Adding the Everywhere Operator to Propositional Logic

David Gries [*] and Fred B. Schneider [†]
Computer Science Department, Cornell University
Ithaca, New York 14853 USA

May 1, 1996

Abstract

A sound and complete modal propositional logic S5c is presented in which $\square P$ has the interpretation " P is *true* in all states". S5c illustrates that the two following approaches to axiomatizing a logic need not be equivalent: (i) give axiom schemes that denote an infinite number of axioms and (ii) write a finite number of axioms in terms of propositional variables and introduce a substitution inference rule.

1 Introduction

Logic gives a syntactic way to derive or certify truths that can be expressed in the language of the logic. The expressiveness of the language impacts the logic's utility —the more expressive the language, the more useful the logic (at least if the intended use is to prove theorems, as opposed to, say, studying logics). We wish to calculate with logic, much as one calculates in algebra to derive equations, and we find it useful for $\square P$ to be a formula of the logic and have the meaning " P is *true* in all states".

When a propositional logic extended with $\square$ is further extended to predicate logic and then to other theories, the logic can be used for proving theorems that could otherwise be handled only at the metalevel, and most likely informally. For example, the statement

(1) P is valid iff $\forall x.P$ is valid

[*] Supported by NSF grant CDA-9214957 and ARPA/ONR grant N00014-91-J-4123.
[†] Supported in part by ARPA/NSF grant CCR-9014363, NASA/ARPA grant NAG-2-893, and AFOSR grant F49620-94-1-0198.

Table 1: Table of abbreviations

$\alpha \wedge \beta$:	$\neg(\neg\alpha \vee \neg\beta)$	$true$:	$p \equiv p$
$\alpha \Rightarrow \beta$:	$\neg\alpha \vee \beta$	$false$:	$\neg true$
$\alpha \equiv \beta$:	$(\alpha \Rightarrow \beta) \wedge (\beta \Rightarrow \alpha)$	$\diamond\alpha$:	$\neg\Box\neg\alpha$

is formalized in our logic as $\Box P \equiv \Box(\forall x.P)$. In contrast, formalizing (1) as the two inference rules

$$\vdash P \longrightarrow \vdash \forall x\,.\,P \qquad \text{and} \qquad \vdash \forall x\,.\,P \longrightarrow \vdash P$$

demotes it to a meta-logical notion. When the equivalence of P and $\forall x.P$ is not expressible by a formula of the logic, it is not directly available for use in calculational reasoning.

As another example, the following fact about set theory, [1]

$$\{x \mid Q\} = \{x \mid R\} \text{ is valid} \quad \text{iff} \quad Q \equiv R \text{ is valid} \quad ,$$

is formalized using $\Box$ as

$$\Box(\{x \mid Q\} = \{x \mid R\}) \quad \equiv \quad \Box(Q \equiv R) \quad ,$$

but it cannot be formalized as a formula without something like $\Box$.

The use of the everywhere operator $\Box P$ was introduced by Dijkstra (using the notation $[P]$) in the early 1980's (see e.g. [3, 2]) and has since been used by researchers in the formal development of programs. It can be used to formalize the Hoare triple $\{P\}\ S\ \{Q\}$, with the meaning "execution of statement S begun in a state in which P is $true$ is guaranteed to terminate in a state in which Q is $true$." [7]. Using weakest-precondition predicate transformer wp [1], we define:

$$\{P\}\ S\ \{Q\} : \quad \Box(P \Rightarrow wp(S, Q)) \quad .$$

Modal logic [2] S5 includes $\Box P$ among its formulas. However, S5 is not complete with respect to the model consisting of all states (total functions from the set of all propositional variables to $\{\mathbf{t}, \mathbf{f}\}$, with the conventional definition of evaluation), where every state is accessible from every other state. For example, the formula $\neg\Box p$ for p a propositional variable is valid with respect this model, but it is not a theorem of S5.

In this paper, we extend S5 to an axiomatization S5c that is sound and complete with respect to this model. We first write S5c using axiom schemes that represent an infinite number of axioms. Later, we present a logic that has a finite number of axioms.

[1] We use $\equiv$ for equality over the booleans and $=$ for equality over any type.
[2] See Hughes and Cresswell [8] for an introduction to modal logic.

2 Preliminaries

Let VP be a set of propositional variables. We use lower-case letters $p, q, r, \ldots$ for elements of VP. A *formula* of S5 has one of the following forms (p is any variable in VP, and metavariables α, β stand for formulas).

(2) $\quad p \qquad (\neg\alpha) \qquad (\alpha \vee \beta) \qquad (\Box\alpha)$

In addition, $(\alpha \wedge \beta)$, $(\alpha \Rightarrow \beta)$, $(\alpha \equiv \beta)$, $\Diamond\alpha$, *true*, and *false* are abbreviations of certain formulas, as shown in Table 1. (Operator $\Diamond$ is read as "possibly" or "somewhere".) Precedences eliminate the need for some parentheses; prefix operators $\neg$, $\Box$, and $\Diamond$ bind tightest, then $\vee$ and $\wedge$, then $\Rightarrow$, and finally $\equiv$.

A formula of S5 that contains neither $\Box$ nor $\Diamond$ is called a *propositional formula.*

A *model* is a triple $\langle W, R, V\rangle$ in which:

- W is a nonempty set of *worlds*.
- R is an *accessibility relation*, a binary relation over W: $w\ R\ u$ signifies that *world* u *is accessible from world* w.
- $V(\alpha, w)$, for α a formula and w a world in W, is a *value assignment* that satisfies the following properties:

$$
\begin{array}{lll}
(3) \quad V(p,w) & \text{is either } \mathbf{t} \text{ or } \mathbf{f} & (\text{for } p \text{ a variable in } VP), \\
V(\neg\alpha,w) & = & \mathbf{if}\ V(\alpha,w)=\mathbf{t}\ \mathbf{then}\ \mathbf{f}\ \mathbf{else}\ \mathbf{t}\ , \\
V(\alpha \vee \beta,w) & = & \mathbf{if}\ V(\alpha,w)=\mathbf{t}\ \mathbf{then}\ \mathbf{t}\ \mathbf{else}\ V(\beta,w)\ , \\
V(\Box\alpha,w) & = & \mathbf{if}\ V(\alpha,u)=\mathbf{t} \text{ for all worlds } u \text{ such that } w\ R\ u\ \mathbf{then}\ \mathbf{t}\ \mathbf{else}\ \mathbf{f}\,.
\end{array}
$$

An S5-*model* is a model $\langle W, R, V\rangle$ in which R is an equivalence relation —reflexive, transitive, and symmetric. An S5-formula α is S5-*valid*, written $\models_{S5} \alpha$, iff for every S5-model $\langle W, R, V\rangle$ and every w in W, $V(\alpha, w) = \mathbf{t}$. ($\models_L$ has lowest precedence —it applies to the longest formula that follows it.)

The first part of Table 2 is a schematic presentation of propositional logic PM. PM consists of one inference-rule scheme and four axiom schemes. The inference-rule scheme denotes the infinite set of inference rules constructed by replacing metavariables α and β by formulas. (Similarly for the axiom schemes). In PM (as in all the logics in this paper), a *theorem* is either an axiom or the conclusion of an inference rule whose premises are theorems. We use the notation $\vdash_L \alpha$ for " α is a theorem of logic L". ($\vdash_L$ has lowest precedence; it applies to the longest formula that follows it.)

The second part of Table 2 extends propositional logic PM to modal logic S5, by adding one inference rule and three axioms. S5 is sound and complete with respect to S5-validity [8].

Table 2: Schematic PM, S5, and S5c

PM:	**Modus Ponens:** $\vdash \alpha,\ \vdash (\alpha \Rightarrow \beta) \quad \longrightarrow \ \vdash \beta$
	Axiom scheme A1: $\alpha \vee \alpha \Rightarrow \alpha$
	Axiom scheme A2: $\alpha \Rightarrow \alpha \vee \beta$
	Axiom scheme A3: $\alpha \vee \beta \Rightarrow \beta \vee \alpha$
	Axiom scheme A4: $(\beta \Rightarrow \gamma) \Rightarrow (\alpha \vee \beta \Rightarrow \alpha \vee \gamma)$
S5:	**Necessitation:** $\vdash \alpha \quad \longrightarrow \ \vdash \Box\alpha$
	Axiom scheme $\Box$-Instantiation: $\Box\alpha \Rightarrow \alpha$
	Axiom scheme Monotonicity: $\Box(\alpha \Rightarrow \beta) \Rightarrow (\Box\alpha \Rightarrow \Box\beta)$
	Axiom scheme Necessarily Possible: $\Diamond\alpha \Rightarrow \Box\Diamond\alpha$
S5c:	**Textual Substitution:** $\vdash \gamma \quad \longrightarrow \ \vdash \gamma[v := \beta]$

3 Logic S5c

The intended model for S5c is the set of all states, where a state associates a value $\mathbf{t}$ or $\mathbf{f}$ with each propositional variable and each state is accessible from all the others. In this model, $\Box\alpha$ has the interpretation "α is true in all states".

Let $\widehat{W}$ be the set of all total functions $w : VP \to \{\mathbf{t}, \mathbf{f}\}$. Let $\widehat{R}$ be the universal relation over $\widehat{W}$, i.e. $w\ \widehat{R}\ u$ holds for all w, u in $\widehat{W}$. Let $\widehat{V}$ be the value assignment (i.e. it satisfies (3)) defined by $\widehat{V}(p, w) = w.p$.[3] Then, $\langle \widehat{W}, \widehat{R}, \widehat{V} \rangle$ is the (only) S5c-model, and a formula α is S5c-valid iff for every w in $\widehat{W}$, $\widehat{V}(\alpha, w) = \mathbf{t}$. The S5c-model is also an S5-model, since $\widehat{R}$ is an equivalence relation. Consequently, $\models_{\mathrm{S5}} \alpha$ implies $\models_{\mathrm{S5c}} \alpha$.

Logic S5 is sound but not complete with respect to S5c-validity. To see this, consider the formula $\neg\Box p$. It is not S5-valid, since it evaluates to $\mathbf{f}$ in the S5-model $(\{w\}, I, V)$, where $V(p, w) = \mathbf{t}$ and I is the identity relation. Since S5 is sound, $\neg\Box p$ is not a theorem of S5. However, $\models_{\mathrm{S5c}} \neg\Box p$ holds, since $\Box p$ evaluates to $\mathbf{f}$ —there is a world w in $\widehat{W}$ that satisfies $\widehat{V}(p, w) = \mathbf{f}$.

We define *textual substitution* $\gamma[v := \beta]$ where v is a propositional variable and γ and β are formulas. The definition treats occurrences of variables in $\Box\alpha$ as if they were bound.

$$
\begin{array}{lcll}
v[v := \beta] & = & \beta & \\
w[v := \beta] & = & w & \text{(for variable } w \text{ different from } v\text{)} \\
(\neg\gamma)[v := \beta] & = & \neg(\gamma[v := \beta]) & \\
(\delta \circ \gamma)[v := \beta] & = & \delta[v := \beta] \circ \gamma[v := \beta] & \text{(for binary connective } \circ\text{)} \\
(\Box\gamma)[v := \beta] & = & \Box\gamma &
\end{array}
$$

We allow simultaneous textual substitution, by letting v and β be lists of distinct propositional variables and formulas, respectively. The formal definition is left to the reader.

[3] For w a one-parameter function, $w.x$ denotes the application of w to argument x.

The third part of Table 2 extends logic S5 with inference rule scheme Textual Substitution to yield logic S5c. It is easy to show that Textual Substitution preserves S5c-validity. Note that only a propositional variable, and not a metavariable, may be replaced using it.

Textual Substitution and all the inference rules of S5 preserve S5c-validity. Also, the axioms of S5 are S5c-valid. Therefore, logic S5c is sound with respect to S5c-validity.

To illustrate S5c, we prove that $\neg\Box p$ is a theorem. We use a calculational style of proof —see [4] or [5]. The first formula is an S5c-theorem. Since the last formula equals the first, the last is also an S5c-theorem.

$$
\begin{array}{ll}
& (\Box p \;\Rightarrow\; p)[p := \mathit{false}] \quad \text{—Textual Substitution in Axiom } \Box\text{-Instantiation} \\
= & \quad \langle \text{Definition of textual substitution for propositional variable } p \rangle \\
& \Box p \;\Rightarrow\; \mathit{false} \\
= & \quad \langle \text{Propositional theorem } Q \;\Rightarrow\; \mathit{false} \;\equiv\; \neg Q \text{, with } Q := \Box p \rangle \\
& \neg\Box p
\end{array}
$$

In the rest of this section, we prove that S5c is complete with respect to S5c-validity. Since S5c is an extension of S5, in this proof, we can use S5-theorems presented in Hughes and Cresswell [8] as S5c-theorems. Also, we rely on the following lemma, which follows directly from the definition of $\widehat{V}(\Box\alpha, w)$.

(4) **Lemma.** For formula α , either $\models_{\mathrm{S5c}} \Box\alpha$ or $\models_{\mathrm{S5c}} \neg\Box\alpha$.

Hughes and Cresswell define *ordered modal conjunctive normal form* (ordered MCNF). A formula is in ordered MCNF if it has the form $C_1 \wedge \ldots \wedge C_n$ and each C_i has the form

(5) $\beta \vee \Box\gamma_1 \vee \ldots \vee \Box\gamma_m \vee \Diamond\delta$,

where β , the γ_i , and δ are propositional formulas (i.e. they don't contain $\Box$ or $\Diamond$).

We prove three lemmas, leading up to a proof that $\models_{\mathrm{S5c}}$ (5) implies $\vdash_{\mathrm{S5c}}$ (5) .

(6) **Lemma.** For propositional formula β , $\models_{\mathrm{S5c}} \beta$ implies $\vdash_{\mathrm{S5c}} \beta$.

Proof. Suppose $\models_{\mathrm{S5c}} \beta$. Since β is a propositional formula, and since S5c contains complete propositional logic PM, $\vdash_{\mathrm{S5c}} \beta$. □

(7) **Lemma.** For propositional formula δ , $\models_{\mathrm{S5c}} \Diamond\delta$ implies $\vdash_{\mathrm{S5c}} \Diamond\delta$.

Proof. If $\models_{\mathrm{S5c}} \Diamond\delta$, there is a world w such that $\widehat{V}(\delta, w) = \mathbf{t}$. Thus, there is an assignment $[\overline{p} := \overline{c}]$ of constants $\overline{c}$ (each constant being *true* or *false*) to the propositional variables $\overline{p}$ of δ such that $\delta[\overline{p} := \overline{c}]$ evaluates to $\mathbf{t}$ in w . Since $\delta[\overline{p} := \overline{c}]$ contains no variables, it evaluates to $\mathbf{t}$ in all worlds and is valid. By Lemma (6), $\vdash_{\mathrm{S5c}} \delta[\overline{p} := \overline{c}]$.

The following calculational proof shows that $(\Box\neg\delta \;\Rightarrow\; \neg\delta)[\overline{p} := \overline{c}]$ is equivalent to $\Diamond\delta$. Further, since the first formula is a theorem (it is an instance of axiom $\Box$-Instantiation on which Textual Substitution is performed), the last formula is also a theorem. This establishes $\vdash_{\mathrm{S5c}} \diamond\delta$.

$$
\begin{array}{ll}
 & (\Box\neg\delta \ \Rightarrow\ \neg\delta)[\overline{p} := \overline{c}] \qquad \text{—Textual Substitution in } \Box\text{-Instantiation} \\
= & \quad \langle \text{Contrapositive; Double negation} \rangle \\
 & (\delta \ \Rightarrow\ \neg\Box\neg\delta)[\overline{p} := \overline{c}] \\
= & \quad \langle \text{Definition of textual substitution} \rangle \\
 & \delta[\overline{p} := \overline{c}] \ \Rightarrow\ \neg\Box\neg\delta \\
= & \quad \langle\, \delta[\overline{p} := \overline{c}] \ \equiv\ \mathit{true} \ \ (\text{since } \vdash_{\mathrm{S5c}} \delta[\overline{p} := \overline{c}]\,);\ \text{Abbreviation (see Table 1)} \rangle \\
 & \mathit{true} \ \Rightarrow\ \diamond\delta \\
= & \quad \langle \text{Left identity of } \Rightarrow \rangle \\
 & \diamond\delta
\end{array}
$$

□

(8) **Lemma.** For propositional formula γ , $\models_{\mathrm{S5c}} \Box\gamma$ implies $\vdash_{\mathrm{S5c}} \Box\gamma$.

Proof. If $\models_{\mathrm{S5c}} \Box\gamma$, then $\widehat{V}(\gamma, w) = \mathbf{t}$ in all worlds w . Hence, $\models_{\mathrm{S5c}} \gamma$. By Lemma (6), $\vdash_{\mathrm{S5c}} \gamma$. By inference rule Necessitation, $\vdash_{\mathrm{S5c}} \Box\gamma$. □

(9) **Theorem.** $\models_{\mathrm{S5c}}$ (5) implies $\vdash_{\mathrm{S5c}}$ (5) .

Proof. Suppose $\models_{\mathrm{S5c}}$ (5) . By Lemma (4), each of $\Box\gamma_i$ and $\diamond\delta$ (i.e. $\neg\Box\neg\delta$) evaluates to $\mathbf{f}$ in all worlds or to $\mathbf{t}$ in all worlds. The proof uses a three-case analysis: $\diamond\delta$ evaluates to $\mathbf{t}$, $\Box\gamma_i$ evaluates to $\mathbf{t}$ for some γ_i , and all $\Box\gamma_i$ and $\diamond\delta$ evaluate to $\mathbf{f}$ in all worlds.

Case $\diamond\delta$ evaluates to t in all worlds. Then $\models_{\mathrm{S5c}} \diamond\delta$ and, by Lemma (7), $\vdash_{\mathrm{S5c}} \diamond\delta$. Note that $\diamond\delta \Rightarrow (5)$ is of the form $P \Rightarrow P \vee Q$, which is a theorem of propositional logic PM, so $\vdash_{\mathrm{S5c}} \diamond\delta \Rightarrow (5)$. By Modus ponens, $\vdash_{\mathrm{S5c}}$ (5) .

Case $\Box\gamma_i$ evaluates to t in all worlds. The proof is similar to the proof of the previous case, using Lemma (8) instead of (7).

Case the $\Box\gamma_i$ and $\diamond\delta$ evaluate to f in all worlds. Since (5), i.e. $\beta \vee \Box\gamma_1 \vee \ldots \vee \Box\gamma_m \vee \diamond\delta$, is S5c-valid, β evaluates to true in all worlds, so $\models_{\mathrm{S5c}} \beta$. The rest of this proof is similar to the proof of the first case, using Lemma (6) instead of (7). □

Hughes and Cresswell [8, p. 117] prove the following theorem.

(10) **Ordered MCNF Theorem.** For any formula α , there exists an ordered MCNF formula $\mathit{mcnf}.\alpha$ such that $\vdash_{\mathrm{S5}} \alpha \ \equiv\ \mathit{mcnf}.\alpha$.

(11) **Corollary**. $\vdash_{\mathrm{S5c}} \alpha \ \equiv\ \mathit{mcnf}.\alpha$

(12) **Corollary**. $\vdash_{\mathrm{S5c}} \alpha$ iff $\vdash_{\mathrm{S5c}} \mathit{mcnf}.\alpha$

(13) **Corollary**. $\models_{\mathrm{S5c}} \alpha$ iff $\models_{\mathrm{S5c}} \mathit{mcnf}.\alpha$.

Corollary (11) holds because S5c is an extension of S5. Corollary (12) follows from (11), the definition of abbreviation $\equiv$, and Modus ponens. For Corollary (13), note that (11) together with the soundness of S5c yields $\models_{\mathrm{S5c}} \alpha \equiv \mathit{mcnf}.\alpha$ and use properties of $\widehat{V}$.

To prove completeness of S5c, we use the following properties of propositional logic (which is included in logic S5c) and the S5c model.

(14) $\vdash_{\mathrm{S5c}} \alpha \wedge \beta$ iff $(\vdash_{\mathrm{S5c}} \alpha)$ and $(\vdash_{\mathrm{S5c}} \beta)$

(15) $\models_{\mathrm{S5c}} \alpha \wedge \beta$ iff $(\models_{\mathrm{S5c}} \alpha)$ and $(\models_{\mathrm{S5c}} \beta)$

(16) **Theorem.** For any formula α, $\models_{\mathrm{S5c}} \alpha$ implies $\vdash_{\mathrm{S5c}} \alpha$.

Proof. $mcnf.\alpha$ has the form $C_1 \wedge \ldots \wedge C_n$ where each C_i has form (5). We have,

$$
\begin{array}{ll}
 & \models_{\mathrm{S5c}} \alpha \\
\text{iff} & \quad \langle (13) \text{ —where } mcnf.\alpha \text{ is } C_1 \wedge \ldots \wedge C_n \rangle \\
 & \models_{\mathrm{S5c}} C_1 \wedge \ldots \wedge C_n \\
\text{iff} & \quad \langle (15),\ n-1 \text{ times} \rangle \\
 & (\models_{\mathrm{S5c}} C_1) \text{ and } \ldots \text{ and } (\models_{\mathrm{S5c}} C_n) \\
\text{implies} & \quad \langle \text{Monotonicity of and , Theorem (9) } (n-1 \text{ times}) \rangle \\
 & (\vdash_{\mathrm{S5c}} C_1) \text{ and } \ldots \text{ and } (\vdash_{\mathrm{S5c}} C_n) \\
\text{iff} & \quad \langle (14),\ n-1 \text{ times} \rangle \\
 & \vdash_{\mathrm{S5c}} C_1 \wedge \ldots \wedge C_n \\
\text{iff} & \quad \langle (12) \text{ —where } mcnf.\alpha \text{ is } C_1 \wedge \ldots \wedge C_n \rangle \\
 & \vdash_{\mathrm{S5c}} \alpha
\end{array}
$$

□

4 S5c with a finite number of axioms

An axiomatization with a finite number of axioms is usually derived from one with axiom schemes by (i) replacing the metavariables in the axiom schemes with propositional variables and (ii) introducing inference rule Uniform Substitution to substitute formulas for propositional variables:

$$\text{Uniform Substitution: } \vdash \alpha \quad \longrightarrow \quad \vdash \alpha^v_\beta \quad .$$

Here, v is a propositional variable, α and β are metavariables, and the notation α^v_β denotes a copy of the formula denoted by α in which all occurrences (even those within the scope of $\Box$) of v are replaced by the formula denoted by β. (Simultaneous uniform substitution is allowed, in which case v would be a list of variables and β a list of formulas.)

This method for eliminating axiom schemes does not work in the case of Schematic S5c of Table 2, because Uniform Substitution does not preserve S5c-validity. For example, $\neg\Box p$ is S5c-valid (as proven earlier), but $(\neg\Box p)^p_{true}$, which is $\neg\Box true$, is not S5c-valid.

We obtain a sound axiomatization of S5c that has a finite number of axioms as follows. First, we extend language S5c to a language S5c$'$. The formulas of S5c$'$ include those of S5c; the original formulas of S5c will be called *concrete formulas*. Then, we give an axiomatization of S5c$'$ —using a finite number of axioms. Finally, we show that the theorems of S5c$'$ that are concrete are precisely the theorems of S5c.

Table 3: PM$'$, S5$'$, and S5c$'$

PM$'$:	**Uniform Substitution:** $\vdash \alpha \quad \longrightarrow \vdash \alpha^P_\beta$
	Modus Ponens: $\vdash \alpha, \vdash (\alpha \Rightarrow \beta) \quad \longrightarrow \vdash \beta$
	Axiom A1: $P \vee P \Rightarrow P$
	Axiom A2: $P \Rightarrow P \vee Q$
	Axiom A3: $P \vee Q \Rightarrow Q \vee P$
	Axiom A4: $(Q \Rightarrow R) \Rightarrow (P \vee Q \Rightarrow P \vee R)$
S5$'$:	**Necessitation:** $\vdash \alpha \quad \longrightarrow \vdash \Box\alpha$
	Axiom $\Box$-Instantiation: $\Box P \Rightarrow P$
	Axiom Monotonicity: $\Box(P \Rightarrow Q) \Rightarrow (\Box P \Rightarrow \Box Q)$
	Axiom Necessarily Possible: $\Diamond P \Rightarrow \Box\Diamond P$
S5c$'$:	**Textual Substitution:** $\vdash \gamma \quad \longrightarrow \vdash \gamma[v := \beta] \qquad (\gamma, \beta$ concrete)

Let VF be a new set of *formula variables*. We use upper-case letters $P, Q, R, \ldots$ for formula variables. Formulas of S5c$'$ are defined as in (2), except that a formula variable is also a formula. For example, $p \vee q$, $P \vee Q$, and $p \vee Q$ are formulas of S5c$'$.

Call a formula *concrete* if it does not contain a formula variable. For example, $p \vee q$ is concrete, but $P \vee Q$ and $p \vee Q$ are not concrete. Language S5c contains exactly the concrete formulas of S5c$'$. For a formula α, let $\underline{\alpha}$ denote the formula obtained by replacing every formula variable P of α by the corresponding propositional variable p.

We present an axiomatization for S5c$'$ in Table 3. Its axioms are those of S5c, except that metavariables have been replaced by formula variables. The inference rules of S5c$'$ include those of S5c (even to requiring that, in Textual Substitution, γ and β be concrete). Inference rule Uniform Substitution is included, but only for replacing formula variables.

One may view logic S5c$'$ as simulating metavariables by formula variables. Note that neither Uniform Substitution nor Textual Substitution can be used to derive a non-concrete theorem from a concrete theorem.

We wish to prove that S5c and S5c$'$ have the same concrete theorems. To this end, call a Hilbert-style S5c$'$ proof *concrete* iff the only non-concrete theorems in it are axioms. This implies that an axiom is used only as the premise of an instance of Uniform Substitution whose conclusion is concrete. For example, here is a concrete proof of $\neg p \vee (p \vee p) \Rightarrow \neg p \vee p$.

1.	$(Q \Rightarrow R) \Rightarrow (P \vee Q \Rightarrow P \vee R)$	Axiom A4
2.	$(p \vee p \Rightarrow p) \Rightarrow (\neg p \vee (p \vee p) \Rightarrow \neg p \vee p)$	Uniform Substitution
3.	$P \vee P \Rightarrow P$	Axiom A1
4.	$p \vee p \Rightarrow p$	Uniform Substitution
5.	$\neg p \vee (p \vee p) \Rightarrow \neg p \vee p$	Modus Ponens, 4, 2

In a concrete S5c$'$ proof, each theorem that is derived using a Uniform-Substitution inference is an axiom of S5c. Thus, the concrete S5c$'$ proof can be turned into an S5c proof simply by deleting each axiom and changing every hint "Uniform Substitution" to "Axiom". For example, the S5c proof corresponding to the above concrete proof is as follows.

1.	$(p \vee p \Rightarrow p) \;\Rightarrow\; (\neg p \;\vee\; (p \vee p) \;\Rightarrow\; \neg p \vee p)$	Axiom A4
2.	$p \;\vee\; p \;\Rightarrow\; p$	Axiom A1
3.	$\neg p \;\vee\; (p \vee p) \;\Rightarrow\; \neg p \vee p$	Modus Ponens, 2, 1

Further, the reverse transformation turns a S5c proof into a concrete S5c$'$ proof. Hence:

(17) **Theorem.** Every theorem of S5c has a concrete S5c$'$ proof, and every concrete theorem of S5c$'$ that has a concrete proof is a theorem of S5c.

We now prove the important

(18) **Theorem.** Every concrete theorem of S5c$'$ has a concrete S5c$'$ proof.

Proof. Consider an arbitrary concrete theorem α and a Hilbert-style proof for it. We prove by induction on the length of its proof that there exists a concrete proof of α. Since α is concrete, the proof uses at least one inference rule. Below, we consider the four possibilities for the last inference rule.

Textual Substitution, $\vdash \gamma \quad \longrightarrow \; \vdash \gamma[v := \beta]$. Hence, γ is concrete, and by the induction hypothesis, it has a concrete proof. Since the step $\vdash \gamma \quad \longrightarrow \; \vdash \gamma[v := \beta]$ does not introduce a non-concrete theorem, the result follows in this case.

Necessitation, $\vdash \alpha \quad \longrightarrow \; \vdash \Box\alpha$. Similar to the previous case.

Modus Ponens, $\vdash \gamma,\ \gamma \Rightarrow \alpha \quad \longrightarrow \; \vdash \alpha$. Thus, $\vdash \gamma$ and $\vdash \gamma \Rightarrow \alpha$, so by Uniform Substitution, $\vdash \underline{\gamma}$ and $\vdash \underline{\gamma} \Rightarrow \alpha$ (recall that α is already concrete). By the induction hypothesis, there are concrete proofs of $\underline{\gamma}$ and $\underline{\gamma} \Rightarrow \alpha$. Now use Modus Ponens, $\vdash \underline{\gamma},\ \underline{\gamma} \Rightarrow \alpha \quad \longrightarrow \; \vdash \alpha$, to complete a concrete proof of α.

Uniform Substitution, $\vdash \gamma \quad \longrightarrow \; \vdash \gamma^P_\beta$. Here, α is γ^P_β. In the left column of Table 4 are the five ways in which the last two steps of the proof could be written. We have abbreviated the names of inference rules by their initials, and we have listed either the premises or the numbers of lines on which the premises appear in the proof. In the right column, we give alternative concrete proofs —the formulas on lines with boldface numbers are concrete formulas for which the inductive hypothesis is assumed, so they have concrete proofs. Since the remaining lines of these proofs contain concrete formulas, the proofs are concrete. □

(19) **Corollary of Theorems (17), (18).** For concrete formula α, $\vdash_{\mathrm{S5c}} \alpha$ iff $\vdash_{\mathrm{S5c}'} \alpha$.

A bit more can be said about formulas of S5 and S5$'$. Call a formula *abstract* if it contains no propositional variables. Thus, $P \vee P$ is abstract. Consider any formula α that

Table 4: The five possible proofs of γ^P_β

(0) γ Axiom (1) γ^P_β U.S. (0)	The proof to the left is concrete
(0) $\delta[v := \sigma]$ T.S. δ (1) $(\delta[v := \sigma])^P_\beta$ U.S. (0)	**(0)** $\delta[v := \sigma]$ T.S. δ (Textual Substitution requires δ to be concrete, so substituting for P has no effect)
(0) $\Box\gamma$ N. γ (1) $(\Box\gamma)^P_\beta$ U.S. (0)	**(0)** γ^P_β U.S. γ (1) $\Box(\gamma^P_\beta)$ (i.e. $(\Box\gamma)^P_\beta$) N. (0)
(0) γ M.P. δ , $\delta \Rightarrow \gamma$ (1) γ^P_β U.S. (0)	**(0)** $\delta^{P,Q}_{\beta,q}$ U.S. δ **(1)** $(\delta \Rightarrow \gamma)^{P,Q}_{\beta,q}$ U.S. $\delta \Rightarrow \gamma$ (2) $\gamma^{P,Q}_{\beta,q}$ M.P. (0), (1)
(γ^P_β is concrete, so P contains the formula variables in γ . Let Q be a list of formula variables in δ except those in P and let q be a corresponding list of propositional variables. Then γ^P_β is concrete and $\gamma^{P,Q}_{\beta,q}$ is the same as γ^P_β .)	
(0) δ^Q_σ U.S. δ (1) $(\delta^Q_\sigma)^P_\beta$ U.S. (0)	**(0)** $\delta^{P,Q}_{\beta,\sigma^P_\beta}$ U.S. δ

does not contain both a formula variable P and the corresponding propositional variable p . Let $\overline{\alpha}$ denote the abstract formula obtained by changing all (lower-case) propositional variables to (upper-case) formula variables. For example, if α is $p \vee Q$, then $\overline{\alpha}$ is $P \vee Q$ and $\underline{\alpha}$ is $p \vee q$. The proof of the following theorem is left to the reader.

(20) **Theorem.** Let α be a formula that does not contain both a formula variable (e.g. P) and its propositional counterpart (e.g. p). Then the following are all equivalent: $\vdash_{\mathrm{S5}} \underline{\alpha}$, $\vdash_{\mathrm{S5'}} \overline{\alpha}$, $\vdash_{\mathrm{S5'}} \alpha$, $\vdash_{\mathrm{S5'}} \underline{\alpha}$, and $\models_{\mathrm{S5}} \underline{\alpha}$.

5 Discussion

It is possible to give a schematic axiomatization S5c without using an inference rule like Textual Substitution. Add as axioms to schematic S5 all formulas of the form $\Diamond\delta$ for δ a satisfiable propositional formula (i.e. a propositional formula that evaluates to $\mathbf{t}$ in at least one world of the S5c-model). Lemma (7) now holds trivially, and we can prove completeness with respect to S5c-validity in the same way that we proved completeness of

S5c. This general approach is mentioned in [6].

This axiomatization appeared unsatisfactory to us because it refers to the semantic concept of satisfiability. However, in private communications with us, both Rob Goldblatt and Joe Halpern explained how this semantic concept can be eliminated, leading to a complete (syntactic) axiomatization. A propositional formula is satisfiable iff its disjunctive normal form contains a disjunct that does not contain some literal together with its negation.

However, this solution is still unsatisfactory. It would require us, whenever we wish to prove a conjectured theorem $\diamond\delta$ (for δ a propositional formula), to put δ in disjunctive normal form. The introduction of inference rule Textual Substitution provides an alternative that is more in tune with the way we typically prove theorems.

For us, the interesting part of this exercise was the discovery that the following approaches to logic are not equivalent: (i) give axiom schemes and (ii) give a finite number of axioms in terms of propositional variables and introduce inference rule Uniform Substitution.

Acknowledgements

We thank Rutger Dijkstra, Rob Goldblatt, K. Rustan M. Leino, and members of the Eindhoven Tuesday Afternoon Club for their extremely helpful comments on drafts of this paper.

References

[1] Dijkstra, E.W. *A Discipline of Programming.* Prentice Hall, Englewood Cliffs, 1976.

[2] Dijkstra, E.W. The everywhere operator once more. EWD1086, 26 November 1990.

[3] Dijkstra, E.W., and C.S. Scholten. *Predicate Calculus and Program Semantics.* Springer Verlag, New York, 1990.

[4] Gries, D., and F.B. Schneider. *A Logical Approach to Discrete Math.* Springer-Verlag, New York, 1993.

[5] Gries, D., and F.B. Schneider. Equational propositional logic. *IPL 53* (1995), 145-152.

[6] Halpern, Y.J., and Kapron, B. Zero-one laws for modal logic. *Annals of Pure and Applied Logic 69* (1994), 157-193.

[7] C.A.R. Hoare. An axiomatic basis for computer programming. *Comm. ACM 12* (October 1969), 576-580, 583.

[8] Hughes, G.E., and M.J. Cresswell. *An Introduction to Modal Logic.* Mehuen and Co., New York, 1968.