcal{I}}, (r, m)) \models \varphi.$$

We remark that we can also reason about time in interpreted systems. That is, we can enrich the logic so that it contains temporal modal operators such as $\Box$ and $\Diamond$ and give them analogous definitions to those given in van Benthem's chapter in this Volume. For example, $\Box\varphi$ is true at a point if φ is true at that point and at all later points:

$$(\mathcal{I}, r, m) \models \Box\varphi \text{ iff } (\mathcal{I}, r, m') \models \varphi \text{ for all } m' \geq m.$$

In general, temporal operators are used for reasoning about events that happen along a single run. By combining temporal and knowledge operators, we can make assertions about the evolution of knowledge in the system.

This particular way of capturing knowledge in distributed systems is taken from [Halpern and Fagin 1989]. Slight variants of it have been used in most of the papers that attempt to define formal models for knowledge in distributed systems, such as [Chandy and Misra 1986; Fischer and Immerman 1986; Halpern and Moses 1990; Parikh and Ramanujam 1985]. Interestingly, essentially the identical notion of knowledge was developed independently by Rosenschein and his coworkers (cf. [Rosenschein 1985; Rosenschein and Kaelbling 1986]) and used for describing and analyzing situated automata in AI applications.

Note that in this model, knowledge is an "external" notion. We don't imagine a process scratching its head wondering whether or not it knows a certain fact φ. Rather, a programmer reasoning about a particular protocol would say, from the outside, that the process knows φ because in all global states consistent with its current state (intuitively, all the global states that the process could be in, for all it knows) φ is true. This notion of knowledge is information based, and does *not* take into account, for example, the difficulty involved in computing knowledge. Nor could a process necessarily answer questions based on its knowledge, with respect to this definition of knowledge. So on what basis can we even view this as knowledge?

There are two reasonable answers to this question. The first is that it corresponds to one common usage of the word. When trying to prove properties such as lower bounds on the number of rounds required to complete a given protocol, the kinds of arguments that one often hears have the form "We can't stop after only three rounds, because process 1 might not know that process 2 knows that process 3 is faulty." Now this informal use of the word "know" is exactly captured by the definition above. Let φ say that process 2 knows that process 3 is faulty. Then process 1 doesn't know φ exactly if there is a global state of the system that process 1 cannot distinguish from the actual state where φ does not hold; i.e., where process 2 doesn't know that process 3 is faulty.

The second answer is that this notion gives us a useful formalization of our intuitions, one that gives us important insights into the design and verification of distributed protocols. A good illustration of this is the *coordinated attack problem*, from the distributed systems folklore [Gray 1978]. The following presentation is taken from [Halpern and Moses 1990]:

> Two divisions of an army are camped on two hilltops overlooking a common valley. In the valley awaits the enemy. It is clear that if both divisions attack the enemy simultaneously they will win the battle, whereas if only one division attacks it will be defeated. The generals do not initially have plans for launching an attack on the enemy, and the commanding general of the first division wishes to coordinate a simultaneous attack (at some time the next day). Neither general will decide to attack unless he is sure that the other will attack with him. The generals can only communicate by means of a messenger. Normally, it takes the messenger one hour to get from one encampment to the other. However, it is possible that he will get lost in the dark or, worse yet, be captured by the enemy. Fortunately, on this particular night, everything goes smoothly. How long will it take them to coordinate an attack?

Suppose the messenger sent by General A makes it to General B with a message saying "Let's attack at dawn". Will general B attack? Of course not, since General A does not know he got the message, and thus may not attack. So General B sends the messenger back with an acknowledgement. Suppose the messenger makes it. Will General A attack? No, because now General B does not know that General A got the message, so General B thinks General A may think that he (B) didn't get the original message, and thus not attack. So A sends the messenger back with an acknowledgement. But of course, this is not enough either.

In terms of knowledge, each time the messenger makes a transit, the *depth* of the generals' knowledge increases by one. Suppose we let the primitive proposition m stand for "A message saying 'Attack at dawn' was sent by General A." When General B gets the message, $K_B m$ holds. When A gets B's acknowledgment, $K_A K_B m$ holds. The next acknowledgment brings us to $K_B K_A K_B m$. Although more acknowledgments keep increasing the depth of knowledge, it is not hard to show that by following this protocol, the generals never attain common knowledge that the attack is to be held at dawn.

What happens if the generals use a different protocol? That does not help either. As long as there is a possibility that the messenger may get captured or lost, then common knowledge is not attained, even if the messenger in fact does deliver his messages. It would take us too far afield here to completely formalize these results (see [Halpern and Moses 1990] for details), but we can give a rough description. We say a *system $\mathcal{R}$ displays unbounded message delays* if, roughly speaking, whenever there is a run $r \in \mathcal{R}$ such that process i receives a message at time m in r, then for all $m' > m$, there is another run r'

that is identical to r up to time m except that process i receives no messages at time m, and no process receives a message between times m and m'.

Theorem 3.1: [Halpern and Moses 1990] *In any run of a system that displays unbounded message delays, it can never be common knowledge that a message has been delivered.*

This says that no matter how many messages arrive, we cannot attain common knowledge of message delivery. But what does this have to do with coordinated attack? The fact that the generals have no initial plans for attack means that in the absence of message delivery, they will not attack. Since it can never become common knowledge that a message has been delivered, and message delivery is a prerequisite for attack, it is not hard to show that it can never become common knowledge among the generals that they are attacking. More precisely, let *attack* be a primitive proposition that is true precisely at points where both generals attack.

Corollary 3.2: *In any run of a system that displays unbounded message delays, it can never be common knowledge among the generals that they are attacking; i.e., if G consists of the two generals, then $C_G(attack)$ never holds.*

We still do not seem to have dealt with our original problem. What is the connection between common knowledge of an attack and coordinated attack? As the following theorem shows, it is quite deep. Common knowledge is a prerequisite for coordination in any *system for coordinated attack*, that is, in any system which is the set of runs of a protocol for coordinated attack.

Theorem 3.3: [Halpern and Moses 1990] *In any system for coordinated attack, when the generals attack, it is common knowledge among the generals that they are attacking. Thus, if $\mathcal{I}$ is an interpreted system for coordinated attack, and G consists of the two generals, then at every point (r, m) of $\mathcal{I}$, we have*

$$(\mathcal{I}, r, m) \models attack \Rightarrow C_G(attack).$$

Putting together Corollary 3.2 and Theorem 3.3, we get

Corollary 3.4: *In any system for coordinated attack that displays unbounded message delays, the generals never attack.*

This result shows not only that coordinated attack is impossible (a fact that was well known [Yemini and Cohen 1979]), but *why* it is impossible. The problem is due to an unattainability of common knowledge in certain types of systems.

In fact, as results of Halpern and Moses [1990] show, common knowledge is unattainable in a much wider variety of circumstances. Roughly speaking, common knowledge is not attainable whenever there is any uncertainty whatsoever about message delivery time. Common knowledge can be attained in "idealized" systems where we assume, for example, that events can be guaranteed to take place simultaneously. However, in the more common less-than-ideal systems, common knowledge is not attainable. Given that we also showed that common knowledge is a prerequisite for agreement, we seem to have something of a paradox here. After all, we often do reach agreement (or seem to!). Do we in fact get common knowledge, despite the results that say we can not?

Two solutions to the paradox are suggested in [Fagin, Halpern, Moses, and Vardi 1995; Halpern and Moses 1990]. The first involves a number of variants of common knowledge that are attainable under reasonable assumptions, and may suffice in practice. For example, we can consider a temporal variant called *ϵ-common knowledge*, which essentially says that "within ϵ time units everyone knows that within ϵ time units everyone knows that ..." Just as common knowledge corresponds to simultaneous coordination, ϵ common knowledge corresponds to coordinating to within ϵ time units. Further discussion of variants of common knowledge can be found in [Dwork and Moses 1990; Fischer and Immerman 1986; Fagin and Halpern 1994; Halpern and Moses 1990; Halpern, Moses, and Waarts 1990; Moses and Tuttle 1988; Neiger and Toueg 1993; Panangaden and Taylor 1992].

This approach still does not explain the pervasive feeling that we do (occasionally) attain common knowledge. The second approach attempts to deal with this issue. It is based on the observation that whether or not we get common knowledge depends on the granularity at which we model time. For example, suppose Alice and Bob are having a conversation, and Alice sneezes. Is it common knowledge that Alice has sneezed? If we model the situation in such a way that Alice and Bob perceive the sneeze *simultaneously*, then indeed there is common knowledge of the sneeze. If we take a more fine-grained model of time, where we take into account how long it takes for the information about the sneeze to be processed, and only say that Bob perceives that Alice has sneezed when he has processed this information, then not only is it unlikely that Alice and Bob perceive the sneeze simultaneously, but it should be clear that Alice has some uncertainty as to when Bob will perceive the sneeze. We can identify "the time required to perceive the sneeze" with "the message delivery time" in our earlier discussion. The fact that there is some uncertainty in the time required to perceive the sneeze again means that common knowledge of the sneeze is unattainable (no matter how small the uncertainty is!).

When we try to model real-world events, we often use a coarse-grained model of time. For example, when modeling distributed systems, we often assume that events occur in *rounds*, where a round provides sufficient time for a message to be sent by one process and received by its intended recipient, as well as time for some local computation. Is it reasonable to use a coarse-grained model of time? It depends. More precisely, assume that we are trying to show that a situation satisfies some property, or *specification*, σ. We have (at least) two ways of modeling the situation; one results in a coarse-grained system (i.e., one using a coarse-grained notion of time), the other in a fine-grained system. It is reasonable to use the coarse-grained system if σ holding in the coarse-grained system also implies that it holds in the fine-grained system. That is, it is safe to use a coarse-grained system if it does not lead us astray, as far as the specifications of interest go. If in fact Alice and Bob perceive the sneeze within several milliseconds of each other, then using the coarse-grained system (i.e., acting as if the coarse-grained system is a correct model of the world) is safe, provided that coordination to within several milliseconds is acceptable. Typically it is. For some specifications, it may not be.

This clearly is a special case of a more general issue: When is a particular model an accurate model of reality? There are very few general results along these lines; it is a topic that deserves further investigation. See [Fagin, Halpern, Moses, and Vardi 1995; Neiger 1988] for some further discussion.

The analysis of the coordinated attack problem shows the power of a knowledge-based approach to understanding distributed protocols. Numerous other papers have carried out knowledge-based analyses of protocols (for example, [Chandy and Misra 1986; Dwork and Moses 1990; Hadzilacos 1987; Halpern, Moses, and Tuttle 1988; Halpern, Moses, and Waarts 1990; Halpern and Zuck 1992; Mazer and Lochovsky 1990; Mazer 1990; Moses and Roth 1989; Moses and Tuttle 1988; Neiger and Toueg 1993; Panangaden and Taylor 1992]; an overview of the earlier work can be found in [Halpern 1987]). These papers suggest that the knowledge-based approach can indeed give useful insights. In cases where simultaneous agreement is required, as in some variants of the well-studied *Byzantine agreement problem* [Dolev and Strong 1982; Pease, Shostak, and Lamport 1980], common knowledge again turns out to play a key role (see [Dwork and Moses 1990; Moses and Tuttle 1988]). In *eventual* Byzantine agreement, where simultaneity is not required, it turns out that a variant of common knowledge characterizes the level of knowledge that is required [Halpern, Moses, and Waarts 1990]. For other protocols, common knowledge (or one of its variants) is not required; depth two knowledge (A knows that B knows) or depth three knowledge (A knows that B knows that A knows) may suffice [Hadzilacos 1987; Halpern and Zuck 1992; Mazer 1990]. It would be of great interest to have a deeper understanding of the level of knowledge required for various classes of problems; this may help us gain a better understanding of protocol design.

4 The problem of logical omniscience

The model of knowledge described in Section 2 gives rise to a notion of knowledge that seems to require that agents possess a great deal of reasoning power, since they know all the consequences of their knowledge and, in particular, they know all tautologies. Thus, the agents can be described as *logically omniscient.* While this notion of knowledge has been shown to be useful in a number of applications, it is clearly not always appropriate, particularly when we want to represent the knowledge of a resource-bounded agent. What is an appropriate notion of knowledge in this case? That may depend in part on the context and the application. In this section we'll consider a number of approaches to dealing with what has been called the logical omniscience problem.

One approach that has frequently been suggested is the syntactic approach: what an agent knows is simply represented by a set of formulas [Eberle 1974; Moore and Hendrix 1979]. Of course, this set need not be constrained to be closed under logical consequence or to contain all instances of a given axiom scheme. While this approach does allow us to define a notion of knowledge that doesn't suffer from the logical omniscience problem, by using it, we miss out on many of the merits of a knowledge-based analysis. If knowledge is represented by an arbitrary set of formulas, we have no structure or principles to guide us in our analysis. A somewhat more sophisticated approach is taken by Konolige [1986], who considers starting with a set of base facts, and then closing off under a (possibly incomplete) set of deduction rules. But even here we lose the benefits of a good underlying semantics.

A semantic analogue to the syntactic approach can be obtained by using *Montague-Scott structures* [Montague 1960]. The idea here is that a formula corresponds to a set of possible worlds (intuitively, the set of worlds where it is true). Rather than representing what an agent knows by a set of formulas (syntactic objects), we represent what an agent knows by a set of sets of possible worlds. Since each set of possible worlds corresponds to a formula, the two approaches are similar in spirit. Formally, we take a Montague-Scott structure to be a tuple $M = (S, \pi, \mathcal{C}_1, \ldots, \mathcal{C}_n)$, where S is a set of possible worlds and π defines a truth assignment at each possible world, just as in the case of a Kripke structure, while $\mathcal{C}_i(s)$ is a set of subsets of S for each $s \in S$. We can now define $\models$ for all formulas. All clauses are the same as for Kripke structures, except in the case of formulas of the form $K_i\varphi$. In this case we have

$$(M, s) \models K_i\varphi \text{ iff } \{t \,|\, (M, t) \models \varphi\} \in \mathcal{C}_i(s).$$

Thus, agent i knows φ if the set of possible worlds where φ is true is one of the sets of worlds that he considers possible.

The Montague-Scott approach has a great deal of power; by putting appropriate conditions on the sets $\mathcal{C}_i$ we can capture many interesting properties of knowledge, without committing to others. For example, agent i's knowledge is closed under implication (that is, $(K_i\varphi \wedge K_i(\varphi \Rightarrow \psi)) \Rightarrow K_i\psi$ is valid) if $\mathcal{C}_i(s)$ is closed under supersets for each $s \in S$ (that is, $T \in \mathcal{C}_i(s)$ and $T \subseteq T'$ implies $T' \in \mathcal{C}_i(s)$). Similarly, agent i knows all tautologies if $S \in \mathcal{C}_i(s)$ for all $s \in S$. (See [Vardi 1989] for more details on the fine-tuning that is possible with the Montague-Scott approach.) Since we do not require that $\mathcal{C}_i(s)$ be closed under supersets nor that it contain S, the Montague-Scott approach does not suffer from the major problems of logical omniscience. However, because it is a *semantic* approach, it cannot avoid having the following property: if φ and ψ are equivalent, then so are $K_i\varphi$ and $K_i\psi$. An agent cannot distinguish logically equivalent formulas (even if they have different syntactic structure). Thus, we have the following inference rule, which is sound for Montague-Scott structures:

- From $\varphi \equiv \psi$ infer $K_i\varphi \equiv K_i\psi$.

Of course, whether this is a problem depends on the particular application one has in mind.

While the Montague-Scott and the syntactic approach have a great deal of expressive power, one gains very little intuition about knowledge from these approaches. In these approaches knowledge is a primitive construct (much like the primitive propositions in a Kripke structure). Arguably, these approaches give us ways of *representing* knowledge, rather than *modeling* knowledge. We now investigate a few approaches that retain the flavor of the possible-worlds approach, yet still attempt to mitigate the logical omniscience problem.

One approach is to base an epistemic logic on a nonstandard logic, rather than on classical logic. There are a number of well-known nonstandard logics, including *intuitionistic logic* [Heyting 1956], *relevance logic* [Anderson and Belnap 1975], and the four-valued logic of [Belnap 1977a; Belnap 1977b; Dunn 1986]. Typically, these logics attempt to reformulate the notion of implication, to avoid some of the problems perceived with the notion of material implication. For example, in standard logic, from a contradiction one can deduce anything; the formula $(p \wedge \neg p) \Rightarrow q$ is valid. However, consider a knowledge base into which users enter data from time to time. As Belnap [1977b] points out, it is almost certainly the case that in a large knowledge base, there will be some inconsistencies. One can imagine that at some point a user entered the fact that Bob's salary is $50,000, while at another point, perhaps a different user entered the fact that Bob's salary is $60,000.

In [Fagin, Halpern, and Vardi 1990], a logic of knowledge is defined that is based on a nonstandard propositional logic called *NPL*, which is somewhat akin to relevance logic, and where, among other things, a formula such as $(p \wedge \neg p) \Rightarrow q$ is no longer valid. The possible worlds are now models of NPL. Agents are still logically omniscient, but now they know only NPL tautologies, rather than classical tautologies. This has some advantages. In particular, it can be shown that questions of the form "Does $K_i\varphi$ logically imply $K_i\psi$?", where φ and ψ are propositional formulas in conjunctive normal form, can be decided in polynomial time (which is not the case for standard logics of knowledge). This is an important subclass of formulas. If we view φ as representing the contents of a knowledge base and ψ as representing a query to the database, then it essentially amounts to asking whether a knowledge base that knows φ also knows ψ. Thus, under this interpretation of knowledge, queries to a knowledge base of the form "Do you know φ?" can be decided quite efficiently (assuming φ is in conjunctive normal form).

Yet another approach has been called the *impossible-worlds approach.* The idea here is that the possible worlds, where all the customary rules of classical logic hold, are augmented by "impossible" worlds, where they do not [Cresswell 1973; Hintikka 1975; Rantala 1982; Rescher and Brandom 1979; Wansing 1990]. For example, in an impossible world, it may be the case that $p \wedge \neg p$ holds, while this cannot be the case in a possible world. It is still the case that an agent knows φ if φ is true in all the worlds that he considers possible, but now an agent may consider impossible worlds possible. Thus, an agent may not know all tautologies of classical logic, since in some of the worlds he considers possible (namely, the impossible worlds), these tautologies may not hold.

Although there are impossible worlds in a structure, when we consider what are the *valid* formulas in the impossible-worlds approach, we only consider the standard possible worlds. The intuition here is that although the agent may be confused and consider impossible worlds possible, we, the logicians looking at the situation from the outside, know better.

There are many variants of the impossible-worlds approach, depending on how one constructs the impossible worlds. One variant is considered by Levesque [1984b]. In Levesque's impossible worlds, a primitive proposition may be either true, false, both, or neither. This also makes Levesque's approach closely related to relevance logic and to the logic NPL discussed above. Indeed, it can be shown that Levesque's structures are essentially equivalent to NPL structures. The only significant difference between Levesque's approach and that of [Fagin, Halpern, and Vardi 1990] is that Levesque considers only the possible worlds—the ones that obey the laws of classical logic—when considering validity, whereas in [Fagin, Halpern, and Vardi 1990], all worlds are considered. Just as in the context of NPL, checking whether $K_i\varphi$ logically implies $K_i\psi$ for propositional formulas φ and ψ in conjunctive normal form can be decided in polynomial time. (Indeed, this result was first proved in [Levesque 1984b], and then adapted to NPL in [Fagin, Halpern, and Vardi 1990].)

Levesque [1984b] restricts attention to *depth one* formulas, where there are no nested occurrences of K's. He also restricts to the case of a single agent. Lakemeyer [Lakemeyer 1987] has extended Levesque's approach to more deeply nested formulas; his approach can also be extended to deal with multiple agents. Patel-Schneider [1985] and Lakemeyer [1986] have also considered extensions to the first-order case which attempt to preserve decidability for a reasonable fragment of the logic.

Yet another approach to dealing with logical omniscience is to have truth in all possible worlds be a necessary but not sufficient condition for knowledge. Fagin and Halpern [1988] take this approach. Their *logic of general awareness* is essentially a mixture of syntax and semantics. It starts with a standard Kripke structure, and adds to each state a set of formulas that the agent is "aware" of at that state. Now an agent (explicitly) knows a formula φ at state s exactly if φ is true in all worlds the agent considers possible at s *and* φ is one of the formulas the agent is aware of at s. Thus, an agent may not know a tautology, even if it is true at all the worlds that he considers possible, simply because he is not aware of it. Similarly, an agent who knows φ and $\varphi \Rightarrow \psi$ may not know ψ because he is not aware of ψ.

There are a number of different interpretations we can give the notion of awareness. For example, we could say that an agent is aware of a formula if he is aware of all the concepts involved in that formula. Perhaps the most interesting interpretation is a computational one, where an agent is aware of a formula if he can figure out whether the formula is true (perhaps using some specific algorithm) within a prespecified time bound. Under this interpretation, the awareness set at state s would consist of those formulas whose truth the agent can figure out given the information it has acquired at state s.

This interpretation has been investigated in the context of the model for multi-agent systems discussed in the previous section in work of Halpern, Moses, and Vardi [1994], which in turn is based on earlier work of Moses [1988]. The key idea is to add to the agent's local state the algorithm that he is using to compute his knowledge. Thus, the agent's local state at a point (r, m) has the form $(\mathbf{A}, \ell)$, where $\mathbf{A}$ is his algorithm and ℓ is the rest of his local state. We call ℓ the *local data.* In local state $\langle \mathbf{A}, \ell \rangle$, the agent computes whether he knows φ by applying the local algorithm $\mathbf{A}$ to input (φ, ℓ). The output is either "Yes", in which case φ is known to be true, "No", in which case φ is not known to be true, or "?", which intuitively says that the algorithm has insufficient resources to compute the answer. It is the last clause that allows us to deal with resource-bounded reasoners. We can now augment the logic by introducing new modal operators X_i, $i = 1, \ldots, n$, for *algorithmic knowledge*, defined as follows:

$$(\mathcal{I}, r, m) \models X_i\varphi \text{ iff } \mathbf{A}(\varphi, \ell) = \text{“Yes”}, \text{ where } r_i(m) = (\mathbf{A}, \ell).$$

Thus, agent i has algorithmic knowledge of φ at a given point if the agent's algorithm at that point outputs "Yes" when presented with φ and with the agent's local data. (Note that both the outputs "No" and ? result in lack of algorithmic knowledge.)

This definition makes clear that computing whether an agent knows φ has nothing to do in general with computing whether φ is valid. Rather, it is closely related to the *model-checking problem*, that is, the problem of checking whether φ is true at a particular point in a system [Halpern and Vardi 1991]. Because of this, the fact that checking validity is PSPACE-complete in multi-agent S5 [Halpern and Moses 1992] does not indicate that computing knowledge in any particular situation will necessarily be hard. See [Halpern and Vardi 1991] for further discussion of this point.

As defined, there is no necessary connection between $X_i\varphi$ and $K_i\varphi$. An algorithm could very well claim that agent i knows φ (i.e., output "Yes") whenever it chooses to, including at points where $K_i\varphi$ does not hold. Although algorithms that make mistakes are common, we are often interested in local algorithms that are correct. We say that a local algorithm is *sound* for agent i in the system $\mathcal{I}$ if for all points (r, m) of $\mathcal{I}$ and formulas φ, if $r_i(m) = (\mathbf{A}, \ell)$, then (a) $\mathbf{A}(\varphi, \ell) =$ "Yes" implies $(\mathcal{I}, r, m) \models K_i\varphi$, and (b) $\mathbf{A}(\varphi, \ell) =$ "No" implies $(\mathcal{I}, r, m) \models \neg K_i\varphi$. Thus, a local algorithm is sound if its answers are always correct. A local algorithm $\mathbf{A}$ is called *complete* for agent i in the system $\mathcal{I}$ if for all points (r, m) of $\mathcal{I}$ and all formulas φ, if $r_i(m) = (\mathbf{A}, \ell)$, then $\mathbf{A}(\varphi, \ell) \in \{$"Yes", "No"$\}$. Thus, a local algorithm is complete if it always gives a definite answer. Notice that at a point where agent i uses a sound and complete local algorithm, $X_i\varphi \Leftrightarrow K_i\varphi$ holds. If we restrict attention to sound algorithms, then algorithmic knowledge fits into the general awareness framework of Fagin and Halpern [1988]: the agent can be viewed as being aware of φ at a given point precisely if her local algorithm returns "Yes" on input φ at that point.

A number of earlier efforts to solve the logical omniscience problem can be embedded easily into the framework of algorithmic knowledge. The approach of Konolige mentioned above provides one example. Recall that in Konolige's approach, an agent knows precisely the formulas in the set that is obtained by starting with a base set and closing off under a (possibly incomplete) set of deduction rules. In the framework of algorithmic knowledge, the base set of formulas would be part of the agent's local data, while the formal system would characterize her local algorithm.

For another example, consider Levesque's impossible-worlds approach. Levesque is mainly interested in modeling a knowledge base KB that is told a number of facts. This can be modeled in the framework of multiagent systems by having a *Teller* and a KB as agents. The KB's local data at a given point is the sequence of facts it has been told. If we assume that these facts are all propositional and that they describe an unchanging world, then we can identify this sequence with the formula κ consisting of the conjunction of what it has been told. When asked a query φ in state κ, for κ and φ in CNF, suppose we assume that the KB's local algorithm is to test whether $K\kappa \Rightarrow K\varphi$ is valid under Levesque's semantics (or, equivalently, in the approach based on NPL used in [Fagin, Halpern, and Vardi 1990]). If it is, the algorithm outputs "Yes", otherwise it outputs "?". As we mentioned above, this can be done in polynomial time. By Levesque's results, this algorithm is sound, but not complete (even for formulas in CNF).

5 Knowledge, communication, and action

Implicit in much of the previous discussion has been the strong relationship between knowledge, communication, and action. Indeed, much of the motivation for studying knowledge by researchers in all areas has been that of understanding the knowledge required to perform certain actions, and how that knowledge can be acquired through communication. This is a vast area; we briefly review some recent trends here.

Early work of McCarthy and Hayes [1969] argued that a planning program needs to explicitly reason about its ability to perform an action. Moore [1985] took this one step further by emphasizing the crucial relationship between knowledge and action. Knowledge is necessary to perform actions, and new knowledge is gained as a result of performing actions. Moore went on to construct a logic with possible-worlds semantics that allows explicit reasoning about knowledge and action, and then considered the problem of automatically generating deductions within the logic. This work has been extended by Morgenstern [1986]; she views "know" as a syntactic predicate on formulas rather than a modal operator.

Another issue that has received a great deal of attention recently is the relationship between knowledge and communication. Levesque [1984a] considered this from the point of view of a knowledge base that could interact with its domain via $TELL$ and ASK operations. He showed, somewhat surprisingly, that the result of $TELL$ing a knowledge base an arbitrary sentence in a first-order logic of knowledge is always equivalent to the result of $TELL$ing it a purely first-order sentence (i.e. one without any occurrences of K). It is worth remarking here that it is crucial to Levesque's result that there is only one knowledge base, i.e. one agent, in the picture.

Characterizing the states of knowledge that result after communication is also surprisingly subtle. One might think, for example, that after telling someone a fact p he will know p (at least, if it is common knowledge that the teller is honest). But this is not true. For example, consider the sentence "p is true but you don't know it". When told to agent i, this would be represented as $p \wedge \neg K_i p$. Now this sentence might be perfectly true when it is said. But after i is told this fact, it is not the case that $K_i(p \wedge \neg K_i p)$ holds. In fact, this latter formula is provably inconsistent! It is the case, though, that i knows that $p \wedge \neg K_i p$ was true before, although it is no longer true now.

Even if we do not allow formulas that refer to knowledge, there are subtleties in characterizing the knowledge of an agent. Consider the following example from [Fagin, Halpern, and Vardi 1991]. Suppose that Alice has been told only one fact: the primitive proposition p. Intuitively, all she knows is p. Since we are assuming ideal agents, Alice also knows all the logical consequences of p. But is this all she knows? Suppose q is another primitive proposition. Surely Alice doesn't know q, i.e. $\neg K_A q$ holds. But we assume Alice can do perfect introspection, so that she knows about her lack of knowledge of q. Thus $K_A \neg K_A q$ holds. But this means that even if "all Alice knows is p", then she also knows $\neg K_A q$, which is surely not a logical consequence of p! The situation can get even more complicated if we let Bob into the picture. For then Alice knows that Bob doesn't know that Alice knows q. (How can he, since in fact she doesn't know q, and Bob does not know false facts.) And knowing that Bob can also do perfect introspection, Alice knows that Bob knows this fact; i.e., $K_A K_B \neg K_B K_A q$ holds! Thus, despite her limited knowledge, Alice knows a nontrivial fact about Bob's knowledge. (See [Fagin, Halpern, and Vardi 1991; Halpern 1993b; Halpern and Moses 1984; Lakemeyer 1993; Lakemeyer and Levesque 1988; Levesque 1990; Parikh 1991; Stark 1981] for further discussion of these points.) Part of the difficulty here is due to *negative introspection*, i.e., the fact that one has knowledge about one's own lack of knowledge. If we remove this feature from our model (i.e., discard axiom A5), then some of the subtleties disappear (cf. [Halpern 1993b; Vardi 1985]).

One approach that might go a long way to clarifying some of these problems is to use the semantic model of multi-agent systems discussed in Section 3. Rather than describing an agent's knowledge as a collection of formulas, we instead describe (the runs of) the protocol by which the agent acquires knowledge. As we mentioned earlier, Levesque's knowledge base can then be modeled as an agent in such a system, in which the Teller is another agent. As shown by Fagin, Halpern, Moses, and Vardi [1994, 1995], such an approach can be used to capture aspects of knowledge bases more elegantly and concisely than the traditional axiomatic approach, and can help clarify some of the subtleties discussed above.

6 Knowledge and probability

In many of the application areas for reasoning about knowledge, it is important to be able to reason about the probability of certain events as well as the knowledge of agents. This arises in distributed systems, since we want to analyze randomized or probabilistic programs. In game theory and economics, researchers typically want to assume that agents have priors on certain events and make their decisions accordingly. Indeed, although researchers in economics and game theory did not use a logical language with operators for probability, probability has explicitly appeared in their framework all along, going back to the papers of Aumann [1976] and Mertens and Zamir [1985].

It seems straightforward to add probability into the framework that we have developed. As far as syntax goes, we can add statements such as $Pr_i(\varphi) = 1/2$ (according to agent i, the probability that φ holds is 1/2), and then close off under knowledge operators, to allow formulas such as $K_i K_j(Pr_i(\varphi) = 1/2)$ (this syntax is taken from [Fagin and Halpern 1994]). In order to be able to decide if a formula such as $Pr_i(\varphi) = 1/2$ is true at a state s, the obvious approach would be to put a probability on the set of worlds that agent i considers possible at s (where the exact probability used would depend on agent i's prior, or some information contained in the problem statement).

The difficulty comes in deciding what probability space agent i should use. This seems like it should be straightforward. A structure already tells us which worlds agent i considers possible at state s. All that remains is to make this uncertainty a little more quantitative, by assigning a probability to each of the worlds that agent i considers possible in such a way that the probabilities add up to 1. To see that the situation is not quite so straightforward, consider the following example, taken from [Fagin and Halpern 1994]:

> Suppose we have two agents. Agent 2 has an input bit, either 0 or 1. He then tosses a fair coin, and performs an action a if the coin toss agrees with the input bit, i.e., if the coin toss lands heads and the input bit is 1, or if the coin lands tails and the input bit is 0. We assume that agent 1 never learns agent 2's input bit or the outcome of his coin toss. An easy argument shows that according to agent 2, who knows the input bit, the probability (before he tosses the coin) of performing action a is 1/2. There is also a reasonable argument to show that, even according to agent 1 (who does not know the input bit), the probability that the action will be performed is 1/2. Clearly, from agent 1's viewpoint, if agent 2's input bit is 0, then the probability that agent 2 performs action a is 1/2 (since the probability of the coin landing heads is 1/2); similarly, if agent 2's input bit is 1, then the probability of agent 2 performing action a is 1/2. Thus, no matter what agent 2's input bit, the probability according to agent 1 that agent 2 will perform action a is 1/2. It seems reasonable to conclude that agent 1 knows that the *a priori* probability of agent 2 performing action a is 1/2. Note that we do not need to assume a probability distribution on the input bit for this argument to hold. Indeed, it holds independent of the probability distribution, and even if there is no probability distribution on the input bit.

Now suppose we want to capture this argument in our formal system. From agent 1's point of view, there are four possibilities: $(0,h),(0,t),(1,h),(1,t)$ (the input bit was 0 and the coin landed heads, the input bit was 0 and the coin landed tails, etc.). We can view these as the possible worlds or states in a Kripke structure. Call them s_1, s_2, s_3, and s_4 respectively; let S be the set consisting of all four states. Assume that we have primitive propositions A, H, T, B_0, and B_1 in the language, denoting the events that action a is performed, the coin landed heads, the coin landed tails, agent 2's input bit is 0, and agent 2's input bit is 1. Thus H is true at states s_1 and s_3, A is true at states s_2 and s_3, and so on. Now suppose we try to put a probability space on S. It is clear that the event "heads", which corresponds to the set $\{s_1, s_3\}$, should get probability 1/2; similarly the set $\{s_2, s_4\}$ should get probability 1/2. On the other hand, there is no natural probability we can assign to the set $\{s_1, s_2\}$, since this set corresponds to the event "the input bit is 0", an event for which we do not have a probability.

In order to capture our informal argument, we can instead split up S into two separate probability spaces, say S_0 and S_1, where S_0 consists of the points s_1 and s_2, while S_1 consists of the points s_3 and s_4. Intuitively, S_i is the conditional space resulting from conditioning on the event "the input bit is i". We can view S_i as a probability space in the obvious way; for example, in S_0, we give each of the points s_1 and s_2 probability 1/2. In each of S_0 and S_1, the probability of the event A is 1/2. For example, in S_0, the event A holds at the point s_2, which has probability 1/2. The fact that A has probability 1/2 in each of S_0 and S_1 corresponds to our informal argument that, no matter what the input bit is (even if agent 1 does not know the input bit), the probability of A is 1/2. Once we split up S into two subspaces in this way, the statement $Pr_1(A) = 1/2$ holds at all four points in S, and thus $K_1(Pr_1(A) = 1/2)$ holds: agent 1 *knows* that the probability of A is 1/2.

While dividing up S into two subspaces in this way captures our informal argument, it leads to an obvious question: What makes this the right way to divide S into subspaces? Suppose instead we had divided S into four subspaces $T_1, \dots, T_4$, where T_i is the singleton $\{s_i\}$. When we view T_i as a probability space in the obvious way, the point s_i must have probability 1. With this choice of subspaces, $Pr_1(A) = 1$ is true at the points s_2 and s_3, and $Pr(A) = 0$ is true at the points s_1 and s_4. Thus, all we can conclude is $K_1(Pr_1(A) = 0 \vee Pr_1(A) = 1)$. The agent knows that the probability of A is either 0 or 1.

Notice that there is a reasonable interpretation that we can give to the choice of $T_1, \dots, T_4$. Before the coin is tossed, the agent can argue that the probability of A is 1/2. What about after the coin has been tossed? There is one school of thought that would argue that after the coin has been tossed, A has been decided one way or another. Its probability is either 0 or 1, although agent 1 does not know which it is. From this point of view, dividing S into S_0 and S_1 captures the situation before the coin toss, while dividing it into $T_1, \dots, T_4$ captures the situation after the coin toss. It is not a question of which is right or wrong; both choices are appropriate, but capture different situations.

This issue is studied in a more general setting by Halpern and Tuttle [1993]. The argument there is that different partitions of the set of possible worlds into subspaces correspond to playing against different adversaries, with different knowledge. For example, the partition $T_1, \ldots, T_4$ corresponds in a precise sense to playing against an adversary that knows the outcome of the coin toss, while the partition S_0, S_1 corresponds to playing an adversary that does not know the outcome. This point of view allows us to clarify some important philosophical issues regarding the distinction between probability and nondeterminism, as well as providing us with a means of analyzing randomized protocols.

7 Other work and further directions

I have discussed what I see as many of the most important trends in research on reasoning about knowledge but, as I mentioned in the introduction, this is by no means a comprehensive survey. Let me briefly mention a few other topics that were neglected above due to lack of space.

- Using epistemic logics to better understand aspects of nonmonotonicity: see, for example, [Lin and Shoham 1990; Moses and Shoham 1993; Shoham 1988]. For further details see Konolige's chapter in Volume 3 of this *Handbook*.

- Connections between epistemic logics and *zero-knowledge proofs* [Goldwasser, Micali, and Rackoff 1989]: In a zero-knowledge proof, a prover tries to convince a verifier of a certain fact (such that a particular number n is composite) without revealing any additional information (such as the factors of n). To make this precise, we need to invoke notions of computability and probability (since there is allowed to be a small probability of error). These notions can be formalized in epistemic logic by combining the resource-bounded approach of [Moses 1988; Halpern, Moses, and Vardi 1994] with the logic of probability and knowledge of [Fagin and Halpern 1994]; see [Halpern, Moses, and Tuttle 1988] for details.

- Reasoning about knowledge/belief change over time: since the framework for multi-agent systems described in Section 3 has time explicitly built in, it provides a useful tool for studying how knowledge evolves over time. There are a number of assumptions that one can make about how knowledge changes. This assumption can be easily captured in the framework, although it makes formal reasoning about knowledge and time far more complex [Halpern and Vardi 1988; Halpern and Vardi 1989]. As shown by Friedman and Halpern [1994a], this framework is also well-suited to the study of *belief change*, in the spirit of the discussion in Gärdenfors and Rott's chapter in this Volume. The first step in this approach is to add a *plausibility ordering* to the system. Then an agent is said to *believe* φ if he knows that φ is true in all the most plausible worlds (according to the plausibility ordering). Plausibility can be viewed as a qualititative analogue of probability, so many of the issues that arose in the discussion of knowledge and probability in Section 6 arise again here. Friedman and Halpern [1994b] show that this framework can capture the two best-studied notions of belief change—*belief revision* [Alchourrón, Gärdenfors, and Makinson 1985] and *belief update* [Katsuno and Mendelzon 1991]—in a straightforward way. All this can be viewed as further evidence that there are fruitful connections between the notions of knowledge, belief, probability, and plausibility, and that these can all be usefully studied in one framework.

- *Knowledge-based programming:* One of the great advances in computer science was the introduction of high-level programming languages. The goal is to allow a programmer to write a program by saying "what she wants", rather than painfully describing "how to compute what she wants". Since actions are often based on knowledge, we might want to allow a programming language to have explicit tests for knowledge, so that an agent's actions can depend on what he knows. The analyses of [Dwork and Moses 1990; Halpern and Zuck 1992; Moses and Tuttle 1988; Halpern, Moses, and Waarts 1990] suggest that such knowledge-based programs do indeed provide a high-level way to describe the relationship between knowledge and action. Halpern and Fagin [1989] provide a formal semantics for knowledge-based protocols, which is further refined in [Fagin, Halpern, Moses, and Vardi 1995]. We are still a long way from having a full-fledged knowledge-based programming language, where the details of how the knowledge is computed are invisible to the programmer, but the possibility is tantalizing. The *agent-oriented programming* approach suggested by Shoham [1993] can be viewed as a first step along these lines.

Research is currently proceeding in all these areas, as well as the ones mentioned earlier in this article. In earlier overview articles [Halpern 1986a; Halpern 1987; Halpern 1993a], I concluded with suggestions for areas where further research needed to be done. The bibliography of this survey is testimony to the progress that has been made since these overviews were written. Nevertheless, there is much more that could be done. In particular, it seems to me that there are three areas where further research could lead to major progress:

- Analyzing more protocols using tools of knowledge. It would be particularly interesting to see if thinking in terms of adversaries can give us further insight into randomized protocols. Having a larger body of examples will enable us to further test and develop our intuitions.

- Getting more realistic models of knowledge, that incorporate resource-bounded reasoning, probability, and the possibility of errors.

- Getting a deeper understanding of the interplay between various modes of reasoning under uncertainty. I mentioned above the fruitful connections between knowledge, belief, probability, and plausibility. There is undoubtedly much more work to be done in getting a better understanding of the interplay between these notions.

I am optimistic that the next five years will bring us a deeper understanding of all these issues.

Acknowledgments: The presentation of the ideas in this paper owes a great deal to discussions with Ron Fagin, Yoram Moses, and Moshe Vardi in the context of writing a book on reasoning about knowledge [Fagin, Halpern, Moses, and Vardi 1995].

References

Alchourrón, C. E., P. Gärdenfors, and D. Makinson (1985). On the logic of theory change: partial meet functions for contraction and revision. *Journal of Symbolic Logic 50*, 510–530.

Anderson, A. and N. D. Belnap (1975). *Entailment: The Logic of Relevance and Necessity.* Princeton, N.J.: Princeton University Press.

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

Barwise, J. and J. Perry (1983). *Situations and Attitudes.* Cambridge, Mass.: Bradford Books.

Belnap, N. D. (1977a). How a computer should think. In *Contemporary Aspects of Philosophy*, pp. 30–56. Oriel Press.

Belnap, N. D. (1977b). A useful four-valued logic. In G. Epstein and J. M. Dunn (Eds.), *Modern Uses of Multiple-Valued Logic*, pp. 5–37. Dordrecht, Netherlands: Reidel.

Brandenburger, A. (1989). The role of common knowledge assumptions in game theory. In F. Hahn (Ed.), *The Economics of Information, Games, and Missing Markets.* Oxford, U.K.: Oxford University Press.

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

Clark, H. H. and C. R. Marshall (1981). Definite reference and mutual knowledge. In A. K. Joshi, B. L. Webber, and I. A. Sag (Eds.), *Elements of discourse understanding.* Cambridge, U.K.: Cambridge University Press.

Cresswell, M. J. (1973). *Logics and Languages.* London: Methuen and Co.

Dolev, D. and H. R. Strong (1982). Requirements for agreement in a distributed system. In H. J. Schneider (Ed.), *Distributed Data Bases*, pp. 115–129. Amsterdam: North-Holland.

Dunn, J. M. (1986). Relevance logic and entailment. In D. Gabbay and F. Guenthner (Eds.), *Handbook of Philosophical Logic, Vol. III*, pp. 117–224. Dordrecht, Netherlands: Reidel.

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

Eberle, R. A. (1974). A logic of believing, knowing and inferring. *Synthese 26*, 356–382.

Fagin, R. (Ed.) (1994). *Theoretical Aspects of Reasoning about Knowledge: Proc. Fifth Conference*. San Francisco, Calif.: Morgan Kaufmann.

Fagin, R. and J. Y. Halpern (1988). Belief, awareness, and limited reasoning. *Artificial Intelligence 34*, 39–76.

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

Fagin, R., J. Y. Halpern, Y. Moses, and M. Y. Vardi (1994). Knowledge-based programming. Research Report RJ 9711, IBM.

Fagin, R., J. Y. Halpern, Y. Moses, and M. Y. Vardi (to appear, 1995). *Reasoning about Knowledge*. Cambridge, Mass.: MIT Press.

Fagin, R., J. Y. Halpern, and M. Y. Vardi (1990). A nonstandard approach to the logical omniscience problem. In R. Parikh (Ed.), *Theoretical Aspects of Reasoning about Knowledge: Proc. Third Conference*, pp. 41–55. San Francisco, Calif.: Morgan Kaufmann. To appear in *Artificial Intelligence*.

Fagin, R., J. Y. Halpern, and M. Y. Vardi (1991). A model-theoretic analysis of knowledge. *Journal of the ACM 91*(2), 382–428. A preliminary version appeared in *Proc. 25th IEEE Symposium on Foundations of Computer Science*, 1984.

Fischer, M. J. and N. Immerman (1986). Foundations of knowledge for distributed systems. In J. Y. Halpern (Ed.), *Theoretical Aspects of Reasoning about Knowledge: Proc. 1986 Conference*, pp. 171–186. San Francisco, Calif.: Morgan Kaufmann.

Friedman, N. and J. Y. Halpern (1994a). A knowledge-based framework for belief change. Part I: Foundations. In R. Fagin (Ed.), *Theoretical Aspects of Reasoning about Knowledge: Proc. Fifth Conference*, pp. 44–64. San Francisco, Calif.: Morgan Kaufmann.

Friedman, N. and J. Y. Halpern (1994b). A knowledge-based framework for belief change. Part II: revision and update. In J. Doyle, E. Sandewall, and P. Torasso (Eds.), *Principles of Knowledge Representation and Reasoning: Proc. Fourth International Conference (KR '94)*, pp. 190–201. San Francisco, Calif.: Morgan Kaufmann.

Geanaokoplos, J. and H. Polemarchakis (1982). We can't disagree forever. *Journal of Economic Theory 28*(1), 192–200.

Gettier, E. (1963). Is justified true belief knowledge? *Analysis 23*, 121–123.

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

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

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

Halpern, J. Y. (1986a). Reasoning about knowledge: an overview. In J. Y. Halpern (Ed.), *Theoretical Aspects of Reasoning about Knowledge: Proc. 1986 Conference*, pp. 1–17. San Francisco, Calif.: Morgan Kaufmann. Reprinted in *Proc. National Computer Conference*, 1986, pp. 219–228.

Halpern, J. Y. (Ed.) (1986b). *Theoretical Aspects of Reasoning about Knowledge: Proc. 1986 Conference*. San Francisco, Calif.: Morgan Kaufmann.

Halpern, J. Y. (1987). Using reasoning about knowledge to analyze distributed systems. In J. F. Traub, B. J. Grosz, B. W. Lampson, and N. J. Nilsson (Eds.), *Annual Review of Computer Science, Vol. 2*, pp. 37–68. Palo Alto, Calif.: Annual Reviews Inc.

Halpern, J. Y. (1993a). Reasoning about knowledge: a survey circa 1991. In A. Kent and J. G. Williams (Eds.), *Encyclopedia of Computer Science and Technology, Volume 27 (Supplement 12)*, pp. 275–296. New York: Marcel Dekker.

Halpern, J. Y. (1993b). Reasoning about only knowing with many agents. In *Proc. National Conference on Artificial Intelligence (AAAI '93)*, pp. 655–661.

Halpern, J. Y. and R. Fagin (1989). Modelling knowledge and action in distributed systems. *Distributed Computing 3*(4), 159–179. A preliminary version appeared in *Proc. 4th ACM Symposium on Principles of Distributed Computing*, 1985, with the title "A formal model of knowledge, action, and communication in distributed systems: Preliminary report".

Halpern, J. Y. and Y. Moses (1984). Towards a theory of knowledge and ignorance. In *Proc. AAAI Workshop on Non-monotonic Logic*, pp. 125–143. Reprinted in K. Apt (Ed.), *Logics and Models of Concurrent Systems*, Springer-Verlag, Berlin/New York, pp. 459–476, 1985.

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

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

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

Halpern, J. Y., Y. Moses, and M. Y. Vardi (1994). Algorithmic knowledge. In R. Fagin (Ed.), *Theoretical Aspects of Reasoning about Knowledge: Proc. Fifth Conference*, pp. 255–266. San Francisco, Calif.: Morgan Kaufmann.

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

Halpern, J. Y. and M. R. Tuttle (1993). Knowledge, probability, and adversaries. *Journal of the ACM 40*(4), 917–962.

Halpern, J. Y. and M. Y. Vardi (1988). The complexity of reasoning about knowledge and time in asynchronous systems. In *Proc. 20th ACM Symp. on Theory of Computing*, pp. 53–65.

Halpern, J. Y. and M. Y. Vardi (1989). The complexity of reasoning about knowledge and time, I: lower bounds. *Journal of Computer and System Sciences 38*(1), 195–237.

Halpern, J. Y. and M. Y. Vardi (1991). Model checking vs. theorem proving: a manifesto. In J. A. Allen, R. Fikes, and E. Sandewall (Eds.), *Principles of Knowledge Representation and Reasoning: Proc. Second International Conference (KR '91)*, pp. 325–334. San Francisco, Calif.: Morgan Kaufmann. An expanded version appears in *Artificial Intelligence and Mathematical Theory of Computation (Papers in Honor of John McCarthy)* (ed. V. Lifschitz), Academic Press, 1991, pp. 151–176.

Halpern, J. Y. and L. D. Zuck (1992). 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.

Heyting, A. (1956). *Intuitionism: An Introduction.* Amsterdam: North-Holland.

Hintikka, J. (1962). *Knowledge and Belief.* Ithaca, N.Y.: Cornell University Press.

Hintikka, J. (1975). Impossible possible worlds vindicated. *Journal of Philosophical Logic 4*, 475–484.

Hopcroft, J. E. and J. D. Ullman (1979). *Introduction to Automata Theory, Languages and Computation.* New York: Addison-Wesley.

Katsuno, H. and A. Mendelzon (1991). On the difference between updating a knowledge base and revising it. In *Principles of Knowledge Representation and Reasoning: Proc. Second International Conference (KR '91)*, pp. 387–394. San Francisco, Calif.: Morgan Kaufmann.

Konolige, K. (1986). *A Deduction Model of Belief.* San Francisco, Calif.: Morgan Kaufmann.

Kripke, S. (1963). A semantical analysis of modal logic I: normal modal propositional calculi. *Zeitschrift für Mathematische Logik und Grundlagen der Mathematik 9*, 67–96. Announced in *Journal of Symbolic Logic*, **24**, 1959, p. 323.

Ladner, R. E. (1977). The computational complexity of provability in systems of modal propositional logic. *SIAM Journal on Computing 6*(3), 467–480.

Lakemeyer, G. (1986). Steps towards a first-order logic of explicit and implicit belief. In J. Y. Halpern (Ed.), *Theoretical Aspects of Reasoning about Knowledge: Proc. 1986 Conference*, pp. 325–340. San Francisco, Calif.: Morgan Kaufmann.

Lakemeyer, G. (1987). Tractable meta-reasoning in propositional logics of belief. In *Proc. Tenth International Joint Conference on Artificial Intelligence (IJCAI '87)*, pp. 402–408.

Lakemeyer, G. (1993). All they know: a study in multi-agent autoepestemic reasoning. In *Proc. Thirteenth International Joint Conference on Artificial Intelligence (IJCAI '93)*, pp. 376–381.

Lakemeyer, G. and H. J. Levesque (1988). A tractable knowledge representation service with full introspection. In M. Y. Vardi (Ed.), *Proc. Second Conference on Theoretical Aspects of Reasoning about Knowledge*, pp. 145–159. San Francisco, Calif.: Morgan Kaufmann.

Lenzen, W. (1978). Recent work in epistemic logic. *Acta Philosophica Fennica 30*, 1–219.

Levesque, H. J. (1984a). Foundations of a functional approach to knowledge representation. *Artificial Intelligence 23*, 155–212.

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

Levesque, H. J. (1990). All I know: A study in autoepistemic logic. *Artificial Intelligence 42*(3), 263–309.

Lewis, D. (1969). *Convention, A Philosophical Study.* Cambridge, Mass.: Harvard University Press.

Lin, F. and Y. Shoham (1990). Epistemic semantics for fixed-point nonmonotonic logics. In *Theoretical Aspects of Reasoning about Knowledge: Proc. Third Conference*, pp. 111–120. San Francisco, Calif.: Morgan Kaufmann.

Mazer, M. S. (1990). A link between knowledge and communication in faulty distributed systems. In R. Parikh (Ed.), *Theoretical Aspects of Reasoning about Knowledge: Proc. Third Conference*, pp. 289–304. San Francisco, Calif.: Morgan Kaufmann.

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

McCarthy, J. and P. J. Hayes (1969). Some philosophical problems from the standpoint of artificial intelligence. In D. Michie (Ed.), *Machine Intelligence 4*, pp. 463–502. Edinburgh: Edinburgh University Press.

Mertens, J. F. and S. Zamir (1985). Formulation of Bayesian analysis for games of incomplete information. *International Journal of Game Theory 14*(1), 1–29.

Montague, R. (1960). Logical necessity, physical necessity, ethics, and quantifiers. *Inquiry 4*, 259–269.

Moore, R. C. (1985). A formal theory of knowledge and action. In J. Hobbs and R. C. Moore (Eds.), *Formal Theories of the Commonsense World*, pp. 319–358. Norwood, N.J.: Ablex Publishing Corp.

Moore, R. C. and G. Hendrix (1979). Computational models of beliefs and the semantics of belief sentences. Technical Note 187, SRI International, Menlo Park, Calif.

Morgenstern, L. (1986). A first order theory of planning, knowledge, and action. In J. Y. Halpern (Ed.), *Theoretical Aspects of Reasoning about Knowledge: Proc. 1986 Conference*, pp. 99–114. San Francisco, Calif.: Morgan Kaufmann.

Moses, Y. (1988). Resource-bounded knowledge. In M. Y. Vardi (Ed.), *Proc. Second Conference on Theoretical Aspects of Reasoning about Knowledge*, pp. 261–276. San Francisco, Calif.: Morgan Kaufmann.

Moses, Y. (Ed.) (1992). *Theoretical Aspects of Reasoning about Knowledge: Proc. Fourth Conference.* San Francisco, Calif.: Morgan Kaufmann.

Moses, Y. and G. Roth (1989). On reliable message diffusion. In *Proc. 8th ACM Symp. on Principles of Distributed Computing*, pp. 119–128.

Moses, Y. and Y. Shoham (1993). Belief as defeasible knowledge. *Artificial Intelligence 64*(2), 299–322.

Moses, Y. and M. R. Tuttle (1988). Programming simultaneous actions using common knowledge. *Algorithmica 3*, 121–169.

Neiger, G. (1988). Knowledge consistency: a useful suspension of disbelief. In M. Y. Vardi (Ed.), *Proc. Second Conference on Theoretical Aspects of Reasoning about Knowledge*, pp. 295–308. San Francisco, Calif.: Morgan Kaufmann.

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

Panangaden, P. and S. Taylor (1992). Concurrent common knowledge: Defining agreement for asynchronous systems. *Distributed Computing 6*(2), 73–94.

Parikh, R. (1991). Monotonic and nonmonotonic logics of knowledge. *Fundamenta Informaticae 15*(3,4), 255–274.

Parikh, R. and P. Krasucki (1990). Communication, consensus, and knowledge. *Journal of Economic Theory 52*(1), 178–189.

Parikh, R. and R. Ramanujam (1985). Distributed processing and the logic of knowledge. In R. Parikh (Ed.), *Proc. Workshop on Logics of Programs*, pp. 256–268.

Parikh, R. J. (Ed.) (1990). *Theoretical Aspects of Reasoning about Knowledge: Proc. Third Conference*. San Francisco, Calif.: Morgan Kaufmann.

Patel-Schneider, P. F. (1985). A decidable first-order logic for knoweledge representation. In *Proc. Ninth International Joint Conference on Artificial Intelligence (IJCAI '85)*, pp. 455–458.

Pease, M., R. Shostak, and L. Lamport (1980). Reaching agreement in the presence of faults. *Journal of the ACM 27*(2), 228–234.

Perrault, C. R. and P. R. Cohen (1981). It's for your own good: a note on inaccurate reference. In A. K. Johsi, B. L. Webber, and I. A. Sag (Eds.), *Elements of discourse understanding*. Cambridge, U.K.: Cambridge University Press.

Rantala, V. (1982). Impossible worlds semantics and logical omniscience. *Acta Philosophica Fennica 35*, 18–24.

Rescher, N. and R. Brandom (1979). *The Logic of Inconsistency*. Totowa, N.J.: Rowman and Littlefield.

Rosenschein, S. J. (1985). Formal theories of AI in knowledge and robotics. *New Generation Computing 3*, 345–357.

Rosenschein, S. J. and L. P. Kaelbling (1986). The synthesis of digital machines with provable epistemic properties. In J. Y. Halpern (Ed.), *Theoretical Aspects of Reasoning about Knowledge: Proc. 1986 Conference*, pp. 83–97. San Francisco, Calif.: Morgan Kaufmann.

Shoham, Y. (1988). Chronological ignorance: experiments in nonmonotonic temporal reasoning. *Artificial Intelligence 36*, 271–331.

Shoham, Y. (1993). Agent oriented programming. *Artificial Intelligence 60*(1), 51–92.

Stark, W. R. (1981). A logic of knowledge. *Zeitschrift für Mathematische Logik und Grundlagen der Mathematik 27*, 371–374.

Vardi, M. Y. (1985). A model-theoretic analysis of monotonic knowledge. In *Proc. Ninth International Joint Conference on Artificial Intelligence (IJCAI '85)*, pp. 509–512.

Vardi, M. Y. (Ed.) (1988). *Proc. Second Conference on Theoretical Aspects of Reasoning about Knowledge*. San Francisco, Calif.: Morgan Kaufmann.

Vardi, M. Y. (1989). On the complexity of epistemic reasoning. In *Proc. 4th IEEE Symp. on Logic in Computer Science*, pp. 243–252.

Wansing, H. (1990). A general possible worlds framework for reasoning about knowledge and belief. *Studia Logica 49*(4), 523–539.

Wright, G. H. v. (1951). *An Essay in Modal Logic*. Amsterdam: North-Holland.

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