Adding the Everywhere Operator to Propositional Logic

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

May 29, 1996

Abstract

Sound and complete modal propositional logic C is presented, in which $\Box P$ has the interpretation " P is *true* in all states". The interpretation is already known as the Carnapian extension of S5. A new axiomatization for C provides two insights. First, introducing an inference rule *textual substitution* allows seamless integration of the propositional and modal parts of the logic, giving a more practical system for writing formal proofs. Second, the two following approaches to axiomatizing a logic are shown to be not 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 $\Box P$ to be a formula of the logic and have the meaning " P is *true* in all states".

When a propositional logic extended with $\Box$ 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.

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

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

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

The use of the everywhere operator $\Box P$ was introduced to researchers in the formal development of programs by Dijkstra (using the notation $[P]$) in the early 1980's (see e.g. [5, 4]). 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*." [10]. Using weakest-precondition predicate transformer wp [3], we define:

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

Modal logic [2] S5 includes $\Box P$ among its formulas. As is well known, S5 is not complete with respect to model C, which consists 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.

A number of sound an complete axiomatizations for C are known [13, 2, 1, 11, 9], dating from as early as 1973 —see Gottlob' survey [6]. In Sec. 3, we give a new axiomatization for C, compare it with previous ones, and argue why we believe the new axiomatization is more suitable for actually writing formal proofs.

The axiomatization presented in Sec. 3 uses an infinite number of axioms, specified by a finite set of axiom schemes. In Sec. 4 we present an axiomatization C$'$ that has a finite number of axioms. We show that such a finite axiomatization cannot be obtained simply by replacing the metavariables of the axiom schemes of C of Sec. 3 by propositional variables and adding inference rule Uniform Substitution. This, then demonstrates that the two approaches to axiomatizing a logic are not necessarily equivalent.

[1] We use $\equiv$ for equality over the booleans and $=$ for equality over any type.
[2] See Hughes and Cresswell [12] for an introduction to modal logic.

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*
$\alpha \equiv \beta$:	$(\alpha \Rightarrow \beta) \wedge (\beta \Rightarrow \alpha)$	$\diamond\alpha$:	$\neg\Box\neg\alpha$

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

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:

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

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

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: $\Diamond\alpha \Rightarrow \Box\Diamond\alpha$
C:	**Textual Substitution:** $\vdash \gamma \quad \longrightarrow \ \vdash \gamma[v := \beta]$

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 [12].

3 Logic C

The intended model for C 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". We now define this model formally.

Let $\widehat{W}$ be the set of all total functions $w : VP \rightarrow \{\mathbf{t}, \mathbf{f}\}$. 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) C-model, and a formula α is C-valid iff for every w in $\widehat{W}$, $\widehat{V}(\alpha, w) = \mathbf{t}$. The C-model is also an S5-model, since $\widehat{R}$ is an equivalence relation. Consequently, $\models_{\mathrm{S5}} \alpha$ implies $\models_{\mathrm{C}} \alpha$.

Logic S5 is sound but not complete with respect to C-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, $\models_{\mathrm{C}} \neg\Box p$ 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.

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

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

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.

The third part of Table 2 extends logic S5 with inference-rule scheme Textual Substitution to yield logic C. It is easy to show that Textual Substitution preserves C-validity.

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

To illustrate C, we prove that $\neg\Box p$ is a theorem. We use a calculational style of proof —see [7] or [8]. The first formula is a C-theorem. Since the last formula equals the first, the last is also a C-theorem.

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

Proving completeness of logic C with respect to C-validity

We now prove that C is complete with respect to C-validity. Since C is an extension of S5, in this proof, we can use S5-theorems presented in Hughes and Cresswell [12] as C-theorems. Also, we rely on the following lemma, which follows directly from the definition of $\widehat{V}(\Box\alpha, w)$.

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

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 $\models_{\mathrm{C}}$ (5) implies $\vdash_{\mathrm{C}}$ (5).

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

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

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

Proof. If $\models_{\mathrm{C}} \diamond\delta$, 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), $\vdash_{\mathrm{C}} \delta[\overline{p} := \overline{c}]$.

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 $\vdash_{\mathrm{C}} \diamond\delta$.

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

□

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

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

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

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

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

Hughes and Cresswell [12, 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**. $\vdash_{\mathrm{C}} \alpha \equiv mcnf.\alpha$

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

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

Corollary (11) holds because C 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 C yields $\models_{\mathrm{C}} \alpha \equiv mcnf.\alpha$ and use properties of $\widehat{V}$.

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

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

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

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

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

	$\models_{\mathrm{C}} \alpha$
iff	$\langle$(13) —where $mcnf.\alpha$ is $C_1 \wedge \ldots \wedge C_n$ $\rangle$
	$\models_{\mathrm{C}} C_1 \wedge \ldots \wedge C_n$
iff	$\langle$(15), $n-1$ times$\rangle$
	$(\models_{\mathrm{C}} C_1)$ and $\ldots$ and $(\models_{\mathrm{C}} C_n)$
implies	$\langle$Monotonicity of and , Theorem (9) ($n-1$ times)$\rangle$
	$(\vdash_{\mathrm{C}} C_1)$ and $\ldots$ and $(\vdash_{\mathrm{C}} C_n)$
iff	$\langle$(14), $n-1$ times$\rangle$
	$\vdash_{\mathrm{C}} C_1 \wedge \ldots \wedge C_n$
iff	$\langle$(12) —where $mcnf.\alpha$ is $C_1 \wedge \ldots \wedge C_n$ $\rangle$
	$\vdash_{\mathrm{C}} \alpha$ $\square$

Comparison with earlier complete axiomatizations

As mentioned in Sec. 1, a number of complete axiomatizations of C have been given [13, 2, 1, 11, 9]. All of them are similar in nature to the following one, which we take from [9]. Begin with Schematic S5 (see Table 2). Instead of adding inference rule Textual Substitution, add as axioms 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 model C). Lemma (7) now holds trivially, and we can prove completeness with respect to C-validity in the same way that we proved completeness of C.

This axiomatization is unsatisfactory to us because it refers to the semantic notion of satisfiability. However, this semantic notion 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.

Hence, to discover whether $\diamond\delta$ (for δ a propositional formula) is a theorem, convert δ to disjunctive normal form and determine whether one of its disjuncts contains a literal and its negation. (Private communications with Rob Goldblatt and Joe Halpern).

The resulting axiomatization is still unsatisfactory to us, because of the need to reformulate δ of a conjectured theorem $\diamond\delta$ in disjunctive normal form. This reformulation is not in keeping with our usual way of proving theorems (using a calculational approach, where suitable [7, 8, 5]). For example, to prove $\diamond\delta \vee \diamond\gamma$, we would be forced to prove that one of δ and γ were satisfiable, rather than simply performing syntactic manipulations to obtain $\diamond\delta \vee \diamond\gamma$, as is our preference. Inference rule Textual Substitution provides an alternative that is more in tune with the way we prove theorems calculationally; it allows for a more seamless integration of proofs of the various kinds of theorems of C.

4 C 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 an inference rule to substitute formulas for propositional variables:

(17) $\vdash \alpha \quad \longrightarrow \vdash \alpha^v_\beta \quad .$

Here, α is a metavariable, v is a list of propositional variables, and β is a corresponding list of metavariables. 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 the variables of v are replaced by the formulas denoted by the corresponding variables of β.

This method for eliminating axiom schemes does not work in the case of Schematic C of Table 2, because (17) does not preserve C-validity. For example, $\neg\Box p$ is C-valid (as proven earlier), but $(\neg\Box p)^p_{true}$, which is $\neg\Box true$, is not C-valid.

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

Let VF be a new set of *formula variables.* We use upper-case letters $P, Q, R, \ldots$ for formula variables. Formulas of C′ 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 C′.

A formula of C′ is 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 C contains exactly the concrete formulas of C′. 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.

An axiomatization for C′ is given in Table 3. Its axioms are those of C, except that metavariables have been replaced by formula variables. The inference rules of C′ include

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

PM$'$:	**Uniform Substitution:** $\vdash \alpha \quad \longrightarrow \quad \vdash \alpha^{\gamma}_{\beta}$ $\quad$ (γ a list of formula variables)
	Modus Ponens: $\vdash \alpha, \vdash (\alpha \Rightarrow \beta) \quad \longrightarrow \quad \vdash \beta$
	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:** $\vdash \alpha \quad \longrightarrow \quad \vdash \Box\alpha$
	Axiom $\Box$-Instantiation: $\Box P \Rightarrow P$
	Axiom Monotonicity: $\Box(P \Rightarrow Q) \Rightarrow (\Box P \Rightarrow \Box Q)$
	Axiom Necessarily Possible: $\Diamond P \Rightarrow \Box\Diamond P$
C$'$:	**Textual Substitution:** $\vdash \gamma \quad \longrightarrow \quad \vdash \gamma[v := \beta]$ $\quad$ (γ, β concrete)

those of C (even to requiring that, in Textual Substitution, γ and β be concrete). The new inference rule Uniform Substitution is used only for replacing formula variables: α^{γ}_{β} denotes a copy of the formula denoted by α in which all occurrences (even those within the scope of $\Box$) of the formula variables in list γ are replaced by the formulas denoted by the corresponding variables of β.

One may view logic C$'$ 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 C and C$'$ have the same concrete theorems. To this end, call a Hilbert-style C$'$ 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 C$'$ proof, each theorem that is derived using a Uniform-Substitution inference is an axiom of C. Thus, the concrete C$'$ proof can be turned into an C proof simply by deleting each axiom and changing every hint "Uniform Substitution" to "Axiom". For example, the C 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 C proof into a concrete C′ proof. Hence:

(18) **Theorem.** Every theorem of C has a concrete C′ proof, and every concrete theorem of C′ that has a concrete proof is a theorem of C.

We now prove the important

(19) **Theorem.** Every concrete theorem of C′ has a concrete C′ 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, $\vdash \gamma \quad \longrightarrow \vdash \gamma^P_\beta$ for P a list of formula variables. 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. □

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

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 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.

(21) **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 $\models_{\mathrm{S5}} \underline{\alpha}$.

Table 4: The five possible proofs of γ^P_β

(0) γ (1) γ^P_β	Axiom U.S. (0)	The proof to the left is concrete	
(0) $\delta[v := \sigma]$ (1) $(\delta[v := \sigma])^P_\beta$	T.S. δ U.S. (0)	**(0)** $\delta[v := \sigma]$ (Textual Substitution requires δ to be concrete, so substituting for P has no effect)	T.S. δ
(0) $\Box\gamma$ (1) $(\Box\gamma)^P_\beta$	N. γ U.S. (0)	**(0)** γ^P_β (1) $\Box(\gamma^P_\beta)$ (i.e. $(\Box\gamma)^P_\beta$)	U.S. γ N. (0)
(0) γ (1) γ^P_β	M.P. δ , $\delta \Rightarrow \gamma$ U.S. (0)	**(0)** $\delta^{P,Q}_{\beta,q}$ **(1)** $(\delta \Rightarrow \gamma)^{P,Q}_{\beta,q}$ (2) $\gamma^{P,Q}_{\beta,q}$	U.S. δ U.S. $\delta \Rightarrow \gamma$ M.P. (0), (1)
(γ^P_β is concrete, so P contains all 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 (1) $(\delta^Q_\sigma)^P_\beta$	U.S. δ U.S. (0)	**(0)** $\delta^{P,Q}_{\beta,\sigma^P_\beta}$	U.S. δ

Acknowledgements

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

References

[1] Carroll, M.J. An axiomatization of S13. *Philosophia, Philosophical Quarterly of Israel 8*, 2–3 (November 1978), 381–382.

[2] Cocciarella, N.B. Logical atomism and modal logic. *Philosophia, Philosophical Quarterly of Israel 4*, 1 (1974), 41–66.

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

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

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

[6] Gottlob, G. Survey on a Carnapian extension of S5. Technical report, Institut für Informationssysteme, Technische Universität Wien, 1996.

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

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

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

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

[11] Hendry, H.E., and M.L. Pokriefka. Carnapian extensions of S5. *J. Philosophical Logic 14* (1985), 111-128.

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

[13] Thomason, S.K. A new representation of S5. *Notre Dame J. Formal Logic XIV*, 2 (April 1973), 281–284.