From: AAAI-86 Proceedings. Copyright ©1986, AAAI (www.aaai.org). All rights reserved.

What Can Machines Know? On the Epistemic Properties of Machines

Ronald Fagin
Joseph Y. Halpern
Moshe Y. Vardi

IBM Almaden Research Center
650 Harry Road
San Jose, California 95120-6099

Abstract: It has been argued that knowledge is a useful tool for designing and analyzing complex systems in AI. The notion of knowledge that seems most relevant in this context is an *external, information-based* notion that can be shown to satisfy all the axioms of the modal logic S5. We carefully examine the properties of this notion of knowledge, and show that they depend crucially, and in subtle ways, on assumptions we make about the system. We present a formal model in which we can capture the types of assumptions frequently made about systems (such as whether they are deterministic or nondeterministic, whether knowledge is *cumulative*, and whether or not the environment affects the transitions of the system). We then show that under some assumptions certain states of knowledge are *not* attainable, and the axioms of S5 do not completely characterize the properties of knowledge; extra axioms are needed. We provide complete axiomatizations for knowledge in a number of cases of interest.

1. Introduction

A fundamental problem in many branches of AI and computer science (including planning, distributed computing systems, and robotics) is to design, analyze, and understand complex systems composed of interacting parts. An increasingly useful tool in this design and analysis process is the concept of *knowledge*. In AI, there have been two approaches to ascribing knowledge to machines or components of systems. The classical AI approach, which has been called the *interpreted-symbolic-structures* approach ([Ro]), ascribes knowledge on the basis of the information stored in certain data structures (such as semantic nets, frames, or data structures to encode formulas of predicate logic; (cf. [BL]). The second, called the *situated-automata* approach, can be viewed as ascribing knowledge on the basis of the information carried by the state of the machine ([Ro]).

Since we concentrate on the second approach in this paper, we describe the intuition in more detail. Imagine a machine composed of various components, each of which may be in various states. (Although we talk here of a "machine composed of components", everything we say goes through perfectly well for a system of sensors taking readings, or a distributed system composed of robots, processes, or people, observing the environment.) We assume some sort of environment about which the system gains information. At any point in time, the system is in some *global state*, defined by the state of the environment and the local states of the components. We say a process or component p *knows* a fact φ in global state s if φ is true in all global states s' of the system where p has the same local state as it does in s.

This notion of knowledge is *external*. A process cannot answer questions based on its knowledge with respect to this notion of knowledge. Rather, this is a notion meant to be used by the system designer reasoning about the system. This approach to knowledge has attracted a great deal of interest recently among researchers in both AI ([Ro,RK]) and distributed systems ([HM1, PR, HF, CM, FI]) precisely because it does seem to capture the type of intuitive reasoning that goes on by system designers. (See [RK] for some detailed examples.)

If we are to use this notion of knowledge to analyze systems, then it becomes important to understand its properties. It is easy to show that the external notion of knowledge satisfies all the axioms of the classical modal logic S5 (we discuss this in more detail in Section 2; an overview of S5 and related logics of knowledge and belief can be found in [HM2]). Indeed, the abstract model most frequently used to capture this notion (for example, in [Ro]), has been the classical Kripke-style *possible-worlds* model for S5 ([Kr]). But, *a priori*, it is not the least bit clear that this is the appropriate abstraction for the notion of knowledge in which we are interested. Does each Kripke structure really correspond to some state of knowledge that the system can be in? As we shall show, the answer to this question depends crucially, and in surprising ways, on the assumptions that one makes about the system.

In order to explain our results, it is helpful to briefly review some material from [FV] which directly inspired our work here. In [FV] a particular scenario is considered, which intuitively can be viewed as robots observing their environment and then communicating about their observations. These robots are assumed never to forget information they have learned. Moreover, the communication is assumed to proceed in rounds, and messages either arrive within one round or do not arrive at all. In addition, messages are assumed to be *honest*; for example, if Alice sends Bob a message φ, then it must be the case that Alice knows φ. Under these assumptions (and, as we shall see, under several more that are implicit in the model), it is shown that certain states of knowledge are not attainable. In particular, suppose we let p be a fact that characterizes the environment (for example, if all we care about is the weather in San Francisco, we could take p to be "It is raining in San Francisco"), and suppose we have a system with exactly two robots, say Alice and Bob. Consider a situation where Alice doesn't know whether p is true or false, and Alice knows that either p is true and Bob knows p, or p is false and Bob doesn't know that p is false. Alice's state of knowledge can be captured by the formula:

$$(*) \quad \begin{aligned} &\sim K_{Alice} p \wedge \sim K_{Alice} \sim p \wedge \\ &K_{Alice}((p \wedge K_{Bob} p) \vee (\sim p \wedge \sim K_{Bob} \sim p)). \end{aligned}$$

Although the state of knowledge described by this formula is perfectly consistent with the axioms of S5, it is not attainable under the assumptions of [FV].

To see that it is not attainable, suppose that it were attainable. Then we can reason as follows:

Suppose p is false. Then Alice's state of knowledge implies that neither Alice nor Bob knows that p is false. But if Bob sends Alice a message saying "I don't know p", then, since Alice knows that either p is true and Bob knows p or p is false and Bob doesn't know that p is false, Alice will know that p must be false. But it is impossible for Alice and Bob to discover that p is false simply by communicating if neither of them had any knowledge about p beforehand. So p must be true. And since this argument holds for all states where Alice has the same information, Alice knows p. But this contradicts the assumption that Alice doesn't know p.

In [FV] a formal proof of the unattainability of this state of knowledge is given for their model. In order to show the subtlety of the assumptions required to make the proof go through, we now give three situations where the state of knowledge *is* attainable. For the first case, suppose p is now the statement "the communication line between us is up". Suppose p is true and Alice sends Bob the message "Hello", which Bob receives (since, after all, the communication line is up). At this point, Bob knows p (since he received the message) and Alice doesn't know whether p is true or false (since she doesn't know whether Bob received her message). But Alice does know that either p is true and Bob knows it, or p is false and Bob doesn't know p is false (since if p is false, Bob will have no way of knowing whether he didn't receive a message because the line was down or because Alice didn't send one in the first place). Thus, we exactly have the state of knowledge previously shown to be unattainable!

Of course, there is nothing wrong with either proof. The first proof of impossibility breaks down in the latter scenario because of Alice's seeming innocuous assumption that Bob could send her (and she could receive) a message saying "I don't know p". If p is false in the latter scenario, then she could never receive such a message because the communication line would be down. Implicitly, there is an assumption in the model of [FV] that the primitive propositions talk about "nature" and have no impact on the transitions of the system. In particular, a proposition cannot say that a communication line is down or a message buffer is full.

Now suppose we slightly modify the example so that there is a communication link from Alice to Bob and a separate one from Bob to Alice. Further suppose that the link from Bob to Alice is guaranteed to be reliable. Let p say that the communication link from Alice to Bob is up. Just as before, suppose Alice sends Bob a message saying "Hello", which Bob gets. The same reasoning as above shows that we have again attained the "unattainable" state of knowledge. But in this case, Bob can send Alice a message saying "I don't know p", and Alice would be guaranteed to receive it. So now where does the reasoning in the proof of [FV] break down? This time it is in the assumption that Alice and Bob cannot gain knowledge of a fact that they did not even have implicit knowledge of beforehand (this is called the principle of "Conservation of Implicit Knowledge" in [FV]).[1] Although neither Alice nor Bob had any knowledge of p before the message was sent, when the message arrived Bob knew p was true, so implicit knowledge is gained in this situation. The point is that while implicit knowledge of the environment cannot be gained if the processes first observe the environment and then communicate about it (so that, intuitively, all transitions are independent of the environment), this may not be true in a more general setting.

A third critical assumption being made in the argument from [FV] is that neither robot forgets; i.e. their knowledge is cumulative. We implicitly assume that if at some point neither Alice nor Bob knows that p is false, then they never had any knowledge of p beforehand. But if knowledge is not cumulative, then it may have been the case that Bob knew that p was false, imparted a piece of this knowledge to Alice, and then forgot about it. For example, suppose Alice knows that if Bob knows p, he will never forget it, while if Bob knows $\sim p$, then he may forget it. Suppose in fact that p is true and Bob knows it, and Bob sends Alice two messages. The first one says "either I know p or I know $\sim p$" (i.e., $K_{Bob}p \vee K_{Bob}\sim p$), while the second says "I don't know that p is false" (i.e., $\sim K_{Bob}\sim p$). At this point, Alice knows that either p is true and Bob knows it, or that p is false and Bob doesn't know that p is false (he may have known this before and forgotten). Again, we have shown that the "unattainable" knowledge state is attainable!

While this example may seem a little contrived, it is in fact easy to capture if we view Alice and Bob as finite-state machines. Indeed, an agent that does not forget must in general have unbounded memory (in order to remember all the messages it has sent and received), so that, in a sense, a finite-state machine can be viewed as the canonical example of a machine that does forget.

In order to examine the properties of knowledge carefully, we define an abstract model of knowledge for machines. Using this model, we are able to give precise formulations of a number of parameters of interest when analyzing systems. The exact setting of the parameters depends, of course, on the system being analyzed, although some choices of parameters seem more common in AI applications than distributed systems applications, and vice versa. Typical parameters include:

- Is a process' knowledge cumulative? Most papers that consider reasoning about knowledge over time implicitly assume that knowledge is cumulative. Indeed, this is one of the major reasons that Moore ([Mo]) considers knowledge rather than belief. As Moore points out, "If we observe or perform a physical action, we generally know everything we knew before, plus whatever we have learned from the action." For example, when considering a robot learning a telephone number, we don't worry about the robot forgetting the number a few steps later. A similar assumption is often made in the distributed systems literature, (cf. [HM1,HF,Lc,PR,DM]). This assumption is, of course, an idealization, since cumulative knowledge in general requires unbounded memory. Bounded memory is a more realistic assumption (and has been used in papers such as ([CM,FI,RK]). But note that for limited interactions, knowledge can be cumulative even with bounded memory; all that is required is enough memory to store the history.[2]
- Are transitions of the system independent of the environment? In the case of processes or sensors observing the environment and then communicating about it, transitions would be independent if nothing about the state can effect the possibility of communication. But suppose we are observing the weather. One can well imagine that the presence of a heavy thunderstorm

[1] We discuss implicit knowledge formally in the next section. Roughly speaking, a system has implicit knowledge of a fact if by putting all their information together, the components of the system could deduce that fact.

[2] We remark that Halpern and Vardi have shown that that the assumption that processes' knowledge is cumulative has a drastic effect on the complexity of the decision procedure for validity of formulas involving knowledge and time ([HV]).

could affect communication, and so affect the transitions in the system.

- Is the system deterministic or nondeterministic? The answer to this question might depend partly on the granularity of our analysis. A system that seems deterministic at one level of analysis may seem nondeterministic if we get down to the level of electrons. Note that even if the individual processes or components of the system are deterministic, the system as a whole may be nondeterministic, since we may decide to ignore certain components of the system (such as a message buffer or a particular and-gate) in doing our analysis.
- Do we view the system as embedded in its environment, so that the initial state of the system is a (possibly nondeterministic) function of the environment, or do we take the system to be the total environment, so that the initial state of the system completely determines the state of the environment? The former is appropriate if we consider the system to consist of sensors observing nature, while the latter is more appropriate in certain distributed systems applications where we identify the "environment" with the initial setting of certain variables. Of course, there may well be applications for which some point between these poles might be more appropriate.
- Is the system synchronous or asynchronous?

The list of parameters mentioned above is not meant to be exhaustive. The interesting point for us is the subtle interplay between these parameters and the states of knowledge that are attainable. For example, if (1) processes'/components' knowledge is cumulative, (2) the system is embedded in the environment, and (3) transitions of the system are independent of the environment, then it turns out that the axiom system ML of [FV] gives a complete characterization of the knowledge states attainable (i.e., is sound and complete), independent of the choices of the other parameters. If we assume that the system is deterministic, we get yet another axiom. On the other hand, if we assume that knowledge is not cumulative or that the state of the environment can affect the transitions of the system, we find that S5 does provide a complete characterization of the states of knowledge attainable. To us, the moral of this story is that a reasonable analysis of a system in terms of knowledge must take into account the relationship between the properties of the system and the properties of knowledge.

The rest of the paper is organized as follows. In Section 2 we describe our abstract model and show how all the various assumptions we would like to consider can be captured in the model. In Section 3 we briefly review the semantics of knowledge in systems. In Section 4 we characterize what states of knowledge are attainable under a number of different reasonable assumptions about systems. We conclude in Section 5 with some directions for further research.

2. The model

Consider a system with n processes (or components). A global state of the system is a tuple that describes the state of the environment and the local state of each process. We consider the system to be evolving over time. A complete description of one particular way the system could have evolved is a run. We identify a system with a set of runs.

More formally, a *system* M is a tuple (E, C, R, L, g), where E is a set of *primitive environment states*; C is a finite set of *processes*, which, for convenience, we shall usually take to be the set $\{1,...,n\}$ if n is the total number of processes; R is a set of *runs*; L is the set of *local states* that the processes can take on; and g associates with each run $r \in R$ and each natural number $m \in \mathbb{N}$ (which we are viewing as a time) a *global state* $g(r,m)$, where a global state is a tuple $<e,l_1,...,l_n>$, with $e \in E$ and $l_i \in L$ for $i = 1,...,n$. Following [HM1], we may refer to the pair (r,m) as a *point*.

A few comments about the model are now in order.

We view a primitive environment state as being a complete description of "nature" (or whatever the domain of discourse is). We could instead have started with a set of primitive propositions, say $p_1,...,p_m$. In this case a primitive environment state would just be one of the 2^m truth assignments to the primitive propositions. We prefer to start with these primitive environment states, since they seem to us more basic than primitive propositions (and, as we shall see, our axioms are more naturally expressed in terms of them), but everything we say can be easily reformulated in terms of primitive propositions.

For the rest of this paper we assume that the primitive environment state does not change throughout the run. Formally, for all runs $r \in R$ and all $m, m' \in \mathbb{N}$, if $g(r,m) = (e, l_1, ..., l_n)$ and $g(r,m') = (e', l'_1, ..., l'_n)$, then $e = e'$. One can certainly imagine applications where the environment does change over time (if we have sensors observing some terrain, we surely cannot assume that the terrain does not change over time!). But even in such applications the sensors usually communicate about a particular reading taken at a particular time. In this case we can think of the primitive environment states as describing the possible states of the environment at that time.

We have taken time here to be discrete (ranging over the natural numbers). This is mainly an assumption of convenience. We could have taken time to range, for example, over the non-negative reals, and defined a global state $g(r,t)$ at all non-negative real times t; none of the essential ideas would change in this case. Of course, we do not assume that there is necessarily a source of time within the system. Time is external to the system, just like knowledge.

Note that we have made no commitment here as to how the transitions between global states occur. There is no notion of messages or communication in our model, as there is, for example, in the model of [HF]. While it is easy to incorporate messages and have the transitions occurring as a result of certain messages occurring, transitions might also be a result of physical interactions or even random events internal to some component.

As it stands, this model is still too general for many purposes. We now discuss how a number of reasonable restrictions can be captured in our model. There are two general types of restrictions we consider: restrictions on the possible initial global states and restrictions on the possible transitions. In terms of knowledge, these restrictions can be viewed as restrictions on the initial state of knowledge and restrictions on how knowledge can be acquired.

Definition 2.1. Fix a system $M = (E, C, R, L, g)$. We say $s = <e, l_1, ..., l_n>$ is a *global state in* M if $s = g(r,m)$ for some run $r \in R$ and time m; e is the *environment component* of s while $<l_1,...,l_n>$ is the *process component*. Let $s = <e, l_1, ..., l_n>$ and $s' = <e', l'_1, ..., l'_n>$ be two global states in M. We say s and s' are *indistinguishable to process* i, written $s \sim_i s'$, if $l_i = l'_i$. We say s and s' are *process equivalent* if they are indistinguishable to all processes $i = 1,...,n$, i.e., if s and s' have the same process component. *Process i's view of run r up to time m* is the sequence $l_0, ..., l_k$ of states that process i takes on in run r up to time m, with consecutive repetitions omitted. For example, if from time 0 through 4 in run r process i goes through

the sequence l, l, l', l, l of states, its view is just l, l', l. Finally, s is an *initial state* if $s = g(r, 0)$ for some run r.

We can now precisely state a few reasonable restrictions on systems.

1. Restrictions on possible initial states.
   a. In many applications we view the system as embedded in an environment, where the processes' initial states are a function of observations made in that environment. Thus if process i is placed in environment e, then its initial state is some function of e. This function is in general not deterministic; i.e., for a given state of the environment, there may be a number of initial local states that a given process can be in. Formally, we say that *the environment determines the initial states* if for each process i and each primitive state e, there is a subset $L(i,e)$ of local states such that the set of initial states is $\{<e,l_1,...,l_n> \mid l_i \in L(i,e)\}$. Intuitively, $L(i,e)$ is the set of states that i could be in initially if the environment is in state e. If we imagine that i is a sensor, then these states represent all the ways i could have partial information about e. For example, if facts p and q are true in an environment e, we can imagine a sensor that sometimes may observe both p and q, neither, or just one of the two; it would have a different local state in each case. Note we have also implicitly assumed that there is initially no interaction between the observations. That is, if in a given primitive environment state e it is possible for process i to make an observation that puts it into state l_i and for j to make an observation that puts it into l_j, then it is possible for both of these observations to happen simultaneously. This assumption precludes a situation where, for example, exactly one of two processes can make a certain observation. An important special case occurs when the initial state of the processes is a deterministic function of the environment. We say *the environment uniquely determines the initial state* if the environment determines the initial states and, in addition, if $L(i,e)$ is a singleton set for all i and e.
   b. At the other extreme, we have the view that the system determines the environment. For example, in some distributed systems applications we may want to take the "environment" to be just the initial setting of certain local variables in each process. We say *the initial state uniquely determines the environment* if, whenever two initial global states are process equivalent, they are in fact identical.

   Of course, many situations between these extremes are possible.
2. Restrictions on state transitions.
   a. If a process' knowledge is cumulative, then the process can and does "remember" its view of a run. Thus, if two global states are indistinguishable to process i, then it must be the case that process i has the same view of the run in both. More formally, *knowledge is cumulative* if for all processes i, all runs r, r', and all times m, m', if $g(r,m) \sim_i g(r',m')$, then process i's view of run r up to time m is identical to its view of run r' up to time m'. Note that cumulative knowledge requires an unbounded number of local states in general, since it must be possible to encode all possible views of a run in the state. In particular, the knowledge of finite-state machines will, in general, not be cumulative.
   b. In a *synchronous system*, every process has access to a global clock that ticks at every instant of time, and the clock reading is part of its state. Thus, in a synchronous system, each process always "knows" the time. Note that in particular, this means that in a synchronous system processes have unbounded memory. More formally, we say a system is synchronous if for all processes i and runs r, if $g(r,m) \sim_i g(r,m')$, then $m = m'$. An easy proof (by induction on m) shows that in a synchronous system where knowledge is cumulative, if $g(r,m) \sim_i g(r',m')$, then $m = m'$ and, if $m > 0$, $g(r,m-1) \sim_i g(r',m-1)$. A system that is not synchronous is called *asynchronous*.
   c. We say that *transitions are independent of the environment* if, whenever we have two process-equivalent initial states, then the same sequence of transitions is possible from each of them; i.e., if $s = <e, l_1, ..., l_n>$ and $s' = <e', l_1, ..., l_n>$ are process-equivalent initial states and r is a run with initial state s (i.e., $g(r,0) = s$), then there is a run r' with initial state s' such that $g(r,m)$ is process equivalent to $g(r',m)$ for all times m.
   d. We say a system is *deterministic* if the initial state completely determines the run; i.e., whenever r and r' are runs with $g(r,0) = g(r',0)$, then $r = r'$.[3]
   e. Note that in both of the previous definitions we considered only initial states. Even if transitions are independent of the environment, it will not in general be the case that the same sequence of transitions is possible starting from two arbitrary process-equivalent global states, since the transitions may depend on the whole history of the run, including, for example, messages that were sent but did not yet arrive. Similarly, even in a deterministic system there may be two different transitions possible from a given global state, depending on the previous history. The point is that there may be some information about the system not described in the global state (such as the fact that certain messages have not yet been delivered). Intuitively, this "incompleteness" in the global state arises because we choose not to describe certain features of the system. For example, we may choose the components of C to be only the processors in the system, ignoring the message buffers. We say there are *no hidden components* in a system if, whenever $r, r' \in R$ are two runs such that $g(r,m) = g(r',m')$, then there is a run $r'' \in R$ which has the same prefix as r up to time m and continues as r' (i.e., $g(r'',k) = g(r,k)$ if $k \leq m$, and $g(r'',k) = g(r', m' + k - m)$ if $k \geq m$). Intuitively, since the global state contained all the relevant information, starting with r it could have been the case that from time m on we could have continued as in run r'. Note that in a deterministic system with no hidden components, if $g(r,m) = g(r',m')$, then $g(r,m+1) = g(r',m'+1)$. Similarly, in a system where transitions are independent of the environment with no hidden components, the same sequence of transitions is

[3] It is not hard to see that in a deterministic system where transitions are independent of the environment, the initial process component completely determines the run; i.e. if r and r' are runs where $g(r,0)$ and $g(r',0)$ are process-equivalent, then $g(r,m)$ and $g(r',m)$ are process-equivalent for every m.

always possible from two process equivalent global states.

We have outlined a few reasonable restrictions on possible initial states and on state transitions. Certainly it is possible to come up with others. The main point we want to make here is that many reasonable restrictions on systems can be easily captured within our model.

3. States of knowledge

Consider the language that results when we take primitive environment states $e, e', \ldots$, and close off under negation, conjunction, and knowing, so that if φ and φ' are formulas, so are $\sim\varphi$, $\varphi \wedge \varphi'$, and $K_i\varphi$, $i = 1, \ldots, n$. We also find it convenient at times to have *implicit knowledge* in the language. Intuitively, implicit knowledge (which is formally introduced in [HM2] and has also been used in [CM, DM, FV, RK]) is the knowledge that can be obtained when the members of a group pool their knowledge together. Put differently, it is what someone who had all the knowledge that each member in the group had could infer. We use $I\varphi$ to denote implicit knowledge of φ.

We now define what it means for a formula φ in the language to be true at time m in run r of system $M = (E, C, R, L, g)$, written $M, r, m \models \varphi$:

$M, r, m \models e$, where e is a primitive environment state, if $g(r, m) = < e, \ldots >$

$M, r, m \models \sim\varphi$ if $M, r, m \not\models \varphi$

$M, r, m \models \varphi_1 \wedge \varphi_2$ if $M, r, m \models \varphi_1$ and $M, r, m \models \varphi_2$

$M, r, m \models K_i\varphi$ if $M, r', m' \models \varphi$ for all r', m' such that $g(r, m) \sim_i g(r', m')$

$M, r, m \models I\varphi$ if $M, r', m' \models \varphi$ for all r', m' such that $g(r, m)$ is process equivalent to $g(r', m')$.

It is helpful to comment on the last two clauses of the above definition, which describe when $K_i\varphi$ and $I\varphi$ hold. Let $S_i = \{(r', m') \mid g(r, m) \sim_i g(r', m')\}$. Intuitively, $(r', m') \in S_i$ precisely if at time m in run r, it is possible, as far as process i is concerned, that it is time m' in run r'. It is easy to verify that $K_i\varphi$ holds at time m in run r precisely if φ holds at every point in S_i. Let S be the intersection of the S_i's. Intuitively, $(r', m') \in S$ precisely if at time m in run r, if all of the processes were to combine their information then they would still consider it possible that it is time m' in run r'. It is easy to verify that $g(r, m)$ is process equivalent to $g(r', m')$ precisely if $(r', m') \in S$. Thus, $I\varphi$ holds at time m in run r precisely if φ holds at every point in S.

Definition 3.1. A formula φ is *valid* if $M, r, m \models \varphi$ for all systems M, runs r, and times m.

It is easy to see that the truth of a formula depends only on the global state; i.e., if $g(r, m) = g(r', m')$, then for all formulas φ, we have $M, r, m \models \varphi$ iff $M, r', m' \models \varphi$. This way of assigning truth gives us a way of ascribing knowledge to components of a system in a given global state. But we still have not defined the notion of a *state of knowledge*. What is the state of knowledge of a system in a given global state? We could identify the state of knowledge with the set of formulas true in the global state, but this definition is too dependent on the particular language chosen. Instead, we give a semantic definition of a state of knowledge. We first need a preliminary definition.

Definition 3.2. A global state s' is *reachable from s in S* if there exist global states $s_0, \ldots, s_k$ in S and (not necessarily distinct) processes $i_1, \ldots, i_k$ such that $s = s_0$, $s' = s_k$, and $s_{j-1} \sim_{i_j} s_j$ for $j = 1, \ldots, k$.

Intuitively, the state of knowledge of the system in global state s depends only on the global states reachable from s. This is borne out in our formal semantics by the fact that the truth of a formula at time m in run r only depends on the global states reachable from $g(r, m)$. Thus, we have the following definition.

Definition 3.3. A *state of knowledge* is a pair (S, s) where S is a set of global states, $s \in S$ is a global state, and every member of S is reachable from s in S. A state (S, s) is *attainable in a system M* if there is a global state s in M such that S consists precisely of those states reachable from s.

In the full paper ([FHV]) we review the classical Kripke semantics for the modal logic S5 and define the analogue of the notion of state of knowledge for Kripke structures. It is fairly easy to show that there is an exact correspondence between states of knowledge in our model and states of knowledge in Kripke structures. This perhaps justifies the choice of Kripke structures as an appropriate abstraction for the notion of knowledge in systems. However, as we show in the next section, under some of the restrictions on systems we have discussed, not all states of knowledge are attainable.

4. The properties of knowledge

We shall not try to give here a complete taxonomy of the properties of knowledge for each choice of parameters that we have discussed (although in the full paper, we do characterize the properties of knowledge for many cases of interest). Instead, we discuss a few illustrative cases, with a view towards showing the subtlety of the interaction between the properties of the system and the properties of knowledge.

As we remarked above, if we put no restrictions on systems, then there is an exact correspondence between states of knowledge in our model and those in Kripke structures. It is well-known that the axiom system S5 captures the properties of knowledge in Kripke structures; i.e., it is *sound* (all the axioms are valid) and *complete* (all valid formulas are provable). $S5_n$ (the extension of the classical axiom system S5 to a situation with n knowers) consists of the following axioms and rules of inference. The axioms are:

A1. All substitution instances of propositional tautologies.[4]

A2. $K_i\varphi_1 \wedge K_i(\varphi_1 \Rightarrow \varphi_2) \Rightarrow K_i\varphi_2$, $i = 1, \ldots, n$

A3. $K_i\varphi \Rightarrow \varphi$, $i = 1, \ldots, n$

A4. $K_i\varphi \Rightarrow K_iK_i\varphi$, $i = 1, \ldots, n$

A5. $\sim K_i\varphi \Rightarrow K_i\sim K_i\varphi$.

There are two rules of inference: modus ponens ("from φ_1 and $\varphi_1 \Rightarrow \varphi_2$ infer φ_2") and knowledge generalization ("from φ infer $K_i\varphi$").

If we extend the language to include implicit knowledge, then we obtain a complete axiomatization by adding axioms that say that I acts like a knowledge operator (i.e., all the axioms above hold with K_i replaced by I) and the additional axiom:

$K_i\varphi \Rightarrow I\varphi$, $i = 1, \ldots, n$.

In [HM2] it is shown that this axiomatization, called $S5I_n$, is sound and complete in Kripke structures for the extended language with implicit knowledge.

For various reasons, philosophers have argued that S5 is an inappropriate axiom system for modelling human knowledge. For example, axiom A2 seems to assume perfect reasoners, that know all logical consequences of their knowledge, while A5 assumes that reasoners can

---

4 Since our base language consists of primitive environment states rather than primitive propositions, we also have tautologies of the form $e \Rightarrow \sim e'$ if e, e' are distinct primitive environment states. In addition, if we have only finitely many primitive environment states, say $e_1, \ldots, e_N$, then $e_1 \vee \ldots \vee e_N$ is a tautology.

do *negative introspection*, and know about their lack of knowledge. While these axioms may be controversial for some notions of knowledge, they are not controversial for the external, information-based notion that we are concerned with here. Moreover, it is easy to see that all of these axioms are still sound even under the restrictions on systems discussed in Section 2. But of course, they may no longer be complete.

Recall the Alice and Bob story discussed in the introduction. What assumptions were really needed to show that the state of knowledge defined by formula (*) was not attainable? As the counterexamples given in the introduction suggest, we need to assume cumulative knowledge (i.e., no "forgetting") and that the environment does not affect the transitions. Also implicit in the story is that Alice and Bob initially study nature independently, so we also need the assumption that the environment determines the initial states. It turns out that these three conditions are sufficient to show that (*) is not attainable, as we shall see below.

Our first step is to get a semantic characterization of the attainable states of knowledge under these assumptions.

Definition 4.1. A knowledge state (S,s) satisfies the *pasting condition* if, whenever $s', s_1, ..., s_n$ are global states in S such that e is the environment component of s_i and $s' \sim_i s_i$, $i = 1, ..., n$, then there exists a global state s'' in S such that s'' is process equivalent to s' and e is the environment component of s''. Thus, a knowledge state (S,s) satisfies the pasting condition if, whenever $s' = (\cdot, l_1, ..., l_n) \in S$, and also $s_1 = (e, l_1, \cdot, ..., \cdot) \in S$, $s_2 = (e, \cdot, l_2, \cdot, ..., \cdot) \in S$, ..., and $s_n = (e, \cdot, ..., \cdot, l_n) \in S$, then $s'' = (e, l_1, ..., l_n) \in S$. (Each $\cdot$ represents a value we do not care about.)

Proposition 4.2. *If M is a system where (1) knowledge is cumulative, (2) the environment determines the initial states, and (3) transitions are independent of the environment, then all the states of knowledge attainable in M satisfy the pasting condition. Conversely, if a state of knowledge satisfies the pasting condition, then it is attainable in some system M satisfying these three assumptions.*

Note that these assumptions are not unreasonable. They hold for "ideal" sensors or robots observing and communicating about an external environment.

Not surprisingly, the fact that the pasting condition holds affects the properties of knowledge. Neither $S5_n$ nor $S5I_n$ is complete. Consider the following axiom in the extended language, where e is a primitive environment state and $\{1, ..., n\}$ is the set of processes:

A6. $I{\sim}e \Rightarrow K_1{\sim}e \vee ... \vee K_n{\sim}e$.

This says that if it is implicit knowledge that the primitive environment state is not e, then it must be the case that some process knows it. The soundness of this axiom, which is *not* a consequence of $S5I_n$, is easily seen to follow from the pasting condition. We remark that the formula (*) discussed in the introduction is a consequence of S5 together with A6 (provided we assume that the primitive proposition p in formula (*) is a primitive environment state; recall that we said it "completely characterizes the environment").

Even without implicit knowledge in the language, we can get an axiom that captures some of the intuition behind the pasting condition. We define a *pure knowledge formula* to be a Boolean combination of formulas of the form $K_i\varphi$, where φ is arbitrary. For example, $K_2p \vee (K_1{\sim}K_3p \wedge {\sim}K_2{\sim}p)$ is a pure knowledge formula, but $p \wedge {\sim}K_ip$ is not. Consider the following axiom, where e is a primitive environment state and φ is a pure knowledge formula:

A6'. $K_i(\varphi \Rightarrow {\sim}e) \Rightarrow K_i(\varphi \Rightarrow (K_1{\sim}e \vee ... \vee K_n{\sim}e))$.

The intuition behind this rather mysterious formula is discussed in [FV]. Let ML_n (resp. ML_n^-) be $S5I_n$ (resp. $S5_n$) together with A6 (resp. A6').

Theorem 4.3. *ML_n^- (resp. ML_n) is a sound and complete axiomatization (for the extended language) for systems of n processes where (1) knowledge is cumulative, (2) the environment determines the initial states, and (3) transitions are independent of the environment.*

Soundness and completeness theorems for ML_n^- and ML_n are also proven by Fagin and Vardi in [FV], but in a rather different setting from ours. The model in [FV] is much more concrete than the model here; in particular there is in their model a particular notion of communication by which the system changes its state. Here we have an abstract model in which, by analyzing the implicit and explicit assumptions in [FV], we have captured the essential assumptions required for the pasting property and A6 to hold. While soundness in [FV] follows easily from soundness in the model here, they have to work much harder to prove completeness.

Recall from our Alice and Bob story in the introduction that the assumptions we made all seemed to be necessary. The following theorem confirms this fact. It shows that if we drop any one of assumptions (1), (2), or (3), all states of knowledge are attainable, and $S5_n$ (resp. $S5I_n$) becomes complete.

Theorem 4.4. *All states of knowledge are attainable in systems that satisfy only two of the restrictions of Proposition 4.2 and Theorem 4.3. Thus $S5_n$ (resp. $S5I_n$) is a sound and complete axiomatization (for the extended language) for such systems.*

We remark that in Proposition 4.2 and Theorem 4.3 we assumed that the environment determines the initial states. If we make the stronger assumption that the environment *uniquely determines* the initial state, then a smaller set of knowledge states is attainable, and again knowledge has extra properties. This is discussed in detail in the full paper.

Finally, we turn our attention to systems where the initial state uniquely determines the environment. Recall that this assumption is appropriate for distributed systems applications where the environment is just the initial setting of certain local variables in each process. If knowledge is not cumulative, then it again can be shown that all states of knowledge can be attained. But if knowledge is cumulative, then not only is it initially the case that the processes' state uniquely determines the environment, but this is true at all times.

Definition 4.5. (S,s) is a state of knowledge where *the processes' state uniquely determines the environment* if, whenever two global states s and s' in S are process equivalent, then s and s' have the same environment component.

Proposition 4.6. *If M is a system where knowledge is cumulative and the initial state uniquely determines the environment, then in every state of knowledge attainable in M, the processes' state uniquely determines the environment. Moreover, every state of knowledge where the processes' state uniquely determines the environment is attainable in a system where knowledge is cumulative and the initial state uniquely determines the environment.*

We can show that if the processes' state always uniquely determines the environment, then $S5I_n$ is not complete. The fact that the processes' state uniquely determines the environment can be characterized by the following axiom:

A7. $\varphi \Rightarrow I\varphi$.

Note that this is an axiom in the extended language. Somewhat surprisingly (and in contrast to the situation in Theorem 4.3), it turns out that if we restrict our attention to formulas involving only knowledge, then $S5_n$

is a complete axiomatization. No new axioms are required! Thus we have:

Theorem 4.7. *$S5_n$ is a sound and complete axiomatization for systems of n processes whose knowledge is cumulative where the initial state uniquely determines the environment. In the extended language, $S5I_n$ together with A7 forms a sound and complete axiomatization for such systems.*

This theorem shows that there are cases where the language may not be sufficiently powerful to capture the fact that not all states of knowledge are attainable.

Details of the proofs of theorems stated above and further results along these lines can be found in the full paper.

5. Conclusions

We have presented a general model for the knowledge of components of a system and shown how the properties of knowledge may depend on the subtle interaction of the parameters of the system. Although we have isolated a few parameters of interest here, we clearly have not made an exhaustive study of the possibilities. Rather, we see our contributions here as (1) showing that the standard S5 possible-worlds model for knowledge may not always be appropriate, even for the external notion of knowledge which does satisfy the S5 axioms, (2) providing a general model in which these issues may be examined, (3) isolating a few crucial parameters and formulating them precisely in our model, and (4) providing complete axiomatizations of knowledge for a number of cases of interest (complete axiomatizations are provided for many choices of parameters in the full paper).

We intend to push this work further by seeing what happens when we add common knowledge and time to the language. By results of [HM1] (since reproved and generalized in [CM,FI]), we know that for many choices of parameters, common knowledge will not be attainable in a system. Thus, we expect that even in cases where the axioms of S5 are complete, when we add common knowledge to the language we will need extra axioms beyond the standard S5 axioms for common knowledge (see [Le,HM2] for a discussion of the S5 axioms of common knowledge). We expect to find yet other complexities if we allow the language to talk explicitly about time by adding temporal modalities (as is done in [Le,RK,HV]). We can then explicitly axiomatize cumulative knowldge, although results of [HV] imply that it may often be impossible to get complete axiomatizations in some cases.

6. References

[BL] R. Brachman and H. Levesque, *Readings in Knowledge Representation*, Morgan Kaufmann, 1985.

[CM] M. Chandy and J. Misra, How processes learn, *Proc. 4th ACM Symposium on Principles of Distributed Computing*, 1985, pp. 204-214.

[DM] C. Dwork and Y. Moses, Knowledge and common knowledge in a Byzantine environment I: crash failures, *Theoretical Aspects of Reasoning about Knowledge* (ed. J.Y. Halpern), Morgan Kaufmann, 1986, pp. 149-170.

[FHV] R. Fagin, J.Y. Halpern, and M.Y. Vardi, What can machines know? On the properties of knowledge in distributed systems, to appear as an IBM Research Report, 1986.

[FV] R. Fagin and M.Y. Vardi, Knowledge and implicit knowledge in a distributed environment, *Theoretical Aspects of Reasoning about Knowledge* (ed. J.Y. Halpern), Morgan Kaufmann, 1986, pp. 187-206.

[FI] M.J. Fischer and N. Immerman, Foundations of knowledge for distributed systems, *Theoretical Aspects of Reasoning about Knowledge* (ed. J.Y. Halpern), Morgan Kaufmann, 1986, pp. 171-186.

[HF] J.Y. Halpern and R. Fagin, A formal model of knowledge, action, and communication in distributed systems: preliminary report, *Proc. 4th ACM Symp. on Principles of Distributed Computation*, 1985, pp. 224-236.

[HM1] J.Y. Halpern and Y.O. Moses, Knowledge and common knowledge in a distributed environment, *Proc. 3rd ACM Symp. on Principles of Distributed Computing*, 1984, pp. 50-61. Revised version appears as IBM Research Report RJ 4421, 1986.

[HM2] J.Y. Halpern and Y.O. Moses, A guide to the modal logics of knowledge and belief, *Proc. 9th International Joint Conference on Artificial Intelligence (IJCAI-85)*, 1985, pp. 480-490.

[HV] J.Y. Halpern and M.Y. Vardi, The complexity of reasoning about knowledge and time: Extended abstract, *Proc. 18th ACM Symposium on the Theory of Computing*, Berkeley, May 1986, pp. 304-315.

[Hi] J. Hintikka, *Knowledge and belief.* Cornell University Press, 1962.

[Kr] S. Kripke, Semantical analysis of modal logic, *Zeitschrift fur Mathematische Logik und Grundlagen der Mathematik* 9 (1963), pp. 67-96.

[Le] D. Lehmann, Knowledge, common knowledge, and related puzzles, *Proc. 3rd ACM Symp. on Principles of Distributed Computing*, 1984, pp. 467-480.

[Mo] R.C. Moore, Reasoning about knowledge and action, Technical Note 191, Artificial Intelligence Center, SRI International, 1980.

[PR] R. Parikh and R. Ramanujam, Distributed processing and the logic of knowledge, *Proc. Workshop on Logics of Programs*, Brooklyn, June 1985, Springer-Verlag, Lecture Notes in Computer Science - Vol. 193, pp. 256-268.

[Ro] S.J. Rosenschein, Formal theories of knowledge in AI and robotics, *New Generation Computing* 3, 1985, pp. 345-357.

[RK] S.J. Rosenschein and L.P. Kaelbling, The synthesis of digital machines with provable epistemic properties, *Theoretical Aspects of Reasoning about Knowledge* (ed. J.Y. Halpern), Morgan Kaufmann, 1986, pp. 83-98.