ELSEVIER

Artificial Intelligence 75 (1995) 361-372

Artificial Intelligence

Research Note

The effect of bounding the number of primitive propositions and the depth of nesting on the complexity of modal logic

Joseph Y. Halpern

IBM Research Division, Almaden Research Center, Department K53/802, San Jose, CA 95120, USA *

Received October 1994; revised February 1995

Abstract

A well-known result of Ladner says that the satisfiability problem for K45, KD45, and S5 is NP-complete. This result implicitly assumes that there are infinitely many primitive propositions in the language; it is easy to see that the satisfiability problem for these logics becomes linear time if there are only finitely many primitive propositions in the language. By way of contrast, we show that the PSPACE-completeness results of Ladner and Halpern and Moses hold for the modal logics $K_n, T_n, S4_n$, $n \geqslant 1$, and $K45_n, KD45_n, S5_n$, $n \geqslant 2$, even if there is only one primitive proposition in the language. We go on to examine the effect on complexity of bounding the depth of nesting of modal operators. If we restrict to finite nesting, then the satisfiability problem is NP-complete for all the modal logics considered, but S4. If we then further restrict the language to having only finitely many primitive propositions, the complexity goes down to linear time in all cases.

1. Introduction

In [3,5], the complexity of the satisfiability problem for various modal logics is characterized. For the single-agent case, Ladner showed that for K, T, and S4, the problem is PSPACE-complete, while for K45, KD45, and S5, it is NP-complete. Halpern and Moses showed that if we allow two or more agents, the satisfiability problem for all these logics is PSPACE-complete.[1] All the lower bound results implicitly assume that

* Email: halpern@almaden.ibm.com.

[1] Halpern and Moses actually did not consider K45, but their proof for KD45 can be easily modified to deal with K45.

SSDI 0004-3702(95)00018-6

there are an unbounded number of primitive propositions in the language. This is a standard assumption in such complexity results. Indeed, if we consider propositional logic, if the set of primitive propositions is finite and has size, say, K, then the satisfiability problem becomes linear time: To test the satisfiability of a formula φ, we simply test each of the 2^K truth assignments, and see if any of them satisfies φ.

While the assumption that there are infinitely many primitive propositions in the language is standard, it might not always be reasonable. One might well be interested in the complexity of reasoning about knowledge and belief in a particular application, where there are only, say, 10 primitive propositions. Do we get linear time algorithms in this case? It is easy to see that in the three cases where we had NP-completeness before—K45, KD45, and S5—the complexity of satisfiability drops to linear time when there are only finitely many primitive propositions, just as it does with propositional logic. However, as we show in this paper, for all the cases where we had PSPACE-completeness before, we still get PSPACE-completeness even if there is only one primitive proposition in the language. The upper bound, of course, follows immediately from the upper bounds in [3,5]; bounding the number of primitive propositions can only make things easier. The lower bounds apply a technique that may be of independent interest: We isolate some key properties of primitive propositions that are needed to prove the lower bound, and show that the existence of an infinite *pp-like* (primitive-proposition-like) family of formulas suffices for the proof. We then show how to construct such an infinite family for each of the logics in question, using formulas that involve only one primitive proposition.

A closer look at the PSPACE lower bounds shows that, except in the case of S4, they make crucial use of formulas with deeply nested occurrences of modal operators. What happens if we restrict the depth of nesting of modal operators to some fixed k? First suppose we have an infinite number of primitive propositions in the language. In the case of $S4_n$, little changes. As long as we allow formulas of depth $k \geqslant 2$, the PSPACE lower bound still holds. On the other hand, for all the other logics, the complexity goes down to NP-complete: the lower bound is immediate since all these logics contain propositional logic, while the upper bound follows easily from the algorithms given in [3,5].

What happens if, in addition to having a bound on the depth of nesting, we also assume that the language has only finitely many primitive propositions? In that case, the complexity goes down to linear time for all the logics we are considering. The new and old results are summarized in Table 1, where the first row describes the results of [3,5], and the remaining rows describe the results of this paper; Φ is the set of primitive propositions. As these results show, both depth of nesting and the number of primitive propositions in the language play a critical role in complexity.

The strengthened lower bounds show the expressive power of modal logic. They have already found application as a technique for proving other complexity results [2]; we hope they will find further applications.

The rest of this paper is organized as follows. In the next section, we briefly review the semantics of the various logics we are interested in, and discuss why the satisfiability problem for K45, KD45, and S5 is linear time if there is a bound on the number of primitive propositions. In Section 3, we prove the PSPACE lower bound in row 2 of the

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

	K45, KD45, S5	$S4_n$, $n \geqslant 1$	K_n, T_n, $n \geqslant 1$; $K45_n$, $KD45_n$, $S5_n$, $n \geqslant 2$
Φ infinite, unbounded depth	NP-complete	PSPACE-complete	PSPACE-complete
Φ finite, unbounded depth	linear time	PSPACE-complete	PSPACE-complete
Φ infinite, depth $\leqslant k$, $k \geqslant 2$	NP-complete	PSPACE-complete	NP-complete
Φ finite, bounded depth	linear time	linear time	linear time

table. In Section 4, we discuss the effects of bounding the nesting of modal operators. We conclude in Section 5.

2. A brief review of modal logic

We briefly review some standard notions of modal logic here. Further details can be found in, for example, [1,3,4].

In this paper we focus on six logics known as K_n, T_n, $S4_n$, $K45_n$, $KD45_n$, and $S5_n$. The subscript n in all these logics is meant to emphasize the fact that we are considering the n-agent version of the logic. We omit it when considering the single-agent case.

The language we use for all these logics is propositional logic augmented by the modal operators $K_1, \ldots, K_n$, where $K_i\varphi$ can be read "agent i knows (or believes) φ". Formally, we start with a finite or infinite set Φ of primitive propositions. The set of modal formulas, denoted $\mathcal{L}_n(\Phi)$, is the least set containing φ closed under conjunction, negation, and application of $K_1, \ldots, K_n$. Thus, if φ and ψ are formulas in $\mathcal{L}_n(\Phi)$, then so are $\varphi \wedge \psi$, $\neg\varphi$, and $K_i\varphi$.

Consider the following collection of axioms:

P. All instances of axioms of propositional logic.

K. $(K_i\varphi \wedge K_i(\varphi \Rightarrow \psi)) \Rightarrow K_i\psi$.

T. $K_i\varphi \Rightarrow \varphi$.

4. $K_i\varphi \Rightarrow K_iK_i\varphi$.

5. $\neg K_i\varphi \Rightarrow K_i\neg K_i\varphi$.

D. $\neg K_i \mathit{false}$.

and rules of inference:

R1. From φ and $\varphi \Rightarrow \psi$ infer ψ.

R2. From φ infer $K_i\varphi$.

The axioms **4** and **5** are called the *positive introspection axiom* and *negative introspection axiom*, respectively. They are appropriate for agents that are sufficiently introspective so that they know what they know and do not know.

We get various systems by combining some subset of **K**, **T**, **4**, **5**, and **D** with **P**, **R1**, and **R2**. In particular, we get K_n by combining **K** with **P**, **R1**, and **R2**, T_n by adding

T to these axioms, S4$_n$ by adding **4**, S5$_n$ by adding **5**, K45$_n$ by deleting **T** from S5$_n$, and KD45$_n$ by adding **D** to K45$_n$. Numerous other modal logics can be constructed by considering other combinations of axioms.

We give semantics to all these logics by using Kripke structures. A Kripke structure is a tuple $(W, \pi, \mathcal{K}_1, \ldots, \mathcal{K}_n)$, where W is a set of *worlds*, π associates with each world a truth assignment to the primitive propositions, so that $\pi(w)(p) \in \{\mathbf{true}, \mathbf{false}\}$ for each world w and primitive proposition $p \in \Phi$, and $\mathcal{K}_1, \ldots, \mathcal{K}_n$ are binary *accessibility* relations.

Recall that a binary relation $\mathcal{K}$ on W is *reflexive* if $(w,w) \in \mathcal{K}$ for all $w \in W$, *transitive* if $(u,v) \in \mathcal{K}$ and $(v,w) \in \mathcal{K}$ implies $(u,w) \in \mathcal{K}$, *Euclidean* if $(u,v) \in \mathcal{K}$ and $(u,w) \in \mathcal{K}$ implies $(v,w) \in \mathcal{K}$, and *serial* if for all $w \in W$, there is some w' such that $(w,w') \in \mathcal{K}$. Let $\mathcal{M}_n(\Phi)$ be the class of all Kripke structures for the language $\mathcal{L}_n(\Phi)$. Thus, in every structure in $\mathcal{M}_n(\Phi)$, the interpretation π gives semantics to the primitive propositions in Φ, and there are accessibility relations $\mathcal{K}_1, \ldots, \mathcal{K}_n$. We restrict $\mathcal{M}_n(\Phi)$ by using superscripts r, s, t, and e, to denote reflexive, serial, transitive, and Euclidean, structures, respectively. Thus, $\mathcal{M}_n^{\mathrm{rt}}(\Phi)$ denotes the class of all structures where the $\mathcal{K}_i$ relations are reflexive and transitive knowledge, $\mathcal{M}_n^{\mathrm{est}}(\Phi)$ denotes the class of all structures where the $\mathcal{K}_i$ relations are Euclidean, serial, and transitive, and so on.

A *situation* is a pair (M,w) consisting of a Kripke structure and a world w in M. We give semantics to formulas with respect to situations. If p is a primitive proposition, then $(M,w) \models p$ if $\pi^M(w)(p) = \mathbf{true}$. Conjunctions and negations are dealt with in the standard way. Finally,

$$(M,w) \models K_i\alpha \quad \text{iff} \quad (M,w') \models \alpha \text{ for all } w' \in \mathcal{K}_i^M(w).$$

As usual, we say that a formula φ is *valid in a structure* M, written $M \models \varphi$, if $(M,w) \models \varphi$ for all worlds w in M. We say that φ is *valid with respect to a class* $\mathcal{N}$ *of structure* if $M \models \varphi$ for all structures $M \in \mathcal{N}$. Similarly, we say that φ is *satisfiable* with respect to $\mathcal{N}$ if $(M,w) \models \varphi$ for some $M \in \mathcal{N}$ and some world w in M.

It is well known that there is a close connection between conditions placed on $\mathcal{K}$ and the axioms. In particular, **T** corresponds to the $\mathcal{K}_i$'s being reflexive, **4** to the $\mathcal{K}_i$'s being transitive, **5** to the $\mathcal{K}_i$'s being Euclidean, and **D** to the $\mathcal{K}_i$'s being serial. Thus, we get the following result (see [1,3,4] for proofs):

Theorem 2.1. K$_n$ *(respectively* T$_n$, S4$_n$, KD45$_n$, K45$_n$, S5$_n$*) is a sound and complete axiomatization for the language* $\mathcal{L}_n(\Phi)$ *with respect to* $\mathcal{M}_n(\Phi)$ *(respectively* $\mathcal{M}_n^{\mathrm{r}}(\Phi)$, $\mathcal{M}_n^{\mathrm{rt}}(\Phi)$, $\mathcal{M}_n^{\mathrm{est}}(\Phi)$, $\mathcal{M}_n^{\mathrm{st}}(\Phi)$, $\mathcal{M}_n^{\mathrm{ret}}(\Phi)$*).* [2]

An $\mathcal{S}$*-situation* (for $\mathcal{S} \in \{\mathrm{K}_n, \mathrm{T}_n, \mathrm{S4}_n, \mathrm{K45}_n, \mathrm{KD45}_n, \mathrm{S5}_n\}$) is a situation (M,w) where M satisfies the appropriate restriction; thus, for example (M,w) is a S4$_n$-situation if $M \in \mathcal{M}_n^{\mathrm{rt}}(\Phi)$. We say that a formula is $\mathcal{S}$*-satisfiable* if it is true in some $\mathcal{S}$-situation.

[2] The more common characterization of structures that characterize S5$_n$ is that the $\mathcal{K}_i$'s in these structures are equivalence relations. As observed in [3], $\mathcal{K}$ is an equivalence relation iff $\mathcal{K}$ is reflexive, Euclidean, and transitive, so these characterizations coincide.

In the single-agent case of $\mathrm{KD45}_n$, $\mathrm{K45}_n$, and $\mathrm{S5}_n$, we can consider a simpler class of structures. We define a *K45 situation* to be a pair (W, w), where W is a set of truth assignments that, intuitively, characterize the worlds the agent considers possible, and w is a truth assignment that, intuitively, characterizes the "real world". A *KD45 situation* is a K45 situation (W, w) such that $W \neq \emptyset$. An *S5 situation* (W, w) is a K45 situation such that $w \in W$.

We again give semantics to formulas with respect to situations. If p is a primitive proposition, then $(W, w) \models p$ if p is true under truth assignment w. Conjunctions and negations are dealt with in the standard way. Finally,

$$(W, w) \models K\alpha \quad \text{iff} \quad (W, w') \models \alpha \text{ for all } w' \in W.[3]$$

It is well known (again, see any of [1,3,4] for a proof) that a formula is provable in K45 (respectively KD45, S5) if and only if it is true in all K45 (respectively KD45, S5) situations.

Notice that if we start with a finite set Φ of primitive propositions, there are $2^{|\Phi|}$ truth assignments to the propositions in Φ, and hence no more than $2^{2^{|\Phi|}}2^{|\Phi|}$ K45 (respectively KD45, S5) situations. Checking whether a formula φ is satisfied in any one of these situations can be done in time linear in the length of $|\varphi|$ (see [3, Proposition 3.1]). Thus, to see if φ is satisfiable, we can simply check each of these structures. Since the number of structures is independent of the size of φ, we get:

Proposition 2.2. *If Φ is finite, deciding if a formula is* K45- *(respectively,* KD45-, S5-*) satisfiable can be done in linear time.*

Note that if Φ is infinite, the number of structures we have to check is not independent of the formula (it depends on the number of primitive propositions in the formula), so this argument fails.

3. The PSPACE lower bounds

We begin by reviewing the lower bound proofs of [3,5], since we plan to follow the same strategy here. The proofs proceed by a reduction from the logic of *quantified Boolean formulas* (QBF). For our purposes, we can take a QBF to be of the form $Q_1 p_1 Q_2 p_2 \dots Q_m p_m A'$, where $Q_i \in \{\forall, \exists\}$ and A' is a propositional formula whose only primitive propositions are among $p_1, \dots, p_m$. 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 (respectively B_1) is B with all occurrences of p_i replaced by *true* (respectively *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

[3] When dealing with logics like K45, where only one agent is involved, we typically do not subscript the K operator, writing, for example, $K\alpha$ rather than $K_1\alpha$.

original formula (in fact, exponential in the size of the original formula). It is known that the problem of determining which QBFs are true is PSPACE-complete [6].

Following [3], we present the lower bound proof for S4, and then show how to modify it to deal with all the other logics. Suppose we are given a QBF $A = Q_1p_1 \ldots Q_mp_mA'$. We construct a formula ψ_A^{S4} that is satisfiable in a structure in $\mathcal{M}_1^{rt}$ iff A is true. The idea is to use ψ_A^{S4} to force the structure to look like a tree of truth assignments. Each of the leaves of the tree encodes a distinct truth assignment to the primitive propositions $p_1, \ldots, p_m$ that appear in A. If A is satisfiable, then we want this tree to contain all the truth assignments necessary to show that A is true.

We proceed as follows. 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. Notice that the number of propositions used depends on A. As the depth of A increases, we need more and more primitive propositions. This is precisely where the proof implicitly assumes that the set of primitive propositions is infinite (although, of course, for any fixed A, we use only finitely many of them). Let *depth* be the following formula, which clearly captures the intended relation between the d_i's:

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

Let *determined* be a formula that intuitively says that the truth value of p_i is determined by depth i in the tree, in that if p_i is true (respectively false) at a given node s of depth j with $j \geqslant i$, then it is true (respectively false) at all the $\mathcal{K}$-successors of s of depth at least i. (If we restrict to structures that look like trees, then all the $\mathcal{K}$-successors of s will have depth $i+1$; however, there may be nonstandard structures satisfying this formula that do not look like trees, and we need to be able to deal with these as well.)

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

Let $branching_A$ be a formula that intuitively says that if Q_{i+1}, the $(i+1)$st quantifier in A, is $\forall$, then each node s at depth i in the tree has two successors of depth $i+1$, one at which p_{i+1} is true, and one at which p_{i+1} is false, while if Q_{i+1} is $\exists$, then s has at least one successor of depth $i+1$ (which intuitively gives p_{i+1} the truth value which results in A being true).

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

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

We take ψ_A^{S4} to be

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

As shown in [3], ψ_A^{S4} is satisfiable in a structure in $\mathcal{M}_1^{rt}$ iff A is true. One direction is easy: we can use the truth assignments that make A true to guide the construction of a tree that satisfies ψ_A^{S4}. 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 $Q_{j+1}p_{j+1} \ldots Q_m p_m A'$ and replacing all occurrences of p_i, $i \leqslant j$, by *true* if $\pi(t)(p_i) = \mathbf{true}$, and by *false* otherwise. Note that $A_0^t = A$ and that 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_m^t 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_{m-j}^t is true. Since $(M, s) \models d_0$, in particular we have that $A_0^s = A$ is true. Since ψ_A^{S4} is polynomial in the length of A, this gives us the desired polynomial reduction from QBF to S4. It follows that S4 satisfiability is PSPACE-hard.

To deal with the other logics, we modify ψ_A^{S4} as follows: We take ψ_A^{T} to be

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

where $K^m\varphi$ is an abbreviation for $K \ldots K\varphi$, with m K's; we take ψ_A^{K} to be

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

finally, we take ψ_A^{K45}, ψ_A^{KD45}, and ψ_A^{S5} all to be the result of replacing all occurrences of K in ψ_A^{T} by K_2K_1. It is shown in [3] that A is true iff ψ_A^{T} (respectively ψ_A^{K}, ψ_A^{K45}, ψ_A^{KD45}, ψ_A^{S5}) is satisfiable in a structure $\mathcal{M}_1^{r}(\Phi)$ (respectively, $\mathcal{M}_1(\Phi)$, $\mathcal{M}_2^{et}(\Phi)$, $\mathcal{M}_2^{est}(\Phi)$, $\mathcal{M}_2^{ret}(\Phi)$). This proves all the other PSPACE-hardness results.

As we observed above, this proof seems to make crucial use of the fact that we have an unbounded number of primitive propositions in Φ. In addition, the depth of nesting of the K operator in the formulas constructed is unbounded in the case of all logics other than S4. We defer the issue of depth to the next section, and focus here on the number of primitive propositions required, showing how the proof can be carried out with only one primitive proposition in the language. We first need to isolate what properties of primitive propositions we actually use. One property we obviously use is that primitive propositions are *independent*. To make this precise, given any formulas $\varphi_1, \ldots, \varphi_m$, we define an *atom* over $\varphi_1, \ldots, \varphi_m$ to be one of the 2^m formulas of the form $\varphi_1' \wedge \cdots \wedge \varphi_m'$, where φ_i' is either φ_i or $\neg\varphi_i$. We say that $\varphi_1, \ldots, \varphi_m$ are *independent with respect to logic* $\mathcal{S}$ if each of the atoms over $\varphi_1, \ldots, \varphi_m$ is $\mathcal{S}$-consistent. But independence alone does not suffice. For example, our proof implicitly uses the fact that both $p_i \wedge \neg K \neg p_{i+1}$ and $p_i \wedge \neg K p_{i+1}$ are satisfiable. But suppose φ_1 is p and φ_2 is Kp. Then φ_1 and φ_2 are easily seen to be independent with respect to K, yet we have that $\varphi_2 \Rightarrow K\varphi_1$ is valid. We can construct similar examples for each of the other logics we are interested in. We need a notion that is stronger than independence. What we really want to be able to do is to construct an arbitrary tree of truth assignments to $\varphi_1, \ldots, \varphi_m$, as we did for the primitive propositions in our lower bound proof.

To make this precise, we define a *tree formula* over $\varphi_1, \ldots, \varphi_m$ inductively to be either an atom over $\varphi_1, \ldots, \varphi_m$ or a conjunction of the form $\psi \wedge \neg K \neg \sigma_1 \wedge \cdots \wedge \neg K \neg \sigma_k$, where ψ is an atom over $\varphi_1, \ldots, \varphi_m$ and $\sigma_1, \ldots, \sigma_k$ are tree formulas over $\varphi_1, \ldots, \varphi_m$.

We can think of a tree formula as describing a tree, each of whose nodes is labeled by a truth assignment to the propositions $\varphi_1, \ldots, \varphi_m$. We say that the formulas $\varphi_1, \ldots, \varphi_m$ are *completely independent* with respect to a logic $\mathcal{S}$ if each tree formula over $\varphi_1, \ldots, \varphi_m$ is $\mathcal{S}$-consistent. Finally, we say that an infinite family $\varphi_1, \varphi_2, \ldots$ of formulas is *pp-like* for a logic $\mathcal{S}$ if each finite subset of these formulas is completely independent with respect to $\mathcal{S}$. Clearly, any infinite set of distinct primitive propositions is indeed pp-like, no matter what the logic. Our goal is to construct pp-like families of formulas $\varphi_1, \varphi_2, \ldots$ for each of the logics of interest to us that involve just one primitive proposition in such a way that the length of the formula φ_n is polynomial in n. Once we do this, we can replace the primitive propositions that appear in formulas such as ψ_A^{K} by the pp-like formulas; it is easy to see that the lower bound proof then goes through unchanged. Thus, we can view the notion of a pp-like family as encapsulating what we really needed from primitive propositions in our lower bound proof.

Constructing a pp-like family for the logic K is quite simple: Let q_j, $j \geqslant 1$, be the formula $\neg K \neg(\neg p \wedge \neg K^j \neg p)$. It is easy to see that q_j is true at a state s precisely if there is a path of length $j+1$ starting at s whose second state satisfies $\neg p$ and whose last state satisfies p. More precisely, $(M, s) \models \neg K^j \neg p$ precisely if there is a sequence $s_0, \ldots, s_j$ such that (a) $s_0 = s$, (b) $(s_i, s_{i+1}) \in \mathcal{K}_1$ for $i < j$, (c) $(M, s_1) \models \neg p$, and (d) $(M, s_j) \models p$. It is easy to see that the family $q_1, q_2, \ldots$ is indeed pp-like for the logic K. We can thus replace of $p_1, \ldots, p_m, d_0, \ldots, d_{m+1}$ in ψ_A^{K} by $q_1, \ldots, q_{2m+2}$ respectively. The resulting formula is satisfiable in $\mathcal{M}$ iff A is true. This shows that deciding K-satisfiability for formulas in $\mathcal{L}_1(\{p\})$ is PSPACE-hard.

This argument no longer works for T. The problem is that in T, $q_j \Rightarrow q_{j'}$ is valid if $j' \geqslant j$: if there is an appropriate path of length $j+1$ to a state satisfying p, reflexivity guarantees that there will also be longer paths. We deal with this as follows. Let r_1 be q_1, and let r_j be an abbreviation for $q_j \wedge \neg q_{j-1}$ for $j > 1$. Thus, r_j says that there is an appropriate path of length j to p, but no shorter paths. It is easy to see that the r_j's are satisfiable and mutually exclusive. Moreover, the formulas $\neg K \neg r_j$, $j = 1, 2, \ldots$, form a pp-like family for T. Thus, we can replace the occurrences of $p_1, \ldots, p_m, d_0, \ldots, d_{m+1}$ in ψ_A^{T} by $\neg K \neg r_1, \ldots, \neg K \neg r_{2m+2}$, respectively, and still get the PSPACE lower bound in this case.

This argument breaks down for S4. In transitive structures, it is easy to see that $q_j \Leftrightarrow q_{j'}$ is valid for all j, j', so r_j is inconsistent! In fact, it can be shown that there is no infinite pp-like family for S4 if we have only a finite number of primitive propositions in the language. We can get an infinite family satisfying a slightly weaker property though, as we now explain. Consider the PSPACE lower bound proof again. We say that a formula φ is *evident in structure M* if the formula $\varphi \Rightarrow K\varphi$ is valid in M. Notice that if ψ_A^{S4} is satisfiable, then it is satisfiable in a structure where all the primitive propositions are evident. In the case of the primitive propositions $d_0, \ldots, d_{m+1}$ that are meant to denote the depth, it is clear that we want them to be evident. As for the propositions $p_1, \ldots, p_m$, notice that we can make them all false at the root. In fact, we can make p_i false at all nodes of depth less than i (i.e., at all nodes not satisfying d_i). To satisfy the formula $branching_A$, there may need to be nodes of depth i satisfying p_i, but once p_i is true at such a node, the formula *determined* guarantees that it remains true. Thus, in the structure constructed in this way, each primitive proposition p_i is evident.

We define an *evident tree formula* over $\varphi_1, \ldots, \varphi_m$ inductively to be either an atom over $\varphi_1, \ldots, \varphi_m$ or a conjunction of the form $\psi \wedge \neg K \neg \sigma_1 \wedge \cdots \wedge \neg K \neg \sigma_k$, where ψ is an atom over $\varphi_1, \ldots, \varphi_m$ and $\sigma_1, \ldots, \sigma_k$ are evident tree formulas over $\varphi_1, \ldots, \varphi_m$ such that if φ_i appears as a conjunct of ψ, then $\neg\varphi_i$ does not appear as a conjunct of $\sigma_1, \ldots, \sigma_k$. Thus, an evident tree formula describes a tree whose nodes are labeled with truth assignments to the propositions $\varphi_1, \ldots, \varphi_m$ with the added property that if φ_i is true at a node, it is true at all the successors of that node—i.e., , φ_i is evident—for $i = 1, \ldots, m$. We say that the formulas $\varphi_1, \ldots, \varphi_m$ are *weakly independent* with respect to a logic $\mathcal{S}$ if each evident tree formula over $\varphi_1, \ldots, \varphi_m$ is $\mathcal{S}$-consistent. Finally, we say that an infinite family $\varphi_1, \varphi_2, \ldots$ of formulas is *weakly pp-like* for a logic $\mathcal{S}$ if each finite subset of these formulas is weakly independent with respect to $\mathcal{S}$. Of course, a pp-like family is weakly pp-like, but the converse may not hold. By our observations above, to get a PSPACE lower bound for S4, it actually suffices to construct a weakly pp-like family for S4. We now show how to do this.

We take $\widehat{q_1}$ to be $\neg K \neg Kp$. Suppose we have defined $\widehat{q_1}, \ldots, \widehat{q_m}$. We define $\widehat{q_{m+1}}$ to be $\neg K \neg (p \wedge \neg K \neg (\neg p \wedge \widehat{q_m}))$. It is not hard to show that $(M,s) \models \widehat{q_j}$ iff there is a path $s_0, s_1, \ldots, s_{2j-1}$ such that (a) $s_0 = s$, (b) $(s_i, s_{i+1}) \in \mathcal{K}_1$ for $i < 2j-1$, (c) $(M, s_{2k-1}) \models p$ for $k = 1, \ldots, j-1$, (d) $(M, s_{2k}) \models \neg p$ for $k = 1, \ldots, j-1$, and (d) $(M, s_{2j-1}) \models Kp$. In transitive structures, $\widehat{q_k} \Rightarrow \widehat{q_j}$ is valid if $k \geqslant j$, so we still do not have independence, let alone a weakly pp-like family. Let $\widehat{q'_j}$ be an abbreviation for $\widehat{q_j} \wedge \neg \widehat{q_{j+1}}$. Note that $\widehat{q'_j}$ holds if there is a path such as the one above of length $2j-1$ and no longer path. Thus, the formulas $\widehat{q'_j}$, $j = 1, 2, 3, \ldots$ are mutually exclusive. Let $\widehat{r_j}$ be an abbreviation for $\neg K \neg \widehat{q'_j} \wedge K(\widehat{q'_j} \Rightarrow \neg K \neg K \neg p)$. It is not hard to show that $\widehat{r_1}, \widehat{r_2}, \ldots$ forms a weakly pp-like family for S4. Given a tree labeled with truth assignments to the φ_i's, each of which is evident (so that if φ_i is true at a given node in the tree, then φ_i is also true at all nodes below it), we must construct an S4 structure corresponding to this tree. Roughly speaking, the idea is that we augment the tree in such a way that for each node where $\neg \widehat{r_k}$ is true, we make sure that there is a successor that satisfies $\widehat{q'_j} \wedge K \neg K \neg p$. We leave details to the reader.

We remark that $\widehat{r_1}, \widehat{r_2}, \ldots$ is not a pp-like family. For example, it is not hard to show that $\varphi =_{\text{def}} \widehat{r_j} \wedge \neg K(\widehat{r_j} \wedge \neg K \neg \widehat{r_j})$ is not S4-satisfiable. To see this, suppose that $(M,s) \models \varphi$. Then there must be states t and u such that $(s,t) \in \mathcal{K}$, $(t,u) \in \mathcal{K}$, and $(M,s) \models \widehat{r_j}$, $(M,t) \models \neg \widehat{r_j}$, and $(M,u) \models \widehat{r_j}$. Since $(M,s) \models \widehat{r_j}$, it follows that

$$(M,s) \models K(\widehat{q'_j} \Rightarrow \neg K \neg K \neg p).$$

Since we are dealing with S4, it follows that

$$(M,t) \models K(\widehat{q'_j} \Rightarrow \neg K \neg K \neg p). \tag{1}$$

By assumption, $(M,t) \models \neg \widehat{r_j}$. From (1) and the definition of $\widehat{r_j}$, it follows that

$$(M,t) \models K \neg \widehat{q'_j}.$$

Since we are dealing with S4, we must also have

$$(M,u) \models K\neg\widehat{q'_j}. \tag{2}$$

But (2) contradicts the assumption that $(M,u) \models \widehat{r_j}$. If we now take φ to be an atom that includes $\widehat{r_j}$ as a conjunct and ψ to be an atom that includes $\neg\widehat{r_j}$ as a conjunct, it follows that the tree formula $\varphi \wedge \neg K\neg(\psi \wedge \neg K\neg\varphi)$ is not S4-consistent. Thus, $\widehat{r_1}, \widehat{r_2}, \ldots$ is not pp-like.[4] Fortunately, as we observed above, it suffices to have a weakly pp-like family to get the lower bound, so we still get the PSPACE lower bound in the case of S4.

These techniques will not give us a weakly pp-like family in the case of K45, KD45, or S5. Indeed, we cannot find a weakly pp-like family if we have only a finite number of primitive propositions and one agent in these cases. But once we have two agents, it is easy to check that we obtain a pp-like family for each of $K45_n$, $KD45_n$, and $S5_n$, $n \geqslant 2$, by replacing each occurrence of K in the family $q_1, q_2, \ldots$ constructed for T by K_2K_1. We again leave details to the reader.

We can summarize this discussion by the following theorem:

Theorem 3.1.
(1) *The satisfiability problem for the logics* K, T, *and* S4 *is PSPACE-hard with respect to the language* $\mathcal{L}_1(\{p\})$.
(2) *The satisfiability problem for* $K45_2$, $KD45_2$, *and* $S5_2$ *is PSPACE-hard with respect to the language* $\mathcal{L}_2(\{p\})$.

4. Bounding the depth

As we have seen, the PSPACE lower bound for K_n, T_n, $n \geqslant 1$, and $K45_n$, $KD45_n$, $S5_n$, $n \geqslant 2$, uses formulas with unbounded nesting of the modal operators. What happens if we bound the depth?

To make this precise, we formally define the depth of nesting in a formula φ, denoted $depth(\varphi)$, as follows: We define $depth(p) = 0$ if p is a primitive proposition, $depth(\neg\varphi) = depth(\varphi)$, $depth(\varphi \wedge \psi) = \max(depth(\varphi), depth(\psi))$, and $depth(K_i\varphi) = 1 + depth(\varphi)$. Thus, the formulas of depth 0 are precisely the propositional formulas, and a formula such as $K_1(K_2 \wedge \neg K_2K_2q)$ has depth 3. Let $\mathcal{L}_n^k(\Phi)$ consist of all formulas in the language $\mathcal{L}_n(\Phi)$ whose depth is at most k.

Notice that the formula ψ_A^{S4} has depth 2, independent of A. This shows that even if we restrict to $\mathcal{L}_n^2(\Phi)$, the PSPACE lower bound holds for S4 as long as Φ has infinitely many primitive propositions. On the other hand, the formulas ψ_A^{K} and ψ_A^{T} have depth $m+2$, where m is the number of primitive propositions in A. Is such unbounded depth really necessary? As the upper bound proofs given in [3,5] show, the answer is yes. Roughly speaking, for K_n, T_n, $K45_n$, $KD45_n$, and $S5_n$, $n \geqslant 1$, if a formula of depth k is satisfiable at all, it is satisfiable in a structure which looks like a tree of depth at most k with outdegree at most the length of the formula. Thus, to check if a formula is satisfiable, it suffices to guess such a small treelike structure that satisfies it. Since

[4] We remark that this argument can be extended to show that there is no infinite pp-like family for S4 that uses only a finite number of primitive propositions.

a treelike structure of depth k and outdegree m has fewer than m^{k+1} nodes, it follows that checking satisfiability for formulas in $\mathcal{L}_n^k\Phi$ is in NP for these logics. Since all these logics contain propositional logic as a sublanguage, we immediately get that satisfiability is NP hard. Thus, we get:

Theorem 4.1. *For any fixed k, if Φ is infinite, the satisfiability problem for* K_n, T_n, $\mathrm{K45}_n$, $\mathrm{KD45}_n$, $\mathrm{S5}_n$, $n \geqslant 1$, *with respect to the language $\mathcal{L}_n^k(\Phi)$ is NP-complete.*

By way of contrast, we have

Theorem 4.2. *Suppose Φ is infinite.*

(a) *If $k \geqslant 2$ the satisfiability problem for* $\mathrm{S4}_n$, $n \geqslant 1$, *with respect to the language $\mathcal{L}_n^k(\Phi)$ is PSPACE-complete.*
(b) *The satisfiability problem for* $\mathrm{S4}_n$, $n \geqslant 1$, *with respect to the language $\mathcal{L}_n^1(\Phi)$ is NP-complete.*

What happens if we further restrict to finite Φ?

Theorem 4.3. *For any fixed k, if Φ is finite, deciding if a formula in $\mathcal{L}_n^k(\Phi)$ is satisfiable with respect to any of the logics* K_n, T_n, $\mathrm{S4}_n$, $\mathrm{K45}_n$, $\mathrm{KD45}_n$, $\mathrm{S5}_n$, $n \geqslant 1$, *can be done in linear time.*

Proof. A straightforward induction on k shows that for each of these logics, there are only finitely many inequivalent formulas in $\mathcal{L}_n^k(\Phi)$. Indeed, given n and k, we can easily construct formulas $\varphi_1, \ldots, \varphi_N$ such that every formula is equivalent to one of $\varphi_1, \ldots, \varphi_N$ in the logic K_n, and hence in all the other logics. Fix a logic $\mathcal{S}$, and consider the subset of $\varphi_1, \ldots, \varphi_N$ that is $\mathcal{S}$-satisfiable. This means that there is a finite collection of structures $M_1, \ldots, M_K$ (where $K \leqslant N$), such that every formula in $\mathcal{L}_n^k(\Phi)$ that is $\mathcal{S}$-satisfiable is satisfiable in one of these structures. Thus, to check if a formula φ is satisfiable, we simply check if it is satisfiable in each of these structures. This can be done in time linear in the size of φ. (Of course, the constant depends on K and the size of the structures $M_1, \ldots, M_K$. While this means it may be huge, it is nonetheless a constant.) □

We remark that since it is well known that, in the logics K45, KD45, and S5, every formula is equivalent to a depth-one formula [4, p. 55] for these logics,[5] we do not need to bound k to get the linear time result (as we observed in Proposition 2.2).

5. Conclusions

We have shown the effect of bounding the depth and bounding the number of primitive propositions on the complexity of reasoning about knowledge. Basically, we get linear

[5] The proof in [4] is given only for S5, but the identical arguments work for KD45 and K45.

time algorithms only in the case that our language is restrictive enough so that there are only finitely many inequivalent formulas. These results show how little it takes to get up to NP or PSPACE complexity.

Our results form an interesting contrast to those of Vardi [7]. By working in the framework of *Montague structures*, which are more general than the Kripke structures we consider here, he is able to do a fine-grained analysis of which axioms cause the complexity of knowledge to increase. He shows that, in a precise sense, it is the property of closure under conjunction—$(K\varphi \wedge K\psi) \Rightarrow K(\varphi \wedge \psi)$—that increases the complexity of satisfiability from NP to PSPACE if we have infinitely many propositions. It would be interesting to understand the effect of restricting the depth and the number of primitive propositions in this more general framework as well. It would also be of interest to see the effect of such limitations on other modal logics, such as temporal logic and dynamic logic. More generally, given a modal logic characterized by a collection $\mathcal{F}$ of *frames* [1] (as S4 is characterized, for example, by the transitive reflexive frames), it would be interesting to find conditions of $\mathcal{F}$ that guarantee that the satisfiability problem is polynomial time (or NP or PSPACE) if we restrict the language to having only finitely many primitive propositions and/or finite depth of nesting.

Acknowledgments

I wish to thank the two referees of their paper for their very useful comments and suggestions.

References

[1] B.F. Chellas, *Modal Logic* (Cambridge University Press, Cambridge, England, 1980).

[2] J.Y. Halpern, A theory of knowledge and ignorance for many agents, Research Report RJ 9894, IBM Research Division, Almaden Research Center, San Jose, CA (1994).

[3] J.Y. Halpern and Y. Moses, A guide to completeness and complexity for modal logics of knowledge and belief, *Artif. Intell.* **54** (1992) 319–379.

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

[5] R.E. Ladner, The computational complexity of provability in systems of modal propositional logic, *SIAM J. Comput.* **6** (3) (1977) 467–480.

[6] L.J. Stockmeyer and A.R. Meyer, Word problems requiring exponential time: preliminary report, in: *Proceedings Fifth ACM Symposium on Theory of Computing*, (1973) 1–9.

[7] M.Y. Vardi, On epistemic logic and logical omniscience, in: J.Y. Halpern, ed., *Theoretical Aspects of Reasoning about Knowledge: Proceedings of the 1986 Conference*, San Francisco, CA (1986) 293–305.