AXIOMATIZATION OF COMPUTER-ORIENTED MODEL LOGIC AND DECISION PROCEDURE

Sawamura, Hajime
Fundamental Information Unit, International Institute for Advanced Study of Social Information Science (IIAS-SIS), Fujitsu Ltd

https://doi.org/10.5109/13367

出版情報：Bulletin of informatics and cybernetics. 21 (3/4), pp.57-66, 1985-03. Research Association of Statistical Sciences
バージョン：
権利関係：

Bulletin of Informatics and Cybernetics Vol. 21, No. 3~4, 1985

AXIOMATIZATION OF COMPUTER-ORIENTED MODAL LOGIC AND DECISION PROCEDURE

By

Hajime SAWAMURA*

Abstract

Modal logic has various applications in Computer Science. In this paper, an axiomatization of computer-oriented modal logic has been given through the modification of Gentzen-type axiomatization of modal propositional logic S4. The characterization theorems of the axiom system, i. e, soundness, decidability were also proved. Finally, some provable and unprovable examples are shown together with the computer implementation of our proof procedure which is now being developed.

1. Introduction

The main roles of logics in computer science are to provide a language with high expressive power in which the object to be investigated is appropriately described, and to provide an inference mechanism. Among various logics, intensional logic such as modal logic has specific applications in computational linguistics (e.g., [1]), programming language semantics (e. g., [2]) and data semantics (e. g., [3]). The objects in these fields are sentences in natural language, programs and data respectively. Since meanings of these objects generally vary with time or situation, it is difficult or even impossible to capture such an aspect within the framework of extensional logic. Intensional logic, on the other hand, allows us to express time or situation dependency of the meanings of objects in these cases.

In spite of these significances of intensional logic, there is not so much work done about automated proof procedure for it.

In this paper, we investigated a proof procedure for propositional modal logic S4 with modalities, necessity and possibility, which is considered to be best suited for computer implementation. Before obtaining the proof procedure, we first give an axiomatization of computer-oriented propositional modal logic S4, which can be derived from the Gentzen formalism for logical system [4]. Our formalism of propositional modal logic S4 is simplified one of Ohnishi and Matsumoto [5] in which Gentzen-type axiomatization of modal logic was first given. More specifically, we employ the method of Rasiowa and Sikorski [6] who derived a variant of Gentzen formalism LK for first-order

* Fundamental Informatics Unit, International Institute for Advanced Study of Social Information Science (IIAS-SIS), Fujitsu Ltd, 140. Miyamoto, Numazu-shi, Shizuoka 410-03, Japan.

predicate calculus. In their formalism, axioms are allowed to include superfluous formulas, and the number of inference rules are reduced by transforming a sequent into the sequence consisting of formulas only. This type of formalism makes it easy to construct the decision procedure and to implement it by a computer without losing the naturalness of axiom system which was one of the Gentzen's intension for it.

The axiom system obtained by applying the method of Rasiowa and Sikorski for Ohnishi and Matsumoto's Gentzen-type axiom system for modal propositional logic S4, is called Sequence Calculus S4 (SC-S4 for short) in this paper. In Section 2, after introducing the propositional modal language and its semantics, we show the process of transforming the original Gentzen-type formalism into the corresponding Sequence Calculus and deductive equivalence of those axiom systems. In Section 3, we prove soundness theorem and decidability theorem for our Sequence Calculus. In Section 4, we briefly discuss about a computer implementation based on the decidability result, together with some provable and unprovable examples. In Section 5, conclusions and future research plans are included.

2. Sequence Calculus S4(SC-S4)

We first introduce an object language for propositional modal logic S4 and its semantics. Next, we construct an axiom system, Sequence Calculus, for it through the modification of Gentzen-type axiom system. In the remainder of this section, we prove that the both systems are deductively equivalent.

2.1. Language

In this paper we consider symbols and formulas of the following kind.

2.1.1. Symbols

1) Propositional variables: $P, Q, R, \cdots$.
2) Logical symbols: $\neg$(not), $\wedge$(and), $\Box$(necessary).
3) Auxiliary symbols:), (, , .

Greek letters $\alpha, \beta, \gamma, \cdots$ (with or without subscripts) serve as syntactical variables representing formulas.

2.1.2. Formulas

1) A propositional variable is a formula.
2) If α is a formula, so are $\neg\alpha$, $\Box\alpha$.
3) If α, β are formulas, so is $(\alpha\wedge\beta)$.

2.1.3. Defined symbols

$(\alpha\vee\beta)=\neg(\neg\alpha\wedge\neg\beta)$.
$(\alpha\supset\beta)=\neg(\alpha\wedge\neg\beta)$.
$(\alpha\equiv\beta)=(\alpha\supset\beta)\wedge(\beta\supset\alpha)$.
$\Diamond\alpha=\neg\Box\neg\alpha$.
$(\alpha\prec\beta)=\Box(\alpha\supset\beta)$.

2.2. Model

2.2.1. Kripke frame

An ordered pair $\langle W, R\rangle$ is called Kripke frame, where W is a non-empty set and R is a binary relation over W.

2.2.2. Valuation

A valuation V over $\langle W, R\rangle$ is a function which maps any ordered pair (α, w) with propositional variable α and $w \in W$ to either t or f. We extend the valuation for any formula α, and any $w \in W$ as follows:

1) $V(\neg\alpha, w)=t$ iff $V(\alpha, w)=f$.

2) $V(\alpha\wedge\beta, w)=t$ iff $V(\alpha, w)=t$ and $V(\beta, w)=t$.

3) $V(\Box\alpha, w)=t$ iff $V(\alpha, w')=t$ for every w' such that wRw'.

2.2.3. Kripke model

A Kripke model is an ordered triple $\langle W, R, V\rangle$, consisting of a Kripke frame $\langle W, R\rangle$ and a valuation V over it.

Kripke showed that various modal logics are well characterized in terms of binary relations R [7]. Propositional modal logic S4 is characterized by letting R be a reflexive and transitive relation over W.

2.2.4. Truth, validity

A formula α is true in a Kripke model $\langle W, R, V\rangle$ if $V(\alpha, w)=t$, for every $w \in W$. If a formula α is true in every Kripke model, it is valid and we write $\models\alpha$ to indicate that the formula α is valid.

2.3. Gentzen-type axiom system L-S4 for S4

2.3.1. Sequent

A sequent is an expression of the form $\alpha_1, \cdots, \alpha_n \rightarrow \beta_1, \cdots, \beta_m (n, m \geqq 0)$, where α_i, β_i represent any formula and $\rightarrow$ is an auxiliary symbol and not a logical symbol. Capital Greek letters $\Gamma, \Delta, \Theta, \Lambda, \Pi$ represent arbitrary (possibly empty) sequences of formulas separated by commas. $\neg\Gamma$, $\Box\Gamma$ represent sequences obtained by prefixing $\neg$, $\Box$ to all formulas of Γ respectively.

2.3.2. Axiom schema and inference schemata of L-S4

Axiom schema: $\alpha\rightarrow\alpha$

Inference schemata:

(Thinning) $$\frac{\Gamma\rightarrow\Theta}{\alpha, \Gamma\rightarrow\Theta}, \quad \frac{\Gamma\rightarrow\Theta}{\Gamma\rightarrow\Theta, \alpha};$$

(Contraction) $$\frac{\alpha, \alpha, \Gamma\rightarrow\Theta}{\alpha, \Gamma\rightarrow\Theta}, \quad \frac{\Gamma\rightarrow\Theta, \alpha, \alpha}{\Gamma\rightarrow\Theta, \alpha};$$

(Interchange) $$\frac{\Gamma, \alpha, \beta, \Delta\rightarrow\Theta}{\Gamma, \beta, \alpha, \Delta\rightarrow\Theta}, \quad \frac{\Gamma\rightarrow\Theta, \alpha, \beta, \Lambda}{\Gamma\rightarrow\Theta, \beta, \alpha, \Lambda};$$

(Cut) $$\frac{\Gamma\rightarrow\Theta, \alpha \quad \alpha, \Delta\rightarrow\Lambda}{\Gamma, \Delta\rightarrow\Theta, \Lambda};$$

($\neg$) $$\frac{\alpha, \Gamma\rightarrow\Theta}{\Gamma\rightarrow\Theta, \neg\alpha}, \quad \frac{\Gamma\rightarrow\Theta, \alpha}{\neg\alpha, \Gamma\rightarrow\Theta};$$

($\wedge$) $$\frac{\alpha, \beta, \Gamma\rightarrow\Theta}{\alpha\wedge\beta, \Gamma\rightarrow\Theta}, \quad \frac{\Gamma\rightarrow\Theta, \alpha \quad \Gamma\rightarrow\Theta, \beta}{\Gamma\rightarrow\Theta, \alpha\wedge\beta};$$

($\Box$) $$\frac{\alpha, \Gamma\rightarrow\Theta}{\Box\alpha, \Gamma\rightarrow\Theta}, \quad \frac{\Box\Gamma\rightarrow\alpha}{\Box\Gamma\rightarrow\Box\alpha}.$$

2.3.3. Provability of a sequent

We define the provability of a sequent recursively as follows:

1) An axiom is provable.

2) If $\frac{\Gamma\ (\Theta)}{\Delta}$ is one of the inference schemata and Γ(and Θ) is provable, then Δ is provable.

A formula α is said to be provable when a sequent $\rightarrow \alpha$ is provable and we write $\vdash\alpha$ to indicate that the formula α is provable.

THEOREM 1. (Cut Elimination Theorem) *L-S4 is deductively equivalent to L-S4 without cut rule* (*see* [2] *for the proof*).

Therefore, in what follows, we consider L-S4 without cut rule.

2.4. Construction of Sequence Calculus

By the left rule of ($\neg$) in L-S4 we can transpose all formulas from left to right, replacing each sequent $\alpha_1, \cdots, \alpha_n \rightarrow \beta_1, \cdots, \beta_m$ by an equivalent sequent $\rightarrow \beta_1, \cdots, \beta_m, \neg\alpha_1, \cdots, \neg\alpha_n$. The symbol $\rightarrow$ is then discarded. This simplifies a sequent to only a sequence of formulas $\gamma_1, \cdots, \gamma_k$. Furthermore, we identify a sequence of formulas Γ with simply a finite set Γ of formulas. Then, an axiom schema in Sequence Calculus SC-S4 is of the form $\Gamma, \alpha, \Delta, \neg\alpha, \Theta$. Inference schemata in SC-S4 are as follows:

(Thinning) $\quad \dfrac{\Gamma}{\Gamma, \alpha}$;

($\neg\neg$) $\quad \dfrac{\Gamma, \alpha}{\Gamma, \neg\neg\alpha}$;

($\wedge$) $\quad \dfrac{\Gamma, \alpha \quad \Gamma, \beta}{\Gamma, \alpha\wedge\beta}$;

($\neg\wedge$) $\quad \dfrac{\Gamma, \neg\alpha, \neg\beta}{\Gamma, \neg(\alpha\wedge\beta)}$;

($\Box$) $\quad \dfrac{\neg\Box\Gamma, \alpha}{\neg\Box\Gamma, \Box\alpha}$;

($\neg\Box$) $\quad \dfrac{\Gamma, \neg\alpha}{\Gamma, \neg\Box\alpha}$.

2.4.1. Provability of a sequence

Provability concept of a sequence in SC-S4 is the same as that of a sequent in L-S4.

THEOREM 2. *A sequent $\Gamma \rightarrow \Delta$ is provable in L-S4 iff a sequence $\neg\Gamma, \Delta$ is provable in SC-S4, in other words, the two axiom systems are deductively equivalent.*

PROOF. ($\Rightarrow$) This case is obvious from the construction of SC-S4.
($\Leftarrow$) The axiom $\Gamma, \alpha, \Delta, \neg\alpha, \Lambda$ is deducible as follows in L-S4:

$$\begin{array}{cl} \alpha\rightarrow\alpha & \\ \hline\hline \Gamma', \alpha\rightarrow\alpha, \Delta & \text{possibly several thinnings} \\ \hline \Gamma'\rightarrow\alpha, \Delta, \neg\alpha & \\ \hline\hline \Gamma'\rightarrow\alpha, \Delta, \neg\alpha, \Lambda & \\ \hline \rightarrow\Gamma, \alpha, \Delta, \neg\alpha, \Lambda & \text{, where } \Gamma=\neg\Gamma'. \end{array}$$

(Thinning):
$$\begin{array}{c} \to \Gamma \\ \hline \to \Gamma, \alpha \end{array}$$

$(\neg\neg)$:
$$\begin{array}{c} \to \Gamma, \alpha \\ \hline \neg\alpha \to \Gamma \\ \hline \to \Gamma, \neg\neg\alpha \end{array}$$

$(\wedge)$:
$$\begin{array}{c} \to \Gamma, \alpha \to \Gamma, \beta \\ \hline \to \Gamma, \alpha\wedge\beta \end{array}$$

$(\neg\wedge)$:
$$\begin{array}{c} \to \Gamma, \neg\alpha, \neg\beta \\ \hline \alpha, \beta \to \Gamma \\ \hline \alpha\wedge\beta \to \Gamma \\ \hline \to \Gamma, \neg(\alpha\wedge\beta) \end{array}$$

$(\Box)$:
$$\begin{array}{c} \to \neg\Box\Gamma, \alpha \\ \hline \Box\Gamma \to \alpha \\ \hline \Box\Gamma \to \Box\alpha \\ \hline \to \neg\Box\Gamma, \Box\alpha \end{array}$$

$(\neg\Box)$:
$$\begin{array}{c} \to \Gamma, \neg\alpha \\ \hline \alpha \to \Gamma \\ \hline \Box\alpha \to \Gamma \\ \hline \to \Gamma, \neg\Box\alpha \end{array}$$

Note that in the proof the double negation signs are freely eliminated since in L-S4, a sequent including double negations is provable if and only if the sequent obtained by eliminating double negations is provable.

3. Soundness, Completeness and Decidability

In this section, we describe the proofs of soundness and decidability theorems of SC-S4. With respect to the completeness of SC-S4, we rely on the completeness of L-S4 and the deductive equivalence of L-S4 and SC-S4.

3.1. Truth of a sequence

A sequence of formulas $\alpha_1, \cdots, \alpha_n$ is true in a Kripke model $\langle W, R, V\rangle$ if and only if for every $w \in W$ there exists some $\alpha_i (1 \leqq i \leqq n)$ such that $V(\alpha_i, w) = t$. If a sequence $\alpha_1, \cdots, \alpha_n$ is true in every Kripke model, then it is valid.

THEOREM 3. *SC-S4 is sound, i.e. for any sequence of formulas $\alpha_1, \cdots, \alpha_n$, if $\alpha_1, \cdots, \alpha_n$ is provable, then $\alpha_1, \cdots, \alpha_n$ is valid.*

PROOF. We only prove the validity-preserving property of the inference schemata $(\Box)$ and $(\neg\Box)$ since the propositional logic part is easy. We take any Kripke model $K = \langle W, R, V\rangle$.

(i) The validity of axiom schema is obvious.

(ii) The case where the inference schema is $(\Box)$: In the inference schema $\dfrac{\neg\Box\alpha_1, \cdots, \neg\Box\alpha_n, \beta}{\neg\Box\alpha_1, \cdots, \neg\Box\alpha_n, \Box\beta}$, we assume that the upper sequence $\neg\Box\alpha_1, \cdots, \neg\Box\alpha_n, \beta$ is true in K and the lower sequence $\neg\Box\alpha_1, \cdots, \neg\Box\alpha_n, \Box\beta$ is not valid. Then, for some

$w \in W$, we obtain,

$$V(\neg\Box\alpha_i,\ w)=f\,(i=1,\ \cdots,\ n) \qquad (1),$$

$$V(\Box\beta,\ w)=f \qquad (2).$$

From (2), we have, for some w' such that wRw', $V(\beta,\ w')=f$. From (1), since $V(\Box\alpha_i,\ w)=t\,(i=1,\ \cdots,\ n)$, at any w'' such that wRw'', we have $V(\Box\alpha_i,\ w'')=t\,(i=1,\ \cdots,\ n)$ using the transitivity of R. Therefore, for the w', we have $V(\Box\alpha_i,\ w')=t\,(i=1,\ \cdots,\ n)$ and $V(\beta,\ w')=f$, that is, $V(\neg\Box\alpha_i,\ w')=f\,(i=1,\ \cdots,\ n)$ and $V(\beta,\ w')=f$. This contradicts to the assumption. Since K is any model of SC-S4, the inference schema ($\Box$) is validity-preserving.

The case where the inference schema is ($\neg\Box$): In the inference schema $\dfrac{\alpha_1,\ \cdots,\ \alpha_n,\ \neg\beta}{\alpha_1,\ \cdots,\ \alpha_n,\ \neg\Box\beta}$, we assume the upper sequence $\alpha_1,\ \cdots,\ \alpha_n,\ \neg\beta$ is true in K and the lower sequence $\alpha_1,\ \cdots,\ \alpha_n,\ \neg\Box\beta$ is not valid. Then, we obtain, for some $w \in W$,

$$V(\alpha_i,\ w)=f\,(i=1,\ \cdots,\ n) \qquad (1),$$

$$V(\neg\Box\beta,\ w)=f \qquad (2).$$

From (2), we have $V(\Box\beta,\ w)=t$. Therefore, for all w' such that wRw', we have,

$$V(\beta,\ w')=t,$$

$$V(\neg\beta,\ w')=f.$$

By the reflexivity of R, letting w' be w, we have,

$$V(\alpha_i,\ w)=f\,(i=1,\ \cdots,\ n) \quad \text{and} \quad V(\neg\beta,\ w)=f.$$

This contradicts to the assumption. Hence, the inference schema ($\neg\Box$) is validity-preserving.

From the soundness theorem, it can be easily seen that SC-S4 is consistent, i. e. it is not true that both a sequence α consisting of a signle formula α and its negated sequence $\neg\alpha$ are provable.

3.2. Decision procedure for a sequence

The axiom system SC-S4 is convenient for designing a decision procedure for the system. It is because the upper sequence and the inference schema to be applied are uniquely determined except for only one schema (Thinning) when the lower sequence is given. The schema (Thinning) is not needed in the case of first-order predicate calculus whose axiom system is given as sequence calculus in this paper. However, by the existence of thinning rule in SC-S4, the resulting decision procedure turns out to be complex only in the point of combinatorial complexity.

3.2.1. Degree of a modal formula

We define the degree of a modal formulas as follows:

1) The degree of a propositional variable is 0.

2) If the degrees of formulas α and β are m and n respectivery, then the degree of $\neg\alpha$ is m, the degree of $\Box\alpha$ is $m+1$ and the degree of $\alpha\wedge\beta$ is $\max(m,\ n)$.

3.2.2. Decidability

THEOREM 4. *SC-S4 is decidable.*

PROOF. It is sufficient to show that the following procedure is a decision procedure for SC-S4. In the procedure, we assume that whenever a double negated formula appears, it is replaced by a formula without double negation sign. The procedure is described as ALGOL-like interation procedure with the help of natural language and mathematical notation. The symbols are used as follows:

seq_i denotes a sequence of formulas, i. e. a set of formulas;

p_i denotes a sets of sequences of formulas, e. g., $\{seq_1, \cdots, seq_m\}$, where $m \geqq 1$, representing the most upper sequences of formulas in the proof tree which is currently made backwards from the sequence to be proved;

P denotes a set, $\{p_1, \cdots, p_n\}$, where $n \geqq 1$, representing all possible candidates of proofs of a given sequence to be proveed.

Axiom (*seq*) denotes the predicate such that if *seq* is an axiom, then true, otherwise false.

Not-*Axiom* (*seq*) denotes the negated predicate of *Axiom* (*seq*).

```
Procedure DPSC-S4(P):
begin
1: if there exists a $p_i \in P = \{p_1, \cdots, p_n\}$ such that $Axiom\ (seq_j)$ for every $seq_j \in p_i$,
      then return ("PROVABLE");
   if there exists a $p_m \in P$ such that it does not contain any empty sequence in it, then
      begin
      choose such a $p_i \in P$ and a $seq_j \in p_i$;
      if $seq_j = \Gamma, \alpha \wedge \beta$ and $Not\text{-}Axiom\ (seq_j)$, then
         $P := (P - p_i) \cup \{(p_i - seq_j) \cup \{\{\Gamma, \alpha\}, \{\Gamma, \beta\}\}\}$;
      if $seq_j = \Gamma, \neg(\alpha \wedge \beta)$ and $Not\text{-}Axiom\ (seq_j)$, then
         $P := (P - p_i) \cup \{(p_i - seq_j) \cup \{\{\Gamma, \neg\alpha, \neg\beta\}\}\}$;
      if $seq_j = \neg\Box\Gamma, \Box\alpha$ and $Not\text{-}Axiom\ (seq_j)$, then
         $P := (P - p_i) \cup \{(p_i - seq_j) \cup \{\{\neg\Box\Gamma, \alpha\}\}\}$;
      if $seq_j = \Gamma, \neg\Box\alpha_1, \cdots, \neg\Box\alpha_t$ (where any formula in $\Gamma$ is not of the form $\neg\Box\alpha$)
         and $Not\text{-}Axiom\ (seq_j)$, then
         $P := (P - p_i) \cup \{(p_i - seq_j) \cup \{\{\Gamma, \neg\Box\alpha_1, \cdots, \neg\alpha_r, \cdots, \neg\Box\alpha_t\}\} \mid r = 1, \cdots, t\}$;
      if $seq_j = \alpha_1, \cdots, \alpha_s$ and $Not\text{-}Axiom\ (seq_j)$, then
         $P := (P - p_i) \cup \{(p_i - seq_j) \cup \{\{\alpha_1, \cdots, \alpha_s\} - \{\alpha_u\}\} \mid u = 1, \cdots, s\}$;
      go to 1;
      end
   return ("UNPROVABLE");
end
```

In the procedure. we use the clause "choose such a $p_i \in P$ and a $seq_j \in p_i$". We do not intend this as a nondeterministic step. It is just that it does not matter in what specific order the sets p_i and seq_j are maintained. We also omit the algorithms of *Axiom* and *Not-Axiom* since it is easy to construct the algorithms which test whether *seq* is an axiom or not.

This procedure terminates in a finite number of steps since the number of subformulas contained in the initial sequence to be proved are finite and by the applications of the inference shemata ($\Box$) and ($\neg\Box$), the degree n of a modal formula reduces to $n-1$.

4. Proof Examples

In this section, we list some proof examples. Some are provable examples, but others are unprovable in SC-S4. In the following proofs, justification for each line is indicated in the right margin.

EXAMPLE 1. The proof of a formula $\Box P\text{-}\!3\,\Box\Diamond\Box P$ is as follows; the left is a success and the right is a failure:

(success)

$$\begin{array}{cl} \neg\Box P,\ \Box P & \\ \hline \neg\Box P,\ \neg\Box\neg\Box P & (\neg\Box) \\ \hline \neg\Box P,\ \Box\neg\Box\neg\Box P & (\Box) \\ \hline \neg(\Box P\wedge\neg\Box\neg\Box\neg\Box P & (\neg\wedge) \\ \hline \Box\neg(\Box P\wedge\neg\Box\neg\Box\neg\Box P) & (\Box) \end{array}$$

(failure)

$$\begin{array}{cl} \neg P,\ \Box\neg\Box\neg\Box P & \\ \hline \neg\Box P,\ \Box\neg\Box\neg\Box P & (\neg\Box) \\ \hline \neg(\Box P\wedge\neg\Box\neg\Box\neg\Box P) & (\neg\wedge) \\ \hline \Box\neg(\Box P\wedge\neg\Box\neg\Box\neg\Box P) & (\Box) \end{array}$$

EXAMPLE 2. $\vdash\Box Q\supset\Box\Box Q\vee R$.

(success)

$$\begin{array}{cl} \neg\Box Q,\ \Box Q & \\ \hline \neg\Box Q,\ \Box\Box Q & (\Box) \\ \hline \neg\Box Q,\ \Box\Box Q,\ R & (\text{Thinning}) \\ \hline \neg(\Box Q\wedge\neg\Box\Box Q\wedge\neg R) & (\neg\wedge) \end{array}$$

(failure)

$$\begin{array}{cl} \neg Q,\ \Box\Box Q,\ R & \\ \hline \neg\Box Q,\ \Box\Box Q,\ R & (\neg\Box) \\ \hline \neg(\Box Q\wedge\neg\Box\Box Q\wedge\neg R) & (\neg\wedge) \end{array}$$

EXAMPLE 3. Unprovability of S5-aviom $\Diamond P\supset\Box\Diamond P$.

$$\begin{array}{cl} \Box\neg P & \\ \hline \Box\neg P,\ \Box\neg\Box\neg P & (\text{Thinning}) \\ \hline \neg(\neg\Box\neg P\wedge\neg\Box\neg\Box\neg P) & (\neg\wedge) \end{array}$$

EXAMPLE 4. Unprovability of Brouwerian axiom $P\supset\Box\Diamond P$.

$$\begin{array}{cl} \neg P,\ \Box\Diamond P & \\ \hline \neg(P\wedge\neg\Box\Diamond P) & (\neg\Box) \end{array}$$

EXAMPLE 5. $\vdash\Box P\vee\Box Q\supset\Box(\Box P\vee\Box Q)$.

$$\begin{array}{cl} \begin{array}{cl} \neg\Box P,\ \Box P,\ \Box Q & \\ \hline \neg\Box P,\ \neg(\neg\Box P\wedge\neg\Box Q) & (\neg\wedge) \\ \hline \neg\Box P,\ \Box\neg(\neg\Box P\wedge\neg\Box Q) & (\Box) \end{array} \quad \begin{array}{cl} \neg\Box Q,\ \Box P,\ \Box Q & \\ \hline \neg\Box Q,\ \neg(\neg\Box P\wedge\neg\Box Q) & (\neg\wedge) \\ \hline \neg\Box Q,\ \Box\neg(\neg\Box P\wedge\neg\Box Q) & (\Box) \end{array} & \\ \hline \neg\Box P\wedge\neg\Box Q,\ \Box\neg(\neg\Box P\wedge\neg\Box Q) & (\wedge) \\ \hline \neg(\neg(\neg\Box P\wedge\neg\Box Q)\wedge\neg\Box\neg(\neg\Box P\wedge\neg\Box Q)) & (\neg\Box) \end{array}$$

EXAMPLE 6. $\vdash$(Sometime At(End)$\wedge$Always(At(end)$\supset Q$)$\supset$Sometime (At(End)$\wedge Q$). Here we assume that the two interdefinable modal operators "Always" and "Sometime" correspond to □ and ◇ in SC-S4 respectively. This formula says that if assertions about termination of a program "Sometime At(End)" and weak correctness of the program "Always (At(End)$\supset Q$)" hold, then an assertion about strong correctness of the program "Sometime (At(End)$\wedge Q$)" holds [8]. The proof of the formula is illustrated below in the computer output form of theorem prover TP-PML implemented on FACOM M180II, where a propositional variable P denotes the proposition At(End).

```
TP-PML ((NOT (AND (AND (NOT (NEC (NOT P))) (NEC (NOT (AND P (NOT Q)))))
    (NEC (NOT (AND P Q))))))
(THE PROOF OF)
(NOT (AND (AND (NOT (NEC (NOT P))) (NEC (NOT (AND P (NOT Q)))))
    (NEC (NOT (AND P Q)))))
((0 ((NOT (AND (AND (NOT (NEC (NOT P))) (NEC (NOT (AND P (NOT Q)))))
    (NEC (NOT (AND P Q))))))))
(1 ((NOT (AND (NOT (NEC (NOT P))) (NEC (NOT (AND P (NOT Q))))))
    (NOT (NEC (NOT (AND P Q))))) (0 NOTAND))
(2 ((NOT (NOT (NEC (NOT P)))) (NOT (NEC (NOT (AND P (NOT Q)))))
    (NOT (NEC (NOT (AND P Q))))) (1 NOTAND))
(3 ((NOT (NEC (NOT (AND P Q)))) (NEC (NOT P)) (NOT (NEC (NOT (AND P
    (NOT Q)))))) (2 NOTNOT))
(4 ((NOT (NEC (NOT (AND P Q)))) (NOT (NEC (NOT (AND P (NOT Q)))))
    (NOT P)) (3 NEC))
(5 ((NOT (NEC (NOT (AND P (NOT Q))))) (NOT P) (NOT (NOT (AND P Q))))
    (4 NOTNEC))
(6 ((NOT P) (NOT (NOT (AND P Q))) (NOT (NOT (AND P (NOT Q))))) (5 NOTNEC))
(7 ((NOT P) (AND P Q) (NOT (NOT (AND P (NOT Q))))) (6 NOTNOT))
(8 ((NOT P) (AND P Q) (AND P (NOT Q))) (7 NOTNOT))
(9 (P (NOT P) (AND P (NOT Q))) (8 NOTAND))
(9 (P (NOT P) (AND P (NOT Q))) IS-AXIOM)
(10 (Q (NOT P) (AND P (NOT Q))) (8 NOTAND))
(11 (P Q (NOT P)) (10 NOTAND))
(11 (P Q (NOT P)) IS-AXIOM)
(12 ((NOT Q) Q (NOT P)) (10 NOTAND))
(12 ((NOT Q) Q (NOT P)) IS-AXIOM)
=PROVABLE
```

Note that in the proof, formulas are represented by Polish notation, and a sequence is represented by the list of the form (sequence no. (wff_1 $wff_2 \cdots wff_n$) justification list), where justification list is of the form (sequence no. inference rule name or "IS-AXIOM"), denoting the justification of a derivation of current sequence. Then, the line of a proof should be read as: Either the current sequence of "sequence no." has been obtained by applying "inference rule name" to the sequence of the former "sequence no.", or it is an axiom.

5. Conclusions

We have presented a computer-oriented axiomatization of propositional modal logic S4 and a proof procedure. Our method seems to be easily contracted or extended to other propositional modal logic such as T or S5 [7] since it employs completely proof-theoretic approach. If we used the model-theoretic concepts such as world and accessibility relation over the set of worlds in a decision procedure, we would have to alter the structure itself of our proof procedure of DPSC-S4 for the systems T and S5. That is, our proof-theoretic approach to proof procedure for modal logic is not only adequate for a computer implementation, but also flexible for the uniform treatment of various modal systems.

Our future research plans are to introduce the measure of proof complexity for a modal formula and to develop our Sequence Calculus so as to be able to obtain proof procedures for modal predicate logic and non-monotonic logic [9].

Acknowledgements

The author would like to acknowledge the continuing guidance and encouragement of Dr. Tosio Kitagawa, the president of his institute, and the useful comments of Mr. T. Takeshima.

References

[1] J.R. HOBBS: *Making computational sense of Montague's intensional logic.* Artificial Intelligence, Vol. 9 (1978), 287-306.

[2] V.R. PRATT: *Semantical considerations on Floyd-Hoare logic.* 17th Ann. IEEE Symposium on Foundations of Computer Science (1976), 109-121.

[3] B.R. GAINES: *Logical foundations for database systems.* Int. J. Man-Machine Studies, Vol. 19 (1979), 481-500.

[4] G. GENTZEN: *Untersuchungen über das logische schliessen, I, II. Mathematische Zeitschrift,* Vol. 39 (1935), 176-210, 405-431; English translation in the collected papers of Gerhard Gentzen (M.E. Szabo, Editor), North-Holland, Amsterdam, (1969).

[5] M. OHNISHI and K. MATSUMOTO: *Gentzen method in modal calculi, I, II.* Osaka Journal of Mathematics, Vol. 9 (1957), 113-130 and Vol. II (1959), 115-120.

[6] H. RASIOWA and R. SIKORSKI: *On the Gentzen theorem.* Fundamenta Mathematicae XLVIII (1960), 57-69.

[7] G.E. HUGHES and M.J. CRESSWELL: *An introduction to modal logic.* Methuen and Co. Ltd., (1968).

[8] R.M. BURSTALL: *Program proving as hand simulation with a little induction.* IFIP 74 (1974), 308-312.

[9] D. McDERMOTT and J. DOYLE: *Non-monotonic logic I.* Artificial Intelligence, Vol. 13, No. 1-2 (1980), 41-72.

Communicated by T. Kitagawa
Received July 25, 1984