What Can Machines Know? On the Properties of Knowledge in Distributed Systems*

Ronald Fagin Joseph Y. Halpern
Moshe Y. Vardi

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

October 2, 1996

Abstract

It has been argued that knowledge is a useful tool for designing and analyzing complex systems. 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 and about the language used for describing knowledge. We present a formal model in which we can capture various assumptions frequently made about systems, such as whether they are deterministic or nondeterministic, whether knowledge is cumulative (which means that processes never "forget"), and whether or not the "environment" affects the state transitions of the processes. We then show that under some assumptions about the system and the language, 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.

*This version of the paper is almost identical to one that appears in *Journal of the ACM* **39**:2, 1992, pp. 328–376. A preliminary version appears in *Proc. AAAI-86* under the title "What can machines know? On the epistemic properties of machines".

1 Introduction

A fundamental problem in many branches of AI and computer science (including distributed systems, robotics, and planning) 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 the study of protocols in distributed systems, it has long been common to argue informally about knowledge, with statements such as "processor 1 cannot safely commit at this point, since it does not yet *know* whether processor 2 *knows* that processor 3 is still up". Although notions of knowledge have frequently been used in informal descriptions, traditional formal analyses of distributed protocols avoid any explicit mention of knowledge. Recent work has shown that these informal arguments can be completely formalized (see [HM1, DM, MT, RK] for some detailed examples).

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]. This second approach describes the way knowledge is ascribed to processors in a distributed system, in the papers mentioned above.

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

This notion of knowledge is *external*. It need not involve any cognitive activity on the part of the process or component. It is a notion meant to be used by the system designer in reasoning about the system.

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 it satisfies all the axioms of the classical modal logic S5 (we discuss this in more detail in Section 5; an overview of S5 and related logics of knowledge and belief can be found in [HM2]). Indeed, the abstract models most frequently used to capture this notion (for example, in [Ro]) have been variants of 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 state of an S5 Kripke structure really correspond to some "knowledge situation" that the system can be in? As we shall show, the answer to this question depends crucially, and in surprising ways, on the assumptions made about the system and about the language used for describing knowledge.

In order to explain our results, it is helpful to briefly review some material from [FV2], which directly inspired our work here. In [FV2], a very restricted type of system is considered, which intuitively can be viewed as one where robots observe their external environment and then communicate about their observations. These robots are assumed never to forget information they have learned. 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 at a given time (for example, if all we care about is whether or not it rained in San Francisco on January 1, we could take p to be "It rained in San Francisco on January 1"). Suppose also that we have a system with exactly two robots, which we will call 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 that p is true, or p is false and Bob doesn't know that p is false. Alice's state of knowledge can be captured by the formula

$$\neg K_{Alice} p \ \wedge \ \neg K_{Alice} \neg p \ \wedge \ K_{Alice}((p \wedge K_{Bob} p) \vee (\neg p \wedge \neg K_{Bob} \neg p)). \qquad (1)$$

Although this formula is perfectly consistent with S5, it is not attainable under the assumptions of [FV2].

To see that it is not attainable, it is convenient to consider the notion of *distributed knowledge*[1] introduced in [HM1] and studied in [CM, DM, FV2, MT, PR, RK]). Roughly speaking, a group has distributed knowledge of a fact if by putting all their information together, the members of the group could deduce that fact. In the class of systems considered in [FV2], distributed knowledge obeys a certain *conservation principle*—it can neither be lost not gained. Suppose now that the above state of knowledge 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 Alice could then receive a message from Bob saying "I (Bob) don't know p". Then, since Alice knows that either (a) p is true and Bob knows that p is true, or (b) p is false and Bob does not know that p is false, it follows that Alice would know that p must be false. But originally Alice and Bob did not have distributed knowledge that p is false. It follow that it is impossible for Alice and Bob to discover that p is false, since that would contradict the conservation principle for distributed knowledge. So p must be true. And since this argument holds for all states

[1]In [HM1], a previous version of this paper [FHV], and other papers, what we are now calling distributed knowledge was called *implicit knowledge*. We have changed the name here to avoid conflict with the usage of the term "implicit knowledge" in papers such as [FH, Lev].

where Alice has this same information, Alice knows p. But this contradicts the assumption that Alice doesn't know p.

In [FV2] a formal proof is given that the above state of knowledge is unattainable for the class of systems considered. In order to show the subtlety of the assumptions required to make the proof go through, we now give four situations where the state of knowledge *is* attainable. For the first case, suppose p is now the statement "the communication line between Alice and Bob 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 that p is true, or else p is false and Bob doesn't know that 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 have exactly the state of knowledge previously shown to be unattainable!

Actually, there is nothing wrong with either the proof of impossibility in [FV2] or with the counterexample just given. The proof of impossibility breaks down in the counterexample because of Alice's seemingly 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 counterexample, then she could never receive such a message because the communication line would be down. Implicitly, there is an assumption in the model of [FV2] that the primitive propositions are determined by some external environment and that this environment has no impact on the possible transitions of the system. In particular, a primitive proposition cannot say that a communication line is down or a message buffer is full.

Now suppose we slightly modify the counterexample 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 receives. 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 impossibility that we gave above break down? This time it is in the claim that Alice could not discover that p is false if Alice and Bob do not have distributed knowledge that p is false beforehand, which we inferred from the conservation principle for distributed knowledge. Although Alice and Bob do not have distributed knowledge as to whether p is true or false before any messages are sent, if p is true then they can obtain distributed knowledge that p is true by communication, and if p is false then they can obtain distributed knowledge that p is false by communication. Thus, if p is true (and so the communication link from Alice to Bob is up), then Alice can send Bob a message, and upon receipt of the message Bob will know that p is true; hence, distributed knowledge that p is true is gained. Similarly, if p is false, and if Alice attempts to send Bob a message, and then Bob sends Alice a message saying that he does not know that p is true (and so, in particular, Bob did not receive Alice's message), then Alice knows that p is false, and so distributed knowledge that p is false is gained. Why does the conservation principle hold for the systems of [FV2] but not in this counterexample? In the systems of [FV2], the primitive propositions are determined by some external environment and this environment has no impact on the transitions of the system. This assumption, which does not hold in this counterexample, is sufficient to prove that distributed knowledge of propositional facts does not increase with time.

The other half of the conservation principle, which says that distributed knowledge of propositional facts does not *decrease* with time, follows from another critical assumption from [FV2], namely, that neither robot forgets (i.e., their knowledge is cumulative). This may be reasonable when we are considering a scenario that lasts only a few rounds. However, no forgetting requires an unbounded number of states in general; thus, this is certainly not an assumption to take lightly. Once we drop this assumption, we can construct yet a third counterexample where the "unattainable" state of knowledge is attained.

In the proof of impossibility it is implicitly assumed that if at some point neither Alice nor Bob knows that p is false, then they never had any knowledge about 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, let p be the statement "Bob has a perfect memory". In particular, Alice knows that if Bob knows p, then p is true,[2] so Bob has a perfect memory, and so Bob never forgets that he knows p. In addition, Alice knows that if Bob knows $\neg p$, then p is false, and so Bob does not have a perfect memory, and so Bob might forget that he knows $\neg p$. Suppose in fact that p is true and Bob knows this, and Bob sends Alice two messages. The first one says "either I know p or I know $\neg p$" (i.e., $K_{Bob}p \vee K_{Bob}\neg p$), and the second says "I don't know $\neg p$" (i.e., $\neg K_{Bob}\neg p$). At this point, Alice knows that either p is true and Bob knows that p is true, or that p is false and Bob doesn't know that p is false (Bob knew that p was false and then forgot). Again, we have shown that the "unattainable" state of knowledge is attainable!

Our final counterexample shows how properties of the language can affect the properties of knowledge. In particular, this counterexample shows that our assumption that the primitive proposition p characterizes the environment at a given time was another crucial assumption in our proof of impossibility. Up to now, we have made this assumption. Thus, in our Alice and Bob example, we assumed that the environment is characterized by whether or not it rains in San Francisco, and we let p be "It rained in San Francisco on January 1". For our final counterexample, we now assume instead that the environment is characterized both by whether it is rainy or dry and whether the temperature is cold

[2] A basic property of *knowledge* (as opposed to *belief*) is that everything that is *known* is necessarily true.

or warm. We thus assume that there are *two* primitive propositions p and q, where p is "It rained in San Francisco on January 1", and q is "It was cold in San Francisco on January 1". Assume that Alice knows that either (a) it was rainy and cold (that is, p and q are both true), or else (b) it was dry and warm (that is, p and q are both false). Assume that Bob knows that it was rainy and cold (that is, he knows that p and q are both true). Assume that Bob tells Alice that either (a′) he knows that it was rainy and cold (which is the actual situation), or else (b′) he knows that it was warm but doesn't know whether it was rainy (that is, he knows that q is false but doesn't know whether or not p is true). After Alice receives this information from Bob, she still doesn't know whether it was rainy or dry. She knows that if it was rainy, then case (a) occurred, so it was cold, so case (b′) is impossible, so case (a′) occurred, so Bob knows that it was rainy. Thus, Alice knows that if p is true (it was rainy), then Bob knows that p is true. Further, she knows that if it was dry, then case (b) occurred, so it was warm, so case (a′) is impossible, so case (b′) occurred, so Bob doesn't know whether it was rainy. Thus, Alice knows that if p is false (it was dry), then Bob doesn't know that p is false. So, yet again, we have attained the "unattainable" state of knowledge!

What goes wrong with the proof of impossibility this time? After receiving Bob's message above that (a′) or (b′) holds, Alice knows that if p is false, then they have distributed knowledge that p is false. This is because Alice knows that if p is false, then she still of course knows that either p and q are both true or both false, and Bob knows that q is false. Therefore, it would be possible to infer by combining their information that p is false, and so it would be distributed knowledge. But the impossibility proof depends in a crucial way on the fact that Alice and Bob do not have distributed knowledge about p to start with.

The above discussion should convince the reader that the properties of knowledge depend in a subtle way on assumptions made about a system and the language. The main contribution of this paper is to isolate precisely a number of parameters of interest when analyzing systems, to show how these parameters can be described formally, and to discuss the effects of various combinations of these parameters on the properties of knowledge in the system. Some choices of parameters are more common in AI applications than distributed systems applications, and vice versa. Typical parameters include:

- Are the initial states of the processes a (possibly nondeterministic) function of the environment? This would be the case if the system consists of sensors observing nature, and the "environment" is a description of nature.

- Is the knowledge of the processes cumulative? Many 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. [CM, HM1, HF, Le, 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 [FI, RK]). But note that for limited interactions, knowledge can be cumulative even with bounded memory. We remark that Halpern and Vardi have shown that the assumption that knowledge is cumulative has a drastic effect on the complexity of the decision procedure for validity of formulas involving knowledge and time [HV1, HV2].

- 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.

- Is the system synchronous? That is, is there a “global clock” that every process can “see”, so that every process “knows the time”?

- Are process state transitions independent of the environment? In the case of processes or sensors observing an external environment and then communicating about it, the possible state transitions of the processes usually depend only on the local states of the processes, and not on the state of the environment. On the other hand, if the “environment” includes a description of the weather, then the presence of a heavy thunderstorm could affect communication, and so affect the transitions in the system. Thus, in this latter case, the possible state transitions of the processes would depend also on the state of the environment. As another example, if the environment describes the status of various message buffers in a distributed system, then process state transitions would *not* be independent of the environment.

- What do the primitive propositions talk about? In our Alice and Bob example, the primitive proposition p that Alice and Bob communicate about describes what “nature” was like at some fixed time, namely whether it rained in San Francisco on January 1. Similarly, in a distributed systems application, the primitive propositions might describe the initial values of certain registers. In both of these cases, we might say that “the primitive propositions are determined by the initial global state”. In our Alice and Bob example, the primitive proposition in fact are determined by the initial environment, and we indeed assume that it *characterizes* the initial environment.

The list of parameters mentioned above is not meant to be exhaustive. In fact, we shall discuss a number of other parameters of interest, and there are still other interesting parameters that we do not discuss. The interesting point for us is the subtle interplay between such parameters and the states of knowledge that are attainable. For example, if (1) the initial states of the processes are a (possibly nondeterministic) function of the environment, (2) knowledge is cumulative, (3) process state transitions are independent of the environment, and (4) the primitive propositions characterize the initial environment, then it turns out that the states of knowledge that are attainable are completely characterized by the axiom system ML^- of [FV2], which has strictly more axioms than S5. That is, if we consider only systems that satisfy these four properties, then ML^- is a sound and complete axiomatization. If we assume that the initial states of the processes are a *deterministic* function of the environment, rather than just a nondeterministic function of the environment, then yet another axiom is required. On the other hand, for systems that satisfy only three of these four conditions, every state of knowledge is attainable (that is, S5 then provides a complete characterization of the attainable states of knowledge). 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 language for discussing knowledge (that is, what the primitive propositions "talk about"). In this paper, we do such an analysis.

The rest of the paper is organized as follows. In Section 2 we describe our abstract model and show how various assumptions we would like to consider can be captured in the model. In Section 3 we introduce a language of primitive propositions, and consider various assumptions about what the primitive propositions talk about. In Section 4 we define our semantics, and introduce the notion of a "knowledge situation". In Section 5 we characterize what knowledge situations are attainable under a number of different reasonable assumptions about systems and about what the primitive propositions talk about. In Section 6, we give our conclusions. In the appendix, we prove the results stated (and those alluded to) in the body of the paper.

2 The model

In modeling a system, the first task is to decide which features are relevant for the application at hand. If we wanted, we could call every component of interest a "process". However, it is often natural instead to select certain components of the system and call them "processes", and to lump everything else together into an "environment" component. For example, in a distributed system of communicating processors, it might be natural to consider the processors as the processes of the system, and to keep track of other salient features by considering them as part of an environment component. Such features would include which messages have been sent but not delivered, and information as to whether the communication links are up or down. There are other times when we might wish, for example, to consider the message buffer as one of the processes, where the state of the message buffer would reflect which messages have not yet been delivered.

Thus, we consider systems with n processes (or components), along with an environment, which represents other things that are relevant. Essentially, the environment can be viewed as just another process, but one that we can often ignore, since we are not usually interested in what the environment knows. We assume that at any given time, each process is in some "local state", and the environment is also in some state. We think of the "global state" of the system at a given time as a description of the local states of the processes and the state of the environment. Thus, the global state describes the state, at a given time, of what we are focusing on. For example, in a message-based distributed system, the local state of a process might reflect all of the messages that have been sent or received by that process, and the state of the environment might reflect which messages are in message buffers. In our Alice and Bob example, the "environment" is a description of an external "nature" that is being observed by the processes.

We consider the system to be evolving over time. A description of one particular way that the system could have evolved is a run. Thus, for each system, we assume that there 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. We assume also that there is a set L of *local states* that the processes can take on, and a set E of *environment states*. The set G of *global states* consists of tuples $(e, l_1, ..., l_n)$, with $e \in E$ and $l_i \in L$ for $i = 1, ..., n$. A global state consists of those things that we choose the focus on; it need not represent all aspects of the system. For example, in a model of a distributed system we often choose to ignore several aspects of the underlying hardware. A *run* is a function $r : \mathbb{N} \rightarrow G$ which associates with each time m a global state $r(m)$.[3] We may refer to $r(0)$ as the *initial global state* of the run r, or simply the *initial state*. Following [HM1], we may refer to a pair (r, m) consisting of a run r and a time m as a *point*. Points, rather than global states, are our "possible worlds". Thus, we shall define what it means for a formula to be true at a *point*, rather than what it means for a formula to be true at a global state. Finally, a *system* is a set $\mathcal{R}$ of runs.

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 being received or sent, transitions might also be a result of physical interactions or events (perhaps random!) internal to some process.

If $\mathcal{R}$ is a system (a set of runs), then we say that $s = (e, l_1, ..., l_n)$ is a *global state of* $\mathcal{R}$ if $s = r(m)$ for some run $r \in \mathcal{R}$ and some time m; we call e the *environment component* of s and $(l_1, ..., l_n)$ the *process component*. Let $s = (e, l_1, ..., l_n)$ and $s' = (e', l'_1, ..., l'_n)$ be two global states in $\mathcal{R}$. We say s and s' are *indistinguishable to process* i, written $s \sim_i s'$, if $l_i = l'_i$. If (r, m) and (r', m') are points, then we say that (r, m) and (r', m') are indistinguishable to process i, written $(r, m) \sim_i (r', m')$, if $r(m) \sim_i r'(m')$. If s and

[3]We chose to model time by the set $\mathbb{N}$ of the natural numbers. This assumption is not crucial; we could just as well have chosen to model time by the set $\mathbb{R}$ of the real numbers. We do, however, want to make the assumption that in each run, only a finite number of events (which correspond to changes in the global state) happen up to any given time. This assumption is automatically true when time is modelled by $\mathbb{N}$.

s' are two global states (or two points), and if $s \sim_i s'$ for every process i, then we say s and s' are *process equivalent*, written $s \sim s'$. Thus, two global states s and s' are process equivalent precisely if they have the same process component. *Process i's history in run r up to time m* is the sequence $\langle s_1, ..., s_k \rangle$ 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 $\langle l, l, l', l, l \rangle$ of states, its history in run r up to time 4 is just $\langle l, l', l \rangle$.

The main thrust of this paper is to consider natural conditions that certain systems satisfy, to show how they can be formalized in our model, and to consider the effect that some of these conditions (along with conditions as to what the primitive propositions talk about) have on the properties of knowledge. We begin by giving some possible conditions on the set of global states that can serve as initial states.

If $\mathcal{R}$ is a system, then the *set of initial states in* $\mathcal{R}$ is the set of global states $r(0)$ for $r \in \mathcal{R}$. One natural condition on the set of initial states of $\mathcal{R}$ might be that *the initial local states of the processes and the initial state of the environment are all independent of each other.* This means that there is a set E of environment states and a set $INIT_i$ of local states for each process i such that the set of initial states of R is $\{(e, l_1, ..., l_n) | \, e \in E$ and $l_i \in INIT_i$ for each $i\}$. An interesting special case occurs when *there is a unique initial global state* (which means that E and each $INIT_i$ are singleton sets).

A slightly less restrictive condition on the set of initial global states arises in many applications, such as our Alice and Bob example, where the processes' initial states are a function of observations made of "nature". We can assume that the state of nature is modelled as part of the environment. Thus, in this example, we can take the processes' initial state to be some function of the initial state of the environment. 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* in $\mathcal{R}$ if there is a set E of environment states and a set $INIT_{i,e}$ of local states for each $e \in E$ and each process i, such that the set of initial states of $\mathcal{R}$ is $\{(e, l_1, ..., l_n) | \, e \in E$ and $l_i \in INIT_{i,e}$ for each $i\}$.

Intuitively, $INIT_{i,e}$ 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 observing an external environment e, then the states in $INIT_{i,e}$ 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 would be in four possible distinct local states, depending on which of p and q it observes. Thus, the four possible local states correspond to the situations where (a) it observes that both p and q are true, (b) it observes that p is true but does not observe that q is true, (c) it observes that q is true but does not observe that p is true, and (d) it does not observe that p is true and does not observe that q is true. Note that our definition assumes that there is initially no interaction between the observations of different processes. That is, if in a given 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 interesting special case where the environment determines the initial states occurs when the initial state of the processes is a *deterministic* function of the environment. In our example above of sensors placed in some external environment, this corresponds to the case where the sensors operate in a completely deterministic fashion. We say that *the environment uniquely determines the initial state* if there are no two distinct initial states with the same environment component.

We now describe some natural conditions on state transitions.

The knowledge of the processes is said to be *cumulative* if the process "remembers" the entire history of the various local states it has been in during the course of a run. Thus, if two points (r, m) and (r', m') are indistinguishable to process i, and if the knowledge of the processes is cumulative, then it must be the case that process i had the same history in run r up to time m as it did in run r' up to time m'. It is easy to see that if the knowledge of the processes is cumulative, this limits the combinations of state transitions that can occur. For example, it would not be possible for process i to be able to enter local state s immediately after being in local state s' and also immediately after being in local state $s'' \neq s'$, since this would mean that in state s process i has "forgotten" what its previous state was. Formally, we say that *knowledge is cumulative* if for all processes i, all runs r, r', and all times m, m', if $r(m) \sim_i r'(m')$, then process i's history in run r up to time m is identical to its history in run r' up to time m'. That is, knowledge is cumulative if each process always knows its history. Note that cumulative knowledge requires an unbounded number of local states in general, since for each possible history there must be a state that encodes it. In particular, the knowledge of finite-state machines will not in general be cumulative.

The notion of cumulative knowledge that we just discussed corresponds to "no forgetting". The dual notion is "no learning". Let us say that process *i considers run r' possible* at point (r, m) if there is m' such that $r(m) \sim_i r'(m')$; otherwise, we say that process i *considers run r' impossible* at (r, m). Now "no forgetting" implies that if at some point (r, m) process i considers run r' impossible, then at every point (r, m') with $m' \geq m$ process i considers run r' impossible. Similarly, "no learning" implies that if at some point (r, m) process i considers run r' possible, then at every point (r, m') with $m' \geq m$ process i considers run r' possible. It is not hard to verify that knowledge is cumulative in a system $\mathcal{R}$ iff for all runs $r, r' \in \mathcal{R}$ and times m, m', k, if $(r, m) \sim_i (r', m')$ and $k \leq m$, then for some time k' with $k' \leq m'$ we have $(r, k) \sim_i (r', k')$. Dually, we now say that *processes do not learn* if for all runs $r, r' \in \mathcal{R}$ and times m, m', k, if $(r, m) \sim_i (r', m')$ and $k \geq m$, then for some time k' with $k' \geq m'$ we have $(r, k) \sim_i (r', k')$. Thus, if no forgetting means that the set of possible runs does not increase with time, no learning means that the set of possible runs does not decrease with time. This notion was introduced by Ladner and Reif [LR], who viewed it as a situation where each player's strategy is preordained and cannot be changed over time (and thus is *nonadaptive*), and studied further by Halpern and Vardi [HV1, HV2].

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 that a system is *synchronous* if for all processes i and runs r, r', if $r(m) \sim_i r'(m')$, then $m = m'$. An easy proof shows that in a synchronous system where knowledge is cumulative, if $r(m) \sim_i r'(m)$ and $0 \leq k < m$, then $r(k) \sim_i r'(k)$.

In general, the set of global states that a system could be in at time $m + 1$ can depend on the whole history up to time m, and not just on the global state at time m. However, there are many situations where the current global state contains enough information that "what can happen next" is completely determined by the current global state. Formally, we say that a system is *history independent* if whenever $r, r' \in \mathcal{R}$ are two runs such that $r(m) = r'(m')$, then there is a run $r'' \in \mathcal{R}$ which has the same prefix as r up to time m and continues as r' (i.e., $r''(k) = r(k)$ if $k \leq m$, and $r''(k) = r'(m' + k - m)$ if $k \geq m$).[4] Intuitively, if global states contain all the relevant information, and if up to time m the situation is as in run r, then it could have been the case that from time m on we could have continued as in run r'. If a system is history independent, then there are "no hidden components". An example where this would *not* be the case would be a distributed system where we choose not to represent the message buffers of the system by any of the components of the processes or by the environment component. Thus, in such a system, it could be possible that there is a run r where no messages are ever sent and a run r' where a message is sent at time 3 and received at time 5, and such that r and r' are in the same global state at time 4 (the process that sent the message at time 3 in run r' "has forgotten" by time 4 that it ever sent a message). Although $r(4) = r'(4)$, there is no run r'' with the same prefix as r up to time 4 and which continues as r', since then in r'' a message would be received that was never sent. Intuitively, the environment component does *not* in general represent "everything else" (that is, everything other than the local states of the processes), but rather "everything else that we choose to focus on". When the environment component does not represent "everything else", we should not expect the system to be history independent.

We now consider another condition that is closely related to history independence. We say that *process state transitions are independent of the environment* if, whenever $r, r' \in \mathcal{R}$ are two runs such that $r(m) \sim r'(m')$, then there is a run $r'' \in \mathcal{R}$ which, when we ignore the environment component, has the same prefix as r up to time m and continues as r' (i.e., $r''(k) \sim r(k)$ if $k \leq m$, and $r''(k) \sim r'(m' + k - m)$ if $k \geq m$). Intuitively, this condition says that process state transitions depend only on the current states of the processes.

There are situations where a slightly weaker condition holds. In a distributed systems application where the environment represents the state of message buffers, process state transitions are *not* independent of the environment. However, if the message buffers

[4]Such a run r'' is said to be in the *fusion closure* of r and r' [Pr, Em].

are initially empty, then it would be natural to assume that process state transitions are independent of the *initial* environment. This means that whenever we have two process-equivalent initial states, then the same sequence of transitions of states of the processes is possible from each of them. This new condition is obtained from our previous condition by letting $m = 0$. Thus, *process state transitions are independent of the initial environment* if whenever s and s' are initial states where $s \sim s'$, and r is a run with initial state s (i.e., $r(0) = s$), then there is a run r' with initial state s' such that $r(m) \sim r'(m)$ for all times m (this run r' corresponds to the run r'' in the previous definition). In our Alice and Bob example where the environment is "nature" that is being observed by Alice and Bob, process state transitions are independent of the initial environment. However, in our modification of the example where the environment could describe whether the communication line between Alice and Bob is up, it is *not* the case that process state transitions are independent of the initial environment.

We say that a system is *deterministic* if the initial state completely determines the run; i.e., the system is deterministic if whenever r and r' are runs with $r(0) = r'(0)$, then $r = r'$. A natural way to strengthen this condition would be to say that the "next" global state is uniquely determined. Formally, let us say that a system is *strongly deterministic* if for all runs r, r' and times m, m', if $r(m) = r'(m')$, then $r(m+1) = r'(m'+1)$. We leave to the reader the verification that a system is strongly deterministic if and only if it is deterministic and history independent. We note also that in a deterministic system where process state transitions are independent of the initial environment, the initial process component completely determines the process components in the run at every time; i.e., if r and r' are runs where $r(0) \sim r'(0)$, then $r(m) \sim r'(m)$ for every time m.

We have outlined a few natural conditions 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 conditions on systems can be easily captured within our model. In the next section we shall discuss some natural conditions on what the primitive propositions talk about.

3 Introducing a language

In order to reason about a system, we select certain basic facts of interest and consider how each process's knowledge of these facts changes over time. In our Alice and Bob example, the fact of interest was whether or not it was raining on January 1; as Alice and Bob communicate, their knowledge about this fact, along with their knowledge about each other's knowledge, etc., changes as a result of the messages they send and receive.

We can represent these basic facts of interest by primitive propositions. We assume that the number of basic facts of interest is finite, so that the set of primitive propositions is finite, say $\{p_1, ..., p_k\}$ (we will see later that this a crucial assumption). We define an *interpreted system* to be a pair $(\mathcal{R}, \pi)$, where $\mathcal{R}$ is a system (a set of runs), and where π is a function which maps each point (r, m) onto a truth assignment $\pi(r, m)$ on the set $\{p_1, ..., p_k\}$. Thus, $\pi(r,m)(p_j) \in \{\text{true}, \text{false}\}$ for each j, so that $\pi(r,m)(p_j)$ tells us whether the fact represented by p_j is true at the point (r,m).

Just as we considered certain natural conditions on systems, we can consider also other natural conditions on interpreted systems by making various assumptions about what the primitive propositions talk about.

In our definition of interpreted systems, π is a function on points, not on global states. Thus, π has as its argument the point (r,m) rather than the global state $r(m)$. We say that *the primitive propositions are determined by the current global state* if whenever (r,m) and (r',m') are points where $r(m) = r'(m')$, then $\pi(r,m) = \pi(r',m')$. Thus, in this case the truth of the facts of interest are determined completely by the current global state. A related but contrasting situation is where *the primitive propositions are determined by the initial global state*; i.e., whenever $r(0) = r'(0)$, then $\pi(r,m) = \pi(r',m')$ for all r, r', m, m'. An example of this latter condition is when the primitive propositions talk about whether some variables were initially zero.

Instead of just talking about the current (respectively, initial) global state, the primitive propositions might talk about just some portion of the global state. We say that *the primitive propositions are determined by the current states of the processes* if whenever $r(m) \sim r'(m')$, then $\pi(r,m) = \pi(r',m')$ for all r, r', m, m'. Similarly, we say that *the primitive propositions are determined by the initial state of the processes* if whenever $r(0) \sim r'(0)$, then $\pi(r,m) = \pi(r',m')$ for all r, r', m, m'. As an interesting special case, we say that *each primitive proposition is determined by the current (*respectively*, initial) state of some process* if for each primitive proposition p there is a process i such that whenever $r(m) \sim_i r'(m')$ (respectively, $r(0) \sim_i r'(0)$), then $(\pi(r,m))(p) = (\pi(r',m'))(p)$. A situation where each primitive proposition is determined by the current (respectively, initial) state of some process occurs when for each primitive proposition p there is a process i such that p talks about the current (respectively, initial) value of some variable local to process i.

It is straightforward to see that if the primitive propositions are determined by the initial states of the processes, and if knowledge is cumulative, then the primitive propositions are determined by the current states of the processes. Intuitively, this is because the current states of the processes uniquely determine the initial states of the processes when knowledge is cumulative. Similarly, if each primitive proposition is determined by the initial state of some process, and if knowledge is cumulative, then each primitive proposition is determined by the current state of some process.

Instead of the primitive propositions talking only about the processes, there are situations where the primitive propositions talk about the environment. Formally, we say that *the primitive propositions are determined by the current (*respectively*, initial) environment* if whenever $r(m)$ and $r'(m')$ (respectively, $r(0)$ and $r'(0)$) have the same environment component, then $\pi(r,m) = \pi(r',m')$. In our Alice and Bob example, the primitive propositions are determined by the initial environment.

In fact, in our Alice and Bob example, it is the case that the primitive propositions *characterize* the initial environment. We can modify all of our definitions we have just given about primitive propositions "being determined by" something into definitions where they "characterize" something. For example, we say that *the primitive propositions characterize the current (*respectively*, initial) environment* if $r(m)$ and $r'(m')$ (respectively, $r(0)$ and $r'(0)$) have the same environment component *if and only if* $\pi(r, m) = \pi(r', m')$.

In this section we have given certain natural conditions on the language. As in the previous section, our goal is not to be exhaustive but rather to demonstrate the point that many natural conditions can be captured easily within our framework.

4 Knowledge situations

We wish to reason not only about basic facts, but about each process's *knowledge* of basic facts, and a process's knowledge of another process's knowledge of the basic facts, etc. In order to do so, we first augment the language so that we can explicitly talk about the knowledge of processes. In this section, we give this semantics, which is based on the familiar Kripke semantics.

Once we have defined the syntax and semantics of the language, we can define the *state of knowledge* at a point in an interpreted system to be simply the set of formulas true at the point. This definition is syntactic in nature, and will depend on the particular language chosen. In this section, we define a semantic analogue to the state of knowledge at a point, which we call the *knowledge situation* at a point in an interpreted system. This will later allow us to describe a correspondence between the knowledge situations that are "attainable" in an interpreted system and the axioms for knowledge that hold in that interpreted system.

We start by giving a formal syntax. Let Φ be the finite set of primitive propositions. Let $\mathcal{L}_n(\Phi)$ (or simply $\mathcal{L}_n(\Phi)$, if Φ is understood) be the set of formulas that results when we take Φ and close under negation, conjunction, and the modal operators $K_1, ..., K_n$, so that if φ and φ' are formulas, then so are $\neg\varphi$, $\varphi \wedge \varphi'$, and $K_i\varphi$, for $i = 1, ..., n$. The formula $K_i\varphi$ is interpreted as "process i knows φ". Let $\mathcal{L}_n^D(\Phi)$ (or simply $\mathcal{L}_n^D(\Phi)$, if Φ is understood) be defined like $\mathcal{L}_n(\Phi)$, but where we also let $D\varphi$ be a formula when φ is. The formula $D\varphi$ represents *distributed knowledge* of φ. Intuitively, distributed knowledge (which is formally introduced in [HM1] and has also been used in [CM, DM, FV2, MT, PR, RK]) is the knowledge that can be obtained when the members of a group pool their knowledge. Put differently, it is what someone who had all the knowledge that each member in the group had could infer.

We shall soon give the semantics that tells us when a formula is true at a point in an interpreted system. Since our semantics is based on the standard Kripke semantics [Kr], we begin by reviewing the Kripke semantics for the truth of a formula at a state of a Kripke structure (where we follow Halpern and Moses [HM2] in the definition of the truth of a formula involving distributed knowledge). A *Kripke structure* is a tuple

$(S, \pi, \mathcal{K}_1, ..., \mathcal{K}_n)$, where S is a set of "states", where $\pi(s)$ is a truth assignment to the primitive proposition in Φ for each $s \in S$, and where each $\mathcal{K}_i$ is a binary relation on S. Intuitively, if s and s' are states with $(s, s') \in \mathcal{K}_i$, then in state s it is consistent with process i's information that state s' is possible. We are interested only in *S5 Kripke structures*, where each $\mathcal{K}_i$ is an equivalence relation on S (i.e., a reflexive, symmetric, and transitive binary relation on S). Such a Kripke structure satisfies the S5 properties of knowledge, which we shall discuss in the next section.

We now define when a formula φ of $\mathcal{L}_n^D(\Phi)$ is *satisfied* (or *holds*, or *is true*) at a state s of Kripke structure M, written $(M, s) \models \varphi$. Intuitively, the formula $K_i\varphi$ ("process i knows φ") is satisfied at a state s precisely if φ is satisfied at every state s' that process i "cannot distinguish" from state s. Also intuitively, the formula $D\varphi$ ("φ is distributed knowledge") is satisfied at a state s precisely if φ is satisfied at every state s' that no process can distinguish from state s.

$(M, s) \models p$, where p is a primitive proposition, if p is true under the truth assignment $\pi(s)$

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

$(M, s) \models \varphi_1 \wedge \varphi_2$ if $(M, s) \models \varphi_1$ and $(M, s) \models \varphi_2$

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

$(M, s) \models D\varphi$ if $(M, s') \models \varphi$ for all s' such that $(s, s') \in \cap_{i=1}^n \mathcal{K}_i$.

To define the truth of a formula at a point in an interpreted system, we associate each interpreted system $(\mathcal{R}, \pi)$ with a Kripke structure $Kr(\mathcal{R}, \pi) = (S, \pi, \mathcal{K}_1, ..., \mathcal{K}_n)$, where the states of $Kr(\mathcal{R}, \pi)$ are the points (r, m) of the interpreted system, and where for each i we have that $((r, m), (r', m')) \in \mathcal{K}_i$ iff $r(m) \sim_i r'(m')$. Thus, if $r(m) = (e, l_1, ..., l_n)$ and $r'(m') = (e', l'_1, ..., l'_n)$, then $((r, m), (r', m')) \in \mathcal{K}_i$ iff $l_i = l'_i$. It is easy to see that $Kr(\mathcal{R}, \pi)$ is an S5 Kripke structure.

We now say that a formula φ of $\mathcal{L}_n^D(\Phi)$ is *satisfied* (or *holds*, or *is true*) at a point (r, m) in an interpreted system $\mathcal{I} = (\mathcal{R}, \pi)$, written $(\mathcal{I}, r, m) \models \varphi$, if it is satisfied at the corresponding state of the associated Kripke structure $Kr(\mathcal{R}, \pi)$. In particular,

$(\mathcal{I}, r, m) \models K_i\varphi$ iff $(\mathcal{I}, r', m') \models \varphi$ for all r', m' such that $r(m) \sim_i r'(m')$

$(\mathcal{I}, r, m) \models D\varphi$ iff $(\mathcal{I}, r', m') \models \varphi$ for all r', m' such that $r(m) \sim r'(m)$.

As we did for Kripke semantics, we describe the intuition behind the definition of $\models$ for $K_i\varphi$ and $D\varphi$ hold. Let $V_i(r, m) = \{(r', m') | \, r(m) \sim_i r'(m')\}$. Intuitively, $(r', m') \in V_i(r, m)$ 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 $V_i(r, m)$. Let $V(r, m) = \bigcap_{i=1}^n V_i(r, m)$. Intuitively,

$(r', m') \in V(r, m)$ 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 $r(m) \sim r'(m')$ precisely if $(r', m') \in V(r, m)$. Thus, $D\varphi$ holds at time m in run r precisely if φ holds at every point in $V(r, m)$.

If $\mathcal{C}$ is a class of interpreted systems, then we say that a formula φ is *valid* (or *sound*) with respect to $\mathcal{C}$ if $(\mathcal{I}, r, m) \models \varphi$ for all interpreted systems $\mathcal{I} \in \mathcal{C}$, runs r of $\mathcal{I}$, and times m.

We are almost ready to define the semantic notion of the "knowledge situation" of a point in an interpreted system. We first need a few preliminaries.

It is sometimes convenient to consider a truth assignment α to the primitive propositions $p_1, \ldots, p_k$ as a formula in the languages $\mathcal{L}_n(\Phi)$ and $\mathcal{L}_n^D(\Phi)$, by taking it to be an abbreviation for the conjunction $p'_1 \wedge ... \wedge p'_k$, where p'_i is p_i (respectively, $\neg p_i$) if p_i is true (respectively, false) under the truth assignment α. We shall often find it convenient to refer to α, whether it is thought of as a truth assignment or as a formula, as a *primitive state*. Note that we could not identify a primitive state with a formula in our language if we did not take Φ, the set of primitive propositions, to be finite.

We now need to define the notion of an "interpreted state". Assume that there is some fixed set of primitive propositions and fixed set of local states. We then define an *interpreted state* to be a tuple $(l_1, ..., l_n, \alpha)$, where α is a primitive state and each l_i is a local state. We call α the *primitive state component* and $(l_1, ..., l_n)$ the *process component*. Note that there is no environment component. The reason we omit the environment component is that, while that component may play a role in the evolution of runs, it plays no role, as we shall see, in determining the truth or falsity of formulas. Let $s = (l_1, ..., l_n, \alpha)$ and $s' = (l'_1, ..., l'_n, \alpha')$ be interpreted states. By analogy to our definition for global states, we write $s \sim_i s'$ if $l_i = l'_i$. Again as before, we say that s and s' are *process equivalent*, written $s \sim s'$, if $s \sim_i s'$ for every process i, that is, if s and s' have the same process component. Given an interpreted system $(\mathcal{R}, \pi)$, there is a natural interpreted state associated with the point (r, m). Namely, if $r(m) = (e, l_1, ..., l_n)$ and $\pi(r, m) = \alpha$, then we define $\hat{r}(m)$, the *interpreted state at the point* (r, m), to be the tuple $(l_1, ..., l_n, \alpha)$. We then say that $\hat{r}(m)$ is an *interpreted state in* $(\mathcal{R}, \pi)$. We refer to $\hat{r}(0)$ as an *initial interpreted state*, or simply an *initial state* if it is clear that we are discussing interpreted states.

If S is a set of interpreted states and $s \in S$, then we can define what it means for the pair (S, s) to satisfy a formula φ, written $(S, s) \models \varphi$, by associating with S a Kripke structure $(S, \pi, \mathcal{K}_1, ..., \mathcal{K}_n)$. The set of states of the Kripke structure is S itself. We let $\pi(l_1, ..., l_n, \alpha)$ be α, and we let $\mathcal{K}_i$ be the set of all pairs (s, s') such that $s, s' \in S$ and $s \sim_i s'$. Once again, it is easy to see that this gives us an S5 Kripke structure. If φ is a formula in $\mathcal{L}_n^D(\Phi)$, then we define $(S, s) \models \varphi$ to hold if φ is satisfied at the corresponding state of the associated Kripke structure. In particular,

$(S, s) \models K_i\varphi$ iff $(S, s') \models \varphi$ for all $s' \in S$ such that $s \sim_i s'$

$(S, s) \models D\varphi$ iff $(S, s') \models \varphi$ for all $s' \in S$ such that $s \sim s'$.

If S is a set of interpreted states and if $s, s' \in S$, then we say that s' is *reachable in S from s* if there exist $s_1, ..., s_k$ in S and (not necessarily distinct) processes $i_1, ..., i_{k-1}$ such that $s = s_1$, $s' = s_k$, and $s_j \sim_{i_j} s_{j+1}$ for $j = 1, ..., k-1$. In other words, s' is reachable from s in S if the pair (s, s') is in the reflexive-transitive closure of $\bigcup_{i=1}^{n} \mathcal{K}_i$. A set S of interpreted states is *connected* if every member is reachable in S from every other member. By a similar definition, if s, s' are points in an interpreted system $\mathcal{I}$ or states in a Kripke structure, we define what it means for s' to be *reachable in $\mathcal{I}$ from s*; we similarly define connectedness in these cases.

We are now ready to define a "knowledge situation".

Definition 4.1: A *knowledge situation* is a pair (S, s) where S is a connected set of interpreted states with $s \in S$. If $\mathcal{I}$ is an interpreted system and (r, m) is a point in $\mathcal{I}$, then (S, s) is the *knowledge situation at* $(\mathcal{I}, r, m)$ (and (S, s) is *attainable in* $\mathcal{I}$) if $s = \hat{r}(m)$ and S is the set of all $\widehat{r'}(m')$ such that (r', m') is reachable in $\mathcal{I}$ from (r, m). ∎

In our definition, the knowledge situation of an interpreted system $\mathcal{I}$ at a point (r, m) depends only on the points reachable in $\mathcal{I}$ from (r, m). This reflects the fact that in our formal semantics, the truth of a formula at time m in run r depends only on points reachable in $\mathcal{I}$ from (r, m).

The following proposition, whose straightforward inductive proof is left to the reader, shows that our definition of the satisfaction of a formula in a knowledge situation (S, s) is equivalent to our definition of the satisfaction of the formula at a point in an interpreted system.

Proposition 4.2: *Let (S, s) be the knowledge situation at $(\mathcal{I}, r, m)$ and let φ be a formula. Then $(\mathcal{I}, r, m) \models \varphi$ if and only if $(S, s) \models \varphi$.*

In particular, Proposition 4.2 justifies our earlier statement that the environment component has no effect on the satisfiability of a formula at a point in an interpreted system.

If M is an S5 Kripke structure and s is a state of M, then we can similarly define the *knowledge situation at* (M, s). Not surprisingly, the analogue to Proposition 4.2 still holds.

5 The properties of knowledge

Our goal is to consider some combinations of the parameters we have discussed, and to analyze the resulting properties of knowledge. Thus, we consider some classes $\mathcal{C}$ of interpreted systems (where $\mathcal{C}$ consists of those interpreted systems that satisfy certain combinations of the conditions we have discussed) and analyze the properties of knowledge for the class $\mathcal{C}$. We shall not try to give here a complete taxonomy of the properties of knowledge for each choice of parameters that we have discussed. Instead, we discuss a few illustrative cases, with a view towards showing the subtlety of the interaction between the properties of the interpreted system and the properties of knowledge.

It is often convenient for us to refer to the extension of the classical axiom system S5 to a situation with n "knowers" as S5$_n$. It is well-known that the axiom system S5$_n$ captures the properties of knowledge in S5 Kripke structures with n knowers (which is, of course, why we call them S5 Kripke structures); i.e., S5$_n$ is *sound* (all the axioms and rules of inference are valid) and *complete* (all valid formulas are provable); cf. [HM2]. The axioms of S5$_n$ are (where $i = 1, ..., n$):

A1. All substitution instances of propositional tautologies

A2. $K_i\varphi_1 \wedge K_i(\varphi_1 \Rightarrow \varphi_2) \Rightarrow K_i\varphi_2$

A3. $K_i\varphi \Rightarrow \varphi$

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

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

There are two rules of inference:

R1. *Modus ponens*: from φ_1 and $\varphi_1 \Rightarrow \varphi_2$ infer φ_2.

R2. *Knowledge generalization*: from φ infer $K_i\varphi$.

In the sequel we will consider several extensions of S5$_n$, by additional axioms. All these extensions will use the inference rules R1 and R2.

If we allow also distributed knowledge (that is, if we consider the language $\mathcal{L}_n^D(\Phi)$ rather than $\mathcal{L}_n(\Phi)$), then we need additional axioms. Following Halpern and Moses [HM2], we define the axiom system S5D$_n$ by taking S5$_n$ and adding axioms that say that D acts like a knowledge operator (i.e., all the axioms above hold with K_i replaced by D) and the following additional axiom, which we name "KD" for later reference:

KD. $K_i\varphi \Rightarrow D\varphi$.

Furthermore, if $n = 1$, then we add the additional axiom

$D\varphi \Rightarrow K_1\varphi$.

(This axiom is needed to guarantee that knowledge and distributed knowledge coincide when there is single process.)

Halpern and Moses [HM2] state without proof that S5D$_n$ is sound and complete in S5 Kripke structures for the language $\mathcal{L}_n^D(\Phi)$. We give a proof in the appendix.

If we put no conditions on interpreted systems, then there is an exact correspondence between knowledge situations in our model and those in S5 Kripke structures. For each S5 Kripke structure M, it is straightforward to find an interpreted system $(\mathcal{R}, \pi)$ such that $Kr(\mathcal{R}, \pi) = M$. The following result then follows easily.

Proposition 5.1: *$S5_n$ (respectively, $S5D_n$) is a sound and complete axiomatization with respect to interpreted systems of n processes for the language $\mathcal{L}_n(\Phi)$ (respectively, $\mathcal{L}_n^D(\Phi)$).*

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 seems to assume that reasoners can do *negative introspection*, and know about their lack of knowledge. While these axioms may be controversial for some notions of knowledge, they clearly hold for the external, information-based notion that we are concerned with here. Thus, they are sound for interpreted systems. But of course, under some of the conditions that we discussed earlier on interpreted systems and on what the primitive propositions talk about, they may no longer be complete.

In our Alice and Bob example, what assumptions were really needed to show that the state of knowledge defined by formula (1) in the introduction is not attainable? As the counterexamples given in the introduction suggest, we need to assume that knowledge is cumulative (i.e., that there is "no forgetting"), that process state transitions are independent of the initial environment, and that the primitive propositions characterize the initial environment. Also implicit in the story is that Alice and Bob initially study nature independently and that this determines their initial states, so we also need the assumption that the environment determines the initial states. It turns out that these conditions are sufficient to show that formula (1) is not attainable, as we shall see below.

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

Definition 5.2: A set S of interpreted states satisfies the *pasting condition* if whenever

1. $s_1, ..., s_n, t' \in S$,
2. α is the primitive state component of s_i, for $i = 1, ..., n$, and
3. $t' \sim_i s_i$, for $i = 1, ..., n$,

then there exists $t \in S$ such that $t \sim t'$ and α is the primitive state component of t. The knowledge situation (S, s) is said to obey the pasting condition if S does. ∎

Thus, S satisfies the pasting condition precisely if whenever

$s_1 = (l_1, \cdot, ..., \cdot, \alpha) \in S$,

$s_2 = (\cdot, l_2, \cdot, ..., \cdot, \alpha) \in S$,

...

$s_n = (\cdot, ..., \cdot, l_n, \alpha) \in S$,

$t' = (l_1, ..., l_n, \cdot) \in S$,

then $t = (l_1, ..., l_n, \alpha) \in S$. (Each $\cdot$ represents a value we do not care about.) We have called this condition the "pasting condition", since it says that if certain interpreted states are in S, then another interpreted state which is the result of "pasting together" these interpreted states is also in S.

Similarly, we can modify the definition in the obvious way to define what it means for a Kripke structure to satisfy the pasting condition.

We shall show that the conditions we have discussed that hold for our Alice and Bob example (or other natural conditions) are sufficient to guarantee that every attainable knowledge situation satisfies the pasting condition.

Not surprisingly, the fact that the pasting condition holds affects the properties of knowledge. Neither $S5_n$ nor $S5D_n$ is complete. Consider the following axiom in the language $\mathcal{L}_n^D(\Phi)$, where α is a primitive state and $\{1, ..., n\}$ is the set of processes:

A6. $D\neg\alpha \Rightarrow K_1\neg\alpha \vee ... \vee K_n\neg\alpha$.

This new axiom says that if it is distributed knowledge that the primitive state is not α, then the stronger fact is true that some process knows that the primitive state is not α. Axiom A6 is *not* a consequence of $S5D_n$; however, as we shall see, it follows easily from the pasting condition. We remark that formula (1) discussed in the introduction is a consequence of $S5D_2$ + A6, where the two processes are Alice and Bob (provided we assume that the primitive proposition p in formula (1) is a primitive state; recall that we said that it characterizes the initial environment).

Even in the language $\mathcal{L}_n(\Phi)$, which does not have the distributed knowledge operator D in it, 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, $K_2p \vee (K_1\neg K_3p \wedge \neg K_2\neg p)$ is a pure knowledge formula, but $p \wedge \neg K_1p$ is not. Consider the following axiom, where α is a primitive state and β is a pure knowledge formula:

A6′. $\beta \wedge K_i(\beta \Rightarrow \neg\alpha) \Rightarrow (K_1\neg\alpha \vee ... \vee K_n\neg\alpha)$.

Since β is a pure knowledge formula, we can think of β as describing certain knowledge of the processes. So axiom A6′ says that if the processes have certain knowledge, and if some process knows that such knowledge would be incompatible with the primitive state α, then some process knows that the primitive state α is impossible. For historical reasons we refer to $S5_n$ + A6′ (respectively $S5D_n$ + A6) as ML_n^- (respectively ML_n), where "ML" stands for "Message Logic" [FV2]. It turns out that ML_n implies ML_n^-, that is, axiom A6′ can be proven in ML_n (we shall prove this in the appendix in the proof of Proposition 5.3 below). However, the converse is not true.

The next proposition says that in a precise sense, the pasting condition corresponds to the new axiom A6 if distributed knowledge is in the language, and to A6′ if it is not.

Proposition 5.3: *Let $\mathcal{C}$ be a class of interpreted systems with n processes. If every attainable knowledge situation in every member of $\mathcal{C}$ satisfies the pasting condition, then ML_n^- (respectively ML_n) is sound with respect to $\mathcal{C}$. Conversely, if every knowledge situation that satisfies the pasting condition is attainable in some member of $\mathcal{C}$, then ML_n^- (respectively ML_n) is complete with respect to $\mathcal{C}$ for the language $\mathcal{L}_n(\Phi)$ (respectively, $\mathcal{L}_n^D(\Phi)$).*

Proof: In the case of $\mathcal{L}_n^D(\Phi)$, the proof of the theorem proceeds by showing that a set S of interpreted states satisfies the pasting condition if and only if (S, s) satisfies every instance of axiom A6 for every $s \in S$. In the case of $\mathcal{L}_n(\Phi)$, the proof is somewhat more delicate. The details are in the appendix. ∎

The first part of Proposition 5.3 tells us that if $\mathcal{I}$ is an interpreted system where every attainable knowledge situation satisfies the pasting condition, then the new axioms A6 and A6′ are guaranteed to be satisfied at every point of $\mathcal{I}$. The second part of Proposition 5.3 tells us that if every knowledge situation that satisfies the pasting condition is attainable in some member of the class $\mathcal{C}$ of interpreted systems, then no new axioms other than A6′ (respectively, A6) are required to prove every formula of $\mathcal{L}_n(\Phi)$ (respectively, $\mathcal{L}_n^D(\Phi)$) that is sound with respect to $\mathcal{C}$.

Before we give our first application of Proposition 5.3, we consider the assumptions that we have noted were implicit in our Alice and Bob example, and show (among other things) that these conditions guarantee the pasting condition.

Proposition 5.4: *If $\mathcal{I}$ is an interpreted system where (1) the environment determines the initial states, (2) knowledge is cumulative, (3) process state transitions are independent of the initial environment, and (4) the primitive propositions characterize the initial environment, then all the knowledge situations attainable in $\mathcal{I}$ satisfy the pasting condition. Conversely, if a knowledge situation satisfies the pasting condition, then it is attainable in some interpreted system $\mathcal{I}$ satisfying these four assumptions.*

Proof: See the appendix. ∎

Note that the assumptions of Proposition 5.4 are not unreasonable. They hold for "ideal" sensors or robots observing and communicating about an external environment.

The next theorem is an immediate consequence of Propositions 5.3 and 5.4.

Theorem 5.5: *ML_n^- (respectively, ML_n) is a sound and complete axiomatization with respect to interpreted systems of n processes where (1) the environment determines the initial states, (2) knowledge is cumulative, (3) process state transitions are independent of the initial environment, and (4) the primitive propositions characterize the initial environment, for the language $\mathcal{L}_n(\Phi)$ (respectively, $\mathcal{L}_n^D(\Phi)$).*

Soundness and completeness theorems for ML_n^- and ML_n are also proven by Fagin and Vardi in [FV2], but in a rather different setting from ours. The model in [FV2] is much more concrete than the model here; in particular there is in their model a particular notion of communication by which processes change their states. Here we have an abstract model in which, by analyzing the implicit and explicit assumptions in [FV2], we have captured the essential assumptions required for the pasting property (and hence axiom A6) to hold. While soundness in [FV2] follows easily from soundness in the model here, the completeness proof is much more difficult there.

Recall from our Alice and Bob example in the introduction that the assumptions we made all seemed to be necessary. The following theorem, which is proven in the appendix, confirms this fact. It shows that if we drop any one of the assumptions of Theorem 5.5, then all knowledge situations are attainable, and $\mathrm{S5}_n$ (respectively $\mathrm{S5D}_n$) becomes complete.

Theorem 5.6: *Let $\mathcal{A}$ be a proper subset of the four conditions of Theorem 5.5, and let $\mathcal{C}$ be the class of all interpreted systems that satisfy the conditions $\mathcal{A}$. Then each knowledge situation is attainable in a member of $\mathcal{C}$. Thus, $S5_n$ (respectively $S5D_n$) is a sound and complete axiomatization with respect to $\mathcal{C}$ for the language $\mathcal{L}_n(\Phi)$ (respectively, $\mathcal{L}_n^D(\Phi)$).*

Theorems 5.5 and 5.6 show that we have captured those conditions that cause an extra axiom to hold in our Alice and Bob example. This provides an excellent example of how special properties of an interpreted system can cause extra properties of knowledge to hold.

We now comment briefly on how our results would change if the set Φ of primitive propositions were infinite. As we noted earlier, a primitive state α could not then be identified with a formula in our language, since each truth assignment would require an infinite description. Hence, the axioms A6 and A6$'$ would not even be formulas in our language. However, it is still *a priori* conceivable that there could be other "extra axioms" for knowledge situations that satisfy the pasting condition. We note that interestingly enough, this is not the case; if Φ is infinite, then we can replace ML_n^- (respectively ML_n) everywhere in Proposition 5.3 and Theorem 5.5 by $\mathrm{S5}_n$ (respectively $\mathrm{S5D}_n$). Put differently, there are no new axioms if we have infinitely many propositions in our language![5]

We remark that in Proposition 5.4 and Theorem 5.5 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 situations is attainable, and again knowledge has extra properties. This is discussed in detail in the appendix.

[5]The idea of the proof is as follows. Let φ be a formula that is consistent with $\mathrm{S5D}_n$. Assume that Ψ is the (finite) set of primitive propositions that appear in φ. Let M be an S5 Kripke structure over the primitive propositions in Ψ such that φ is satisfiable in M. It is easy to show that there is an S5 Kripke structure M' over all of Φ that satisfies the pasting condition, and is identical to M when restricted to Ψ. The result then follows.

A natural condition for distributed systems applications that is sufficient to guarantee the pasting condition (and hence axiom A6) is that each primitive proposition is determined by the current state of some process. For example, the primitive propositions can talk about the current values of various variables local to individual processes. As we noted earlier, the condition that each primitive proposition is determined by the current state of some process holds also if each primitive proposition is determined by the *initial* state of some process, and if knowledge is cumulative.

In fact, if each primitive proposition is determined by the current state of some process, then not only is the axiom A6 sound, but even more, the stronger axiom A7 below is sound, where α is a primitive state and $\{1, ..., n\}$ is the set of processes.

A7. $\neg\alpha \Rightarrow K_1\neg\alpha \vee ... \vee K_n\neg\alpha$.

Axiom A7 indeed implies A6, since $D\neg\alpha \Rightarrow \neg\alpha$.

We now explain why axiom A7 is sound when each primitive proposition is determined by the current state of some process. Let α' be the current primitive state. Assume that α is false; hence, $\alpha \neq \alpha'$. Then there is some primitive proposition p whose truth value is different under α than under α'. Since each primitive proposition is determined by the current state of some process, there is some process i such that the primitive proposition p talks about the current state of process i. Then process i knows that α is not the current primitive state.

We note that when we consider the class of interpreted systems where each primitive proposition is determined by the current state of some process, then although $S5_n$ + A7 is sound, it is not complete; further axioms are required. For example, if q is of the form either p or $\neg p$ for some primitive proposition p, then we have the axiom

$$q \Rightarrow K_1 q \vee \ldots K_n q.$$

Let us now consider the slightly more general situation where the primitive propositions are determined by the current states of the processes. Recall that this means that whenever $r(m) \sim r'(m')$, then $\pi(r, m) = \pi(r', m')$. It is easy to see that this condition is not sufficient to guarantee the pasting condition. However, in this case, every knowledge situation that is attainable satisfies the following condition.

Definition 5.7: *The process component uniquely determines the primitive state* in a set S of interpreted states if whenever t and t' are members of S where $t \sim t'$, then $t = t'$. *The process component uniquely determines the primitive state in the knowledge situation* (S, s) if the process component uniquely determines the primitive state component in S. ∎

We have the following proposition, which is completely analogous to Proposition 5.4.

Proposition 5.8: *If $\mathcal{I}$ is an interpreted system where the primitive propositions are determined by the current states of the processes, then in every knowledge situation attainable in $\mathcal{I}$, the process component uniquely determines the primitive state. Conversely, each knowledge situation where the process component uniquely determines the primitive state is attainable in some interpreted system where the primitive propositions are determined by the current states of the processes.*

Proof: See the appendix. ∎

What is the analogue to Proposition 5.3? That is, what new axiom corresponds to the condition that the process component uniquely determines the primitive state? This condition can be characterized by the following axiom:

A8. $\varphi \Rightarrow D\varphi$.

The converse to axiom A8, namely $D\varphi \Rightarrow \varphi$, which says that everything that is distributed knowledge is true, is one of the axioms of $\mathrm{S5D}_n$. Axiom A8 says that everything that is true is distributed knowledge. The following axiom, where α is a primitive state, is of course a special case of axiom A8:

A8*. $\alpha \Rightarrow D\alpha$.

As we shall show in the appendix, $\mathrm{S5D}_n$ + A8 is equivalent to $\mathrm{S5D}_n$ + A8*. We note also that clearly axiom A8 (where we take φ to be $\neg\alpha$) along with axiom A6 imply axiom A7. These axioms hold in the situation where each primitive proposition is determined by the current state of some process.

We have just said that if the process component uniquely determines the primitive state, then axiom A8 (in the language $\mathcal{L}_n^D(\Phi)$) is sound. What new axiom is needed in the language $\mathcal{L}_n(\Phi)$? Somewhat surprisingly (and in contrast to the situation in Theorem 5.5), it turns out that if we restrict our attention to $\mathcal{L}_n(\Phi)$, then $\mathrm{S5}_n$ *is* a complete axiomatization. No new axioms are required![6] Thus, we have the following analogues to Proposition 5.3 and Theorem 5.5:

Proposition 5.9: *Let $\mathcal{C}$ be a class of interpreted systems with $n \geq 2$ processes. If for every attainable knowledge situation in every member of $\mathcal{C}$, the process component uniquely determines the primitive state, then $S5_n$ (respectively, $S5D_n$ + A8) is sound with respect to $\mathcal{C}$. Conversely, if every knowledge situation where the process component uniquely determines the primitive state is attainable in some member of $\mathcal{C}$, then $S5_n$ (respectively, $S5D_n$ + A8) is complete with respect to $\mathcal{C}$ for the language $\mathcal{L}_n(\Phi)$ (respectively, $\mathcal{L}_n^D(\Phi)$).*

[6] Actually, in the degenerate case $n = 1$, where there is exactly one process, there is a new axiom, namely $\varphi \Rightarrow K_1\varphi$. For simplicity, in the remaining results of this section we consider only the case where $n \geq 2$.

Proof: See appendix. ∎

Theorem 5.10: *$S5_n$ (respectively, $S5D_n$ + A8) is a sound and complete axiomatization with respect to interpreted systems of $n \geq 2$ processes where the primitive propositions are determined by the current states of the processes, for the language $\mathcal{L}_n(\Phi)$ (respectively, $\mathcal{L}_n^D(\Phi)$).*

Proof: This follows immediately from Propositions 5.8 and 5.9. ∎

In the case of $\mathcal{L}_n(\Phi)$, this theorem shows that there are cases where the language may not be sufficiently powerful to capture the fact that not all knowledge situations are attainable.

If the primitive propositions are determined by the *initial* states of the processes, and if knowledge is cumulative, then as we have noted, the primitive propositions are determined by the *current* states of the processes. So from Theorem 5.10, we know that if the primitive propositions are determined by the initial states of the processes, and if knowledge is cumulative, then $S5D_n$ + A8 is sound. In fact, it is also complete. Thus, Proposition 5.8 and Theorem 5.10 both hold when we replace every occurrence of "the primitive propositions are determined by the current states of the processes" by "the primitive propositions are determined by the initial states of the processes and knowledge is cumulative". In particular, we have

Theorem 5.11: *$S5_n$ (respectively, $S5D_n$ + A8) is a sound and complete axiomatization with respect to interpreted systems of $n \geq 2$ processes where the primitive propositions are determined by the initial states of the processes and knowledge is cumulative, for the language $\mathcal{L}_n(\Phi)$ (respectively, $\mathcal{L}_n^D(\Phi)$).*

The proofs of these results can be found in the appendix.

6 Conclusions

We have presented a general model for the knowledge of processes in a system and shown how the properties of knowledge may depend on the subtle interaction of the parameters of the system and on what the primitive propositions talk about. 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 various cases of interest.

An interesting direction in which to extend the line of research of this paper is to consider the effect of adding to the language common knowledge and/or time.[7] By results of [HM1] (since reproved and generalized in [CM, FI]), we know that under certain natural assumptions, common knowledge will not be attainable. For example, if at most one process changes its state at any given time, then no common knowledge is gained beyond the common knowledge that the processes already had in the initial state. So this assumption, along with suitable assumptions about the initial states, will cause extra axioms to hold 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, HV1, HV2]). Results of [HV1, HV2] imply that under certain combinations of our parameters, there is no complete, recursively enumerable axiomatization in a language that includes both common knowledge and time. For example, knowledge being cumulative is by itself enough to cause this unpleasant phenomenon.

7 Appendix

In this appendix, we prove the results stated (and those alluded to) in the body of the paper.

7.1 Completeness of $\mathbf{S5D}_n$

In this subsection, we shall prove that $\mathrm{S5D}_n$ is a sound and complete axiomatization with respect to S5 Kripke structures of n processes, for the language $\mathcal{L}_n^D(\Phi)$. This result was stated without proof in [HM2]. We shall also develop some general tools that will be useful later in the paper; indeed, we feel that these tools will be useful in other contexts in the future.

We begin by sketching a completeness proof for $\mathrm{S5}_n$ for the language $\mathcal{L}_n(\Phi)$, since our later proofs will be in the same spirit. (The completeness result was proved by Kripke [Kr] and Hintikka [Hi] for S5, and extended by Halpern and Moses to $\mathrm{S5}_n$ [HM2]. The "maximal consistent sets" construction that we use is due to Makinson [Ma].)

Theorem 7.1: *$S5_n$ is a sound and complete axiomatization with respect to S5 Kripke structures of n processes, for the language $\mathcal{L}_n(\Phi)$.*

Sketch of proof: Soundness is straightforward, as usual, so we focus on completeness.

[7]A formula φ is *common knowledge* if $E^k\varphi$ holds for every $k \geq 1$, where $E\psi$ is an abbreviation for $\bigwedge_{i=1}^n K_i\psi$ and $E^k\varphi$ is $E(E^{k-1}\varphi)$. Namely, φ is common knowledge if every process knows φ, every process knows that every process knows φ, etc.

A formula φ is said to be *inconsistent* if its negation $\neg\varphi$ can be proven in $S5_n$. Otherwise, φ is said to be *consistent*. A set Σ of formulas is said to be inconsistent if there is a finite subset $\{\sigma_1, ..., \sigma_k\} \subseteq \Sigma$ such that the formula $\sigma_1 \wedge ... \wedge \sigma_k$ is inconsistent; otherwise, Σ is said to be consistent. A *maximal consistent set of formulas* is a consistent set V of formulas such that whenever ψ is a formula not in V, then $V \cup \{\psi\}$ is inconsistent. Let *MAXCON* be the set of all maximal consistent sets of formulas.

Let us say that a formula $\varphi \in \mathcal{L}_n(\Phi)$ is *satisfiable* if there is an S5 Kripke structure M and a state s of M such that $(M, s) \models \varphi$. In order to prove completeness, we must show that every valid formula is provable, or equivalently, that every consistent formula is satisfiable. We shall now construct a certain S5 Kripke structure M^c, called the "canonical Kripke structure", such that for every consistent formula $\varphi \in \mathcal{L}_n(\Phi)$, there is a state s such that $(M^c, s) \models \varphi$. This is sufficient to prove completeness.

If V is a set of formulas and i is a process, define V/K_i to be $\{\varphi |\, K_i\varphi \in V\}$. For each $V \in MAXCON$, we define a new, distinct state s_V, and let $S = \{s_V | V \in MAXCON\}$. Let $\mathcal{K}_i = \{(s_{V_1}, s_{V_2}) |\, V_1, V_2 \in MAXCON, \text{ and } V_1/K_i \subseteq V_2\}$, for each process i. By making use of the axioms, it is not hard to verify that each $\mathcal{K}_i$ is an equivalence relation. As an illustration, we show that each $\mathcal{K}_i$ is symmetric. Assume that $(s_{V_1}, s_{V_2}) \in \mathcal{K}_i$; we must show that $(s_{V_2}, s_{V_1}) \in \mathcal{K}_i$. That is, assume that $V_1/K_i \subseteq V_2$; we must show that $V_2/K_i \subseteq V_1$. Assume not; then there is some formula φ such that $K_i\varphi \in V_2$ but $\varphi \notin V_1$. If $K_i\varphi \in V_1$, then $\varphi \in V_1$, since $K_i\varphi \Rightarrow \varphi$ is one of the axioms, and since V_1 is a maximal consistent set. Therefore, $K_i\varphi \notin V_1$. Since $K_i\varphi \notin V_1$ and V_1 is maximal, it follows that $\neg K_i\varphi \in V_1$. So $K_i\neg K_i\varphi \in V_1$, since $\neg K_i\varphi \Rightarrow K_i\neg K_i\varphi$ is one of the axioms, and since V_1 is a maximal consistent set. Therefore, since $V_1/K_i \subseteq V_2$, it follows that $\neg K_i\varphi \in V_2$. But by assumption, $K_i\varphi \in V_2$. So V_2 is inconsistent, a contradiction.

We define π by letting $\pi(s_V)$ be the truth assignment where the primitive proposition p is true if $p \in V$, and false if $p \notin V$. Let M^c be the S5 Kripke structure $(S, \pi, \mathcal{K}_1, ..., \mathcal{K}_n)$. It can be shown (see, for example, [Ma] or [HM2]) that for every formula φ in $\mathcal{L}_n(\Phi)$, we have $(M^c, s_V) \models \varphi$ iff $\varphi \in V$. Since every consistent formula is contained in some maximal consistent set, this shows that every consistent formula is satisfiable, as desired. ∎

Before we prove completeness of $S5D_n$, we need some preliminary definitions and results, which will prove to be useful again later on.

Assume that $\mathcal{B} \subseteq \mathcal{L}_n^D(\Phi)$. We shall typically take $\mathcal{B}$ to be the set of all instances of some set of axioms. A formula φ is said to be *$\mathcal{B}$-inconsistent* if its negation $\neg\varphi$ can be proven in $S5D_n + \mathcal{B}$. Otherwise, φ is said to be *$\mathcal{B}$-consistent*. We define what it means for a set of formulas to be $\mathcal{B}$-consistent or to be a maximal $\mathcal{B}$-consistent set of formulas just as we did earlier for consistency. A *$\mathcal{B}$-Kripke structure* is an S5 Kripke structure M such that $(M, s) \models \psi$ for every state s of M and every $\psi \in \mathcal{B}$.

A *pseudo-Kripke structure* $M^* = (S, \pi, \mathcal{K}_1, ..., \mathcal{K}_n, \mathcal{K}_D)$ is an S5 Kripke structure (of $n+1$ processes) where D is considered just another process. Further, if $n = 1$ (that is, if there is exactly one process), and if $M^* = (S, \pi, \mathcal{K}_1, \mathcal{K}_D)$, then we demand that

$\mathcal{K}_D = \mathcal{K}_1$. If $s \in S$, we define what it means for (M^*, s) to *pseudo-satisfy* a formula $\varphi \in \mathcal{L}_n^D(\Phi)$, written $(M^*, s) \models^* \varphi$, just as we ordinarily define satisfaction for Kripke structures, except that $(M^*, s) \models^* D\psi$ iff $(M^*, s') \models^* \psi$ for all s' such that $(s, s') \in \mathcal{K}_D$. Thus, pseudo-satisfaction is defined for pseudo-Kripke structures by treating D as just another process, rather than as "distributed knowledge". A *$\mathcal{B}$-pseudo-Kripke structure* is a pseudo-Kripke structure M^* such that $(M^*, s) \models^* \psi$ for every state s of M^* and every $\psi \in \mathcal{B}$.

Let us say that a formula $\varphi \in \mathcal{L}_n^D(\Phi)$ is *$\mathcal{B}$-satisfiable* if there is a $\mathcal{B}$-Kripke structure M and a state s of M such that $(M, s) \models \varphi$. We may then say that φ is *$\mathcal{B}$-satisfiable in M*. We say that φ is *satisfiable (in M)* if φ is $\emptyset$-satisfiable (in M). Define φ to be *$\mathcal{B}$-pseudo-satisfiable* if there is a $\mathcal{B}$-pseudo-Kripke structure M^* and a state s of M^* such that $(M^*, s) \models^* \varphi$. We may then say that φ is *$\mathcal{B}$-pseudo-satisfiable in M^**.

We shall prove the following two facts:

1. If φ is $\mathcal{B}$-consistent, then φ is $\mathcal{B}$-pseudo-satisfiable.

2. If φ is $\mathcal{B}$-pseudo-satisfiable, then φ is $\mathcal{B}$-satisfiable.

Putting these two facts together, it follows that if φ is $\mathcal{B}$-consistent, then φ is $\mathcal{B}$-satisfiable, and in particular, φ is satisfiable. Therefore, by letting $\mathcal{B} = \emptyset$, we immediately obtain completeness of S5D$_n$.

We begin by proving the first fact.

Proposition 7.2: *Assume that $\varphi \in \mathcal{L}_n^D(\Phi)$ and $\mathcal{B} \subseteq \mathcal{L}_n^D(\Phi)$. If φ is $\mathcal{B}$-consistent, then φ is $\mathcal{B}$-pseudo-satisfiable.*

Proof: The proof follows by using the maximal consistent sets construction as in the proof of Theorem 7.1, where we use $\mathcal{B}$-consistency instead of consistency. The construction gives us a pseudo-Kripke structure M^*, with states s_V for every maximal $\mathcal{B}$-consistent set $V \subseteq \mathcal{L}_n^D(\Phi)$, such that $(M^*, s_V) \models^* \psi$ iff $\psi \in V$. In particular, since it is easy to see that every maximal $\mathcal{B}$-consistent set contains $\mathcal{B}$, it follows that M^* is a $\mathcal{B}$-pseudo-Kripke structure. Actually, there is one remaining point to verify: if $n = 1$ (that is, if there is exactly one process), and if $M^* = (S, \pi, \mathcal{K}_1, \mathcal{K}_D)$, then we must show that $\mathcal{K}_D = \mathcal{K}_1$. In this case, $\mathcal{B}$ contains the axioms $K_1\psi \Rightarrow D\psi$ and $D\psi \Rightarrow K_1\psi$, for every formula ψ. It follows easily from the construction that therefore $\mathcal{K}_D = \mathcal{K}_1$. Note that the only place in the proof where we use the fact that $\mathcal{B}$ contains every axiom of S5D$_n$ occurs in the case $n = 1$. ∎

Our next step towards the proof of completeness of S5D$_n$ is to show that if $\mathcal{B}$ contains every axiom of S5D$_n$, and if φ is $\mathcal{B}$-pseudo-satisfiable, then φ is $\mathcal{B}$-satisfiable. Before we can prove this, we need further preliminary results.

Let M be an S5 Kripke structure of n processes, and let s be a state of M. We define the *$\mathcal{L}_n(\Phi)$-type of* (M, s) to be the set of formulas $\varphi \in \mathcal{L}_n(\Phi)$ such that $(M, s) \models \varphi$.

Let $M = (S, \pi, \mathcal{K}_1, ..., \mathcal{K}_n)$ be a Kripke structure, and let s, t be states (members of S). We call a sequence $\langle v_1, i_1, v_2, i_2, \ldots, i_{k-1}, v_k\rangle$ where $k \geq 1$ a *path from s to t (in M)* if

1. $v_1 = s$,
2. $v_k = t$,
3. $v_1, \ldots, v_k$ are states,
4. $i_1, \ldots, i_{k-1}$ are processes, and
5. $(v_j, v_{j+1}) \in \mathcal{K}_{i_j}$, for $1 \leq j < k$.

The *reduction* of a path $\langle v_1, i_1, v_2, i_2, \ldots, i_{k-1}, v_k\rangle$ is obtained by replacing each maximal consecutive subsequence $\langle v_q, i_q, v_{q+1}, i_{q+1}, \ldots, i_{r-1}, v_r\rangle$ where $i_q = i_{q+1} = \cdots = i_{r-1}$ by $\langle v_q, i_q, v_r\rangle$. The reduction of a path from s to t is a path from s to t, by transitivity of the $\mathcal{K}_j$'s. A path is said to be *reduced* if it equals its own reduction. Thus, a path $\langle v_1, i_1, v_2, i_2, \ldots, i_{k-1}, v_k\rangle$ is reduced precisely if $i_j \neq i_{j+1}$ for $1 \leq j < k$. We say that M is *tree-like* if whenever s and t are states of M, then there is at most one reduced path from s to t in M. Note in particular that if M is tree-like and s and t are distinct states of M such that $(s, t) \in \mathcal{K}_i$ and $(s, t) \in \mathcal{K}_j$, then $i = j$.

Proposition 7.3: *Let M be an S5 Kripke structure of n processes. There is an S5 Kripke structure M' of n processes such that*

1. *M' is tree-like, and*
2. *M and M' have precisely the same $\mathcal{L}_n(\Phi)$-types. That is,*
   (a) *For every state s of M there is a state s' of M' such that the $\mathcal{L}_n(\Phi)$-types of (M, s) and (M', s') are the same.*
   (b) *For every state s' of M' there is a state s of M such that the $\mathcal{L}_n(\Phi)$-types of (M, s) and (M', s') are the same.*

Proof: Intuitively, we shall "unwind" the Kripke structure $M = (S, \pi, \mathcal{K}_1, ..., \mathcal{K}_n)$ into a forest (not necessarily a tree, since the structure we create is not necessarily connected). Our approach is to create states at various "levels". The first level T_1 contains precisely S, the set of states of M. Assume inductively that we have defined the set T_k of states at level k. Then for each $s \in S$, each $v \in T_k$, and each process i, we define a new, distinct state $z_{s,v,i}$. We may refer to $z_{s,v,i}$ as an *i-child of v*, and to v as the *parent* of $z_{s,v,i}$. The set T_{k+1} of states at level $k+1$ consist of all these states $z_{s,v,i}$. Let $T = \cup\{T_k |\, k \geq 1\}$. Define $g : T \rightarrow S$ by letting $g(s) = s$ if $s \in T_1$, and $g(z_{s,v,i}) = s$ for $z_{s,v,i} \in T_k$ where $k \geq 2$. Intuitively, we shall construct M' with state space T such that the state $s \in T$

"mimics" the state $g(s) \in S$ (by "mimics", we mean that the $\mathcal{L}_n(\Phi)$-types of (M', s) and $(M, g(s))$ will be the same).

Define $\mathcal{K}_i''$ for each process i by letting (s,t) be in $\mathcal{K}_i''$ iff t is an i-child of s and $(g(s), g(t)) \in \mathcal{K}_i$. Let $\mathcal{K}_i'$ be the reflexive, symmetric, transitive closure of $\mathcal{K}_i''$ (that is, the smallest equivalence relation that contains $\mathcal{K}_i''$). In order to give a useful characterization of $\mathcal{K}_i'$, we need some more definitions.

If $s, t \in T$, we call a sequence $\langle v_1, i_1, v_2, i_2, \ldots, i_{k-1}, v_k\rangle$ where $k \geq 1$ a *primitive path from s to t* if

1. $v_1 = s$,
2. $v_k = t$,
3. $v_1, \ldots, v_k \in T$
4. $i_1, \ldots, i_{k-1}$ are processes, and
5. either $(v_j, v_{j+1}) \in \mathcal{K}_{i_j}''$ or $(v_{j+1}, v_j) \in \mathcal{K}_{i_j}''$, for $1 \leq j < k$.

We call the primitive path an *i-primitive path* if $i_j = i$ for $1 \leq j < k$. We say that a primitive path $\langle v_1, i_1, v_2, i_2, \ldots, i_{k-1}, v_k\rangle$ is *nonredundant* if there is no j such that $v_j = v_{j+2}$ and $i_j = i_{j+1}$. Thus, in a nonredundant path there is no consecutive subsequence $\langle v_j, i_j, v_{j+1}, i_j, v_j\rangle$, where intuitively, we go forward along some edge and then immediately backward along the same edge. Since every state has at most one parent, and since the only $\mathcal{K}_j''$ edges are between adjacent levels, it is easy to see that whenever $s, t \in T$, then there is at most one nonredundant primitive path from s to t.

It is straightforward to see that $(s,t) \in \mathcal{K}_i'$ iff there is an i-primitive path from s to t. Using this fact and the fact that $\mathcal{K}_i$ is an equivalence relation, it follows fairly easily that if $(s,t) \in \mathcal{K}_i'$, then $(g(s), g(t)) \in \mathcal{K}_i$.

Define π' by letting $\pi'(s) = \pi(g(s))$ for each $s \in T$. Let M' be the S5 Kripke structure $(T, \pi', \mathcal{K}_1', ..., \mathcal{K}_n')$.

We now show that M' is tree-like. Let $P = \langle v_1, i_1, v_2, i_2, \ldots, i_{k-1}, v_k\rangle$ and $P' = \langle v_1', i_1', v_2', i_2', \ldots, i_{k-1}', v_k'\rangle$ be reduced paths from s to t. Since $(v_j, v_{j+1}) \in \mathcal{K}_{i_j}'$, for $1 \leq j < k$, it follows as we noted earlier that there is an i_j-primitive path from v_j to v_{j+1}, and hence a nonredundant i_j-primitive path from v_j to v_{j+1}. Let $\widehat{P}$ be the primitive path from s to t obtained from P by "splicing in" a nonredundant i_j-primitive path from v_j to v_{j+1} in place of $\langle v_j, i_j, v_{j+1}\rangle$, for $1 \leq j < k$. Note that $\widehat{P}$ is nonredundant, and when we think of $\widehat{P}$ as a path rather than a primitive path, P is the reduction of $\widehat{P}$. Similarly, let $\widehat{P'}$ be a nonredundant primitive path from s to t with reduction P'. By uniqueness of nonredundant primitive paths from s to t, we know that $\widehat{P} = \widehat{P'}$. So the reductions of $\widehat{P}$ and $\widehat{P'}$ are the same, that is, $P = P'$. Hence, M' is tree-like.

We now show that the $\mathcal{L}_n(\Phi)$-types of (M', s) and $(M, g(s))$ are the same. We prove, by induction on the structure of $\varphi \in \mathcal{L}_n(\Phi)$, that

$$(M', s) \models \varphi \text{ iff } (M, g(s)) \models \varphi. \tag{2}$$

If φ is a primitive proposition, this follows from the fact that $\pi'(s) = \pi(g(s))$. The case where φ is a Boolean combination of formulas is immediate. We now consider the interesting case, where φ is of the form $K_i\psi$. Assume first that $(M', s) \not\models K_i\psi$. Thus, there is a state t of M' such that $(s, t) \in \mathcal{K}'_i$ and $(M', t) \not\models \psi$. As we showed above, $(g(s), g(t)) \in \mathcal{K}_i$. Since $(M', t) \not\models \psi$, it follows by inductive assumption that $(M, g(t)) \not\models \psi$. So $(M, g(s)) \not\models K_i\psi$, as desired. Assume now that $(M, g(s)) \not\models K_i\psi$. Therefore, there is a state w of M such that $(g(s), w) \in \mathcal{K}_i$ and $(M, w) \not\models \psi$. By construction, $z_{w,s,i} \in T$. Further $(s, z_{w,s,i}) \in \mathcal{K}''_i$, since $g(z_{w,s,i}) = w$ and $(g(s), w) \in \mathcal{K}_i$. Therefore, $(s, z_{w,s,i}) \in \mathcal{K}'_i$. By inductive assumption, since $(M, w) \not\models \psi$, also $(M', z_{w,s,i}) \not\models \psi$. So $(M', s) \not\models K_i\psi$, as desired.

We just showed that the $\mathcal{L}_n(\Phi)$-types of (M', s) and $(M, g(s))$ are the same. Then 2(a) in the statement of the proposition holds by letting $s' = s$, since $S = T_1 \subset T$ and $g(s) = s$ for $s \in S$. Also, 2(b) holds, by letting $s = g(s')$. ∎

The next proposition shows how to convert pseudo-satisfaction into satisfaction. As we noted earlier, we shall use it in our completeness proof of $\mathrm{S5D}_n$.

Proposition 7.4: *Assume that $\varphi \in \mathcal{L}_n^D(\Phi)$ and $\mathcal{B} \subseteq \mathcal{L}_n^D(\Phi)$. If φ is $\mathcal{B}$-pseudo-satisfiable, then φ is $\mathcal{B}$-satisfiable.*

Proof: Assume that $\varphi \in \mathcal{L}_n^D(\Phi)$. Consider first the case where the number n of processes is one. Assume that φ is pseudo-satisfiable in the $\mathcal{B}$-pseudo-Kripke structure $M^* = (S, \pi, \mathcal{K}_1, \mathcal{K}_D)$. Let M be the S5 Kripke structure $(S, \pi, \mathcal{K}_1)$. Since M^* is a pseudo-Kripke structure, $\mathcal{K}_D = \mathcal{K}_1$. Therefore, it is easy to show by induction on the structure of formulas that for every $\psi \in \mathcal{L}_n^D(\Phi)$ and for every state s of M^*, we have $(M^*, s) \models^* D\psi$ iff $(M, s) \models K_1\psi$ iff $(M, s) \models D\psi$. Since φ is pseudo-satisfiable in M^*, it therefore follows that φ is satisfiable in M. Furthermore, since M^* is a $\mathcal{B}$-pseudo-Kripke structure, M is a $\mathcal{B}$-Kripke structure. So φ is $\mathcal{B}$-satisfiable.

Now consider the case $n \geq 2$. By Proposition 7.3, we can assume without loss of generality that there is a tree-like $\mathcal{B}$-pseudo-structure M^* and a state s such that $(M^*, s) \models \varphi$. Let M^* be $(S, \pi, \mathcal{K}_1^*, \ldots, \mathcal{K}_n^*, \mathcal{K}_D^*)$. Define $\mathcal{K}_i$, for $1 \leq i \leq n$, to be the transitive closure of $\mathcal{K}_i^* \cup \mathcal{K}_D^*$. Let $M = (S, \pi, \mathcal{K}_1, \ldots, \mathcal{K}_n)$. Note that M and M^* have the same state space S and the same π. Since $\mathcal{K}_i^*$ and $\mathcal{K}_D^*$ are reflexive and symmetric, it follows easily that $\mathcal{K}_i$ is an equivalence relation. Therefore, M is an S5 Kripke structure. We now show, by induction on the structure of formulas $\gamma \in \mathcal{L}_n^D(\Phi)$, that

$$(M, s) \models \gamma \text{ iff } (M^*, s) \models^* \gamma. \tag{3}$$

Since φ is $\mathcal{B}$-pseudo-satisfiable in M^*, this is clearly sufficient to prove the lemma. If γ is a primitive proposition, then (3) is immediate, since M and M^* have the same π. The case where γ is a Boolean combination of formulas for which the analog of (3) holds is immediate. We now consider the case where γ is of the form $K_i\psi$.

Assume first that $(M^*, s) \not\models^* K_i\psi$. Thus, there is a state $t \in S$ such that $(s,t) \in \mathcal{K}_i^*$ and $(M^*, t) \not\models^* \psi$. Since $\mathcal{K}_i^* \subseteq \mathcal{K}_i$, we have $(s,t) \in \mathcal{K}_i$. By inductive assumption, $(M,t) \not\models \psi$. So $(M,s) \not\models K_i\psi$.

Assume now that $(M^*, s) \models^* K_i\psi$. To show that $(M,s) \models K_i\psi$, we must show that $(M,t) \models \psi$ whenever $(s,t) \in \mathcal{K}_i$. Assume that $(s,t) \in \mathcal{K}_i$. Since $\mathcal{K}_i$ is the transitive closure of $\mathcal{K}_i^* \cup \mathcal{K}_D^*$, there are $v_1, \ldots, v_k \in S$ such that

1. $v_1 = s$,
2. $v_k = t$,
3. either $(v_j, v_{j+1}) \in \mathcal{K}_i^*$ or $(v_j, v_{j+1}) \in \mathcal{K}_D^*$, for $1 \leq j < k$.

We now show, by induction on j (where $1 \leq j \leq k$) that $(M^*, v_j) \models^* K_i\psi$. The case $j = 1$ is by assumption. Assume inductively on j that $(M^*, v_j) \models^* K_i\psi$ (for some j where $1 \leq j \leq k-1$); we must show that $(M^*, v_{j+1}) \models^* K_i\psi$. Since $(M^*, v_j) \models^* K_i\psi$, it follows that $(M^*, v_j) \models^* K_iK_i\psi$. We know that either $(v_j, v_{j+1}) \in \mathcal{K}_i^*$ or $(v_j, v_{j+1}) \in \mathcal{K}_D^*$. In the first case, since $(M^*, v_j) \models^* K_iK_i\psi$, it follows that $(M^*, v_{j+1}) \models^* K_i\psi$, as desired. In the second case, since the formula $K_iK_i\psi \Rightarrow DK_i\psi$ is an instance of axiom KD, it follows that $(M^*, v_j) \models^* (K_iK_i\psi \Rightarrow DK_i\psi)$, and so $(M^*, v_j) \models^* DK_i\psi$. Hence, since $(v_j, v_{j+1}) \in \mathcal{K}_D^*$, it follows again that $(M^*, v_{j+1}) \models^* K_i\psi$. This completes the induction. It follows that $(M^*, t) \models^* K_i\psi$. Therefore, $(M^*, t) \models^* \psi$. So by inductive assumption, $(M,t) \models \psi$. This was to be shown.

Finally, consider the case where γ is of the form $D\psi$. Assume first that $(M^*, s) \not\models^* D\psi$. Thus, there is a state $t \in S$ such that $(s,t) \in \mathcal{K}_D^*$ and $(M^*, t) \not\models^* \psi$. By inductive assumption, $(M,t) \not\models \psi$. Since $\mathcal{K}_D^* \subseteq \mathcal{K}_i$ for each i (with $1 \leq i \leq n$), it follows that $(s,t) \in \mathcal{K}_i$ for each i, and so $(M,s) \not\models D\psi$.

Assume now that $(M,s) \not\models D\psi$. Thus, there is a state $t \in S$ such that $(s,t) \in \mathcal{K}_i$, for $1 \leq i \leq n$, and $(M,t) \not\models \psi$. So $(M^*, t) \not\models^* \psi$ by inductive assumption. Since $(s,t) \in \mathcal{K}_1$, there is a path $P_1 = \langle v_1, i_1, v_2, i_2, \ldots, i_{k-1}, v_k \rangle$ from s to t in M^*, such that each i_j is either 1 or D. By taking the reduction, we can assume that P_1 is reduced. Similarly, there is a reduced path $P_2 = \langle v_1, i_1, v_2, i_2, \ldots, i_{k-1}, v_k \rangle$ from s to t in M^*, such that each i_j is either 2 or D (recall that by assumption, $n \geq 2$). Since M^* is tree-like, there is at most one reduced path from s to t in M^*. Therefore, $P_1 = P_2$. So every i_j in P_1 is D. Since P_1 is reduced, the length k of the path is 1. Thus, $(s,t) \in \mathcal{K}_D^*$. Since $(M^*, t) \not\models^* \psi$, it follows that $(M^*, s) \not\models^* D\psi$. This completes the proof. ∎

The following theorem now follows immediately from Propositions 7.2 and 7.4:

Theorem 7.5: *Assume that $\varphi \in \mathcal{L}_n^D(\Phi)$ and $\mathcal{B} \subseteq \mathcal{L}_n^D(\Phi)$. If φ is $\mathcal{B}$-consistent, then φ is $\mathcal{B}$-satisfiable.*

As we noted earlier, completeness of $S5D_n$ follows almost immediately from Theorem 7.5.

Theorem 7.6: [HM2] *$S5D_n$ is a sound and complete axiomatization with respect to S5 Kripke structures of n processes, for the language $\mathcal{L}_n^D(\Phi)$.*

If we study our completeness proof for S5D$_n$, we see that we actually proved something stronger than "if $\varphi \in \mathcal{L}_n^D(\Phi)$ is consistent, then φ is satisfiable". Our proof shows that if $\varphi \in \mathcal{L}_n^D(\Phi)$ is consistent, then φ is satisfiable in a Kripke structure of n processes. If we were willing to settle for the weaker statement "if $\varphi \in \mathcal{L}_n^D(\Phi)$ is consistent, then φ is satisfiable in a Kripke structure of $n+1$ processes", we could give a simpler proof that does not make use of Proposition 7.4 (or of Proposition 7.3, which was used in the proof of Proposition 7.4). Instead we use only the relatively simple Proposition 7.2 (and its proof) and a little bit more. From Proposition 7.2, there is a pseudo-Kripke structure $M^* = (S, \pi, \mathcal{K}_1, ..., \mathcal{K}_n, \mathcal{K}_D)$ such that φ is pseudo-satisfiable in M^*. In fact, in the construction, it turns out that $\mathcal{K}_D \subseteq \cap\{\mathcal{K}_i | 1 \leq i \leq n\}$, as we now show. Assume that $(s_V, s_W) \in \mathcal{K}_D$; we must show that $(s_V, s_W) \in \mathcal{K}_i$, for each process i. Thus, by definition of $\mathcal{K}_D$ and $\mathcal{K}_i$, we assume that $V/D \subseteq W$, and we must show that $V/K_i \subseteq W$. Let ψ be an arbitrary member of V/K_i; we must show that $\psi \in W$. Since $\psi \in V/K_i$, by definition $K_i\psi \in V$. Since $K_i\psi \Rightarrow D\psi$ is an instance of axiom KD, it follows by maximality of V that $D\psi \in V$. So, since $V/D \subseteq W$, it follows that $\psi \in W$, as desired.

The idea now is to let D play the role of the $(n+1)$st process. More formally, let $\mathcal{K}_{n+1} = \mathcal{K}_D$, and let $M = (S, \pi, \mathcal{K}_1, ..., \mathcal{K}_n, \mathcal{K}_{n+1})$. For each $\psi \in \mathcal{L}_n^D(\Phi)$, let $\widehat{\psi}$ be the result of replacing every occurrence of D in ψ by K_{n+1}. Since φ is pseudo-satisfiable in M^*, it follows immediately from the definition of pseudo-satisfaction that $\widehat{\varphi}$ is satisfiable in M. Since $\mathcal{K}_{n+1} = \mathcal{K}_D \subseteq \cap\{\mathcal{K}_i | \, 1 \leq i \leq n\}$, it follows immediately that $(s,t) \in \mathcal{K}_{n+1}$ iff $(s,t) \in \mathcal{K}_i$ for each i with $1 \leq i \leq n+1$. Therefore, for every $\psi \in \mathcal{L}_n^D(\Phi)$, and every state s of M, we have $(M,s) \models D\psi$ iff $(M,s) \models K_{n+1}\psi$. It follows easily that for every $\psi \in \mathcal{L}_n^D(\Phi)$, and every state s of M, we have $(M,s) \models \psi$ iff $(M,s) \models \widehat{\psi}$. Therefore, since $\widehat{\varphi}$ is satisfiable in M, so is φ.

7.2 The pasting condition

Lemma 7.7: *Let S be either a set of interpreted states or an S5 Kripke structure. Then S satisfies the pasting condition if and only if (S,s) satisfies every instance of axiom A6 for every state s of S.*[8]

Proof: We prove the result when S is a set of interpreted states; the proof when S is an S5 Kripke structure is almost identical and is left to the reader.

Assume first that S satisfies the pasting condition and $s \in S$, but (S,s) does not satisfy $D\neg\alpha \Rightarrow K_1\neg\alpha \vee ... \vee K_n\neg\alpha$. Let s be $(l_1, ..., l_n, \alpha')$. By assumption, $(S,s) \models D\neg\alpha$ and $(S,s) \models \neg K_i\neg\alpha$, for each process i. Since $(S,s) \models \neg K_i\neg\alpha$, we know that S contains

[8]We are assuming that the set $\{1, ..., n\}$ of processes and the set E of primitive states are fixed. So the set of "instances" of axiom A6 is $\{D\neg\alpha \Rightarrow K_1\neg\alpha \vee ... \vee K_n\neg\alpha | \, \alpha \in E\}$.

interpreted states of the form $(l_1, \cdot, ..., \cdot, \alpha)$, $(\cdot, l_2, \cdot, ..., \cdot, \alpha)$, ..., $(\cdot, ..., \cdot, l_n, \alpha)$. Since S also contains $s = (l_1, ..., l_n, \alpha')$, it follows from the pasting condition that S contains $(l_1, ..., l_n, \alpha)$, which is process equivalent to s. So $(S, s) \models \neg D\neg\alpha$, a contradiction.

Conversely, assume that (S, s) satisfies every instance of axiom A6 for every $s \in S$ but that S does not satisfy the pasting condition. Since S violates the pasting condition, S contains some $s_1 = (l_1, \cdot, ..., \cdot, \alpha)$, $s_2 = (\cdot, l_2, \cdot, ..., \cdot, \alpha)$, ..., $s_n = (\cdot, ..., \cdot, l_n, \alpha)$, $t' = (l_1, ..., l_n, \alpha')$, but not $t = (l_1, ..., l_n, \alpha)$. Since $t \notin S$, it follows that $(S, t') \models D\neg\alpha$. Since $s_i \in S$, it follows that $(S, t') \models \neg K_i\neg\alpha$, for each process i. So (S, t') does not satisfy $D\neg\alpha \Rightarrow K_1\neg\alpha \vee ... \vee K_n\neg\alpha$, an instance of axiom A6. This is a contradiction. ∎

We now prove Proposition 5.3, which we restate here for convenience.

Proposition 5.3: *Let $\mathcal{C}$ be a class of interpreted systems with n processes. If every attainable knowledge situation in every member of $\mathcal{C}$ satisfies the pasting condition, then ML_n^- (respectively ML_n) is sound with respect to $\mathcal{C}$. Conversely, if every knowledge situation that satisfies the pasting condition is attainable in some member of $\mathcal{C}$, then ML_n^- (respectively ML_n) is complete with respect to $\mathcal{C}$ for the language $\mathcal{L}_n(\Phi)$ (respectively, $\mathcal{L}_n^D(\Phi)$).*

Proof: We first consider soundness. Let $\mathcal{C}$ be a class of interpreted systems with n processes, such that every attainable knowledge situation in every member of $\mathcal{C}$ satisfies the pasting condition. By Lemma 7.7 (along with Proposition 4.2), it follows that every instance of axiom A6 is sound with respect to $\mathcal{C}$. Hence, ML_n is sound with respect to $\mathcal{C}$, as desired. To prove that ML_n^- is sound with respect to $\mathcal{C}$, we now need only show that every instance of axiom A6$'$ can be proven in ML_n. Thus, let α be a primitive state, let β be a pure knowledge formula, and let σ be $\beta \wedge K_i(\beta \Rightarrow \neg\alpha) \Rightarrow (K_1\neg\alpha \vee ... \vee K_n\neg\alpha)$. We must show that σ can be proven from ML_n. It is not hard to show that since β is a pure knowledge formula, the formula $\beta \Rightarrow D\beta$ is provable in ML_n (we leave this to the reader). An axiom of ML_n says that $K_i\varphi \Rightarrow D\varphi$ for every formula φ, and in particular $K_i(\beta \Rightarrow \neg\alpha) \Rightarrow D(\beta \Rightarrow \neg\alpha)$. So it is provable in ML_n that the left-hand side of σ implies $D\beta \wedge D(\beta \Rightarrow \neg\alpha)$, from which in turn it is provable in ML_n that $D\neg\alpha$, from which in turn it is provable in ML_n (using axiom A6) that $(K_1\neg\alpha \vee ... \vee K_n\neg\alpha)$. It follows that σ is provable in ML_n, as desired. This completes the proof of soundness.

We now consider completeness. Assume that every knowledge situation that satisfies the pasting condition is attainable in some member of $\mathcal{C}$; we must show that ML_n^- (respectively ML_n) is complete with respect to $\mathcal{C}$ for the language $\mathcal{L}_n(\Phi)$ (respectively, $\mathcal{L}_n^D(\Phi)$). We consider first the case of $\mathcal{L}_n^D(\Phi)$. Assume that $\varphi \in \mathcal{L}_n^D(\Phi)$ is consistent with respect to ML_n; we must show that φ is satisfiable in some member of $\mathcal{C}$. Let $\mathcal{B}$ be the set of all axioms of ML_n. Since φ is $\mathcal{B}$-consistent, it follows from Theorem 7.5 that φ is satisfiable in a $\mathcal{B}$-Kripke structure M. By Lemma 7.7, we know that M satisfies the pasting condition.

Let s be a state of M such that $(M, s) \models \varphi$, and let (S, s) be the knowledge situation at (M, s). Then $(S, s) \models \varphi$. Since M satisfies the pasting condition, it is easy to see that so does S. Hence, by assumption, (S, s) is attainable in some member of $\mathcal{C}$. So φ is satisfiable in some member of $\mathcal{C}$, as desired.

We have proven completeness in the $\mathcal{L}_n^D(\Phi)$ case; we now prove completeness in the $\mathcal{L}_n(\Phi)$ case. Thus, assume that every knowledge situation that satisfies the pasting condition is attainable in some member of $\mathcal{C}$; we must show that ML_n^- is complete with respect to $\mathcal{C}$ for the language $\mathcal{L}_n(\Phi)$.

Although the pasting condition is characterized (in the sense of Lemma 7.7) by axiom A6, it is not characterized by axiom A6′. Therefore, we must work harder to prove completeness in the $\mathcal{L}_n(\Phi)$ case than we did in the $\mathcal{L}_n^D(\Phi)$ case. In particular, let M^- be the analogue of the canonical structure M^c in the proof of Theorem 7.1, where "consistency" is now with respect to ML_n^-. We must look closely at M^-, and show that because of the details of how it is constructed, it obeys the pasting condition. Note that unlike before, we do not "unwind" M^- to a tree-like pseudo-Kripke structure; the purpose of the unwinding was to deal with the distributed knowledge operator, which is not in ML_n^-.

Let $MAXCON^-$ be the analogue of $MAXCON$; thus, every member of $MAXCON^-$ contains every axiom of ML_n^-. We define $\mathcal{K}_1^-, ..., \mathcal{K}_n^-$ analogously to how we defined $\mathcal{K}_1, ..., \mathcal{K}_n$ in the proof of Theorem 7.1. The states of M^- are $\{s_V \mid V \in MAXCON^-\}$; we denote this set by S^-. As before, for each formula $\psi \in \mathcal{L}_n(\Phi)$ and each $V \in MAXCON^-$, we have

Fact A. $(M^-, s_V) \models \psi$ iff $\psi \in V$.

In particular, (M^-, s_V) satisfies every instance of axiom A6′ for every $V \in MAXCON^-$. Further, as before, if φ is consistent, then φ is satisfiable in M^-. If we can show that M^- satisfies the pasting condition, then the completeness proof for $\mathcal{L}_n(\Phi)$ is completed just like the completeness proof for $\mathcal{L}_n^D(\Phi)$.

Assume that M^- is $(S^-, \pi^-, \mathcal{K}_1^-, ..., \mathcal{K}_n^-)$. Assume that $s_{V_1}, ..., s_{V_n}, s_V \in S^-$, and that for each process i, we have $\pi^-(s_{V_i}) = \alpha$ and $(s_V, s_{V_i}) \in \mathcal{K}_i^-$. To show that M^- satisfies the pasting condition, we must show that there is some $s_W \in S^-$ such that $(s_V, s_W) \in \mathcal{K}_i^-$ for each process i, and $\pi^-(s_W) = \alpha$. Let V^p be the pure knowledge formulas in V. We first show that $V^p \cup \{\alpha\}$ is consistent.

Assume not. Now a set is inconsistent iff some finite subset is inconsistent. Therefore, there is some finite subset Z of V^p such that $Z \cup \{\alpha\}$ is inconsistent. Let β be the pure knowledge formula which is the conjunction of the members of Z. So $\beta \Rightarrow \neg\alpha$ is provable. Therefore, by the knowledge generalization rule, the formula $K_i(\beta \Rightarrow \neg\alpha)$ is provable. Therefore, the formula $K_i(\beta \Rightarrow \neg\alpha)$ is in V. Since $Z \subseteq V^p \subseteq V$, we know also that $\beta \in V$. Since the formulas β and $K_i(\beta \Rightarrow \neg\alpha)$ are both in V, as is the formula $\beta \wedge K_i(\beta \Rightarrow \neg\alpha) \Rightarrow (K_1\neg\alpha \vee \ ... \ \vee K_n\neg\alpha)$ (which is an instance of axiom A6′), it follows that the formula $(K_1\neg\alpha \vee \ ... \ \vee K_n\neg\alpha)$ is also in V. So by Fact A above, $(M^-, s_V) \models (K_1\neg\alpha \vee \ ... \ \vee K_n\neg\alpha)$. But this is a contradiction, since for each process i, we have $\pi^-(s_{V_i}) = \alpha$ and $(s_V, s_{V_i}) \in \mathcal{K}_i^-$.

Therefore, $V^p \cup \{\alpha\}$ is consistent. Thus, there is $W \in MAXCON^-$ such that $V^p \cup \{\alpha\} \subseteq W$. We now show that $V/K_i \subseteq W$ for each process i. If $\varphi \in V/K_i$, then $K_i\varphi \in V$, so $K_i\varphi \in V^p$, so $K_i\varphi \in W$, so $\varphi \in W$ (because of the axiom $K_i\varphi \Rightarrow \varphi$). This shows that $V/K_i \subseteq W$ for each process i, as desired. So by construction, $(s_V, s_W) \in \mathcal{K}_i^-$ for each process i. Further, since $\alpha \in W$, it follows from Fact A above that $(M^-, s_W) \models \alpha$. Hence, $\pi^-(s_W) = \alpha$. Therefore, the pasting condition is satisfied, as desired. ∎

We are about to prove Proposition 5.4, which gives some natural conditions that guarantee the pasting condition. First, we need another definition and a lemma.

Definition 7.8: If s and t are interpreted states of an interpreted system $(\mathcal{R}, \pi)$, then s is said to be an *initial ancestor* of t if there is a run $r \in \mathcal{R}$ and a time m such that $s = \hat{r}(0)$ and $t = \hat{r}(m)$. ∎

Note that an interpreted state may have more than one initial ancestor, since it may appear in more than one run.

The next lemma follows immediately from our definitions.

Lemma 7.9: *Let s_1 and s_2 be interpreted states of an interpreted system where knowledge is cumulative. Let s_1' (respectively, s_2') be an initial ancestor of s_1 (respectively, s_2). If $s_1 \sim_i s_2$, then $s_1' \sim_i s_2'$.*

We can now prove Proposition 5.4, which we restate here for convenience.

Proposition 5.4: *If $\mathcal{I}$ is an interpreted system where (1) the environment determines the initial states, (2) knowledge is cumulative, (3) process state transitions are independent of the initial environment, and (4) the primitive propositions characterize the initial environment, then all the knowledge situations attainable in $\mathcal{I}$ satisfy the pasting condition. Conversely, if a knowledge situation satisfies the pasting condition, then it is attainable in some interpreted system $\mathcal{I}$ satisfying these four assumptions.*

Proof: Assume that (S, s) is a knowledge situation attainable in an interpreted system $\mathcal{I}$ that satisfies the four assumptions above. Since the primitive propositions characterize the initial environment, it follows immediately that *the primitive state never changes*; that is, for every run r there is a primitive state α such that $\pi(r, m) = \alpha$ for all times m. Since the environment determines the initial states and the primitive propositions characterize the initial environment, it follows immediately that *the primitive state determines the initial interpreted states*; that is, for each process i and each primitive state α, there is a subset $INIT_{i,\alpha}$ of local states such that the set of initial interpreted states with primitive state component α is $\{(l_1, \ldots, l_n, \alpha) | \; l_i \in INIT_{i,\alpha}\}$. Since process state transitions are independent of the initial environment, and the primitive propositions characterize the initial environment, it follows that *process state transitions are independent of the primitive state*, that is, whenever s and s' are initial interpreted states where $s \sim s'$, and also r is a run with initial interpreted state s (i.e., $\hat{r}(0) = s$), then there is a run r' with initial interpreted state s' such that $\hat{r}(m) \sim \widehat{r'}(m)$ for all times m.

We now show that (S, s) satisfies the pasting condition. Assume that

$s_1 = (l_1, \cdot, ..., \cdot, \alpha) \in S,$

$s_2 = (\cdot, l_2, \cdot, ..., \cdot, \alpha) \in S,$

$\dots$

$s_n = (\cdot, ..., \cdot, l_n, \alpha) \in S.$

$t' = (l_1, ..., l_n, \alpha') \in S,$

We must show that $(l_1, ..., l_n, \alpha) \in S$. Let $u' = (l'_1, ..., l'_n, \alpha')$ be an initial ancestor of t' (we have used the fact that the primitive state never changes), and let u_i be an initial ancestor of s_i, for each process i. Since knowledge is cumulative, and since $t' \sim_i s_i$, it follows from Lemma 7.9 that $u' \sim_i u_i$. Thus, $u_i = (\cdot, ..., \cdot, l'_i, \cdot, ..., \cdot, \alpha)$, where we have once again used the fact that the primitive state never changes. Since the primitive state determines the initial states, and since $u_1, ..., u_n$ are initial states, it follows that $u = (l'_1, ..., l'_n, \alpha)$ is an initial state. Since u' is an initial ancestor of t', there is a run r' and a time m such that $\widehat{r'}(0) = u'$ and $\widehat{r'}(m) = t'$. Since u and u' are process-equivalent initial states, and since process state transitions are independent of the primitive state, it follows that there is a run r with initial state u such that $t = \hat{r}(m)$ is process equivalent to $t' = \widehat{r'}(m)$. Since $t' = (l_1, ..., l_n, \alpha')$, it follows that $t = (l_1, ..., l_n, \alpha)$ (the primitive state component is α, since the primitive state component of $u = \hat{r}(0)$ is α and the primitive state never changes). We have shown that $t = (l_1, ..., l_n, \alpha)$ is an interpreted state of $\mathcal{I}$. Further, $t \in S$, since S consists of all interpreted states reachable in $\mathcal{I}$ from s (clearly t is reachable in $\mathcal{I}$ from s, since t is obviously reachable in $\mathcal{I}$ from t', and t' is reachable in $\mathcal{I}$ from s because $t' \in S$). This was to be shown.

Conversely, let (S, s) be a knowledge situation that satisfies the pasting condition. We now define an interpreted system $\mathcal{I}$ satisfying the four assumptions of Proposition 5.4 such that (S, s) is a knowledge situation attainable in $\mathcal{I}$.

If $t = (l_1, ..., l_n, \alpha)$ is an interpreted state, then we say that the primitive state α and the local states l_i *appear* in t. For each primitive state α, we define a distinct environment state e_α. The set of environment states of the interpreted system $\mathcal{I}$ we are constructing contains precisely those environment states e_α such that the primitive state α appears in some member of S. Let L_0 be the set of local states that appear in some member of S. Let L_1 and L_2 be sets with the same cardinality as L_0, such that L_0, L_1, and L_2 are pairwise disjoint. Define the set L of local states of the interpreted system $\mathcal{I}$ to be $L_0 \cup L_1 \cup L_2$. Let f_1 (respectively, f_2) be a one-to-one map from L_0 onto L_1 (respectively, onto L_2). Let $T = \{(e_\alpha, f_1(l_1), ..., f_1(l_n)) | \; (l_1, ..., l_n, \alpha) \in S\}$. Let $T' = \{(e, l_1, ..., l_n) | \; (e, l_1, \cdot, ..., \cdot) \in T$ and $(e, \cdot, l_2, \cdot, ..., \cdot) \in T$ and ... and $(e, \cdot, ..., \cdot, l_n) \in T\}$. Intuitively, T' is the result of closing T off under Cartesian product for each fixed environment state e. The set of initial states of our interpreted system $\mathcal{I}$ is precisely T'. It follows easily that in $\mathcal{I}$, the environment determines the initial states.

We now define the set $\mathcal{R}$ of runs in $\mathcal{I}$. There is one run r_w for each $w \in T'$, where $r_w(0) = w$. It is clear that every member of T' is of the form $(e, f_1(l_1), ..., f_1(l_n))$. Assume that $w = (e, f_1(l_1), ..., f_1(l_n)) \in T'$. Let us say that w is of *type 1* if $(l_1, ..., l_n, \beta) \in S$ for some β (where possibly, but not necessarily, $e = e_\beta$). Otherwise, we say that w is of *type 2*. If w is of type 1, then let $r_w(m) = (e, l_1, ..., l_n)$ for each $m \geq 1$. If w is of type 2, then let $r_w(m) = (e, f_2(l_1), ..., f_2(l_n))$ for each $m \geq 1$.

We define π in the obvious way: if the environment component of the global state $r(m)$ is e_α, then we let $\pi(r, m) = \alpha$. It is easy to see that the primitive propositions characterize the initial environment.

To show that knowledge is cumulative, we must show that the local state of each process uniquely determines its history. If $l \in L_0$ is the local state, then it is easy to see that the history is $\langle f_1(l), l\rangle$. If $l \in L_1$ is the local state, then l is of the form $f_1(l')$, and the history is simply $\langle l\rangle$. If $l \in L_2$ is the local state, then l is of the form $f_2(l')$, and the history is $\langle f_1(l'), f_2(l')\rangle$. Thus, knowledge is cumulative.

We now show that process state transitions are independent of the initial environment. Assume that w and w' are process-equivalent initial states. Like all initial states in our interpreted system $\mathcal{I}$, the initial state w is of the form $(e, f_1(l_1)..., f_1(l_n))$. Since $w \sim w'$, there is e' such that $w' = (e', f_1(l_1)..., f_1(l_n))$. It is clear that w and w' are either both of type 1 or both of type 2. In either case, it is easy to see that $r_w(m) \sim r_{w'}(m)$ for every time m. Since r_w is the only run with initial state w, and since $r_{w'}$ has initial state w', it follows immediately that process state transitions are independent of the initial environment.

We have shown that $\mathcal{I}$ satisfies each of the four assumptions of Proposition 5.4. We conclude by showing that (S, s) is a knowledge situation attainable in $\mathcal{I}$.

First, every member of S is an interpreted state of $\mathcal{I}$. This is because if $t = (l_1, ..., l_n, \alpha) \in S$, then $w = (e_\alpha, f_1(l_1), ..., f_1(l_n))$ is of type 1, and so $\widehat{r_w}(1) = t$. Therefore, we will be done if we can show that the only interpreted states that are reachable in $\mathcal{I}$ from a member of S are in S. Assume that $t = (l_1, ..., l_n, \alpha)$ is an interpreted state that is reachable in $\mathcal{I}$ from a member of S; we must show that $t \in S$. We must have $t = \widehat{r_w}(m)$ for some run r_w of $\mathcal{R}$ and some time m. By definition of π, we know that $r_w(m) = (e_\alpha, l_1, ..., l_n)$. Since L_1 and L_2 are disjoint from L_0, and since t is reachable in $\mathcal{I}$ from a member of S, it follows from our construction that w is of type 1 and $w = (e_\alpha, f_1(l_1)..., f_1(l_n))$. Since w is of type 1, there is α' such that $t' = (l_1, ..., l_n, \alpha') \in S$. Since $w \in T'$, it follows by construction of T' that for each process i, there is some global state $s'_i = (e_\alpha, \cdot, ..., \cdot, f_1(l_i), \cdot, ..., \cdot) \in T$. So, by construction of T, we know that for each process i, there is some $s_i = (\cdot, ..., \cdot, l_i, \cdot, ..., \cdot, \alpha) \in S$. Since $s_1, ..., s_n, t' \in S$, it follows from the pasting condition that $t = (l_1, ..., l_n, \alpha) \in S$, as desired. This was to be shown. ∎

We note that the interpreted system $\mathcal{I}$ constructed in the above proof is deterministic. Further, by an easy modification of the proof, we could have made $\mathcal{I}$ synchronous. Thus, the converse of Proposition 5.4 can be strengthened to say that if a knowledge situation satisfies the pasting condition, then it is attainable in some deterministic, synchronous interpreted system $\mathcal{I}$ satisfying the four assumptions of Proposition 5.4.

We now prove Theorem 5.6, which again we restate for convenience.

Theorem 5.6: *Let $\mathcal{A}$ be a proper subset of the four conditions of Theorem 5.5 (or of Proposition 5.4), and let $\mathcal{C}$ be the class of all interpreted systems that satisfy the conditions $\mathcal{A}$. Then each knowledge situation is attainable in a member of $\mathcal{C}$. Thus, $S5_n$ (respectively $S5D_n$) is a sound and complete axiomatization with respect to $\mathcal{C}$ for the language $\mathcal{L}_n(\Phi)$ (respectively, $\mathcal{L}_n^D(\Phi)$).*

Proof: It is clear that we need only consider the four possible subsets $\mathcal{A}$ consisting of precisely three of the four conditions. Let (S, s) be an arbitrary knowledge situation. We shall show that for each of the four candidates for $\mathcal{A}$, it is the case that (S, s) is attainable in some interpreted system $\mathcal{I}$ satisfying $\mathcal{A}$. In each case, the set E of environment states is $\{e_\alpha |$ the primitive state α appears in some member of $S\}$.

Case 1: Knowledge is cumulative, process state transitions are independent of the initial environment, and the primitive propositions characterize the initial environment. For each $t = (l_1, ..., l_n, \alpha) \in S$, there is one run r_t, where $r_t(m) = (e_\alpha, l_1, ..., l_n)$ for all times m. If the primitive state component of t is α, then let $\pi(r_t, m) = \alpha$ for all times m. It is easy to see that the resulting interpreted system $\mathcal{I}$ satisfies each of the three conditions of Case 1, and (S, s) is attainable in $\mathcal{I}$.

Case 2: The environment determines the initial states, process state transitions are independent of the initial environment, and the primitive propositions characterize the initial environment. Let L_0 be the set of local states that appear in some member of S. Let L_1 be a set which is disjoint from L_0 and which contains a new local state s_α for every primitive state α that appears in some member of S. The set L of local states is $L_0 \cup L_1$. There is one run r_t for each $t \in S$. If $t = (l_1, ..., l_n, \alpha)$, then let $r_t(0) = (e_\alpha, s_\alpha, ..., s_\alpha)$, and let $r_t(m) = (e_\alpha, l_1, ..., l_n)$ for $m \geq 1$. As before, if the primitive state component of t is α, then let $\pi(r_t, m) = \alpha$ for all times m.

Again, the primitive propositions characterize the initial environment. The environment determines the initial states, since no two distinct initial states have the same environment component (thus, using our terminology, the environment even uniquely determines the initial state). Process state transitions are independent of the initial environment, since no two distinct initial states are process equivalent. Finally, (S, s) is a knowledge situation attainable in $\mathcal{I}$, since every member of S is an interpreted state of $\mathcal{I}$, and the only interpreted states that are reachable in $\mathcal{I}$ from a member of S are members of S.

Case 3: The environment determines the initial states, knowledge is cumulative, and the primitive propositions characterize the initial environment. Let $L_0, L_1, L_2, L, f_1, f_2, T, T'$ be as in the proof of Proposition 5.4. As in the proof of Proposition 5.4, the set of initial states of our interpreted system $\mathcal{I}$ is precisely T', and as before, the environment determines the initial states.

We now define the set $\mathcal{R}$ of runs in $\mathcal{I}$. There is one run r_w for each $w \in T'$, where $r_w(0) = w$. It is clear that every member w of T' is of the form $(e, f_1(l_1), ..., f_1(l_n))$. Assume that $w = (e, f_1(l_1), ..., f_1(l_n)) \in T'$. If $w \in T$, then $r_w(m) = (e, l_1, ..., l_n)$, for

$m \geq 1$. If $w \notin T$, then $r_w(m) = (e, f_2(l_1), ..., f_2(l_n))$, for $m \geq 1$. It is perhaps useful to comment on how this construction differs from that of Proposition 5.4. Assume that $e = e_\alpha$. In the construction of Proposition 5.4, for $r_w(m)$ to be $(e, l_1, ..., l_n)$, rather than $r_w(m)$ being $(e, f_2(l_1), ..., f_2(l_n))$, for $m \geq 1$, it was necessary that w be of type 1 (and so for $(l_1, ..., l_n, \beta) \in S$ for some β, not necessarily $\beta = \alpha$). However, in the current construction, for $r_w(m)$ to be $(e, l_1, ..., l_n)$, rather than $r_w(m)$ being $(e, f_2(l_1), ..., f_2(l_n))$, for $m \geq 1$, it is necessary that w be in T (and so for $(l_1, ..., l_n, \alpha) \in S$). In particular, unlike the construction of Proposition 5.4, it turns out that process state transitions will not necessarily be independent of the initial environment.

By the identical argument to that in the proof of Proposition 5.4, knowledge is cumulative. We define π as in the proof of Proposition 5.4, and again it is clear that the primitive propositions characterize the initial environment.

We now show that (S, s) is a knowledge situation attainable in $\mathcal{I}$. First, every member of S is an interpreted state of $\mathcal{I}$. This is because if $t = (l_1, ..., l_n, \alpha) \in S$, then $\widehat{r_w}(1) = t$, where $w = (e_\alpha, f_1(l_1), ..., f_1(l_n))$. Since L_1 and L_2 are disjoint from L_0, it follows from our construction that the only interpreted states that are reachable in $\mathcal{I}$ from a member of S are members of S. So (S, s) is a knowledge situation attainable in $\mathcal{I}$, as desired.

Case 4: The environment determines the initial states, knowledge is cumulative, and process state transitions are independent of the initial environment. For each $t \in S$, we define a new, distinct environment state e_t. For each $t \in S$, there is one run r_t, where $r_t(m) = (e_t, l_1, ..., l_n)$ for all times m. If the primitive state component of t is α, then let $\pi(r_t, m) = \alpha$ for all times m. It is easy to see that the resulting interpreted system $\mathcal{I}$ satisfies each of the three conditions of Case 4, and (S, s) is attainable in $\mathcal{I}$. ∎

7.3 Equivalence of S5D$_n$ + A8 and S5D$_n$ + A8*

As we promised in the body of the paper, we now prove that S5D$_n$ + A8 is equivalent to S5D$_n$ + A8*. Recall that A8 and A8* are as follows, where φ is an arbitrary formula and where α is a primitive state:

A8. $\varphi \Rightarrow D\varphi$.

A8*. $\alpha \Rightarrow D\alpha$.

Proposition 7.10: *S5D$_n$ + A8 is equivalent to S5D$_n$ + A8*.*

Proof: Since A8* is a special case of A8, it is clear that every formula that can be proven in S5D$_n$ + A8* can also be proven in S5D$_n$ + A8. We now prove the converse. We shall show, by induction on the structure of formulas φ, that $\varphi \Rightarrow D\varphi$ and $(\neg\varphi) \Rightarrow D\neg\varphi$ are each provable in S5D$_n$ + A8*.

1. φ *is propositional*: Since by assumption, there are only a finite number of primitive propositions, it follows that φ is logically equivalent to a formula $\alpha_1 \vee ... \vee \alpha_k$, where each α_j is a primitive state. Since $\alpha_j \Rightarrow D\alpha_j$ is an axiom for each j, it follows by propositional reasoning (that is, using axiom A1 and modus ponens) that the following formula is provable:

$$(\alpha_1 \vee ... \vee \alpha_k) \Rightarrow (D\alpha_1 \vee ... \vee D\alpha_k) \tag{4}$$

Using the axiom $D\varphi_1 \wedge D(\varphi_1 \Rightarrow \varphi_2) \Rightarrow D\varphi_2$, where φ_1 is α_j and φ_2 is $\alpha_1 \vee ... \vee \alpha_k$, we see by propositional reasoning that the following formula is provable for each j $(1 \leq j \leq k)$:

$$D\alpha_j \Rightarrow D(\alpha_1 \vee ... \vee \alpha_k) \tag{5}$$

(we also used the fact that $\varphi_1 \Rightarrow \varphi_2$ is a propositional tautology, and so provable, and hence $D(\varphi_1 \Rightarrow \varphi_2)$ is provable by distributed knowledge generalization). Putting together (4) and the k instances of (5), we see by propositional reasoning that the following formula is provable:

$$(\alpha_1 \vee ... \vee \alpha_k) \Rightarrow D(\alpha_1 \vee ... \vee \alpha_k) \tag{6}$$

Since φ is logically equivalent to $\alpha_1 \vee ... \vee \alpha_k$, the following formula is provable:

$$\varphi \Leftrightarrow (\alpha_1 \vee ... \vee \alpha_k) \tag{7}$$

From (6) and (7), we see that the formula $\varphi \Rightarrow D\varphi$ is provable, as desired. Since $\neg\varphi$ is also a propositional formula, an identical argument shows that $(\neg\varphi) \Rightarrow D\neg\varphi$ is provable. This concludes the case where φ is propositional.

2. φ *is* $\neg\psi$: This case follows immediately from our inductive assumption.

3. φ *is* $\varphi_1 \wedge \varphi_2$:

   (a) By inductive assumption, $\varphi_1 \Rightarrow D\varphi_1$ and $\varphi_2 \Rightarrow D\varphi_2$ are each provable. Also, we see from the axioms that $(D\varphi_1 \wedge D\varphi_2) \Rightarrow D(\varphi_1 \wedge \varphi_2)$ is provable. By propositional reasoning, we then see that $(\varphi_1 \wedge \varphi_2) \Rightarrow D(\varphi_1 \wedge \varphi_2)$ is provable, that is, $\varphi \Rightarrow D\varphi$ is provable.

   (b) We now must show that $\neg\varphi \Rightarrow D\neg\varphi$ is provable. It is provable that $\neg\varphi$ implies $\neg\varphi_1 \vee \neg\varphi_2$. From $\neg\varphi_1 \vee \neg\varphi_2$ we can infer that $D\neg\varphi_1 \vee D\neg\varphi_2$ (since by inductive assumption $\neg\varphi_j \Rightarrow D\neg\varphi_j$ for $j = 1, 2$). From $D\neg\varphi_1 \vee D\neg\varphi_2$ we can infer that $D(\neg\varphi_1 \vee \neg\varphi_2)$ (since for $j = 1, 2$, the fact that $\neg\varphi_j \Rightarrow \neg\varphi_1 \vee \neg\varphi_2$ is provable shows that $D\neg\varphi_j \Rightarrow D(\neg\varphi_1 \vee \neg\varphi_2)$ is provable). From $D(\neg\varphi_1 \vee \neg\varphi_2)$ we can infer that $D\neg\varphi$.

4. φ *is* $K_i\psi$:

(a) It is provable that $K_i\psi$ implies $K_iK_i\psi$, from which $DK_i\psi$ is provable. Hence, $K_i\psi \Rightarrow DK_i\psi$ is provable.

(b) It is provable that $\neg K_i\psi$ implies $K_i\neg K_i\psi$, from which $D\neg K_i\psi$ is provable. Hence, $(\neg K_i\psi) \Rightarrow D(\neg K_i\psi)$ is provable.

5. *φ is $D\psi$*: $D\psi \Rightarrow DD\psi$ is one of the axioms, as is $(\neg D\psi) \Rightarrow D\neg D\psi$. ∎

7.4 The process component uniquely determines the primitive state

In this subsection, we prove the results stated earlier that deal with the situation where the process component uniquely determines the primitive state. We first restate and prove Proposition 5.8.

Proposition 5.8: *If $\mathcal{I}$ is an interpreted system where the primitive propositions are determined by the current states of the processes, then in every knowledge situation attainable in $\mathcal{I}$, the process component uniquely determines the primitive state. Conversely, each knowledge situation where the process component uniquely determines the primitive state is attainable in some interpreted system where the primitive propositions are determined by the current states of the processes.*

Proof: First, let $\mathcal{I}$ be an interpreted system where the primitive propositions are determined by the current states of the processes, and let (S, s) be a knowledge situation attainable in $\mathcal{I}$. Let t and t' be members of S where $t \sim t'$. Since the primitive propositions are determined by the current states of the processes, it follows immediately that $t = t'$. Hence, the process component uniquely determines the primitive state.

Conversely, let (S, s) be a knowledge situation where the process component uniquely determines the primitive state. Let the interpreted system $\mathcal{I}$ be constructed as in Case 1 of the proof of Theorem 5.6. Because the process component uniquely determines the primitive state in (S, s), it follows easily that the primitive propositions are determined by the current states of the processes in $\mathcal{I}$. ∎

We claimed in the body of the paper that Proposition 5.8 and Theorem 5.10 both hold when we replace every occurrence of "the primitive propositions are determined by the current states of the processes" by "the primitive propositions are determined by the initial states of the processes and knowledge is cumulative". We first show that Proposition 5.8 holds after this replacement.

Proposition 7.11: *If $\mathcal{I}$ is an interpreted system where the primitive propositions are determined by the initial states of the processes and knowledge is cumulative, then in every knowledge situation attainable in $\mathcal{I}$, the process component uniquely determines the primitive state. Conversely, each knowledge situation where the process component uniquely determines the primitive state is attainable in some interpreted system where the* *primitive propositions are determined by the initial states of the processes and knowledge is cumulative.*

Proof: The first part of the proposition follows from the first part of Proposition 5.8 and the fact that if the primitive propositions are determined by the initial states of the processes and knowledge is cumulative, then the primitive propositions are determined by the current states of the processes. The second part follows from the proof of the second part of Proposition 5.8 and the fact that in the interpreted system $\mathcal{I}$ constructed there, the primitive propositions are determined by the initial states of the processes and knowledge is cumulative. ∎

Theorem 5.11 is the result of replacing every occurrence of "the primitive propositions are determined by the current states of the processes" by "the primitive propositions are determined by the initial states of the processes and knowledge is cumulative". Theorem 5.11 follows immediately from Propositions 5.9 and 7.11.

Before we prove the proposition which relates axiom A8 to interpreted systems where the process component uniquely determines the primitive state, we need a lemma that is completely analogous to Lemma 7.7. The definition of "the process component uniquely determines the primitive state in a Kripke structure" is obtained by modifying the definition of "the process component uniquely determines the primitive state in a set S of interpreted states" in the obvious manner.

Lemma 7.12: *Let S be either a set of interpreted states or an S5 Kripke structure. The process component uniquely determines the primitive state in S if and only if (S, s) satisfies every instance of axiom A8 for every state s of S.*

Proof: We shall prove the result when S is a set of interpreted states; the proof when S is an S5 Kripke structure is almost identical.

Because of Lemma 7.10, it follows easily that (S, s) satisfies every instance of axiom A8 if and only if (S, s) satisfies every instance of axiom A8*. So we need only show that the process component uniquely determines the primitive state in S if and only if (S, s) satisfies every instance of axiom A8* for every state s of S.

Assume first that the process component does not uniquely determine the primitive state in S. So, there are members s and s' of S that are process equivalent but that have distinct primitive state components. Let α be the primitive state component of s. Then $(S, s) \models \alpha$, but $(S, s) \models \neg D\alpha$. So (S, s) does not satisfy the instance $\alpha \Rightarrow D\alpha$ of axiom A8. The converse is very similar. ∎

We now prove Proposition 5.9, which says the following:

Proposition 5.9: *Let $\mathcal{C}$ be a class of interpreted systems with $n \geq 2$ processes. If for every attainable knowledge situation in every member of $\mathcal{C}$, the process component uniquely determines the primitive state, then $S5_n$ (respectively, $S5D_n$ + A8) is sound with respect* *to $\mathcal{C}$. Conversely, if every knowledge situation where the process component uniquely determines the primitive state is attainable in some member of $\mathcal{C}$, then S5$_n$ (respectively, S5D$_n$ + A8) is complete with respect to $\mathcal{C}$ for the language $\mathcal{L}_n(\Phi)$ (respectively, $\mathcal{L}_n^D(\Phi)$).*

Proof: Lemma 7.12 can be used to prove soundness and completeness of S5D$_n$ + A8 in the same way as Lemma 7.7 was used to prove soundness and completeness of ML$_n$ (in the proof of Proposition 5.3).

We now show completeness of S5$_n$ in the language $\mathcal{L}_n(\Phi)$ for $\mathcal{C}$ (we already know S5$_n$ is sound, by Proposition 5.1). Let φ be a formula of $\mathcal{L}_n(\Phi)$ that is consistent with S5$_n$. To show completeness, we need only show that φ is satisfiable in some member of $\mathcal{C}$. Since φ is consistent with S5$_n$, we know by Theorem 7.1 that φ is satisfiable. It then follows immediately from Proposition 7.3 that φ is satisfiable in an S5 Kripke structure $M = (S, \pi, \mathcal{K}_1, ..., \mathcal{K}_n)$ that is tree-like. Let (S, s) be a knowledge situation attainable in M that satisfies φ. As we noted earlier, one consequence of M being tree-like is that if t_1 and t_2 are distinct states of M such that $(t_1, t_2) \in \mathcal{K}_i$ and $(t_1, t_2) \in \mathcal{K}_j$, then $i = j$. So, since $n \geq 2$, it follows that if two states in S are process equivalent, then they are identical. Therefore, in S the process component uniquely determines the primitive state. So by assumption, (S, s) is attainable in some member of $\mathcal{C}$. Therefore, φ is satisfiable in some member of $\mathcal{C}$. This was to be shown. ∎

7.5 The environment uniquely determines the initial state

In Proposition 5.4 and Theorem 5.5 one of our assumptions is that the environment determines the initial states. We claimed in Section 5 that if we make the stronger assumption that the environment *uniquely* determines the initial state, then a smaller set of knowledge situations is attainable, and again knowledge has extra properties. We now discuss and prove this claim.

Thus, we are interested in considering interpreted systems where (1) the environment uniquely determines the initial state, (2) knowledge is cumulative, (3) process state transitions are independent of the initial environment, and (4) the primitive propositions characterize the initial environment. As before, our first step is to get a semantic characterization of the attainable knowledge situations under these assumptions.

Definition 7.13: A set S of interpreted states satisfies the *matching condition* if whenever

1. $s_1, ..., s_n, s'_1, ..., s'_n \in S$,
2. α is the primitive state component of s_j, for $j = 1, ..., n$,
3. α' is the primitive state component of s'_j, for $j = 1, ..., n$, and
4. $s_j \sim_j s'_j$, for $j = 1, ..., n$,

then for each $t' \in S$ with primitive state component α', there is $t \in S$ with primitive state component α such that $t \sim t'$. The knowledge situation (S, s) is said to satisfy the matching condition if S does. ∎

Thus, S satisfies the matching condition if whenever

$s_1 = (l_1, \cdot, ..., \cdot, \alpha) \in S,$

$s_2 = (\cdot, l_2, \cdot, ..., \cdot, \alpha) \in S,$

...

$s_n = (\cdot, ..., \cdot, l_n, \alpha) \in S,$

$s'_1 = (l_1, \cdot, ..., \cdot, \alpha') \in S,$

$s'_2 = (\cdot, l_2, \cdot, ..., \cdot, \alpha') \in S,$

...

$s'_n = (\cdot, ..., \cdot, l_n, \alpha') \in S,$

$t' = (l'_1, ..., l'_n, \alpha') \in S,$

then $t = (l'_1, ..., l'_n, \alpha) \in S$. As we shall see, the matching condition implies the pasting condition.

Similarly, we can modify the definition in the obvious way to define what it means for a Kripke structure to satisfy the matching condition.

Proposition 7.14: *If $\mathcal{I}$ is an interpreted system where (1) the environment uniquely determines the initial state, (2) knowledge is cumulative, (3) process state transitions are independent of the initial environment, and (4) the primitive propositions characterize the initial environment, then all the knowledge situations attainable in $\mathcal{I}$ satisfy the matching condition. Conversely, if a knowledge situation satisfies the matching condition, then it is attainable in some interpreted system satisfying these four assumptions.*

Proof: Assume first that (S, s) is a knowledge situation attainable in an interpreted system $\mathcal{I} = (\mathcal{R}, \pi)$ that satisfies the four assumptions; we must show that S satisfies the matching condition. As in the proof of Proposition 5.4, we see that the primitive state never changes, and process state transitions are independent of the primitive state. Furthermore, since the environment uniquely determines the initial state and the primitive propositions characterize the initial environment, it follows that *the primitive state uniquely determines the initial interpreted state*, that is, there are no two distinct initial interpreted states with the same primitive state component.

Let $s_1, ..., s_n, s'_1, ..., s'_n, t' \in S$ be as in Definition 7.13. Let w be an initial ancestor of s_1. Since the primitive state never changes and α is the primitive state component of s_1, it follows that w is an initial interpreted state of $\mathcal{I}$ with primitive state component α. Since the primitive state uniquely determines the initial interpreted state, w is the unique initial interpreted state of $\mathcal{I}$ with primitive state component α. Similarly, there is a unique initial interpreted state w' with primitive state component α'. So w (respectively, w') is the unique initial ancestor of $s_1, ..., s_n$ (respectively, $s'_1, ..., s'_n$). Since knowledge is cumulative, and since $s_i \sim_i s'_i$, it follows from Lemma 7.9 that $w \sim_i w'$ for each process i. So $w \sim w'$. Since t' has primitive state component α', its (unique) initial ancestor is w'. Since process state transitions are independent of the primitive state, and since $w \sim w'$, it follows that there is an interpreted state t with initial ancestor w such that $t \sim t'$. Since (S, s) is a knowledge situation of $\mathcal{I}$ and since $t' \in S$, it follows that $t \in S$. Also, t has α as its primitive state component (since w does, and since the primitive state never changes). Hence, we have constructed t with all the desired properties.

Conversely, let (S, s) be a knowledge situation that satisfies the matching condition. We now define an interpreted system $\mathcal{I}$ satisfying the four assumptions, such that (S, s) is a knowledge situation attainable in $\mathcal{I}$.

For each primitive state α, we define a distinct environment state e_α. As before, the set of environment states of the interpreted system $\mathcal{I}$ we are constructing contains precisely those environment states e_α such that the primitive state α appears in some member of S. Let L_0 be the set of local states that appear in some member of S. For each process i and each primitive state α that appears in S, we define new, distinct elements $t_{i,\alpha}$. For each i, we form certain equivalence classes of the $t_{i,\alpha}$'s, by putting $t_{i,\alpha}$ and $t_{i,\alpha'}$ in the same equivalence class iff S contains interpreted states s_i and s'_i such that $s_i \sim_i s'_i$ and the primitive state of s_i (respectively, s'_i) is α (respectively, α'). We denote the equivalence class of $t_{i,\alpha}$ by $[t_{i,\alpha}]$. We let $L_1 = \{[t_{i,\alpha}] | i$ is a process and α is a primitive state appearing in $S\}$. The set L of local states is then $L_0 \cup L_1$.

We now define the set $\mathcal{R}$ of runs in $\mathcal{I}$. There is one run r_w for each $w \in S$. Assume that $w = (l_1, ..., l_n, \alpha)$. Then $r_w(0) = (e_\alpha, [t_{1,\alpha}], ..., [t_{n,\alpha}])$, and $r_w(m) = (e_\alpha, l_1, ..., l_n)$ for $m \geq 1$.

We define π in the obvious way, as before: if the environment component of the global state $r(m)$ is e_α, then we let $\pi(r, m) = \alpha$. Again, the primitive propositions characterize the initial environment.

By construction, the primitive state uniquely determines the initial interpreted state, and so (since the primitive propositions characterize the initial environment), the environment uniquely determines the initial state.

We now show that knowledge is cumulative. Assume that $r_w(m) \sim_i r_{w'}(m')$. We must show that process i's history in run r_w up to time m is the same as process i's history in run $r_{w'}$ up to time m'. If either m or m' is 0, then by construction we know that both are, and the history is simply $\langle r_w(0) \rangle$. So assume that neither m nor m' is 0. Let $w = (l_1, ..., l_n, \alpha)$ and $w' = (l'_1, ..., l'_n, \alpha')$ (where we know that $l_i = l'_i$). Then process i's history in run r_w up to time m is $\langle [t_{i,\alpha}], l_i \rangle$, and process i's history in run r'_w up to time m' is $\langle [t_{i,\alpha'}], l'_i \rangle$. Since $w \sim_i w'$ and the primitive state of w (respectively, w') is α (respectively, α'), it follows from our definition of the equivalence classes that $[t_{i,\alpha}] = [t_{i,\alpha'}]$. Since also $l_i = l'_i$, it follows that process i's history in run r_w up to time m is the same as process i's history in run $r_{w'}$ up to time m'. Thus, knowledge is cumulative.

We now show that process state transitions are independent of the initial environment. Thus, assume that u and u' are initial states where $u \sim u'$ and that r is a run with initial state u; we must show that there is a run r' with initial state u' such that $r(m) \sim r'(m)$ for all times $m \geq 1$. Since $r'(m) = r'(1)$ for each run r' and for each $m' \geq 1$, we need only show that there is a run r' with initial state u' such that $r(1) \sim r'(1)$. If $u = u'$, then let $r' = r$. So assume that $u \neq u'$. Assume that the environment state of u (respectively, u') is e_α (respectively, $e_{\alpha'}$). Since $u \sim u'$, it follows that for each process i, we have $[t_{i,\alpha}] = [t_{i,\alpha'}]$. That is, $t_{i,\alpha}$ and $t_{i,\alpha'}$ are in the same equivalence class for each process i. By definition of the equivalence classes, this means that for each process i, the set S contains primitive states s_i and s'_i such that $s_i \sim_i s'_i$ and the primitive state of s_i (respectively, s'_i) is α (respectively, α'). Since the primitive state component of $r(1)$ is α and since S satisfies the matching condition, it follows that there is some $w \in S$ with primitive state component α' such that $w \sim r(1)$. Then r_w is the run r' that we have been seeking, since $r_w(1) = w$.

Finally, (S, s) is a knowledge situation attainable in $\mathcal{I}$, since every member of S is an interpreted state of $\mathcal{I}$, and the only interpreted states that are reachable in $\mathcal{I}$ from a member of S are members of S. ∎

We now define a condition that, taken together with the pasting condition, is equivalent to the matching condition.

Definition 7.15: A set S of interpreted states satisfies the *weak matching condition* if whenever

1. $s_1, ..., s_n, s'_1, ..., s'_n \in S$,
2. α is the primitive state component of s_j, for $j = 1, ..., n$,
3. α' is the primitive state component of s'_j, for $j = 1, ..., n$, and
4. $s_j \sim_j s'_j$, for $j = 1, ..., n$,

then for each $t' \in S$ with primitive state component α' and for each process i, there is some $t \in S$ with primitive state component α such that $t \sim_i t'$. The knowledge situation (S, s) is said to satisfy the weak matching condition if S does. ∎

Thus, S satisfies the weak matching condition if whenever

$s_1 = (l_1, \cdot, ..., \cdot, \alpha) \in S$,

$s_2 = (\cdot, l_2, \cdot, ..., \cdot, \alpha) \in S$,

...

$s_n = (\cdot, ..., \cdot, l_n, \alpha) \in S$,

$s'_1 = (l_1, \cdot, ..., \cdot, \alpha') \in S$,

$s'_2 = (\cdot, l_2, \cdot, ..., \cdot, \alpha') \in S$,

...

$s'_n = (\cdot, ..., \cdot, l_n, \alpha') \in S$,

$t'_i = (..., l'_i, ..., \alpha') \in S$,

then $t_i = (..., l'_i, ..., \alpha) \in S$.

Similarly, we can modify the definition in the obvious way to define what it means for a Kripke structure to satisfy the weak matching condition.

Lemma 7.16: *The matching condition is equivalent to the pasting condition along with the weak matching condition.*

Proof: Let S be a set of interpreted states that satisfies the matching condition. It is clear that S satisfies the weak matching condition. We now show that S satisfies the pasting condition. In the definition of the matching condition, let

$s_1 = (l_1, \cdot, ..., \cdot, \alpha)$,

$s_2 = (\cdot, l_2, \cdot, ..., \cdot, \alpha)$,

...

$s_n = (\cdot, ..., \cdot, l_n, \alpha)$,

$s'_j = t' = (l_1, ..., l_n, \alpha')$ for $j = 1, ..., n$.

The matching condition then tells us that if $s_1, ..., s_n, t'$ as above are all in S, then so is $t = (l_1, ..., l_n, \alpha)$. Therefore, the pasting condition is satisfied.

Conversely, assume that the weak matching condition and the pasting condition are both satisfied. Assume that

$s_1 = (l_1, \cdot, ..., \cdot, \alpha) \in S$,

$s_2 = (\cdot, l_2, \cdot, ..., \cdot, \alpha) \in S$,

...

$s_n = (\cdot, ..., \cdot, l_n, \alpha) \in S,$

$s'_1 = (l_1, \cdot, ..., \cdot, \alpha') \in S,$

$s'_2 = (\cdot, l_2, \cdot, ..., \cdot, \alpha') \in S,$

...

$s'_n = (\cdot, ..., \cdot, l_n, \alpha') \in S,$

$t' = (l'_1, ..., l'_n, \alpha') \in S.$

By the weak matching condition, we know that for each process i there is some $t_i \in S$ where $t_i = (..., l'_i, ..., \alpha)$. Since $t_1, ..., t_n, t' \in S$, it follows from the pasting condition that $t = (l'_1, ..., l'_n, \alpha) \in S$. Therefore, the matching condition is satisfied, as desired. ∎

Assume that the processes are 1,...,n. Let $E\varphi$ be a shorthand for $K_1\varphi \wedge ... \wedge K_n\varphi$, that is, "Every process knows φ". Let $E^0\varphi$ be φ, and let $E^k\varphi$ abbreviate $EE^{k-1}\varphi$ for $k \geq 1$. φ is *common knowledge* (written $C\varphi$) if $E^k\varphi$ holds for every k. If S is a set of interpreted states, and if $s, t \in S$, then we say that t *is distance at most k from s in S* if there are $s_1, ..., s_{k+1} \in S$ and processes $i_1, ..., i_k$ where

1. $s_1 = s$,
2. $s_{k+1} = t$, and
3. $s_j \sim_j s_{j+1}$, for $1 \leq j \leq k$.

The following simple lemma, whose proof is left to the reader, will be useful later.

Lemma 7.17: *Let S be a set of interpreted states. $(S, s) \models E^k\varphi$ iff $(S, t) \models \varphi$ for every t at most distance k from s in S.*

Consider the following axiom (which involves common knowledge), where α and α' are primitive states, and $\{1, ..., n\}$ is the set of processes.

A9*. $(\alpha' \wedge K_i \neg\alpha) \Rightarrow \bigvee_{j=1}^{n} C(\alpha' \Rightarrow K_j \neg\alpha)$

Intuitively, (the contrapositive of) A9* says that the weak matching condition holds. This is because the contrapositive says that if there are states s'_j (for $1 \leq j \leq n$) where $\alpha' \Rightarrow K_j\neg\alpha$ fails (that is, where α' holds and where there is s_j such that α holds and such that $s_j \sim_j s'_j$, then it is not possible for there to be a state t' where α' holds and where there is no t_i such that α holds and $t_i \sim_i t'$. Since we do not have common knowledge in the languages $\mathcal{L}_n(\Phi)$ and $\mathcal{L}_n^D(\Phi)$ that we are considering, we make use of the following axiom instead of A9*, where α and α' are primitive states, $\{1, ..., n\}$ is the set of processes, and k is a positive integer:

A9. $(\alpha' \wedge K_i\neg\alpha) \Rightarrow \bigvee_{j=1}^{n} E^k(\alpha' \Rightarrow K_j\neg\alpha)$

(The contrapositive of) this new axiom says roughly that within a ball of radius k, the weak matching condition holds. The next lemma, which is analogous to Lemmas 7.7 and 7.12, makes the correspondence between the weak matching condition and axiom A9 precise. Note that unlike Lemmas 7.7 and 7.12, we need to assume in the next lemma that S is connected. Of course, for knowledge situations (S, s) it is always the case (by assumption) that S is connected.

Lemma 7.18: *Let S be either a connected set of interpreted states or a connected S5 Kripke structure. Then S satisfies the weak matching condition if and only if (S, s) satisfies every instance of axiom A9 for every state s of S.*

Proof: We shall prove the result when S is a connected set of interpreted states; the proof when S is a connected S5 Kripke structure is almost identical.

Assume first that S satisfies the weak matching condition, but that there is $t' \in S$ such that (S, t') does not satisfy some instance of axiom A9, say the instance $(\alpha' \wedge K_i\neg\alpha) \Rightarrow \bigvee_{j=1}^{n} E^k(\alpha' \Rightarrow K_j\neg\alpha)$. It follows that

$$(S, t') \models (\alpha' \wedge K_i\neg\alpha) \qquad (8)$$

and that for every process j,

$$(S, t') \not\models E^k(\alpha' \Rightarrow K_j\neg\alpha). \qquad (9)$$

By (9) and by Lemma 7.17, we know that for each process j there is $s'_j \in S$ (where s'_j is distance at most k from t') such that $(S, s'_j) \not\models \alpha' \Rightarrow K_j\neg\alpha$. Hence, for each j we know that $(S, s'_j) \models \alpha'$ and that there is some s_j such that $s_j \sim_j s'_j$ and $(S, s_j) \models \alpha$. It follows immediately that the four conditions of Definition 7.15 hold. By (8) we know that the primitive state component of t' is α'. So by the weak matching condition there is $t \in S$ with primitive state component α such that $t \sim_i t'$. Hence, $(S, t') \models \neg K_i\neg\alpha$. But this contradicts (8).

Conversely, assume that (S, s) satisfies every instance of axiom A9 for every state s of S. Let $s_1, ..., s_n$, $s'_1, ..., s'_n, t'$ be as in Definition 7.15. Since S is connected, there is k such that each of $s'_1, ..., s'_n$ is distance at most k from t'. It is straightforward to verify that for every process j, we have $(S, s'_j) \not\models (\alpha' \Rightarrow K_j\neg\alpha)$. So by Lemma 7.17, for each process j we know that (9) above holds. Since (S, t') satisfies axiom A9, and since (9) holds for every j, it follows that $(S, t') \not\models (\alpha' \wedge K_i\neg\alpha)$. But $(S, t') \models \alpha'$, and so $(S, t') \not\models K_i\neg\alpha$. Hence, there is $t \in S$ with primitive state component α such that $t \sim_i t'$. Therefore, the weak matching condition is satisfied. ∎

The next lemma is immediate from the equivalence of the matching condition with the pasting condition and the weak matching condition (Lemma 7.16), along with the characterizations of the pasting condition by axiom A6 (Lemma 7.7) and the weak matching condition by axiom A9 (Lemma 7.18).

Lemma 7.19: *Let S be either a connected set of interpreted states or a connected S5 Kripke structure. Then S satisfies the matching condition if and only if (S, s) satisfies every instance of axioms A6 and A9 for every state s of S.*

We now prove the analogue to Propositions 5.3 and 5.9.

Proposition 7.20: *Let $\mathcal{C}$ be a class of interpreted systems with n processes. If every attainable knowledge situation in every member of $\mathcal{C}$ satisfies the matching condition, then $S5_n$ + A6′ + A9 (respectively, $S5D_n$ + A6 + A9) is sound with respect to $\mathcal{C}$. Conversely, if every knowledge situation that satisfies the matching condition is attainable in some member of $\mathcal{C}$, then $S5_n$ + A6′ + A9 (respectively, $S5D_n$ + A6 + A9) is complete with respect to $\mathcal{C}$ for the language $\mathcal{L}_n(\Phi)$ (respectively, $\mathcal{L}_n^D(\Phi)$).*

Proof: Lemma 7.19 can be used to prove soundness and completeness of $\mathrm{S5D}_n$ + A6 + A9 in the language $\mathcal{L}_n^D(\Phi)$ in the same way as Lemma 7.7 was used to prove soundness and completeness of ML_n (in the proof of Proposition 5.3). We now consider soundness and completeness of $\mathrm{S5}_n$ + A6′ + A9 in the language $\mathcal{L}_n(\Phi)$. The soundness of A6′ follows from the soundness of A6, as we showed in the proof of Proposition 5.3. Therefore, the soundness of $\mathrm{S5}_n$ + A6′ + A9 follows from the soundness of $\mathrm{S5D}_n$ + A6 + A9, which we already showed. We now consider completeness.

Let M be the canonical Kripke structure as constructed in the proof of Theorem 7.1, except that "consistency" is now with respect to $\mathrm{S5}_n$ + A6′ + A9 rather than $\mathrm{S5}_n$. Since every maximal consistent set contains every instance of axiom A9, it follows as before that (M, s) satisfies every instance of axiom A9 for every state s of M. It then follows from Lemma 7.18 that every connected component of M satisfies the weak matching condition. Also, the techniques given in the proof of Proposition 5.3 can be carried over to show that M (and hence every connected component of M) satisfies the pasting condition. So by Lemma 7.16, every connected component of M satisfies the matching condition.

Assume now that $\varphi \in \mathcal{L}_n(\Phi)$ is consistent with respect to $\mathrm{S5}_n$ + A6′ + A9. Then there is a state s of M such that $(M, s) \models \varphi$. Let (S, s) be the knowledge situation at (M, s). Then $(S, s) \models \varphi$. Since, as we noted above, every connected component of M satisfies the matching condition, it follows that S does also (recall that S is connected, by definition of a knowledge situation). Hence, by assumption, (S, s) is attainable in some member of $\mathcal{C}$. So φ is satisfiable in some member of $\mathcal{C}$. This proves completeness. ∎

The next theorem is immediate from Propositions 7.14 and 7.20.

Theorem 7.21: *$S5_n$ + A6′ + A9 (respectively, $S5D_n$ + A6 + A9) is a sound and complete axiomatization with respect to interpreted systems of n processes where (1) the environment uniquely determines the initial state, (2) knowledge is cumulative, (3) process state transitions are independent of the initial environment, and (4) the primitive propositions characterize the initial environment, for the language $\mathcal{L}_n(\Phi)$ (respectively, $\mathcal{L}_n^D(\Phi)$).*

References

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

[CM] K. M. Chandy and J. Misra, How processes learn, *Distributed computing* **1**,*1* (1986), pp. 40–52.

[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.

[Em] E. A. Emerson, Alternative semantics for temporal logics, *Theoretical Comp. Sci.* **26** (1983), pp. 121–130.

[FH] R. Fagin and J. Y. Halpern, Belief, Awareness, and Limited Reasoning, *Artificial Intelligence* **1**, *34*, 1988, pp. 39–76.

[FHV] R. Fagin, J. Y. Halpern, and M. Y. Vardi, What can Machines Know? On the Epistemic Properties of Machines, *Proc. of National Conference on Artificial Intelligence (AAAI-86)*, 1986, pp. 428–434.

[FV1] R. Fagin and M.Y. Vardi, An internal semantics for modal logic. *Proc. 17th ACM Symp. on Theory of Computing*, May 1985, pp. 305–315.

[FV2] 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.

[HV1] J.Y. Halpern and M.Y. Vardi, The complexity of reasoning about knowledge and time, I: lower bounds, *J. Computer and System Sciences* 38(1989), pp. 195–237.

[HV2] J.Y. Halpern and M.Y. Vardi, The complexity of reasoning about knowledge and time in asynchronous systems, *20th ACM Symp. on Theory of Computing*, 1988, pp. 53–65.

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

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

[LR] R. Ladner and J. H. Reif, The logic of distributed protocols, *Theoretical Aspects of Reasoning about Knowledge* (ed. J.Y. Halpern), Morgan Kaufmann, 1986, pp. 207–221.

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

[Lev] H. Levesque, A logic of implicit and explicit belief, *Proc. of National Conference on Artificial Intelligence (AAAI-84)*, 1984, pp. 198–202.

[Ma] D. Makinson, On some completeness theorems in modal logic, *Zeitschrift für mathematische Logik und Grundlagen der Mathematik* **12** (1966), pp. 379–384.

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

[MT] Y. Moses and M. Tuttle, Programming simultaneous actions using common knowledge, *Proceedings of the 27th Annual Symposium on Foundations of Computer Science*, 1986, pp. 208–221.

[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.

[Pr] V. R. Pratt, Process logic, *Proc. 6th ACM Symp. on Principles of Programming Languages* (1979).

[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.

[Sa] L.J. Savage, *The Foundations of Statistics*, Wiley, New York, 1954.