INFORMATION AND CONTROL 61, 147–158 (1984)

# Diamond Formulas: A Fragment of Dynamic Logic with Recursively Enumerable Validity Problem

PETER H. SCHMITT

*Mathematisches Institut, Universität Heidelberg, West Germany*

The set of diamond formulas is built up inductively from first-order formulas by closing off under first-order quantifiers, conjunction, disjunction, and $\langle\alpha\rangle$, for regular programs $\alpha$ which may themselves contain tests which are diamond formulas. A complete proof system for diamond formulas is presented with a recursive set of axioms and only finitary rules. As a consequence the dual set of box formulas satisfies the compactness theorem. 

## 1. INTRODUCTION

The main purpose of this paper is to prove a generalization of a result due to Meyer and Halpern. They showed in [Meyer and Halpern, 1982] that the set of valid termination assertions is recursively enumerable; a termination assertion is a formula of the form $\varphi \to \langle\alpha\rangle\psi$ with $\varphi$ and $\psi$ first-order formulas and $\alpha$ a regular program possibly containing tests which themselves are, inductively, such termination assertions. Meyer and Halpern present, in fact, a complete proof system for termination assertions with a recursive set of axioms and only finitary rules.

We will define below the set $\langle\ \rangle$ DL of diamond formulas which contains all termination assertions and is closed under first-order quantifiers, and $\langle\ \rangle$, and under conjuction and disjunction but not under negation. Diamond formulas do not contain boxes. We exhibit a complete effective proof system for diamond formulas thus showing the validity problem to be recursively enumerable. This result seems quite sharp when compared with the theorem proved in [Harel, Meyer, and Pratt, 1977] that the validity problem for fomulas of the form $\exists y[\alpha]\varphi$ with $\varphi$ first-order is already $\Pi_1^1$-complete.

It does not seem possible to extend the method of proof used by Meyer and Halpern so that it also covers our more general theorem. Instead we embed first-order dynamic logic in the infinitary logic $L_{\omega_1\omega}$, an idea that was already employed by Engeler in his pioneering paper [Engeler, 1967] and is systematically used, e.g., in [Harel, 1982] to prove uninterpreted completeness results. We carry this idea a small step further in that we do not copy the model existence theorem for $L_{\omega_1\omega}$ in the setting of dynamic



0019-9958/84 $3.00

logic, we rather use the embedding mentioned above to apply it. This trick is not restricted to the treatment of diamond formulas; it may just as well be used to prove the known infinitary completeness theorems for dynamic logic and its various extensions.

Among the corollaries to our main result discussed in Section 4 we want to mention here the compactness theorem for box formulas. Box formulas are defined in the obvious way dually to diamond formulas. We prove that an arbitrary set $\Delta$ of box formulas is satisfiable if every finite subset of $\Delta$ is satisfiable.

I would like to thank the referee for pointing out some incorrectnesses in the first version of this paper.

## 2. Prerequisites

We assume some aquaintance with the infinitary logic $L_{\omega_1\omega}$. The first four chapters of [Keisler, 1971] will suffice. We follow in syntax and semantics the standard definition of the set RG of first-order regular programs and the set QDL of formulas of first-order dynamic logic, see, e.g., [Harel, 1979, p. 12 ff.]. But let us describe in greater detail how we deal with substitution. We adopt the idea suggested in [Harel, 1982, Sect. 3.4.2]. For a regular program using the variables $\mathbf{y} = y_1,\ldots, y_k$ and $\varphi$ in QDL we accept only $[\mathbf{y} \leftarrow \mathbf{x}; \alpha; \mathbf{x} \leftarrow \mathbf{y}]\,\varphi$ (resp. $\langle \mathbf{y} \leftarrow \mathbf{x}; \alpha; \mathbf{x} \leftarrow \mathbf{y} \rangle\,\varphi$) with $y_1,\ldots, y_k$ not occuring in $\alpha$ and $\mathbf{x} = x_1,\ldots, x_k$ not occuring in $\varphi$ as well-formed expressions in QDL, i.e., the input–output variables $\mathbf{x}$ are not allowed to be used in the execution of the program $\alpha$ and the working variables $\mathbf{y}$ of $\alpha$ should not appear outside the box (resp. the diamond). Of course $\mathbf{y} \leftarrow \mathbf{x}$ abbreviates $y_1 \leftarrow x_1;\ldots; y_k \leftarrow x_k$. We will, however, for denotational simplicity continue to write $[\alpha]\,\varphi$, $\langle\alpha\rangle\,\varphi$ whenever possible; this is to be understood as an abbreviation for the corresponding syntactically correct expressions. The technical complication just described is one way to arrive at a satisfying concept of bound and free variables. An occurence of a variable $z$ in a QDL-formula $\varphi$ is called bound, if it occurs

(i) within a subformula of the form $\forall z\varphi_0$ or $\exists z\varphi_0$;

(ii) within a subformula of the form $[\ldots; z \leftarrow x;\ldots; \alpha;\ldots; x \leftarrow z;\ldots]\,\varphi_0$ or $\langle\ldots; z \leftarrow x;\ldots; \alpha;\ldots; x \leftarrow z;\ldots\rangle\,\varphi_0$;

(iii) within a subformula of the form $[\ldots; y \leftarrow z;\ldots; \alpha;\ldots; z \leftarrow y;\ldots]\,\varphi_0$ or $\langle\ldots; y \leftarrow z;\ldots; \alpha;\ldots; z \leftarrow y;\ldots\rangle\,\varphi_0$ with the exception that in the input assignment $y \leftarrow z$, $z$ is not counted as a bound occurence.

Every occurence of a variable which is not bound is called free. For $\sigma$ a term, $x$ a variable, $\varphi$ a QDL-formula, we obtain $\varphi(\sigma/x)$ by substituting for

every free occurence of $x$ in $\varphi$ the term $\sigma$ renaming bound variables to avoid capture if necessary. The care we have taken in defining substitution guarantees that for every structure $\mathfrak{A}$ and every state $s$ (which for our purposes will just be a function assigning elements of $\mathfrak{A}$ to the variables of QDL) we have

LEMMA 2.1. $(\mathfrak{A}, s) \models \varphi(\sigma/x)$ *iff* $(\mathfrak{A}, s(\sigma/x)) \models \varphi$.

Here $s(\sigma/x)$ denotes the function that differs from $s$ only at the argument $x$, where it assumes the value $(\mathfrak{A}, s)(\sigma)$. Later definitions and proofs will proceed by induction on the complexity of QDL-formulas and regular programs. We give for definiteness one possible complexity function $c$ taking countable ordinals as values:

$$c(x \leftarrow \sigma) = 1$$

$$c(\alpha;\beta) = c(\alpha) + c(\beta) + 2 \qquad c(\alpha \cup \beta) = \max\{c(\alpha), c(\beta)\} + 1$$

$$c(\alpha^*) = c(\alpha) \times \omega \qquad c(\varphi?) = c(\varphi) + 1$$

$$c(\varphi) = 0 \qquad \text{if } \varphi \text{ is a quantifierfree first-order formula}$$

$$c(\varphi \wedge \psi) = c(\varphi \vee \psi) = \max\{c(\varphi), c(\psi)\} + 1$$

$$c(\neg\varphi) = c(\varphi) + 1$$

$$c(\forall x\varphi) = c(\exists x\varphi) = c(\varphi) + 1$$

$$c([\alpha]\varphi) = c(\langle\alpha\rangle\varphi) = c(\alpha) + c(\varphi) + 1.$$

The embedding $v$ of QDL into $L_{\omega_1\omega}$ is a first example of a definition using induction on $c(\varphi)$.

$$v(\varphi) = \varphi \qquad \text{if } \varphi \text{ is a first-order formula}$$

$$v(\neg\varphi) = \neg v(\varphi)$$

$$v(\varphi_1 \wedge \varphi_2) = v(\varphi_1) \wedge v(\varphi_2), \qquad v(\varphi_1 \vee \varphi_2) = v(\varphi_1) \vee v(\varphi_2)$$

$$v(\forall x\varphi) = \forall x v(\varphi) \qquad v(\exists x\varphi) = \exists x v(\varphi)$$

$$v([x \leftarrow \sigma]\varphi) = v(\langle x \leftarrow \sigma\rangle\varphi) = v(\varphi)(\sigma/x)$$

$$v([\alpha;\beta]\varphi) = v([\alpha][\beta]\varphi) \qquad v(\langle\alpha;\beta\rangle\varphi) = v(\langle\alpha\rangle\langle\beta\rangle\varphi)$$

$$v([\alpha \cup \beta]\varphi) = v([\alpha]\varphi) \wedge v([\beta]\varphi) \qquad v(\langle\alpha \cup \beta\rangle\varphi) = v(\langle\alpha\rangle\varphi \vee v(\langle\beta\rangle\varphi)$$

$$v([\psi?]\varphi) = v(\psi) \to v(\varphi) \qquad v(\langle\psi?\rangle\varphi) = v(\psi) \wedge v(\varphi)$$

$$v([\alpha^*]\varphi) = \bigwedge_{n \geq 0} v([\alpha]^n\varphi) \qquad v(\langle\alpha^*\rangle\varphi) = \bigvee_{n \geq 0} v([\alpha]^n\varphi).$$

We use $[\alpha]^n \varphi$ as shorthand for $[\alpha] \cdots [\alpha] \varphi$ with $n$ repetitions of $\alpha$. In particular, $[\alpha]^0 \varphi$ is just $\varphi$.

LEMMA 2.2. *For any* QDL*-formula* $\varphi$ *and structure* $\mathfrak{A}$ *and state* $s$,

$$(\mathfrak{A}, s) \models \varphi \quad \textit{iff} \quad (\mathfrak{A}, s) \models \nu(\varphi).$$

*Proof.* Immediate by induction on the complexity of $\varphi$.

## 3. THE MAIN THEOREM

We assume for the remainder of this paper a fixed set of relation, function, and constant symbols. The dependence of subsequent definitions on this set will not be displayed in our notation.

The sets $\langle\ \rangle$RG of diamond regular programs and $\langle\ \rangle$DL of diamond formulas are defined by simultaneous induction. The definition of $\langle\ \rangle$RG is literally the same as for RG with the sole exception, that tests $\varphi?$ are only included for $\langle\ \rangle$DL-formulas $\varphi$. For diamond formulas we have the following inductive clauses:

(i) first-order formulas are in $\langle\ \rangle$DL;

(ii) if $\varphi, \psi$ are in $\langle\ \rangle$DL so are $\varphi \wedge \psi$, $\varphi \vee \psi$, $\forall x \varphi$, $\exists x \varphi$;

(iii) if $\varphi$ is in $\langle\ \rangle$DL, $\alpha$ in $\langle\ \rangle$RG then $\langle \alpha \rangle \varphi$ is in $\langle\ \rangle$DL.

Thus diamond formulas do not contain boxes. We use $\varphi \to \psi$ as an abbreviation for $\neg\varphi \vee \psi$. For $\varphi \to \psi$ to be in $\langle\ \rangle$DL, $\varphi$ has to be a first-order formula. Dually we define box regular programs, $[\ ]$RG, and box formulas, $[\ ]$DL, by replacing in clause (iii), $\langle \alpha \rangle$ by $[\alpha]$.

In trying to write down a proof system we encounter the difficulty that for a diamond formula $\varphi$ we want to use the valid QDL-formula $\varphi \to \exists x \varphi$ as an axiom. But this is in general, not a diamond formula. We therefore replace the axiom $\varphi \to \exists x \varphi$ by the rule $\varphi \vdash \exists x \varphi$ which allows to pass from the $\langle\ \rangle$DL-formula $\varphi$ to the $\langle\ \rangle$DL-formula $\exists x \varphi$. In the same way we deal with all the other candidates for axioms.

### *Axioms: All Valid First-order Formulas*

I. *Structure Rules*

DISJUNCTION RULE. $\varphi_1 \vee \cdots \vee \varphi_k \vdash \psi_1 \cdots \psi_k$ *if* $\{\varphi_1, \ldots, \varphi_k\} \subseteq \{\psi_1, \ldots, \psi_k\}$.

RENAMING RULE. $\varphi \vee \psi \vdash \varphi \vee \psi'$, *where* $\psi'$ *arises from* $\psi$ *be renaming bound variables; of course in such a way that formerly free variables remain free.*

The disjunction rule serves many purposes. It allows to permute the disjunctive components in $\varphi_1 \vee \cdots \vee \varphi_k$ by taking $\psi_i = \varphi_{s(i)}$ for a permutation $s$ of $\{1,...,k\}$. It allows to weaken $\varphi_1 \vee \cdots \vee \varphi_k$ to $\varphi_1 \vee \cdots \vee \varphi_k \vee \varphi_{k+1}$ and to get rid of duplications, e.g., $\varphi \vee \varphi \vdash \varphi$.

II. *Logic Rules*

CONJUNCTION RULE. $\varphi \vee \psi_1, \varphi \vee \psi_2 \vdash \varphi \vee (\psi_1 \wedge \psi_2)$.

$\exists$-RULE. $\varphi \vee \psi(\sigma/x) \vdash \varphi \vee \exists x \psi$.

$\forall$-RULE. $\varphi \vee \psi(x) \vdash \varphi \vee \forall x \psi(x)$, *where $x$ does not occur free in $\varphi$.*

CHAIN RULE. $\varphi \vee \chi, \neg\chi \vee \psi \vdash \varphi \vee \psi$ *for first-order formulas $\chi$.*

III. *Program Rules*

ASSIGNMENT RULE. $\varphi \vee \psi(\sigma/x) \vdash \varphi \vee \langle x \leftarrow \sigma \rangle \psi$.

COMPOSITION RULE. $\varphi \vee \langle \alpha \rangle \langle \beta \rangle \psi \vdash \varphi \vee \langle \alpha; \beta \rangle \psi$.

UNION RULE. $\varphi \vee \langle \alpha \rangle \psi \vee \langle \beta \rangle \psi \vdash \varphi \vee \langle \alpha \cup \beta \rangle \psi$.

TEST RULE. $\varphi \vee (\psi \wedge \chi) \vdash \varphi \vee \langle \psi? \rangle \chi$.

ITERATION RULE. $\varphi \vee \langle \alpha \rangle^n \psi \vdash \varphi \vee \langle a^* \rangle \psi$.

It is to be understood that the side formula $\varphi$ in the above rules may be missing. We say that a $\langle\ \rangle$ DL-formula $\varphi$ is a syntactic consequence of $\langle\ \rangle$ DL-formula $\psi$, in symbols $\psi \vdash \varphi$, if there is a finite sequence $\varphi_1,..., \varphi_k$ of $\langle\ \rangle$ DL-formulas with $\varphi_k = \varphi$ and for all $i$, $1 \leqslant i \leqslant k$, $\varphi_i$ is either $\psi$ or an axiom or follows from one or two of the formulas $\varphi_j$, $j < i$ by one application of a rule. The obvious modification yields the concept of a syntactically valid $\langle\ \rangle$ DL-formula $\varphi$, in symbols $\vdash \varphi$.

Since diamond formulas are not closed under negation a proof system for $\langle\ \rangle$ DL cannot contain the general modus ponens rule. The following lemma will be used as a substitute.

LEMMA 3.1. *If $\psi_1 \vdash \psi_2$ and $\chi \vdash \psi_1 \vee \varphi$ then $\chi \vdash \psi_2 \vee \varphi$.*

*Proof.* Let $\chi_1,..., \chi_k$ be a proof of $\psi_1 \vee \varphi$ from $\chi$ and $\rho_1,..., \rho_r$ a proof of $\psi_2$ from $\psi_1$. Observing that replacing a free variable in a proof by a new variable not occuring there before yields again a correct proof and using the renaming rule, we may assume without loss of generality that in the proof $\rho_1,..., \rho_r$ the $\forall$-rule is never applied to a variable $x$ which occurs free in $\varphi$.

This suffices to see that the concatenated sequence $\chi_1,\ldots,\chi_k, \rho_1 \vee \varphi,\ldots,\rho_r \vee \varphi$ is again a correct proof verifying $\chi \vdash \psi_2 \vee \varphi$.

Before we state the main theorem let us take a closer look at the embedding $v$. We have the useful property $v(\varphi(\sigma/x)) = v(\varphi)(\sigma/x)$ of which the reader may easily convince himself. With respect to other syntactic operations $v$ does not behave so smoothly. To mention just one example $v(\varphi)$ may be of the form $\chi_1 \vee \chi_2$ while $\varphi$ itself is not a disjunction; $\varphi$ might be $\langle x \leftarrow \sigma\rangle$, $(\varphi_1 \vee \varphi_2)$, or even $\langle \alpha \cup \beta\rangle \varphi_0$. This behaviour of $v$ complicates proofs by induction on the complexity of $v(\varphi)$. The way we cope with this problem is summarized in Lemma 3.2. This lemma is phrased so that it will be immediately applicable in the proof of the main theorem; in particular it uses a technical device already introduced in [Keisler, 1971] which we found also useful in our context. For a QDL-formula $\varphi$ we let $\varphi\neg$ denote the formula which arises from $\neg\varphi$ by moving the negation sign inside till it is in front of an atomic formula, where also multiple occurences of $\neg$ are eliminated.

The formal inductive definition reads: If $\varphi$ is atomic then $\varphi\neg = \neg\varphi$,

$$(\neg\varphi)\neg = \varphi$$

$$(\varphi \wedge \psi)\neg = \varphi\neg \vee \psi\neg \qquad (\varphi \vee \psi)\neg = \varphi\neg \wedge \psi\neg$$

$$(\forall x\varphi)\neg = \exists x\varphi\neg \qquad (\exists x\varphi)\neg = \forall x\varphi\neg$$

$$([\alpha]\varphi)\neg = \langle\alpha\rangle\varphi\neg \qquad (\langle\alpha\rangle\varphi)\neg = [\alpha]\varphi\neg.$$

If $\varphi$ is a diamond formula then $\varphi\neg$ is a box formula and vice versa, while in general $\neg\varphi$ is neither a diamond nor a box formula.

LEMMA 3.2. *Let $\varphi$ be a diamond formula.*

(i) *If $v(\varphi\neg) = \chi$ for $\chi$ an atomic or negated atomic formula then $\vdash \neg\chi \to \varphi$.*

(ii) *If $v(\varphi\neg) = \chi_1 \wedge \chi_2$ then there are diamond formulas $\varphi_1; \varphi_2$ such that $v(\varphi_i\neg) = \chi_i$ and $\varphi_i \vdash \varphi$ for $i = 1, 2$.*

(iii) *If $v(\varphi\neg) = \chi_1 \vee \chi_2$ then there are diamond formulas $\varphi_1, \varphi_2$ such that $v(\varphi_i\neg) = \chi_i$ for $i = 1, 2$ and $\varphi_1 \wedge \varphi_2 \vdash \varphi$.*

(iv) *If $v(\varphi\neg) = \bigwedge_{n\geqslant 0} \chi_n$ then there are a diamond formula $\psi$ and a program $\alpha$ such that for all $n \geqslant 0$, $v((\langle\alpha\rangle^n \psi)\neg) = \chi_n$ and $\langle\alpha^*\rangle \psi \vdash \varphi$.*

(v) *If $v(\varphi\neg) = \forall x\chi(x)$ then there is a diamond formula $\psi$ such that $v(\psi\neg) = \chi$ and $\exists x\psi \vdash \varphi$.*

(vi) *If $v(\varphi\neg) = \exists x\chi(x)$ then there is a diamond formula $\psi$ such that $v(\psi\neg) = \chi$ and $\forall x\psi \vdash \varphi$.*

Since the proofs of these six assertions all follow along the same pattern

we give details only for parts (i), (ii), and (iv). In each case we proceed by induction on the complexity of $\nu(\varphi)$.

(i) If $\varphi$ is a first-order formula then we have $\nu(\varphi\neg) = \varphi\neg = \chi$ and $\neg\chi \to \varphi$ is a first-order axiom. If $\varphi$ is not first-order there are two cases.

*Case* 1. $\varphi = \langle x \leftarrow \sigma\rangle \varphi_0$. By definition of $\nu$ we have $\nu(\varphi\neg) = \nu(\varphi_0\neg)(\sigma/x) = \nu(\varphi_0\neg(\sigma/x)) = \chi$. By induction hypothesis we have $\neg\chi \to \varphi_0(\sigma/x)$, from which an application of the assignment rule yields $\vdash \neg\chi \to \langle x \leftarrow \sigma\rangle \varphi_0$.

*Case* 2. $\varphi = \langle \alpha;\beta\rangle \varphi_0$. By definition of $\nu$ we have $\nu(\varphi\neg) = \nu(\langle\alpha\rangle\langle\beta\rangle \varphi_0)\neg) = \chi$. By induction hypothesis $\vdash \neg\chi \to \langle\alpha\rangle\langle\beta\rangle \varphi_0$ from which an application of the composition rule yields $\vdash \neg\chi \to \langle \alpha;\beta\rangle \varphi_0$.

(ii) If $\varphi$ is first-order or of one of the forms $\langle x \leftarrow \sigma\rangle \varphi_0$ or $\langle \alpha;\beta\rangle \varphi_0$ we proceed as in the verification of (i). This time however there are two additional possibilities.

*Case* A. $\varphi = \varphi_1 \vee \varphi_2$. Then $\nu(\varphi\neg) = \nu(\varphi_1\neg) \wedge \nu(\varphi_2\neg)$. Thus $\nu(\varphi_i\neg) = \chi_i$ and $\varphi_i \vdash \varphi$ follows from the disjunction rule.

*Case* B. $\varphi = \langle \alpha \cup \beta\rangle \varphi_0$. From $\nu(\varphi\neg) = \nu(\langle\alpha\rangle \varphi_0)\neg) \wedge \nu(\langle\beta\rangle \varphi_0)\neg)$ we get $\nu(\langle\alpha\rangle \varphi_0)\neg) = \chi_1$ and $\nu(\langle\beta\rangle \varphi_0)\neg) = \chi_2$. Furthermore we derive from $\langle\alpha\rangle \varphi_0$ by the disjunction rule $\langle\alpha\rangle \varphi_0 \vee \langle\beta\rangle \varphi_0$ and, by a subsequent application of the union rule, $\langle \alpha \cup \beta\rangle \varphi_0$. Similarly $\langle\beta\rangle \varphi_0 \vdash \varphi$ is obtained.

(iv) Apart from the cases that $\varphi$ is first-order or $\langle x \leftarrow \sigma\rangle \psi$ or $\langle \alpha;\beta\rangle \psi$ there is one additional case $\varphi = \langle \alpha^*\rangle \psi$. But in this case the assertion is trivially satisfied.

MAIN THEOREM 3.3. *For any diamond formula* $\varphi$:

$$\vdash \varphi \qquad \textit{iff } \varphi \textit{ is universally valid.}$$

That every syntactically valid diamond formula is universally valid is easily checked and left to the reader. For the converse implication we will construct a model $\mathfrak{A}$ and a state $s$ such that $(\mathfrak{A}, s) \models \neg\varphi$ for every diamond formula $\varphi$ such that not $\vdash \varphi$. More precisely we will first obtain $(\mathfrak{A}, s) \models \nu(\neg\varphi)$ by applying the model existence theorem for $L_{\omega_1\omega}$ and then use Lemma 2.2.

For the convenience of the reader we repeat the definition of a consistency property which is the central notion involved in formulating the model existence theorem for $L_{\omega_1\omega}$. We begin by adding to the primitive symbols that have been present so far a countably infinite set $C$ of new constant symbols. The set of $L_{\omega_1\omega}$ formulas that may be built using constants from $C$ is denoted by $L_{\omega_1\omega}(C)$.

DEFINITION. A consistency property $S$ is a set of countable subsets of sentences from $L_{\omega_1\omega}(C)$ (i.e., formulas from $L_{\omega_1\omega}(C)$ with free variables) such that for all $s \in S$ the following hold:

(C1) for every atomic or negated atomic $\varphi \in L_{\omega_1\omega}(C)$ either $\varphi \notin s$ or $\neg\varphi \notin s$;

(C2) if $\neg\varphi \in s$ then $s \cup \{\varphi\neg\} \in S$;

(C3) if $\bigwedge_{n<m} \varphi_n \in s$, $0 < m \leqslant \infty$, then $s \cup \{\varphi_n\} \in S$ for all $n$, $0 \leqslant n < m$;

(C4) if $\forall x\varphi \in s$ then $s \cup \{\varphi(c)\} \in S$ for all $c \in C$;

(C5) if $\bigvee_{n<m} \varphi_n \in s$ then for some $n < m$, $s \cup \{\varphi_n\} \in S$;

(C6) if $\exists x\varphi \in s$ then $s \cup \{\varphi(c)\} \in S$ for some $c \in C$,

(C7) for every term $\sigma$ and $c, d \in C$:

(a) if $(c = d) \in s$ then $s \cup \{d = c\} \in S$;

(b) if $c = \sigma$, $\varphi(\sigma/x) \in s$ for some atomic or negated atomic formula $\varphi$ then $s \cup \{\varphi(c/x)\} \in S$;

(c) for some $e \in C$, $s \cup \{e = \sigma\} \in S$.

MODEL EXISTENCE THEOREM. *If $S$ is a consistency property and $s \in S$, then $s$ has a model.*

See [Keisler, 1971] for a proof.

We are now ready to deal with the difficult direction of the main theorem. Let $\langle\ \rangle\mathrm{DL}(C)$ be the set of diamond formulas that may use, in addition to the up-to-now fixed vocabulary, the new constants from the countably infinite set $C$. We define a collection $S$ of finite subsets $s \subseteq L_{\omega_1\omega}(C)$ by requiring $s \in S$ iff there are sentences $\varphi_1, \ldots, \varphi_k$ in $\langle\ \rangle\mathrm{DL}(C)$ such that

$$s = \{v(\varphi_1\neg), \ldots, v(\varphi_k\neg)\} \qquad \text{and} \qquad \text{not} \vdash \varphi_1 \vee \cdots \vee \varphi_k.$$

*Claim* 3.4. $S$ is a consistency property. Let $s = \{v(\varphi_1), \ldots, v(\varphi_k)\} \in S$ be given. Thus we have not $\vdash \varphi_1 \vee \cdots \vee \varphi_k$.

(C1) Assume for the sake of a contradiction that for some atomic sentence $\chi$ we have, say, $v(\varphi_1\neg) = \chi$ and $v(\varphi_2\neg) = \neg\chi$. Since $\neg\chi \vee \neg\neg\chi$ is among our axioms we obtain by the chain rule and Lemma 3.2(i), $\vdash \varphi_1 \vee \varphi_2$ and therefore by the disjunction rule, $\vdash \varphi_1 \vee \cdots \vee \varphi_k$, a contradiction.

(C2) If $\neg\varphi \in s$ then $\neg\varphi = v(\psi\neg)$ for some diamond formula $\psi$. Since $\neg$-signs occur in $\psi\neg$ only in front of atomic formulas, $\varphi$ has to be atomic. Thus $\neg\varphi = \varphi\neg$ and $s \cup \{\varphi\neg\} = s$.

(C3) We treat the case of a finite conjunction first. It certainly suffices to consider conjunctions of two formulas. Let $v(\varphi_1\neg) = \chi_1 \wedge \chi_2$. By

Lemma 3.2(ii) there are diamond sentences $\psi_1, \psi_2$ such that $v(\psi_i\neg) = \chi_i$ and $\psi_i \vdash \varphi_1$ for $i = 1, 2$. Assume $\vdash \varphi_1 \vee \cdots \vee \varphi_k \vee \psi_i$, then we get by Lemma 3.1, $\vdash \varphi_1 \vee \cdots \vee \varphi_k \vee \varphi_1$ which yields by the disjunction rule, the contradiction $\vdash \varphi_1 \vee \cdots \vee \varphi_k$. Thus $s \cup \{v(\psi_i\neg)\} = s \cup \{\chi_i\} \in S$.

Turning to the infinite conjunction, let $v(\varphi_1) = \bigwedge_{n\geqslant 0} \chi_n$. By Lemma 3.2(iv) there are $\alpha$ and $\psi$ with $v((\langle\alpha\rangle^n \psi)\neg) = \chi_n$ for all $n$ and $\langle\alpha^*\rangle \psi \vdash \varphi$. If for some $n$ $\vdash \varphi_1 \vee \cdots \vee \varphi_k \vee \langle\alpha\rangle^n \psi$ we obtain by the iteration rule $\vdash \varphi_1 \vee \cdots \vee \varphi_k \vee \langle\alpha^*\rangle \psi$ from which we get by Lemma 3.1 and the disjunction rule, the contradiction $\vdash \varphi_1 \vee \cdots \vee \varphi_k$.

(C4) Use Lemma 3.2(v) and the $\exists$-rule.

(C5) Since $v(\varphi\neg)$ for a diamond formula $\varphi$ can never be an infinite disjunction it suffices to treat the case $v(\varphi_1\neg) = \chi_1 \vee \chi_2$. By Lemma 3.2(iii) there are diamond sentences $\psi_1, \psi_2$ such that $v(\psi_i\neg) = \chi_i$ for $i = 1, 2$ and $\psi_1 \wedge \psi_2 \vdash \varphi_1$. Assume for the sake of a contradiction $\vdash \varphi_1 \vee \cdots \vee \varphi_k \vee \psi_1$ and $\vdash \varphi_1 \vee \cdots \vee \varphi_k \vee \psi_2$. By the conjunction rule we obtain $\vdash \varphi_1 \vee \cdots \vee \varphi_k \vee (\psi_1 \wedge \psi_2)$ and, using Lemma 3.1, we arrive at the contradiction $\vdash \varphi_1 \vee \cdots \vee \varphi_k \vee \varphi_1$.

(C6) Let $v(\varphi_1\neg) = \exists x\chi$. By Lemma 3.2(vi) there is a diamond formula $\psi$ such that $v(\psi\neg) = \chi$ and $\forall x\psi \vdash \varphi_1$. Assume $\vdash \varphi_1 \vee \cdots \vee \varphi_k \vee \psi(c/x)$ for $c \in C$ not occuring in $\varphi_1 \vee \cdots \vee \varphi_k$. By the $\forall$-rule this yields $\vdash \varphi_1 \vee \cdots \vee \varphi_k \vee \forall x\psi$ and therefore the contradiction $\vdash \varphi_1 \vee \cdots \vee \varphi_k \vee \varphi_1$ by Lemma 3.1. Thus $s \cup \{v(\psi\neg(c/x))\} \in S$ which is because of $v(\psi\neg) = \chi$, which we want.

(C7) (a) Let $v(\varphi_1\neg) = (c = d)$. By Lemma 3.2(i) $\vdash \neg(c = d) \to \varphi_1$. Using the chain rule and the tautology $\neg(d = c) \to \neg(c = d)$ we obtain $\vdash \neg(d = c) \to \varphi_1$. Thus $\vdash \varphi_1 \vee \cdots \vee \varphi_k \vee \neg(d = c)$ would yield the contradiction $\vdash \varphi_1 \vee \cdots \vee \varphi_k$. Therefore we have $s \cup \{d = c\} \in S$.

(b) Let $v(\varphi_1\neg) = (c = \sigma)$ and $v(\varphi_2\neg) = \varphi(\sigma/x)$. By Lemma 3.2(i) we have $\vdash \neg(c = \sigma) \to \varphi_1$ and $\vdash \neg\varphi(\sigma/x) \to \varphi_2$. The disjunction rule yields $\vdash \neg(c = \sigma) \to \varphi_1 \vee \varphi_2$ and $\vdash \neg\varphi(\sigma/x) \to \varphi_1 \vee \varphi_2$, from which we obtain by the conjunction rule $\vdash \neg(c = \sigma \wedge \varphi(\sigma/x)) \to \varphi_1 \vee \varphi_2$. If $\vdash \varphi_1 \vee \cdots \vee \varphi_k \vee \varphi(c/x)$ then we get from Lemma 3.1 and the first-order tautology $\neg\varphi(c/x) \to \neg(c = \sigma \wedge \varphi(\sigma/x))$, the contradiction $\vdash \varphi_1 \vee \cdots \vee \varphi_k \vee \varphi_1 \vee \varphi_2$.

(c) Choose some $e \in C$ that does not occur in $\varphi_1 \cdots \varphi_k$ nor in $\sigma$. If $\vdash \varphi_1 \vee \cdots \vee \varphi_k \vee \neg\ (e = \sigma)$ then we get by the $\forall$-rule $\vdash \varphi_1 \vee \cdots \vee \varphi_k \vee \forall x(\neg(x = \sigma))$. Since $\neg\forall x\neg(x = \sigma)$ is a first-order tautology, we obtain by the chain rule again the contradiction $\vdash \varphi_1 \vee \cdots \vee \varphi_k$.

Now there remains only one small wrinkle in the proof of the main theorem. Assume not $\vdash \varphi$ for some diamond formula $\varphi$. We cannot infer right away that $\{v(\varphi\neg)\}$ is in $S$ since $v(\varphi\neg)$ and therefore $\varphi$ is not allowed to

contain free variables. We therefore replace first the free variables $x_1,\ldots, x_m$ in $\varphi$ consistently by new constant symbols $e_1,\ldots, e_m$ to obtain a sentence $\varphi'$. This may necessitate an extension of the vocabulary. If $\vdash\varphi'$ we may assume using the renaming rule that in a proof of $\varphi'$ the $\forall$-rule is never applied for one of the variables $x_1,\ldots, x_m$. Now we see that replacing the constants $e_i$ by the variables they were originally substituted for yields a proof of $\varphi$, i.e., not $\vdash\varphi$ implies not $\vdash\varphi'$. Now $\{v(\varphi'\neg)\}\in S$ and the model existence theorem provides us with a structure $\mathfrak{A}$ such that $\mathfrak{A}\models\neg\varphi'$. For any state $s$ such that $s(x_i)=\mathfrak{A}(e_i)$, for $1\leqslant i\leqslant m$ we then have $(\mathfrak{A}, s)\models\neg\varphi$.

COROLLARY 3.5. *The set of valid diamond formulas is recursively enumerable.*

## 4. EXTENSIONS AND COROLLARIES

We mention two extensions of the main theorem. We may extend the class RG of regular programs to the class CF of context-free programs. This is done by replacing the iteration operator * by the recursion operator $\mu X\tau(X)$ (or more generally, $\mu X_1\cdots X_n(\tau_1,\ldots,\tau_n)$). The language CFDL is defined much as QDL but now using programs from CF instead of RG. For details see [Harel, 1979].

Diamond context-free programs $\langle\ \rangle$CF and diamond context-free formulas $\langle\ \rangle$CFDL are defined by the same simultaneous induction as before by simply replacing RG by CF and DL by CFDL. If we exchange the iteration rule for the (recursion rule) $\varphi\vee\langle\tau^n(\text{false?})\rangle\psi\vdash\varphi\vee\langle\mu X\tau(X)\rangle\psi$, using the notation from [Harel, 1979, p. 46], the main theorem remains true for diamond context-free formulas.

A second extension of the main theorem consists in not only providing a proof system for validity but for the more general notion of semantic consequence $\Delta\models\varphi$, which asserts that for all structures $\mathfrak{A}$ and all states $s$, if $(\mathfrak{A}, s)\models\psi$ for all $\psi$ in $\Delta$ then $(\mathfrak{A}, s)\models\varphi$.

Let us first observe that no complete finitary proof system is possible if we allow the set $\Delta$ to be infinite and contain diamond formulas. Consider $\Delta=\{d\neq f^{(n)}(c): n\geqslant 1\}\cup\{\forall x\langle z\leftarrow c;(z\leftarrow f(z))^*; y\leftarrow z\rangle(x=y)\}$. Then we have $\Delta\models(d=c)$. If a complete finitary proof system existed there should be a finite subset $\Delta_0$ of $\Delta$ with $\Delta_0\models(d=c)$. But there is a counterexample to this: interpret $f$ as the successor function on the natural numbers, $c$ as 0 and interpret $d$ by the first integer $m$ which is greater than any $n$ for which $d=f^{(n)}(c)$ appears in $\Delta_0$.

On the other hand, if we restrict $\Delta$ to be countable and contain only box formulas but require, as before, $\varphi$ to be a diamond formula then we are able to derive a complete finitary proof system for $\Delta\models\varphi$. We define for a coun-

table set $\Delta$ of box formulas and any diamond formula $\varphi$ the syntactic relation $\Delta \Vdash \varphi$ by $\Delta \Vdash \varphi$ iff there are finitely many $\psi_1, \ldots, \psi_k \in \Delta$ such that

$$\vdash \psi_1\neg \vee \cdots \vee \psi_k\neg \vee \varphi.$$

THEOREM 4.1. *$\Delta \Vdash \varphi$ iff $\Delta \models \varphi$.*

*Proof.* The implication from left to right is easily checked. Now for the converse implication. Replacing free variables by new constants we may assume that $\Delta \cup \{\varphi\}$ contains only sentences. Since $\psi\neg\neg = \psi$ is not always true, but $\psi\neg\neg\neg = \psi\neg$ is, we need to change $\Delta$ into $\Delta' = \{\psi\neg\neg : \psi \in \Delta\}$. Let us assume that not $\Delta \Vdash \varphi$. Obviously this implies not $\Delta' \Vdash \varphi$. By $S$ let us denote the consistency property defined in the proof of the Main Theorem 3.3. It is easily checked that the set $S_1$ defined by $s \in S_1$ iff $s \cup \{\nu(\psi\neg\neg) : \psi \in \Delta_0\} \in S$ for every finite subset $\Delta_0$ of $\Delta$ is again a consistency property and $\{\nu(\varphi\neg)\} \in S_1$. Finally set $S_2 = \{s \cup \{\nu(\psi\neg\neg) : \psi \in \Delta, s \in S_1\}$. It is again easily observed that $S_2$ is a consistency property. By the Model Existence Theorem there is a structure $\mathfrak{A}$ such that $\mathfrak{A} \models \nu(\psi\neg\neg)$ for all $\psi \in \Delta$ and $\mathfrak{A} \models \nu(\varphi\neg)$. Thus not $\Delta \models \varphi$.

For $\Delta$ a countable set of box formulas and $\varphi$ a diamond formula we have the following easy corollaries:

COROLLARY 4.2. *$\Delta \models \varphi$ iff for some finite $\Delta_0 \subseteq \Delta$ $\Delta_0 \models \varphi$.*

COROLLARY 4.3. *$\Delta$ has a model iff every finite subset of $\Delta$ has a model.*

We conclude with the following interpolation theorem:

COROLLARY 4.4. *If $\varphi$ is a box formula, $\psi$ a diamond formula such that $\models \varphi \to \psi$, then there is a first-order formula $\chi$ such that $\models \varphi \to \chi$ and $\models \chi \to \psi$.*

*Proof.* By assumption, the diamond formula $\varphi\neg \vee \psi$ is universally valid; thus $\vdash \varphi\neg \vee \psi$ and we proceed by induction on the length of proof.

If $\varphi\neg \vee \psi$ is an axiom, then it is a first-order formula and the result is trivially true. Let us assume as another case that $\varphi\neg \vee \psi$ is derived by an application of the conjunction rule, i.e., $\psi = \psi_1 \wedge \psi_2$ and $\varphi\neg \vee \psi_1$, $\varphi\neg \vee \psi_2$ appear earlier in the proof. Since $\psi_i$ are again box formulas we apply the induction hypothesis to obtain first-order formulas $\chi_1, \chi_2$ such that $\models \varphi \to \chi_1$, $\models \chi_1 \to \psi_1$ and $\models \varphi \to \chi_2$, $\models \chi_2 \to \varphi_2$ from which we infer $\models \varphi \to \chi_1 \wedge \chi_2$ and $\models \chi_1 \wedge \chi_2 \to \psi$. The conjunction rule might also have been applied in the following way: $\varphi = \varphi_1 \vee \varphi_2$ and $\varphi_1\neg \vee \psi$ and $\varphi_2\neg \vee \psi$ appear earlier in the proof. By induction hypothesis there are first-order sentences $\chi_1, \chi_2$ such that $\models \varphi_1 \to \chi_1$, $\models \chi_1 \to \psi$, $\models \varphi_2 \to \chi_2$, $\models \chi_2 \to \psi$, from which $\models \varphi \to \chi_1 \vee \chi_2$ and $\models \chi_1 \vee \chi_2 \to \psi$ follow.

Let us comment on one further case in the induction on the length of proof

for $\varphi\neg \vee \psi$, on the iteration rule, leaving the remaining easy verifications to the reader. Thus we assume $\psi = \langle \alpha^* \rangle \psi_0$ and $\varphi\neg \vee \psi$ is obtained from $\varphi\neg \vee \langle \alpha \rangle^n \psi_0$ by an application of the the iteration rule. The induction hypothesis yields a first-order sentence $\chi$ such that $\models \varphi \to \chi$ and $\models \chi \to \langle \alpha \rangle^n \psi_0$, but this also gives $\models \chi \to \langle \alpha^* \rangle \psi_0$. The iteration rule can also be applied in case $\varphi \models [\alpha^*] \varphi_0$. Here we get by induction hypothesis a first-order sentence $\chi$ satisfying $\models [\alpha]^n \varphi_0 \to \chi$ and $\models \chi \to \varphi$ which implies $\models [\alpha^*] \varphi_0 \to \chi$.

RECEIVED May 14, 1984; ACCEPTED July 17, 1984

## REFERENCES

ENGELER, E. (1967), Algorithmic properties of structures, *Math. Systems Theory* **1**, 183–195.

HAREL, D. (1979), First-Order dynamic Logic, "Lecture Notes in Computer Sciences," Vol. 68, Springer-Verlag, Berlin/Heidelberg/New-York.

HAREL, D., MEYER, A. R., AND PRATT, V. R. (1977), Computability and completeness in Logics of Programs," *in* "Proc. 9th ACM Sympos. Theory of Computing, 261–268.

HAREL, D. (1982), Dynamic logic, *in* the "Handbook of Philosophical Logic," to appear.

KEISLER, J. (1971), Model theory for infinitary logic, "Studies in Logic and the Foundations of Mathematics," Vol. 62, North-Holland, Amsterdam/London.

MEYER, A. R., AND HALPERN, J. Y. (1982), Axiomatic definitions of programming languages: A theoretical assessment, *J. Assoc. Comput. Mach.* **29**, 555–576.