SIAM J. COMPUT.
Vol. 6, No. 3, September 1977

THE COMPUTATIONAL COMPLEXITY OF PROVABILITY IN SYSTEMS OF MODAL PROPOSITIONAL LOGIC*

RICHARD E. LADNER†

Abstract. The computational complexity of the provability problem in systems of modal propositional logic is investigated. Every problem computable in polynomial space is log space reducible to the provability problem in any modal system between K and $S4$. In particular, the provability problem in K, T, and $S4$ are log space complete in polynomial space. The nonprovability problem in S5 is log space complete in nondeterministic polynomial time.

Key words. modal logic, computational complexity

Introduction. We investigate the computational complexity of deciding whether or not a modal propositional formula is provable in certain systems of modal propositional logic, including K, T, $S4$, and $S5$. In terms to be defined later we show (using a suggestion of S. K. Thomason) that if S is a modal system between K and $S4$, then every problem computable in polynomial space is log space reducible to the provability problem in S. We then show that there are polynomial space bounded algorithms for deciding if a formula is provable in any one of K, T, and $S4$. This implies that the provability problem for each of systems K, T, and $S4$ is log space complete in polynomial space. We also obtain upper and lower bounds on the space complexity of the provability problem in each of the systems K, T, and $S4$.

We show that the nonprovability problem for $S5$ is log space complete in nondeterministic polynomial time. Hence the provability problem in $S5$ and the provability problem in the classical propositional calculus have the same complexity modulo polynomial time.

All our proofs depend heavily on the semantic models for modal systems developed by Kripke [6].

As evidence that modal logic has some applications in Computer Science, we point to the work of V. R. Pratt and R. Moore [8], who have developed a system of modal logic as a basis for proving correctness and termination of programs. We briefly explain their application. Assume we have some underlying programming language and some underlying assertion language. For each program p define a new syntactic object $[p]$ which is understood to be a modal operator. We can now form new assertions of the form $[p]A$ where A is an arbitrary assertion. The intuitive meaning of $[p]A$ is that "if p terminates, then A holds." The fact that p always terminates can be expressed by the assertion $\langle p\rangle$true (where $\langle p\rangle A \overset{\text{def}}{=} \sim[p]\sim A$). The Hoare formula $A\{p\}B$ is equivalent to the formula $A \supset [p]B$. An advantage of this modal system over the Hoare system is that more complicated assertions about programs are possible. For instance one can express the fact that "if a program p terminates with A true, then subsequently the

* Received by the editors August 21, 1975, and in revised form July 21, 1976.

† Department of Computer Science, University of Washington, Seattle, Washington 98195. This research was supported by the National Science Foundation under Grant GJ-43264.

program q can terminate with B being true" by the assertion "$[p](A \supset \langle q \rangle B)$".

If Σ is a finite alphabet, then define Σ^* to be the set of all finite words from letters in Σ and λ to be the empty word and $\Sigma^+ = \Sigma^* - \{\lambda\}$. If $x, y \in \Sigma^*$, then $|x|$ denotes the length of x, xy denotes x concatenated to y, and x^n denotes x concatenated to itself n times ($x^0 = \lambda$, $x^k = x \cdot x^{k-1}$ for $k \geqq 1$). Let $N = \{0, 1, 2, \cdots\}$. If $n \geqq 1$, then $\log n$ is defined to be $\lceil \log_2 n \rceil$ and $\log 0 = 0$.

1. Modal logic. We define formulas so that they are words in a finite alphabet. A *variable* is a member of VAR $= ¢\{0, 1\}^*\$$. A *Boolean formula* is either a variable or has the form $(A \wedge B)$ or $\sim A$ where A and B are Boolean formulas. The set of Boolean formulas, denoted by BF, is a subset of Δ_{BF}^* where $\Delta_{\mathrm{BF}} = \{¢, \$, 0, 1, \wedge, \sim, (,)\}$. A *modal formula* is either a variable or has the form $(A \wedge B)$, $\sim A$, or $\Box A$ where A and B are modal formulas. The set of modal formulas, denoted by MF, is a subset of Δ_{MF}^* where $\Delta_{\mathrm{MF}} = \Delta_{\mathrm{BF}} \cup \{\Box\}$. A formula of the form $\Box A$ is read "necessarily A". Another modal operator, $\Diamond$, is defined by $\Diamond A \equiv \sim\Box\sim A$ and $\Diamond$ can be read as "possibly". Technically $\Diamond$ and the standard logical operators $\wedge$, $\supset$, $\equiv$ do not appear in modal formulas, but for convenience they do appear in modal text. We also may drop parentheses from formulas to improve readability.

We will systematically use $\wedge$, $\vee$, $\sim$ as both logical symbols and as the Boolean operations on $\{T, F\}$ they represent.

Let PC (for propositional calculus) be some complete set of axioms for the valid Boolean formulas where the rules of inference are substitution and modus ponens. A *modal system* is a set of modal formulas. If S is a modal system, then define the provability relation, $\vdash_S$, inductively as follows.

(i) $\vdash_S A$ if $A \in \mathrm{PC} \cup S$,

(ii) $\vdash_S A'$ if $\vdash_S A$ and A' is the result of substituting uniformly in A a modal formula for a propositional variable (Rule of Substitution),

(iii) $\vdash_S B$ if $\vdash_S A$ and $\vdash_S A \supset B$ (Modus Ponens),

(iv) $\vdash_S \Box A$ if $\vdash_S A$ (Rule of Necessity),

(v) $\vdash_S$ is the smallest relation satisfying (i)–(iv).

If $\vdash_S A$, then we say that A is *provable in S* and we define S-PROVABLE $= \{A \in \mathrm{MF} : \vdash_S A\}$.

There are at least four important modal systems, K, T, $S4$, and $S5$ which are defined by

$$K = \{\Box(X \supset Y) \supset (\Box X \supset \Box Y)\},$$

$$T = K \cup \{\Box X \supset X\},$$

$$S4 = T \cup \{\Box X \supset \Box\Box X\},$$

$$S5 = S4 \cup \{\Diamond X \supset \Box\Diamond X\},$$

(X and Y are specific members of VAR.)

The reader unfamiliar with modal logic can appeal to Hughes and Cresswell [4].

Very useful semantic models for many modal systems were discovered by Kripke [6]. In particular, there are such semantics for the four systems K, T, $S4$, and $S5$. In the remainder of this section the facts we state are either due to Kripke [6] or are attributed to him.

A *model structure* is a triple (W, R, V) where W is a set, R is a binary relation on W, and V is a mapping from $\mathrm{VAR} \times W$ into $\{T, F\}$. The set W is a set of "possible worlds", R determines which worlds are "accessible" from other worlds, and V determines what is true in each of the worlds. Given a model structure (W, R, V) the mapping V can be extended to $\mathrm{MF} \times W$ inductively as follows:

$$V(A \wedge B, w) = T \quad \text{iff} \quad V(A, w) = T \text{ and } V(B, w) = T,$$

$$V({\sim}A, w) = T \quad \text{iff} \quad V(A, w) = F,$$

$$V(\Box A, w) = T \quad \text{iff} \quad \text{for all } w' \in W, \text{ if } wRw', \text{ then } V(A, w') = T.$$

Define (W, R, V) to be a *K-model* if it is a model structure and to be a (i) *T-model*, (*ii*) *$S4$-model*, (iii) *$S5$-model* if R is respectively (i) reflexive, (ii) reflexive and transitive, (iii) reflexive, transitive and symmetric.

Let $S \in \{K, T, S4, S5\}$. Define a modal formula A to be *S-satisfiable* if there is an S-model (W, R, V) and a world $w \in W$ such that $V(A, w) = T$. Let S-SATISFIABLE $= \{A \in \mathrm{MF}: A$ is S-satisfiable$\}$. Define A to be *S-valid* if ${\sim}A$ is not S-satisfiable. Let S-VALID $= \{A \in \mathrm{MF}: A$ is S-valid$\}$. The crucial fact we use later is:

FACT 1.1 (Kripke). *For all* $S \in \{K, T, S4, S5\}$ S-VALID $= S$-PROVABLE.

The *modal degree* of a formula is defined inductively: the modal degee of a variable is 0; degree of ${\sim}A$ = degree of A; degree of $A \wedge B = \max\{$degree of A, degree of $B\}$; and degree of $\Box A = 1 +$ degree of A.

2. Computational complexity. We adopt the *Turing machine* model of computation to measure time and space complexity. The reader may refer to Hopcroft and Ullman [3, Chap. 10] for background.

To be specific, our Turing machines will have three tapes: a two-way read-only input tape, one-way write-only output tape, and a two-way read-write work tape. Associated with such a machine are finite alphabets: Σ (input alphabet), Δ (output alphabet), and Γ (work tape alphabet); also a finite set of states Q, a start state q_0 and a transition function

$$\delta: Q \times \Sigma \times \Gamma \to 2^{Q \times \Gamma \times (\Delta \cup \{\lambda\}) \times \{R, L\}^2}.$$

Given a state, a symbol being read on the input tape, a symbol being read on the work tape, the machine does one of a finite number of "moves" each of which consists of going to a new state, writing a symbol on the work tape, outputting either a symbol or λ and moving the input tape and work tape heads. As defined, our Turing machines are *nondeterministic*. A Turing machine is *deterministic* if the cardinality of $\delta(q, \sigma, \tau) \leqq 1$ for each triple $(q, \sigma, \tau) \in Q \times \Sigma \times \Gamma$.

Given an input $x \in \Sigma^*$,a *computation of T on input x* is a finite sequence of configurations of the Turing machine which begins in the starting configuration (the machine is in state q_0, input tape contains x with the input head on the first letter of x, and the other tapes empty), each other configuration follows from the previous one via the transition rule, and ends in a configuration from which no configuration can follow. A Turing machine *T runs in time $t: N \to N$* if for each n and each $x \in \Sigma^*$ such that $|x| = n$ every computation of T on input x has length $\leqq t(n)$. A Turing machine *T runs in space s: $N \to N$* if for each n and each $x \in \Sigma^*$ of length n at most $s(n)$ distinct tape cells on the *work* tape are scanned in each computation of T on input x.

A *set* $L \subseteq \Sigma^*$ *is computable in nondeterministic time* (space) r if there is a Turing machine T that runs in time (space) r such that for all $x \in \Sigma^*$, $x \in L$ iff there is a computation of T on input x such that T outputs some symbol during that computation. A *set L is computable in time* (space) r if in the above definition the Turing machine is deterministic. A *function* $f: \Sigma^* \to \Delta^*$ *is computable in time* (space) r if there is a deterministic Turing machine T that runs in time (space) r such that for all $x \in \Sigma^*$, when T halts on input x the machine has outputted the string $f(x)$.

We define NP-TIME (NP-SPACE) to be the class of sets L such that there is a polynomial p such that L is computable in nondeterministic time (space) p. Similarly P-TIME (P-SPACE) is the class of sets L such that there is a polynomial p such that L is computable in time (space) p. A result of Savitch [9] implies P-SPACE $=NP$-SPACE. There is the obvious containment relationship P-TIME $\subseteq NP$-TIME $\subseteq P$-SPACE. It is open whether or not either containment is proper.

If $s: N \to N$, then define (N)SPACE($s(n)$) = the class of sets computable in (nondeterministic) space s. We don't define the analogous time complexity classes for the same reason that we don't bother with multiple work tapes; the methods we use cannot be used to distinguish polynomial time complexity up to the degree of the polynomial.

Given sets $L \in \Sigma^*$ and $M \in \Delta^*$ we say that *L is log space reducible to M* ($L \leqq_{\log} M$) if there is a function $f: \Sigma^* \to \Delta^*$ such that f is computable in space log and for all $x \in \Sigma^*$, $x \in L$ iff $f(x) \in M$. We sometimes say $L \leqq_{\log} M$ *via f*. The relation $\leqq_{\log}$ is reflexive and transitive (cf. Stockmeyer and Meyer [12] or Jones [5]).

Let $\mathcal{S}$ be a class of sets. *A* set *L is log space complete in* $\mathcal{S}$ if $L \in \mathcal{S}$ and for all $M \in \mathcal{S}$, $M \leqq_{\log} L$. Cook [2] implicitly showed the existence of log space complete sets in NP-TIME while Stockmeyer and Meyer [12] showed the existence of log space complete sets in P-SPACE.

There is a well known relationship between complete problems and open problems concerning P-TIME, NP-TIME, and P-SPACE.

FACT 2.1. *If L is log space complete in* NP-TIME,*then* $L \in P$-TIME *if and only if* P-TIME $= NP$-TIME.

FACT 2.2. *If L is log space complete in* P-SPACE, *then*

(i) $L \in P$-TIME *if and only if* P-TIME $= P$-SPACE,

(ii) $L \in NP$-TIME *if and only if* NP-TIME $= P$-SPACE.

If $l: N \to N$ and $f: \Sigma^* \to \Delta^*$ then f is *length* $l(n)$ *bounded* if for all $x \in \Sigma^*$, $|f(x)| \leqq l(|x|)$. The following fact due to Stockmeyer and Meyer [12] and Jones [5] is helpful later in establishing lower bounds.

FACT 2.3 (Stockmeyer and Meyer, and Jones). *If* $A \leqq_{\log} B$ *via f where f is length* $l(n)$ *bounded, then A is in* (N)SPACE($s(l(n)) + \log n$) *should B be in* (N)SPACE($s(n)$).

Let $\Delta_{\mathrm{QBF}} = \Delta_{\mathrm{BF}} \cup \{\forall, \exists\}$. A *quantified Boolean formula* (QBF) is a member of Δ^*_{QBF} of the form $Q_1X_1Q_2X_2 \cdots Q_nX_nA(X_1, \cdots, X_n)$ where $Q_i \in \{\forall, \exists\}$, $X_i \in$ VAR for $1 \leqq i \leqq n$ and $A(X_1, \cdots, X_n) \in$ BF whose variables are contained in

$X_1, \cdots, X_n$. The propositional variables range over $\{T, F\}$ so that if $A \in \mathrm{QBF}$, then the value of A is either T or F.

Define

$$\mathbf{B}_\omega = \{A \in \mathrm{QBF} : A \equiv T\},$$

$$\mathbf{B}_1 = \{A \in \mathrm{QBF} \cap (\exists\mathrm{VAR})^*\mathrm{BF} : A \equiv T\}.$$

The set $\mathbf{B}_\omega$ is the set of all valid quantified Boolean formulas, while $\mathbf{B}_1$ is essentially the set of all satisfiable Boolean formulas.

Stockmeyer and Meyer [12] have shown

FACT 2.4 (Stockmeyer and Meyer). *$\mathbf{B}_\omega$ is log space complete in P*-SPACE.

A more precise delineation of $\mathbf{B}_\omega$ is given in Stockmeyer [11].

FACT 2.5 (Stockmeyer). *Let d be an integer* $\geqq 1$. *If $A \in \mathrm{NSPACE}(n^d)$, then there is a function f and a constant $a > 0$ such that $A \leqq_{\log} \mathbf{B}_\omega$ via f and f is length an^{2d}* $\log$ *n bounded.*

When we investigate the complexity of $S5$ we will need a result of Cook [2].

FACT 2.6 (Cook). *$\mathbf{B}_1$ is log space complete in NP*-TIME.

Because of the transitivity of $\leqq_{\log}$ we can show that every problem computable in polynomial space is log space reducible to, say, L if we can show that $\mathbf{B}_\omega$ is log space reducible to L. In what follows we use $\mathbf{B}_\omega$ as a cornerstone in analyzing the space complexity of modal systems between K and $S4$.

One useful fact that we use later is:

FACT 2.7. *If $\mathbf{B}_\omega \leqq_{\log} A$ via a length $l(n)$ bounded function, then $\mathbf{B}_\omega \leqq_{\log} \bar{A}$ via a length $l(n+5)$ bounded function.*

Proof Let f be such that $\mathbf{B}_\omega \leqq_{\log} A$ via f and f is length $l(n)$ bounded. There is a g such that $\mathbf{B}_\omega \leqq_{\log} \bar{\mathbf{B}}_\omega$ via g and g is length $n+5$ bounded. Let $x \in \Delta^*_{\mathrm{QBF}}$ and let $n = |x|$. It can be determined in space $\log n$ whether or not $x \in \mathrm{QBF}$. If $x \notin \mathrm{QBF}$, then define $g(x) = (\exists \cancel{c}\$)\cancel{c}\$$. If $x \in \mathrm{QBF}$ then define $g(x) = \bar{Q}_1X_1 \cdots \bar{Q}_mX_m \sim A$ where $x = Q_1X_1 \cdots Q_mX_mA$, $A \in \mathrm{BF}$, $\bar{\forall} = \exists$ and $\bar{\exists} = \forall$. Clearly, $x \in \mathbf{B}_\omega$ if and only if $g(x) \notin \mathbf{B}_\omega$. Now, $\mathbf{B}_\omega$ is log space reducible via $f \cdot g$ which is length $l(n+5)$ bounded. Q.E.D.

3. Log space reduction of $\mathbf{B}_\omega$ to modal systems between *K* and *S*4. We say that a modal system S *is between* S_1 *and* S_2 if S_1-PROVABLE $\subseteq S$-PROVABLE $\subseteq S_2$-PROVABLE. In this section we prove the following.

THEOREM 3.1. *If S is between K and $S4$, then $\mathbf{B}_\omega$ is log space reducible to* S-PROVABLE.

Proof. The crux of the proof is to show that given any quantified Boolean formula A, a modal formula B can be constructed (using only logarithmic space) with the properties: (i) $A \in \mathbf{B}_\omega$ implies $B \in S4$-SATISFIABLE and (ii) $B \in K$-SATISFIABLE implies $A \in \mathbf{B}_\omega$.

In light of Fact 2.7, the following claim yields the theorem.

CLAIM. $A \in \mathbf{B}_\omega$ if and only if $\sim B \notin S$-PROVABLE.

If $A \in \mathbf{B}_\omega$, then by (i) $B \in S4$-SATISFIABLE and hence $\sim\sim B \in S4$-SATISFIABLE. By the definition of $S4$-VALID, $\sim B \notin S4$-VALID. By Fact 1.1, $\sim B \notin S4$-PROVABLE. Since S-PROVABLE $\subseteq S4$-PROVABLE, then $\sim B \notin S$-PROVABLE. On the other hand if $\sim B \notin S$-PROVABLE, then because K-PROVABLE $\subseteq S$-PROVABLE, then $\sim B \notin K$-PROVABLE. Again using Fact 1.1, $B \in K$-SATISFIABLE, which in turn implies by (ii) that $A \in \mathbf{B}_\omega$.

Let $A = Q_1X_1 \cdots Q_mX_mA'(X_1, \cdots, X_m) \in \mathrm{QBF}$ where $Q_i \in \{\forall, \exists\}$ and $X_i \in \mathrm{VAR}$ for $1 \leqq i \leqq m$, and $A'(X_1, \cdots, X_m) \in \mathrm{BF}$. Let $Y_0, \cdots, Y_m$, and $Z_1, \cdots, Z_{1+\log m}$ be new variables an for $0 \leqq i \leqq m$and $1 \leqq j \leqq 1+\log m$ define $\beta_{ij} = \sim$ if the jth bit of i written as a binary number of length $1+\log m$ is 1 and $\beta_{ij} = \lambda$ otherwise.

Define B to be the conjunction of the following formulas:

$$\Box^{(m)}(Y_i \equiv \beta_{i1}Z_i \wedge \beta_{i2}Z_2 \wedge \cdots \wedge \beta_{i(1+\log m)}Z_{1+\log m}) \quad \text{for } 0 \leqq i \leqq m, \tag{1}$$

$$Y_0, \tag{2}$$

$$\Box^{(m)}(Y_i \supset \Diamond Y_{i+1}) \quad \text{for } 0 \leqq i < m, \tag{3}$$

$$\Box^{(m)}(Y_i \supset ((X_i \supset \Box^{(m)}X_i) \wedge (\sim X_i \supset \Box^{(m)} \sim X_i))) \quad \text{for } 0 < i \leqq m, \tag{4}$$

$$\Box^{(m)}(Y_i \supset (\Diamond(Y_{i+1} \wedge X_{i+1}) \wedge \Diamond(Y_{i+1} \wedge \sim X_{i+1}))) \quad \text{if } Q_{i+1} = \forall \text{ and } 0 \leqq i < m, \tag{5}$$

$$\Box^{(m)}(Y_m \supset A'), \tag{6}$$

where $\Box^{(m)}D \equiv D \wedge \Box D \wedge \Box^2 D \wedge \cdots \wedge \Box^m D$.

The intuitive meaning of $\Box^{(m)}D$ is that in any model structure (W, R, V), $V(\Box^{(m)}D, w) = T$ if and only if D is true in any world reachable from w in i steps where $0 \leqq i \leqq m$.

The idea behind the formula B is to "simulate" the quantifiers of A. The variables Y_i are used to set up levels corresponding to the levels of quantification in A. The formula Y_i is true in each world on level i. If the ith quantifier of A is universal, then (5) guarantees a splitting for each of the two possibilities for X_i. At the final level, m, A' must be true. We begin by showing (i) mentioned earlier.

$A \in \mathbf{B}_\omega$ implies $B \in S4$-SATISFIABLE. Suppose $A \in \mathbf{B}_\omega$; then B is satisfied in the $S4$-model, (W_A, R_A, V_A). The set of worlds is a finite subset of $\{0, 1\}^*$ defined inductively by

(a) $\lambda \in W_A$,

(b) if $w \in W_A$ and $|w| = i < m$, then

(b1) $w0$ and $w1 \in W_A$ if and only if $Q_{i+1} = \forall$,

(b2) $w0 \in W_A$ if and only if $Q_{i+1} = \exists$,

(c) W_A is the smallest set satisfying (a) and (b).

The members of W_A form a tree with respect to extension. The tree is binary branching at level i if $Q_{i+1} = \forall$ and is unary branching if $Q_{i+1} = \exists$. The accessibility relation R_A is defined by

$$x\, R_A\, y \quad \text{iff} \quad x \text{ is a prefix of } y.$$

Clearly R_A is reflexive and transitive. Finally we define V_A inductively on the length of w in such a way that

(a′) if $|w| = i$, then $V_A(Y_i, w) = T$,

(b′) if $|w0| = |w1| = i$ and $Q_i = \forall$, then $V_A(X_i, w0) \neq V_A(X_i, w1)$,

(c′) if $|w| = i > j$, then $V_A(X_j, w) = V_A(X_j, w')$ where w' is the prefix of w of length $i-1$,

(d′) if $|w| = i$, then $Q_{i+1}X_{i+1} \cdots Q_mX_mA'(V_A(X_1, w), \cdots, V_A(X_i, w), X_{i+1}, \cdots, X_m) = T$.

Assume (a′)–(d′) hold for all numbers $<i$. Let $|w|=i$. Set $V_A(Z_j, w)=T$ if $\beta_{ij}=\lambda$, $V_A(Z_j, w)=F$ if $\beta_{ij}=\sim$, $V_A(Y_j, w)=F$ if $j\neq i$, and $V_A(Y_i, w)=T$. If $1\leqq j<i$, then set $V_A(X_j, w)=V_A(X_j, w')$ where w' is the prefix of w of length $i-1$. If $j>i$, then set $V_A(X_j, w)=T$.

If $i=0$, then (a′)–(c′) hold by definition and (d′) holds because $A\in\mathbf{B}_\omega$. Assume then that $i>0$; then all that remains is the value of $V_A(X_i, w)$.

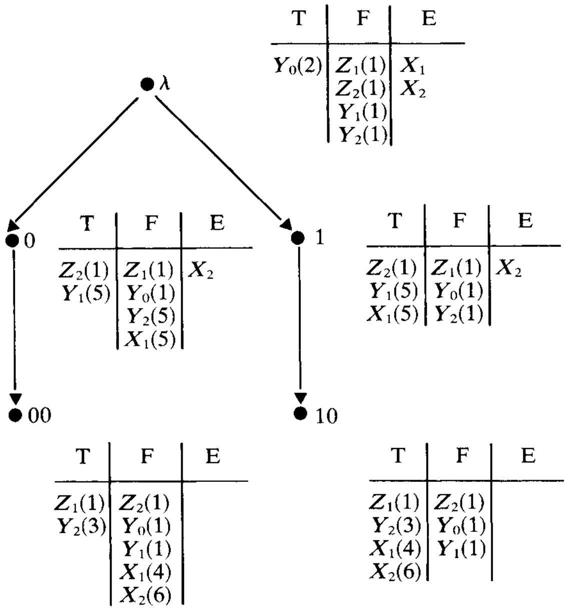
If $Q_i=\forall$, then set $V_A(X_i, w)=T$ if the last letter of w is 1 and set $V_A(X_i, w)=F$ otherwise.

If $Q_i=\exists$, then set $V_A(X_i, w)=V$ where $Q_{i+1}X_{i+1}\cdots Q_mX_mA'(V_A(X_1, w), \cdots, V_A(X_{i-1}, w), V, X_{i+1}, \cdots, X_m)=T$. Such a $V\in\{T, F\}$ exists because of the induction hypothesis.

It is straightforward to check that the induction hypothesis holds at i.

To establish that $B\in S4$-SATISFIABLE, we show that $V_A(B, \lambda)=T$. Clauses (1) through (5) in the definition of B hold by the construction of V_A. Clause (6) holds because by (a′) if $|w|<m$, then $V_A(Y_m, w)=F$ and by (d′) if $|w|=m$, then $V_A(A', w)=T$.

As an example of the preceding proof consider the formula $\forall X_1\exists X_2(X_1\equiv X_2)$. This formula is true and its modal companion B is satisfied in the $S4$-model graphically displayed in Fig. 1.



T	F	E
$Y_0(2)$	$Z_1(1)$	X_1
	$Z_2(1)$	X_2
	$Y_1(1)$	
	$Y_2(1)$	

T	F	E
$Z_2(1)$	$Z_1(1)$	X_2
$Y_1(5)$	$Y_0(1)$	
	$Y_2(5)$	
	$X_1(5)$	

T	F	E
$Z_2(1)$	$Z_1(1)$	X_2
$Y_1(5)$	$Y_0(1)$	
$X_1(5)$	$Y_2(1)$	

T	F	E
$Z_1(1)$	$Z_2(1)$	
$Y_2(3)$	$Y_0(1)$	
	$Y_1(1)$	
	$X_1(4)$	
	$X_2(6)$	

T	F	E
$Z_1(1)$	$Z_2(1)$	
$Y_2(3)$	$Y_0(1)$	
$X_1(4)$	$Y_1(1)$	
$X_2(6)$		

T—*variables that must be true.*
F—*variables that must be false.*
E—*variables that can have either value.*
The number in parentheses indicates the clause of B that forces the value of the variable.
The arrows represent the Hasse diagram of the accessibility relation.

FIG 1. *$S4$-model satisfying B associated with* $\forall X_1\exists X_2(X_1\equiv X_2)$

$B \in K$-SATISFIABLE *implies* $A \in \mathbf{B}_\omega$. Suppose that B is K-satisfiable in a model structure (W, R, V). We define a mapping, σ, of W_A into W inductively as follows.

(a″) Choose $\sigma(\lambda)$ such that $V(B, \sigma(\lambda)) = T$,

(b″) if $|w| = i > 0$ then choose $\sigma(w) \in W$ such that $\sigma(w')\, R\, \sigma(w)$ where w' is the prefix of w of length $i-1$ and

1. $V(\Box^{(m-i)}(Y_j \equiv \beta_{j1}Z_1 \wedge \cdots \wedge \beta_{j(1+\log m)}Z_{1+\log m}),\ \sigma(w)) = T$ for $0 \leqq j \leqq m$ (by clause (1) of B),
2. $V(Y_i, \sigma(w)) = T$ (by clause (3) of B),
3. $V(\Box^{(m-i)}(Y_j \supset \Diamond Y_{j+1}), \sigma(w)) = T$ for $0 \leqq j < m$ (by clause (3) of B),
4. $V(\Box^{(m-i)}(Y_j \supset ((X_j \supset \Box^{(m)}X_j) \wedge (\sim X_j \supset \Box^{(m)} \sim X_j))),\ \sigma(w)) = T$ for $0 < j \leqq m$ (by clause (4) of B),
5. $V(\Box^{(m-i)}(Y_j \supset (\Diamond(Y_{j+1} \wedge X_{j+1}) \wedge \Diamond(Y_{j+1} \wedge \sim X_{j+1}))),\ \sigma(w)) = T$ if $Q_{j+1} = \forall$ and $0 \leqq j < m$ (by clause (5) of B),
6. $V(\Box^{(m-i)}(Y_m \supset A'), \sigma(w)) = T$ (by clause (6) of B),
7. either $V(\Box^{(m-i)}X_j, \sigma(w)) = T$ or $V(\Box^{(m-i)} \sim X_j, \sigma(w)) = T$ for $j \leqq i$ (by the induction hypothesis for $j < i$ and by 2 and 4 above for $j = i$),
8. $V(X_j, \sigma(w)) = V(X_j, \sigma(w'))$ if $j < i$ and w' is the prefix of w of length $i-1$ (by 7 above),
9. $V(X_i, \sigma(w)) = T$ if $Q_i = \forall$ and w ends in 1 (by 2 and 5 above),
10. $V(X_i, \sigma(w)) = F$ if $Q_i = \forall$ and w ends in 0 (by 2 and 5 above).

We leave it to the reader to convince himself that such a mapping exists because B is K-satisfiable.

We may show by induction on $m - i$ that if $|w| = i$, then

$$Q_{i+1}X_{i+1} \cdots Q_mX_mA'(V(X_1, \sigma(w)), \cdots, V(X_i, \sigma(w)), X_{i+1}, \cdots, X_m) = T.$$

If $|w| = m$ then $V(Y_m, \sigma(w)) = T$ and $V(Y_m \supset A', \sigma(w)) = T$. Thus $V(A', \sigma(w)) = T$, which implies the equality for $m - i = 0$.

Let $|w| = i - 1$. It is straightforward to show that 8, 9, 10 above and the induction hypothesis imply the equality for w.

It remains to be shown that B can be constructed in logarithmic space given A. Technically speaking, we should be considering a mapping from Δ^*_{QBF} to Δ^*_{MF}, but it takes only space $\log n$ to check that a member of Δ^*_{QBF} is a member of QBF, so that we can essentially ignore non-well formed formulas. The ability to count the number of quantifiers in A is really all that is necessary in order to construct B. This amounts to a $\log n$ space bound. We leave the details to the reader. Q.E.D.

We originally just showed that $\mathbf{B}_\omega$ was log space reducible to each of T and $S4$. Subsequently S. K. Thomason showed us how to extend the proof to obtain the result for all systems between K and $S4$.

4. Space lower bounds for provability in *K*, *T*, and *S*4. We begin by trying to find the most efficient log space reductions of $\mathbf{B}_\omega$ to each of K, T, and $S4$.

LEMMA 4.1. *For each $S \in \{K, T, S4\}$ there is a function f_S such that $\mathbf{B}_\omega \leqq_{\log} S$-*PROVABLE *via f_S where f_S is length $l(n)$ bounded and*

(i) *$S = K$ implies $l(n) = O(n^3/\log^2 n)$,*

(ii) *$S = T$ implies $l(n) = O(n^2/\log^2 n)$,*

(iii) *$S = S4$ implies $l(n) = O(n \log n)$.*

Proof. Let $A = Q_1X_1 \cdots Q_mX_mA'$ where $A' \in \mathrm{BF}$ and let $n = |A|$. Without loss of generality we can assume that $X_i = ¢\#i\$$ where $\#i$ is the ith binary string in the ordering $\lambda, 0, 1, 00, 01, 10, 11, 000, \cdots$. It is important to notice that $|X_i| \leqq 2 + \log i$. If the Z_i's and Y_i's are chosen as follows, $Y_i = ¢\#(m+i)\$$ for $1 \leqq i \leqq m$, and $Z_i = ¢\#(2m+i)\$$ where $1 \leqq i \leqq 1 + \log m$ then $|Y_i| \leqq 1 + \log m$ and $|Z_i| \leqq 2 + \log m$. Note that $m = O(n/\log n)$.

Technically speaking, in Theorem 3.1 we reduced $\mathbf{B}_\omega$ to the complement of S-PROVABLE. By Fact 2.7 there is no loss (except for constant factors) in using the length bound of the reduction of $\mathbf{B}_\omega$ to the complement of S-PROVABLE as the length bound of the reduction of $\mathbf{B}_\omega$ to S-PROVABLE itself.

Case (i). $S = K$. To begin with we more efficiently encode $\Box^{(m)}D$ as $D \wedge \Box(D \wedge \Box(D \wedge \cdots (D \wedge \Box D)) \cdots)$ so that $|\Box^{(m)}D| = O(m|D|)$. Another improvement is to factor $\Box^{(m)}$ out using the rule $\Box^{(m)}(C \wedge D) \equiv \Box^{(m)}C \wedge \Box^{(m)}D$. Notice also that $|Y_i \equiv \beta_{i1}Z_1 \wedge \cdots \wedge \beta_{i(1+\log m)}Z_{1+\log m}| = O(\log^2 m)$. From this we can see that (4)· and (6) dominate the length of B with lengths $O(m^3 \log m)$ and $O(mn)$ respectively. We have that $|B|$ is $O(n^3/\log^2 n)$,

Case (ii). $S = T$. We may replace $\Box^{(m)}$ with just $\Box^m$. Again (4) and (6) dominate with lengths $O(m^2)$ and $O(n)$ respectively. Hence $|B|$ is $O(n^2/\log n)$.

Case (iii). We replace $\Box^{(m)}$ with simply $\Box$. In this case (1) and (6) dominate with lengths $O(m \log m)$ and $O(n)$ respectively. Hence $|B|$ is $O(n \log n)$. Q.E.D.

In the spirit of Stockmeyer [11] we use the lemma to show lower bounds on the space complexity of provability in K, T, and $S4$.

THEOREM 4.2. *If* $S \in \{K, T, S4\}$ *and* S-PROVABLE $\in$ NSPACE$(s(n))$, *then there is a constant* $c > 0$ *such that*

(i) $S = K$ *implies* $s(n) > c(n/\log n)^{1/6}$ *for infinitely many* n,

(ii) $S = T$ *implies* $s(n) > cn^{1/4}$ *for infinitely many* n,

(iii) $S = S4$ *implies* $s(n) > c(n/\log^2 n)^{1/2}$ *for infinitely many* n.

Proof. We begin with a proof of (i) which parallels almost exactly a proof of Stockmeyer [11, Cor. 6.6]. Suppose to the contrary that K-PROVABLE $\in$ NSPACE$(s(n))$ where for all $c > 0$, $s(n) \leqq c(n/\log^4 n)^{1/6}$ for all but finitely many n. We may assume that s is a nondecreasing function.

By the hierarchy theorem of Seiferas, Fischer and Meyer [10], we can conclude that there is a set $A \in$ NSPACE(n) such that for all s' if $\lim_n (s'(n+1)/n) = 0$, then $A \notin$ NSPACE$(s'(n))$. By Fact 2.5 $A \leqq_{\log} \mathbf{B}_\omega$ via a length $O(n^2 \log n)$ bounded function. By Lemma 4.1, $\mathbf{B}_\omega \leqq_{\log} K$-PROVABLE via a length $O(n^3/\log^2 n)$ bounded function. Hence $A \leqq_{\log} K$-PROVABLE via a length $O(n^6 \log n)$ bounded function. By Fact 2.3, $A \in$ NSPACE$(s(an^6 \log n) + \log n)$ for some constant $a > 0$. For all $c > 0$ $s(an^6 \log n) \leqq cn$ for all but finitely many n, contradicting the fact that $A \notin$ NSPACE$(s'(n))$ if $\lim_n (s'(n+1)/n) = 0$.

The proofs of (ii) and (iii) are analogous if we use the facts that if $A \in$ NSPACE(n), then $A \leqq_{\log} T$-PROVABLE via a length $O(n^4)$ bounded function and $A \leqq_{\log} S4$-PROVABLE via a length $O(n^2 \log^2 n)$ bounded function. Q.E.D.

5. Space upper bounds for provability in *K*, *T*, and *S*4. In this section we show that for $S \in \{K, T, S4\}$, S-PROVABLE $\in P$-SPACE. In essence we actually show that S-SATISFIABLE $\in P$-SPACE. This may be surprising to some since there are modal formulas A_n of length $O(n \log(n) \operatorname{loglog}(n))$ with the property that $A_n \in S4$-SATISFIABLE and if (W, R, V) is an $S4$-model of A_n, then the cardinality of W is $\geqq 2^n$. What allows us to compute S-SATISFIABLE in polynomial space is the fact that if A is S-SATISFIABLE, then it is in a tree-like model structure, with each branch of only polynomial length. Hence the structure can be constructed one branch at a time.

THEOREM 5.1. *For* $S \in \{K, T, S4\}$, S-PROVABLE $\in P$-SPACE.

Proof. The algorithms that we will give are simply reformulations of the corresponding algorithms of Kripke [6] in such a way to optimize the space used. We do not necessarily give the most efficient algorithms, because we wish to present algorithms that are both understandable and run in polynomial space.

We begin with a procedure K-WORLD which has parameters $(\mathcal{T}, \mathcal{F}, \tilde{\mathcal{T}}, \tilde{\mathcal{F}})$, where each parameter is a finite set of modal formulas; the value of K-WORLD $(\mathcal{T}, \mathcal{F}, \tilde{\mathcal{T}}, \tilde{\mathcal{F}})$ is **true** if there is a K-model (W, R, V) and a $w \in W$ such that

$$V\Big(\bigwedge_{A \in \mathcal{T}} A \wedge \bigwedge_{A \in \mathcal{F}} \sim A \wedge \bigwedge_{A \in \tilde{\mathcal{T}}} \Box A \wedge \bigwedge_{A \in \tilde{\mathcal{F}}} \sim \Box A, w\Big) = T,$$

otherwise its value is **false.** More intuitively, K-WORLD$(\mathcal{T}, \mathcal{F}, \tilde{\mathcal{T}}, \tilde{\mathcal{F}})$ is **true** if there is a world w in which all the formulas of $\mathcal{T}$ are true, all the formulas of $\mathcal{F}$ are false, in each world accessible from w each member of $\tilde{\mathcal{T}}$ is true, and for each member, B, of $\tilde{\mathcal{F}}$ there is a world accessible from w where B is false.

```
procedure K-WORLD($\mathcal{T}, \mathcal{F}, \tilde{\mathcal{T}}, \tilde{\mathcal{F}}$):
begin
  if $\mathcal{T} \cup \mathcal{F} \not\subseteq$ VAR then
    begin
1.    choose $A \notin \mathcal{T} \cup \mathcal{F} -$ VAR;
2.    if $A = \sim B$ and $A \in \mathcal{T}$ then return K-WORLD($\mathcal{T} - \{A\}, \mathcal{F} \cup \{B\}, \tilde{\mathcal{T}}, \tilde{\mathcal{F}}$);
3.    if $A = \sim B$ and $A \in \mathcal{F}$ then return K-WORLD($\mathcal{T} \cup \{B\}, \mathcal{F} - \{A\}, \tilde{\mathcal{T}}, \tilde{\mathcal{F}}$);
4.    if $A = B \wedge C$ and $A \in \mathcal{T}$ then return K-WORLD($(\mathcal{T} \cup \{B, C\}) - \{A\}, \mathcal{F},$
        $\tilde{\mathcal{T}}, \tilde{\mathcal{F}}$);
5.    if $A = B \wedge C$ and $A \in \mathcal{F}$ then return K-WORLD($\mathcal{T}, (\mathcal{F} \cup \{B\}) - \{A\}, \tilde{\mathcal{T}},$
        $\tilde{\mathcal{F}}$) $\vee$ K-WORLD($\mathcal{T}, (\mathcal{F} \cup \{C\}) - \{A\}, \tilde{\mathcal{T}}, \tilde{\mathcal{F}}$);
6.    if $A = \Box B$ and $A \in \mathcal{T}$ then return K-WORLD($\mathcal{T} - \{A\}, \mathcal{F}, \tilde{\mathcal{T}} \cup \{B\}, \tilde{\mathcal{F}}$);
7.    if $A = \Box B$ and $A \in \mathcal{F}$ then return K-WORLD($T, F - \{A\}, \tilde{\mathcal{T}}, \tilde{\mathcal{F}} \cup \{B\}$)
    end;
  if $\mathcal{T} \cup \mathcal{F} \subseteq$ VAR then
    begin
8.    if $\mathcal{T} \cap \mathcal{F} \neq \varnothing$ then return false;
9.    if $\mathcal{T} \cap \mathcal{F} = \varnothing$ then return $\bigwedge_{B \in \tilde{\mathcal{F}}}$ K-WORLD($\tilde{\mathcal{T}}, \{B\}, \varnothing, \varnothing$)
    end
end
```

(*Note.* The conjunction over the empty set is defined to be **true**.)

On line 1 we say "choose $A \in \mathcal{T} \cup \mathcal{F} -$ VAR". We do not intend this as a nondeterministic step; it is just that it does not matter in what specific order the lists $\mathcal{T}$ and $\mathcal{F}$ are maintained.

The proof that K-WORLD works is essentially the same as that for Kripke's corresponding algorithm [6]. We can now give an algorithm for testing whether or not a modal formula $A \in K$-PROVABLE.

Test for $A \in K$-PROVABLE.

```
begin
  read $A$;
  $v \leftarrow \sim K$-WORLD($\{\sim A\}, \emptyset, \emptyset, \emptyset$);
end
```

The value of v determines if A is K-provable. This of course exploits the fact that $A \in K$-PROVABLE if and only if $\sim A \notin K$-SATISFIABLE.

We now examine the space complexity of this algorithm. The recursive nature of the algorithm is implemented on a Turing machine by simulating a stack. At each level of recursion the members of $\mathcal{T}$, $\mathcal{F}$, $\tilde{\mathcal{T}}$, $\tilde{\mathcal{F}}$ are just sets of subformulas of $\sim A$ so that their values an be indicated by using "pointers" to $\sim A$.

To implement the pointers, copy the original formula onto the stack and place a mark on the major connective of each subformula pointed to. There are four types of marks, one for each of the four subsets. The storage at each level of recursion is $O(n)$. We will also show that the numbers of levels of recursion is $O(n)$ so that the total space used is $O(n^2)$.

If $\mathcal{S}$ is a finite set of formulas then define $|\mathcal{S}| = \sum_{A \in \mathcal{S}} |A|$. We show by induction on $n = |\mathcal{T}| + |\mathcal{F}| + |\tilde{\mathcal{T}}| + |\tilde{\mathcal{F}}|$ that K-WORLD($\mathcal{T}, \mathcal{F}, \tilde{\mathcal{T}}, \tilde{\mathcal{F}}$) has at most $2n+1$ levels of recursion. Assume the result for all numbers $<n$. Let the first recursive call of K-WORLD($\mathcal{T}, \mathcal{F}, \tilde{\mathcal{T}}, \tilde{\mathcal{F}}$) be to ($\mathcal{T}', \mathcal{F}', \tilde{\mathcal{T}}', \tilde{\mathcal{F}}'$). If $\mathcal{T} \cup \mathcal{F} \neq \emptyset$, then by a case-by-case analysis $|\mathcal{T}'| + |\mathcal{F}'| + |\tilde{\mathcal{T}}'| + |\tilde{\mathcal{F}}'| < n$. If $\mathcal{T} \cup \mathcal{F} = \emptyset$, then we must be at line 9 of the program, so that $\tilde{\mathcal{F}}' \neq \emptyset$, which reduces us to the case $\mathcal{T}' \cup \mathcal{F}' \neq \emptyset$. Hence every two levels of recursion reduces $|\mathcal{T}| + |\mathcal{F}| + |\tilde{\mathcal{T}}| + |\tilde{\mathcal{F}}|$ by at least 1. Thus K-WORLD($\{\sim A\}, \emptyset, \emptyset, \emptyset$) has recursion depth $\leqq 2|A| + 1$.

We now argue that T-PROVABLE $\in$ SPACE(n^3). Only slight modifications of the procedure K-WORLD are necessary to produce the analogous procedure T-WORLD.

(T-1) Replace all K's with T's.

(T-2) Replace line 6 with
"**if** $A = \Box B$ and $A \in \mathcal{T}$ **then return** T-WORLD($(T \cup \{B\}) - \{A\}$, $\mathcal{F}$, $\tilde{\mathcal{T}} \cup \{B\}$, $\tilde{\mathcal{F}}$)."

If $\mathcal{S}$ is a set of modal formulas then define deg($\mathcal{S}$) = max{modal degree of $C : C \in \mathcal{S}$}. In this case the storage at each level of recursion remains $O(n)$ but the recursion depth is $O(n^2)$. This can be seen by noticing that there can be at most $O(n)$ successive recursive calls all with $\mathcal{T} \cup \mathcal{F} \not\subseteq$ VAR; and if $\tilde{\mathcal{T}}'$, $\tilde{\mathcal{F}}'$ and $\tilde{\mathcal{T}}''$, $\tilde{\mathcal{F}}''$ are the values of $\tilde{\mathcal{T}}$ and $\tilde{\mathcal{F}}$ on successive calls with $\mathcal{T} \cup \mathcal{F} \subseteq$ VAR, then deg($\tilde{\mathcal{T}}'' \cup \tilde{\mathcal{F}}''$) < deg($\tilde{\mathcal{T}}' \cup \tilde{\mathcal{F}}'$). Since the degree of any set of subformulas of $\sim A$ is $\leqq n$, then there can be at most $O(n^2)$ levels of recursion. The total space is $O(n^3)$.

More elaborate changes to T-WORLD are necessary to obtain an analogous procedure $S4$-WORLD. We need the ability to check if the current world is exactly the same as a prior world. To do this we introduce a new parameter $\mathcal{L}$ which is a sequence $\{(\mathcal{T}_1, B_1), (\mathcal{T}_2, B_2), \cdots, (\mathcal{T}_k, B_k)\}$ where $\mathcal{T}_1 \subseteq \mathcal{T}_2 \subseteq \cdots \subseteq \mathcal{T}_k$ are sets of modal formulas and $B_1, \cdots, B_k$ are modal formulas. The value of

$S4$-WORLD$(\mathscr{T}, \mathscr{F}, \tilde{\mathscr{T}}, \tilde{\mathscr{F}}, \mathscr{L})$ is **true** if there is an $S4$-model (W, R, V) and a sequence of words $w_1, \cdots, w_k, w$ in W with the properties: (a) w_{i+1} is accessible from w_i and w is accessible from w_k; (b) $V(\bigwedge_{A\in\mathscr{T}_i} A \wedge \sim B_i, w_i) = T$ for each i; and (c) $V(\bigwedge_{A\in\mathscr{T}} A \wedge \bigwedge_{A\in\mathscr{F}} \sim A \wedge \bigwedge_{A\in\tilde{\mathscr{T}}} \Box A \wedge \bigwedge_{A\in\tilde{\mathscr{F}}} \sim\Box A, w) = T$.

For clarity we give the entire algorithm for $S4$-WORLD. The major changes are in lines 6 and 9.

```
 procedure S4-WORLD($\mathscr{T}, \mathscr{F}, \tilde{\mathscr{T}}, \tilde{\mathscr{F}}, \mathscr{L}$);
 begin
   if $\mathscr{T} \cup \mathscr{F} \not\subseteq$ VAR then
     begin
1.     choose $A \in \mathscr{T} \cup \mathscr{F} -$ VAR;
2.     if $A = \sim B$ and $A \in \mathscr{T}$ then return S4-WORLD($\mathscr{T} - \{A\}, \mathscr{F} \cup \{B\}, \tilde{\mathscr{T}}, \tilde{\mathscr{F}},$
         $\mathscr{L}$);
3.     if $A = \sim B$ and $A \in \mathscr{F}$ then return S4-WORLD($\mathscr{T} \cup \{B\}, \mathscr{F} - \{A\}, \tilde{\mathscr{T}}, \tilde{\mathscr{F}},$
         $\mathscr{L}$);
4.     if $A = B \wedge C$ and $A \in \mathscr{T}$ then return S4-WORLD($(\mathscr{T} \cup \{B, C\}) - \{A\}, \mathscr{F},$
         $\tilde{\mathscr{T}}, \tilde{\mathscr{F}}, \mathscr{L}$);
5.     if $A = B \wedge C$ and $A \in \mathscr{F}$ then return S4-WORLD($\mathscr{T}, (\mathscr{F} \cup \{B\}) - \{A\}, \tilde{\mathscr{T}},$
         $\tilde{\mathscr{F}}, \mathscr{L}$) $\vee$ S4-WORLD($\mathscr{T}, (\mathscr{F} \cup \{C\}) - \{A\}, \tilde{\mathscr{T}}, \tilde{\mathscr{F}}, \mathscr{L}$);
6.     if $A = \Box B$ and $A \in \mathscr{T}$ then return S4-WORLD($(\mathscr{T} \cup \{B\}) - \{A\}, \mathscr{F},$
         $\tilde{\mathscr{T}} \cup \{B\}, \tilde{\mathscr{F}}, \mathscr{L}$);
7.     if $A = \Box B$ and $A \in \mathscr{F}$ then return S4-WORLD($\mathscr{T}, \mathscr{F} - \{A\}, \tilde{\mathscr{T}}, \tilde{\mathscr{F}} \cup \{B\},$
         $\mathscr{L}$)
     end;
   if $\mathscr{T} \cup \mathscr{F} \subseteq$ VAR then
     begin
8.     if $\mathscr{T} \cap \mathscr{F} \neq \varnothing$ then return false;
9.     if $\mathscr{T} \cap \mathscr{F} = \varnothing$ and $\tilde{\mathscr{F}} \neq \varnothing$ then return $\bigwedge_{B\in\tilde{\mathscr{F}},(\tilde{\mathscr{T}},B)\notin\mathscr{L}}$ S4-WORLD($\tilde{\mathscr{T}}, \{B\},$
         $\tilde{\mathscr{T}}, \varnothing, \mathscr{L} \cdot (\tilde{\mathscr{T}}, B)$);
       return true
     end
 end
```

(Note that $\mathscr{L} \cdot (\tilde{\mathscr{T}}, B)$ is the sequence $\mathscr{L}$ extended with $(\tilde{\mathscr{T}}, B)$ and the conjunction over the empty set defined to be **true**.)

Test for $A \in S4$-PROVABLE.

```
begin
  read $A$
  $v \leftarrow \sim$S4-WORLD($\{\sim A\}, \varnothing, \varnothing, \varnothing, \varnothing$);
end
```

The value of v determines if A is $S4$-provable. We leave the verification of the algorithm to the reader.

Let $n = |A|$. In order to improve the space complexity of the algorithm we should let $\mathscr{L}$ be a global stack. If $\mathscr{L} = \{(\mathscr{T}_1, B_1), (\mathscr{T}_2, B_2), \cdots, (\mathscr{T}_k, B_k)\}$ then $\mathscr{T}_1 \subseteq \mathscr{T}_2 \subseteq \cdots \subseteq \mathscr{T}_k \subseteq$ subformulas of A. Since no repetitions can occur in the sequence $\mathscr{L}$, then $k \leqq n^2$. Hence $O(n^3)$ storage suffices for $\mathscr{L}$. What remains is an analysis of the number of levels of recursion in $S4$-WORLD. Since $\mathscr{L}$ is now a global stack, then $O(n)$ is all that is needed at each level of recursion. As before there can be at most $O(n)$ successive recursive calls all with $\mathcal{T} \cup \mathcal{F} \not\subseteq \mathrm{VAR}$. Further, because the cardinality of $\mathcal{L}$ is bounded by $O(n^2)$ there can be at most $O(n^2)$ depth of recursive calls with $\mathcal{T} \cup \mathcal{F} \subseteq \mathrm{VAR}$. Thus the recursion depth is $O(n^3)$. Total space is bounded by $O(n^4)$. Q.E.D.

To summarize the specific space bounds we give this corollary to the proof of Theorem 5.1.

COROLLARY 5.2. *K-PROVABLE $\in$ SPACE(n^2), T-PROVABLE $\in$ SPACE(n^3), and $S4$-PROVABLE $\in$ SPACE(n^4).*

We do not claim that these bounds are best possible, but they do guarantee that these problems are computable in polynomial space.

COROLLARY 5.3. *For $S \in \{K, T, S4\}$, S-PROVABLE is log space complete in P-SPACE.*

6. The complexity of provability in $S5$. The provability problem in $S5$ seems to be easier than that for the systems we have considered so far. For example, in T, K and $S4$ we can construct satisfiable formulas which are only satisfiable in exponential size model structures. This phenomenon does not happen for $S5$-satisfiability. Hence we can only show that $S5$-SATISFIABLE is log space complete in NP-TIME.

LEMMA 6.1. *If $A \in S5$-SATISFIABLE has m modal connectives, then A is $S5$-satisfiable in an $S5$-model with $\leqq m+1$ worlds.*

PROOF. Let A be satisfied in an $S5$-model (W, R, V). We may assume that $u\,R\,v$ for all $u, v \in W$. We construct a mapping σ from all instances of subformulas of A into W in such a way that A is $S5$-satisfied in $(\mathrm{Range}(\sigma), R|\mathrm{Range}(\sigma), V|\mathrm{Range}(\sigma))$ and the cardinality of $\mathrm{Range}(\sigma) \leqq m+1$.

The function σ is defined inductively on the instances of subformulas of A.

(i) Choose $\sigma(A) \in w$ such that $V(A, \sigma(A)) = T$,

(ii) $\sigma(C) = \sigma(B)$ if $B = \sim C$,

(iii) $\sigma(C) = \sigma(D) = \sigma(B)$ if $B = C \wedge D$,

(iv) $\sigma(C) = \sigma(B)$ if $B = \Box C$ and $V(B, \sigma(B)) = T$,

(v) if $B = \Box C$ and $V(B, \sigma(B)) = F$, then choose $\sigma(C) \in W$ in such a way that $V(C, \sigma(C)) = F$.

Clearly the cardinality of $\mathrm{Range}(\sigma) \leqq m+1$. Let $W' = \mathrm{Range}(\sigma)$ and let R' and V' be respectively R and V restricted to W'. We may show inductively that for each instance of a subformula B of A, $V(B, \sigma(B)) = V'(B, \sigma(B))$. Q.E.D.

THEOREM 6.2. *$S5$-SATISFIABLE is log space complete in NP-TIME.*

Proof. Trivially $\mathbf{B_1}$ is log space reducible to $S5$-SATISFIABLE, $\exists X_1 \cdots \exists X_m A \in \boldsymbol{B_1}$ if and only if $A \in S5$-SATISFIABLE.

It remains to show that $S5$-SATISFIABLE $\in NP$-TIME. Let $A \in$ MF and let $|A| = n$. By Lemma 6.1 $A \in S5$-SATISFIABLE if and only if there is an $S5$-model (W, R, V) with the cardinality of $W \leqq n+1$ and a $w \in W$ such that $V(A, w) = T$. Such a model can be "guessed" nondeterministically and checked in polynomial time. Q.E.D.

7. Conclusion. It would be interesting to determine cut off points between $S4$ and $S5$ where the complexity of satisfiability changes from complete in P-SPACE to complete in NP-TIME. We conjecture that $S4.3$-SATISFIABLE is log space complete in NP-TIME.

Another interesting area is the complexity of provability or validity in intuitionistic propositional logic (IC). J. Cherniavsky [1] claimed that the nonvalid formulas in IC can be determined in *NP*-TIME. He has since informed us of mistakes in his proof. We conjecture that provability in IC is log space complete in *P*-SPACE. There is a very simple reduction of IC to $S4$ given by McKinsey and Tarski [7]. Define τ inductively:

(i) $\tau(A)=A$ if A is a variable,
(ii) $\tau(A\wedge B)=\tau(A)\wedge\tau(B)$,
(iii) $\tau(A\supset B)=\Box(\tau(A)\supset\tau(B))$,
(iv) $\tau(\sim A)=\Box\sim\tau(A)$.

Now, A is IC-provable if and only if $\tau(A)$ is $S4$-provable. Thus IC-PROVABLE$\in P$-SPACE. All that remains is to show that $\mathbf{B}_\omega$ or some other complete set is log space reducible to IC-PROVABLE.

Acknowledgments. We appreciate the suggestions of S. K. Thomason in obtaining the results of § 3. Also we are indebted to J. Cherniavsky in providing helpful ideas that we used in our algorithms for *K*, *T*, and $S4$.

Note added in proof. M. J. Fischer has suggested a new construction which improves the bounds of § 4. For example $S=K$ implies $l(n)=O(n^2/\log n)$ in Lemma 4.1.

REFERENCES

[1] J. Cherniavsky, *The complexity of some non-classical logics*, 14th Ann. IEEE Symp. on Switching and Automata Theory (1973), pp. 209–213.

[2] S. A. Cook, *The complexity of theorem proving procedures*, Proc. 3rd Ann. ACM Symp. on Theory of Computing (1971), pp. 151–158.

[3] J. E. Hopcroft and J. D. Ullman, *Formal Languages and their Relation to Automata*, Addison-Wesley, Reading, MA, 1969.

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

[5] N. D. Jones, *Space-bounded reducibility among combinatorial problems*, J. Comput. System Sci., 11 (1975), pp. 68–85.

[6] S. A. Kripke, *Semantical analysis of modal logic, I. Normal modal propositional calculi*, Z. Math. Logik Grundlagen Math., 9 (1963), pp. 67–96.

[7] J. C. C. McKinsey and A. Tarski, *Some theorems about the sentential calculi of Lewis and Heyting*, J. Symbolic Logic, 13 (1948), pp. 1–15.

[8] V. R. Pratt, *Semantical considerations on Floyd–Hoare logic*, 17th Ann. IEEE Symposium on Foundations of Computer Science (1976), pp. 109–121.

[9] W. J. Savitch, *Relationship between nondeterministic and deterministic tape complexities*, J. Comput. System Sci., 4 (1970), pp. 177–192.

[10] J. I. Seiferas, M. J. Fischer and A. R. Meyer, *Refinements of the nondeterministic time and space hierarchies*, 14th Ann. IEEE Symp. on Switching and Automata Theory (1973), pp. 130–137.

[11] L. J. Stockmeyer, *The polynomial-time hierarchy*, IBM Tech. Rep. Yorktown Heights, NY, 1975.

[12] L. J. Stockmeyer and A. R. Meyer, *Word problems requiring exponential time: Preliminary report*, Proc. 5th Ann. ACM Symp. on Theory of Computing (1973), pp. 1–9.