Sequence-Indexed Linear-Time Temporal Logic: Proof System and Application*

Ken Kaneiwa
Department of Electrical Engineering and Computer Science, Iwate University
4-3-5 Ueda, Morioka, Iwate 020-8551, Japan
kaneiwa@cis.iwate-u.ac.jp

Norihiro Kamide
Waseda Institute for Advanced Study, Waseda University
1-6-1 Nishi Waseda, Shinjuku-ku, Tokyo 169-8050, Japan
logician-kamide@aoni.waseda.jp

Abstract

In this paper, we propose a proof system for reasoning on certain specifications of secure authentication systems. For this purpose, a new logic, sequence-indexed linear-time temporal logic (SLTL), is obtained semantically from standard linear-time temporal logic (LTL) by adding a sequence modal operator that represents a sequence of symbols. By this sequence modal operator, we can appropriately express message flows between clients and servers and states of servers in temporal reasoning. A Gentzen-type sequent calculus for SLTL is introduced, and the completeness and cut-elimination theorems for it are proved. SLTL is also shown to be PSPACE-complete and embeddable into LTL.

Keywords: Sequential information, secure password authentication system, linear-time temporal logic, sequent calculus, completeness theorem, cut-elimination theorem.

1 Introduction

1.1 Comparison between LTL, SLTL, and CTLS*

Linear-time temporal logic (LTL) [1, 8] is known to be one of the most useful logics for temporal reasoning and system verification. In this paper, a new logic, *sequence-*

*This paper is an extended and refined version of the conference presentation [2].

indexed linear-time temporal logic (SLTL), is obtained from LTL by adding a sequence modal operator that represents a sequence of symbols. By this sequence modal operator we can appropriately express "sequential information" in temporal reasoning.

A sequence modal operator in temporal reasoning was studied by Kamide and Kaneiwa [3] based on *branching-time temporal logic*, CTLS*. This logic can suitably represent hierarchical tree structures where the sequence modal operator of CTLS* is applied to tree structures. CTLS* is, however, not appropriate for obtaining a proof-theoretical basis for temporal reasoning with sequential information. Indeed, it is difficult to construct a good proof system for CTLS* since the branching-time formalism is not suitable for obtaining a simple proof theory. A proof system such as a *Gentzen-type sequent calculus* with completeness and cut-elimination theorems has been required for developing foundations of automated temporal theorem proving with sequential information.

In this paper, we improve on the shortcoming for CTLS*, i.e., that it lacks a good proof theory. Compared with CTLS*, the SLTL logic introduced in this paper has the following advantages:

1. A natural and simple Gentzen-type sequent calculus, SLT$_\omega$, also introduced in this paper

2. The completeness and cut-elimination theorems for SLT$_\omega$ hold

3. Some secure password authentication systems can suitably be specified and verified using SLT$_\omega$.

SLTL thus allows us to obtain a good proof theory for temporal reasoning with sequential information.

1.2 Sequential Information and Sequence Modal Operator

The reason underlying the use of the notion of sequences in the sequence modal operator is explained below. The notion of sequences is fundamental for practical reasoning in computer science because it can appropriately represent sequences such as data, program-execution, action, time, word (character or alphabet), and DNA. This notion is thus useful for representing the notions of information, attributes, trees, orders, preferences, strings, vectors, and ontologies. Sequential information can appropriately be represented by sequences because a sequence structure gives a *monoid* $\langle M, ;, \emptyset \rangle$ with an *informational interpretation* [10]:

1. M is a set of pieces of (sequential, ordered or prioritized) information (i.e., a set of sequences),

2. ; is a binary operator (on M) that combines two pieces of information (i.e., a concatenation operator on sequences),

3. $\emptyset$ is an empty piece of information (i.e., the empty sequence).

The sequence modal operator $[b]$ represents labels as "sequential information." A formula of the form $[b_1 \ ; \ b_2 \ ; \cdots ; \ b_n]\alpha$ intuitively means that "α is true based on a sequence $b_1 \ ; \ b_2 \ ; \cdots ; \ b_n$ of information pieces." A formula of the form $[\emptyset]\alpha$, which coincides with α, also intuitively means that "α is true without any information (i.e., it is an eternal truth in the sense of classical logic)." Simple and intuitive satisfaction relations by indexing sequences are required to formalize the sequence modal operator. These satisfaction relations are regarded as natural extensions of the standard satisfaction relation of classical logic. The proposed satisfaction relations, denoted as $\models^{\hat{d}}$, are indexed by a sequence $\hat{d}$, and the special case $\models^{\emptyset}$ corresponds to the classical satisfaction relation. Then, $\models^{\hat{d}} \alpha$ means that "α is true based on a sequence $\hat{d}$ of information pieces" and $\models^{\emptyset} \alpha$ means that "α is eternally true without any information."

SLTL is obtained by adding a sequence modal operator to the standard linear-time temporal logic LTL. Because of the sequence modal and temporal operators, our proposed proof system is useful for reasoning on some specifications of secure authentication systems. In particular, the sequence modal and temporal operators can appropriately express message flows between clients and servers and states of servers on networks. The proof system is designed as a Gentzen-type sequent calculus for SLTL, together with the completeness and cut-elimination theorems for this calculus. We show the complexity result that reasoning for SLTL is PSPACE-complete.

1.3 Contents of This Paper

The contents of this paper are summarized as follows. Section 2 discusses some specifications of secure password authentication systems, which are based on SLTL. Section 3 introduces SLTL semantically, and a Gentzen-type sequent calculus, SLT_{ω}, is constructed for SLTL. In Section 4, some theorems for embedding SLTL into LTL are proved, and by using these embedding theorems, the cut-elimination and completeness theorems for SLT_{ω} are shown. SLTL is also shown to be PSPACE-complete. Section 5 concludes the paper.

2 Secure Password Authentications in SLTL

We provide an example of specifying secure password authentication on a network using SLTL. Consider a case in which two clients input their user ID and password to access a server on a network (as shown in Fig. 1). To access the server, each client must try to log in to the system in less than three attempts. As a password-protected system, a client that inputs an incorrect password for a user ID three times is locked out for 30 seconds, i.e., the client must wait 30 seconds to try again. If a client inputs an incorrect user ID, the server system counts the number of attempts

for the incorrect user ID. Each client can log in to the server via a correct user ID and password when it is not locked out. In our approach, sequence modal operators $[b_1 ; b_2 ; \cdots ; b_n]$ in SLTL can be used to express the specifications of secure password authentications and the behaviors of clients and servers. For the specifications, SLTL can determine whether a client has successfully logged in to the server.

Let c_i and s respectively denote a client and a server. We use SLTL formulas $[b_1 ; b_2 ; \cdots ; b_n]\alpha$ to represent message flows and server states. By using the sequence modal operators, the SLTL formula $[c_i ; s]\alpha$ indicates that client c has sent a message α to server s. The SLTL formula $[s ; c_i]\alpha$ implies that server s returns a message α to client c_i. Accordingly, the sequence modal operators $[c_i ; s]$ and $[s ; c_i]$ represent the orders of message flows between client c_i and server s.

c_i: client i
s: server
$[c_i ; s]\alpha$: message flows from client c_i to server s
$[s ; c_i]\alpha$: message flows from server s to client c_i
$[s]\beta$: states of server s

In addition, the SLTL formula $[s]\beta$ implies that server s has a state β.

For the above formulas, the messages α and states β are expressed by propositional variables. The messages of user IDs and passwords are specified by the following propositional variables.

u_i: user ID for client i
p_i: password of user ID u_i
iu_i: incorrect user ID for client i
ip_i: incorrect password of user ID u_i

Moreover, the states of accept, reject, and lockout are expressed by the following propositional variables.

a_i: accept for user ID u_i
r_i: reject for lockout of user ID u_i
ir_i: reject for an incorrect password of user ID u_i
l_i: lockout for client i

Let $i \in \{1, 2\}$, and let X (next), G (always), and F (eventually) be temporal operators in SLTL formulas. The protocol rules of a secure password authentication system are described using SLTL formulas as follows:

(Accept Rule)
$R1 : \mathrm{G}(\neg[s]l \wedge [c_i ; s](u_i \wedge p_i)) \rightarrow \mathrm{X}[s ; c_i]a_i),$

(Lockout Rule)
$R2 : \mathrm{G}([s ; c_i]ir_i \rightarrow \mathrm{GX}([s ; c_i]ir_i \rightarrow \mathrm{GX}([s ; c_i]ir_i \rightarrow \mathrm{X}^1[s]l_i \wedge \cdots \wedge \mathrm{X}^{30}[s]l_i))),$

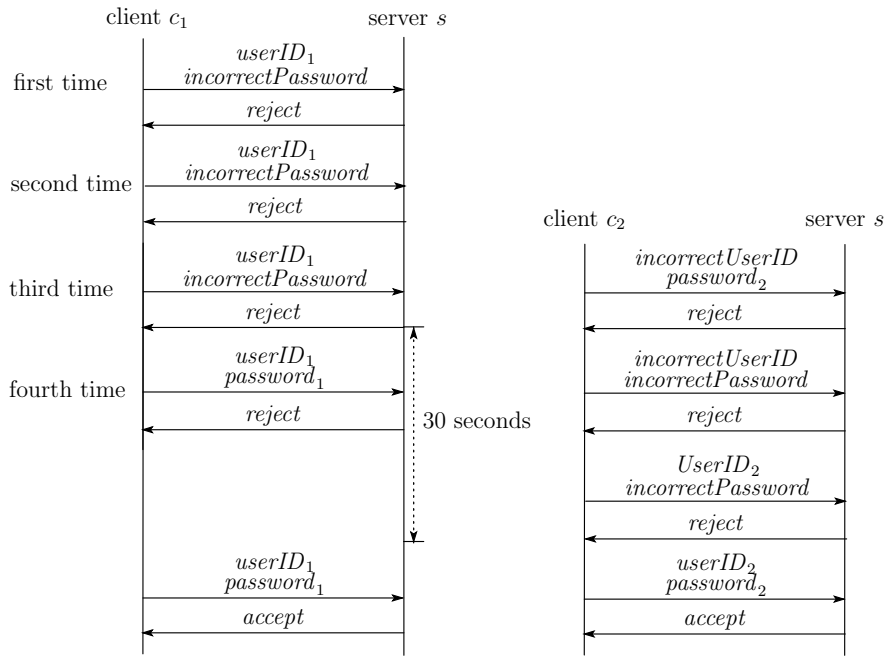

Figure 1: Message flow during successful and failed logins

(Reject Rule)
$R3 : \mathrm{G}([c_i\ ;\ s](iu_i \wedge p_i)) \rightarrow \mathrm{X}[s\ ;\ c_i]r_i),$
$R4 : \mathrm{G}(\neg[s]l_i \wedge [c_i\ ;\ s](u_i \wedge p_i) \rightarrow \mathrm{X}[s\ ;\ c_i]ir_i),$
$R5 : \mathrm{G}([c_i\ ;\ s](iu_i \wedge ip_i) \rightarrow \mathrm{X}[s\ ;\ c_i]r_i),$
$R6 : \mathrm{G}([s]l_i \wedge [c_i\ ;\ s](u_i \wedge ip_i) \rightarrow \mathrm{X}[s\ ;\ c_i]r_i).$

The formula $\mathrm{X}\alpha$ indicates that α is true in the next interval $(0, 1] = \{x \mid 0 < x \leq 1\}$ of seconds. The formula $\mathrm{G}\alpha$ (resp. $\mathrm{F}\alpha$) indicates that α is true in every future interval (resp. some future interval).

We describe a case of attempts made by a client to login to the server system. Fig. 1 shows examples of message flows between two clients $c1$ and $c2$ and server s. On the left-hand side of the figure, client $c1$ sends pairs of user ID and password to the server system s.

$A1 : \mathrm{X}(\neg[s]l_1 \wedge [c_1\ ;\ s](u_1 \wedge p_1)),$
$A2 : \mathrm{X}^3(\neg[s]l_1 \wedge [c_1\ ;\ s](u_1 \wedge p_1)),$
$A3 : \mathrm{X}^5(\neg[s]l_1 \wedge [c_1\ ;\ s](u_1 \wedge p_1)),$
$A4 : \mathrm{X}^{15}[c_1\ ;\ s](u_1 \wedge p_1).$

The formulas $A1$, $A2$, and $A3$ represent the three attempts of client c_1's logging in when no correct passwords are input. The formula $A4$ expresses the fourth attempt

of the logging in when the pair of a correct user ID and password is input. However, the fourth attempt cannot be accepted by the server system because the client is locked out.

In order to infer the state of lockout in server s, using the proof system SLT_ω for SLTL (defined in Definition 3.7), we consider the validity of the SLTL formula

$$\mathrm{X}^{15}[s]l_1$$

that implies that server s locks out client c_1 at the 15th interval. On the basis of the abovementioned specifications and assumptions, this proof system can prove the formula $\mathrm{X}^{15}[s]l_1$ as shown in the following proof-figure. We assume the following abbreviations:

$$S_1 \equiv \mathrm{X}[s]l_1 \wedge \mathrm{X}^2[s]l_1 \wedge \cdots \wedge \mathrm{X}^{30}[s]l_1,$$
$$S_2 \equiv [s\ ;\ c_1]ir_1 \rightarrow S_1,$$
$$S_3 \equiv [s\ ;\ c_1]ir_1 \rightarrow \mathrm{GX}S_2,$$
$$S_4 \equiv [s\ ;\ c_1]ir_1 \rightarrow \mathrm{GX}S_3.$$

In order to decide the validity of the formula $\mathrm{X}^{15}[s]l_1$, a proof of the required sequent:

$$R2, R4, A1, A2, A3 \Rightarrow \mathrm{X}^{15}[s]l_1$$

is obtained as follows:

$$\dfrac{\vdots\ R_1 \qquad \dfrac{\vdots\ R_2 \qquad \dfrac{\vdots\ R_3 \qquad \dfrac{\dfrac{\dfrac{\mathrm{X}^{15}[s]l_1 \Rightarrow \mathrm{X}^{15}[s]l_1}{\mathrm{X}^6(\mathrm{X}[s]l_1 \wedge \mathrm{X}^9[s]l_1) \Rightarrow \mathrm{X}^{15}[s]l_1}\ (\wedge\mathrm{left1}^s)}{\vdots\ (\wedge\mathrm{left1}^s), (\wedge\mathrm{left2}^s)}}{\mathrm{X}^6 S_1 \Rightarrow \mathrm{X}^{15}[s]l_1}\ (\rightarrow\mathrm{left}^s)}{\mathrm{X}^6 S_2, R4, A3 \Rightarrow \mathrm{X}^{15}[s]l_1}\ (\mathrm{Gleft}^s)}{\mathrm{X}^4 \mathrm{GX}S_2, R4, A3 \Rightarrow \mathrm{X}^{15}[s]l_1}\ (\rightarrow\mathrm{left}^s)}{\mathrm{X}^4 S_3, R4, A3, A2 \Rightarrow \mathrm{X}^{15}[s]l_1}\ (\mathrm{Gleft}^s)}{\mathrm{X}^2 \mathrm{GX}S_3, R4, A3, A2 \Rightarrow \mathrm{X}^{15}[s]l_1}\ (\rightarrow\mathrm{left}^s)}{\mathrm{X}^2 S_4, R4, A3, A2, A1 \Rightarrow \mathrm{X}^{15}[s]l_1}\ (\mathrm{Gleft}^s)}{R2, R4, A3, A2, A1 \Rightarrow \mathrm{X}^{15}[s]l_1}\ (\mathrm{Gleft}^s)$$

where R_1 is of the form:

$$\dfrac{\dfrac{A1 \Rightarrow A1 \quad \mathrm{X}^2[s\ ;\ c_1]ir1 \Rightarrow \mathrm{X}^2[s\ ;\ c_1]ir_1}{\mathrm{X}(\neg[s]l_1 \wedge [c_1\ ;\ s](u_1 \wedge p) \rightarrow \mathrm{X}[s\ ;\ c_1]ir_1), A1 \Rightarrow \mathrm{X}^2[s\ ;\ c_1]ir_1}\ (\rightarrow\mathrm{left}^s)}{R4, A1 \Rightarrow \mathrm{X}^2[s\ ;\ c_1]ir_1}\ (\mathrm{Gleft}^s),$$

R_2 is of the form:

$$\dfrac{\dfrac{A2 \Rightarrow P2 \quad \mathrm{X}^4[s\ ;\ c_1]ir1 \Rightarrow \mathrm{X}^4[s\ ;\ c_1]ir_1}{\mathrm{X}^3(\neg[s]l_1 \wedge [c_1\ ;\ s](u_1 \wedge p) \rightarrow \mathrm{X}[s\ ;\ c_1]ir_1), A2 \Rightarrow \mathrm{X}^4[s\ ;\ c_1]ir_1}\ (\rightarrow\mathrm{left}^s)}{R4, A2 \Rightarrow \mathrm{X}^4[s\ ;\ c_1]ir_1}\ (\mathrm{Gleft}^s)$$

and R_3 is of the form:

$$\dfrac{\dfrac{A3 \Rightarrow A3 \quad \mathrm{X}^6[s\ ;\ c_1]ir1 \Rightarrow \mathrm{X}^6[s\ ;\ c_1]ir_1}{\mathrm{X}^5(\neg[s]l_1 \wedge [c_1\ ;\ s](u_1 \wedge p) \rightarrow \mathrm{X}[s\ ;\ c_1]ir_1), A3 \Rightarrow \mathrm{X}^6[s\ ;\ c_1]ir_1}\ (\rightarrow\mathrm{left}^s)}{R4, A3 \Rightarrow \mathrm{X}^6[s\ ;\ c_1]ir_1}\ (\mathrm{Gleft}^s).$$

3 Sequence-Indexed Linear-Time Temporal Logic

In this section, we formalize the SLTL for reasoning on certain specifications of secure authentication systems.

3.1 Semantics

In this subsection, firstly, we present a semantical definition of LTL, and secondly, we introduce SLTL by extending LTL with a sequence modal operator.

Formulas of LTL are constructed from countably many propositional variables, $\rightarrow$ (implication), $\wedge$ (conjunction), $\vee$ (disjunction), $\neg$ (negation), X (next), G (globally), and F (eventually). Lower-case letters $p, q, ...$ are used to denote propositional variables, and Greek lower-case letters $\alpha, \beta, ...$ are used to denote formulas. We write $A \equiv B$ to indicate the syntactical identity between A and B. The symbol ω is used to represent the set of natural numbers. Lower-case letters i, j and k are used to denote any natural numbers. The symbol $\geq$ or $\leq$ is used to represent a linear order on ω.

Definition 3.1 (LTL) *Let S be a non-empty set of states. A structure $M := (\sigma, I)$ is a* model *if*

1. *σ is an infinite sequence $s_0, s_1, s_2, ...$ of states in S,*
2. *I is a mapping from the set Φ of propositional variables to the power set of S.*

A satisfaction relation $(M, i) \models \alpha$ for any formula α, where M is a model (σ, I) and i ($\in \omega$) represents some position within σ, is defined inductively by

1. *for any $p \in \Phi$, $(M, i) \models p$ iff $s_i \in I(p)$,*
2. *$(M, i) \models \alpha \wedge \beta$ iff $(M, i) \models \alpha$ and $(M, i) \models \beta$,*
3. *$(M, i) \models \alpha \vee \beta$ iff $(M, i) \models \alpha$ or $(M, i) \models \beta$,*

4. $(M,i) \models \alpha \rightarrow \beta$ *iff* $(M,i) \models \alpha$ *implies* $(M,i) \models \beta$,

5. $(M,i) \models \neg\alpha$ *iff not-*$[(M,i) \models \alpha]$,

6. $(M,i) \models \mathrm{X}\alpha$ *iff* $(M,i+1) \models \alpha$,

7. $(M,i) \models \mathrm{G}\alpha$ *iff* $\forall j \geq i[(M,j) \models \alpha]$,

8. $(M,i) \models \mathrm{F}\alpha$ *iff* $\exists j \geq i[(M,j) \models \alpha]$.

A formula α *is* valid *in* LTL *if* $(M,0) \models \alpha$ *for any model* $M := (\sigma, I)$.

Formulas of SLTL are constructed from countably many propositional variables, $\rightarrow$, $\wedge$, $\vee$, $\neg$, X, G, F, and $[b]$ (sequence modal operator) where b is a sequence. *Sequences* are constructed from countable atomic sequences, $\emptyset$ (empty sequence) and ; (composition). Lower-case letters $b, c, ...$ are used for sequences. An expression $[\emptyset]\alpha$ means α, and expressions $[\emptyset\ ;\ b]\alpha$ and $[b\ ;\ \emptyset]\alpha$ mean $[b]\alpha$.

Definition 3.2 *Formulas and sequences are defined by the following grammar, assuming* p *and* e *represent propositional variables and atomic sequences, respectively:*

$$\alpha ::= p \mid \alpha \wedge \alpha \mid \alpha \vee \alpha \mid \alpha \rightarrow \alpha \mid \neg\alpha \mid \mathrm{X}\alpha \mid \mathrm{G}\alpha \mid \mathrm{F}\alpha \mid [b]\alpha$$
$$b ::= e \mid \emptyset \mid b\ ;\ b$$

The set of sequences (including $\emptyset$) is denoted as SE. An expression $[\hat{d}]$ is used to represent $[d_0][d_1]\cdots[d_i]$ with $i \in \omega$ and $d_0 \equiv \emptyset$. Note that $[\hat{d}]$ can be the empty sequence. Also, an expression $\hat{d}$ is used to represent $d_0\ ;\ d_1\ ;\ \cdots\ ;\ d_i$ with $i \in \omega$.

Definition 3.3 (SLTL) *Let* S *be a non-empty set of states. A structure* $M := (\sigma, \{I^{\hat{d}}\}_{\hat{d} \in \mathrm{SE}})$ *is a* sequence model *if*

1. σ *is an infinite sequence* $s_0, s_1, s_2, ...$ *of states in* S,

2. $I^{\hat{d}}$ $(\hat{d} \in \mathrm{SE})$ *are mappings from the set* Φ *of propositional variables to the power set of* S.

Satisfaction relations $(M,i) \models^{\hat{d}} \alpha$ $(\hat{d} \in \mathrm{SE})$ *for any formula* α, *where* M *is a sequence model* $(\sigma, \{I^{\hat{d}}\}_{\hat{d} \in \mathrm{SE}})$ *and* i $(\in \omega)$ *represents some position within* σ, *is defined inductively by*

1. *for any* $p \in \Phi$, $(M,i) \models^{\hat{d}} p$ *iff* $s_i \in I^{\hat{d}}(p)$,

2. $(M,i) \models^{\hat{d}} \alpha \wedge \beta$ *iff* $(M,i) \models^{\hat{d}} \alpha$ *and* $(M,i) \models^{\hat{d}} \beta$,

3. $(M,i) \models^{\hat{d}} \alpha \vee \beta$ *iff* $(M,i) \models^{\hat{d}} \alpha$ *or* $(M,i) \models^{\hat{d}} \beta$,

4. $(M,i) \models^{\hat{d}} \alpha \rightarrow \beta$ *iff* $(M,i) \models^{\hat{d}} \alpha$ *implies* $(M,i) \models^{\hat{d}} \beta$,

5. $(M,i) \models^{\hat{d}} \neg\alpha$ *iff* *not-*$[(M,i) \models^{\hat{d}} \alpha]$,

6. $(M,i) \models^{\hat{d}} \mathrm{X}\alpha$ *iff* $(M,i+1) \models^{\hat{d}} \alpha$,

7. $(M,i) \models^{\hat{d}} \mathrm{G}\alpha$ *iff* $\forall j \geq i[(M,j) \models^{\hat{d}} \alpha]$,

8. $(M,i) \models^{\hat{d}} \mathrm{F}\alpha$ *iff* $\exists j \geq i[(M,j) \models^{\hat{d}} \alpha]$.

9. *for any atomic sequence* e, $(M,i) \models^{\hat{d}} [e]\alpha$ *iff* $(M,i) \models^{\hat{d}\ ;\ e} \alpha$,

10. $(M,i) \models^{\hat{d}} [b\ ;\ c]\alpha$ *iff* $(M,i) \models^{\hat{d}} [b][c]\alpha$.

A formula α *is* valid *in* SLTL *if* $(M,0) \models^{\emptyset} \alpha$ *for any sequence model* $M := (\sigma, \{I^{\hat{d}}\}_{\hat{d}\in \mathrm{SE}})$.

Remark that $\models^{\emptyset}$ of SLTL includes $\models$ of LTL, and hence SLTL is an extension of LTL.

Proposition 3.4 *The following clauses hold for any formula* α *and any sequences* c *and* $\hat{d}$,

1. $(M,i) \models^{\hat{d}} [c]\alpha$ *iff* $(M,i) \models^{\hat{d}\ ;\ c} \alpha$,

2. $(M,i) \models^{\emptyset} [\hat{d}]\alpha$ *iff* $(M,i) \models^{\hat{d}} \alpha$.

Proof. Since (2) is derived from (1), we show only (1) below. (1) is proved by induction on c.

Case ($c \equiv \emptyset$): Obvious.

Case ($c \equiv e$ for an atomic sequence e): By the definition of $\models^{\hat{d}}$.

Case ($c \equiv b_1\ ;\ b_2$): $(M,i) \models^{\hat{d}} [b_1\ ;\ b_2]\alpha$ iff $(M,i) \models^{\hat{d}} [b_1][b_2]\alpha$ iff $(M,i) \models^{\hat{d}\ ;\ b_1} [b_2]\alpha$ (by induction hypothesis) iff $(M,i) \models^{\hat{d}\ ;\ b_1\ ;\ b_2} \alpha$ (by induction hypothesis). ∎

An expression $\alpha \leftrightarrow \beta$ means $(\alpha \rightarrow \beta) \wedge (\beta \rightarrow \alpha)$.

Proposition 3.5 *The following formulas are valid in* SLTL*: for any formulas* α *and* β *and any* $b, c \in$ *SE,*

1. $[b](\alpha \circ \beta) \leftrightarrow ([b]\alpha) \circ ([b]\beta)$ *where* $\circ \in \{\wedge, \vee, \rightarrow\}$,

2. $[b](\sharp\alpha) \leftrightarrow \sharp([b]\alpha)$ *where* $\sharp \in \{\neg, \mathrm{X}, \mathrm{G}, \mathrm{F}\}$,

3. $[b\ ;\ c]\alpha \leftrightarrow [b][c]\alpha$.

3.2 Sequent Calculus

In this subsection, firstly, we present a sequent calculus for LTL, and secondly, we introduce a sequent calculus for SLTL.

Greek capital letters $\Gamma, \Delta, ...$ are used to represent finite (possibly empty) sets of formulas. An expression $\mathrm{X}^i\alpha$ for any $i \in \omega$ is defined inductively by $\mathrm{X}^0\alpha \equiv \alpha$ and $\mathrm{X}^{n+1}\alpha \equiv \mathrm{X}^n\mathrm{X}\alpha$. An expression of the form $\Gamma \Rightarrow \Delta$ is called a *sequent*. An expression $L \vdash S$ is used to denote the fact that a sequent S is provable in a sequent calculus L. A rule R of inference is said to be *admissible* in a sequent calculus L if the following condition is satisfied: for any instance

$$\frac{S_1 \cdots S_n}{S}$$

of R, if $L \vdash S_i$ for all i, then $L \vdash S$.

Kawai's sequent calculus LT_ω [7] for LTL is presented below.

Definition 3.6 ($\mathbf{LT}_\omega$) *The initial sequents of* LT_ω *are of the form: for any propositional variable* p*,*

$$\mathrm{X}^i p \Rightarrow \mathrm{X}^i p.$$

The structural rules of LT_ω *are of the form:*

$$\frac{\Gamma \Rightarrow \Delta, \alpha \quad \alpha, \Sigma \Rightarrow \Pi}{\Gamma, \Sigma \Rightarrow \Delta, \Pi}\ (\mathrm{cut})$$

$$\frac{\Gamma \Rightarrow \Delta}{\alpha, \Gamma \Rightarrow \Delta}\ (\text{we-left}) \quad \frac{\Gamma \Rightarrow \Delta}{\Gamma \Rightarrow \Delta, \alpha}\ (\text{we-right}).$$

The logical inference rules of LT_ω *are of the form:*

$$\frac{\Gamma \Rightarrow \Sigma, \mathrm{X}^i\alpha \quad \mathrm{X}^i\beta, \Delta \Rightarrow \Pi}{\mathrm{X}^i(\alpha{\to}\beta), \Gamma, \Delta \Rightarrow \Sigma, \Pi}\ (\to\text{left}) \quad \frac{\mathrm{X}^i\alpha, \Gamma \Rightarrow \Delta, \mathrm{X}^i\beta}{\Gamma \Rightarrow \Delta, \mathrm{X}^i(\alpha{\to}\beta)}\ (\to\text{right})$$

$$\frac{\mathrm{X}^i\alpha, \Gamma \Rightarrow \Delta}{\mathrm{X}^i(\alpha \wedge \beta), \Gamma \Rightarrow \Delta}\ (\wedge\text{left1}) \quad \frac{\mathrm{X}^i\beta, \Gamma \Rightarrow \Delta}{\mathrm{X}^i(\alpha \wedge \beta), \Gamma \Rightarrow \Delta}\ (\wedge\text{left2})$$

$$\frac{\Gamma \Rightarrow \Delta, \mathrm{X}^i\alpha \quad \Gamma \Rightarrow \Delta, \mathrm{X}^i\beta}{\Gamma \Rightarrow \Delta, \mathrm{X}^i(\alpha \wedge \beta)}\ (\wedge\text{right}) \quad \frac{\mathrm{X}^i\alpha, \Gamma \Rightarrow \Delta \quad \mathrm{X}^i\beta, \Gamma \Rightarrow \Delta}{\mathrm{X}^i(\alpha \vee \beta), \Gamma \Rightarrow \Delta}\ (\vee\text{left})$$

$$\frac{\Gamma \Rightarrow \Delta, \mathrm{X}^i\alpha}{\Gamma \Rightarrow \Delta, \mathrm{X}^i(\alpha \vee \beta)}\ (\vee\text{right1}) \quad \frac{\Gamma \Rightarrow \Delta, \mathrm{X}^i\beta}{\Gamma \Rightarrow \Delta, \mathrm{X}^i(\alpha \vee \beta)}\ (\vee\text{right2})$$

$$\frac{\Gamma \Rightarrow \Delta, \mathrm{X}^i\alpha}{\mathrm{X}^i\neg\alpha, \Gamma \Rightarrow \Delta}\ (\neg\text{left}) \quad \frac{\mathrm{X}^i\alpha, \Gamma \Rightarrow \Delta}{\Gamma \Rightarrow \Delta, \mathrm{X}^i\neg\alpha}\ (\neg\text{right})$$

$$\frac{\mathrm{X}^{i+k}\alpha, \Gamma \Rightarrow \Delta}{\mathrm{X}^i\mathrm{G}\alpha, \Gamma \Rightarrow \Delta}\ (\mathrm{Gleft}) \quad \frac{\{\ \Gamma \Rightarrow \Delta, \mathrm{X}^{i+j}\alpha\ \}_{j\in\omega}}{\Gamma \Rightarrow \Delta, \mathrm{X}^i\mathrm{G}\alpha}\ (\mathrm{Gright})$$

$$\frac{\{\ \mathrm{X}^{i+j}\alpha, \Gamma \Rightarrow \Delta\ \}_{j\in\omega}}{\mathrm{X}^i\mathrm{F}\alpha, \Gamma \Rightarrow \Delta}\ (\mathrm{Fleft}) \quad \frac{\Gamma \Rightarrow \Delta, \mathrm{X}^{i+k}\alpha}{\Gamma \Rightarrow \Delta, \mathrm{X}^i\mathrm{F}\alpha}\ (\mathrm{Fright}).$$

Remark that (Gright) and (Fleft) have infinite premises. The sequents of the form: $\mathrm{X}^i\alpha \Rightarrow \mathrm{X}^i\alpha$ for any formula α are provable in cut-free LT_ω. This fact can be proved by induction on the complexity of α. The cut-elimination and completeness theorems for LT_ω were proved by Kawai [7].

Prior to introduce a sequent calculus for SLTL, we have to introduce some notations. The symbol K is used to represent the set $\{\mathrm{X}\}\cup\{[b] \mid b \in \mathrm{SE}\}$, and the symbol K^* is used to represent the set of all words of finite length of the alphabet K. For example, $\mathrm{X}^i[\hat{b}]\mathrm{X}^j[\hat{c}]$ is in K^*. Remark that K^* includes $\emptyset$, and hence $\{\dagger\alpha \mid \dagger \in K^*\}$ includes α. An expression $\sharp$ is used to represent an arbitrary member of K^*.

A sequent calculus SLT_ω for SLTL is then introduced below.

Definition 3.7 (SLT_ω) *The initial sequents of* SLT_ω *are of the form: for any propositional variable* p*,*

$$\sharp p \Rightarrow \sharp p.$$

The structural rules of SLT_ω *are* (cut)*,* (we-left) *and* (we-right) *in Definition 3.6. The logical inference rules of* SLT_ω *are of the form:*

$$\frac{\Gamma \Rightarrow \Sigma, \sharp\alpha \quad \sharp\beta, \Delta \Rightarrow \Pi}{\sharp(\alpha\to\beta), \Gamma, \Delta \Rightarrow \Sigma, \Pi}\ (\to\mathrm{left}^s) \qquad \frac{\sharp\alpha, \Gamma \Rightarrow \Delta, \sharp\beta}{\Gamma \Rightarrow \Delta, \sharp(\alpha\to\beta)}\ (\to\mathrm{right}^s)$$

$$\frac{\sharp\alpha, \Gamma \Rightarrow \Delta}{\sharp(\alpha \wedge \beta), \Gamma \Rightarrow \Delta}\ (\wedge\mathrm{left}1^s) \qquad \frac{\sharp\beta, \Gamma \Rightarrow \Delta}{\sharp(\alpha \wedge \beta), \Gamma \Rightarrow \Delta}\ (\wedge\mathrm{left}2^s)$$

$$\frac{\Gamma \Rightarrow \Delta, \sharp\alpha \quad \Gamma \Rightarrow \Delta, \sharp\beta}{\Gamma \Rightarrow \Delta, \sharp(\alpha \wedge \beta)}\ (\wedge\mathrm{right}^s) \qquad \frac{\sharp\alpha, \Gamma \Rightarrow \Delta \quad \sharp\beta, \Gamma \Rightarrow \Delta}{\sharp(\alpha \vee \beta), \Gamma \Rightarrow \Delta}\ (\vee\mathrm{left}^s)$$

$$\frac{\Gamma \Rightarrow \Delta, \sharp\alpha}{\Gamma \Rightarrow \Delta, \sharp(\alpha \vee \beta)}\ (\vee\mathrm{right}1^s) \qquad \frac{\Gamma \Rightarrow \Delta, \sharp\beta}{\Gamma \Rightarrow \Delta, \sharp(\alpha \vee \beta)}\ (\vee\mathrm{right}2^s)$$

$$\frac{\Gamma \Rightarrow \Delta, \sharp\alpha}{\sharp\neg\alpha, \Gamma \Rightarrow \Delta}\ (\neg\mathrm{left}^s) \qquad \frac{\sharp\alpha, \Gamma \Rightarrow \Delta}{\Gamma \Rightarrow \Delta, \sharp\neg\alpha}\ (\neg\mathrm{right}^s)$$

$$\frac{\sharp\mathrm{X}^k\alpha, \Gamma \Rightarrow \Delta}{\sharp\mathrm{G}\alpha, \Gamma \Rightarrow \Delta}\ (\mathrm{Gleft}^s) \qquad \frac{\{\ \Gamma \Rightarrow \Delta, \sharp\mathrm{X}^j\alpha\ \}_{j\in\omega}}{\Gamma \Rightarrow \Delta, \sharp\mathrm{G}\alpha}\ (\mathrm{Gright}^s)$$

$$\frac{\{\ \sharp\mathrm{X}^j\alpha, \Gamma \Rightarrow \Delta\ \}_{j\in\omega}}{\sharp\mathrm{F}\alpha, \Gamma \Rightarrow \Delta}\ (\mathrm{Fleft}^s) \qquad \frac{\Gamma \Rightarrow \Delta, \sharp\mathrm{X}^k\alpha}{\Gamma \Rightarrow \Delta, \sharp\mathrm{F}\alpha}\ (\mathrm{Fright}^s)$$

$$\frac{\sharp[b]\mathrm{X}\alpha, \Gamma \Rightarrow \Delta}{\sharp\mathrm{X}[b]\alpha, \Gamma \Rightarrow \Delta}\ (\mathrm{Xleft}) \qquad \frac{\Gamma \Rightarrow \Delta, \sharp[b]\mathrm{X}\alpha}{\Gamma \Rightarrow \Delta, \sharp\mathrm{X}[b]\alpha}\ (\mathrm{Xright}).$$

The sequence inference rules of SLT_ω *are of the form:*

$$\frac{\sharp[b][c]\alpha, \Gamma \Rightarrow \Delta}{\sharp[b\ ;\ c]\alpha, \Gamma \Rightarrow \Delta}\ (;\mathrm{left}) \qquad \frac{\Gamma \Rightarrow \Delta, \sharp[b][c]\alpha}{\Gamma \Rightarrow \Delta, \sharp[b\ ;\ c]\alpha}\ (;\mathrm{right}).$$

The sequents of the form $\sharp\alpha \Rightarrow \sharp\alpha$ for any formula α are provable in cut-free SLT_ω. This fact can be proved by induction on the complexity of α.

Proposition 3.8 *The rules of the form:*

$$\frac{\Gamma \Rightarrow \Delta}{[b]\Gamma \Rightarrow [b]\Delta}\ (\text{regu}) \quad \frac{\sharp \mathrm{X}[b]\alpha, \Gamma \Rightarrow \Delta}{\sharp[b]\mathrm{X}\alpha, \Gamma \Rightarrow \Delta}\ (\mathrm{Xleft}^{-1}) \quad \frac{\Gamma \Rightarrow \Delta, \sharp \mathrm{X}[b]\alpha}{\Gamma \Rightarrow \Delta, \sharp[b]\mathrm{X}\alpha}\ (\mathrm{Xright}^{-1})$$

are admissible in cut-free SLT_ω.

Proof. We show only the case for (regu) by induction on the proofs P of $\Gamma \Rightarrow \Delta$ in cut-free SLT_ω. We distinguish the cases according to the last inference of P. We show some cases.

Case ($\rightarrow$left^s): The last inference of P is of the form:

$$\frac{\Gamma_1 \Rightarrow \Delta_1, \sharp\alpha \quad \sharp\beta, \Gamma_2 \Rightarrow \Delta_2}{\sharp(\alpha \rightarrow \beta), \Gamma_1, \Gamma_2 \Rightarrow \Delta_1, \Delta_2}\ (\rightarrow\text{left}).$$

By induction hypothesis, we have $\mathrm{SLT}_\omega - (\mathrm{cut}) \vdash [b]\Gamma_1 \Rightarrow [b]\Delta_1, [b]\sharp\alpha$ and $\mathrm{SLT}_\omega - (\mathrm{cut}) \vdash [b]\sharp\beta, [b]\Gamma_2 \Rightarrow [b]\Delta_2$. Then, we obtain the required fact:

$$\frac{\begin{array}{cc}\vdots & \vdots \\ [b]\Gamma_1 \Rightarrow [b]\Delta_1, [b]\sharp\alpha & [b]\sharp\beta, [b]\Gamma_2 \Rightarrow [b]\Delta_2\end{array}}{[b]\sharp(\alpha \rightarrow \beta), [b]\Gamma_1, [b]\Gamma_2 \Rightarrow [b]\Delta_1, [b]\Delta_2}\ (\rightarrow\text{left}^s).$$

Case (Gright^s): The last inference of P is of the form:

$$\frac{\{\ \Gamma \Rightarrow \Delta', \sharp \mathrm{X}^j\alpha\ \}_{j\in\omega}}{\Gamma \Rightarrow \Delta', \sharp \mathrm{G}\alpha}\ (\mathrm{Gright}^s).$$

By induction hypothesis, we have $\mathrm{SLT}_\omega - (\mathrm{cut}) \vdash [b]\Gamma \Rightarrow [b]\Delta', [b]\sharp\mathrm{X}^j\alpha$ for all $j \in \omega$. Then, we obtain the required fact:

$$\frac{\begin{array}{c}\vdots \\ \{\ [b]\Gamma \Rightarrow [b]\Delta', [b]\sharp\mathrm{X}^j\alpha\ \}_{j\in\omega}\end{array}}{[b]\Gamma \Rightarrow [b]\Delta', [b]\sharp\mathrm{G}\alpha}\ (\mathrm{Gright}^s).$$

∎

Remark that the rule (regu) in Proposition 3.8 is more expressive than the following standard inference rules for the normal modal logics K and KD:

$$\frac{\Gamma \Rightarrow \alpha}{\bigcirc\Gamma \Rightarrow \bigcirc\alpha} \qquad \frac{\Gamma \Rightarrow \gamma}{\bigcirc\Gamma \Rightarrow \bigcirc\gamma}$$

where γ can be empty. Thus, the sequence modal operator $[b]$ in SLTL is stronger than the modal operators $\bigcirc$ in K and KD.

An expression $\alpha \Leftrightarrow \beta$ means the sequents $\alpha \Rightarrow \beta$ and $\beta \Rightarrow \alpha$.

Proposition 3.9 *The following sequents are provable in cut-free* SLT_ω*: for any formulas* α *and* β *and any* $b, c \in SE$*,*

1. $[b](\alpha \circ \beta) \Leftrightarrow ([b]\alpha) \circ ([b]\beta)$ *where* $\circ \in \{\wedge, \vee, \rightarrow\}$*,*
2. $[b](\sharp\alpha) \Leftrightarrow \sharp([b]\alpha)$ *where* $\sharp \in \{\neg, \mathrm{X}, \mathrm{G}, \mathrm{F}\}$*,*
3. $[b\ ;\ c]\alpha \Leftrightarrow [b][c]\alpha$*.*

Proof. We show only the case $[b]\mathrm{G}\alpha \Rightarrow \mathrm{G}[b]\alpha$ as follows:

$$\dfrac{\dfrac{\dfrac{\{\ \mathrm{X}^j[b]\alpha \Rightarrow \mathrm{X}^j[b]\alpha\ \}_{j\in\omega}}{\vdots\ (\mathrm{Xleft})}}{\{\ [b]\mathrm{X}^j\alpha \Rightarrow \mathrm{X}^j[b]\alpha\ \}_{j\in\omega}}\ (\mathrm{Gleft}^s)}{\{\ [b]\mathrm{G}\alpha \Rightarrow \mathrm{X}^j[b]\alpha\ \}_{j\in\omega}}\ (\mathrm{Gright}^s).$$

∎

4 Completeness, Complexity, and Cut-Elimination

Firstly, we introduce a translation of SLTL into LTL, and by using this translation, we show two theorems for semantically and syntactically embedding SLTL into LTL. As corollaries of these embedding theorems, we obtain the cut-elimination, completeness, and complexity theorems for SLT_ω (or SLTL).

Definition 4.1 *Let* Φ *be a non-empty set of propositional variables and* $\Phi^{\hat{d}}$ *be the set* $\{p^{\hat{d}} \mid p \in \Phi\}$ *(*$\hat{d} \in \mathrm{SE}$*) of propositional variables where* $p^\emptyset := p$ *(i.e.,* $\Phi^\emptyset := \Phi$*). The language* $\mathcal{L}^s$ *(the set of formulas) of* SLTL *is defined using* Φ*,* $[b]$*,* $\wedge, \vee, \rightarrow, \neg$*,* X*,* F*, and* G *by the same way as in Definition 3.2. The language* $\mathcal{L}$ *of* LTL *is obtained from* $\mathcal{L}^s$ *by adding* $\Phi^{\hat{d}}$ *and deleting* $[b]$*.*

A mapping f *from* $\mathcal{L}^s$ *to* $\mathcal{L}$ *is defined by:*

1. *for any* $p \in \Phi$*,* $f([\hat{d}]p) := p^{\hat{d}} \in \Phi^{\hat{d}}$*, esp.,* $f(p) = p \in \Phi^\emptyset$*,* [1]
2. $f(\sharp(\alpha \circ \beta)) := f(\sharp\alpha) \circ f(\sharp\beta)$ *where* $\circ \in \{\wedge, \vee, \rightarrow\}$*,*
3. $f(\sharp\dagger\alpha) := \dagger f(\sharp\alpha)$ *where* $\dagger \in \{\neg, \mathrm{X}, \mathrm{G}, \mathrm{F}\}$*,*
4. $f(\sharp[b\ ;\ c]\alpha) := f(\sharp[b][c]\alpha)$*.*

Remark that we can derive the following clause for f:

5. $f(\sharp[b]\mathrm{X}\alpha) := f(\sharp\mathrm{X}[b]\alpha)$.

[1] Remark that $f([b\ ;\ c]p) = p_{b\ ;\ c} = f([b][c]p)$ for any $b, c \in \mathrm{SE}$.

Lemma 4.2 *Let f be the mapping defined in Definition 4.1, and S be a non-empty set of states. For any sequence model $M := (\sigma, \{I^{\hat{d}}\}_{\hat{d}\in \mathrm{SE}})$ of* SLTL*, any satisfaction relations $\models^{\hat{d}}$ ($\hat{d} \in$ SE) on M, and any state s_i in σ, we can construct a model $N := (\sigma, I)$ of* LTL *and a satisfaction relation $\models$ on N such that for any formula α in $\mathcal{L}^s$,*

$$(M, i) \models^{\hat{d}} \alpha \text{ iff } (N, i) \models f([\hat{d}]\alpha).$$

Proof. Let Φ be a non-empty set of propositional variables and $\Phi^{\hat{d}}$ be the set $\{p^{\hat{d}} \mid p \in \Phi\}$. Suppose that M is a sequence model $(\sigma, \{I^{\hat{d}}\}_{\hat{d}\in \mathrm{SE}})$ where

$I^{\hat{d}}$ ($\hat{d} \in$ SE) are mappings from Φ to the power set of S.

Suppose that N is a model (σ, I) where

I is a mapping from $\bigcup_{\hat{d}\in \mathrm{SE}} \Phi^{\hat{d}}$ to the power set of S.

Suppose moreover that M and N satisfy the following condition: for any s_i in σ and any $p \in \Phi$,

$s_i \in I^{\hat{d}}(p)$ iff $s_i \in I(p^{\hat{d}})$.

Then, the lemma is proved by induction on the complexity of α.

• Base step:

Case $\alpha \equiv p \in \Phi$: We obtain: $(M, i) \models^{\hat{d}} p$ iff $s_i \in I^{\hat{d}}(p)$ iff $s_i \in I(p^{\hat{d}})$ iff $(N, i) \models p^{\hat{d}}$ iff $(N, i) \models f([\hat{d}]p)$ (by the definition of f).

• Induction step:

Case $\alpha \equiv \beta \wedge \gamma$: We obtain: $(M, i) \models^{\hat{d}} \beta \wedge \gamma$ iff $(M, i) \models^{\hat{d}} \beta$ and $(M, i) \models^{\hat{d}} \gamma$ iff $(N, i) \models f([\hat{d}]\beta)$ and $(N, i) \models f([\hat{d}]\gamma)$ (by induction hypothesis) iff $(N, i) \models f([\hat{d}]\beta) \wedge f([\hat{d}]\gamma)$ iff $(N, i) \models f([\hat{d}](\beta \wedge \gamma))$ (by the definition of f).

Case $\alpha \equiv \beta \vee \gamma$: Similar to Case $\alpha \equiv \beta \wedge \gamma$.

Case $\alpha \equiv \beta \rightarrow \gamma$: We obtain: $(M, i) \models^{\hat{d}} \beta \rightarrow \gamma$ iff $(M, i) \models^{\hat{d}} \beta$ implies $(M, i) \models^{\hat{d}} \gamma$ iff $(N, i) \models f([\hat{d}]\beta)$ implies $(N, i) \models f([\hat{d}]\gamma)$ (by induction hypothesis) iff $(N, i) \models f([\hat{d}]\beta) \rightarrow f([\hat{d}]\gamma)$ iff $(N, i) \models f([\hat{d}](\beta \rightarrow \gamma))$ (by the definition of f).

Case $\alpha \equiv \neg\beta$: We obtain: $(M, i) \models^{\hat{d}} \neg\beta$ iff not-$[(M, i) \models^{\hat{d}} \beta]$ iff not-$[(N, i) \models f([\hat{d}]\beta)]$ (by induction hypothesis) iff $(N, i) \models \neg f([\hat{d}]\beta)$ iff $(N, i) \models f([\hat{d}]\neg\beta)$ (by the definition of f).

Case $\alpha \equiv \mathrm{X}\beta$: We obtain: $(M, i) \models^{\hat{d}} \mathrm{X}\beta$ iff $(M, i + 1) \models^{\hat{d}} \beta$ iff $(N, i + 1) \models f([\hat{d}]\beta)$ (by induction hypothesis) iff $(N, i) \models \mathrm{X}f([\hat{d}]\beta)$ iff $(N, i) \models f([\hat{d}]\mathrm{X}\beta)$ (by the definition of f).

Case $\alpha \equiv \mathrm{G}\beta$: We obtain: $(M, i) \models^{\hat{d}} \mathrm{G}\beta$ iff $\forall j \geq i[(M, j) \models^{\hat{d}} \beta]$ iff $\forall j \geq i[(N, j) \models f([\hat{d}]\beta)]$ (by induction hypothesis) iff $(N, i) \models \mathrm{G}f([\hat{d}]\beta)$ iff $(N, i) \models f([\hat{d}]\mathrm{G}\beta)$ (by the definition of f).

Case $\alpha \equiv \mathrm{F}\beta$: Similar to Case $\alpha \equiv \mathrm{G}\beta$.

Case ($\alpha \equiv [b]\beta$): $(M,i) \models^{\hat{d}} [b]\beta$ iff $(M,i) \models^{\hat{d}\,;\,b} \beta$ (by Proposition 3.4) iff $(N,i) \models f([\hat{d}\,;\,b]\beta)$ (by induction hypothesis) iff $(N,i) \models f([\hat{d}][b]\beta)$ by the definition of f. ∎

Lemma 4.3 *Let f be the mapping defined in Definition 4.1, and S be a non-empty set of states. For any model $N := (\sigma, I)$ of* LTL*, any satisfaction relation $\models$ on N, and any state s_i in σ, we can construct a sequence model $M := (\sigma, \{I^{\hat{d}}\}_{\hat{d}\in \mathrm{SE}})$ of* SLTL *and satisfaction relations $\models^{\hat{d}}$ ($\hat{d} \in$ SE) on M such that for any formula α in $\mathcal{L}^s$,*

$$(N,i) \models f([\hat{d}]\alpha) \text{ iff } (M,i) \models^{\hat{d}} \alpha.$$

Proof. Similar to the proof of Lemma 4.2. ∎

Theorem 4.4 (Semantical embedding) *Let f be the mapping defined in Definition 4.1. For any formula α, α is valid in* SLTL *iff $f(\alpha)$ is valid in* LTL*.*

Proof. By Lemmas 4.2 and 4.3. ∎

An expression $f(\Gamma)$ denotes the result of replacing every occurrence of a formula α in Γ by an occurrence of $f(\alpha)$.

Theorem 4.5 (Syntactical embedding) *Let Γ and Δ be sets of formulas in $\mathcal{L}^s$, and f be the mapping defined in Definition 4.1. Then:*

1. $\mathrm{SLT}_\omega \vdash \Gamma \Rightarrow \Delta$ *iff* $\mathrm{LT}_\omega \vdash f(\Gamma) \Rightarrow f(\Delta)$.
2. $\mathrm{SLT}_\omega - (\mathrm{cut}) \vdash \Gamma \Rightarrow \Delta$ *iff* $\mathrm{LT}_\omega - (\mathrm{cut}) \vdash f(\Gamma) \Rightarrow f(\Delta)$.

Proof. Since the case (2) can be obtained as the subproof of the case (1), we show only (1) in the following. In the following proof, we assume that the total number of X in $\sharp$ is i, and that the sequence which is obtained from $\sharp$ by deleting all X is $[\hat{d}]$.

• ($\Longrightarrow$) : By induction on the proofs P of $\Gamma \Rightarrow \Delta$ in SLT_ω. We distinguish the cases according to the last inference of P, and show some cases.

Case ($\sharp p \Rightarrow \sharp p$): The last inference of P is of the form: $\sharp p \Rightarrow \sharp p$. In this case, we obtain the required fact $\mathrm{LT}_\omega \vdash f(\sharp p) \Rightarrow f(\sharp p)$ since $f(\sharp p)$ coincides with $\mathrm{X}^i p^{\hat{d}}$ by the definition of f.

Case ($\wedge\mathrm{left}^s$): The last inference of P is of the form:

$$\frac{\Gamma \Rightarrow \Delta, \sharp\alpha \quad \Gamma \Rightarrow \Delta, \sharp\beta}{\Gamma \Rightarrow \Delta, \sharp(\alpha \wedge \beta)}\ (\wedge\mathrm{left}^s).$$

By induction hypothesis, we have: $\mathrm{LT}_\omega \vdash f(\Gamma) \Rightarrow f(\Delta), f(\sharp\alpha)$ and $\mathrm{LT}_\omega \vdash f(\Gamma) \Rightarrow f(\Delta), f(\sharp\beta)$ where $f(\sharp\alpha)$ and $f(\sharp\beta)$ respectively coincide with $\mathrm{X}^i f([\hat{d}]\alpha)$ and $\mathrm{X}^i f([\hat{d}]\beta)$ by the definition of f. Then, we obtain:

$$\frac{\begin{array}{cc}\vdots & \vdots \\ f(\Gamma) \Rightarrow f(\Delta), \mathrm{X}^i f([\hat{d}]\alpha) & f(\Gamma) \Rightarrow f(\Delta), \mathrm{X}^i f([\hat{d}]\beta)\end{array}}{f(\Gamma) \Rightarrow f(\Delta), \mathrm{X}^i(f([\hat{d}]\alpha) \wedge f([\hat{d}]\beta))}\ (\wedge\text{left})$$

where $\mathrm{X}^i(f([\hat{d}]\alpha) \wedge f([\hat{d}]\beta))$ coincides with $f(\sharp(\alpha \wedge \beta))$ by the definition of f.

Case (Fleft^s): The last inference of P is of the form:

$$\frac{\{\ \sharp\mathrm{X}^j\alpha, \Gamma \Rightarrow \Delta\ \}_{j\in\omega}}{\sharp\mathrm{F}\alpha, \Gamma \Rightarrow \Delta}\ (\text{Fleft}^s).$$

By induction hypothesis, we have: $\mathrm{LT}_\omega \vdash f(\sharp\mathrm{X}^j\alpha), f(\Gamma) \Rightarrow f(\Delta)$ for any $j \in \omega$, where $f(\sharp\mathrm{X}^j\alpha)$ coincides with $\mathrm{X}^{i+j} f([\hat{d}]\alpha)$ by the definition of f. Then, we obtain:

$$\frac{\begin{array}{c}\vdots \\ \{\ \mathrm{X}^{i+j} f([\hat{d}]\alpha), f(\Gamma) \Rightarrow f(\Delta)\ \}_{j\in\omega}\end{array}}{\mathrm{X}^i\mathrm{F} f([\hat{d}]\alpha), f(\Gamma) \Rightarrow f(\Delta)}\ (\text{Fleft})$$

where $\mathrm{X}^i\mathrm{F} f([\hat{d}]\alpha)$ coincides with $f(\sharp\mathrm{F}\alpha)$ by the definition of f.

Case (;left): The last inference of P is of the form:

$$\frac{\sharp[b][c]\alpha, \Gamma \Rightarrow \Delta}{\sharp[b\ ;\ c]\alpha, \Gamma \Rightarrow \Delta}\ (;\text{left}).$$

By induction hypothesis, we obtain: $\mathrm{LT}_\omega \vdash f(\sharp[b][c]\alpha), f(\Gamma) \Rightarrow f(\Delta)$ where $f(\sharp[b][c]\alpha)$ coincides with $f(\sharp[b\ ;\ c]\alpha)$ by the definition of f.

Case (Xleft): The last inference of P is of the form:

$$\frac{\sharp[b]\mathrm{X}\alpha, \Gamma \Rightarrow \Delta}{\sharp\mathrm{X}[b]\alpha, \Gamma \Rightarrow \Delta}\ (\text{Xleft}).$$

By induction hypothesis, we obtain: $\mathrm{LT}_\omega \vdash f(\sharp[b]\mathrm{X}\alpha), f(\Gamma) \Rightarrow f(\Delta)$ where $f(\sharp[b]\mathrm{X}\alpha)$ coincides with $f(\sharp\mathrm{X}[b]\alpha)$ by the definition of f.

• ($\Longleftarrow$) : By induction on the proofs Q of $f(\Gamma) \Rightarrow f(\Delta)$ in LT_ω. We distinguish the cases according to the last inference of Q, and show only the following cases.

Case (cut): The last inference of Q is of the form:

$$\frac{f(\Gamma_1) \Rightarrow f(\Delta_1), \beta \quad \beta, f(\Gamma_2) \Rightarrow f(\Delta_2)}{f(\Gamma_1), f(\Gamma_2) \Rightarrow f(\Delta_1), f(\Delta_2)}\ (\text{cut}).$$

Since β is in $\mathcal{L}$, we have the fact $\beta = f(\beta)$. This fact can be shown by induction on β. Then, by induction hypothesis, we have: $\mathrm{SLT}_\omega \vdash \Gamma_1 \Rightarrow \Delta_1, \beta$ and $\mathrm{SLT}_\omega \vdash$

$\beta, \Gamma_2 \Rightarrow \Delta_2$. We then obtain the required fact: $\mathrm{SLT}_\omega \vdash \Gamma_1, \Gamma_2 \Rightarrow \Delta_1, \Delta_2$ by using (cut) in SLT_ω.

Case (Gleft): The last inference of Q is of the form:

$$\frac{\mathrm{X}^{j+k} f(\sharp\alpha), f(\Gamma) \Rightarrow f(\Delta)}{\mathrm{X}^{j} \mathrm{G} f(\sharp\alpha), f(\Gamma) \Rightarrow f(\Delta)}\ (\mathrm{Gleft})$$

where $\mathrm{X}^{j+k} f(\sharp\alpha)$ and $\mathrm{X}^{j}\mathrm{G} f(\sharp\alpha)$ respectively coincide with $f(\sharp \mathrm{X}^{j+k}\alpha)$ and $f(\sharp \mathrm{X}^{j}\mathrm{G}\alpha)$ by the definition of f. By induction hypothesis, we have: $\mathrm{SLT}_\omega \vdash \sharp \mathrm{X}^{j+k}\alpha, \Gamma \Rightarrow \Delta$, and hence obtain the required fact:

$$\frac{\begin{array}{c}\vdots\\ \sharp \mathrm{X}^{j+k}\alpha, \Gamma \Rightarrow \Delta\end{array}}{\sharp \mathrm{X}^{j}\mathrm{G}\alpha, \Gamma \Rightarrow \Delta}\ (\mathrm{Gleft}^s).$$

∎

Theorem 4.6 (Cut-elimination) *The rule* (cut) *is admissible in cut-free* SLT_ω.

Proof. Suppose $\mathrm{SLT}_\omega \vdash \Gamma \Rightarrow \Delta$. Then, we have $\mathrm{LT}_\omega \vdash f(\Gamma) \Rightarrow f(\Delta)$ by Theorem 4.5 (1), and hence $\mathrm{LT}_\omega - (\mathrm{cut}) \vdash f(\Gamma) \Rightarrow f(\Delta)$ by the cut-elimination theorem for LT_ω. By Theorem 4.5 (2), we obtain $\mathrm{SLT}_\omega - (\mathrm{cut}) \vdash \Gamma \Rightarrow \Delta$. ∎

Remark that in order to obtain Theorem 4.6, it is sufficient to prove the following restricted statements of Theorem 4.5:

1. if $\mathrm{SLT}_\omega \vdash \Gamma \Rightarrow \Delta$, then $\mathrm{LT}_\omega \vdash f(\Gamma) \Rightarrow f(\Delta)$.
2. if $\mathrm{LT}_\omega - (\mathrm{cut}) \vdash f(\Gamma) \Rightarrow f(\Delta)$, then $\mathrm{SLT}_\omega - (\mathrm{cut}) \vdash \Gamma \Rightarrow \Delta$.

To show the second statement, we do not need to prove the case for (cut) as in Theorem 4.5. By these restricted statements and the cut-elimination theorem for SLT_ω, we can show Theorem 4.5 again.

Theorem 4.7 (Completeness) *For any formula* α, $\mathrm{SLT}_\omega \vdash \Rightarrow \alpha$ *iff* α *is valid in* SLTL.

Proof. $\mathrm{SLT}_\omega \vdash \Rightarrow \alpha$ iff $\mathrm{LT}_\omega \vdash \Rightarrow f(\alpha)$ (by Theorem 4.5) iff $f(\alpha)$ is valid in LTL (by the completeness theorem for LTL) iff α is valid in SLTL (by Theorem 4.4). ∎

Theorem 4.8 (Complexity) SLTL *is PSPACE-complete.*

Proof. (Propositional) LTL (without until operator) is known to be PSPACE-complete [9]. By decidability of LTL, for each α, it is possible to decide if $f(\alpha)$ is valid in LTL. Then, by Theorem 4.4, SLTL is decidable. Since f is a polynomial-time reduction, SLTL is also PSPACE-complete. ∎

5 Conclusions

The new logic, SLTL, was obtained semantically from LTL by adding a sequence modal operator $[b]$ that can appropriately represent sequential information. Gentzen-type sequent calculus SLT_ω for SLTL was constructed. The semantical and syntactical embedding theorems were proved. By using these embedding theorems, the cut-elimination and completeness theorems for SLT_ω were proved. SLTL is also shown to be PSPACE-complete. As an application of SLT_ω, some specifications of secure authentication systems were proposed. It was thus shown in this paper that SLT_ω provides a good proof theory for temporal reasoning with sequential information, and that SLT_ω is useful for specifying and verifying secure password authentication systems.

Acknowledgments. K. Kaneiwa has been partially supported by the Japanese Ministry of Education, Science, Sports and Culture, Grant-in-Aid for Young Scientists (B) 20700147. N. Kamide was supported by the Japanese Ministry of Education, Culture, Sports, Science and Technology, Grant-in-Aid for Young Scientists (B) 20700015.

References

[1] E.A. Emerson, Temporal and modal logic, In Handbook of Theoretical Computer Science, Formal Models and Semantics (B), Jan van Leeuwen (Ed.), pp. 995–1072, Elsevier and MIT Press, 1990.

[2] N. Kamide, A proof system for temporal reasoning with sequential information, Proceedings of the 20th Brazilian Symposium on Artificial Intelligence (SBIA 2010), to appear.

[3] N. Kamide and K. Kaneiwa, Extended full computation-tree logic with sequence modal operator: representing hierarchical tree structures, Proceedings of the 22nd Australasian Joint Conference on Artificial Intelligence, Lecture Notes in Artificial Intelligence 5866, pp. 485–494, 2009.

[4] K. Kaneiwa, Order-sorted logic programming with predicate hierarchy, *Artificial Intelligence* 158 (2), pp. 155–188, 2004.

[5] K. Kaneiwa and R. Mizoguchi, Distributed reasoning with ontologies and rules in order-sorted logic programming, *Journal of Web Semantics* 7 (3), pp. 252–270, 2009.

[6] K. Kaneiwa and K. Satoh, On the complexities of consistency checking for restricted UML class diagrams, *Theoretical Computer Science* 411(2), pp. 301–323, 2010.

[7] H. Kawai, Sequential calculus for a first order infinitary temporal logic, *Zeitschrift für Mathematische Logik und Grundlagen der Mathematik* 33, pp. 423–432, 1987.

[8] A. Pnueli, The temporal logic of programs, Proceedings of the 18th IEEE Symposium on Foundations of Computer Science, pp. 46–57, 1977.

[9] A.P. Sistla and E.M. Clarke, The complexity of propositional linear temporal logics, *Journal of the ACM* 32 (3), pp. 733-749, 1985.

[10] H. Wansing, The logic of information structures, Lecture Notes in Artificial Intelligence 681, 163 pages, 1993.