A guide to completeness and complexity for modal logics of knowledge and belief*

Joseph Y. Halpern
IBM Almaden Research Center
San Jose, CA 95120

Yoram Moses
Weizmann Institute of Science
Rehovot, 76100 Israel

Abstract: We review and re-examine possible-worlds semantics for propositional logics of knowledge and belief with three particular points of emphasis: (1) we show how general techniques for finding decision procedures and complete axiomatizations apply to models for knowledge and belief, (2) we show how sensitive the difficulty of the decision procedure is to such issues as the choice of modal operators and the axiom system, and (3) we discuss how notions of common knowledge and distributed knowledge among a group of agents fit into the possible-worlds framework, As far as complexity is concerned, we show, among other results, that while the problem of deciding satisfiability of an S5 formula with one agent is *NP*-complete, the problem for many agents is *PSPACE*-complete. Adding a distributed knowledge operator does not change the complexity, but once a common knowledge operator is added to the language, the problem becomes complete for exponential time.

October 2, 1996

*Some material in this paper appeared in preliminary form in "A guide to the modal logics of knowledge and belief", which appeared in the Proceedings of IJCAI-85. This version is almost identical to one that appears in *Artificial Intelligence* **54**, 1992, pp. 311–379. The second author's work was supported in part by DARPA contract N00039-82-C-0250.

1 Introduction

Reasoning about knowledge and belief has long been an issue of concern in philosophy and artificial intelligence (cf. [Hin62, MH69, Moo85, Ros85]). We have argued [HM90] that reasoning about knowledge is also useful in understanding and reasoning about protocols in distributed systems, since messages can be viewed as changing the state of knowledge of a system; since the appearance of [HM90], a number of other papers have confirmed the important role of reasoning about knowledge in distributed systems (see [Hal87] for an overview and further references). Reasoning about knowledge also seems to be of importance in cryptography theory [GMR89, HMT88, Mer83] and database theory [Imi87, Rei88].

In order to formally reason about knowledge, we need a good semantic model. Part of the difficulty in providing such a model is that there is no agreement on exactly what the properties of knowledge are or should be. For example, is it the case that you know what facts you know? Do you know what you don't know? Do you know only true things, or can something you "know" actually be false?

Possible-worlds semantics provide a good formal tool for customizing a logic so that, by making minor changes in the semantics, we can capture different sets of axioms. The idea, first formalized by Hintikka [Hin62], is that in each state of the world, an agent has other states or worlds that he considers possible. An agent knows φ exactly if φ is true in all the worlds that he considers possible. As Kanger, Kripke, Hintikka (and probably others) pointed out [Hin61, Kan57b, Kan57a, Kri63], by imposing various conditions on this possibility relation, we can capture a number of interesting axioms. For example, if we require that the world that the agent finds himself in is always one of the worlds he considers possible (which amounts to saying that the possibility relation is reflexive), then it follows that the agent does not know false facts. Similarly, we can show that if the relation is transitive, then an agent that knows a given fact knows that he knows it. If we impose no restrictions on the class of structures, then the resulting logic is the well-known modal logic K. If we restrict attention to structures where the possibility relation is reflexive (resp., reflexive and transitive; reflexive, symmetric, and transitive), the resulting logic is the modal logic T (resp., S4, S5).

In this paper, we focus on the possible-worlds approach to modeling knowledge and belief. In this framework, we consider how hard it is to reason about knowledge. In particular, how hard is it to decide whether a given formula is valid (or satisfiable)? The answer to this question depends crucially on the choice of axioms. For example, in the propositional version of the single-agent case, Ladner [Lad77] has shown that for K, T, and S4 the problem of deciding satisfiability is polynomial-space complete,[1] while for S5 it is *NP*-complete, and thus no harder than the satisfiability problem for propositional logic. We extend these results to the multi-agent case, showing that, as long as there are at least two agents in the picture, then the complexity of the satisfiability problem is polynomial-space complete in all cases. We also consider what happens when we add more modal operators, including modal operators for *common knowledge* and *distributed knowledge*, to the language.

More generally, our aim in this paper is to review and re-examine the possible-worlds framework for knowledge and belief with three particular points of emphasis: (1) we show how general techniques for finding decision procedures and complete axiomatizations apply to models for

[1] All complexity-theoretic notions used in this paper are defined carefully in Section 6.

knowledge and belief, (2) we show how sensitive the difficulty of the decision procedure is to such issues as the choice of modal operators and the axiom system, and (3) we discuss how notions of common knowledge and distributed knowledge among a group of agents fit into the possible-worlds framework.

We begin in Section 2 by reviewing possible-world semantics for knowledge and belief in detail, and extending the standard techniques for proving complete axiomatizations to the multi-agent versions of K, T, S4, KD45, and S5. In Section 3 we consider issues of decidability, and show how a minor modification of our proof of completeness yields a procedure for effectively deciding whether a formula is valid.

In Sections 4 and 5 we extend the syntax so that we can reason about *common knowledge* and *distributed knowledge* in the language. Roughly speaking, a group has common knowledge of a fact φ exactly when everyone knows that everyone knows that everyone knows ... that φ is true. A group has distributed knowledge of φ if, roughly speaking, the agents' combined knowledge implies φ.[2] Common knowledge is essentially what McCarthy's "fool" knows; cf. [MSHI79]. By way of contrast, distributed knowledge is what a wise man, who knows what every member of the group knows, would know. As shown in [HM90], common knowledge is an essential state for reaching agreements and coordinating action. For very similar reasons, common knowledge also seems to play an important role in human understanding of speech acts (cf. [CM81]). The notion of distributed knowledge arises when reasoning about what states of knowledge a group can attain through communication, and thus is also crucial when reasoning about the efficacy of speech acts and about communication protocols in distributed systems (cf. [DM90, FV86, MT88]). We show that complete axiomatizations can also be obtained for reasoning about common knowledge and distributed knowledge, although the classical techniques that suffice for obtaining completeness in the case of reasoning about knowledge alone need to be extended. In the case of common knowledge, we use ideas from the complete axiomatization for PDL (Propositional Dynamic Logic) [KP81]; distributed knowledge requires some new techniques (cf. [FHV92]).

In Section 6 we turn to complexity-theoretic issues. We review some standard notions from complexity theory, and then reprove and extend Ladner's results to show that the satisfiability problem for the multi-agent versions of K, T, S4, and S5 is polynomial-space complete. This suggests that for S5, reasoning about many agents' knowledge is qualitatively harder than just reasoning about one agent's knowledge of the real world and of his own knowledge. It turns out that adding a distributed knowledge operator to the language does not substantially change the complexity of deciding the satisfiability of formulas in the language, but this is not the case for common knowledge. Techniques first applied to PDL [FL79, Pra79] can be used to show that when we add common knowledge to the language, the satisfiability problem for the resulting logic (whether it is based on K, T, S4, or S5) is complete for deterministic exponential time. (For S4 and S5, when there is only one agent, common knowledge is equivalent to knowledge, so we need to assume that there are at least two agents to get this result.) Thus, adding a common knowledge operator renders the decision procedure qualitatively more complex.

We conclude in Section 7 with some discussion of the appropriateness of the possible-worlds

[2]In an earlier version of this paper, what we are now calling distributed knowledge was called *implicit knowledge*. We have changed the name here to avoid conflict with the usage of the phrase "implicit knowledge" in [Lev84b] and [FH88].

approach for capturing knowledge and belief, particularly in light of our results on computational complexity.

2 Logics of knowledge and their properties

2.1 Syntax

A logic of any kind needs a language. Although we consider a number of different logics here, the syntax for all of them is essentially the same. We wish to reason about a world consisting of a propositional reality ("nature") and n agents, creatively named $1, \ldots, n$. Given a nonempty set Φ of primitive propositions, which we typically label $p, p', q, q', \ldots$ and a set of n agents, we define $\mathcal{L}_n(\Phi)$ to be the least set of formulas containing Φ, closed under negation, conjunction, and the modal operators $K_1, \ldots, K_n$. Thus, if φ and ψ are formulas of $\mathcal{L}_n(\Phi)$, then so are $(\neg\varphi)$, $(\varphi \wedge \psi)$, and $K_i(\varphi)$, for $i = 1, \ldots, n$ (where $K_i(\varphi)$ is read "agent i *knows* φ"). We omit parentheses if they are unnecessary for readability. We use the standard abbreviations $\varphi \vee \psi$ for $\neg(\neg\varphi \wedge \neg\psi)$ and $\varphi \Rightarrow \psi$ for $\neg(\varphi \wedge \neg\psi)$. We take *true* to be an abbreviation for some valid formula, such as $p \vee \neg p$; we abbreviate $\neg$*true* by *false*. The *size* of a formula φ in $\mathcal{L}_n(\Phi)$, denoted $|\varphi|$, is its length over the alphabet $\Phi \cup \{\neg, \wedge, (,), K_1, \ldots, K_n\}$. The *depth* of a formula φ, denoted $dep(\varphi)$, is the depth of nesting of K operators in φ. More formally, $dep(p) = 0$ for a primitive proposition $p \in \Phi$; $dep(\neg\psi) = dep(\psi)$; $dep(\varphi \wedge \psi) = \max(dep(\varphi), dep(\psi))$; $dep(K_i\psi) = 1 + dep(\psi)$, for $1 \leq i \leq n$. Notice that $dep(\varphi) < |\varphi|$ for all formulas φ. For later reference, we also define what it means for ψ to be a *subformula* of φ. Intuitively, ψ is a subformula of φ if it is a formula that is a substring of φ. Formally, we proceed by induction on the structure of φ: ψ is a subformula of φ if either $\psi = \varphi$ (so they are syntactically identical), or φ is of the form $\neg\varphi'$ (resp., $\varphi' \wedge \varphi''$, $K_i\varphi'$), and ψ is a subformula of φ' (resp., ψ is either a subformula of φ' or of φ'', ψ is a subformula of φ'). Let $Sub(\varphi)$ be the set of all subformulas of φ. We leave it to the reader to check that $|Sub(\varphi)| \leq |\varphi|$; that is, the length of φ is an upper bound on the number of subformulas of φ.

2.2 Possible-worlds semantics

Following Hintikka [Hin62], Sato [Sat77], Moore [Moo85], and others, we use a *possible-worlds* semantics to model knowledge. This provides us with a general framework for our semantical investigations of knowledge and belief. (Everything we say about "knowledge" in this subsection applies equally well to belief.) The essential idea behind possible-worlds semantics is that an agent's state of knowledge corresponds to the extent to which he can determine what world he is in. In a given world, we can associate with each agent the set of worlds that, according to the agent's knowledge, could possibly be the real world. An agent is then said to know a fact φ exactly if φ is true in all the worlds in this set; he does not know φ if there is at least one world that he considers possible where φ does not hold.

Kripke [Kri63] introduced *Kripke structures* as a formal model for a possible-worlds semantics for the modal logic of necessity and possibility. A *Kripke structure for n agents* is a tuple $M = (S, \pi, \mathcal{K}_1, \ldots, \mathcal{K}_n)$, where S is a set of *states* or *possible worlds*, π is a truth assignment to the primitive propositions of Φ for each state $s \in S$ (i.e., $\pi(s) : \Phi \rightarrow \{\mathbf{true}, \mathbf{false}\}$ for each state

$s \in S$), and $\mathcal{K}_i$ is a binary relation on the states of S, for $i = 1, \ldots, n$. The *size* of structure M is the number of states in S. We allow structures with infinite size.

The truth assignment π tells us, for each state s and each primitive proposition p, whether p is true or false in s. Thus, if p denotes the fact "It is raining in San Francisco", then $\pi(s)(p) = \mathbf{true}$ captures the situation in which it is raining in San Francisco in state s of structure M. $\mathcal{K}_i$ is intended to capture the possibility relation according to agent i: $(s,t) \in \mathcal{K}_i$ if in world s in structure M, agent i considers t a possible world.

One of the advantages of Kripke-style semantics is that given a Kripke structure, we can construct a corresponding labeled directed graph, where the nodes are the states of S and there is an edge from s to t labeled i exactly if $(s,t) \in \mathcal{K}_i$. This graph-theoretic viewpoint will turn out to be useful in our decision procedures (see Section 6).

For example, suppose $\Phi = \{p\}$ and $n = 2$, so that our language only has one primitive proposition p and there are only two agents. Further suppose that $M = (S, \pi, \mathcal{K}_1, \mathcal{K}_2)$, where $S = \{s, t, u\}$, p is true at states s and t, but false at u (so that $\pi(s)(p) = \pi(t)(p) = \mathbf{true}$ and $\pi(u)(p) = \mathbf{false}$), agent 1 considers state t possible in state s, state u possible in state u, and does not consider any states possible in state t (so that $\mathcal{K}_1 = \{(s,t),(u,u)\}$), and agent 2 considers state u possible in state s and also considers u possible in u (so that $\mathcal{K}_2 = \{(s,u),(u,u)\}$). The graph corresponding to this Kripke structure is described in Figure 1.

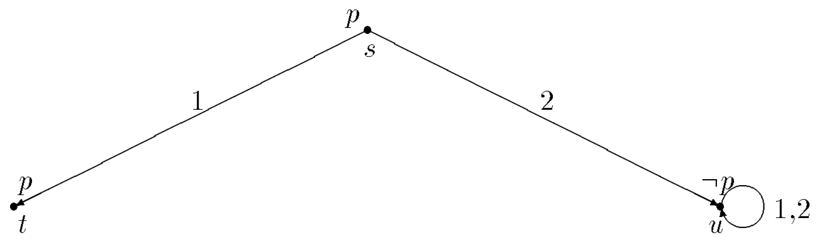


Figure 1: A simple Kripke structure

We now formally define a binary relation $\models$ between a formula φ and a pair (M,s) consisting of a structure M and a state s in M, where $(M,s) \models \varphi$ is read as either "φ is *true at* (M,s)", "(M,s) *satisfies* φ" or "φ *holds at* (M,s)".

$(M,s) \models p$ (for $p \in \Phi$) iff $\pi(s)(p) = \mathbf{true}$

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

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

$(M,s) \models K_i\varphi$ iff $(M,t) \models \varphi$ for all t satisfying $(s,t) \in \mathcal{K}_i$.

The first three clauses in this definition correspond to the standard clauses in the definition of truth for propositional logic. The last clause captures the intuition that agent i knows φ in world s of structure M exactly if φ is true at all worlds that i considers possible in s. For future reference, the reader should check that $\neg K_i \neg \varphi$ is true at a state s exactly if there is some t

such that $(s,t) \in \mathcal{K}_i$ and $(M,t) \models \varphi$. Thus, $\neg K_i \neg \varphi$ is true at s if agent i thinks there is some possible world where φ is true.

We leave it to the reader to check that in the structure M described in Figure 1, we have $(M,t) \models K_2 \mathit{false}$ and $(M,s) \models K_1 p \wedge K_2 \neg p \wedge K_1 K_2 \mathit{false}$.

We have suggested that a formula such as $K_i\varphi$ should be read "agent i knows φ". But is this a reasonable way of reading this formula? Does our semantics really capture the properties of knowledge in a reasonable way? How can we even answer this question?

One way of characterizing the properties of our interpretation of knowledge is by characterizing the formulas that are *valid*; i.e., those that are true in every state of every structure. More formally, given a structure $M = (S, \pi, \mathcal{K}_1, \ldots, \mathcal{K}_n)$, we say that φ is *valid in M*, and write $M \models \varphi$, if $(M,s) \models \varphi$ for every state $s \in S$; we say φ is *satisfiable in M* if $(M,s) \models \varphi$ for some state $s \in S$. We say φ is *valid with respect to a class $\mathcal{M}$ of structures* and write $\mathcal{M} \models \varphi$, if φ is valid in all structures in $\mathcal{M}$, and say φ is *satisfiable with respect to $\mathcal{M}$* if it is satisfiable in some structure in $\mathcal{M}$. It is easy to check that φ is valid in M (resp., valid with respect to $\mathcal{M}$) iff $\neg\varphi$ is not satisfiable in M (resp., not satisfiable with respect to $\mathcal{M}$). We use $\mathcal{M}_n$ to denote the class of all Kripke structures for n agents.

The following well-known theorem captures some of the formal properties of $\models$:

Theorem 2.1: *For all formulas $\varphi, \psi \in \mathcal{L}_n(\Phi)$, structures $M \in \mathcal{M}_n$, and agents $i = 1, \ldots, n$,*

1. *if φ is an instance of a propositional tautology, then $M \models \varphi$,*
2. *if $M \models \varphi$ and $M \models \varphi \Rightarrow \psi$, then $M \models \psi$.*
3. *$M \models (K_i\varphi \wedge K_i(\varphi \Rightarrow \psi)) \Rightarrow K_i\psi$,*
4. *if $M \models \varphi$ then $M \models K_i\varphi$.*

Proof: Parts (1) and (2) follows immediately from the fact the interpretation of $\wedge$ and $\neg$ in the definition of $\models$ is the same as in the propositional calculus. For part (3), if $(M,s) \models K_i\varphi \wedge K_i(\varphi \Rightarrow \psi)$, then for all states t such that $(s,t) \in \mathcal{K}_i$, we have both that $(M,t) \models \varphi$ and $(M,t) \models \varphi \Rightarrow \psi$. By propositional reasoning, it follows that $(M,t) \models \psi$ for all such t, and therefore $(M,s) \models K_i\psi$. For part (4), if $M \models \varphi$ then $(M,t) \models \varphi$ for all states t in M. In particular, for any fixed state s in M, it follows that $(M,t) \models \varphi$ for all t such that $(s,t) \in \mathcal{K}_i$. Thus, $(M,s) \models K_i\varphi$ for all states s in M, and hence $M \models K_i\varphi$. ∎

2.3 Axiom systems for knowledge

Theorem 2.1 tells us that by subscribing to Kripke semantics we are forced to accept a number of constraints on the type of notions of knowledge that we can model.[3] We now show that in a precise sense these are the only constraints that we are forced to accept by using Kripke semantics. We do so by defining an axiom system K_n whose axioms and rules of inference correspond to the clauses in Theorem 2.1, and then proving that this axiom system characterizes

[3] We discuss the ramifications of this point in Section 7.

Kripke structures for knowledge.[4] Such results are well known (cf. [Che80, HC68, Sat77]). We reprove them here using techniques originally due to Kaplan and Makinson [Kap66, Mak66] that show the close correspondence between the axioms and a particular Kripke structure called the *canonical structure*. We reprove these results here in order to make this paper self-contained, and to make it easier for the reader to follow our later discussion, where we show how the standard techniques need to be modified in order to deal with new modal operators.

K_n consists of two axioms:[5]

A1. All instances of tautologies of the propositional calculus

A2. $(K_i\varphi \wedge K_i(\varphi \Rightarrow \psi)) \Rightarrow K_i\psi$, $i = 1, \ldots, n$

and two rules of inference:

R1. From $\vdash \varphi$ and $\vdash \varphi \Rightarrow \psi$ infer $\vdash \psi$ (Modus ponens)

R2. From $\vdash \varphi$ infer $\vdash K_i\varphi$ (Generalization)

A formula φ is said to be K_n *provable*, denoted $\mathrm{K}_n \vdash \varphi$, if φ is an instance of one of the axioms, or if φ follows from provable formulas by one of the inference rules R1 and R2 (we omit the qualifier K_n if it is clear from context). Provability relative to an arbitrary axiom system $\mathcal{S}$ is defined in an analogous fashion.[6] A formula φ is (K_n) *consistent* if $\neg\varphi$ is not K_n provable. A finite set $\{\varphi_1, \ldots, \varphi_k\}$ of formulas is consistent exactly if $\varphi_1 \wedge \ldots \wedge \varphi_k$ is consistent, and an infinite set of formulas is consistent exactly if all of its finite subsets are consistent. A formula or set of formulas is said to be *inconsistent* exactly if it is not consistent. A set F of formulas is a *maximal consistent set* if it is consistent and for all $\varphi \notin F$, the set $F \cup \{\varphi\}$ is inconsistent.

Using standard techniques of propositional reasoning (i.e., using A1 and R1), we can show

Lemma 2.2: *In any axiom system AX that includes A1 and R1, every consistent set F can be extended to a maximal (AX-)consistent set. In addition, if F is a maximal consistent set, then it satisfies the following properties:*

(a) for every formula φ, exactly one of φ and $\neg\varphi$ is in F,

(b) $\varphi \wedge \psi \in F$ iff $\varphi \in F$ and $\psi \in F$,

(c) if φ and $\varphi \Rightarrow \psi$ are both in F, then ψ is in F, and

(d) if φ is K_n provable, then $\varphi \in F$.

[4] The name K_n is inspired by the fact that for one agent, the system reduces to the well-known modal logic K.

[5] Technically, these are axiom schemas, not axioms, since they represent a family of axioms. We abuse notation and refer to them as axioms in this paper.

[6] It is perhaps worth pointing out that the use of the *deduction theorem*, which is legitimate in standard axiomatizations of propositional logic, is not legitimate for K_n. Roughly speaking, the deduction theorem holds for an axiom system $\mathcal{S}$ if, whenever we can infer ψ from φ, then $\psi \Rightarrow \varphi$ is provable in $\mathcal{S}$. Notice that by the generalization rule, we can infer $K_i\varphi$ from φ in K_n. However, it does *not* follow that $\varphi \Rightarrow K_i\varphi$ is provable in K_n (in fact, it is not).

Proof: Let F be a consistent set of formulas, and let $\varphi \in \mathcal{L}_n(\Phi)$. An easy argument shows that one of $F \cup \{\varphi\}$ or $F \cup \{\neg\varphi\}$ is consistent. For assume to the contrary that neither of them is consistent. Then $F \cup \{\varphi \vee \neg\varphi\}$ is also inconsistent, and it follows that F is inconsistent because $\varphi \vee \neg\varphi$ is a propositional tautology. A set that results from successively adding either φ or $\neg\varphi$ to F in a consistent fashion for all $\varphi \in \mathcal{L}_n(\Phi)$ is clearly a maximal consistent extension of F.

In order to see that maximal consistent sets have all the properties we claimed, let F be a maximal consistent set. If $\varphi \in \mathcal{L}_n(\Phi)$, we know by the argument above that one of $F \cup \{\varphi\}$ and $F \cup \{\neg\varphi\}$ is consistent. If $F \cup \{\varphi\}$ is consistent, then we must have $\varphi \in F$ since F is a maximal consistent set. Similarly, if $F \cup \{\neg\varphi\}$ is consistent then $\neg\varphi \in F$. Thus, one of φ or $\neg\varphi$ is in F. It is clear that we cannot have both φ and $\neg\varphi$ in F, since otherwise F would be inconsistent.

This observation is enough to let us prove all the other properties we claimed. For example, if $\varphi \wedge \psi \in F$, then we must have $\varphi \in F$, for otherwise, as we just showed, we would have $\neg\varphi \in F$, and F would be inconsistent. Similarly, we must have $\psi \in F$. Conversely, if φ and ψ are both in F, we must have $\varphi \wedge \psi \in F$, for otherwise we would have $\neg(\varphi \wedge \psi) \in F$, and, again, F would be inconsistent. We leave the proof that F has properties (c) and (d) to the reader. ∎

An axiom system $\mathcal{S}$ is *sound* with respect to a class $\mathcal{M}$ of structures if every formula provable from $\mathcal{S}$ is valid with respect to $\mathcal{M}$. $\mathcal{S}$ is *complete* with respect to $\mathcal{M}$ if every formula that is valid with respect to $\mathcal{M}$ is provable from $\mathcal{S}$. We think of an axiom system as characterizing a class of structures exactly if it provides a sound and complete axiomatization of that class.

Theorem 2.3: *K_n is a sound and complete axiomatization with respect to $\mathcal{M}_n$.*

Proof: Theorem 2.1 implies that K_n is sound with respect to $\mathcal{M}_n$. In order to prove completeness, we must show that every formula that is valid with respect to $\mathcal{M}_n$ is K_n provable. It suffices to prove

$$\text{Every } \mathrm{K}_n\text{-consistent formula is satisfiable in some structure in } \mathcal{M}_n. \quad (*)$$

For suppose we can prove $(*)$ and φ is a valid formula. If φ is not provable, then neither is $\neg\neg\varphi$, so, by definition, $\neg\varphi$ is consistent. It follows from $(*)$ that $\neg\varphi$ is satisfiable in some structure in $\mathcal{M}_n$, contradicting the validity of φ with respect to $\mathcal{M}_n$.

We prove $(*)$ using a general technique that works for a wide variety of modal logics. We construct a special Kripke structure $M^c \in \mathcal{M}_n$, which we call the *canonical* Kripke structure for K_n, in which *every* K_n-consistent formula is satisfiable. M^c has a state s_V corresponding to every maximal consistent set V. We will show

$$(M^c, s_V) \models \varphi \text{ iff } \varphi \in V. \quad (**)$$

Note that $(**)$ suffices to prove $(*)$, for by Lemma 2.2, if φ is consistent, then φ is contained in some maximal consistent set V. From $(**)$ it follows that $(M^c, s_V) \models \varphi$, showing that φ is satisfiable in M^c.

We proceed as follows. Given a set V of formulas, define $V/K_i = \{\varphi : K_i\varphi \in V\}$. Let $M^c = (S, \pi, \mathcal{K}_1, \ldots, \mathcal{K}_n)$, where

$$\begin{aligned} S &= \{s_V : V \text{ is a maximal consistent set}\} \\ \pi(s_V)(p) &= \begin{cases} \textbf{true} & \text{if } p \in V \\ \textbf{false} & \text{if } p \notin V \end{cases} \\ \mathcal{K}_i &= \{(s_V, s_W) : V/K_i \subseteq W\}. \end{aligned}$$

We now show, by induction on the structure of φ, that for all V we have $(M^c, s_V) \models \varphi$ iff $\varphi \in V$. More precisely, assuming that the claim holds for all subformulas of φ, we will also show that it holds for φ. If φ is a primitive proposition p, this is immediate from the definition of $\pi(s_V)$ above. The cases where φ is a conjunction or a negation follow easily from parts (a) and (b) of Lemma 2.2; we leave details to the reader.

Finally, suppose that φ is of the form $K_i\psi$ and that $\varphi \in V$. Then $\psi \in V/K_i$ and, by definition of $\mathcal{K}_i$, if $(s_V, s_W) \in \mathcal{K}_i$, then $\psi \in W$. Thus, using the induction hypothesis, $(M^c, s_W) \models \psi$ for all W such that $(s_V, s_W) \in \mathcal{K}_i$. By the definition of $\models$, it follows that $(M^c, s_V) \models K_i\psi$.

For the other direction, assume $(M^c, s_V) \models K_i\psi$. It follows that the set $(V/K_i) \cup \{\neg\psi\}$ is inconsistent. For suppose not. Then by Lemma 2.2 it would have a maximal consistent extension W, and, by construction, we would have $(s_V, s_W) \in \mathcal{K}_i$. By the induction hypothesis we have $(M^c, s_W) \models \neg\psi$, and so $(M^c, s_V) \models \neg K_i\psi$, contradicting our original assumption. Since $(V/K_i) \cup \{\neg\psi\}$ is inconsistent, some finite subset, say $\{\varphi_1, \ldots, \varphi_k, \neg\psi\}$, must be inconsistent. Thus, by propositional reasoning, we have

$$\vdash \varphi_1 \Rightarrow (\varphi_2 \Rightarrow (\ldots(\varphi_k \Rightarrow \psi)\ldots)).$$

By R2, we have

$$\vdash K_i(\varphi_1 \Rightarrow (\varphi_2 \Rightarrow (\ldots(\varphi_k \Rightarrow \psi)\ldots)).$$

By induction on k, together with axiom A2 and propositional reasoning, we can show

$$\vdash K_i(\varphi_1 \Rightarrow (\varphi_2 \Rightarrow (\ldots(\varphi_k \Rightarrow \psi)\ldots)) \Rightarrow (K_i\varphi_1 \Rightarrow (K_i\varphi_2 \Rightarrow (\ldots(K_i\varphi_k \Rightarrow K_i\psi)\ldots)).$$

Now from R1, we get

$$\vdash K_i\varphi_1 \Rightarrow (K_i\varphi_2 \Rightarrow (\ldots(K_i\varphi_k \Rightarrow K_i\psi)\ldots)).$$

Thus, it follows that the set $\{K_i\varphi_1, \ldots, K_i\varphi_k, \neg K_i\psi\}$ is inconsistent. Since $\varphi_1, \ldots, \varphi_k \in V/K_i$, we must have $K_i\varphi, \ldots, K_i\varphi_i \in V$. Since V is consistent and one of $K_i\psi$ or $\neg K_i\psi$ is in V, we must in fact have $K_i\psi \in V$, as desired. ∎

In the philosophical literature, one finds a great deal of discussion as to which axioms truly characterize knowledge (see [Len78] for a discussion and review). Some of the ones more commonly considered include:

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

the *knowledge axiom*, which states that only true facts can be known (this is usually taken as the essential property distinguishing knowledge from belief);

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

the *positive introspection axiom*, which states that an agent knows what facts he knows;

A5. $\neg K_i\varphi \Rightarrow K_i\neg K_i\varphi, \quad i = 1, \ldots, n,$

the *negative introspection axiom*, which says that an agent knows what facts he does not know; and

A6. $\neg K_i(\mathit{false})$,

which says that the agent does not know inconsistent facts.

In the case of a single agent, K+A3 has been called T, T+A4 has been called S4, S4+A5 is known as S5, while K+{A4,A5,A6} has been called KD45. Clearly other combinations are possible; we focus on these here, since they have been most commonly used for reasoning about knowledge. In the case of n agents, where the language is based on the set Φ of primitive propositions, we denote these systems by T_n, $\mathrm{S4}_n$, $\mathrm{S5}_n$, and $\mathrm{KD45}_n$ respectively. We occasionally omit the subscript if $n = 1$, in line with more traditional notation.

Philosophers have spent years trying to determine which of these systems (if any) best captures knowledge (again, see [Len78]). Theorem 2.1 shows that we are modeling a rather idealized reasoner, who knows all tautologies and all the logical consequences of his knowledge. The classical view of knowledge is that it is true, justified belief. That is, an agent knows φ if he believes that φ holds, φ actually does hold, and the agent is justified in believing that φ holds. Under this view of knowledge, an axiom such as A3 is necessary. On the other hand, as pointed out in, say [Lev84a], the knowledge represented in a knowledge based is typically not required to be true. Thus, the propositional attitude that the philosophers have called *belief*—where agents may have false beliefs—seems more appropriate than knowledge for formalizing the reasoning and deduction of a knowledge base. A3 would therefore not be appropriate for describing the knowledge of a knowledge base. Since we do want to assume that knowledge bases do not believe inconsistent facts, we typically require A6 rather than A3. (We remark that it is easy to show that A6 in fact follows by propositional reasoning from A3.)

Philosophers have also shown that axiom A5 does *not* hold with respect to the interpretation of knowledge as true, justified belief [Get63, Len78]. However, the S5 axioms do capture an interesting interpretation of knowledge appropriate for reasoning about distributed systems (see [HM90] and Section 7). We continue here with our investigation of all these logics and, in particular, the relationship between the axioms mentioned above and the properties of the $\mathcal{K}_i$ relation. We defer further comments on the appropriateness of the axioms to Section 7.

Theorem 2.3 implies that the provable formulas of K_n correspond precisely to the formulas that are valid with respect to Kripke structures. As Kripke showed [Kri63], there are simple conditions that we can impose on the possibility relations $\mathcal{K}_i$ so that the valid formulas of the resulting structures are exactly the provable formulas of T_n, $\mathrm{S4}_n$, $\mathrm{S5}_n$ and $\mathrm{KD45}_n$, respectively. We try to motivate these conditions here; we first need a few definitions.

We say that a binary relation $\mathcal{K}$ on a set S is *reflexive* if $(s,s) \in \mathcal{K}$ for all $s \in S$; $\mathcal{K}$ is *transitive* if, for all $s,t,u \in S$, if $(s,t) \in \mathcal{K}$ and $(t,u) \in \mathcal{K}$, then $(s,u) \in \mathcal{K}$; $\mathcal{K}$ is *symmetric* if, for all $s,t \in S$, whenever $(s,t) \in \mathcal{K}$ then $(t,s) \in \mathcal{K}$; $\mathcal{K}$ is *Euclidean* if, for all $s,t,u \in S$, whenever $(s,t) \in \mathcal{K}$ and $(s,u) \in \mathcal{K}$, then $(t,u) \in \mathcal{K}$; finally, $\mathcal{K}$ is *serial* if, for all $s \in S$, there is some t such that $(s,t) \in \mathcal{K}$. A relation that is reflexive, symmetric, and transitive is also commonly called an *equivalence relation*.

Some of the relationships between these notions are described by the following lemma, whose straightforward proof is left to the reader (cf. [Che80]):

Lemma 2.4:

1. *If $\mathcal{K}$ is symmetric and transitive, then $\mathcal{K}$ is Euclidean.*
2. *$\mathcal{K}$ is symmetric, transitive, and serial iff $\mathcal{K}$ is reflexive and Euclidean iff $\mathcal{K}$ is reflexive, symmetric, and transitive.* ∎

Let $\mathcal{M}_n^r$ (resp., $\mathcal{M}_n^{rt}$; $\mathcal{M}_n^{rst}$; $\mathcal{M}_n^{elt}$) be the class of all structures for n agents where the possibility relations are reflexive (resp., reflexive and transitive; reflexive, symmetric, and transitive; Euclidean, serial, and transitive).

To see the relationship between these notions and the axioms described above, consider the canonical Kripke structure M^c defined in Theorem 2.3. Recall that $(s_V, s_W) \in \mathcal{K}_i$ in M^c exactly if $V/K_i \subseteq W$, where $V/K_i = \{\varphi : K_i\varphi \in V\}$. Now suppose that all instances of A3 are true at s_V. Then it is easy to see that $(s_V, s_V) \in \mathcal{K}_i$, since $V/K_i \subseteq V$. This suggests that A3 corresponds to reflexivity. Indeed, it is easy to check that A3 is valid in all structures where the possibility relation is reflexive. Semantically, it is not hard to see the connection between A3 and reflexivity. If s is a world in a structure $M \in \mathcal{M}^r$, then agent i must consider s to be one of his possible worlds in s. Thus, if agent i knows φ in s, then φ must be true in s; i.e., $(M, s) \models K_i\varphi \Rightarrow \varphi$.

Similarly, we can show that A4 corresponds to transitivity. It is easy to see that A4 is valid in all structures where the possibility relation is transitive. Moreover, we can show that A4 forces the possibility relations in the canonical structure to be transitive. To see this, suppose that $(s_V, s_W), (s_W, s_X) \in \mathcal{K}_i$ and that all instances of A4 are true at s_V. Then if $K_i\varphi \in V$, by A4 we have $K_iK_i\varphi \in V$, and, by the construction of M^c, we have $K_i\varphi \in W$ and $\varphi \in X$. Thus, $V/K_i \subseteq X$ and $(s_V, s_X) \in \mathcal{K}_i$, as desired.

Similar reasoning shows that axiom A5 corresponds to the possibility relation being Euclidean. We remark that similar reasoning also shows that symmetry corresponds to the axiom

$$\varphi \Rightarrow K_i \neg K_i \neg \varphi,$$

which can be shown to be a consequence of A3 and A5. This corresponds to the observation of Lemma 2.4 that a relation that is both reflexive and Euclidean is also symmetric.[7]

Finally, we can show that A6 corresponds the possibility relation being serial. It is easy to see that A6 is valid in all structures where the possibility relation is serial. Moreover, as we now show, A6 forces the possibility relation in the canonical model to be serial. To see this, suppose that $\neg K_i \mathit{false} \in V$. Consider the set V/K_i. If V/K_i is not consistent, then there are formulas $\varphi_1, \ldots, \varphi_k$ in V/K_i such that $\mathrm{K}_n \vdash \varphi_1 \Rightarrow (\varphi_2 \Rightarrow \ldots(\varphi_k \Rightarrow \mathit{false})\ldots))$. Now using techniques similar to those used in the proof of Theorem 2.3, we can show (using A6) that the set $\{K_i\varphi_1, \ldots, K_i\varphi_k, \neg K_i \mathit{false}\}$ is inconsistent, contradicting the consistency of V Thus, it follows that V/K_i is consistent, and can therefore be extended to a maximal consistent set W. Our construction guarantees that $(s_V, s_W) \in \mathcal{K}_i$.

We say that a structure M is a *model of K_n* if every K_n-provable formula is valid in M. We can similarly say that a structure is a model of T_n, $\mathrm{S4}_n$, $\mathrm{S5}_n$, or $\mathrm{KD45}_n$.

Arguments essentially identical to those of Theorem 2.3 can now be used to show:

[7] Since Lemma 2.4 says that a relation that is both reflexive and Euclidean must also be transitive, the reader may suspect that axiom A4 is redundant in S5. This indeed is the case.

Theorem 2.5:

1. *T_n is a sound and complete axiomatization with respect to $\mathcal{M}_n^r$.*
2. *$S4_n$ is a sound and complete axiomatization with respect to $\mathcal{M}_n^{rt}$.*
3. *$S5_n$ is a sound and complete axiomatization with respect to $\mathcal{M}_n^{rst}$.*
4. *$KD45_n$ is a sound and complete axiomatization with respect to $\mathcal{M}_n^{elt}$.*

Proof: For part (1), observe that from the above discussion, it follows that every reflexive structure satisfies all the axioms of T_n, and thus is a model of T_n. Consequently, T_n is sound with respect to $\mathcal{M}_n^r$. For completeness, we need to show that any T_n-consistent formula is satisfiable in a reflexive structure. This is done exactly as in the proof of Theorem 2.3. We define a canonical Kripke structure M^c for T_n whose states each corresponds to a maximal T_n-consistent set of formulas V. The $\mathcal{K}_i$ relations are defined as in the proof of Theorem 2.3. By our above remarks, these relations are reflexive; a proof identical to that used in Theorem 2.3 can now be used to show that $\varphi \in V$ iff $(M^c, s_V) \models \varphi$ for all maximal T_n-consistent sets V. The proofs of parts (2), (3), and (4) are analogous. ∎

Theorem 2.5 shows that every structure in $\mathcal{M}_n^r$ (resp., $\mathcal{M}_n^{rt}$, $\mathcal{M}_n^{rst}$, $\mathcal{M}_n^{elt}$) is a model of T_n (resp., $\mathrm{S4}_n$, $\mathrm{S5}_n$, $\mathrm{KD45}_n$). We might be tempted to conjecture that the converse also holds. This is not quite true, as the following example shows. Suppose $n = 1$ and $\Phi = \{p\}$, and let M be the structure consisting of two states s and t, such that $\pi(s)(p) = \pi(t)(p) = \mathbf{true}$ and $\mathcal{K}_1 = \{(s,t),(t,t)\}$, as shown in Figure 2:

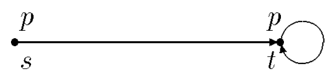


Figure 2: A model of S5 where $\mathcal{K}_1$ is not reflexive

Clearly $\mathcal{K}_1$ is not reflexive or symmetric. Nevertheless, it is easy to see that M is a model of S5 and *a fortiori* a model of S4 and T. (The proof proceeds by first showing that for all formulas φ in $\mathcal{L}(\{p\})$, we have $(M,s) \models \varphi$ iff $(M,t) \models \varphi$.) Using a slight variant of this construction, we can construct a model of KD45 where the $\mathcal{K}_1$ relation is not Euclidean. We leave details to the reader.

However, the intuition behind such a conjecture is almost correct in two senses. For one thing, to every model of T_n (resp., $\mathrm{S4}_n$, $\mathrm{S5}_n$, $\mathrm{KD45}_n$) there corresponds an *equivalent* model where the $\mathcal{K}_i$ relations are transitive; equivalence relations; Euclidean, serial, and transitive). To make this precise, define the *reflexive closure* of a structure M, where $M = (S, \pi, \mathcal{K}_1, \ldots, \mathcal{K}_n)$, to be the structure M^r, where $M^r = (S, \pi, \mathcal{K}_1^r, \ldots, \mathcal{K}_n^r)$, in which $\mathcal{K}_i^r$ is the reflexive closure of $\mathcal{K}_i$; i.e., $\mathcal{K}_i^r = \mathcal{K}_i \cup \{(s,s) : s \in S\}$. Similarly, the *rt-closure* (resp. , *rst-closure*, *et-closure*) of M is the structure M^{rt} (resp., M^{rst}, M^{et}) in which the possibility relations are the reflexive, transitive closure (resp., reflexive, symmetric and transitive closure; Euclidean and transitive closure) of the $\mathcal{K}_i$ relations in M. We say that (M, s) and (M', s') are *equivalent*, and write $(M, s) \equiv (M', s')$, if they satisfy exactly the same formulas, i.e., $(M, s) \equiv (M', s')$ if for all formulas φ we have $(M, s) \models \varphi$ iff $(M', s') \models \varphi$.

Theorem 2.6:

1. *If M is a model of T_n, then so is its reflexive closure M^r; moreover, $M^r \in \mathcal{M}_n^r$ and $(M, s) \equiv (M^r, s)$ for all states s in M.*
2. *If M is a model of $S4_n$, then so is its rt-closure M^{rt}; moreover, $M^{rt} \in \mathcal{M}_n^{rt}$ and $(M, s) \equiv (M^{rt}, s)$ for all states s in M.*
3. *If M is a model of $S5_n$, then so is its rst-closure M^{rst}; moreover, $M^{rst} \in \mathcal{M}_n^{rst}$ and $(M, s) \equiv (M^{rst}, s)$ for all states s in M.*
4. *If M is a model of $KD45_n$, then so is its et-closure M^{et}; moreover, $M^{et} \in \mathcal{M}_n^{elt}$ and $(M, s) \equiv (M^{et}, s)$ for all states s in M.*

Proof: For part (1), let M be a model of T_n and let M^r be its reflexive closure. We prove by induction on the structure of φ that $(M, s) \models \varphi$ iff $(M^r, s) \models \varphi$ for all states s of M. The claim is immediate for primitive propositions, and straightforward for negations and conjunctions. Assume that the claim holds for ψ, and let φ be of the form $K_i\psi$. The induction hypothesis, together with the fact that $\mathcal{K}_i \subseteq \mathcal{K}_i^r$, imply that if $(M^r, s) \models K_i\psi$ then $(M, s) \models K_i\psi$. For the other direction, suppose that $(M, s) \models K_i\psi$. If $(s, t) \in \mathcal{K}_i^r$ then either $(s, t) \in \mathcal{K}_i$ or $s = t$. In the former case $(M, t) \models \psi$ since $(M, s) \models K_i\psi$, so by the induction hypothesis we have $(M^r, t) \models \psi$. In the latter case, we have that $(M, t) \models K_i\psi \Rightarrow \psi$ since M is a model of T_n, and since $s = t$, we also have $(M, t) \models K_i\psi$. Thus, $(M, t) \models \psi$. By the induction hypothesis, we have that $(M^r, t) \models \psi$. We have just shown that $(M^r, t) \models \psi$ for all t such that $(s, t) \in \mathcal{K}_i^r$, so it follows that $(M^r, s) \models K_i\psi$.

The proofs of parts (2), (3), and (4) are similar and left to the reader. Observe that in part (4), there is no need to ensure that the possibility relations in M^{et} are serial. This already follows from the fact that A6 ensures that the possibility relations in M are serial. ∎

We conclude by taking a closer look at the single-agent case of S5 and KD45. The following result shows that in the case of S5, we can further restrict our attention to structures where the possibility relation is *universal*; i.e., in every state, all states of S are considered possible. Intuitively, this means that in the case of S5, we can talk about *the* set of worlds the agent considers possible; this set is the same in every state, and consists of all the worlds. Similarly, for KD45 we can restrict attention to structures with one distinguished state, which intuitively describes what is true in the "real" world, and a set of states (which does not in general include the real world) corresponding to the worlds that the agent thinks possible in every state. More formally, we have

Proposition 2.7:

1. *If a formula φ is S5 consistent, then φ is satisfiable in a structure $M = (S, \pi, \mathcal{K}_1)$, where $\mathcal{K}_1$ is universal, i.e., $\mathcal{K}_1 = \{(s, t) : s, t \in S\}$.*

2. *If a formula φ is KD45 consistent, then φ is satisfiable in a structure $M = (\{s_0\} \cup S, \pi, \mathcal{K}_1)$, where S is nonempty and $\mathcal{K}_1 = \{(s,t) : s \in \{s_0\} \cup S, t \in S\}$.*

Proof: We first consider the case of KD45. Suppose φ is KD45 consistent. Then by Theorem 2.5, it follows that there is a structure $M' = (S', \pi', \mathcal{K}_1')$ with $M' \in \mathcal{M}_1^{elt}$ and a state $s_0 \in S'$ such that $(M', s_0) \models \varphi$. Let $\mathcal{K}_1'(s_0) = \{t : (s_0, t) \in \mathcal{K}_1'\}$. Since $\mathcal{K}_1'$ is serial, $\mathcal{K}_1'(s_0)$ must be nonempty. It is also easy to check that since $\mathcal{K}_1'$ is Euclidean, we have $(s,t) \in \mathcal{K}_1'$ for all $s, t \in \mathcal{K}_1'(s_0)$. Finally, since $\mathcal{K}_1'$ is transitive, if $s \in \mathcal{K}_1'(s_0)$ and $(s,t) \in \mathcal{K}_1'$, then $t \in \mathcal{K}_1'(s_0)$. Let $M = (\{s_0\} \cup \mathcal{K}_1'(s_0), \pi, \mathcal{K}_1\}$, where π is the restriction of π' to $\{s_0\} \cup \mathcal{K}_1'(s_0)$ and $\mathcal{K}_1 = \{(s,t) : s \in \{s_0\} \cup \mathcal{K}_1'(s_0), t \in \mathcal{K}_1'(s_0)\}$. By the observations above, $\mathcal{K}_1$ is the restriction of $\mathcal{K}_1'$ to $\{s_0\} \cup \mathcal{K}_1'(s_0)$. Note that $\mathcal{K}_1$ is serial (since $\mathcal{K}_1'(s_0)$ is nonempty), Euclidean, and transitive. A straightforward induction on the structure of formulas now shows that for all $s \in \{s_0\} \cup \mathcal{K}_1'(s_0)$ and all formulas ψ, we have $(M, s) \models \psi$ iff $(M', s) \models \psi$. In particular, this means that $(M, s_0) \models \varphi$, so that φ is satisfiable in a structure of the form claimed. We leave details to the reader.

In the case of S5, we proceed just as above, except that we start with a structure $M \in \mathcal{M}_1^{rst}$. Using the fact that $\mathcal{K}_1'$ is now reflexive, it is easy to show that the relation $\mathcal{K}_1$ we construct is universal. The rest of the proof proceeds as before. ∎

It follows from Proposition 2.7 that we can assume without loss of generality that models of S5 have a particularly simple form, namely (S, π), where we do not mention the $\mathcal{K}_1$ relation, but simply assume that $(s,t) \in \mathcal{K}_1$ for all $s, t \in S$. Similarly, we can take models of KD45 to have the form (s_0, S, π), where as we mentioned above, the intuition is that s_0 is the "real" world, and S is the set of worlds that the agent considers possible. As we shall see, this simple representation of models for S5 and KD45 has important implications when it comes to the difficulty of deciding whether a formula is S5 or KD45 provable. We cannot in general get such simple representations for the other logics we have considered, nor can we get them if we have two or more agents in the picture.

We mentioned earlier that the intuition that every model of T_n (resp. $\mathrm{S4}_n$, $\mathrm{S5}_n$, $\mathrm{KD45}_n$) is in $\mathcal{M}^r$ (resp. $\mathcal{M}^{rt}$, $\mathcal{M}^{rst}$, $\mathcal{M}^{elt}$) is essentially correct in *two* senses. The second sense involves the notion of a *frame*. We define a *frame for n agents* to be a tuple $(S, \mathcal{K}_1, \ldots, \mathcal{K}_n)$, where S is a set of states and $\mathcal{K}_1, \ldots, \mathcal{K}_n$ are binary relations on S. Thus, a frame is like a Kripke structure without the truth assignment π. We say that the Kripke structure $(S, \pi, \mathcal{K}_1, \ldots, \mathcal{K}_n)$ is *based on* the frame $(S, \mathcal{K}_1, \ldots, \mathcal{K}_n)$. It turns out that if we look at the level of frames rather than at the level of structures, the converse to Theorem 2.5 does hold. More precisely, suppose we let $\mathcal{F}_n$ be the class of all Kripke frames. Just as for structures, we can consider subclasses of $\mathcal{F}_n$ such as $\mathcal{F}_n^r$, $\mathcal{F}_n^{rt}$, $\mathcal{F}_n^{rst}$, and $\mathcal{F}_n^{elt}$. We say a frame F is a *model of* T_n (resp., $\mathrm{S4}_n$, $\mathrm{S5}_n$, $\mathrm{KD45}_n$) if every structure based on F is a model of T_n (resp., $\mathrm{S4}_n$, $\mathrm{S5}_n$, $\mathrm{KD45}_n$). We have the following:

Theorem 2.8:

1. *F is a model of T_n iff $F \in \mathcal{F}_n^r$,*
2. *F is a model of $\mathrm{S4}_n$ iff $F \in \mathcal{F}_n^{rt}$,*
3. *F is a model of $\mathrm{S5}_n$ iff $F \in \mathcal{F}_n^{rst}$,*

4. F is a model of $\mathrm{KD45}_n$ *iff* $F \in \mathcal{F}_n^{elt}$.

Proof: We do the proof for part (1) here, leaving the remainder (which are all similar) to the reader. It follows immediately from Theorem 2.5 that if $F \in \mathcal{F}^r$ then F is a model of T_n. For the converse, suppose that $F = (S, \mathcal{K}_1, \ldots, \mathcal{K}_n)$ is a model of T_n and $F \notin \mathcal{F}_n^r$. Then for some state $s \in S$ and some agent i, we have $(s,s) \notin \mathcal{K}_i$. Let p be a primitive proposition in Φ and define π so that $\pi(s)(p) = \mathbf{false}$ and $\pi(t)(p) = \mathbf{true}$ for all states $t \neq s$. For a primitive proposition $q \neq p$, we take $\pi(s')(q) = \mathbf{true}$ for all states $s' \in S$. Let $M = (S, \pi, \mathcal{K}_1, \ldots, \mathcal{K}_n)$. Clearly M is based on F. It is easy to see that $(M,s) \models \neg p \wedge K_i p$, contradicting the assumption that F is a model of T_n. ∎

This result suggests that the viewpoint of frames might be more appropriate than that of structures. Although we have shown that, for example, we can find a structure that is a model of $\mathrm{S5}_n$ but is not in $\mathcal{M}_n^{rst}$ (or even $\mathcal{M}_n^r$), this is not the case at the level of frames. If a frame is a model of $\mathrm{S5}_n$, then it must be in $\mathcal{F}_n^{rst}$. Conversely, if a frame is in $\mathcal{F}_n^{rst}$, then it is a model of $\mathrm{S5}_n$. All the results in this paper can be easily converted to the frame level. Nevertheless, we work at the level of structures since most papers in the AI literature work at that level too. We remark that the idea of using frames to characterize axiom systems is well known in modal logic; it appears, for example, in [Gol92, HC84, Ben85].

3 Decidability

In the preceding section we showed that the set of valid formulas of $\mathcal{M}_n$ are indeed characterized by K_n, and that the valid formulas of various interesting subclasses of $\mathcal{M}_n$ are characterized by other systems, such as T_n, $\mathrm{S4}_n$, and $\mathrm{S5}_n$. However, our results were not constructive; they gave no indication of how to tell whether a given formula was indeed provable (and thus also valid in the appropriate class of structures).

In this section, we present results showing that the question of whether a formula is valid is effectively decidable. However, before we do so, it is worth considering to what extent this problem is of interest. Notice that for many of the formulas we are interested in, checking validity is straightforward (as evidenced by Theorem 2.1). As well, the nonconstructive proof suffices to show that our axioms do characterize the properties of the notion(s) of knowledge that we have defined.

It seems that the most obvious situation where we might want decidability is if we have an agent whose situation is characterized by a collection of axioms φ, who wants to know whether a formula ψ also holds in this situation. This amounts to asking whether $\varphi \Rightarrow \psi$ is valid. An example of this type of situation is if we take the agent to be a knowledge base. The agent can then draw conclusions about the state of the world based on what validly follows from the information in the knowledge base. In such situations, what the agent knows is exactly what can be proved from a given set of formulas using the axioms of the logic.

However, rather than characterizing the agent's situation by a collection of axioms, we may instead be able to characterize the agent as being in some state s in a Kripke structure. This is the case, for example, in many distributed systems applications (see [HM90, Hal87]). Now suppose the agent knows φ in state s. Then the agent is really interested in determining whether

ψ is also true *in state* s, not whether $\varphi \Rightarrow \psi$ is true in all states. (Note that the latter is what the agent learns by checking if $\varphi \Rightarrow \psi$ is valid.) Moreover, as we shall see, the difficulty in deciding whether a formula is valid does not necessarily imply difficulty in deciding whether the same formula is true in a given situation. The question of whether to consider the problem of deciding validity or of deciding truth in a state depends on the application. We actually consider both issues here.

We start our investigation by considering the *model-checking problem*, that is, the problem of deciding if a formula is satisfiable in a given Kripke structure. As we shall see, the model-checking problem is closely related to the problem of deciding whether a formula is true in a given state in a given structure. Moreover, model checking turns out to be an essential component in our algorithm to decide validity.

There is no general procedure for doing model checking in an infinite Kripke structure. Indeed, it is not even clear how we could represent an arbitrary infinite structure effectively. On the other hand, in finite Kripke structures, model checking is relatively straightforward. Given a finite Kripke structure $M = (S, \pi, \mathcal{K}_1, \ldots, \mathcal{K}_n)$, define $||M||$ to be the sum of the number of states in S and the number of pairs in $\mathcal{K}_i$, $i = 1, \ldots, n$. In the theorem below (and in later results), we use the notation $O(f(n))$, read "*order* of f" or "(big) O of f". Roughly speaking, this denotes $cf(n)$, for some constant $c > 0$ independent of n. Thus, for example, we say that the running time of the algorithm below is $O(||M|| \times |\varphi|)$, this means that there is some constant $c > 0$, independent of the structure M and the formula φ, such that for all structures M and formulas φ, the time to check if φ is satisfied in M is at most $c(||M|| \times |\varphi|)$.

Proposition 3.1: *Given a structure M and a formula $\varphi \in \mathcal{L}_n$, there is an algorithm for checking if φ is satisfied in M that runs in time $O(||M|| \times |\varphi|)$.*

Proof: Let $\varphi_1, \ldots, \varphi_k$ be the subformulas of φ (i.e., the members of $Sub(\varphi)$) listed in order of length, with ties broken arbitrarily. Thus we have $\varphi_k = \varphi$, and if φ_i is a subformula of φ_j, then $i < j$. There are at most $|\varphi|$ subformulas of φ, so we must have $k \leq |\varphi|$. An easy induction on k' shows that we can label each state s in M with φ_j or $\neg\varphi_j$, for $j = 1, \ldots, k'$, depending on whether or not φ_j is true at s, in time $O(k'||M||)$. The only nontrivial case is if φ_j is of the form $K_i\varphi_{j'}$, where $j' < j$. We label a state s with $K_i\varphi_{j'}$ iff each state t such that $(s,t) \in \mathcal{K}_i$ is labeled with $\varphi_{j'}$. Assuming inductively that each state has already been labeled with $\varphi_{j'}$ or $\neg\varphi_{j'}$, this step can clearly be carried out in time $O(||M||)$, as desired. ∎

Observe that this result holds independent of the number of agents. It continues to hold if we restrict attention to particular classes of structures, such as $\mathcal{M}_n^{rt}$ or $\mathcal{M}_n^{rst}$. We can easily modify the algorithm to check whether φ holds at a particular state s in M.

We now turn our attention to the problem of checking whether a given formula is provable. We start with K_n. Our first step is to show that if a formula is K_n consistent, not only is it satisfiable in some structure (in particular, the canonical structure constructed in the proof of Theorem 2.3), in fact it is satisfiable in a finite structure (which the canonical structure is certainly not!). The proof is actually just a slight variant of the proof of Theorem 2.3. The idea is that rather than considering maximal K_n consistent subsets of $\mathcal{L}_n(\Phi)$ when trying to construct a structure satisfying a formula φ, we restrict attention to sets of subformulas of φ.

Theorem 3.2: *If φ is* K_n *consistent then φ is satisfiable in a structure in $\mathcal{M}_n$ with at most $2^{|\varphi|}$ states where every primitive proposition in Φ which is not a subformula of φ is false at every state.*

Proof: Let $Sub^+(\varphi)$ consist of all the subformulas of φ and their negations, i.e., $Sub^+(\varphi) = Sub(\varphi) \cup \{\neg\psi : \psi \in Sub(\varphi)\}$. Let $Con(\varphi)$ be the set of maximal K_n consistent subsets of $Sub^+(\varphi)$. A proof almost identical to that of Lemma 2.2 can be used to show that every K_n consistent subset of $Sub^+(\varphi)$ can be extended to an element of $Con(\varphi)$. Moreover, a member of $Con(\varphi)$ contains either ψ or $\neg\psi$ for every formula $\psi \in Sub(\varphi)$ (but not both, since otherwise it would not be consistent). So the cardinality of $Con(\varphi)$ is at most $2^{|Sub(\varphi)|}$, which is at most $2^{|\varphi|}$, since $|Sub(\varphi)| \leq |\varphi|$.

We now construct a structure $M_\varphi = (S_\varphi, \pi, \mathcal{K}_1, \ldots, \mathcal{K}_n)$. The construction is essentially the same as that of Theorem 2.3, except that we take $S_\varphi = \{s_V : V \in Con(\varphi)\}$. Notice that our construction guarantees that all primitive propositions in Φ that are not subformulas of φ will be false in every state (since they will not be in any set V). We can now show that if $V \in Con(\varphi)$, then for all $\psi \in Sub^+(\varphi)$ we have $(M_\varphi, s_V) \models \psi$ iff $\psi \in V$. The proof is identical to that of Theorem 2.3, and so is omitted here. ∎

From Theorem 3.2, we can get an effective (although not particularly efficient) procedure for checking if a formula φ is satisfiable with respect to $\mathcal{M}_n$. We simply construct every Kripke structure of size $2^{|\varphi|}$ where every primitive proposition in Φ which is not a subformula of φ is false at every state (there are only a finite, albeit very large, number of these) and check if φ is true at some state of one of these structures. The latter check is done using the model-checking algorithm of Proposition 3.1. If φ is true at some state in one of these structures, then clearly φ is satisfiable with respect to $\mathcal{M}_n$. Conversely, if φ is satisfiable with respect to $\mathcal{M}_n$, by Theorem 3.2, it must be satisfiable in one of these structures.

As a consequence, we can now show that the validity problem for $\mathcal{M}_n$, and hence the provability problem for K_n, is *decidable*; that is, there is an algorithm that decides whether a given formula is valid with respect to $\mathcal{M}_n$.

Corollary 3.3: *The validity problem for $\mathcal{M}_n$ and the provability problem for* K_n *are decidable.*

Proof: Since φ is K_n provable iff φ is valid with respect to $\mathcal{M}_n$ (by Theorem 2.3) iff $\neg\varphi$ is not satisfiable with respect to $\mathcal{M}_n$ (by definition), we can simply check (using the above procedure) if $\neg\varphi$ is satisfiable. ∎

Note that by Corollary 3.3, we have a way of checking whether a formula is K_n provable without deriving a single proof using the axiom system! (Actually, with some additional effort, we can extend the ideas in the proof of Theorem 3.2 so that if a formula is K_n provable, then we can effectively find a proof of it. The interested reader can consult [EH85] for an analogous proof in the context of a temporal logic.)

We can extend the arguments of Theorem 3.2 to the other logics we have been considering.

Theorem 3.4: *If φ is* T_n *(resp.,* $\mathrm{S4}_n$*,* $\mathrm{S5}_n$*,* $\mathrm{KD45}_n$*) consistent, then φ is satisfiable in a structure in $\mathcal{M}_n^r$ (resp., $\mathcal{M}_n^{rt}$, $\mathcal{M}_n^{rst}$, $\mathcal{M}_n^{elt}$) with at most $2^{|\varphi|}$ states where every primitive proposition in Φ which is not a subformula of φ is false at every state.*

Proof: The proof in the case of T_n is identical to that of Theorem 3.2, except that we consider maximal T_n consistent subsets of $Sub^+(\varphi)$ rather than maximal K_n consistent subsets of $Sub^+(\varphi)$. Note that in the case of T_n, the axiom $K_i\varphi \Rightarrow \varphi$ guarantees that $V/K_i \subseteq V$, so we get reflexivity of $\mathcal{K}_i$ even if we restrict attention to subsets of $Sub^+(\varphi)$.

The obvious modification of the proof of Theorem 3.2 does not work for $S4_n$, since the $\mathcal{K}_i$ relations may not be transitive if we define $(s_V, s_W) \in \mathcal{K}_i$ iff $V/K_i \subseteq W$. For example, if φ is the formula $K_1 p$, then the maximal $S4_n$ consistent subsets of $Sub^+(\varphi)$ are $V_1 = \{K_1 p, p\}$, $V_2 = \{\neg K_1 p, p\}$, and $V_3 = \{\neg K_1 p, \neg p\}$. Note that $V_1/K_1 \subseteq V_2$ and $V_2/K_1 \subseteq V_3$, but $V_1/K_1 \not\subseteq V_3$. Although $V_1/K_1 \subseteq V_2$, intuitively it should be clear that we do not want to have $(s_{V_1}, s_{V_2}) \in \mathcal{K}_1$. The reason is that every maximal $S4_n$ consistent extension of V_1 contains $K_1 K_1 p$, and so in such an extension, no consistent extension of V_2 would be considered possible.

In the case of $S4_n$, we deal with this problem as follows: We repeat the construction of Theorem 3.2 except that we take $\mathcal{K}_i$ to consist of $\{(s_V, s_W) : V/K_i \subseteq W/K_i\}$. Clearly this definition guarantees that $\mathcal{K}_i$ is transitive. For $S5_n$, we take $\mathcal{K}_i$ to consist of $\{(s_V, s_W) : V/K_i = W/K_i\}$. This guarantees that $\mathcal{K}_i$ is an equivalence relation. Note that in the case of $S4_n$ and $S5_n$, the axiom $K_i\varphi \Rightarrow \varphi$ guarantees that if $V/K_i \subseteq W/K_i$, then $V/K_i \subseteq W$. For $KD45_n$ we do not have this axiom, so we take $\mathcal{K}_i$ to consist of $\{(s_V, s_W) : V/K_i = W/K_i, V/K_i \subseteq W\}$. We leave it to the reader to check that with this definition, $\mathcal{K}_i$ is Euclidean, transitive, and serial. The proof in all cases now continues along the lines of Theorem 2.3; we leave details to the reader. ∎

Just as in the case of K_n, we can use this result to give us an effective technique for deciding whether a formula is T_n (resp., $S4_n$, $S5_n$, $KD45_n$) provable.

Corollary 3.5: *The validity problem for $\mathcal{M}_n^r$ (resp., $\mathcal{M}_n^{rt}$, $\mathcal{M}_n^{rst}$, $\mathcal{M}_n^{elt}$) and the provability problem for T_n (resp., $S4_n$, $S5_n$, $KD45_n$) are decidable.*

It turns out that in fact there are more efficient ways of checking whether a formula is provable than those suggested by the results we have just proved; we discuss this issue in Section 6.

4 Incorporating common knowledge

In a number of situations it is useful to be able to reason about the state of knowledge of a group of agents, not just that of an individual agent. For example, we may sometimes want to reason about facts that everyone in the group knows. At other times, we wish to talk about facts that are part of a group's "culture": not only does everyone know them, but everyone knows that everyone knows them, and everyone knows that everyone knows that everyone knows them, and so on. These facts are said to be *common knowledge*. Put another way, these can be thought of as the facts that "any fool knows" (cf. [MSHI79]).

To capture these notions, define the language $\mathcal{L}_n^C(\Phi)$ to be the result of extending $\mathcal{L}_n(\Phi)$ by adding two new operators: E and C. Thus, if φ is a formula, then so are $E\varphi$ ("everyone knows φ") and $C\varphi$ ("φ is common knowledge"). We view $E\varphi$ as an abbreviation for $K_1\varphi \wedge \ldots \wedge K_n\varphi$, while $C\varphi$ is intended to represent the infinite conjunction $E\varphi \wedge EE\varphi \wedge \ldots$. Note that if $n = 1$

then $E\varphi \equiv K\varphi$.[8]

We can capture the intended meaning of these constructs quite straightforwardly in our semantics. We denote $E^1\varphi = E\varphi$, and define $E^{k+1}\varphi = E(E^k\varphi)$ for $k \geq 1$. Now, given a structure $M = (S, \pi, \mathcal{K}_1, \ldots, \mathcal{K}_n)$, define

$$\begin{array}{l}(M,s) \models E\varphi \text{ iff } (M,s) \models K_i\varphi \text{ for all } i = 1,\ldots,n, \text{ and} \\ (M,s) \models C\varphi \text{ iff } (M,s) \models E^k\varphi \text{ for } k = 1,2,\ldots\end{array}$$

Our definition of common knowledge has an interesting graph-theoretical interpretation, which turns out to be useful in many of our applications. Define a state t to be *reachable from state s in k steps* ($k \geq 1$) if there exist states $s_0, s_1, \ldots, s_k$ such that $s_0 = s$, $s_k = t$ and for all j with $0 \leq j \leq k-1$, there exists an agent i_j such that $(s_j, s_{j+1}) \in \mathcal{K}_{i_j}$. We say t is *reachable from s* if t is reachable from s in k steps for some $k \geq 1$. Thus, t is reachable from s exactly if there is a path from s to t in the graph.

Lemma 4.1:

1. *$(M,s) \models E^k\varphi$ if and only if $(M,t) \models \varphi$ for all t that are reachable from s in k steps.*
2. *$(M,s) \models C\varphi$ if and only if $(M,t) \models \varphi$ for all t that are reachable from s.*

Proof: Part (1) follows from a straightforward induction on k, while part (2) is immediate from part (1). ∎

This lemma gives us another way to think about common knowledge. Suppose we define the relation $\mathcal{E} = \mathcal{K}_1 \cup \ldots \cup \mathcal{K}_n$, and define $\mathcal{C}$ to be the transitive closure of $\mathcal{E}$. Then $(M,s) \models E\varphi$ iff $(M,t) \models \varphi$ for all t such that $(s,t) \in \mathcal{E}$ and $(M,s) \models C\varphi$ iff $(M,t) \models \varphi$ for all t such that $(s,t) \in \mathcal{C}$. Thus, the E and C modalities can be viewed as corresponding to the knowledge of two artificial individuals (whose possible world relations are defined by $\mathcal{E}$ and $\mathcal{C}$ respectively).

The most important properties of the modal operators E and C are captured in the following theorem.

Theorem 4.2: *For all formulas $\varphi, \psi \in \mathcal{L}_n^C(\Phi)$ and all structures $M \in \mathcal{M}_n$ we have*

1. *$M \models E\varphi \Leftrightarrow (K_1\varphi \wedge \ldots \wedge K_n\varphi)$*
2. *$M \models C\varphi \Leftrightarrow E(\varphi \wedge C\varphi)$,*
3. *if $M \models \varphi \Rightarrow E(\psi \wedge \varphi)$ then $M \models \varphi \Rightarrow C\psi$.*

Proof: Part (1) follows immediately from the semantics of E. To prove the other parts, we use the characterization of common knowledge provided by Lemma 4.1, namely that $(M,s) \models C\varphi$ iff $(M,t) \models \varphi$ for all states t that are reachable from s.

[8] In practice it is quite useful to have a family of modal operators E_G and C_G, where $G \subseteq \{1,\ldots,n\}$, so that we can talk about facts that every agent in G knows, or about a fact being common knowledge among the agents in G. All the results we prove here for E and C can be easily extended to deal with the more general case.

For part (2), suppose $(M,s) \models C\varphi$. Thus, $(M,t) \models \varphi$ for all states t that are reachable from s. In particular, if u is reachable from s in one step, then $(M,u) \models \varphi$. Moreover, since any state reachable from u is also reachable from s, we have that $(M,t) \models \varphi$ for all t that are reachable from u. Thus, $(M,u) \models \varphi \wedge C\varphi$ for all u that are reachable from s in one step, so $(M,s) \models E(\varphi \wedge C\varphi)$. For the converse, suppose $(M,s) \models E(\varphi \wedge C\varphi)$. If we now restrict attention to structures in $\mathcal{M}^r$, where the possibility relation is reflexive, it is easy to see that $(M,s) \models C\varphi$. The result actually holds in arbitrary structures, although we have to work a little harder to prove it. Suppose that t is reachable from s and s' is the first node after s on a path from s to t. Since $(M,s) \models E(\varphi \wedge C\varphi)$, it follows that $(M,s') \models \varphi \wedge C\varphi$. Either $s' = t$ or t is reachable from s'. In the former case, $(M,t) \models \varphi$ since $(M,s') \models \varphi$, while in the latter case, $(M,t) \models \varphi$ using Lemma 4.1 and the fact that $(M,s') \models C\varphi$. Since $(M,t) \models \varphi$ for all t reachable from s, it follows that $(M,s) \models C\varphi$.

Finally, for part (3), suppose $M \models \varphi \Rightarrow E(\psi \wedge \varphi)$ and $(M,s) \models \varphi$. We show by induction on k that for all k, we have $(M,t) \models \psi \wedge \varphi$ for all t reachable from s in k steps. Suppose t is reachable from s in one step. Since $M \models \varphi \Rightarrow E(\psi \wedge \varphi)$, we have $(M,s) \models E(\psi \wedge \varphi)$. Since t is reachable from s in one step, by Lemma 4.1, we have $(M,t) \models \psi \wedge \varphi$ as desired. If $k = k' + 1$, then there is some t' that is reachable from s in k' steps such that t is reachable from t' in one step. By induction hypothesis, we have $(M,t') \models \psi \wedge \varphi$. Now the same argument as in the base case shows that $(M,t) \models \psi \wedge \varphi$. This completes the induction proof. Since ψ holds at every state reachable from s, it follows that $(M,s) \models C\psi$, as desired. ∎

The two properties of C described in the previous theorem are quite important in practice. The fact that $C\varphi \equiv E(\varphi \wedge C\varphi)$ says that $C\varphi$ can be viewed as a solution of the fixed point equation $X \equiv E(\varphi \wedge X)$. Intuitively, this says that common knowledge of φ holds in a situation X where everyone in the group knows that φ holds and that they are in situation X. This viewpoint helps to explain how common knowledge of φ may arise without the agents learning each of the facts $E\varphi, E^2\varphi, \ldots$ one by one. In fact, $C\varphi$ turns out to be the *greatest* solution of this fixed point equation, in that it is implied by all other solutions (see [HM90] for more details).

The last property gives us a way of deducing that common knowledge holds in a structure. It is often called the *Induction Rule*. The proof of its soundness shows why: the antecedent gives us the essential ingredient for proving that $\varphi \Rightarrow E^k(\psi \wedge \varphi)$ is valid by induction on k.

Somewhat surprisingly, even though C is an "infinitary" operator, the properties described in the previous theorem are enough to completely characterize it. Consider the following axioms (cf. [Leh84, Mil81, MSHI79]):

A7. $E\varphi \equiv K_1\varphi \wedge \ldots \wedge K_n\varphi$

A8. $C\varphi \Rightarrow E(\varphi \wedge C\varphi)$

and the rule of inference:

R3. From $\vdash \varphi \Rightarrow E(\psi \wedge \varphi)$ infer $\vdash \varphi \Rightarrow C\psi$.

Let K_n^C (resp. T_n^C, $\mathrm{S4}_n^C$, $\mathrm{S5}_n^C$) be the system that results from adding A7, A8, and R3 to the axioms for K_n (resp., T_n, $\mathrm{S4}_n$, $\mathrm{S5}_n$).

Theorem 4.3: *For formulas in the language $\mathcal{L}_n^C(\Phi)$, the system K_n^C (resp., T_n^C, $S4_n^C$, $S5_n^C$, $KD45_n^C$) is a sound and complete axiomatization with respect to $\mathcal{M}_n$ (resp., $\mathcal{M}_n^r$, $\mathcal{M}_n^{rt}$, $\mathcal{M}_n^{rst}$, $\mathcal{M}_n^{elt}$).*

Proof: Soundness follows from Theorem 2.3 (resp. Theorem 2.5) and Theorem 4.2. For completeness, we proceed as in the proof of Theorem 2.3 to show that if φ is consistent, then φ is satisfiable. However, for technical reasons (which are explained below) we need to restrict to finite structures as is done in the proof of Theorem 3.2. We deal with the case K_n^C here; we leave it to the reader to make the obvious modifications to deal with the other cases.

We define $Sub_C(\varphi)$ to consist of all subformulas of φ together with the formulas $E(\psi \wedge C\psi)$, $\psi \wedge C\psi$, $K_1(\psi \wedge C\psi)$, $\ldots$, $K_n(\psi \wedge C\psi)$, for each subformula $C\psi$ of φ, and $K_1\psi$, $\ldots$, $K_n\psi$ for each subformula $E\psi$ of φ. It is easy to see that $|Sub_C(\varphi)| \leq (n+3)\,|\varphi|$. We define $Sub_C^+(\varphi)$ to consist of all formulas of $Sub_C(\varphi)$ and their negations, and define $Con_C(\varphi)$ to consist of all maximal K_n^C-consistent subsets of $Sub_C^+(\varphi)$. Let $M_\varphi = (S_\varphi, \pi, \mathcal{K}_1, \ldots, \mathcal{K}_n)$, where S_φ consists of $\{s_V : V \in Con_C(\varphi)\}$ and $\pi(s_V)(p) = \mathbf{true}$ for $p \in \Phi$ iff $p \in V$. It is easy to see that S_φ has at most $2^{(n+3)|\varphi|}$ elements. As in the proof of Theorem 2.3 we take $\mathcal{K}_i = \{(s_V, s_W) : V/K_i \subseteq W\}$.

We again want to show that for every formula $\psi \in Sub_C^+(\varphi)$, we have $(M_\varphi, s_V) \models \psi$ iff $\psi \in V$. We proceed by induction on the structure of formulas. The argument in the case that ψ is a primitive proposition, a conjunction, a negation, or of the form $K_i\psi'$ is essentially identical to that used in Theorem 2.3; we do not repeat it here.

Suppose ψ is of the form $E\psi'$. Since V is a maximal consistent subset of $Sub_C^+(\varphi)$, which includes (by definition) all formulas $K_i\psi'$ for $i = 1, \ldots, n$, by A7 we get that $E\psi' \in V$ iff $K_i\psi' \in V$, $i = 1, \ldots, n$. We have already argued that $K_i\psi' \in V$ iff $(M_\varphi, s_V) \models K_1\psi'$, for $i = 1, \ldots, n$. Thus, $E\psi' \in V$ iff $(M_\varphi, s_V) \models K_1\psi' \wedge \ldots \wedge K_n\psi'$, which in turn holds iff $(M_\varphi, s_V) \models E\psi'$.

Finally, we must consider the case that ψ is of the form $C\psi'$. If $C\psi' \in V$, we show by induction on k that if s_W is reachable from s_V in k steps, then both ψ' and $C\psi'$ are in W. For $k = 1$, observe that A8 and the fact that $V \in Con_C(\varphi)$ together imply that $E(\psi' \wedge C\psi') \in V$. Now our construction guarantees that if s_W is reachable from s_V in one step (so that $(s_V, s_W) \in \mathcal{K}_i$ for some $i \in \{1, \ldots, n\}$), we have $(\psi' \wedge C\psi') \in W$. Since $W \in Con_C(\varphi)$, it follows that both ψ' and $C\psi'$ are in W. Now assume that $k = k' + 1$ and the claim holds for k'. If s_W is reachable from s_V in k steps, then there exists W' such that $s_{W'}$ is reachable from s_V in k' steps and s_W is reachable from $s_{W'}$ in one step. By the induction hypothesis, both ψ' and $C\psi'$ are in W'. The argument for the base case now shows that both $C\psi'$ and ψ' are in W. Our argument shows that, in particular, $\psi' \in W$ for all W such that s_W is reachable from s_V. By our main induction hypothesis, $(M_\varphi, s_W) \models \psi'$ for all s_W reachable from s_V. Thus, $(M_\varphi, s_V) \models C\psi'$.

For the converse, suppose that $(M_\varphi, s_V) \models C\psi'$. If W is a set of formulas, we use φ_W to denote the conjunction of the formulas in W. Let $\mathcal{W} = \{W \in Con_C(\varphi) : (M_\varphi, s_W) \models C\psi'\}$ and let $\varphi_{\mathcal{W}} = \bigvee_{W \in \mathcal{W}} \varphi_W$. Note that it is crucial here that $Con_C(\varphi)$ is finite, and each $W \in Con_C(\varphi)$ is a finite set of formulas (for otherwise $\varphi_{\mathcal{W}}$ would not be a formula in our language). This construction would not work if we had considered maximal consistent subsets of $\mathcal{L}_n^C(\Phi)$, as we do in the proof of Theorem 2.3. Suppose we can prove

$$\vdash \varphi_{\mathcal{W}} \Rightarrow E(\psi' \wedge \varphi_{\mathcal{W}}). \tag{1}$$

Then by R3 we have

$$\vdash \varphi_{\mathcal{W}} \Rightarrow C\psi'.$$

Since $V \in \mathcal{W}$, we have $\vdash \varphi_V \Rightarrow \varphi_{\mathcal{W}}$, so

$$\vdash \varphi_V \Rightarrow C\psi'.$$

Thus, $C\psi' \in V$, as desired.

So it only remains to show (1). By propositional reasoning and axiom A7, it is easy to see that it suffices to show that for each $W \in \mathcal{W}$ and agent i we have

$$\vdash \varphi_W \Rightarrow K_i(\psi' \wedge \varphi_{\mathcal{W}}). \tag{2}$$

We next show that by straightforward propositional reasoning we have:

$$\vdash \varphi_{\mathcal{W}} \equiv \neg(\bigvee_{W' \notin \mathcal{W}} \varphi_{W'}). \tag{3}$$

We will first prove that $\vdash \bigvee_{W \in Con_C(\varphi)} \varphi_W$. Assume not. For ease of exposition let us denote $\bigvee_{W \in Con_C(\varphi)} \varphi_W$ by η. If not $\vdash \eta$ then $\neg\eta$ is consistent. Thus, by Lemma 2.2 there is a maximally consistent set T that extends $\{\neg\eta\}$. In particular, T is a consistent set, and for every formula ψ in $Sub_C(\varphi)$, either ψ or its negation is in T. Denote by W_T the set $T \cap Sub_C^+(\varphi)$. It follows that $W_T \in Con_C(\varphi)$ and, as well, $\varphi_{W_T} \in T$. Moreover, by propositional reasoning we must have that $\vdash \varphi_{W_T} \Rightarrow \bigvee_{W \in Con_C(\varphi)} \varphi_W$, or in other word $\vdash \varphi_{W_T} \Rightarrow \eta$. It follows that $\eta \in T$, and given that T is a consistent set, this contradicts the assumption that $\neg\eta \in T$. We thus conclude that $\vdash \bigvee_{W \in Con_C(\varphi)} \varphi_W$. The claim that (3) holds now follows by propositional reasoning, since the φ_W's are mutually exclusive.

Using (3), propositional reasoning, A2, and R2, we can see that (2) follows from

$$\vdash \varphi_W \Rightarrow K_i(\psi' \wedge (\bigwedge_{W' \notin \mathcal{W}} \neg\varphi_{W'})).$$

Finally, observe that for all agents i and formulas φ_1, φ_2, we have

$$\vdash (K_i\varphi_1 \wedge K_i\varphi_2) \Rightarrow K_i(\varphi_1 \wedge \varphi_2) \tag{4}$$

This follows using A2, propositional reasoning, and the observation that $\varphi_1 \Rightarrow (\varphi_2 \Rightarrow (\varphi_1 \wedge \varphi_2))$ is a propositional tautology. Thus, it suffices to prove, for all agents i, $W \in \mathcal{W}$, and $W' \notin \mathcal{W}$,

$$\vdash \varphi_W \Rightarrow K_i(\neg\varphi_{W'}), \text{ and} \tag{5}$$

$$\vdash \varphi_W \Rightarrow K_i(\psi'). \tag{6}$$

Suppose (5) does not hold. Then $\varphi_W \wedge \neg K_i(\neg\varphi_{W'})$ is consistent. It then follows that $(W/K_i \cup W/E) \subseteq W'$. To see this, suppose, for example, that $E\varphi' \in W$ and $\varphi' \notin W'$. It follows that $\neg\varphi' \in W'$, so that $\vdash \varphi' \Rightarrow \neg\varphi_{W'}$. Using propositional reasoning, R2, and A2, it is easy to show that each implication in the following chain is provable:

$$\varphi_W \Rightarrow E\varphi' \Rightarrow K_i\varphi' \Rightarrow K_i\neg\varphi_{W'}.$$

But this contradicts the assumption that $\varphi_W \wedge \neg K_i(\neg\varphi_{W'})$ is consistent. Thus, we have $(W/K_i \cup W/E) \subseteq W'$, and hence $(s_W, s_{W'}) \in \mathcal{K}_i$. Since $W' \notin \mathcal{W}$, we have, by the definition of $\mathcal{W}$, that $(M_\varphi, s_{W'}) \not\models C\psi'$. This means that there is a state t reachable from $s_{W'}$ such that $(M_\varphi, t) \models \neg\psi'$. But if t is reachable from $s_{W'}$, it is also reachable from s_W, since $(s_W, s_{W'}) \in \mathcal{K}_i$. Thus, $(M_\varphi, s_W) \models \neg C\psi$. But this contradicts our assumption that $s_W \in \mathcal{W}$. Thus (5) holds.

The proof of (6) follows similar lines. If $\varphi_W \wedge \neg K_i\psi'$ is consistent, similar arguments to the ones we have just seen show that there exists $W' \in Con_C(\varphi)$ such that $(s_W, s_{W'}) \in \mathcal{K}_i$ and $(M_\varphi, s_{W'}) \models \neg\psi'$. But again this contradicts our assumption that $(M_\varphi, s_W) \models C\psi'$. This completes the proof of (6) and the theorem. ∎

5 Distributed knowledge

Besides the knowledge common to a group of agents, it is also often desirable to be able to reason about the knowledge that is *distributed* in the group, i.e., what someone who could combine the knowledge of all of the agents in the group would know. Thus, for example, if Alice knows φ and Bob knows $\varphi \Rightarrow \psi$, then the knowledge of ψ is distributed among them, even though it might be the case that neither of them individually knows ψ. Whereas common knowledge, in McCarthy's analogy, essentially corresponds to what "any fool" knows, distributed knowledge corresponds to what a (fictitious) "wise man" (one that knows exactly what each individual agent knows) would know. Distributed knowledge is a useful notion in describing the total knowledge available to a group of agents in a distributed environment (cf. [DM90, FV86, HM90]).

In order to capture the notion of distributed knowledge in our language, we add a new modal operator D that stands for "distributed knowledge".[9] We can then capture distributed knowledge semantically as follows. Given a Kripke structure $M = (S, \pi, \mathcal{K}_1, \ldots, \mathcal{K}_n)$, we define

$(M, s) \models D\varphi$ iff $(M, t) \models \varphi$ for all t such that $(s, t) \in \mathcal{K}_1 \cap \ldots \cap \mathcal{K}_n$.

The intuition behind this definition is that if all the agents could "combine their knowledge", the only worlds they would consider possible are precisely those in the intersection of the sets of worlds that each one individually considers possible. Put another way, if some agent knows that a world t is not the real world, then the "wise man" should know this too. Thus, the wise man would only consider possible the worlds that all agents consider possible. Note that in the case of a single agent (i.e., $n = 1$), we have $D\varphi \equiv K\varphi$; distributed knowledge just reduces to knowledge.

How can we be sure that this definition really does capture our intuitions regarding distributed knowledge? One way is to find a complete axiomatization. If we view $D\varphi$ as saying "the wise man knows φ", one axiom that suggests itself is

A9. $K_i\varphi \Rightarrow D\varphi$, $i = 1, \ldots, n$;

this axiom is easily seen to be valid. We also expect the D operator to act like a knowledge operator, and indeed it is easy to see that it satisfies A2, with K_i replaced by D:

$$(D\varphi \wedge D(\varphi \Rightarrow \psi)) \Rightarrow D\psi.$$

[9] Again, all our results can be easily extended to the case where we have a modal operator D_G for each $G \subseteq \{1, \ldots, n\}$.

Moreover, if the $\mathcal{K}_i$ relations are reflexive, so that knowledge satisfies A3, then so does distributed knowledge; similar remarks hold for A4 and A5. Let K_n^D (resp., T_n^D, $\mathrm{S4}_n^D$, $\mathrm{S5}_n^D$) be the system that results from adding axiom A9 to the axiom system K_n (resp., T_n, $\mathrm{S4}_n$, $\mathrm{S5}_n$) and assuming that D also satisfies the axioms A2, and whichever of A3, A4, and A5 are applicable. Then we have

Theorem 5.1: *For the language of distributed knowledge with $n \geq 2$ agents, K_n^D (resp., T_n^D, $S4_n^D$, $S5_n^D$) is a sound and complete axiomatization with respect to $\mathcal{M}_n$ (resp., $\mathcal{M}_n^r$, $\mathcal{M}_n^{rt}$, $\mathcal{M}_n^{rst}$).*

Proof: Constructing the canonical model for maximal K_n^D-consistent sets, regarding the distributed knowledge operator D as if it were another K_j operator, results in the edges corresponding to D's possibility relation $\mathcal{K}_D$ being a subset of $\mathcal{K}_1 \cap \mathcal{K}_2 \cap \ldots \cap \mathcal{K}_n$. By making multiple copies of states in the canonical model that are in $\bigcap \mathcal{K}_i$ and not in $\mathcal{K}_D$, it is possible to construct an equivalent structure in which $\bigcap \mathcal{K}_i$ and $\mathcal{K}_D$ coincide. The same holds for T_n^D, $\mathrm{S4}_n^D$, and $\mathrm{S5}_n^D$. Details of the proof can be found in [FHV92]. ∎

We remark that if $n = 1$, we can get a complete axiomatization for distributed knowledge simply by adding the axiom $D\varphi \equiv K\varphi$ to the axioms for knowledge.

In the discussion above, we also viewed distributed knowledge as the knowledge the agents would have by pooling their individual knowledge together. This suggests the following rule of inference:

R4. From $\vdash (\psi_1 \wedge \ldots \wedge \psi_n) \Rightarrow \varphi$ infer $\vdash (K_1\psi_1 \wedge \ldots \wedge K_n\psi_n) \Rightarrow D\varphi$.

Again, this inference rule is easily seen to preserve validity. Intuitively it says that if $\psi = \psi_1 \wedge \ldots \wedge \psi_n$ implies φ, and if each of the agents knows a "part" of ψ (in particular, agent i knows ψ_i), then together they have distributed knowledge of ψ, and thus distributed knowledge of φ.

It is easy to check that this inference rule is derivable from axiom A2 (with D substituted for K_i) and A9 by propositional reasoning. Conversely, A9 is derivable from R4 and the other axioms for knowledge. Thus, we can replace A9 by R4 and get another complete axiomatization for distributed knowledge. We omit details here.

6 Deciding the satisfiability of formulas

In this section we examine the inherent difficulty of determining whether a formula in a given logic is satisfiable. Of course, the problem of determining validity is a closely related one, since φ is valid iff $\neg\varphi$ is not satisfiable. We consider this problem in terms of computational complexity. We briefly review the necessary notions here; the reader should consult [HU79] for further details.

Formally, we view everything in terms of the difficulty of determining membership in a set. Thus, the validity problem is viewed as the problem of determining whether a given formula φ is an element of the set of formulas valid with respect to a class of structures. The difficulty of determining set membership is usually measured by the amount of time and/or space (memory) required to do this, as a function of the input size. Since the inputs we consider in this section are formulas, we will typically be interested in the difficulty of determining whether a formula φ is valid or satisfiable as a function of $|\varphi|$. We are usually most interested in *deterministic* computations, where at any point in a computation, the next step of the computation is uniquely determined. However, thinking in terms *nondeterministic* computations—ones where the program may "guess" which of a finite number of steps to take—has been very helpful in classifying the intrinsic difficulty of a number of problems. The complexity classes we will be most concerned with here are *P*, *PSPACE*, *EXPTIME*, and *NP*: those sets such that determining whether a given element x is a member of the set can be done in deterministic polynomial time, deterministic polynomial space, deterministic exponential time, and nondeterministic polynomial time, respectively (as a function of the size of x). It is not hard to show that $P \subseteq NP \subseteq PSPACE \subseteq EXPTIME$; it is also known that $P \neq EXPTIME$. While it is conjectured that all the other inclusions are strict, proving this remains elusive. The $P = NP$ problem is currently considered the most important open problem in the field of computational complexity. It is perhaps worth mentioning that it is known that $PSPACE = NPSPACE$; that is, set membership can be determined in deterministic polynomial space if and only if it can be determined in nondeterministic polynomial space. Nondeterminism does not add any power at the level of polynomial space.

Roughly speaking, a set A is said to be *hard* with respect to a complexity class $\mathcal{C}$ (e.g., *NP*-hard, *PSPACE*-hard, etc.) if every set in $\mathcal{C}$ can be effectively *reduced* to A; i.e., for any set B in $\mathcal{C}$, an algorithm deciding membership in B can be easily obtained from an algorithm for deciding membership in A. A set is *complete* with respect to a complexity class $\mathcal{C}$ if it is both in $\mathcal{C}$ and $\mathcal{C}$-hard.

A well-known result due to Cook [Coo71] shows that the problem of determining whether a formula of propositional logic is satisfiable (i.e., the problem of determining whether a given propositional formula is in the set of satisfiable propositional formulas) is *NP*-complete. In particular, this means that if we could find a polynomial-time algorithm for deciding satisfiability for propositional logic, we would also have polynomial-time algorithms for all other *NP* problems. This is considered highly unlikely.

Given a complexity class $\mathcal{C}$, the class co-$\mathcal{C}$ consists of all of the sets whose *complement* is a member of $\mathcal{C}$. Notice that if we have a deterministic algorithm M for deciding membership in a set A, then it is easy to convert it to an algorithm M' for deciding membership in the complement of A that runs in the same space and/or time bounds: M' accepts an input x iff M rejects. It follows that $\mathcal{C} = \text{co}-\mathcal{C}$ must hold for every deterministic complexity class $\mathcal{C}$. This is not necessarily the case for a nondeterministic algorithm, since in this case we say that the algorithm accepts an input if it accepts for some appropriate sequence of guesses. There is no obvious way to construct an algorithm M' that will accept an element of the complement of A by an appropriate sequence of guesses. Thus, in particular, it is not known whether $NP = co\text{-}NP$. Clearly, if $P = NP$, then it would immediately follow that $NP = \text{co-}NP$, but it is conjectured that in fact $NP \neq \text{co-}NP$. By way of contrast, since $PSPACE = NPSPACE$, it follows that $NPSPACE = \text{co-}NPSPACE$.

Notice that validity and satisfiability are complementary problems. For example, since a propositional formula φ is valid exactly if $\neg\varphi$ is not satisfiable, it is easy to see that the validity problem for propositional logic is in co-*NP*; from the fact that satisfiability is *NP*-complete, it follows that validity is (co-*NP*)-complete. Analogous relationships hold between the satisfiability and validity problems for other logics.

The complexity of the validity and satisfiability problems for numerous other logics has been studied. It is remarkable how many of them can be completely characterized in terms of the complexity classes discussed above. Logics that are particularly relevant to us here are QBF, the logic of quantified Boolean formulas, for which the satisfiability problem was shown by Stockmeyer and Meyer to be *PSPACE*-complete [SM73] (QBF is described in a little more detail below) and PDL, propositional dynamic logic, for which the satisfiability problem was shown to be *EXPTIME*-complete by Fischer and Ladner (who proved the lower bound) and Pratt (who proved the matching upper bound) [FL79, Pra79]. (Observe that it follows from our earlier remarks that the validity problem for QBF is also *PSPACE*-complete and the validity problem for PDL is *EXPTIME*-complete.) Ladner [Lad77] also showed that in the case of one agent, the satisfiability problem for the logics K, T, and S4 are *PSPACE*-complete, while for S5, the satisfiability problem is *NP*-complete. We extend Ladner's results here to the case of many agents and common knowledge. As we shall see, in the case of S5, going from one agent to many agents increases the complexity of the logic (provided that $PSPACE \neq NP$); adding common knowledge causes a further increase in complexity.

Our results are summarized in the Table 1. The results in the table are stated in terms of the satisfiability problem, and are all tight. By our comments above, we can easily restate the results in terms of the validity problem. For example, from the table, we see that the satisfiability problem for $S5_2$ is *PSPACE*-complete: There is an algorithm for deciding whether a formula is satisfiable with respect to $\mathcal{M}_2^{rst}$ (or, equivalently, whether it is $S5_2$ consistent) that runs in polynomial space, and any *PSPACE* problem can be efficiently reduced to the satisfiability problem for $S5_2$. Since *PSPACE* is closed under complement, it follows that the validity problem for $S5_2$ is also *PSPACE*-complete.[10]

We remark that we do not mention the case for languages involving distributed knowledge in our table. This is because adding distributed knowledge to the language does not affect the complexity. Thus, for example, the complexity of the satisfiability problem for $S5_n$ is the same as that for $S5_n^D$. Similarly, we do not mention the single-agent case for $S4^C$, $S5^C$, and $KD45^C$, since in these cases common knowledge reduces to knowledge.[11]

Our upper bound proofs are constructive. For example, we show that the satisfiability problem for $S5_2$ is in *PSPACE* by actually providing a polynomial space algorithm for checking whether an arbitrary formula is $S5_2$ satisfiable. (Of course, the amount of space used by the algorithm is polynomial in the size of the formula.) Our lower bound proofs show that the upper bounds we obtain are essentially optimal. *Any* algorithm for checking $S5_2$ satisfiability

[10] We remark that these results are incorrectly stated in [McA88], a survey on logics of knowledge and belief. In particular, it is stated there (inaccurately citing [HM85]) that the validity problem for K_n, T_n, etc. is *co-PSPACE*-complete, while the validity problem for K_n^C, T_n^c, etc. is *co-EXPTIME*-complete. As we observed above, since *PSPACE* and *EXPTIME* are deterministic complexity classes, co-*PSPACE* is identical to *PSPACE* and co-*EXPTIME* is identical to *EXPTIME*. In addition, on p. 228 of [McA88], it is stated that "It is known that any *NP*-complete problem is *PSPACE*-complete. This is certainly not known and, as we have indicated above, is believed to be false.

[11] We could also consider a language that combined common knowledge and distributed knowledge. The complexity for this language is identical to that of the language with only common knowledge (exponential time complete except in single-agent case for S5 and KD45). No additional technical difficulties arise in this case, so we omit it here.

$S5_1$, $KD45_1$	K_n, T_n, $S4_n$, $n \geq 1$; $S5_n$, $KD45_n$, $n \geq 2$	K_n^C, T_n^C, $n \geq 1$; $S4_n^C$, $S5_n^C$, $KD45_n^C$, $n \geq 2$
NP-complete	*PSPACE*-complete	*EXPTIME*-complete

Table 1: The complexity of the satisfiability problem for logics of knowledge

must take polynomial space for infinitely many inputs.

Our lower bounds suggest that we cannot hope for automatic theorem provers for these logics that are guaranteed to work well (in the sense of providing the right answer quickly) for all inputs. We return to this point in Section 7. In the remainder of this section, we prove the results described in Table 1, as well as further properties of models for these logics.

We remark that, in practice, we may not be interested in checking satisfiability or validity for arbitrary formulas; instead, we may be interested only in a particular subset of formulas. In some papers, a case has been made that we should be particularly interested in the validity of formulas of the form $K_i\varphi \Rightarrow K_i\psi$, since such formulas allow us to express the fact that an agent (or knowledge base) whose knowledge is characterized by φ will also know ψ.[12] Of course, our upper bound results immediately apply to formulas of the form $K_i\varphi \Rightarrow K_i\psi$ (or any other subclass of formulas). It turns out that our lower bounds apply to this set of formulas as well. To see this, note that if we take φ to be the formula *true* then, since $K_i true$ is equivalent to *true* in all the logics we have considered, it follows that any algorithm that checks validity of formulas for all formulas of the form $K_i\varphi \Rightarrow K_i\psi$ must in particular be able to check the validity of formulas of the form $K_i\psi$. It is not hard to check that for all the logics we have considered, $K_i\psi$ is valid if and only if ψ is valid. Thus, the validity problem for the subclass of formulas $K_i\varphi \Rightarrow K_i\psi$ is equivalent in difficulty to the full validity problem. (Notice that it may in some cases be interesting to study the complexity of deciding the validity of $K_i\varphi \Rightarrow K_i\psi$ where φ and ψ are themselves of a restricted form; depending on the particular form of these subformulas, the decision problem may sometimes be more manageable.)

6.1 *NP*-completeness results for S5 and KD45

Since propositional logic is included in all the logics we have considered, the satisfiability problem for all of them is *NP*-hard. Ladner showed that, at least for S5, it is no harder.

Theorem 6.1: *([Lad77]) The satisfiability problem for S5 is NP-complete (and thus the validity problem for S5 is co-NP-complete).*

The key step in the proof of Theorem 6.1 lies in showing that satisfiable S5 formulas can be satisfied in structures with very few states.

Proposition 6.2: *([Lad77]) An S5 formula φ is satisfiable iff it is satisfiable in a structure in $\mathcal{M}_1^{rst}$ with at most $|\varphi|$ states.*

[12] This issue particularly arises in when we try to capture what Levesque has called the *subjective* interpretation of logic—namely, that a sentence being in a theory means that it is believed by the agent—in an *objective* interpretation. See [Lev90] for further discussion of this issue.

Proof: Suppose $(M,s) \models \varphi$. By Proposition 2.7, we can assume without loss of generality that $M = (S, \pi, \mathcal{K})$, where $(t,t') \in \mathcal{K}$ for all $t,t' \in S$. Let F be the set of subformulas of φ of the form $K\psi$ for which $(M,s) \models \neg K\psi$; i.e., F is the set of subformulas of φ that have the form $K\psi$ and are false at the state s. For each formula $K\psi \in F$, there must be a state $s_\psi \in S$ such that $(M, s_\psi) \models \neg\psi$. Let $M' = (S', \pi', \mathcal{K}')$, where $S' = \{s\} \cup \{s_\psi : \psi \in F\}$, π' is the restriction of π to S', and $\mathcal{K}' = \{(t,t') : t,t' \in S'\}$. Since $|F| < |Sub(\varphi)| \le |\varphi|$, it follows that $|S'| \le |\varphi|$. We now show that for all states $s' \in S'$ and for all subformulas ψ of φ (including φ itself), $(M,s') \models \psi$ iff $(M',s') \models \psi$. As usual, we proceed by induction on the structure of ψ. The only nontrivial case is when ψ is of the form $K\psi'$. Suppose $s' \in S'$. If $(M,s') \models K\psi'$, then $(M,t) \models \psi'$ for all $t \in S$, so, in particular, $(M,t') \models \psi'$ for all $t' \in S'$. By the induction hypothesis, $(M',t') \models \psi'$ for all $t' \in S'$, so $(M',s') \models K\psi'$. And if $(M,s') \not\models K\psi'$, then $(M,s') \models \neg K\psi'$. Since M is a model of S5, we have $(M,s') \models K\neg K\psi'$, so that $(M,s) \models \neg K\psi'$ (since $(s',s) \in \mathcal{K}$ by assumption). But then it follows that $K\psi' \in F$, and $(M, s_{\psi'}) \models \neg\psi'$. By construction, $s_{\psi'} \in S'$, and by induction hypothesis, we also have $(M', s_{\psi'}) \models \neg\psi'$. Since $(s', s_{\psi'}) \in \mathcal{K}'$, we have $(M',s') \models \neg K\psi'$, and so $(M',s') \not\models K\psi'$ as desired. This concludes the proof that $(M,s') \models \psi$ iff $(M',s') \models \psi$ for all subformulas ψ of φ and all $s' \in S'$. Since $s \in S'$ and $(M,s) \models \varphi$ by assumption, we also have $(M',s) \models \varphi$. ∎

Proof of Theorem 6.1: Because the propositional calculus is part of S5, Cook's Theorem [Coo71] implies that deciding S5 satisfiability is *NP*-hard. We now give an *NP* algorithm for deciding S5 satisfiability. Intuitively, given a formula φ, we simply guess a structure $M \in \mathcal{M}_1^{rst}$ with a universal possibility relation and at most $|\varphi|$ states, and verify that φ is satisfied in M. More formally, we proceed as follows. Given a formula φ, where $|\varphi| = m$, we nondeterministically guess a Kripke structure $M = (S, \pi, \mathcal{K})$, where S is a set of $k \le m$ states, $(s,t) \in \mathcal{K}$ for all $s,t \in S$, and for all $s \in S$ and primitive propositions p not appearing in φ, $\pi(s)(p) = \mathbf{true}$. (Note that the only "guessing" that enters here is in the choice of k, and of the truth values $\pi(s)(q)$ that the primitive propositions q appearing in φ have in the k states of S.) Since at most m primitive propositions appear in φ, guessing such a Kripke structure can be done in nondeterministic time $O(m^2)$ (i.e., at most cm^2 steps for some constant c). Next, we check whether φ is satisfied at some state $s \in S$. By Proposition 3.1, this can be done deterministically in time $O(m^2)$. By Proposition 6.2, if φ is satisfiable, one of our guesses is bound to be right. (Of course, if φ is not satisfiable, no guess will be right.) Thus, we have a nondeterministic $O(m^2)$ algorithm for deciding if φ is satisfiable. ∎

We can prove essentially the same results for KD45 as for S5. Using Proposition 6.2, we can show

Proposition 6.3: *A KD45 formula φ is satisfiable iff it is satisfiable in a structure in $\mathcal{M}_1^{elt}$ with at most $|\varphi|$ states.*

Using Proposition 6.3 just as we used Proposition 3.1, we can now prove

Theorem 6.4: *The satisfiability problem for KD45 is NP-complete (and thus the validity problem for KD45 is (co-NP)-complete).*

We leave details of the proofs of the latter two results to the reader.

6.2 *PSPACE* lower bounds

As the following result shows, we cannot prove an analogue of Proposition 6.2 for the logics K, T, and S4.

Proposition 6.5: *There is a formula* φ_m^K *(resp.,* φ_m^T*,* φ_m^{S4}*) of size* $O(m^2)$ *(resp.,* $O(m)$*,* $O(m)$*) that is K (resp. T, S4) satisfiable, but every structure in* $\mathcal{M}_1$ *(resp.,* $\mathcal{M}_1^r$*,* $\mathcal{M}_1^{rt}$*) that satisfies it has at least* 2^m *states.*

Proof: We construct φ_m^{S4} first, and then point out the necessary modifications for constructing φ_m^T and φ_m^K. We construct the formula φ_m^{S4} so that it essentially forces the existence of a binary tree of depth m, each of whose leaves encodes a distinct truth assignment to the primitive propositions $p_1, \ldots, p_m$. In addition, we have primitive propositions $d_0, \ldots, d_{m+1}$. Intuitively, d_i is true exactly if we are at a depth $\geq i$ in the tree. Let *depth* be the following formula, which clearly captures the intended relation between the d_i's:

$$\bigwedge_{i=1}^{m+1} (d_i \Rightarrow d_{i-1}).$$

Let *determined* be a formula which intuitively says that the truth value of the proposition p_i is determined by depth i in the tree, in that if p_i is true (resp. false) at a given node s of depth j with $j \geq i$, then it is true (resp. false) at all the successors of s of depth at least i. (If we deal with trees, then all successors of s will have depth at least i; we only put this caveat in the formula to deal with our later extension of these ideas to $S5_2$, where things get a bit more complicated.) We take *determined* to be an abbreviation for:

$$\bigwedge_{i=1}^{m} (d_i \Rightarrow ((p_i \Rightarrow K(d_i \Rightarrow p_i)) \wedge (\neg p_i \Rightarrow K(d_i \Rightarrow \neg p_i)))).$$

We take *branching* to be a formula that intuitively says that for any node at depth i, it is possible to find two successor nodes at depth $i+1$ such that p_{i+1} is true at one and false at the other. More formally, we take *branching* to be an abbreviation for:

$$\bigwedge_{i=0}^{m-1} ((d_i \wedge \neg d_{i+1}) \Rightarrow (\neg K \neg (d_{i+1} \wedge \neg d_{i+2} \wedge p_{i+1}) \wedge \neg K \neg (d_{i+1} \wedge \neg d_{i+2} \wedge \neg p_{i+1}))).$$

Finally, we take φ_m^{S4} to be

$$d_0 \wedge \neg d_1 \wedge K(\mathit{depth} \wedge \mathit{determined} \wedge \mathit{branching}).$$

It is easy to see that $|\varphi_m^{S4}|$ is $O(m)$ and φ_m^{S4} is satisfiable. Moreover, if $M = (S, \pi, \mathcal{K}) \in \mathcal{M}_1^{rt}$ and $(M, s) \models \varphi_m^{S4}$, then we can show by induction on j that if $j \leq m$ and v is a truth assignment to the propositions $p_1, \ldots, p_j$, then there is a state t reachable from s such that $(M, t) \models d_j \wedge \neg d_{j+1}$ (so that t is at depth j) and $(M, t) \models p_i$ iff $v(p_i) = \mathbf{true}$, for $i = 1, \ldots, j$. For the base case, note that since $(M, s) \models d_0 \wedge \neg d_1 \wedge \mathit{branching}$, there must be successors t_0 and t_1 of s such that $(M, t_0) \models d_1 \wedge \neg d_2 \wedge p_1$ and $(M, t_1) \models d_1 \wedge \neg d_2 \wedge \neg p_1$. For the induction step, suppose that $j < m$ and that v is a truth assignment to $p_1, \ldots, p_{j+1}$. By the induction hypothesis, there is a state t reachable from s such that $(M,t) \models d_j \wedge \neg d_{j+1}$ and $(M,t) \models p_i$ iff $v(p_i) = \mathbf{true}$ for $i = 1, \ldots, j$. Since t is reachable from s and $M \in \mathcal{M}_1^{rt}$, we also have that $(M,t) \models \mathit{depth} \wedge \mathit{determined} \wedge \mathit{branching}$. The formula *branching* guarantees that there is a successor t' of t such that $(M,t') \models d_{j+1} \wedge \neg d_{j+2}$ and $(M,t') \models p_{j+1}$ iff $v(p_{j+1}) = \mathbf{true}$. The formula *depth* guarantees that $(M,t) \models d_1 \wedge \ldots \wedge d_j$. From this it follows that the formula *determined* guarantees that $(M,t') \models p_i$ iff $v(p_i) = \mathbf{true}$ for $i = 1, \ldots, j$. This completes the proof of the induction step. Taking $j = m$, it follows that there are at least 2^m distinct states in M where the formula d_m is true.

The key property of φ_m^{S4} is that if $(M,s) \models \varphi_m^{S4}$ and $M \in \mathcal{M}_1^{rt}$, then for every node t reachable in at most m steps from s, we have $(M,t) \models \mathit{depth} \wedge \mathit{determined} \wedge \mathit{branching}$. In order for this property to hold in arbitrary structures, we take φ_m^K to be

$$d_0 \wedge \neg d_1 \wedge \bigwedge_{i=0}^{m} K^i(\mathit{depth} \wedge \mathit{determined} \wedge \mathit{branching}),$$

where $K^0\psi$ is an abbreviation for ψ, and $K^{i+1}\psi$ to be an abbreviation for $KK^i\psi$. In models of T_n, we have $K^i\psi \Rightarrow K^{i-1}\psi$ for $i \geq 1$. Thus, we can take φ_m^T to be

$$d_0 \wedge \neg d_1 \wedge K^m(\mathit{depth} \wedge \mathit{determined} \wedge \mathit{branching}).$$

Note that $|\varphi_m^T|$ is $O(m)$, but $|\varphi_m^K|$ is $O(m^2)$. We leave it to the reader to verify that these formulas have the right properties. ∎

We know from Theorems 3.2 and 3.4 that if a formula φ is K_n (resp., T_n, $\mathrm{S4}_n$, $\mathrm{S5}_n$) satisfiable, then it is satisfiable in a structure of size $\leq 2^{|\varphi|}$. Proposition 6.5 tells us that in the case of K, T, and S4 (and, *a fortiori*, in the case of K_n, T_n, and $\mathrm{S4}_n$ for $n \geq 2$) we can essentially do no better. On the other hand, Proposition 6.2 says that we can do much better for S5. The reader may wonder why the construction of Proposition 6.5 does not also work for S5. While it is hard to give a completely precise answer to this question, it may help to note that formula φ_m^{S4} is not even S5 satisfiable. For suppose it were. By Proposition 2.7, we can assume without loss of generality that it is satisfiable in a structure, say $M = (S, \pi, \mathcal{K})$, where $\mathcal{K}$ is the universal relation. The formula *branching* forces there to be two states s and t in M such that $(M,s) \models d_1 \wedge p_1$ and $(M,t) \models d_1 \wedge \neg p_1$. But since we must have $(M,s) \models \mathit{determined}$, in particular $(M,s) \models (d_1 \wedge p_1) \Rightarrow Kp_1$. Since we have $(s,t) \in \mathcal{K}$, this means $(M,t) \models p_1$, a contradiction.

Proposition 6.5 suggests that we will not be able to get an *NP* decision procedure for the satisfiability problem for T, K and S4. Ladner gives strong support to this conjecture by showing that the satisfiability problem for these logics is *PSPACE*-hard. His proof (which we sketch below) uses a variant of the formula constructed in Proposition 6.5. We also show how to modify his proof to get a *PSPACE*-hardness result for $\mathrm{S5}_2$ and $\mathrm{KD45}_2$. This suggests that for S5 and KD45, deciding satisfiability in the multi-agent case is significantly more difficult than in the single-agent case.

Theorem 6.6:

1. *[Lad77] The satisfiability problem for the logics K, T, and S4 is PSPACE-hard.*

2. *The satisfiability problem for $S5_2$ and $KD45_2$ is PSPACE-hard.*

Proof: Following Ladner, we consider the logic of *quantified Boolean formulae* (QBF). For our purposes, we can take a QBF to be of the form $Q_1p_1Q_2p_2\ldots Q_mp_mA'$, where $Q_i \in \{\forall, \exists\}$ and A' is a propositional formula whose only primitive propositions are among $p_1, \ldots, p_m$. Thus, a typical QBF is $\forall p_1 \exists p_2(p_1 \Rightarrow p_2)$. We can determine whether a QBF is true or false by successively replacing each subformula of the form $\forall p_i(B)$ by $B_0 \wedge B_1$ and each subformula of the form $\exists p_i(B)$ by $B_0 \vee B_1$, where B_0 (resp. B_1) is B with all occurrences of p_i replaced by *true* (resp. *false*), and then using the standard rules of propositional logic. Note that this successive replacement results in a formula that may be much larger than the original formula (in fact, exponential in the size of the original formula). It is known that the problem of determining which QBF are true is *PSPACE*-complete [SM73]. Ladner proves the *PSPACE* lower bound by reducing the problem of deciding whether a QBF is true to that of deciding whether a formula is K (resp., T, S4) satisfiable. We present a slight variant of his proof here, and show how to modify it to deal with $S5_2$ and $KD45_2$.

We deal with S4 first. Suppose we are given a QBF formula $A = Q_1p_1\ldots Q_mp_mA'$. We construct a formula ψ_A^{S4} that is satisfiable in a structure in $\mathcal{M}_1^{rt}$ iff A is true. The construction of ψ_A^{S4} is very similar to that of the formula φ_m^{S4} constructed in the proof of Proposition 6.5, except that instead of forcing all possible truth assignments to the p_i's, ψ_A^{S4} just forces those truth assignments necessary to show that A is true. We proceed as follows.

Again we take as primitive propositions $p_1, \ldots, p_m, d_0, \ldots, d_{m+1}$, where d_i denotes depth at least i in a "tree" of truth assignments. We take the formulas *depth* and *determined* to be just as in the proof of Proposition 6.5, namely

$$depth =_{\text{def}} \bigwedge_{i=1}^{m+1} (d_i \Rightarrow d_{i-1}) \text{ and}$$

$$determined =_{\text{def}} \bigwedge_{i=1}^{m} (d_i \Rightarrow ((p_i \Rightarrow K(d_i \Rightarrow p_i)) \wedge (\neg p_i \Rightarrow K(d_i \Rightarrow \neg p_i)))).$$

We modify *branching* to $branching_A$, which is an abbreviation for

$$\bigwedge_{\{i:Q_{i+1}=\forall\}} ((d_i \wedge \neg d_{i+1}) \Rightarrow (\neg K \neg (d_{i+1} \wedge \neg d_{i+2} \wedge p_{i+1}) \wedge \neg K \neg (d_{i+1} \wedge \neg d_{i+2} \wedge \neg p_{i+1}))) \wedge$$

$$\bigwedge_{\{i:Q_{i+1}=\exists\}} ((d_i \wedge \neg d_{i+1}) \Rightarrow (\neg K \neg (d_{i+1} \wedge \neg d_{i+2} \wedge p_{i+1}) \vee \neg K \neg (d_{i+1} \wedge \neg d_{i+2} \wedge \neg p_{i+1}))).$$

Finally, we take ψ_A^{S4} to be

$$d_0 \wedge \neg d_1 \wedge K(depth \wedge determined \wedge branching_A \wedge (d_m \Rightarrow A')).$$

It is easy to see that ψ_A^{S4} is satisfiable in a structure in $\mathcal{M}_1^{rt}$ if A is true. Conversely, suppose that $M = (S, \pi, \mathcal{K}) \in \mathcal{M}_1^{rt}$ and $(M, s) \models \psi_A^{S4}$. Given a state t in M, let A_j^t be the QBF that results by starting with $Q_{j+1}p_{j+1}\ldots Q_mp_mA'$ and replacing all occurrences of p_i, $i < j$, by *true* if $\pi(t)(p_i) = \mathbf{true}$, and by *false* otherwise. Note that $A_0^t = A$, and A_m^t is the result of starting with A' and replacing all the p_i's by *true* or *false* as appropriate. The fact that

$(M,s) \models K(d_m \Rightarrow A')$ implies that if $(s,t) \in \mathcal{K}$ and $(M,t) \models d_m$, then A^t_m is true. An easy induction on j now shows that if $(s,t) \in \mathcal{K}$ and $(M,t) \models d_{m-j} \wedge \neg d_{m-j+1}$, then the QBF A^t_{m-j} is true. Since $(M,s) \models d_0$, in particular we have that $A^s_0 = A$ is true.

Since, by Theorem 2.6, a formula is S4 satisfiable iff it is satisfiable in a structure in $\mathcal{M}_1^{rt}$, it follows that ψ_A^{S4} is satisfiable iff A is true. Since the size of ψ_A^{S4} is linear in the size of A, it follows that S4 satisfiability is *PSPACE*-hard.

The modifications required to deal with K and T are similar to those in the proof of Proposition 6.5. We take ψ_A^T to be

$$d_0 \wedge \neg d_1 \wedge K^m(\mathit{depth} \wedge \mathit{determined} \wedge \mathit{branching}_A \wedge (d_m \Rightarrow A'))$$

and ψ_A^K to be

$$d_0 \wedge \neg d_1 \wedge \bigwedge_{i=0}^{m} K^i(\mathit{depth} \wedge \mathit{determined} \wedge \mathit{branching}_A \wedge (d_m \Rightarrow A')).$$

Similar arguments to those used for ψ_A^{S4} can now be used to show that A is true iff ψ_A^T (resp. ψ_A^K) is satisfiable in a reflexive structure (resp., satisfiable in a Kripke structure).

Finally, we take both ψ_A^{S5} and ψ_A^{KD45} to be the result of replacing all occurrences of K in ψ_A^T by K_1K_2. Now suppose A is satisfiable. We want to show that ψ_A^{S5} is satisfiable in a structure in $\mathcal{M}_2^{rst}$. We start with the obvious structure $M = (S, \pi, \mathcal{K}) \in \mathcal{M}_1^{rt}$ which satisfies ψ_A^{S4}. M looks like a tree, and at the root s_0 we have $(M, s_0) \models \psi_A^{S4}$. Intuitively, to get a structure in $\mathcal{M}_2^{rst}$ satisfying ψ_A^{S5}, we simply replace every $\mathcal{K}$ edge in M by two edges, the first in $\mathcal{K}_1$ and the second in $\mathcal{K}_2$. More formally, let S' consist of S together with a new node $s_{(s,t)}$ for every edge $(s,t) \in \mathcal{K}$. Define π' on the states of S' so that it agrees with π on S, and so that $\pi'(s_{s,t}) = \pi(s)$. Finally define $\mathcal{K}_1$ to be the reflexive, symmetric, transitive closure of the set of edges $\{(s, s_{(s,t)}) : (s,t) \in \mathcal{K}\}$ and $\mathcal{K}_2$ to be the reflexive, symmetric, transitive closure of $\{(s_{(s,t)}, t) : (s,t) \in \mathcal{K}\}$. By construction, $\mathcal{K}_1$ and $\mathcal{K}_2$ are equivalence relations. We leave it to the reader to check that $(M', s_0) \models \psi_A^{S5}$, where $M' = (S', \pi', \mathcal{K}_1, \mathcal{K}_2)$. In fact, since M' is also in $\mathcal{M}_2^{elt}$ and $\psi_A^{KD45} = \psi_A^{S5}$, the result follows for ψ_A^{KD45}. The proof for the other direction follows the same lines as the proof in the case of ψ_A^{S4}, except that instead of considering edges $(s,t) \in \mathcal{K}$, we consider $\mathcal{K}_1 \circ \mathcal{K}_2$ (i.e., the composition of the relations $\mathcal{K}_1$ and $\mathcal{K}_1$, which consists of all edges (s,t) such that for some u, we have $(s,u) \in \mathcal{K}_1$ and $(u,t) \in \mathcal{K}_2$). We leave further details to the reader. ∎

PSPACE lower bounds for all the logics K_n, T_n, $S4_n$, $n \geq 1$, and $S5_n$, $KD45_n$, $n \geq 2$, follow immediately from Theorem 6.6.

6.3 *PSPACE* decision procedures using the tableau method

In this subsection, we prove upper bounds that match the lower bounds of the previous subsection. All the decision procedures presented are essentially generalizations of Ladner's procedures for K, T, and S4 [Lad77]. The details are quite technical and can be skipped on a first reading. Ladner's construction is based on the *tableau* method, which was first developed for deciding satisfiability in the propositional calculus [Bet59, Smu68], and first applied to modal logics by Kripke [Kri63]. We briefly review the method in the propositional case.

A *propositional tableau* is a set T of formulas such that

1. if $\neg\neg\psi \in T$ then $\psi \in T$,
2. if $\psi \wedge \psi' \in T$ then both ψ, $\psi' \in T$,
3. if $\neg(\psi \wedge \psi') \in T$ then either $\neg\psi \in T$ or $\neg\psi' \in T$, and
4. it is not the case that both ψ and $\neg\psi$ are in T for some formula ψ.

We say that T is a *propositional tableau for* φ if T is a propositional tableau and $\varphi \in T$. It is easy to see that we have

Lemma 6.7: *The propositional formula φ is satisfiable if and only if there is a propositional tableau for φ.*

We want to extend the notion of a propositional tableau to a tableau for a modal logic. A K_n *tableau* is a tuple $T = (S, L, \mathcal{K}_1, \ldots, \mathcal{K}_n)$, where, as for a structure, S is a set of states and $\mathcal{K}_1, \ldots, \mathcal{K}_n$ are possibility relations, while L is a *labeling function* that associates with each state $s \in S$ a set $L(s)$ of formulas such that

1. $L(s)$ is a propositional tableau,
2. if $K_i\psi \in L(s)$ and $(s,t) \in \mathcal{K}_i$, then $\psi \in L(t)$, and
3. if $\neg K_i\psi \in L(s)$, then there exists t with $(s,t) \in \mathcal{K}_i$ and $\neg\psi \in L(t)$.

We say that $T = (S, L, \mathcal{K}_1, \ldots, \mathcal{K}_n)$ is a K_n *tableau for* φ if T is a K_n tableau and $\varphi \in L(s)$ for some state $s \in S$. A T_n *tableau* is a K_n tableau that satisfies in addition

4. if $K_i\psi \in L(s)$ then $\psi \in L(s)$.

An *$S4_n$ tableau* is a T_n tableau that satisfies in addition

5. if $K_i\psi \in L(s)$ and $(s,t) \in \mathcal{K}_i$, then $K_i\psi \in L(t)$.

An *$S5_n$ tableau* is a T_n tableau that satisfies in addition

6. if $(s,t) \in \mathcal{K}_i$ then $K_i\psi \in L(s)$ iff $K_i\psi \in L(t)$.

Clearly clause (6) implies clause (5), so an $\mathrm{S5}_n$ tableau is automatically an $\mathrm{S4}_n$ tableau.

Finally, a *$KD45_n$ tableau* is a K_n tableau that satisfies in addition

7. (a) if $(s,t), (s,u) \in \mathcal{K}_i$, and $K_i\psi \in L(t)$ then both $K_i\psi \in L(u)$ and $\psi \in L(u)$; (b) if $K_i\psi \in L(s)$, then either $\psi \in L(s)$ or there exists t with $(s,t) \in \mathcal{K}_i$; and (c) if $K_i\psi \in L(s)$ and $(s,t) \in \mathcal{K}_i$, then $K_i\psi \in L(t)$.

Intuitively, clause (a) corresponds to the Euclidean property, clause (b) corresponds to seriality, while clause (c) corresponds to transitivity (compare clause 7(c) to clause (5) for $\mathrm{S4}_n$ tableaus).

We say $T = (S, L, \mathcal{K}_1, \ldots, \mathcal{K}_n)$ is a K_n (resp., T_n, $S4_n$, $S5_n$, $KD45_n$) *tableau for* φ if T is a K_n (resp., T_n, $\mathrm{S4}_n$, $\mathrm{S5}_n$, $\mathrm{KD45}_n$) tableau and $\varphi \in L(s)$ for some state $s \in S$.

Proposition 6.8: *The formula φ is K_n (resp., T_n, $S4_n$, $S5_n$, $KD45_n$) satisfiable if and only if there is a K_n (resp., T_n, $S4_n$, $S5_n$, $KD45_n$) tableau for φ.*

Proof: We do the case of $S5_n$ here. All the other cases are similar (and easier). If φ is $S5_n$ satisfiable, suppose it is satisfied in the structure $M = (S, \pi, \mathcal{K}_1, \ldots, \mathcal{K}_n)$. Let $T = (S, L, \mathcal{K}_1, \ldots, \mathcal{K}_n)$, where $L(s) = \{\psi : (M, s) \models \psi\}$. It is easy to see that T is an $S5_n$ tableau for φ.

For the converse, suppose that $T = (S, L, \mathcal{K}_1, \ldots, \mathcal{K}_n)$ is an $S5_n$ tableau for φ. Let $M = (S, \pi, \mathcal{K}'_1, \ldots, \mathcal{K}'_n)$, where $\mathcal{K}'_i$ is the reflexive, symmetric, transitive closure of $\mathcal{K}_i$ (so that $M \in \mathcal{M}_2^{rst}$) and

$$\pi(s)(p) = \begin{cases} \textbf{true} & \text{if } p \in L(s) \\ \textbf{false} & \text{if } p \notin L(s). \end{cases}$$

We now show by induction on the structure of formulas that if $\psi \in Sub(\varphi)$, then $\psi \in L(s)$ implies $(M, s) \models \psi$ and $\neg\psi \in L(s)$ implies $(M, s) \models \neg\psi$. If ψ is a primitive proposition, the result follows immediately from the definition of π and the fact that $L(s)$ is a propositional tableau, so that we cannot have both p and $\neg p$ in $L(s)$. If ψ is of the form $\psi_1 \wedge \psi_2$ or $\neg\psi'$, then the result follows easily using the induction hypothesis and the fact that $L(s)$ is a propositional tableau. Finally, if ψ is of the form $K_i\psi'$, clearly if $\neg K_i\psi' \in L(s)$ then by clause (3) of the definition of K_n (and $S5_n$) tableau, there is some state t such that $(s, t) \in \mathcal{K}_i$ and $\neg\psi' \in L(t)$. Since $\mathcal{K}_i \subseteq \mathcal{K}'_i$ and, since by the induction hypothesis we have $(M, t) \models \neg\psi'$, we must have $(M, s) \models \neg K_i\psi'$. Finally, suppose $K_i\psi' \in L(s)$. We want to show that $(M, s) \models K_i\psi'$. It suffices to show that $(M, t) \models \psi'$ for all t such that $(s, t) \in \mathcal{K}'_i$. But since $\mathcal{K}'_i$ is the reflexive, symmetric, transitive closure of $\mathcal{K}_i$, if $(s, t) \in \mathcal{K}'_i$, then there must exist $k \geq 0$ and states $s_0, \ldots, s_k$, such that $s = s_0$, $t = s_k$, and for $j < k$, either $(s_j, s_{j+1}) \in \mathcal{K}_i$ or $(s_{j+1}, s_j) \in \mathcal{K}_i$. An easy induction on j, using clause (6) of the definition of $S5_n$ tableau, shows that we must have $K_i\psi' \in L(s_j)$ for all $j \leq k$. In particular, $K_i\psi' \in L(t)$. By clause (4), we also have $\psi' \in L(t)$. By the induction hypothesis, $(M, t) \models \psi'$, and we are done. ∎

Given a formula φ, we now present an algorithm that attempts to construct a K_n tableau for φ. We show that the construction succeeds if and only if φ is K_n satisfiable. Finally, we show that there is an algorithm that checks whether our tableau construction succeeds that runs in space polynomial in $|\varphi|$. We then show how to modify the construction to get decision procedures for T_n, $S4_n$, $S5_n$, and $KD45_n$.

Besides proving that the satisfiability problem is in polynomial space, this construction has a number of other payoffs. For one thing, it shows that a formula φ is $K_n$satisfiable if and only if it is satisfiable in a structure whose graph looks like a tree of low depth (at most $dep(\varphi)$). Moreover, we can actually show directly (independent of Theorem 2.3) that our tableau construction succeeds if and only if φ is K_n-consistent. This gives us an alternative proof of completeness of the axiom system K_n. Easy modifications give us completeness for T_n, $S4_n$, $S5_n$, and $KD45_n$.

The K_n tableau construction consists of four independent procedures. We define a set T of formulas to be *fully expanded* if for every formula $\varphi \in T$ and subformula ψ of φ, either $\psi \in T$ or $\neg\psi \in T$. The first procedure expands a set of formulas to a propositional tableau. The second constructs a fully expanded propositional tableau. (We do not need fully expanded sets to deal with K_n, but they are useful when dealing with $S5_n$, so we introduce them now for uniformity.)

The third procedure takes a node whose label is a fully expanded propositional tableau and creates successors to the node so as to satisfy clause (3) of the definition of K_n tableau. The fourth procedure checks for satisfiable labels.

The algorithm as presented seems to be nondeterministic. In particular, step 2 says "Repeat until none of [procedures] (a)–(d) applies." We do not intend the choice of which procedure to apply to be nondeterministic; rather, as we shall show, the correctness of the construction is independent of which choice is made. Thus, any deterministic implementation of these choices will work. In fact, in order to show that satisfiability is in polynomial space, we shall implement these choices in a particularly space-efficient manner.

We remark that, strictly speaking, the K_n tableau construction does not really construct a tableau. Rather, it constructs an object which we call a *pre-tableau* in which the desired tableau is embedded. As we shall see, the desired tableau consists of a subset of the nodes in our construction whose label is a fully expanded propositional tableau; all other nodes are ignored. The pre-tableau is a tree, with nodes labeled by sets of formulas (just as a tableau), and some edges labeled by agents. One of its important properties is that it has low depth: the pre-tableau constructed for the formula φ has depth polynomial in $|\varphi|$. It may seem somewhat surprising that we can check whether φ is satisfiable using only polynomial space, given that Proposition 6.5 shows that some satisfiable formulas are satisfied only in structures whose size is at least exponential in the size of the formula. The key point here is that although the pre-tableau has exponential size, the fact that it has low depth means that it can be traversed efficiently (using depth-first search), using only polynomial space.

If a set T of formulas is not a propositional tableau, then ψ is a *witness* to this if $\psi \in T$ and one of clauses (1)–(3) in the definition of propositional tableau does not apply to ψ (for example, clause (1) would not apply if ψ were of the form $\neg\neg\psi'$ and $\psi' \notin T$). Similarly, if a set T of formulas is not fully expanded, then ψ is a witness to this if ψ is a subformula of some formula $\varphi \in T$, and neither ψ nor $\neg\psi$ is in T. For convenience, we also assume that the formulas are ordered in some way (say by length, with some way of breaking ties), so that it makes sense to choose the "least witness" if there is a witness. Finally, we say that a set T is *blatantly inconsistent* if, for some formula ψ, both ψ and $\neg\psi$ are in T.

The K_n tableau construction for φ_0:

1. Construct a tree consisting of a single node s_0 (the "root"), with $L(s_0) = \{\varphi_0\}$.
2. Repeat until none of (a)–(d) below applies:
   (a) *Forming a propositional tableau*: if s is a leaf of the tree, $L(s)$ is not blatantly inconsistent, $L(s)$ is not a propositional tableau, and ψ is the least witness to this fact, then:
      i. if ψ is of the form $\neg\neg\psi'$, then create a successor s' of s (i.e., add a node s' to the tree and an edge from s to s') and set $L(s') = L(s) \cup \{\psi'\}$,
      ii. if ψ is of the form $\psi_1 \wedge \psi_2$, then create a successor s' of s and set $L(s') = L(s) \cup \{\psi_1, \psi_2\}$,
      iii. if ψ is of the form $\neg(\psi_1 \wedge \psi_2)$, then create two successors s_1 and s_2 of s and set $L(s_i) = L(s) \cup \{\neg\psi_i\}$, $i = 1, 2$,

(b) *Forming a fully expanded propositional tableau*: if s is a leaf of the tree, $L(s)$ is not blatantly inconsistent, $L(s)$ is not a fully expanded propositional tableau, and ψ is the least witness to this fact, then create two successors s' and s'' of s and set $L(s') = L(s) \cup \{\psi\}$, $L(s'') = L(s) \cup \{\neg\psi\}$.

(c) *Creating successor nodes:* If s is a leaf of the tree, $L(s)$ is not blatantly inconsistent, and $L(s)$ is a fully expanded propositional tableau, then for each formula of the form $\neg K_i\psi \in L(s)$ create an i-successor node s' (i.e., add the node s' to the tree and an edge from s to s' labeled i) and let $L(s') = L(s)/K_i \cup \{\neg\psi\}$. (Recall that if L is a set of formulas, then L/K_i consists of all those formulas ψ such that $K_i\psi \in L$).

(d) *Marking nodes "satisfiable"*: If s is not marked "satisfiable" then mark s satisfiable if either (i) $L(s)$ is not a fully expanded propositional tableau and s' is marked "satisfiable" for some successor s' of s, (ii) $L(s)$ is a fully expanded propositional tableau, there are no formulas of the form $\neg K_i\psi \in L(s)$, and $L(s)$ is not blatantly inconsistent, or (iii) $L(s)$ is a fully expanded propositional tableau, s has successors, and all of them are marked "satisfiable".

3. If the root of the tree is marked "satisfiable", then return "φ_0 is satisfiable"; otherwise return "φ_0 is unsatisfiable".

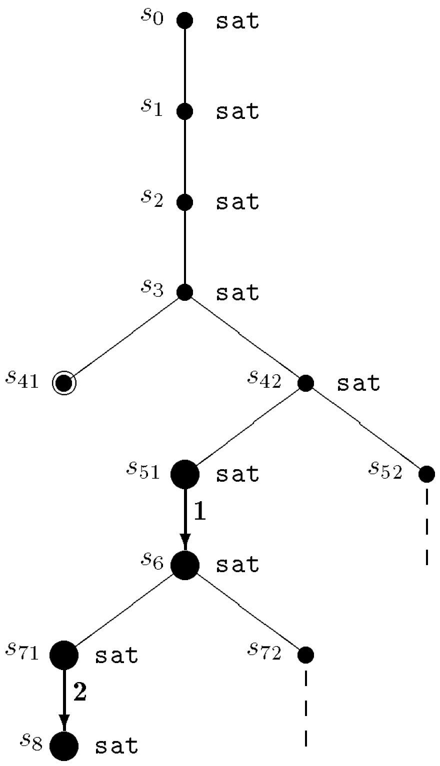
This completes the description of the K_n tableau construction. We now give an example of this construction in operation. Figure 3 contains a pictorial description of the construction, and should be consulted while reading the textual description below. Let

$$\varphi_0 \;=\; (p \wedge \neg(p \wedge q)) \wedge (K_1(\neg p) \wedge \neg K_1 K_2 q).$$

The construction begins by creating a state s_0, with $L(s_0) = \{\varphi_0\}$. It then applies step (a)(ii) three times, adding all of the conjuncts in φ_0 to the labeling set. The resulting labeling set is $L(s_3)$ in Figure 3 The only witness to the fact that $L(s_3)$ is not a propositional tableau is the formula $\neg(p \wedge q)$. This formula is satisfied if one of $\neg p$ or $\neg q$ holds. As a result, by step (a)(iii), two successors s_{41} and s_{42} are created, with $\neg p$ added to the labeling set in forming $L(s_{41})$, and $\neg q$ added to form $L(s_{42})$. Notice at this point that $L(s_{41})$ is blatantly inconsistent, since it contains both p and $\neg p$. As a result, steps (a)-(c) will never create any successors of this node, and step (d) will never mark it "satisfiable". The construction continues with s_{42}. $L(s_{42})$ is a propositional tableau but not a fully expanded one. Moreover, the only witness to the fact that $L(s_{42})$ is not fully expanded is K_2q. By step (b) we thus create two successors of s_{42}, denoted s_{51} and s_{52}, with $L(s_{51}) = L(s_{42}) \cup \{K_2q\}$ and $L(s_{52}) = L(s_{42}) \cup \{\neg K_2 q\}$. At this point s_{51} is a leaf of the tree, and $L(s_{51})$ is a fully expanded propositional tableau that is not blatantly inconsistent. The only formula of the form $\neg K_i\psi$ in $L(s_{51})$ is $\neg K_1K_2q$. By step (c), a 1-successor s_6 of s_{51} is created, with $L(s_6) = \{\neg p\} \cup \{\neg K_2 q\}$. Notice that $\{\neg p\} = L(s_{51})/K_1$, which is why it is included in $L(s_6)$. Now q is a witness to the fact that $L(s_6)$ is not fully expanded, so two successors s_{71} and s_{72} are created by adding q and $\neg q$ two their respective labeling sets. Finally, $L(s_{71}) = \{\neg p, \neg K_2 q, q\}$ is fully expanded, and by step (c) a 2-successor node s_8 with $L(s_8) = \{\neg q\}$ is created. Steps (a)-(c) do not apply to s_8 since $L(s_8)$ is fully expanded and contains no formulas of the form $K_i\psi$. Moreover, since it is not blatantly inconsistent, step (d) marks s_8 "satisfiable" (denoted by `sat` in Figure 3). As a result, one by one step (d) marks the nodes $s_{71}, s_6, s_{51}, s_{42}, s_3, s_2, s_1$ and s_0 "satisfiable".

Formally, the K_n construction will continue and create successors of s_{52} and of s_{72}, and will eventually also mark them "satisfiable". The interested reader is invited to complete this part of the construction. However, as far as finding out that φ_0 is K_n-satisfiable, the part of the construction we have described suffices.

Notice that the path from s_0 to s_{51} has the role of extending the labeling function from $L(s_0) = \{\varphi_0\}$ to a fully expanded propositional tableau ($L(s_{51})$). The same goes for the path from s_6 to s_{71}. In contrast, s_6 is a 1-successor of s_{51}, and s_8 is a 2-successor of s_{71}. The K_n tableau for φ_0 that can be extracted from the pre-tableau we have constructed would thus be $(S', L', \mathcal{K}_1', \mathcal{K}_2')$, where $S' = \{s_{51}, s_{71}, s_8\}$, while the labeling function L' is the restriction of L constructed above to the nodes of S'. Finally, $\mathcal{K}_1' = \{(s_{51}, s_{71})\}$ and $\mathcal{K}_2' = \{(s_{71}, s_8)\}$.



$$L(s_0) = \{\varphi_0\}$$

$$L(s_1) = L(s_0) \cup \{p \wedge \neg(p \wedge q), K_1\neg p \wedge \neg K_1 K_2 q\}$$

$$L(s_2) = L(s_1) \cup \{K_1\neg p, \neg K_1 K_2 q\}$$

$$L(s_3) = L(s_2) \cup \{p, \neg(p \wedge q)\}$$

$$L(s_{41}) = L(s_3) \cup \{\neg p\}; \qquad L(s_{42}) = L(s_3) \cup \{\neg q\}$$

$$L(s_{51}) = L(s_{42}) \cup \{K_2 q\}; \quad L(s_{52}) = L(s_{42}) \cup \{\neg K_2 q\}$$

$$L(s_6) = \{\neg p\} \cup \{\neg K_2 q\}$$

$$L(s_{71}) = L(s_6) \cup \{q\}; \qquad L(s_{72}) = L(s_6) \cup \{\neg q\}$$

$$L(s_8) = \{\neg q\}$$

Figure 3: K_n tableau construction for $\varphi_0 = (p \wedge \neg(p \wedge q)) \wedge (K_1\neg p \wedge \neg K_1 K_2 q)$.

We next prove some important properties of this construction. Define a node s to be an *internal node* if $L(s)$ is not a fully expanded propositional tableau; otherwise we call s a *state*. As we remarked above, the object we construct is not a K_n tableau. However, as we shall show, if the algorithm returns "satisfiable", then the states that are marked satisfiable do form a K_n tableau. A state s' is an $\mathcal{K}_i$-*successor* of a state s if s and s' are consecutive states along a branch in the tree and the first edge on the path between them is labeled with an i. (Note that we distinguish the notion of i-successor from that of $\mathcal{K}_i$-successor. We say s' is an i-successor of s if there is an edge labeled i along the path from s to s' in the pre-tableau.

For s' to be a $\mathcal{K}_i$ successor of s, both s' and s have to be states, and there must be a path $t_0, \ldots, t_k$ in the tree such that $t_0 = s$, $t_k = s'$, t_1 is an i-successor of t_0, and for all j with $0 < j < k$, t_j is an internal node and t_{j+1} is a successor of t_j in the pre-tableau.) Finally, define $dep(L(s)) = \max\{dep(\psi) : \psi \in L(s)\}$.

Lemma 6.9: *For all formulas φ, the K_n tableau construction terminates.*

Proof: Suppose that $|\varphi| = m$. Note that our construction guarantees that for any node s in the tree, $L(s)$ consists only of formulas in $Sub^+(\varphi)$ (i.e., subformulas of φ or their negations). Thus, $|L(s)| \leq 2m$. It follows that we can apply steps 2(a) and 2(b) at most m times before we reach a node s' such that either s' is a state or $L(s')$ is blatantly inconsistent. Moreover, it is immediate from step 2(c) that if $L(s)$ is a state, then $dep(L(s')) < dep(L(s))$ for any successor s' of s. Since $dep(\varphi) \leq m$, it immediately follows that the final tree constructed in the algorithm above has height at most m^2; in particular, it is finite. It immediately follows that the construction terminates. ∎

Theorem 6.10: *A formula φ is K_n satisfiable iff the K_n tableau construction for φ returns "φ is satisfiable".*

Proof: First suppose that the K_n tableau construction for φ returns "φ is satisfiable". We construct a K_n tableau for φ as follows. The nodes in the tableau consist of the states s in the pre-tableau constructed above that are marked "satisfiable". We take $(s, s') \in \mathcal{K}_i$ in the tableau if s' is a $\mathcal{K}_i$-successor of s. It is easy to check that the construction guarantees that this is a tableau for φ (note that in particular we must have a node s such that $L(s)$ is a propositional tableau and $\varphi \in L(s)$, otherwise the root of the pre-tableau would not be labeled "satisfiable"). By Proposition 6.8, it follows that φ is K_n satisfiable.

For the converse, given a node s, let ψ_s be the conjunction of all the formulas in $L(s)$. We show that if a node s in the pre-tableau is not marked "satisfiable" then $\neg\psi_s$ is provable, i.e., ψ_s is inconsistent. Since the axiom system is sound, it follows that ψ_s is indeed unsatisfiable in this case. In particular, if the root is not marked "satisfiable", then $\neg\varphi$ is provable (and, by Proposition 6.8, there can be no tableau for φ).

We proceed by induction on the height of s (i.e., the length of the longest path from s to a leaf of the pre-tableau). From step 2(d), it follows that if the height of s is 0 (i.e., s is a leaf of the tree), then s is not marked "satisfiable" if and only if $L(s)$ is blatantly inconsistent. In this case it is trivial to see that ψ_s is inconsistent. For the general case, first suppose that s an internal node. Then from step 2(d), it follows that s is not marked "satisfiable" if and only if none of s's successors is marked "satisfiable". By the induction hypothesis, it follows that $\psi_{s'}$ is inconsistent for every successor s' of s in the pre-tableau. It is easy to see that ψ_s is inconsistent (all we need here is propositional reasoning). For example, suppose successors s_1 and s_2 of s are created due to the presence of a witness $\neg(\psi_1 \wedge \psi_2)$ to the fact that $L(s)$ is not a propositional tableau. Then, by construction, we have $L(s_i) = L(s) \cup \{\neg\psi_i\}$, $i = 1, 2$. Using only propositional reasoning we can show that $\vdash (\neg\psi_{s_1} \wedge \neg\psi_{s_2}) \Rightarrow \neg\psi_s$. The induction hypothesis tells us that $\mathrm{K}_n \vdash \neg\psi_{s_i}$, $i = 1, 2$. Thus, it follows that $\mathrm{K}_n \vdash \neg\psi_s$; i.e., ψ_s is inconsistent.

If s is not marked "satisfiable" and $L(s)$ is a state, then from step 2(d) it follows that some successor s' of s in the pre-tableau is not marked "satisfiable". By construction, there must be some formula $\neg K_i\psi \in L(s)$ such that $L(s') = L(s)/K_i \cup \{\neg\psi\}$. By the induction hypothesis, we must have that $\psi_{s'}$ is inconsistent. Suppose $L(s)/K_i = \{\varphi_1, \ldots, \varphi_k\}$, so that $L(s') = \{\varphi_1, \ldots, \varphi_k, \neg\psi\}$. Since $\psi_{s'}$ is inconsistent, arguments identical to those used in Theorem 2.3 can be used to show that

$$\vdash K_i\varphi_1 \Rightarrow (K_i\varphi_2 \Rightarrow (\ldots(K_i\varphi_k \Rightarrow K_i\psi)\ldots)).$$

Since $\{K_i\varphi_1, \ldots, K_i\varphi_k, \neg K_i\psi\} \subseteq L(s)$, it immediately follows that ψ_s is inconsistent. This completes the induction step of the proof. ∎

Note that the proof of Theorem 6.10 gives us another proof of the completeness of the K_n axioms. For suppose that φ is valid. In order to show that φ is provable, we apply the tableau construction above to $\neg\varphi$. It must be the case that the root of the pre-tableau will not be marked "satisfiable" (otherwise, by Theorem 6.10, $\neg\varphi$ would be satisfiable, contradicting the validity of φ). It now follows from the proof of Theorem 6.10 that $\neg\neg\varphi$ is provable, and hence so is φ.

Theorem 6.11: *There is an algorithm for deciding satisfiability of K_n formulas that runs in polynomial space.*

Proof: We give an efficient (polynomial space) way of checking whether the root of the tree in the tableau construction for φ will be marked "satisfiable". The intuitive idea is to do a depth-first search of the pre-tableau, using the observation that how a node is marked can be completely determined by its label and how its successors are marked. Thus, once we have determined how a node is marked, we never have to consider the subtree below that node again.

More formally, given a node s with label $L(s)$, we show by induction on h that if we start the tableau construction with a node labeled by $L(s)$ and end with a tree of height h, then we can determine how s will be marked using at most $(3h+1)m$ bits of storage (where $|\varphi| = m$). Roughly speaking, we use m bits to store φ, and $3hm$ bits to explore the tree below s. We can represent the label of any node s' by a bit string of length $2m$. We simply enumerate the $2m$ formulas in $Sub^+(\varphi)$ in some order; the i^{th} formula in the enumeration is in $L(s)$ iff the i^{th} bit in the bit string is 1. Thus, we use $2m$ bits to encode the bit string at each node s', and a further (at most) m bits to keep track of which part of the tree below s' we still need to explore. If $h = 0$, then $L(s)$ is either blatantly inconsistent, in which case s is not marked "satisfiable", or s is a state and $L(s)$ has no formulas of the form $\neg K_i\psi$, in which case s is marked "satisfiable". This completes the base case. If $h > 0$ and s is not an internal node, then the tableau construction creates one or two successors of s; moreover, s is marked "satisfiable" iff one of its successors is marked "satisfiable". Thus, we can easily use the inductive assumption to compute how each of the successors is marked, reusing the space after each computation. (This reuse of space corresponds to deleting all the information about the subtree rooted at the successor, since we no longer need it.) Similar arguments work if s is a state.

Since, as observed in the proof of Theorem 6.10, the tree has depth at most m^2, it follows that we can compute if φ is satisfiable using space $O(m^3)$. ∎

We can easily modify this procedure to deal with T_n formulas. All we need do is to modify step 2(d) so that a node s is not marked satisfiable if both $K_i\psi$ and $\neg\psi$ are in $L(s)$ for some formula ψ and agent i. Note that any node not satisfying this condition is T_n inconsistent by axiom A3. All the results we proved in the case of K_n can now be reproved for T_n in an analogous manner. Thus, we get:

Theorem 6.12: *There is an algorithm for deciding satisfiability of T_n formula that runs in polynomial space.*

Dealing with $\mathrm{S4}_n$ is a bit more complicated. The idea now is to construct an $\mathrm{S4}_n$ tableau. We first must make one obvious modification to the T_n construction; namely, in substep 2(c), we still create an i-successor s' for each formula of the form $\neg K_i\psi \in L(s)$ if $L(s)$ is a state, but now we set $L(s') = \{K_i\psi' : K_i\psi' \in L(s)\} \cup \{\neg\psi\}$ in order to ensure that we get an $\mathrm{S4}_n$ tableau. The only problem with this modification is that now we can no longer prove that the construction terminates. In particular, it is not necessarily the case that $dep(L(s')) < dep(L(s))$ for every successor s' of a state s (as was the case in the proof of Lemma 6.9). We deal with the problem by modifying step 2(c) as follows:

2(c$'$) If s is a leaf of the tree and $L(s)$ is a fully expanded propositional tableau, then for each formula of the form $\neg K_i\psi \in L(s)$, let $L'(s,\psi) = \{K_i\psi' : K_i\psi' \in L(s)\} \cup \{\neg\psi\}$. If there is no ancestor s'' of s in the tree such that $L(s'') = L'(s,\psi)$, then create an i-successor s' of s with $L(s') = L'(s,\psi)$.

We can now show that this construction terminates. We get the following analogue of Lemma 6.9

Lemma 6.13: *For all formulas φ, the $S4_n$ tableau construction for φ terminates.*

Proof: The proof is quite similar to that of Lemma 6.9, so we just briefly outline the differences here. If $|\varphi| = m$, we can again show that there will be at most $m-1$ internal nodes between consecutive states on any given branch of the tree. Next note that if $L(s)$ is a fully expanded propositional tableau, s' is an i-successor of s, and s'' is any descendant of s', then it is easy to see that the depth of any formula in $L(s'')$ not of the form $K_i\psi$ must be strictly less than $dep(L(s))$. That is, the only formulas whose depth might not go down are those of the form $K_i\psi$. It immediately follows that if s, s', and s'' are states, s' is a $\mathcal{K}_i$-successor of s, and s'' is a $\mathcal{K}_j$-successor of s' with $i \neq j$, then $dep(L(s'')) < dep(L(s))$. Finally, note that for all i, a branch can have at most m^2 consecutive states each of which is a $\mathcal{K}_i$-successor of its predecessor. For suppose we have a path (portion of a branch) where all edges coming out of states are labeled i. It is easy to see that if s' is a descendant of s on this path, then $L(s)/K_i \subseteq L(s')/K_i$. Thus, there can be at most m distinct sets of the form $L(s)/K_i$ for s on the path. It then follows that there can be at most m^2 distinct sets of the form $L'(s,\psi)$ for a node s on this path (since there are at most m choices for ψ). Putting all these observations together, it follows from the tableau construction (in particular step 2(c$'$)) that the tree can have depth at most m^4. The remainder of the proof proceeds along the same lines as that of Lemma 6.9. ∎

We can now prove the following analogue of Theorem 6.10:

Theorem 6.14: *A formula φ is $S4_n$ satisfiable iff the $S4_n$ tableau construction for φ returns "φ is satisfiable".*

Proof: Again we only sketch the differences between this proof and that of Theorem 6.10. As before, we can prove that if a node s is not marked "satisfiable", then ψ_s is $\mathrm{S4}_n$ inconsistent. If the root is marked "satisfiable", we construct an $\mathrm{S4}_n$ tableau for φ along the same lines as before. Again, the nodes in the tableau are the states in the construction that are marked "satisfiable". The only difference is that we now take $(s, s') \in \mathcal{K}_i$ in the tableau either if s' is a $\mathcal{K}_i$-successor of s or if s' is the first state on a path starting with an ancestor s'' of s in the pre-tableau such that $L(s'') = L'(s, \psi)$ for some formula $\neg K_i\psi$ in $L(s)$. We leave it to the reader to check that this gives us an $\mathrm{S4}_n$ tableau for φ, proving that φ is indeed $\mathrm{S4}_n$ satisfiable. ∎

Finally, we get the following analogue to Theorem 6.11:

Theorem 6.15: *There is an algorithm for deciding satisfiability of $S4_n$ formulas that runs in polynomial space.*

Proof: We proceed as in Theorem 6.11, except that now we show by induction on the height h of a node that if s is a node of height h and X is a list of labels that have appeared in ancestors of s, then if we start the tableau construction with a node labeled by $L(s)$ and ancestor labelings given by X, we can determine how the node will be marked using at most $(2h+3)m+O(1)+|X|$ bits of storage (where $|\varphi| = m$). Since a node has at most m^4 ancestors, and each labeling requires space $2m$ to store, it follows that $|X| \leq 2m^5$. Since h is also at most m^4, we can compute the labeling using at most $O(m^5)$ bits. ∎

Next we turn our attention to $\mathrm{S5}_n$. We can deal with $\mathrm{S5}_n$ by making only a small modification to the construction for $\mathrm{S4}_n$, which should be quite obvious to the reader who has come this far. We modify step 2(c) so that now an i-successor of a state is labeled with all the formulas of the form $K_i\psi$ *and* the formulas of the form $\neg K_i\psi$ that were in the label of its predecessor, so as to ensure that the extra condition required for an $\mathrm{S5}_n$ tableau will hold. The new step is:

2(c$''$) If s is a leaf of the tree and $L(s)$ is a fully expanded propositional tableau, then for each formula of the form $\neg K_i\psi \in L(s)$, let $L''(s, \psi) = \{K_i\psi' : K_i\psi' \in L(s)\} \cup \{\neg K_i\psi' : \neg K_i\psi' \in L(s)\} \cup \{\neg\psi\}$. If there is no ancestor s'' of s in the tree such that $L(s'') = L''(s, \psi)$, then create an i-successor s' with $L(s') = L''(s, \psi)$.

The same techniques as used in the previous proofs can now be used to show:

Theorem 6.16: *There is an algorithm for deciding satisfiability of $S5_n$ formulas that runs in polynomial space.*

There is a subtlety in the proof of correctness for the $\mathrm{S5}_n$ case that explains our need to use *fully expanded* propositional tableaus. In proving the analogue of Theorem 6.10 for $\mathrm{S5}_n$, we need to show that if the construction returns "φ is satisfiable", then we can construct an $\mathrm{S5}_n$ tableau for φ. We use the same construction as in the proof of Theorem 6.10. The difficulty comes in showing that the additional condition for $\mathrm{S5}_n$ tableaus, namely that if $(s, t) \in \mathcal{K}_i$ then $K_i\psi \in L(s)$ iff $K_i\psi \in L(t)$, is met. Since the nodes in the tableau are the states in our construction, it suffices to show that if t is a $\mathcal{K}_i$-successor of s, then s and t agree on all subformulas of the form $K_i\psi$ and $\neg K_i\psi$. Suppose s' is the i-successor of s on the path from s to t. The new step 2(c$''$) of our construction guarantees that $L(s)$ and $L(s')$ agree on all subformulas of the form $K_i\psi$ or $\neg K_i\psi$. To see that $L(s)$ and $L(t)$ also agree on such formulas, suppose that $K_i\psi \in L(s)$. By construction $K_i\psi \in L(s')$. Since $L(s') \subseteq L(t)$, we must have $K_i\psi \in L(t)$. A similar argument shows that if $\neg K_i\psi \in L(s)$, then $\neg K_i\psi \in L(t)$. Conversely, suppose that $K_i\psi \in L(t)$. From our construction, it follows that $K_i\psi$ must be a subformula of some formula ψ' in $L(s)$. Since $L(s)$ is fully expanded, it must be the case that either $K_i\psi \in L(s)$ or $\neg K_i\psi \in L(s)$. Our earlier arguments showed that if $\neg K_i\psi \in L(s)$, then $\neg K_i\psi \in L(t)$, making $L(t)$ blatantly inconsistent (and thus not a state). It follows that $K_i\psi \in L(s)$, as desired. A similar argument shows that if $\neg K_i\psi \in L(t)$, then $\neg K_i\psi \in L(s)$.

We also observe that a variant of the argument of Lemma 6.13 can be used to show that the depth of the tree we construct in the case of $\mathrm{S5}_n$ is at most m^3, rather than m^4. The reason is that now if we have a path on the tree where all edges coming out of states are labeled with an i, then if s' is a descendant of s on the path, we must have $L(s)/K_i = L(s')/K_i$ (rather than just $L(s)/K_i \subseteq L(s')/K_i$). Thus, such a branch can have at most m consecutive states each of which is a $\mathcal{K}_i$-successor of its predecessor, rather than m^2. We remark that we can slightly vary the construction so that we never have 3 consecutive states s, t, u on a branch such that t is a $\mathcal{K}_i$-successor of s and u is a $\mathcal{K}_i$-successor of t. If we do this, the resulting tree has depth $\leq 2m^2$; we omit details here.

We can modify the $\mathrm{S5}_n$ case to deal with $\mathrm{KD45}_n$ in a straightforward way. Details are left to the reader. We summarize the results of the last two sections in the following theorem:

Theorem 6.17: *The satisfiability problem for K_n, T_n, $S4_n$, $n \geq 1$, $S5_n$, $KD45_n$, and $n \geq 2$, is PSPACE-complete.*

Since the class *PSPACE* consists of deterministic algorithms, it is closed under complementation. It thus follows that the validity problem for all these logics is also *PSPACE*-complete.

We remark that further modifications of these proofs allow us to deal with the distributed knowledge operator. Basically, we treat the distributed knowledge operator D as if it were another K_j operator during the construction, and interpret the edges corresponding to D's possibility relation as if they were edges of all of the $\mathcal{K}_i$ relations. We also need to ensure that if $K_i\psi \in L(s)$ and $D\psi \in Sub(\varphi)$, then $D\psi \in L(s)$ (so that any node whose label contains both $K_i\psi$ and $\neg D\psi$ will be blatantly inconsistent). We again get *PSPACE* completeness results. We leave details to the enthusiastic reader.

It is worth noting that for an important special case, the satisfiability and validity problems simplify. If we restrict attention to formulas of a fixed bounded depth (that is, if we restrict attention to formulas φ such that $dep(\varphi) \leq k$, for some fixed k), then the satisfiability problem for this subclass of formulas is *NP*-complete, for all the logics we have been considering. The lower bound is immediate from propositional logic. For the upper bound, observe that our construction guarantees that a formula φ of depth at most k is satisfied in a structure that looks like a tree, has height at most k, and outdegree at most $|\varphi|$. This structure has at most $|\varphi|^{k+1}$ states, a polynomial number. Thus, we can guess a structure satisfying φ in polynomial time, and verify that it does indeed satisfy φ, giving us the desired *NP* upper bound.

6.4 Decision procedures for common knowledge

The common knowledge operator C adds a great deal of expressive power to the language. It provides means to make universal statements about what is true at all reachable states in the structure. As a result, we shall show that the validity problem for languages with common knowledge is *EXPTIME*-complete.

Recall that given a formula φ in the language without common knowledge, the key step in our *PSPACE* decision procedure for φ comes in constructing a pre-tableau T of polynomial depth and whose root is labeled φ. We then consider the tableau for φ embedded in T, and construct from it a structure M satisfying φ. As a consequence of our construction, it follows that if φ is a satisfiable formula in $\mathcal{L}_n(\Phi)$, there is a structure M satisfying φ with paths of length at most polynomial in $|\varphi|$. As we now show, this is no longer the case when we add common knowledge to the language. It follows that the tableau construction we used to obtain polynomial space bounds in the previous section will not work once we add common knowledge to the language.

Proposition 6.18: *For all m, there is a formula φ_m^K (resp., φ_m^T, φ_m^{S4}, φ_m^{S5} φ_m^{KD45}) of size $O(m^2)$ that is K^C (resp., T^C, $S4_2^C$, $S5_2^C$, $KD45_2^C$) satisfiable, but every structure in $\mathcal{M}_1$ (resp., $\mathcal{M}_1^r$, $\mathcal{M}_2^{rt}$, $\mathcal{M}_2^{rst}$, $\mathcal{M}_2^{elt}$) that satisfies it has a path of length $2^m - 1$.*

Proof: The basic idea is that once we have common knowledge we can write a formula of size in $O(m^2)$ that forces any satisfying model to have a path of length $2^m - 1$.

The primitive propositions are $p_0, \ldots, p_{m-1}$. We use these propositions to encode the bits of an m-bit binary counter, with p_0 encoding the low order bit and p_{m-1} encoding the high-order bit. If p_i is true at a given state, this encodes the fact that the i^{th} bit of the counter is 1. We want to write a formula that forces the counter to take on all the values from 0 to $2^m - 1$ consecutively in a sequence of states (cf. [HV89, Lemma 4.1], where a similar technique is used). Note that if $\mathbf{c} = c_{m-1} \ldots c_0$ and $\mathbf{d} = d_{m-1} \ldots d_0$ are two m-bit counters, then $\mathbf{d} = \mathbf{c}+1$ precisely when the following holds: for some $k \leq m-1$, we have $c_i = 1$ for all $i < k$, $c_k = 0$, $d_i = 0$ for all $i < k$, $d_k = 1$, and $c_j = d_j$ for $k+1 \leq j \leq m-1$.

We first consider K^C. The formula σ_m^K is the conjunction of four formulas, σ_{m1}^K, σ_{m2}^K, σ_{m3}^K, and σ_{m4}^K. These formulas are described below, followed by the intuition behind them:

$$
\begin{aligned}
\sigma_{m1}^K :&\ C(\neg K \neg \mathit{true}),\\
\sigma_{m2}^K :&\ (\neg p_0 \wedge \ldots \wedge \neg p_{m-1}),\\
\sigma_{m3}^K :&\ \textstyle\bigwedge_{i=0}^{m-1} C((\bigwedge_{j=0}^{i-1} p_j) \Rightarrow ((p_i \Rightarrow K\neg p_i) \wedge (\neg p_i \Rightarrow K p_i))),\\
\sigma_{m4}^K :&\ \textstyle\bigwedge_{i=0}^{m-1} C((\bigvee_{j=0}^{i-1} \neg p_j) \Rightarrow ((p_i \Rightarrow K p_i) \wedge (\neg p_i \Rightarrow K\neg p_i))).
\end{aligned}
$$

For the case $i = 0$, we take the conjunction $\bigwedge_{j=0}^{i-1} p_j$ in σ_{m3}^K to be equivalent to *true*, and take the disjunction $\bigvee_{j=0}^{i-1} \neg p_j$ in σ_{m4}^K to be equivalent to *false*.

We give the intuition behind these formulas in the course of showing how they are used. Suppose $(M, s_0) \models \sigma_m^K$. The formula σ_{m1}^K guarantees that there is a sequence $s_0, s_1, \ldots, s_{2^m-1}$ of 2^m (not necessarily distinct) states such that $(s_i, s_{i+1}) \in \mathcal{K}$. We now show that these states are in fact all distinct, by showing that the truth values of $p_0, \ldots, p_{m-1}$ encodes the number i at state s_i. The formula σ_{m2}^K guarantees that s_0 encodes the value 0. The formulas σ_{m3}^K

and σ^K_{m4} guarantee that if s is reachable from s_0 and $(s,t) \in \mathcal{K}$, then p_i has the same truth value in s and t iff some p_j is false for $j < i$. By our earlier comments, this means that if s is reachable from s_0 and $(s,t) \in \mathcal{K}$, then s and t encode consecutive values of the counter. It now follows by an easy induction that s_i encodes the value i. Thus, $s_0, \ldots, s_{2^m-1}$ must all be distinct states. Note that σ^K_m is satisfied in a structure $M = (\{s_0, \ldots, s_{2^m-1}\}, \pi, \mathcal{K})$, where $\mathcal{K} = \{(s_i, s_{i+1}) : i < 2^m - 1\} \cup \{(s_{2^m-1}, s_0)\}$ and π is defined so that s_i encodes the value i.

We now modify this argument to deal with T^C. The formula σ^K_m as it stands is unsatisfiable in reflexive structures. As we showed above, if σ^K_m is true at a state s_0, then in any sequence of states $s_0, s_1, \ldots, s_{2^m-1}$ with $(s_i, s_{i+1}) \in \mathcal{K}$, the state s_i encodes the integer i; in particular, all the states in such a sequence must be distinct. However, in a reflexive model, one such sequence is $s_0, \ldots, s_0$, where the states are clearly not distinct. We deal with this problem by introducing a new primitive proposition p_Δ to mark the fact that a change has taken place. We take σ^T_m to be a formula which is true at a state s_0 if in any sequence $s_0, s_1, \ldots, s_{2^m-1}$ such that $(s_i, s_{i+1}) \in \mathcal{K}$ and p_Δ alternates truth values between consecutive states in the sequence (so that, for example, p_Δ is true at s_i if i is even and false at s_i if i is odd), the state s_i encodes the value of i. Formally, we take σ^T_m to be the conjunction of the four formulas, σ^T_{m1}, σ^T_{m2}, σ^T_{m3}, and σ^T_{m4} described below:

$$
\begin{array}{ll}
\sigma^T_{m1}: & C((p_\Delta \Rightarrow \neg K p_\Delta) \wedge (\neg p_\Delta \Rightarrow \neg K \neg p_\Delta)),\\
\sigma^T_{m2}: & (\neg p_0 \wedge \ldots \wedge \neg p_{m-1}),\\
\sigma^T_{m3}: & \bigwedge_{i=0}^{m-1} C[(\bigwedge_{j=0}^{i-1} p_j) \Rightarrow (((p_\Delta \wedge p_i) \Rightarrow K(\neg p_\Delta \Rightarrow \neg p_i)) \wedge ((p_\Delta \wedge \neg p_i) \Rightarrow K(\neg p_\Delta \Rightarrow p_i)) \wedge \\
& \qquad ((\neg p_\Delta \wedge p_i) \Rightarrow K(p_\Delta \Rightarrow \neg p_i)) \wedge ((\neg p_\Delta \wedge \neg p_i) \Rightarrow K(p_\Delta \Rightarrow p_i)))],\\
\sigma^T_{m4}: & \bigwedge_{i=0}^{m-1} C[(\bigvee_{j=0}^{i-1} \neg p_j) \Rightarrow (((p_\Delta \wedge p_i) \Rightarrow K(\neg p_\Delta \Rightarrow p_i)) \wedge ((p_\Delta \wedge \neg p_i) \Rightarrow K(\neg p_\Delta \Rightarrow \neg p_i)) \wedge \\
& \qquad ((\neg p_\Delta \wedge p_i) \Rightarrow K(p_\Delta \Rightarrow p_i)) \wedge ((\neg p_\Delta \wedge \neg p_i) \Rightarrow K(p_\Delta \Rightarrow \neg p_i)))].
\end{array}
$$

Suppose $(M, s_0) \models \sigma^T_m$. The formula σ^T_{m1} guarantees that there is a sequence $s_0, s_1, \ldots, s_{2^m-1}$ such that $(s_i, s_{i+1}) \in \mathcal{K}$ and p_Δ alternates truth values between consecutive states in the sequence. The formulas σ^T_{m3} and σ^T_{m4} again guarantee that if s is reachable from s_0, $(s,t) \in \mathcal{K}$, and p_Δ has different truth values at s and t, then s and t encode consecutive values of the counter. Again we can prove by a straightforward induction that in any sequence $s_0, s_1, \ldots, s_{2^m-1}$ as above, the state s_i encodes the integer i. The rest of the proof proceeds as in the case of the logic K.

The formula σ^T_m is not satisfiable in transitive structures. To see why, suppose we have a sequence $s_0, s_1, s_2, \ldots$ as above. Then s_0 encodes the integer 0 while s_2 encodes the integer 2. But transitivity implies that $(s_0, s_2) \in \mathcal{K}$. Now σ^T_{m3} and σ^T_{m4} imply that s_2 encodes the integer 1, which contradicts the assumption that s_2 encodes 2. That is why we need to allow two agents when dealing with the logics S4, S5, and KD45. We simply replace all occurrences of K in σ^T_m by $K_1 K_2$. This gives us σ^{S4}_m, σ^{S5}_m, and σ^{KD45}_m. We leave it to the reader to check that these formulas do the job. ∎

The previous result shows that the proof technique we used in the previous section for obtaining *PSPACE* upper bounds will not extend to logics involving common knowledge. This is not an artifact of our particular proof technique. We now prove an exponential time lower bound. Our proof is a minor modification of the proof of the exponential time lower bound for the satisfiability problem for PDL given by Fischer and Ladner [FL79]. Rather than going through the details of the proof here, we review PDL to show the similarity between it and the logics of knowledge we have been considering, and refer the reader to [FL79] for further details of the proof.

PDL is a modal logic for reasoning about programs. We start with primitive programs $a, b, c \ldots$, and form more complicated programs using regular constructors such as ; and *. For example, if α and β are programs, then so are $\alpha;\beta$ and α^*. Intuitively, $\alpha;\beta$ corresponds to running α and then running β, while α^* corresponds to running α some finite (but arbitrary) number of times. Associated with each program is a modal operator $[\alpha]$; if φ is a formula and α is a program, then $[\alpha]\varphi$ is a formula. The formula $\langle\alpha\rangle\varphi$ is just an abbreviation for $\neg[\alpha]\neg\varphi$. We give semantics to PDL using Kripke structures. Corresponding to α is a binary relation $\rho(\alpha)$ on states. Intuitively, $(s,t) \in \rho(\alpha)$ if running program α starting in state s, it is possible to end up in state t. The formula $[\alpha]\varphi$ is true at a state s if φ is true at all states t such that $(s,t) \in \rho(\alpha)$. The analogy to the formula $K\varphi$ should be clear. We define $\rho(\alpha^*)$ to be the transitive closure of $\rho(\alpha)$. Thus, α^* bears the same relationship to α as C bears to E.

Fischer and Ladner prove the lower bound for PDL by showing that for each exponential time Turing machine $\mathbf{A}$ and input x, there is a PDL formula $\varphi_{\mathbf{A},x}$ of size $O(|x|)$ such that $\varphi_{\mathbf{A},x}$ is satisfiable iff $\mathbf{A}$ accepts on input x.[13] The only modal operators in the formula $\varphi_{\mathbf{A},x}$ are $[\vdash]$ and $[\vdash^*]$, where $\vdash$ is taken to be a primitive program. By replacing all occurrences of $[\vdash]$ in $\varphi_{\mathbf{A},x}$ by K and all occurrences of $[\vdash^*]$ by C, we get a formula $\varphi^K_{\mathbf{A},x}$ which is K^C satisfiable iff $\mathbf{A}$ accepts input x. This proves the exponential time lower bound for K^C. In order to get the exponential time lower bound for all the other logics, we modify $\varphi^K_{\mathbf{A},x}$ just as we modified the formula σ^K_m in the previous result. That is, to deal with the logic T^C, we use a new primitive proposition p_Δ to mark that a change has taken place; we then get a formula $\varphi^T_{\mathbf{A},x}$ that is T^C satisfiable iff $\mathbf{A}$ accepts input x. For S4, S5, and KD45, we replace all occurrences of K in $\varphi^T_{\mathbf{A},x}$ by K_1K_2. We omit further details here. As a result, we get

Theorem 6.19: *The satisfiability problem for K^C_n, T^C_n, $n \geq 1$, and $S4^C_n$, $S5^C_n$, $KD45^C_n$, $n \geq 2$, is exponential time hard.*

Finally, we want to prove an exponential time upper bound to match the lower bound. The proof of Theorem 4.3 shows that if a formula φ in $\mathcal{L}^C_n$ is satisfiable, it is satisfiable in a structure of size $\leq 2^{(3+n)|\varphi|}$.[14] Since n is a constant in this context, this immediately gives us a nondeterministic exponential time upper bound. To see if a formula is satisfiable, we simply guess the structure that satisfies it, and verify that it is indeed satisfied in that structure (which, by Proposition 3.1 and the remarks following it, can be done efficiently). We can get a deterministic exponential time algorithm by actually constructing the structure, rather than guessing it. We do so by modifying techniques due to Pratt [Pra79]. (The result in the case of $\mathrm{S5}^C_n$ was also proved in [FI87].)

Theorem 6.20: *The satisfiability problem for K^C_n, T^C_n, $n \geq 1$, and $S4^C_n$, $S5^C_n$, $KD45^C_n$, $n \geq 2$, is complete for exponential time.*

[13] Actually, Fischer and Ladner consider *alternating polynomial space* Turing machines rather than exponential time Turing machines. However, it is well known [CKS81] that alternating polynomial space Turing machines accept precisely the same languages as exponential time Turing machines.

[14] We remark that with a little more effort, we could have shown that if a formula is satisfiable, then it is actually satisfiable in a structure of size at most $2^{|\varphi|}$.

Proof: The lower bound is just Theorem 6.19. For the upper bound, suppose we are given a formula φ. We consider the case of K_n^C here; the modifications for the other logics are straightforward. We want to either construct a structure satisfying φ or show that none exists. The construction has much of the flavor of the proof of Theorem 4.3. Recall that the set $Sub_C(\varphi)$ consists of all the subformulas of φ, together with the formulas $E(\psi \wedge C\psi)$ and $\psi \wedge C\psi$ for each subformula $C\psi$ of φ, while $Sub_C^+(\varphi)$ consists of all formulas in $Sub_C(\varphi)$ and their negations. Let $S^1(\varphi)$ consist of all subsets A of $Sub_C^+(\varphi)$ that are propositional tableaus and are maximal, in that for each formula $\psi \in Sub_C^+(\varphi)$, either $\psi \in A$ or $\neg\psi \in A$. Note that there are at most $2^{3|\varphi|}$ sets in $S^1(\varphi)$. (The set $S^1(\varphi)$ should be considered an approximation to the set $Con_C(\varphi)$ from the proof of Theorem 4.3. We do not know how to compute the set $Con_C(\varphi)$ efficiently – indeed, the point of this theorem is to show that it can be computed in exponential time – so we use $S^1(\varphi)$ instead.) We now inductively construct a sequence of structures $M^j = (S^j, \pi^j, \mathcal{K}_1^j, \ldots, \mathcal{K}_n^j)$, $j = 1, 2, 3, \ldots$, with $S^1(\varphi) = S^1 \supseteq S^2 \supseteq S^3 \ldots$. Suppose we have defined S^j. Then define $\mathcal{K}_i^j = \{(s,t) : s, t \in S^j, (s/K_i \cup s/E) \subseteq t\}$, $i = 1, \ldots, n$, and define $\pi^j(s)(p) = \mathbf{true}$ iff $p \in s$. We say a state $s \in S^j$ is *consistent* if (a) for every formula $\neg K_i\psi \in s$, there is a state $t \in S^j$ such that $(s,t) \in \mathcal{K}_i$ and $\neg\psi \in t$, (b) for every formula $\neg E\psi \in s$, there is a state t such that $(s,t) \in \mathcal{K}_1 \cup \ldots \cup \mathcal{K}_n$ such that $\neg\psi \in t$, and (c) for every formula $\neg C\psi \in s$, there is a state $t \in S^j$ reachable from s such that $\neg\psi \in t$. If every state in S^j is consistent and $\varphi \in s$ for some state $s \in S^j$, then return "φ is satisfiable". If there is no consistent state $s \in S^j$ such that $\varphi \in s$, then return "φ is unsatisfiable". Otherwise, let S^{j+1} consist of all the consistent states in S^j, and continue the construction.

Since $S^j \supseteq S^{j+1}$ and S^1 has at most $2^{3|\varphi|}$ elements, this construction must halt after at most exponentially many stages. Computing which states of S^j are consistent can be done in time polynomial in the size of S^j, which is at most exponential in the size of φ. Thus, the whole construction can be carried out in deterministic exponential time. It is easy to show (by induction on the structure of formulas) that if all the states in S^j are consistent, then for all states $s \in S^j$ and all formulas $\psi \in Sub_C^+(\varphi)$, we have $(M,s) \models \psi$ iff $\psi \in s$. Thus, it follows that ψ is satisfiable. Similar arguments to those used in Theorem 4.3 show that if a state s is inconsistent, then φ_s, the conjunction of all the formulas in s, is provably inconsistent. It follows that if there are no states $s \in S^j$ such that $\varphi \in s$, then φ is inconsistent and hence unsatisfiable. The correctness of the algorithm now follows. We leave further details to the reader. ∎

7 Conclusions

We have investigated various modal logics of knowledge and belief. Our emphasis has been on complete axiomatizations and decision procedures. We showed that the standard complete axiomatizations for the well-known logics K, T, S4, S5, and KD45 extends straightforwardly to the case of many agents, and can accommodate common knowledge and distributed knowledge in a natural way. We also showed that while the single-agent case of S5 and KD45 has a decision procedure no worse than that of propositional logic (*NP*-complete), the complexity increases to *PSPACE*-complete when we move to the multi-agent case. We also obtain *PSPACE*-complete decision procedures for the logics K_n, T_n, and $\mathrm{S4}_n$, for both the single- and multi-agent case. Finally, we showed that adding common knowledge to the language causes another substantial increase in complexity, to exponential time.

In many applications of reasoning about knowledge, we want to reason about time as well as knowledge. Various logics of knowledge and time have been investigated recently [HV89]; it turns out that the complexity of reasoning about knowledge and time depends in subtle ways on the assumptions we make about the interaction between knowledge and time. In particular, if we assume that agents do not forget (an assumption frequently made in the literature, for example in [Moo85]), then the language with common knowledge and time turns out to be highly undecidable, and has no complete axiomatization.

It is reasonable at this point to consider to what extent these logics really do capture our intuitive notions. Our feeling in this regard is that there are several useful notions of knowledge and belief; some of them are captured by these logics, others are not. For example, consider a processor in a given distributed system that has received a certain set of messages (or a robot that has observed a certain set of events). There are a number of global states of the system ("possible worlds") that are consistent with the processor having received these messages (or the robot having made these observations). We can say that the processor knows φ in this case if φ is true in all these global states. Note that this is an "external" interpretation of knowledge, that does not require a processor to perform any reasoning to obtain knowledge, or even to be "aware" of this knowledge. This interpretation of knowledge precisely satisfies the $S5_n$ axioms, and turns out to be quite useful in practice (see [HM90] for further discussion).

When it comes to formalizing the reasoning of a knowledge base or of humans, computational complexity must be taken into account. On the other hand, we must be careful in interpreting the lower bounds on complexity we have presented in the previous sections. These are *worst-case* results, and there is no reason to believe that most cases of interest should act like the worst case. Indeed, the evidence suggests that just the opposite is true. The complexity of deciding formulas that humans are interested in tends to be much better than the worst-case analysis would indicate. Indeed, experience with theorem provers for linear-time temporal logic, a modal logic whose satisfiability problem is *PSPACE*-complete, has been quite promising [BG88]. This suggests that theorem proving may be practically feasible for many cases of interest, even for the many-knower versions of the logics we have been considering. Moreover, what is often needed in practice is not checking for validity, but model checking, that is, checking whether a given formula is true in a given model. As we showed in Proposition 3.1, model checking is also often feasible in practice (at least, as long as our structures do not get too large).

Nevertheless, these observations suggest that although the logics we have been considering may provide good approximations to the reasoning carried out by a knowledge base, they still do not seem to be realistic models for human reasoning. Humans simply do not seem to be *logically omniscient* [Hin75], in the sense of Theorem 2.1: they do not know all tautologies, nor is their knowledge closed under deduction (i.e., it does not satisfy $[K_i\varphi \wedge K_i(\varphi \Rightarrow \psi)] \Rightarrow K_i\psi$). A number of attempts have been made to modify the possible-worlds framework to provide a more realistic semantic model of human reasoning. Most of these attempts have involved either allowing non-classical "impossible" worlds in addition to the regular possible worlds [Cre73, Ran82], using a non-classical truth assignment [Lev84b, FH88], or enriching the possible worlds with a syntactic "awareness" function [FH88]. An attempt that explicitly models agents as being able to perform only computations of bounded complexity appears in [Mos88]. While none of these attempts appears as yet to provide the definitive solution, they do suggest that there is sufficient flexibility in the possible-worlds approach to make it worth pursuing.

Acknowledgements: We would like to thank Martín Abadi, Shai Ben David, Ron Fagin, Haim Gaifman, Adam Grove, Bob Moore, Nils Nilsson, and Moshe Vardi, for their helpful comments and criticisms. The first author would also like to thank the students of Stanford course CS400B in 1985, particularly Ian Pratt, for numerous interesting discussions on distributed knowledge.

References

[Ben85] J. F. A. K. van Benthem. *Modal Logic and Classical Logic.* Bibliopolis, Naples, 1985.

[Bet59] E. W. Beth. *The Foundation of Mathematics.* North-Holland, Amsterdam, 1959.

[BG88] H. Barringer and G. D. Gough. Mechanization of temporal logic, part 1: techniques. Draft available from authors at University of Manchester, 1988.

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

[CKS81] A. K. Chandra, D. Kozen, and L. J. Stockmeyer. Alternation. *Journal of the ACM*, 28:114–133, 1981.

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

[Coo71] S. A. Cook. The complexity of theorem proving procedures. In *Proc. 3rd ACM Symp. on Theory of Computing*, pages 151–158, 1971.

[Cre73] M. J. Cresswell. *Logics and Languages.* Methuen and Co., London, 1973.

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

[EH85] E. A. Emerson and J. Y. Halpern. Decision procedures and expressiveness in the temporal logic of branching time. *Journal of Computer and System Sciences*, 30(1):1–24, 1985.

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

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

[FI87] M. J. Fischer and N. Immerman. Interpreting logics of knowledge in propositional dynamic logic with converse. *Information Processing Letters*, 25(3):175–182, 1987.

[FL79] M. J. Fischer and R. E. Ladner. Propositional dynamic logic of regular programs. *Journal of Computer and System Sciences*, 18(2):194–211, 1979.

[FV86] R. Fagin and M. Y. Vardi. Knowledge and implicit knowledge in a distributed environment: preliminary report. In J. Y. Halpern, editor, *Theoretical Aspects of Reasoning about Knowledge: Proc. 1986 Conference*, pages 187–206. Morgan Kaufmann, San Francisco, Calif., 1986.

[Get63] E. Gettier. Is justified true belief knowledge? *Analysis*, 23:121–123, 1963.

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

[Gol92] R. Goldblatt. *Logics of Time and Computation.* CSLI Lecture Notes Number 7. Center for Studies in Language and Information, Stanford University, 2nd edition, 1992.

[Hal87] J. Y. Halpern. Using reasoning about knowledge to analyze distributed systems. In J. F. Traub, B. J. Grosz, B. W. Lampson, and N. J. Nilsson, editors, *Annual Review of Computer Science, Vol. 2*, pages 37–68. Annual Reviews Inc., Palo Alto, Calif., 1987.

[HC68] G. E. Hughes and M. J. Cresswell. *An Introduction to Modal Logic.* Methuen, London, 1968.

[HC84] G. E. Hughes and M. J. Cresswell. *A Companion to Modal Logic.* Methuen, London, 1984.

[Hin61] J. Hintikka. Modalities and quantification. *Theoria*, 27(61):119–128, 1961.

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

[Hin75] J. Hintikka. Impossible possible worlds vindicated. *Journal of Philosophical Logic*, 4:475–484, 1975.

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

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

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

[HU79] J. E. Hopcroft and J. D. Ullman. *Introduction to Automata Theory, Languages and Computation.* Addison-Wesley, New York, 1979.

[HV89] J. Y. Halpern and M. Y. Vardi. The complexity of reasoning about knowledge and time, I: lower bounds. *Journal of Computer and System Sciences*, 38(1):195–237, 1989.

[Imi87] T. Imielinski. Relative knowledge in a distributed database. In *Proc. 6th ACM Symp. on Principles of Database Systems*, pages 197–209, 1987.

[Kan57a] S. Kanger. On the characterization of modalities. *Theoria*, 23:152–155, 1957.

[Kan57b] S. Kanger. *Provability in Logic.* Stockholm Studies in Philosophy I, 1957.

[Kap66] D. Kaplan. Review of "A semantical analysis of modal logic I: normal modal propositional calculi". *Journal of Symbolic Logic*, 31:120–122, 1966.

[KP81] D. Kozen and R. Parikh. An elementary proof of the completeness of PDL. *Theoretical Computer Science*, 14(1):113–118, 1981.

[Kri63] S. Kripke. A semantical analysis of modal logic I: normal modal propositional calculi. *Zeitschrift für Mathematische Logik und Grundlagen der Mathematik*, 9:67–96, 1963. Announced in *Journal of Symbolic Logic*, **24**, 1959, p. 323.

[Lad77] R. E. Ladner. The computational complexity of provability in systems of modal propositional logic. *SIAM Journal on Computing*, 6(3):467–480, 1977.

[Leh84] D. Lehmann. Knowledge, common knowledge, and related puzzles. In *Proc. 3rd ACM Symp. on Principles of Distributed Computing*, pages 62–67, 1984.

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

[Lev84a] H. J. Levesque. Foundations of a functional approach to knowledge representation. *Artificial Intelligence*, 23:155–212, 1984.

[Lev84b] H. J. Levesque. A logic of implicit and explicit belief. In *Proc. National Conference on Artificial Intelligence (AAAI '84)*, pages 198–202, 1984.

[Lev90] H. J. Levesque. All I know: a study in autoepistemic logic. *Artificial Intelligence*, 42(3):263–309, 1990.

[Mak66] D. Makinson. On some completeness theorems in modal logic. *Zeitschrift für Mathematische Logik und Grundlagen der Mathematik*, 12:379–384, 1966.

[McA88] G. L. McArthur. Reasoning about knowledge and belief: a survey. *Computational Intelligence*, 4:223–243, 1988.

[Mer83] M. J. Merritt. *Cryptographic Protocols.* PhD thesis, Georgia Institute of Technology, 1983.

[MH69] J. McCarthy and P. J. Hayes. Some philosophical problems from the standpoint of artificial intelligence. In D. Michie, editor, *Machine Intelligence 4*, pages 463–502. Edinburgh University Press, Edinburgh, 1969.

[Mil81] P. Milgrom. An axiomatic characterization of common knowledge. *Econometrica*, 49(1):219–222, 1981.

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

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

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

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

[Pra79] V. R. Pratt. Models of program logics. In *Proc. 20th IEEE Symp. on Foundations of Computer Science*, pages 115–122, 1979.

[Ran82] V. Rantala. Impossible worlds semantics and logical omniscience. *Acta Philosophica Fennica*, 35:18–24, 1982.

[Rei88] R. Reiter. On integrity constraints. In M. Y. Vardi, editor, *Proc. Second Conference on Theoretical Aspects of Reasoning about Knowledge*, pages 97–112. Morgan Kaufmann, San Francisco, Calif., 1988.

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

[Sat77] M. Sato. A study of Kripke-style methods for some modal logics by Gentzen's sequential method. *Publications Research Institute for Mathematical Sciences, Kyoto University*, 13(2), 1977.

[SM73] L. J. Stockmeyer and A. R. Meyer. Word problems requiring exponential time: preliminary report. In *Proc. 5th ACM Symp. on Theory of Computing*, pages 1–9, 1973.

[Smu68] R. Smullyan. *First-Order Logic*. Springer-Verlag, Berlin/New York, 1968.