

Norihiro KAMIDE

**PARACONSISTENT SEQUENTIAL
LINEAR-TIME TEMPORAL LOGIC:
COMBINING PARACONSISTENCY AND
SEQUENTIALITY IN TEMPORAL
REASONING**

A b s t r a c t. Inconsistency-tolerant temporal reasoning with sequential (i.e., ordered or hierarchical) information is gaining increasing importance in computer science applications. A logical system for representing such reasoning is thus required for obtaining a theoretical basis for such applications. In this paper, we introduce a new logic called paraconsistent sequential linear-time temporal logic (PSLTL), which is an extension of the standard linear-time temporal logic (LTL). PSLTL can appropriately represent inconsistency-tolerant temporal reasoning with sequential information. The cut-elimination, decidability, and completeness theorems for PSLTL are proved in this paper.

Received 22 January 2016; Revised 12 August 2016

This paper is an extended and refined version of the conference presentation [17]

Keywords and phrases: linear-time temporal logic, paraconsistent logic, sequent calculus.

AMS subject classification: 03B44, 03B53.

1. Introduction

Inconsistency-tolerant temporal reasoning with sequential (i.e., ordered or hierarchical) information is of growing importance in computer science applications such as medical informatics and agent communication. A logical system for representing such reasoning is thus required for obtaining a concrete theoretical basis for such applications. However, to the best of our knowledge, there are no logical systems that can simultaneously represent inconsistency, sequentiality, and temporality. Thus, the aim of our study is to introduce a logical system, both semantically and syntactically, for appropriately representing inconsistency-tolerant temporal reasoning with sequential information.

Hence, we introduce a new logic called *paraconsistent sequential linear-time temporal logic* (PSLTL), which is an extension of the standard *linear-time temporal logic* (LTL) [28]. Inconsistency-tolerant reasoning in PSLTL is expressed via a paraconsistent negation connective, and sequential information is represented by sequence modal operators. Temporal reasoning in PSLTL is, of course, expressed by temporal operators used in LTL. The cut-elimination, decidability, and completeness theorems for PSLTL are proved via theorems for semantically and syntactically embedding PSLTL into its fragments SLTL and LTL, where SLTL is explained later.

PSLTL is regarded as an extension of both LTL and *Nelson's paraconsistent four-valued logic with strong negation* N4 [1, 22, 27, 32]. On the one hand, LTL is known to be one of the most useful temporal logics for verifying concurrent systems. On the other hand, N4 is known to be one of the most important base logics for inconsistency-tolerant reasoning. The combination of LTL and N4 was previously studied in [21], and such a combined logic is called *paraconsistent LTL* (PLTL). PSLTL is obtained from PLTL by adding sequence modal operators.

LTL is known to be one of the most useful temporal logics in Computer Science. Indeed, LTL is one of the most useful and important base logics for formalizing model checking technologies [5, 6]. The SPIN model checker [11], which is known to be one of the most useful model checkers, was developed on the basis of LTL. Gentzen-type sequent calculi for LTL and its neighbors have been introduced and studied by many researchers. A Gentzen-type sequent calculus LT_ω for LTL was introduced by Kawai [26], who proved cut-elimination and Kripke-completeness theorems for this cal-

culus. A 2-sequent calculus $2S_\omega$ for LTL, which is a natural extension of the usual Gentzen-type sequent calculus, was introduced by Baratella and Masini [3]. A direct syntactical equivalence between LT_ω and $2S_\omega$ was shown by introducing the translation functions that preserve cut-free proofs of these calculi [12]. For more information on LTL, see, e.g., [8].

N4, which is also referred to in the literature as N^- , is known to be one of the most important and basic paraconsistent logics in Computer Science and Philosophical Logic. N4 is also known to be a common basis for various extended and useful paraconsistent logics. It is known that N4 is a paraconsistent variant of Nelson's constructive three-valued logic N3, which is also referred to as N and has been studied by several mathematical logicians. Gentzen-type sequent calculi for Nelson's logics have been investigated, and Kripke semantics for Nelson's logics have also been studied. A translation of N3 into intuitionistic logic has been proposed and studied by several researchers [31, 10, 30]. A similar translation for N4 into LJ, which is Gentzen's sequent calculus for positive intuitionistic propositional logic, can also be obtained. For a comprehensive survey on Nelson's logics, see, e.g., [22, 32].

Combining LTL with sequence modal operators was studied in [14, 23, 15], and such combined logics were called *sequence-indexed LTL* (SLTL) and *sequential paraconsistent LTL* (SPLTL). PSLTL is regarded as a modified paraconsistent extension of SLTL, and hence PSLTL is a modified extension of both PLTL [21] and SLTL [23]. In the remainder of this section, we explain an important property of paraconsistent negation and a plausible interpretation of sequence modal operators. Some comparisons among the logics mentioned above will be presented in Section 6.

The paraconsistent negation connective \sim used in PSLTL can appropriately express inconsistency-tolerant reasoning. One reason why \sim is considered is that it can be added in such a way that the extended logic satisfies the property of *paraconsistency*. A consequence relation \models is called paraconsistent with respect to a negation connective \sim if there are formulas α and β such that $\{\alpha, \sim\alpha\} \models \beta$ does not hold (this fact is denoted by $not\text{-}\{\{\alpha, \sim\alpha\} \models \beta\}$). In the case of LTL, this implies that there exists a model M and position i of a sequence $\sigma = t_0, t_1, t_2, \dots$ of time-points in M with $not\text{-}[(M, i) \models (\alpha \wedge \sim\alpha) \rightarrow \beta]$.

Logical systems with paraconsistency can handle inconsistency-tolerant and uncertainty reasoning more appropriately than systems that are non-

paraconsistent. For example, we do not want $(s(x) \wedge \sim s(x)) \rightarrow d(x)$ to be satisfied for any symptom s and disease d , where $\sim s(x)$ means “person x does not have symptom s ” and $d(x)$ means “person x suffers from disease d ,” because there may be situations that support the truth of both $s(a)$ and $\sim s(a)$ for some individual a but do not support the truth of $d(a)$.

If we cannot determine whether someone is healthy, then the vague concept *healthy* can be represented by asserting the inconsistent formula: $healthy(john) \wedge \sim healthy(john)$. This is well-formalized in PSLTL because the formula $healthy(john) \wedge \sim healthy(john) \rightarrow hasCancer(john)$ where $hasCancer(john)$ means “John has cancer” is not valid in PSLTL (i.e., PSLTL is inconsistency-tolerant). On the other hand, the formula $healthy(john) \wedge \neg healthy(john) \rightarrow hasCancer(john)$, where \neg is the classical negation connective is valid in classical logic (i.e., inconsistency has undesirable consequences). For more information on paraconsistency and inconsistency-handling, see e.g., [29, 2] and the references therein.

Some sequence modal operators [19, 14, 23, 25, 15] used in PSLTL can suitably express sequential information. A sequence modal operator $[b]$ represents a sequence b of symbols. The notion of sequences is useful to represent “information,” “trees,” and “ontologies.” Thus, “sequential (i.e., ordered or hierarchical) information” can be represented by sequences because a sequence structure gives *monoid* $\langle M, ;, \emptyset \rangle$ with *informational interpretation* [32] as follows: (1) M is a set of pieces of 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); and (3) \emptyset is the empty piece of information (i.e., the empty sequence).

A formula of the form $[b_1 ; b_2 ; \dots ; b_n]\alpha$ in PSLTL intuitively means that “ α is true based on a sequence $b_1 ; b_2 ; \dots ; b_n$ of ordered or prioritized information pieces.” Further, a formula of the form $[\emptyset]\alpha$ in PSLTL, which coincides with α , intuitively means that “ α is true without any information” (i.e., it is an eternal truth in the sense of classical logic).

The structure of the paper is as follows.

In Section 2, PSLTL is introduced semantically by extending the semantics of LTL with a paraconsistent negation connective and sequence modal operators. Within this section, LTL is first presented with its standard semantics, then SLTL is presented as semantics with some sequence modal operators. Finally, PSLTL is obtained from SLTL by adding a para-

consistent negation connective similar to that of N4.

In Section 3, we introduce a Gentzen-type sequent calculus PSLT_ω for PSLTL , which is an extension of a Gentzen-type sequent calculus LT_ω for LTL . Within this section, we first present a Gentzen-type sequent calculus LT_ω , introduced by Kawai [26], then we present a Gentzen-type sequent calculus SLT_ω for SLTL on the basis of [14, 23]. Finally, PSLT_ω is obtained from SLT_ω by adding inference rules concerning the paraconsistent negation connective.

In Section 4, we prove the cut-elimination, decidability, and completeness theorems for PSLTL (and PSLT_ω) via two theorems for semantically and syntactically embedding PSLTL (and PSLT_ω) into SLTL (SLT_ω) and LTL (LT_ω). A translation algorithm and some translation examples are also provided in this section.

In Section 5, we present some small illustrative examples on the basis of PSLTL and PSLT_ω .

Finally, in Section 6, we conclude our paper and discuss related and future works.

2. Semantics

2.1. LTL

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, \dots are used to denote propositional variables, and Greek lower-case letters α, β, \dots are used to denote formulas. An expression $\alpha \leftrightarrow \beta$ is used to denote $(\alpha \rightarrow \beta) \wedge (\beta \rightarrow \alpha)$. 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 2.1. Formulas of LTL are defined by the following grammar, assuming p represents propositional variables:

$$\alpha ::= p \mid \alpha \wedge \alpha \mid \alpha \vee \alpha \mid \alpha \rightarrow \alpha \mid \neg \alpha \mid X\alpha \mid G\alpha \mid F\alpha$$

Definition 2.2 (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, \dots 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 $(M, i) \not\models \alpha$,
6. $(M, i) \models X\alpha$ iff $(M, i+1) \models \alpha$,
7. $(M, i) \models G\alpha$ iff $\forall j \geq i [(M, j) \models \alpha]$,
8. $(M, i) \models F\alpha$ iff $\exists j \geq i [(M, j) \models \alpha]$.

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

2.2. SLTL

Formulas of SLTL are obtained from that of LTL by adding $[b]$ (sequence modal operator) where b is a sequence. *Sequences* are constructed from countable atomic sequences, \emptyset (empty sequence) and $;$ (composition). Lowercase letters b, c, \dots are used for sequences. An expression $[\emptyset]\alpha$ means α , and expressions $[\emptyset; b]\alpha$ and $[b; \emptyset]\alpha$ mean $[b]\alpha$. The set of sequences (including \emptyset) is denoted as SE. An expression $\overline{[d]}$ is used to represent $[d_0][d_1] \cdots [d_i]$ with $i \in \omega$, $d_i \in \text{SE}$ and $d_0 \equiv \emptyset$. Note that $\overline{[d]}$ can be the empty sequence. Also, an expression \hat{d} is used to represent $d_0; d_1; \cdots; d_i$ with $i \in \omega$, $d_i \in \text{SE}$ and $d_0 \equiv \emptyset$. Note that $[\hat{d}]$ means $[d_0; d_1; \cdots; d_i]$, which differs from $\overline{[d]}$.

Definition 2.3. Formulas and sequences of SLTL are defined by the following grammar, assuming p and e represent propositional variables and atomic sequences, respectively:

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

Definition 2.4 (SLTL). Let S be a non-empty set of states. A structure $M := (\sigma, \{I^{\hat{d}}\}_{\hat{d} \in \text{SE}})$ is a *sequential model* if

1. σ is an infinite sequence s_0, s_1, s_2, \dots of states in S ,
2. $I^{\hat{d}}$ ($\hat{d} \in \text{SE}$) are mappings from the set Φ of propositional variables to the power set of S .

Satisfaction relations $(M, i) \models^{\hat{d}} \alpha$ ($\hat{d} \in \text{SE}$) for any formula α , where M is a sequential model $(\sigma, \{I^{\hat{d}}\}_{\hat{d} \in \text{SE}})$ 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 $(M, i) \not\models^{\hat{d}} \alpha$,
6. $(M, i) \models^{\hat{d}} X\alpha$ iff $(M, i+1) \models^{\hat{d}} \alpha$,
7. $(M, i) \models^{\hat{d}} G\alpha$ iff $\forall j \geq i [(M, j) \models^{\hat{d}} \alpha]$,
8. $(M, i) \models^{\hat{d}} F\alpha$ iff $\exists j \geq i [(M, j) \models^{\hat{d}} \alpha]$.
9. $(M, i) \models^{\hat{d}} [b]\alpha$ iff $(M, i) \models^{\hat{d}} ; b \alpha$.

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

Some remarks on SLTL are addressed as follows.

1. SLTL is an extension of LTL since $\models^{\hat{d}}$ of SLTL includes \models of LTL.

2. The following clauses hold for SLTL: For any sequences b, c, \hat{d} and \hat{k} ,

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

$$(b) (M, i) \models^{\hat{d}} [\overline{k}]\alpha \text{ iff } (M, i) \models^{\hat{d}} ; \hat{k} \alpha.$$

3. The following formulas are valid in SLTL: For any formulas α and β and any $b, c \in \text{SE}$,

$$(a) [b](\alpha \circ \beta) \leftrightarrow ([b]\alpha) \circ ([b]\beta) \text{ where } \circ \in \{\wedge, \vee, \rightarrow\},$$

$$(b) [b](\# \alpha) \leftrightarrow \#[b]\alpha \text{ where } \# \in \{\neg, X, G, F\},$$

$$(c) [b ; c]\alpha \leftrightarrow [b][c]\alpha,$$

$$(d) [\overline{\hat{d}}]\alpha \leftrightarrow [\hat{d}]\alpha.$$

2.3. PSLTL

Formulas of PSLTL are obtained from that of SLTL by adding \sim (paraconsistent negation).

Definition 2.5. Formulas and sequences of PSLTL are defined by the following grammar, assuming p and e represent propositional variables and atomic sequences, respectively:

$$\alpha ::= p \mid \alpha \wedge \alpha \mid \alpha \vee \alpha \mid \alpha \rightarrow \alpha \mid \neg \alpha \mid \sim \alpha \mid X\alpha \mid G\alpha \mid F\alpha \mid [b]\alpha$$

$$b ::= e \mid \emptyset \mid b ; b$$

Definition 2.6 (PSLTL). Let S be a non-empty set of states. A structure $M := (\sigma, \{I^{+\hat{d}}\}_{\hat{d} \in \text{SE}}, \{I^{-\hat{d}}\}_{\hat{d} \in \text{SE}})$ is a *paraconsistent sequential model* if

1. σ is an infinite sequence s_0, s_1, s_2, \dots of states in S ,
2. $I^{*\hat{d}}$ ($* \in \{+, -\}$, $\hat{d} \in \text{SE}$) are mappings from the set Φ of propositional variables to the power set of S .

Satisfaction relations $(M, i) \models^{*\hat{d}} \alpha$ ($* \in \{+, -\}$, $\hat{d} \in \text{SE}$) for any formula α , where M is a paraconsistent sequential model $(\sigma, \{I^{+\hat{d}}\}_{\hat{d} \in \text{SE}}, \{I^{-\hat{d}}\}_{\hat{d} \in \text{SE}})$ and i ($\in \omega$) represents some position within σ , are defined 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 $(M, i) \not\models^{+\hat{d}} \alpha$,
6. $(M, i) \models^{+\hat{d}} \sim \alpha$ iff $(M, i) \models^{-\hat{d}} \alpha$,
7. $(M, i) \models^{+\hat{d}} X\alpha$ iff $(M, i+1) \models^{+\hat{d}} \alpha$,
8. $(M, i) \models^{+\hat{d}} G\alpha$ iff $\forall j \geq i [(M, j) \models^{+\hat{d}} \alpha]$,
9. $(M, i) \models^{+\hat{d}} F\alpha$ iff $\exists j \geq i [(M, j) \models^{+\hat{d}} \alpha]$,
10. for any $p \in \Phi$, $(M, i) \models^{-\hat{d}} p$ iff $s_i \in I^{-\hat{d}}(p)$,
11. $(M, i) \models^{-\hat{d}} \alpha \wedge \beta$ iff $(M, i) \models^{-\hat{d}} \alpha$ and $(M, i) \models^{-\hat{d}} \beta$,
12. $(M, i) \models^{-\hat{d}} \alpha \vee \beta$ iff $(M, i) \models^{-\hat{d}} \alpha$ or $(M, i) \models^{-\hat{d}} \beta$,
13. $(M, i) \models^{-\hat{d}} \alpha \rightarrow \beta$ iff $(M, i) \models^{+\hat{d}} \alpha$ and $(M, i) \models^{-\hat{d}} \beta$,
14. $(M, i) \models^{-\hat{d}} \neg \alpha$ iff $(M, i) \not\models^{-\hat{d}} \alpha$,
15. $(M, i) \models^{-\hat{d}} \sim \alpha$ iff $(M, i) \models^{+\hat{d}} \alpha$,
16. $(M, i) \models^{-\hat{d}} X\alpha$ iff $(M, i+1) \models^{-\hat{d}} \alpha$,
17. $(M, i) \models^{-\hat{d}} G\alpha$ iff $\exists j \geq i [(M, j) \models^{-\hat{d}} \alpha]$,
18. $(M, i) \models^{-\hat{d}} F\alpha$ iff $\forall j \geq i [(M, j) \models^{-\hat{d}} \alpha]$,
19. for any $* \in \{+, -\}$, $(M, i) \models^{*\hat{d}} [b]\alpha$ iff $(M, i) \models^{*(\hat{d}; b)} \alpha$.

A formula α is *valid* in PSLTL iff $(M, 0) \models^{+\hat{0}} \alpha$ for any paraconsistent sequential model $M := (\sigma, \{I^{+\hat{d}}\}_{\hat{d} \in \text{SE}}, \{I^{+\hat{d}}\}_{\hat{d} \in \text{SE}})$.

Some remarks on PSLTL are addressed as follows.

1. PSLTL is an extension of SLTL since $\models^{*\hat{d}}$ of PSLTL includes $\models^{\hat{d}}$ of SLTL.

2. The following clauses hold for PSLTL: For any sequences b, c, \hat{d}, \hat{k} and any $* \in \{+, -\}$,

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

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

3. F and G are duals of each other not only with respect to \neg but also with respect to \sim . X is a self dual not only with respect to \neg but also with respect to \sim . $[b]$ is a self dual not only with respect to \neg but also with respect to \sim . \neg and \sim are self-duals with respect to \sim and \neg , respectively.

4. The falsification conditions for \neg may be felt to be in need of some justification. Suppose that a is a person who is neither rich nor poor and that, as a matter of fact, no one is both rich and poor. Let p stand for the claim that a is poor and r for the claim that a is rich. Intuitively, a state definitely verifies p iff it falsifies r , and vice versa. Suppose now that $\neg p$ is indeed falsified at a state i in model M : $(M, i) \models^{-\hat{d}} \neg p$. This should mean that it is verified at i that p is poor or neither poor or rich. But this is the case iff r is not verified at i , which means that p is not falsified at i .

5. PSLTL can be regarded as a four-valued logic. The reason is presented as follows. For any $i \in \omega$, any $\hat{d} \in \text{SE}$ and any formula α , we can take one of the following four cases:

$$(a) \alpha \text{ is verified at } i, \text{ i.e., } (M, i) \models^{+\hat{d}} \alpha,$$

$$(b) \alpha \text{ is falsified at } i, \text{ i.e., } (M, i) \models^{-\hat{d}} \alpha,$$

$$(c) \alpha \text{ is both verified and falsified at } i, \text{ and}$$

$$(d) \alpha \text{ is neither verified nor falsified at } i.$$

6. PSLTL is paraconsistent with respect to \sim . The reason is presented as follows. Assume a paraconsistent sequential model

$$M := (\sigma, \{I^{+\hat{d}}\}_{\hat{d} \in \text{SE}}, \{I^{+\hat{d}}\}_{\hat{d} \in \text{SE}})$$

such that $s_i \in I^{+\hat{d}}(p)$, $s_i \in I^{-\hat{d}}(p)$ and $s_i \notin I^{+\hat{d}}(q)$ for a pair of distinct propositional variables p and q . Then, $(M, i) \models^{+\hat{d}} (p \wedge \sim p) \rightarrow q$ does not hold.

Proposition 2.7. *The following formulas are valid in PSLTL: For any formulas α, β and any $b, c \in SE$,*

1. $\sim\sim\alpha \leftrightarrow \alpha$,
2. $\sim(\alpha \wedge \beta) \leftrightarrow \sim\alpha \vee \sim\beta$,
3. $\sim(\alpha \vee \beta) \leftrightarrow \sim\alpha \wedge \sim\beta$,
4. $\sim(\alpha \rightarrow \beta) \leftrightarrow \alpha \wedge \sim\beta$,
5. $\sim\neg\alpha \leftrightarrow \neg\sim\alpha$,
6. $\sim X\alpha \leftrightarrow X\sim\alpha$,
7. $\sim F\alpha \leftrightarrow G\sim\alpha$,
8. $\sim G\alpha \leftrightarrow F\sim\alpha$,
9. $[b](\alpha \circ \beta) \leftrightarrow ([b]\alpha) \circ ([b]\beta)$ where $\circ \in \{\wedge, \vee, \rightarrow\}$,
10. $[b](\dagger\alpha) \leftrightarrow \dagger([b]\alpha)$ where $\dagger \in \{\neg, \sim, X, G, F\}$,
11. $[b ; c]\alpha \leftrightarrow [b][c]\alpha$,
12. $[\bar{d}]\alpha \leftrightarrow [\hat{d}]\alpha$.

Proof. We show only the following cases.

- (5) : We show that $\sim\neg\alpha \leftrightarrow \neg\sim\alpha$ is valid in PSLTL. $(M, 0) \models^{+\emptyset} \sim\neg\alpha$ iff $(M, 0) \models^{-\emptyset} \neg\alpha$ iff $(M, 0) \not\models^{-\emptyset} \alpha$ iff $(M, 0) \not\models^{+\emptyset} \sim\alpha$ iff $(M, 0) \models^{+\emptyset} \neg\sim\alpha$.
- (8) : We show that $\sim G\alpha \leftrightarrow F\sim\alpha$ is valid in PSLTL. $(M, 0) \models^{+\emptyset} \sim G\alpha$ iff $(M, 0) \models^{-\emptyset} G\alpha$ iff $\exists j \geq i[(M, 0) \models^{-\emptyset} \alpha]$ iff $\exists j \geq i[(M, 0) \models^{+\emptyset} \sim\alpha]$ iff $(M, 0) \models^{+\emptyset} F\sim\alpha$.
- (10) : We show only the case that $[b](\sim\alpha) \leftrightarrow \sim([b]\alpha)$ is valid in PSLTL. $(M, 0) \models^{+\emptyset} [b](\sim\alpha)$ iff $(M, 0) \models^{+b} \sim\alpha$ iff $(M, 0) \models^{-b} \alpha$ iff $(M, 0) \models^{-\emptyset} [b]\alpha$ iff $(M, 0) \models^{+\emptyset} \sim([b]\alpha)$.

□

3. Proof systems

3.1. LT_ω

Greek capital letters Γ, Δ, \dots are used to represent finite (possibly empty) sets of formulas. An expression $X^i\alpha$ for any $i \in \omega$ is defined inductively by $X^0\alpha \equiv \alpha$ and $X^{n+1}\alpha \equiv X^n X\alpha$. An expression $X\Gamma$ is used to represent the set $\{X\gamma \mid \gamma \in \Gamma\}$. 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 [26] for LTL is presented below.

Definition 3.1 (LT_ω). The initial sequents of LT_ω are of the form: for any propositional variable p ,

$$X^i p \Rightarrow X^i p.$$

The structural rules of LT_ω are of the form:

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

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

The logical inference rules of LT_ω are of the form:

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

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

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

$$\begin{array}{c}
\frac{\Gamma \Rightarrow \Delta, X^i \alpha}{\Gamma \Rightarrow \Delta, X^i(\alpha \vee \beta)} \text{ (\Vright1)} \quad \frac{\Gamma \Rightarrow \Delta, X^i \beta}{\Gamma \Rightarrow \Delta, X^i(\alpha \vee \beta)} \text{ (\Vright2)} \\
\frac{\Gamma \Rightarrow \Delta, X^i \alpha}{X^i \neg \alpha, \Gamma \Rightarrow \Delta} \text{ (\negleft)} \quad \frac{X^i \alpha, \Gamma \Rightarrow \Delta}{\Gamma \Rightarrow \Delta, X^i \neg \alpha} \text{ (\negright)} \\
\frac{X^{i+k} \alpha, \Gamma \Rightarrow \Delta}{X^i G \alpha, \Gamma \Rightarrow \Delta} \text{ (Gleft)} \quad \frac{\{ \Gamma \Rightarrow \Delta, X^{i+j} \alpha \}_{j \in \omega}}{\Gamma \Rightarrow \Delta, X^i G \alpha} \text{ (Gright)} \\
\frac{\{ X^{i+j} \alpha, \Gamma \Rightarrow \Delta \}_{j \in \omega}}{X^i F \alpha, \Gamma \Rightarrow \Delta} \text{ (Fleft)} \quad \frac{\Gamma \Rightarrow \Delta, X^{i+k} \alpha}{\Gamma \Rightarrow \Delta, X^i F \alpha} \text{ (Fright)}.
\end{array}$$

Some remarks on LT_ω are addressed as follows.

1. The rules (Gright) and (Fleft) have infinite premises.
2. The following rule is admissible in cut-free LT_ω :

$$\frac{\Gamma \Rightarrow \Delta}{X\Gamma \Rightarrow X\Delta} \text{ (reguX)}.$$

3. The sequents of the form: $X^i \alpha \Rightarrow X^i \alpha$ for any formula α are provable in cut-free LT_ω . This fact can be proved by induction on the complexity of α .
4. The cut-elimination and completeness theorems for LT_ω were proved by Kawai [26].

3.2. SLT_ω

An expression $[b]\Gamma$ is used to represent the set $\{[b]\gamma \mid \gamma \in \Gamma\}$. The symbol K is used to represent the set $\{X\} \cup \{[b] \mid b \in SE\}$, and the symbol K^* is used to represent the set of all words of finite length of the alphabet K . For example, $X^i \overline{[b]} X^j \overline{[c]}$ 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 3.2 (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 3.1.

The logical inference rules of SLT_ω are of the form:

$$\begin{array}{c}
\frac{\Gamma \Rightarrow \Sigma, \sharp\alpha \quad \sharp\beta, \Delta \Rightarrow \Pi}{\sharp(\alpha \rightarrow \beta), \Gamma, \Delta \Rightarrow \Sigma, \Pi} (\rightarrow\text{left}^s) \quad \frac{\sharp\alpha, \Gamma \Rightarrow \Delta, \sharp\beta}{\Gamma \Rightarrow \Delta, \sharp(\alpha \rightarrow \beta)} (\rightarrow\text{right}^s) \\
\\
\frac{\sharp\alpha, \Gamma \Rightarrow \Delta}{\sharp(\alpha \wedge \beta), \Gamma \Rightarrow \Delta} (\wedge\text{left}1^s) \quad \frac{\sharp\beta, \Gamma \Rightarrow \Delta}{\sharp(\alpha \wedge \beta), \Gamma \Rightarrow \Delta} (\wedge\text{left}2^s) \\
\frac{\Gamma \Rightarrow \Delta, \sharp\alpha \quad \Gamma \Rightarrow \Delta, \sharp\beta}{\Gamma \Rightarrow \Delta, \sharp(\alpha \wedge \beta)} (\wedge\text{right}^s) \quad \frac{\sharp\alpha, \Gamma \Rightarrow \Delta \quad \sharp\beta, \Gamma \Rightarrow \Delta}{\sharp(\alpha \vee \beta), \Gamma \Rightarrow \Delta} (\vee\text{left}^s) \\
\frac{\Gamma \Rightarrow \Delta, \sharp\alpha}{\Gamma \Rightarrow \Delta, \sharp(\alpha \vee \beta)} (\vee\text{right}1^s) \quad \frac{\Gamma \Rightarrow \Delta, \sharp\beta}{\Gamma \Rightarrow \Delta, \sharp(\alpha \vee \beta)} (\vee\text{right}2^s) \\
\frac{\Gamma \Rightarrow \Delta, \sharp\alpha}{\sharp\neg\alpha, \Gamma \Rightarrow \Delta} (\neg\text{left}^s) \quad \frac{\sharp\alpha, \Gamma \Rightarrow \Delta}{\Gamma \Rightarrow \Delta, \sharp\neg\alpha} (\neg\text{right}^s) \\
\frac{\sharp X^k\alpha, \Gamma \Rightarrow \Delta}{\sharp G\alpha, \Gamma \Rightarrow \Delta} (\text{Gleft}^s) \quad \frac{\{\Gamma \Rightarrow \Delta, \sharp X^j\alpha\}_{j \in \omega}}{\Gamma \Rightarrow \Delta, \sharp G\alpha} (\text{Gright}^s) \\
\frac{\{\sharp X^j\alpha, \Gamma \Rightarrow \Delta\}_{j \in \omega}}{\sharp F\alpha, \Gamma \Rightarrow \Delta} (\text{Fleft}^s) \quad \frac{\Gamma \Rightarrow \Delta, \sharp X^k\alpha}{\Gamma \Rightarrow \Delta, \sharp F\alpha} (\text{Fright}^s) \\
\frac{\sharp[b]X\alpha, \Gamma \Rightarrow \Delta}{\sharp X[b]\alpha, \Gamma \Rightarrow \Delta} (\text{Xleft}) \quad \frac{\Gamma \Rightarrow \Delta, \sharp[b]X\alpha}{\Gamma \Rightarrow \Delta, \sharp X[b]\alpha} (\text{Xright}).
\end{array}$$

The sequence inference rules of SLT_ω are of the form:

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

Some remarks on SLT_ω are addressed as follows.

1. 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 .
2. The following rules are admissible in cut-free SLT_ω :

$$\begin{array}{c}
\frac{\Gamma \Rightarrow \Delta}{X\Gamma \Rightarrow X\Delta} (\text{reguX}) \quad \frac{\Gamma \Rightarrow \Delta}{[d]\Gamma \Rightarrow [d]\Delta} (\text{regu}[\bar{d}]) \\
\frac{\sharp X[b]\alpha, \Gamma \Rightarrow \Delta}{\sharp[b]X\alpha, \Gamma \Rightarrow \Delta} (\text{Xleft}^{-1}) \quad \frac{\Gamma \Rightarrow \Delta, \sharp X[b]\alpha}{\Gamma \Rightarrow \Delta, \sharp[b]X\alpha} (\text{Xright}^{-1}).
\end{array}$$

3.3. PSLT_ω

A sequent calculus PSLT_ω for PSLTL is introduced below.

Definition 3.3 (PSLT_ω). PSLT_ω is obtained from SLT_ω by adding the initial sequents of the form: for any propositional variable p ,

$$\# \sim p \Rightarrow \# \sim p,$$

and adding the logical and sequence inference rules of the form:

$$\begin{array}{c} \frac{\# \sim \alpha, \Gamma \Rightarrow \Delta}{\sim \# \alpha, \Gamma \Rightarrow \Delta} (\sim \# \text{left}) \quad \frac{\Gamma \Rightarrow \Delta, \# \sim \alpha}{\Gamma \Rightarrow \Delta, \sim \# \alpha} (\sim \# \text{right}) \\ \\ \frac{\# \alpha, \Gamma \Rightarrow \Delta}{\# \sim \sim \alpha, \Gamma \Rightarrow \Delta} (\sim \sim \text{left}) \quad \frac{\Gamma \Rightarrow \Delta, \# \alpha}{\Gamma \Rightarrow \Delta, \# \sim \sim \alpha} (\sim \sim \text{right}) \\ \\ \frac{\# \alpha, \Gamma \Rightarrow \Delta}{\# \sim (\alpha \rightarrow \beta), \Gamma \Rightarrow \Delta} (\sim \rightarrow \text{left1}) \quad \frac{\# \sim \beta, \Gamma \Rightarrow \Delta}{\# \sim (\alpha \rightarrow \beta), \Gamma \Rightarrow \Delta} (\sim \rightarrow \text{left2}) \\ \\ \frac{\Gamma \Rightarrow \Delta, \# \alpha \quad \Gamma \Rightarrow \Delta, \# \sim \beta}{\Gamma \Rightarrow \Delta, \# \sim (\alpha \rightarrow \beta)} (\sim \rightarrow \text{right}) \quad \frac{\# \sim \alpha, \Gamma \Rightarrow \Delta \quad \# \sim \beta, \Gamma \Rightarrow \Delta}{\# \sim (\alpha \wedge \beta), \Gamma \Rightarrow \Delta} (\sim \wedge \text{left}) \\ \\ \frac{\Gamma \Rightarrow \Delta, \# \sim \alpha}{\Gamma \Rightarrow \Delta, \# \sim (\alpha \wedge \beta)} (\sim \wedge \text{right1}) \quad \frac{\Gamma \Rightarrow \Delta, \# \sim \beta}{\Gamma \Rightarrow \Delta, \# \sim (\alpha \wedge \beta)} (\sim \wedge \text{right2}) \\ \\ \frac{\# \sim \alpha, \Gamma \Rightarrow \Delta}{\# \sim (\alpha \vee \beta), \Gamma \Rightarrow \Delta} (\sim \vee \text{left1}) \quad \frac{\# \sim \beta, \Gamma \Rightarrow \Delta}{\# \sim (\alpha \vee \beta), \Gamma \Rightarrow \Delta} (\sim \vee \text{left2}) \\ \\ \frac{\Gamma \Rightarrow \Delta, \# \sim \alpha \quad \Gamma \Rightarrow \Delta, \# \sim \beta}{\Gamma \Rightarrow \Delta, \# \sim (\alpha \vee \beta)} (\sim \vee \text{right}) \\ \\ \frac{\Gamma \Rightarrow \Delta, \# \sim \alpha}{\# \sim \neg \alpha, \Gamma \Rightarrow \Delta} (\sim \neg \text{left}) \quad \frac{\# \sim \alpha, \Gamma \Rightarrow \Delta}{\Gamma \Rightarrow \Delta, \# \sim \neg \alpha} (\sim \neg \text{right}) \\ \\ \frac{\{ \# X^j \sim \alpha, \Gamma \Rightarrow \Delta \}_{j \in \omega}}{\# \sim G \alpha, \Gamma \Rightarrow \Delta} (\sim G \text{left}) \quad \frac{\Gamma \Rightarrow \Delta, \# X^k \sim \alpha}{\Gamma \Rightarrow \Delta, \# \sim G \alpha} (\sim G \text{right}) \\ \\ \frac{\# X^k \sim \alpha, \Gamma \Rightarrow \Delta}{\# \sim F \alpha, \Gamma \Rightarrow \Delta} (\sim F \text{left}) \quad \frac{\{ \Gamma \Rightarrow \Delta, \# X^j \sim \alpha \}_{j \in \omega}}{\Gamma \Rightarrow \Delta, \# \sim F \alpha} (\sim F \text{right}) \\ \\ \frac{\# \sim [b][c] \alpha, \Gamma \Rightarrow \Delta}{\# \sim [b ; c] \alpha, \Gamma \Rightarrow \Delta} (\sim ; \text{left}) \quad \frac{\Gamma \Rightarrow \Delta, \# \sim [b][c] \alpha}{\Gamma \Rightarrow \Delta, \# \sim [b ; c] \alpha} (\sim ; \text{right}). \end{array}$$

Proposition 3.4. *The following rules are admissible in cut-free PSLT_ω :*

$$\begin{array}{c} \frac{\Gamma \Rightarrow \Delta}{\text{X}\Gamma \Rightarrow \text{X}\Delta} \text{ (reguX)} \quad \frac{\Gamma \Rightarrow \Delta}{[\overline{d}]\Gamma \Rightarrow [\overline{d}]\Delta} \text{ (regu}[\overline{d}]\text{)} \\ \frac{\sharp\text{X}[b]\alpha, \Gamma \Rightarrow \Delta}{\sharp[b]\text{X}\alpha, \Gamma \Rightarrow \Delta} \text{ (Xleft}^{-1}\text{)} \quad \frac{\Gamma \Rightarrow \Delta, \sharp\text{X}[b]\alpha}{\Gamma \Rightarrow \Delta, \sharp[b]\text{X}\alpha} \text{ (Xright}^{-1}\text{)} \\ \frac{\sim\sharp\alpha, \Gamma \Rightarrow \Delta}{\sharp\sim\alpha, \Gamma \Rightarrow \Delta} \text{ (}\sim\sharp\text{left}^{-1}\text{)} \quad \frac{\Gamma \Rightarrow \Delta, \sim\sharp\alpha}{\Gamma \Rightarrow \Delta, \sharp\sim\alpha} \text{ (}\sim\sharp\text{right}^{-1}\text{)} \end{array}$$

Proof. We show only the case for $(\text{regu}[\overline{d}])$ by induction on the proofs P of $\Gamma \Rightarrow \Delta$ in cut-free PSLT_ω . We distinguish the cases according to the last inference of P , and show some cases.

1. Case $(\sim; \text{right})$: The last inference of P is of the form:

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

By induction hypothesis, we obtain the required fact:

$$\frac{\begin{array}{c} \vdots \\ \overline{[d]}\Gamma \Rightarrow \overline{[d]}\Delta, \overline{[d]}\sharp\sim[\overline{k}][b][c]\alpha \end{array}}{\overline{[d]}\Gamma \Rightarrow \overline{[d]}\Delta, \overline{[d]}\sharp\sim[\overline{k}][b ; c]\alpha} \text{ (}\sim; \text{right)}$$

where $\overline{[d]}\sharp$ can be regarded also as \sharp since $\overline{[d]}\sharp$ is in K^* .

2. Case $(\rightarrow \text{left}^s)$: The last inference of P is of the form:

$$\frac{\Gamma \Rightarrow \Sigma, \sharp\alpha \quad \sharp\beta, \Delta \Rightarrow \Pi}{\sharp(\alpha \rightarrow \beta), \Gamma, \Delta \Rightarrow \Sigma, \Pi} \text{ (}\rightarrow \text{left}^s\text{)}.$$

By induction hypothesis, we obtain the required fact:

$$\frac{\begin{array}{c} \vdots \\ \overline{[d]}\Gamma \Rightarrow \overline{[d]}\Sigma, \overline{[d]}\sharp\alpha \end{array} \quad \begin{array}{c} \vdots \\ \overline{[d]}\sharp\beta, \overline{[d]}\Delta \Rightarrow \overline{[d]}\Pi \end{array}}{\overline{[d]}\sharp(\alpha \rightarrow \beta), \overline{[d]}\Gamma, \overline{[d]}\Delta \Rightarrow \overline{[d]}\Sigma, \overline{[d]}\Pi} \text{ (}\rightarrow \text{left}^s\text{)}$$

where $\overline{[d]}\sharp$ can be regarded also as \sharp since $\overline{[d]}\sharp$ is in K^* . \square

Proposition 3.5. *The sequents of the form $\sharp\alpha \Rightarrow \sharp\alpha$ for any formula α are provable in cut-free PSLT_ω .*

Proof. By induction on the complexity of α . We show only some cases.

1. Case $\alpha \equiv \sim\beta$: We can obtain $\text{PSLT}_\omega \vdash \sim\beta \Rightarrow \sim\beta$ by induction on the complexity of β . Then, we obtain the required fact $\text{PSLT}_\omega \vdash \sharp\sim\beta \Rightarrow \sharp\sim\beta$ by:

$$\begin{array}{c} \vdots \\ \sim\beta \Rightarrow \sim\beta \\ \vdots \quad (*) \\ \sharp\sim\beta \Rightarrow \sharp\sim\beta \end{array}$$

where $(*)$ represents some applications of inference rules in PSLT_ω including some admissible inference rules presented in Proposition 3.4.

2. Case $\alpha \equiv [b]\beta$: We can obtain $\text{PSLT}_\omega \vdash [b]\beta \Rightarrow [b]\beta$ by induction on the complexity of β . To show this, we use the admissible rule $(\overline{[d]}\text{regu})$ presented in Proposition 3.4. Then, we obtain the required fact $\text{PSLT}_\omega \vdash \sharp[b]\beta \Rightarrow \sharp[b]\beta$ by:

$$\begin{array}{c} \vdots \\ [b]\beta \Rightarrow [b]\beta \\ \vdots \quad (*) \\ \sharp[b]\beta \Rightarrow \sharp[b]\beta \end{array}$$

where $(*)$ represents some applications of inference rules in PSLT_ω including some admissible inference rules presented in Proposition 3.4. \square

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

Proposition 3.6. *The following sequents are provable in PSLT_ω : For any formulas α, β and any $b, c \in \text{SE}$,*

1. $\sim\sim\alpha \Leftrightarrow \alpha$,
2. $\sim(\alpha \wedge \beta) \Leftrightarrow \sim\alpha \vee \sim\beta$,
3. $\sim(\alpha \vee \beta) \Leftrightarrow \sim\alpha \wedge \sim\beta$,

4. $\sim(\alpha \rightarrow \beta) \Leftrightarrow \alpha \wedge \sim\beta$,
5. $\sim\neg\alpha \Leftrightarrow \neg\sim\alpha$,
6. $\sim X\alpha \Leftrightarrow X\sim\alpha$,
7. $\sim F\alpha \Leftrightarrow G\sim\alpha$,
8. $\sim G\alpha \Leftrightarrow F\sim\alpha$,
9. $[b](\alpha \circ \beta) \Leftrightarrow ([b]\alpha) \circ ([b]\beta)$ where $\circ \in \{\wedge, \vee, \rightarrow\}$,
10. $[b](\#\alpha) \Leftrightarrow \#[b]\alpha$ where $\# \in \{\neg, \sim, X, G, F\}$,
11. $[b; c]\alpha \Leftrightarrow [b][c]\alpha$,
12. $[\bar{d}]\alpha \Leftrightarrow [\hat{d}]\alpha$.

Proof. We show only the following cases.

(5) :

$$\frac{\frac{\frac{\vdots}{\sim\alpha \Rightarrow \sim\alpha}}{\Rightarrow \neg\sim\alpha, \sim\alpha} (\neg\text{right}^s)}{\sim\neg\alpha \Rightarrow \sim\neg\alpha} (\sim\neg\text{left}) \quad \frac{\frac{\frac{\vdots}{\sim\alpha \Rightarrow \sim\alpha}}{\Rightarrow \sim\neg\alpha, \sim\alpha} (\sim\neg\text{right})}{\sim\neg\alpha \Rightarrow \sim\neg\alpha} (\sim\neg\text{left}^s)}$$

(8) :

$$\frac{\frac{\frac{\vdots}{\{X^j \sim\alpha \Rightarrow X^j \sim\alpha\}_{j \in \omega}} (\text{Fright}^s)}{\{X^j \sim\alpha \Rightarrow F\sim\alpha\}_{j \in \omega}} (\sim\text{Gleft})}{\sim G\alpha \Rightarrow F\sim\alpha} \quad \frac{\frac{\frac{\vdots}{\{X^j \sim\alpha \Rightarrow X^j \sim\alpha\}_{j \in \omega}} (\sim\text{Gright})}{\{X^j \sim\alpha \Rightarrow \sim G\alpha\}_{j \in \omega}} (\text{Fleft}^s)}{F\sim\alpha \Rightarrow \sim G\alpha}$$

(10) : We show the cases $\# \equiv \sim$ and $\# \equiv \neg$ below.

Case ($\# \equiv \sim$):

$$\frac{\frac{\frac{\vdots}{[b](\sim\alpha) \Rightarrow [b](\sim\alpha)}}{[b](\sim\alpha) \Rightarrow \sim([b]\alpha)} (\sim\#\text{right}) \quad \frac{\frac{\frac{\vdots}{[b](\sim\alpha) \Rightarrow [b](\sim\alpha)}}{\sim([b]\alpha) \Rightarrow [b](\sim\alpha)} (\sim\#\text{left})$$

Case ($\# \equiv \neg$):

$$\frac{\frac{\frac{\vdots}{[b]\alpha \Rightarrow [b]\alpha} (\neg\text{-right}^s)}{\Rightarrow \neg([b]\alpha), [b]\alpha} (\neg\text{-left}^s)}{[b](\neg\alpha) \Rightarrow \neg([b]\alpha)} (\neg\text{-left}^s) \qquad \frac{\frac{\frac{\vdots}{[b]\alpha \Rightarrow [b]\alpha} (\neg\text{-right}^s)}{[b]\alpha, \neg([b]\alpha) \Rightarrow} (\neg\text{-right}^s)}{\neg([b]\alpha) \Rightarrow [b](\neg\alpha)} (\neg\text{-left}^s)}$$

□

3.4. Hilbert-type systems

Hilbert-type axiomatic systems H-PSLTL, H-PLTL, H-SLTL and H-LTL for PSLTL, PLTL, SLTL and LTL, respectively, are presented below. For typical Hilbert-style axiomatizations of LTL, see, e.g., [9, 3]. H-LTL is from [3].

Definition 3.7 (H-PSLTL, H-PLTL, H-SLTL and H-LTL). Let F be defined by $F\alpha := \neg G\neg\alpha$.

H-LTL is obtained from the axiom schemes and inference rules of the propositional classical logic by adding the inference rules and axiom schemes of the form:

$$\frac{\alpha}{X\beta} \text{ (X)} \qquad \frac{\alpha}{G\beta} \text{ (G)}$$

1. $X(\alpha \rightarrow \beta) \rightarrow (X\alpha \rightarrow X\beta)$,
2. $G(\alpha \rightarrow \beta) \rightarrow (G\alpha \rightarrow G\beta)$,
3. $\neg X\alpha \leftrightarrow X\neg\alpha$,
4. $G\alpha \rightarrow \alpha$,
5. $G\alpha \rightarrow XG\alpha$,
6. $\alpha \rightarrow (G(\alpha \rightarrow X\alpha) \rightarrow G\alpha)$.

H-PLTL is obtained from H-LTL by adding the axioms schemes of the form:

1. $\sim\sim\alpha \leftrightarrow \alpha$,
2. $\sim\neg\alpha \leftrightarrow \neg\sim\alpha$,
3. $\sim(\alpha \wedge \beta) \leftrightarrow \sim\alpha \vee \sim\beta$,
4. $\sim(\alpha \vee \beta) \leftrightarrow \sim\alpha \wedge \sim\beta$,
5. $\sim(\alpha \rightarrow \beta) \leftrightarrow \alpha \wedge \sim\beta$,
6. $\sim X\alpha \leftrightarrow X\sim\alpha$,
7. $\sim F\alpha \leftrightarrow G\sim\alpha$,
8. $\sim G\alpha \leftrightarrow F\sim\alpha$.

H-SLTL is obtained from H-LTL by adding the inference rule and axioms schemes of the form: For any $b, c \in \text{SE}$,

$$\frac{\alpha}{[b]\beta} \text{ ([b])}$$

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

H-PSLTL is obtained from H-PLTL by adding the above presented inference rule and axioms schemes of H-SLTL.

4. Main results

4.1. Semantical embedding and decidability

Definition 4.1 (Translation from SLTL into LTL). Let Φ be a non-empty set of propositional variables and $\Phi^{\hat{d}}$ be the set $\{p^{\hat{d}} \mid p \in \Phi\}$ ($\hat{d} \in \text{SE}$) 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 . 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., $f(p) = p \in \Phi^{\emptyset}$,
2. $f(\#(\alpha \circ \beta)) := f(\#\alpha) \circ f(\#\beta)$ where $\circ \in \{\wedge, \vee, \rightarrow\}$,
3. $f(\#\dagger\alpha) := \dagger f(\#\alpha)$ where $\dagger \in \{\neg, X, G, F\}$,
4. $f(\#[\overline{d}][b]\alpha) := f(\#[\hat{d} ; b]\alpha)$.

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

Some remarks on the mapping f in Definition 4.1 are addressed as follows.

1. We can obtain the condition:

$$f(\#[\overline{d}][b][c]\alpha) = f(\#[\overline{d}][b ; c]\alpha)$$

by using the condition 4 repeatedly:

$$\begin{aligned} & f(\#[\overline{d}][b][c]\alpha) \\ &= f(\#[\hat{d} ; b][c]\alpha) \\ &= f(\#[\hat{d} ; b ; c]) \\ &= f(\#[\overline{d}][b ; c]\alpha). \end{aligned}$$

2. The following is an example of the translation of the formula $[b_1]([b_2 ; b_3]p_1 \wedge ([b_4]p_2 \vee [b_5 ; b_6]p_3))$ where p_1, p_2 and p_3 are distinct propositional variables, and b_1, b_2, b_3, b_4, b_5 and b_6 are distinct atomic sequences.

$$\begin{aligned} & f([b_1]([b_2 ; b_3]p_1 \wedge ([b_4]p_2 \vee [b_5 ; b_6]p_3))) \\ &= f([b_1][b_2 ; b_3]p_1) \wedge f([b_1]([b_4]p_2 \vee [b_5 ; b_6]p_3)) \\ &= f([b_1][b_2 ; b_3]p_1) \wedge (f([b_1][b_4]p_2) \vee f([b_1][b_5 ; b_6]p_3)) \\ &= f([b_1 ; b_2 ; b_3]p_1) \wedge (f([b_1 ; b_4]p_2) \vee f([b_1 ; b_5 ; b_6]p_3)) \\ &= p_1^{b_1 ; b_2 ; b_3} \wedge (p_2^{b_1 ; b_4} \vee p_3^{b_1 ; b_5 ; b_6}). \end{aligned}$$

Proposition 4.2 ([14, 23]). *Let f be the mapping defined in Definition 4.1.*

1. (Semantical embedding): *For any formula α , α is valid in SLTL iff $f(\alpha)$ is valid in LTL.*

2. (*Syntactical embedding*): For any sets Γ and Δ of formulas in \mathcal{L}^s ,
 - (a) $\text{SLT}_\omega \vdash \Gamma \Rightarrow \Delta$ iff $\text{LT}_\omega \vdash f(\Gamma) \Rightarrow f(\Delta)$,
 - (b) $\text{SLT}_\omega - (\text{cut}) \vdash \Gamma \Rightarrow \Delta$ iff $\text{LT}_\omega - (\text{cut}) \vdash f(\Gamma) \Rightarrow f(\Delta)$.
3. (*Cut-elimination*): The rule (cut) is admissible in cut-free SLT_ω .
4. (*Completeness*): For any formula α , $\text{SLT}_\omega \vdash \Rightarrow \alpha$ iff α is valid in SLTL.

We now introduce a translation of PSLTL into SLTL, and by using this translation, we show some theorems for embedding PSLTL into SLTL. A similar translation has been used by Vorob'ev [31], Gurevich [10], and Rautenberg [30] to embed Nelson's three-valued constructive logic [1, 27] into intuitionistic logic.

Definition 4.3 (Translation from PSLTL into SLTL). Let Φ be a non-empty set of propositional variables and Φ' be the set $\{p' \mid p \in \Phi\}$ of propositional variables. The language \mathcal{L}^{ps} (the set of formulas) of PSLTL is defined using Φ , \sim , \rightarrow , \wedge , \vee , \neg , X, F, G and $[b]$. The language \mathcal{L}^s of SLTL is obtained from \mathcal{L}^{ps} by adding Φ' and deleting \sim .

A mapping g from \mathcal{L}^{ps} to \mathcal{L}^s is defined by

1. for any $p \in \Phi$, $g(p) := p$ and $g(\sim p) := p' \in \Phi'$,
2. $g(\alpha \circ \beta) := g(\alpha) \circ g(\beta)$ where $\circ \in \{\wedge, \vee, \rightarrow\}$,
3. $g(\dagger \alpha) := \dagger g(\alpha)$ where $\dagger \in \{\neg, X, F, G, [b]\}$,
4. $g(\sim \sim \alpha) := g(\alpha)$,
5. $g(\sim \dagger \alpha) := \dagger g(\sim \alpha)$ where $\dagger \in \{\neg, X, [b]\}$,
6. $g(\sim(\alpha \wedge \beta)) := g(\sim \alpha) \vee g(\sim \beta)$,
7. $g(\sim(\alpha \vee \beta)) := g(\sim \alpha) \wedge g(\sim \beta)$,
8. $g(\sim(\alpha \rightarrow \beta)) := g(\alpha) \wedge g(\sim \beta)$,
9. $g(\sim F\alpha) := Gg(\sim \alpha)$,
10. $g(\sim G\alpha) := Fg(\sim \alpha)$.

We have: $g(\sharp\alpha) = \sharp g(\alpha)$ for any formula α and any $\sharp \in K^*$.

Lemma 4.4. *Let g be the mapping defined in Definition 4.3, and S be a non-empty set of states. For any paraconsistent sequential model $M := (\sigma, \{I^{+\hat{d}}\}_{\hat{d} \in \text{SE}}, \{I^{-\hat{d}}\}_{\hat{d} \in \text{SE}})$ of PSLTL, any satisfaction relations $\models^{*\hat{d}}$ ($* \in \{+, -\}, \hat{d} \in \text{SE}$) on M , and any state s_i in σ , we can construct a sequential model $N := (\sigma, \{I^{\hat{d}}\}_{\hat{d} \in \text{SE}})$ of SLTL and satisfaction relations $\models^{\hat{d}}$ on N such that for any formula α in \mathcal{L}^{ps} ,*

1. $(M, i) \models^{+\hat{d}} \alpha$ iff $(N, i) \models^{\hat{d}} g(\alpha)$.
2. $(M, i) \models^{-\hat{d}} \alpha$ iff $(N, i) \models^{\hat{d}} g(\sim\alpha)$.

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

$I^{+\hat{d}}$ and $I^{-\hat{d}}$ are mappings from Φ to the power set of S .

Suppose that N is a sequential model $(\sigma, \{I^{\hat{d}}\}_{\hat{d} \in \text{SE}})$ where

$I^{\hat{d}}$ are mappings from $\Phi \cup \Phi'$ to the power set of S .

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

1. $s_i \in I^{+\hat{d}}(p)$ iff $s_i \in I^{\hat{d}}(p)$,
2. $s_i \in I^{-\hat{d}}(p)$ iff $s_i \in I^{\hat{d}}(p')$.

The lemma is then proved by (simultaneous) induction on the complexity of α .

- Base step:

Case $\alpha \equiv p \in \Phi$: For (1), we obtain: $(M, i) \models^{+\hat{d}} p$ iff $s_i \in I^{+\hat{d}}(p)$ iff $s_i \in I^{\hat{d}}(p)$ iff $(N, i) \models^{\hat{d}} p$ iff $(N, i) \models^{\hat{d}} g(p)$ (by the definition of g). For (2), we obtain: $(M, i) \models^{-\hat{d}} p$ iff $s_i \in I^{-\hat{d}}(p)$ iff $s_i \in I^{\hat{d}}(p')$ iff $(N, i) \models^{\hat{d}} p'$ iff $(N, i) \models^{\hat{d}} g(\sim p)$ (by the definition of g).

- Induction step: We show some cases.

1. Case $\alpha \equiv \beta \wedge \gamma$: For (1), 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 $(N, i) \models^{\hat{d}} g(\beta)$ and $(N, i) \models^{\hat{d}} g(\gamma)$ (by induction hypothesis for 1) iff $(N, i) \models^{\hat{d}} g(\beta) \wedge g(\gamma)$ iff $(N, i) \models^{\hat{d}} g(\beta \wedge \gamma)$ (by the definition of g). For (2), we obtain: $(M, i) \models^{-\hat{d}} \beta \wedge \gamma$ iff $(M, i) \models^{-\hat{d}} \beta$ or $(M, i) \models^{-\hat{d}} \gamma$ iff $(N, i) \models^{\hat{d}} g(\sim\beta)$ or $(N, i) \models^{\hat{d}} g(\sim\gamma)$ (by induction hypothesis for 2) iff $(N, i) \models^{\hat{d}} g(\sim\beta) \vee g(\sim\gamma)$ iff $(N, i) \models^{\hat{d}} g(\sim(\beta \wedge \gamma))$ (by the definition of g).
2. Case $\alpha \equiv \beta \rightarrow \gamma$: For (1), 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 $(N, i) \models^{\hat{d}} g(\beta)$ implies $(N, i) \models^{\hat{d}} g(\gamma)$ (by induction hypothesis for 1) iff $(N, i) \models^{\hat{d}} g(\beta) \rightarrow g(\gamma)$ iff $(N, i) \models^{\hat{d}} g(\beta \rightarrow \gamma)$ (by the definition of g). For (2), we obtain: $(M, i) \models^{-\hat{d}} \beta \rightarrow \gamma$ iff $(M, i) \models^{+\hat{d}} \beta$ and $(M, i) \models^{-\hat{d}} \gamma$ iff $(N, i) \models^{\hat{d}} g(\beta)$ and $(N, i) \models^{\hat{d}} g(\sim\gamma)$ (by induction hypothesis for 1 and 2) iff $(N, i) \models^{\hat{d}} g(\beta) \wedge g(\sim\gamma)$ iff $(N, i) \models^{\hat{d}} g(\sim(\beta \rightarrow \gamma))$ (by the definition of g).
3. Case $\alpha \equiv \sim\beta$: For (1), we obtain: $(M, i) \models^{+\hat{d}} \sim\beta$ iff $(M, i) \models^{-\hat{d}} \beta$ iff $(N, i) \models^{\hat{d}} g(\sim\beta)$ (by induction hypothesis for 2). For (2), we obtain: $(M, i) \models^{-\hat{d}} \sim\beta$ iff $(M, i) \models^{+\hat{d}} \beta$ iff $(N, i) \models^{\hat{d}} g(\beta)$ (by induction hypothesis for 1) iff $(N, i) \models^{\hat{d}} g(\sim\sim\beta)$ (by the definition of g).
4. Case $\alpha \equiv X\beta$: For (1), we obtain: $(M, i) \models^{+\hat{d}} X\beta$ iff $(M, i+1) \models^{+\hat{d}} \beta$ iff $(N, i+1) \models^{\hat{d}} g(\beta)$ (by induction hypothesis for 1) iff $(N, i) \models^{\hat{d}} Xg(\beta)$ iff $(N, i) \models^{\hat{d}} g(X\beta)$ (by the definition of g). For (2), we obtain: $(M, i) \models^{-\hat{d}} X\beta$ iff $(M, i+1) \models^{-\hat{d}} \beta$ iff $(N, i+1) \models^{\hat{d}} g(\sim\beta)$ (by induction hypothesis for 2) iff $(N, i) \models^{\hat{d}} Xg(\sim\beta)$ iff $(N, i) \models^{\hat{d}} g(\sim X\beta)$ (by the definition of g).
5. Case $\alpha \equiv G\beta$: For (1), we obtain: $(M, i) \models^{+\hat{d}} G\beta$ iff $\forall j \geq i [(M, j) \models^{+\hat{d}} \beta]$ iff $\forall j \geq i [(N, j) \models^{\hat{d}} g(\beta)]$ (by induction hypothesis for 1) iff $(N, i) \models^{\hat{d}} Gg(\beta)$ iff $(N, i) \models^{\hat{d}} g(G\beta)$ (by the definition of g). For (2), we obtain: $(M, i) \models^{-\hat{d}} G\beta$ iff $\exists j \geq i [(M, j) \models^{-\hat{d}} \beta]$ iff $\exists j \geq i [(N, j) \models^{\hat{d}} g(\sim\beta)]$ (by induction hypothesis for 2) iff $(N, i) \models^{\hat{d}} Fg(\sim\beta)$ iff $(N, i) \models^{\hat{d}} g(\sim G\beta)$ (by the definition of g).
6. Case $\alpha \equiv [b]\beta$: For (1), we obtain: $(M, i) \models^{+\hat{d}} [b]\beta$ iff $(M, i) \models^{+(\hat{d}; b)} \beta$ iff $(N, i) \models^{\hat{d}; b} g(\beta)$ (by induction hypothesis for 1) iff $(N, i) \models^{\hat{d}} [b]g(\beta)$ iff $(N, i) \models^{\hat{d}} g([b]\beta)$ (by the definition of g). For (2), we obtain: $(M, i) \models^{-\hat{d}} [b]\beta$ iff $(M, i) \models^{-(\hat{d}; b)} \beta$ iff $(N, i) \models^{\hat{d}; b} g(\sim\beta)$

(by induction hypothesis for 2) iff $(N, i) \models^{\hat{d}} [b]g(\sim\beta)$ iff $(N, i) \models^{\hat{d}} g(\sim[b]\beta)$ (by the definition of g). \square

Lemma 4.5. *Let g be the mapping defined in Definition 4.3, and S be a non-empty set of states. For any sequential model $N := (\sigma, \{I^{\hat{d}}\}_{\hat{d} \in \text{SE}})$ of SLTL and any satisfaction relations $\models^{\hat{d}}$ ($\hat{d} \in \text{SE}$) on N , and any state s_i in σ , we can construct a paraconsistent sequential model $M := (\sigma, \{I^{+\hat{d}}\}_{\hat{d} \in \text{SE}}, \{I^{-\hat{d}}\}_{\hat{d} \in \text{SE}})$ of PSLTL and satisfaction relations $\models^{*\hat{d}}$ ($* \in \{+, -\}, \hat{d} \in \text{SE}$) on M such that*

1. $(M, i) \models^{+\hat{d}} \alpha$ iff $(N, i) \models^{\hat{d}} g(\alpha)$.
2. $(M, i) \models^{-\hat{d}} \alpha$ iff $(N, i) \models^{\hat{d}} g(\sim\alpha)$.

Proof. Similar to the proof of Lemma 4.4. \square

Theorem 4.6 (Semantical embedding from PSLTL into SLTL). *Let g be the mapping defined in Definition 4.3. For any formula α ,*

α is valid in PSLTL iff $g(\alpha)$ is valid in SLTL.

Proof. By Lemmas 4.4 and 4.5. \square

Theorem 4.7 (Semantical embedding from PSLTL into LTL). *Let f and g be the mappings defined in Definitions 4.1 and 4.3, respectively. For any formula α ,*

α is valid in PSLTL iff $fg(\alpha)$ is valid in LTL.

Proof. By Proposition 4.2 (1) and Theorem 4.6. \square

Theorem 4.8 (Decidability). *PSLTL is decidable.*

Proof. By decidability of LTL, for each α , it is possible to decide if $fg(\alpha)$ is valid in LTL. Then, by Theorem 4.7, PSLTL is also decidable. \square

4.2. Syntactical embedding, cut-elimination and completeness

Theorem 4.9 (Weak syntactical embedding from PSLT_ω into SLT_ω).
 Let Γ and Δ be sets of formulas in \mathcal{L}^{ps} , and g be the mapping defined in Definition 4.3. Then:

1. If $\text{PSLT}_\omega \vdash \Gamma \Rightarrow \Delta$, then $\text{SLT}_\omega \vdash g(\Gamma) \Rightarrow g(\Delta)$.
2. If $\text{SLT}_\omega - (\text{cut}) \vdash g(\Gamma) \Rightarrow g(\Delta)$, then $\text{PSLT}_\omega - (\text{cut}) \vdash \Gamma \Rightarrow \Delta$.

Proof. • (1) : By induction on the proofs P of $\Gamma \Rightarrow \Delta$ in PSLT_ω . We distinguish the cases according to the last inference of P , and show some cases.

1. Case ($\# \sim p \Rightarrow \# \sim p$): The last inference of P is of the form: $\# \sim p \Rightarrow \# \sim p$. In this case, we obtain the required fact $\text{LT}_\omega \vdash g(\# \sim p) \Rightarrow g(\# \sim p)$, since $g(\# \sim p)$ coincides with $\# p'$ by the definition of g .
2. Case ($\sim \sim \text{left}$): The last inference of P is of the form:

$$\frac{\# \alpha, \Gamma \Rightarrow \Delta}{\# \sim \sim \alpha, \Gamma \Rightarrow \Delta} (\sim \sim \text{left}).$$

By induction hypothesis, we have the required fact:

$$\text{SLT}_\omega \vdash g(\# \alpha), g(\Gamma) \Rightarrow g(\Delta)$$

where $g(\# \alpha)$ coincides with $g(\# \sim \sim \alpha)$ by the definition of g .

3. Case ($\sim \neg \text{left}$): The last inference of P is of the form:

$$\frac{\Gamma \Rightarrow \Delta, \# \sim \alpha}{\# \sim \neg \alpha, \Gamma \Rightarrow \Delta} (\sim \neg \text{left}).$$

By induction hypothesis, we have: $\text{SLT}_\omega \vdash g(\Gamma) \Rightarrow g(\Delta), g(\# \sim \alpha)$ where $g(\# \sim \alpha)$ coincides with $\# g(\sim \alpha)$ by the definition of g . Then, we obtain:

$$\frac{\begin{array}{c} \vdots \\ g(\Gamma) \Rightarrow g(\Delta), \# g(\sim \alpha) \end{array}}{\# \neg g(\sim \alpha), g(\Gamma) \Rightarrow g(\Delta)} (\sim \neg \text{left}^s)$$

where $\# \neg g(\sim \alpha)$ coincides with $g(\# \sim \neg \alpha)$ by the definition of g .

4. Case ($\sim\rightarrow$ right): The last inference of P is of the form:

$$\frac{\Gamma \Rightarrow \Delta, \sharp\alpha \quad \Gamma \Rightarrow \Delta, \sharp\sim\beta}{\Gamma \Rightarrow \Delta, \sharp\sim(\alpha\rightarrow\beta)} (\sim\rightarrow\text{right}).$$

By induction hypothesis, we have: $\text{SLT}_\omega \vdash g(\Gamma) \Rightarrow g(\Delta), g(\sharp\alpha)$ and $\text{SLT}_\omega \vdash g(\Gamma) \Rightarrow g(\Delta), g(\sharp\sim\beta)$ where $g(\sharp\alpha)$ and $g(\sharp\sim\beta)$ respectively coincide with $\sharp g(\alpha)$ and $\sharp g(\sim\beta)$ by the definition of g . Then, we obtain:

$$\frac{\begin{array}{c} \vdots \\ g(\Gamma) \Rightarrow g(\Delta), \sharp g(\alpha) \end{array} \quad \begin{array}{c} \vdots \\ g(\Gamma) \Rightarrow g(\Delta), \sharp g(\sim\beta) \end{array}}{g(\Gamma) \Rightarrow g(\Delta), \sharp(g(\alpha) \wedge g(\sim\beta))} (\wedge\text{right})$$

where $\sharp(g(\alpha) \wedge g(\sim\beta))$ coincides with $g(\sharp\sim(\alpha\rightarrow\beta))$ by the definition of g .

5. Case (\sim Gleft): The last inference of P is of the form:

$$\frac{\{ \sharp X^j \sim \alpha, \Gamma \Rightarrow \Delta \}_{j \in \omega}}{\sharp \sim G\alpha, \Gamma \Rightarrow \Delta} (\sim\text{Gleft}).$$

By induction hypothesis, we have: $\text{SLT}_\omega \vdash g(\sharp X^j \sim \alpha), g(\Gamma) \Rightarrow g(\Delta)$ for any $j \in \omega$, where $g(\sharp X^j \sim \alpha)$ coincides with $\sharp X^j g(\sim \alpha)$ by the definition of g . Then, we obtain:

$$\frac{\begin{array}{c} \vdots \\ \{ \sharp X^j g(\sim \alpha), g(\Gamma) \Rightarrow g(\Delta) \}_{j \in \omega} \end{array}}{\sharp Fg(\sim \alpha), g(\Gamma) \Rightarrow g(\Delta)} (\text{Fleft}^s)$$

where $\sharp Fg(\sim \alpha)$ coincides with $g(\sharp \sim G\alpha)$ by the definition of g .

6. Case ($\sim\sharp$ left): The last inference of P is of the form:

$$\frac{\sharp \sim \alpha, \Gamma \Rightarrow \Delta}{\sim \sharp \alpha, \Gamma \Rightarrow \Delta} (\sim\sharp\text{left}).$$

By induction hypothesis, we have the required fact:

$$\text{SLT}_\omega \vdash g(\sharp \sim \alpha), g(\Gamma) \Rightarrow g(\Delta)$$

where $g(\sharp \sim \alpha)$ coincides with $g(\sim \sharp \alpha)$ by the definition of g .

7. Case (\sim ;left): The last inference of P is of the form:

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

By induction hypothesis, we have: $\text{SLT}_\omega \vdash g(\sharp\sim[b][c]\alpha), g(\Gamma) \Rightarrow g(\Delta)$ where $g(\sharp\sim[b][c]\alpha)$ coincides with $\sharp\sim[b][c]g(\alpha)$ by the definition of g . Then, we obtain:

$$\frac{\begin{array}{c} \vdots \\ \sharp\sim[b][c]g(\alpha), g(\Gamma) \Rightarrow g(\Delta) \end{array}}{\sharp\sim[b ; c]g(\alpha), g(\Gamma) \Rightarrow g(\Delta)} (\sim;\text{left})$$

where $\sharp\sim[b ; c]g(\alpha)$ coincides with $g(\sharp\sim[b ; c]\alpha)$ by the definition of g .

• (2) : By induction on the proofs Q of $g(\Gamma) \Rightarrow g(\Delta)$ in cut-free SLT_ω . We distinguish the cases according to the last inference of Q , and show some cases.

1. Case (Gleft^s): The last inference of Q is (Gleft^s).

Subcase (1): The last inference of Q is of the form:

$$\frac{\sharp X^k g(\alpha), g(\Gamma) \Rightarrow g(\Delta)}{\sharp Gg(\alpha), g(\Gamma) \Rightarrow g(\Delta)} (\text{Gleft}^s)$$

where $\sharp X^k g(\alpha)$ and $\sharp Gg(\alpha)$ respectively coincide with $g(\sharp X^k \alpha)$ and $g(\sharp G\alpha)$ by the definition of g . By induction hypothesis, we have: $\text{PSLT}_\omega - (\text{cut}) \vdash \sharp X^k \alpha, \Gamma \Rightarrow \Delta$, and hence obtain the required fact:

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

Subcase (2): The last inference of Q is of the form:

$$\frac{\sharp X^k g(\sim\alpha), g(\Gamma) \Rightarrow g(\Delta)}{\sharp Gg(\sim\alpha), g(\Gamma) \Rightarrow g(\Delta)} (\text{Gleft}^s)$$

where $\sharp X^k g(\sim\alpha)$ and $\sharp Gg(\sim\alpha)$ respectively coincide with $g(\sharp X^k \sim\alpha)$ and $g(\sharp \sim F\alpha)$ by the definition of g . By induction hypothesis, we have: $\text{PSLT}_\omega - (\text{cut}) \vdash \sharp X^k \sim\alpha, \Gamma \Rightarrow \Delta$, and hence obtain the required fact:

$$\frac{\vdots}{\frac{\sharp X^k \sim\alpha, \Gamma \Rightarrow \Delta}{\sharp \sim F\alpha, \Gamma \Rightarrow \Delta} (\sim\text{Fleft})}.$$

2. Case (;left): The last inference of Q is (;left).

Subcase (1): The last inference of Q is of the form:

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

where $\sharp[b][c]g(\alpha)$ and $\sharp[b ; c]g(\alpha)$ respectively coincide with $g(\sharp[b][c]\alpha)$ and $g(\sharp[b ; c]\alpha)$ by the definition of g . By induction hypothesis, we have: $\text{PSLT}_\omega - (\text{cut}) \vdash \sharp[b][c]\alpha, \Gamma \Rightarrow \Delta$, and hence obtain the required fact:

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

Subcase (2): The last inference of Q is of the form:

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

where $\sharp[b][c]g(\sim\alpha)$ and $\sharp[b ; c]g(\sim\alpha)$ coincide with $g(\sharp \sim[b][c]\alpha)$ and $g(\sharp \sim[b ; c]\alpha)$, respectively, by the definition of g . By induction hypothesis, we have: $\text{PSLT}_\omega - (\text{cut}) \vdash \sharp \sim[b][c]\alpha, \Gamma \Rightarrow \Delta$, and hence obtain the required fact:

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

□

Theorem 4.10 (Cut-elimination). *The rule (cut) is admissible in cut-free PSLT_ω .*

Proof. Suppose $\text{PSLT}_\omega \vdash \Gamma \Rightarrow \Delta$. Then, we have $\text{SLT}_\omega \vdash f(\Gamma) \Rightarrow f(\Delta)$ by Theorem 4.9 (1), and hence $\text{SLT}_\omega - (\text{cut}) \vdash f(\Gamma) \Rightarrow f(\Delta)$ by Proposition 4.2 (3). By Theorem 4.9 (2), we obtain $\text{PSLT}_\omega - (\text{cut}) \vdash \Gamma \Rightarrow \Delta$. \square

Theorem 4.11 (Syntactical embedding from PSLT_ω into SLT_ω). *Let Γ and Δ be sets of formulas in \mathcal{L}^{ps} , and g be the mapping defined in Definition 4.3. Then:*

1. $\text{PSLT}_\omega \vdash \Gamma \Rightarrow \Delta$ iff $\text{SLT}_\omega \vdash g(\Gamma) \Rightarrow g(\Delta)$.
2. $\text{PSLT}_\omega - (\text{cut}) \vdash \Gamma \Rightarrow \Delta$ iff $\text{SLT}_\omega - (\text{cut}) \vdash g(\Gamma) \Rightarrow g(\Delta)$.

Proof.

1. (\implies): By Theorem 4.9 (1). (\impliedby): Suppose $\text{SLT}_\omega \vdash g(\Gamma) \Rightarrow g(\Delta)$. We then have $\text{SLT}_\omega - (\text{cut}) \vdash g(\Gamma) \Rightarrow g(\Delta)$ by Proposition 4.2 (3). Thus, we obtain $\text{PSLT}_\omega - (\text{cut}) \vdash \Gamma \Rightarrow \Delta$ by Theorem 4.9 (2). Therefore we have $\text{PSLT}_\omega \vdash \Gamma \Rightarrow \Delta$.
2. (\implies): Suppose $\text{PSLT}_\omega - (\text{cut}) \vdash \Gamma \Rightarrow \Delta$. Then we have $\text{PSLT}_\omega \vdash \Gamma \Rightarrow \Delta$. We then obtain $\text{SLT}_\omega \vdash g(\Gamma) \Rightarrow g(\Delta)$ by Theorem 4.9 (1). Therefore we obtain $\text{SLT}_\omega - (\text{cut}) \vdash g(\Gamma) \Rightarrow g(\Delta)$ by Proposition 4.2 (3). (\impliedby): By Theorem 4.9 (2). \square

Theorem 4.12 (Syntactical embedding from PSLT_ω into LT_ω). *Let Γ and Δ be sets of formulas in \mathcal{L}^{ps} . Let f and g be the mappings defined in Definitions 4.1 and 4.3, respectively. Then:*

1. $\text{PSLT}_\omega \vdash \Gamma \Rightarrow \Delta$ iff $\text{SLT}_\omega \vdash fg(\Gamma) \Rightarrow fg(\Delta)$.
2. $\text{PSLT}_\omega - (\text{cut}) \vdash \Gamma \Rightarrow \Delta$ iff $\text{SLT}_\omega - (\text{cut}) \vdash fg(\Gamma) \Rightarrow fg(\Delta)$.

Proof. By Proposition 4.2 (2) and Theorem 4.11. \square

Theorem 4.13 (Completeness). *For any formula α , $\text{PSLT}_\omega \vdash \Rightarrow \alpha$ iff α is valid in PSLTL .*

Proof. $\text{PSLT}_\omega \vdash \Rightarrow \alpha$ iff $\text{SLT}_\omega \vdash \Rightarrow g(\alpha)$ (by Theorem 4.11) iff $g(\alpha)$ is valid in SLTL (by Proposition 4.2 (4)) iff α is valid in PSLTL (by Theorem 4.6). \square

4.3. Translation examples

We provide an algorithm for translating a PSLTL-formula into an LTL-formula.

Algorithm 4.14. *Let α be a PSLTL-formula. Then, we obtain an LTL-formula $fg(\alpha)$ by the following steps.*

1. *We translate the PSLTL-formula α into an SLTL-formula $g(\alpha)$ by using the translation function g defined in Definition 4.3.*
 - (a) *For α , we apply the condition of g which corresponds to the outer-most PSLTL-connective of α .*
 - (b) *The resulting formula expression β is of the form $g(\alpha_1) \circ g(\alpha_2)$, $\sharp g(\alpha_1)$, $g(\alpha_1) \circ g(\alpha_2)$ or $g(\alpha_1)$ where \circ and \sharp represent the outer-most PSLTL-connectives of α .*
 - (c) *If there is a PSLTL-connective appearing in α_1 and/or α_2 in β , then we apply the same procedure displayed above to α_1 and α_2 , where α_1 and/or α_2 are regarded as α above. If there is no PSLTL-connective appearing in α_1 and/or α_2 in β , then we go to the next step.*
 - (d) *We translate all the formulas of the form $\sim p$ appearing in the resulting formula expression into the SLTL-formulas of the form p' .*
 - (e) *The resulting formula expression is just the required SLTL-formula $g(\alpha)$.*
2. *In a similar way as shown above, we translate the SLTL-formula $g(\alpha)$ into an LTL-formula $fg(\alpha)$ by using the translation function f defined in Definition 4.1. Then, we obtain the required LTL-formula $fg(\alpha)$.*

We show some translation examples below.

Example 4.15. We consider a formula $G(\sim([b]p \wedge \sim[c]q))$ where b, c are atomic sequences, and p, q are propositional variables.

Firstly, we translate this PSLTL-formula into a SLTL-formula by the translation function g as follows:

$$\begin{aligned}
& g(\mathbf{G}(\sim([b]p \wedge \sim[c]q))) \\
&= \mathbf{G}g(\sim([b]p \wedge \sim[c]q)) \\
&= \mathbf{G}(g(\sim[b]p) \vee g(\sim\sim[c]q)) \\
&= \mathbf{G}([b]g(\sim p) \vee g([c]q)) \\
&= \mathbf{G}([b]p' \vee [c]g(q)) \\
&= \mathbf{G}([b]p' \vee [c]q)
\end{aligned}$$

where p' is a propositional variable in SLTL.

Next, we translate this SLTL-formula into a LTL-formula by the translation function f as follows:

$$\begin{aligned}
& f(\mathbf{G}([b]p' \vee [c]q)) \\
&= \mathbf{G}f([b]p' \vee [c]q) \\
&= \mathbf{G}(f([b]p') \vee f([c]q)) \\
&= \mathbf{G}(p'^b \vee q^c)
\end{aligned}$$

where p'^b, q^c are propositional variables in LTL.

Thus, the formula $\mathbf{G}(\sim([b]p \wedge \sim[c]q))$ of PSLTL is translated into the formula $\mathbf{G}(p'^b \vee q^c)$ of LTL.

Example 4.16. We consider a formula $\mathbf{G}(\sim[b]p \rightarrow \sim\mathbf{G}[c]\sim q)$ where b, c are atomic sequences, and p, q are propositional variables.

Firstly, we translate this PSLTL-formula into a SLTL-formula by the translation function g as follows.

$$\begin{aligned}
& g(\mathbf{G}(\sim[b]p \rightarrow \sim\mathbf{G}[c]\sim q)) \\
&= \mathbf{G}g(\sim[b]p \rightarrow \sim\mathbf{G}[c]\sim q) \\
&= \mathbf{G}(g(\sim[b]p) \rightarrow g(\sim\mathbf{G}[c]\sim q)) \\
&= \mathbf{G}([b]g(\sim p) \rightarrow \mathbf{F}g(\sim[c]\sim q)) \\
&= \mathbf{G}([b]p' \rightarrow \mathbf{F}[c]g(\sim\sim q)) \\
&= \mathbf{G}([b]p' \rightarrow \mathbf{F}[c]g(q)) \\
&= \mathbf{G}([b]p' \rightarrow \mathbf{F}[c]q)
\end{aligned}$$

where p' is a propositional variable in SLTL.

Next, we translate this SLTL-formula into a LTL-formula by the translation function f as follows.

$$\begin{aligned}
& f(G([b]p' \rightarrow F[c]q)) \\
&= Gf([b]p' \rightarrow F[c]q) \\
&= G(f([b]p') \rightarrow f(F[c]q)) \\
&= G(p'^b \rightarrow Ff([c]q)) \\
&= G(p'^b \rightarrow Fq^c)
\end{aligned}$$

where p^b, q^c are propositional variables in LTL.

Thus, the formula $G(\sim[b]p \rightarrow \sim G[c]\sim q)$ of PSLTL is translated into the formula $G(p'^b \rightarrow Fq^c)$ of LTL.

5. Illustrative examples

In this section, we provide an illustrative example using PSLTL to model the learning processes of students. The example is based on examples presented in [15, 16]. A model of students should be inconsistency-tolerant since, in general, student understanding is uncertain and vague. PSLTL can be used to express the negation of uncertain concepts such as “*understand*” (or “*understanding*”). For instance, if we cannot determine whether someone understands, then the uncertain concept “*understand*” can be represented by asserting the following inconsistent formula:

$$\textit{understand} \wedge \sim \textit{understand}$$

This is well-formalized because $(\textit{understand} \wedge \sim \textit{understand}) \rightarrow \perp$ is not valid in paraconsistent logic. On the other hand, we can decide whether someone is learning; this decision is represented by $\neg \textit{learning}$, where $(\textit{learning} \wedge \neg \textit{learning}) \rightarrow \perp$ is valid in classical logic.

The following negative expressions can be interpreted differently:

$\neg \textit{understand}$ (someone does not understand).

$\sim \textit{understand}$ (someone does not fully understand).

The first statement indicates that a person does not understand at all. The second statement means that we can say that a person does not deeply or fully understand, but he or she has a shallow understanding. We thus allow $\textit{understand} \wedge \sim \textit{understand}$.

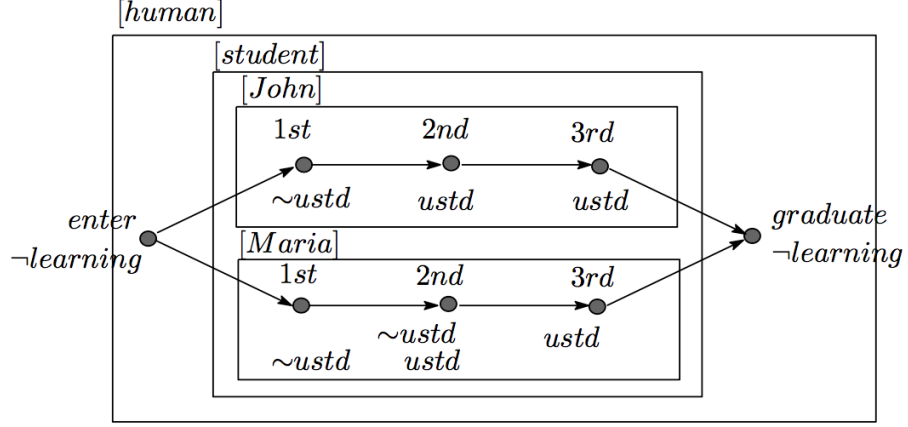


Figure 1: Learning processes of students

In ontology representation, a concept hierarchy is constructed by *ISA-relations* (i.e., “*is a*”-relations or *sub-concept relations*) between concepts, i.e., a concept is a sub-concept of another concept. Below, we use sequence modal operators to represent ISA-relations between concepts. Let c_1, c_2, \dots, c_n be concept symbols. Then, we write a sequence of concept names by $[c_1; c_2; \dots; c_n]$. Each order (c_i, c_j) ($1 \leq i < j \leq n$) of concepts in the sequence modal operator $[c_1; c_2; \dots; c_n]$ can be used to represent an ISA-relation between c_i and c_j . For example, we declare the following order of two concepts as an ISA-relation between “human” and “student”:

$$[student; human]$$

This sequence expresses that the concept “student” is a sub-concept of the concept “human.”

The sequence modal operators in PSLTL are applied to hierarchical structures where each hierarchical structure is a specific model of concepts in a hierarchy. Figure 1 shows a hierarchical structure of the learning processes of students in a high school. A typical high school student graduates in three years. In the figure, $\sim ustd$ (an abbreviation of $\sim understand$) represents uncertain negative information that can exist simultaneously with $ustd$ (an abbreviation of *understand*), which represents positive information.

We show a paraconsistent sequential model

$$M = \langle \sigma_i, \{I^{+\hat{d}}\}_{\hat{d} \in \text{SE}}, \{I^{-\hat{d}}\}_{\hat{d} \in \text{SE}} \rangle$$

that corresponds to a model of the learning processes of students, as shown in Figure 1. For any $* \in \{+, -\}$, we have the following:

1. $S = \{s_0, s_1, s_2, s_3, s_4, s_5, s_6, s_7\}$,
2. $\sigma_1 = s_0 s_1 s_2 s_3 s_7 s_7 s_7 \cdots$,
3. $\sigma_2 = s_0 s_4 s_5 s_6 s_7 s_7 s_7 \cdots$,
4. $I^{*human}(enter) = \{s_0\}$, $I^{*human}(graduate) = \{s_7\}$,
5. $I^{*human}(1st) = I^{*student}(1st) = \{s_1, s_4\}$,
6. $I^{*human}(2nd) = I^{*student}(2nd) = \{s_2, s_5\}$,
7. $I^{*human}(3rd) = I^{*student}(3rd) = \{s_3, s_6\}$,
8. $I^{*human}(learning) = I^{*student}(learning) = \{s_1, s_2, s_3, s_4, s_5, s_6\}$,
9. $I^{+human}(ustd) = \{s_2, s_3, s_5, s_6\}$, $I^{-human}(ustd) = \{s_1, s_4, s_5\}$,
10. $I^{*student}(enter) = I^{*student}(graduate) = \emptyset$,
11. $I^{*John}(enter) = I^{*John}(graduate) = \emptyset$,
12. $I^{*John}(1st) = \{s_1\}$, $I^{*John}(2nd) = \{s_2\}$, $I^{*John}(3rd) = \{s_3\}$,
13. $I^{+John}(ustd) = \{s_2, s_3\}$, $I^{-John}(ustd) = \{s_1\}$,
14. $I^{*Maria}(enter) = I^{*Maria}(graduate) = \emptyset$,
15. $I^{*Maria}(1st) = \{s_4\}$, $I^{*Maria}(2nd) = \{s_5\}$, $I^{*Maria}(3rd) = \{s_6\}$,
16. $I^{+Maria}(ustd) = \{s_5, s_6\}$, $I^{-Maria}(ustd) = \{s_4, s_5\}$.

We then verify “Is there a student who has difficulty in understanding the lectures in the first year?” This statement is expressed as follows:

$$[student; human]F(learning \wedge \sim understand \wedge 1st)$$

The above statement is true because we have path $s_0 \rightarrow s_1$ with

$$s_1 \in I^{-John}(understand), s_1 \in I^{*John}(learning), \text{ and } s_1 \in I^{*John}(1st)$$

with $* \in \{+, -\}$. More specifically, the first year lectures are difficult for John.

We can also verify “Is there a student who is confused while understanding lectures?” This statement is expressed as follows:

$$[student; human]F(learning \wedge \sim understand \wedge understand)$$

The above statement is true because we have path $s_0 \rightarrow s_4 \rightarrow s_5$ with $s_5 \in I^{*Maria}(learning)$, $s_5 \in I^{+Maria}(understand)$, and $s_5 \in I^{-Maria}(understand)$. More specifically, understanding some second year lectures is confusing for Maria.

We can also obtain the corresponding LTL-formulas of the above mentioned PSLTL-formulas, using the translation algorithm presented in the previous section. Thus, we can also verify such resulting LTL-formulas by using LTL.

We show such a translation example below. The PSLTL-formula

$$[s ; h]F(l \wedge \sim u \wedge u),$$

which is an abbreviation of the above mentioned formula, can be transformed into the corresponding LTL-formula

$$F(l^s ; h \wedge u'^s ; h \wedge u^s ; h)$$

as follows.

First, we translate this PSLTL-formula into a SLTL-formula by the translation function g as follows.

$$\begin{aligned} & g([s ; h]F(l \wedge \sim u \wedge u)) \\ &= [s ; h]Fg(l \wedge \sim u \wedge u) \\ &= [s ; h]F(g(l) \wedge g(\sim u) \wedge g(u)) \\ &= [s ; h]F(l \wedge u' \wedge u) \end{aligned}$$

where u' is a propositional variable in SLTL.

Next, we translate this SLTL-formula into a LTL-formula by the translation function f as follows.

$$\begin{aligned}
 & f([s ; h]F(l \wedge u' \wedge u)) \\
 &= Ff([s ; h](l \wedge u' \wedge u)) \\
 &= F(f([s ; h]l) \wedge f([s ; h]u') \wedge f([s ; h]u)) \\
 &= F(l^s ; h \wedge u'^s ; h \wedge u^s ; h)
 \end{aligned}$$

where $l^s ; h$, $u'^s ; h$ and $u^s ; h$ are propositional variables in LTL.

Next, we consider some proof examples using PSLT_ω . We can give a proof of the sequent

$$\Rightarrow [Maria; student; human](\sim understand \wedge understand \wedge 2nd)$$

from the following assumptions:

1. $\Rightarrow \sim[Maria][student][human]understand$
2. $\Rightarrow [Maria][student][human]understand$
3. $\Rightarrow [Maria][student][human]2nd$

by:

$$\begin{array}{l}
 \Rightarrow \sim[M][s][h]ustd \quad \Rightarrow [M][s][h]ustd \quad \Rightarrow [M][s][h]2nd \\
 \Rightarrow [M][s][h]\simustd \quad \quad \quad \vdots \quad \quad \quad \vdots \\
 \quad \quad \quad \vdots \quad \quad \quad \Rightarrow [M; s; h]ustd \quad \Rightarrow [M; s; h]2nd \\
 \Rightarrow [M; s; h]\simustd \quad \quad \quad \Rightarrow [M; s; h](ustd \wedge 2nd) \\
 \hline
 \Rightarrow [M; s; h](\simustd \wedge ustd \wedge 2nd)
 \end{array}$$

where M , s , h and $ustd$ are abbreviations of *Maria*, *student*, *human* and *understand*, respectively.

We can also give a proof of the sequent

$$[Maria; student; human]enter \Rightarrow F[Maria; student; human]graduate$$

from the following assumptions:

1. $[Maria; student; human]enter$
 $\Rightarrow [Maria; student; human]\simunderstand$
2. $[Maria; student; human]enter \Rightarrow [Maria; student; human]learning$
3. $[Maria; student; human]enter \Rightarrow [Maria; student; human]1st$

4. $[Maria; student; human](understand \wedge learning \wedge 1st)$
 $\Rightarrow [Maria; student; human](understand \wedge learning \wedge 2nd)$
5. $[Maria; student; human](understand \wedge learning \wedge 2nd)$
 $\Rightarrow [Maria; student; human](understand \wedge learning \wedge 3rd)$
6. $[Maria; student; human](understand \wedge learning \wedge 3rd)$
 $\Rightarrow [Maria; student; human]graduate$

by:

$$\frac{\begin{array}{c} \vdots P \\ [m]e \Rightarrow [m](\sim u \wedge l \wedge 2) \quad [m](\sim u \wedge l \wedge 2) \Rightarrow [m](u \wedge l \wedge 3) \end{array}}{\frac{[m]e \Rightarrow [m](u \wedge l \wedge 3)}{[m]e \Rightarrow [m]g} \quad [m](u \wedge l \wedge 3) \Rightarrow [m]g}}{\frac{[m]e \Rightarrow [m]g}{[m]e \Rightarrow F[m]g}}$$

where P is:

$$\frac{\frac{[m]e \Rightarrow [m]\sim u \quad [m]e \Rightarrow [m]l}{[m]e \Rightarrow [m](\sim u \wedge l)} \quad [m]e \Rightarrow [m]1}{\frac{[m]e \Rightarrow [m](\sim u \wedge l \wedge 1)}{[m]e \Rightarrow [m](\sim u \wedge l \wedge 2)} \quad [m](\sim u \wedge l \wedge 1) \Rightarrow [m](\sim u \wedge l \wedge 2)}}{[m]e \Rightarrow [m](\sim u \wedge l \wedge 2)}$$

where $[m]$, e , u , l , 1 , 2 , 3 , g are abbreviations of $[Maria; student; human]$, $enter$, $understand$, $learning$, $1st$, $2nd$ and $3rd$, respectively.

6. Conclusions and related works

In this paper, we introduced the semantics of PSLTL by extending the semantics of LTL. PSLTL can appropriately represent inconsistency-tolerant reasoning via the paraconsistent negation connective and sequential information provided by sequence modal operators. By using the semantical embedding theorem of PSLTL into LTL, we showed that PSLTL is decidable. The Gentzen-type sequent calculus $PSLT_{\omega}$ for PSLTL was also introduced,

and the cut-elimination theorem for this calculus was proved using the syntactic embedding theorem of PSLT_ω into its non-paraconsistent fragment SLT_ω . The completeness theorem for PSLTL (and PSLT_ω) was proved using both syntactical and semantical embedding theorems of PSLTL (and PSLT_ω) into SLTL (and SLT_ω). It was thus shown in this paper that PSLTL and PSLT_ω are a good theoretical basis for inconsistency-tolerant temporal reasoning with sequential information.

Closely related works are discussed below. In [15], an extended LTL called *sequential paraconsistent LTL* (SPLTL), was introduced as a Kripke semantics to formalize inconsistency-tolerant temporal reasoning with hierarchical information. A theorem for embedding SPLTL into LTL was proved, and SPLTL was shown to be decidable. Moreover, some illustrative examples for verifying the learning processes of students were presented using the SPLTL semantics. The logic SPLTL in [15] is essentially the same as PSLTL , but only semantics for SPLTL were introduced, i.e., a Gentzen-type sequent calculus was not introduced for SPLTL in [15]. The present paper provides a uniform embedding perspective with both semantics and a proof system. The completeness theorem for PSLTL , which was obtained using both semantical and syntactical embedding theorems, is one of the main contributions of our paper.

In [16, 18], a formal method was proposed for modeling and verifying inconsistency-tolerant temporal reasoning with hierarchical information. To achieve this, temporal logic called *sequential paraconsistent computation tree logic* (SPCTL) was obtained from computation tree logic (CTL) by adding a paraconsistent negation connective and sequence modal operators. The validity, satisfiability, and model-checking problems of SPCTL were shown to be decidable. Illustrative examples for inconsistency-tolerant temporal reasoning with hierarchical information were also presented using SPCTL . For example, in [18], some illustrative examples for medical reasoning are presented based on SPCTL . SPCTL in [16, 18] was formulated as a Kripke semantics, and a proof system for SPCTL was not introduced. Thus, the completeness theorem for SPCTL was not given. Thus, compared with SPCTL , our proposed PSLTL is regarded as a good theoretical basis in the sense that it has both the semantics and proof system in conjunction with the completeness theorem.

The rest of this paper addresses other related works on paraconsistent temporal logics. While the idea of combining paraconsistency and sequen-

tiality within a single temporal logic is new, the idea of introducing a paraconsistent temporal logic is not. In this study, PLTL [21] was used as a base logic for constructing PSLTL. Although there are no other paraconsistent variants of LTL, there are some other paraconsistent variants of CTL.

A *multi-valued computation tree logic*, χ CTL, which has the algebraic structure called *quasi-Boolean logic*, was introduced by Easterbrook and Chechik [7]. The Kripke structure for this logic was based on a multi-valued transition relation and a multi-valued valuation (labeling) function. The multi-valued valuation function was a very general setting because it can express n -valued truth values for any natural number n .

A *quasi-classical temporal logic*, QCTL, was proposed by Chen and Wu [4] in order to formalize reasoning on inconsistent concurrent systems. In this work, *paraKripke structures* were introduced for QCTL. In QCTL, a set of positive and negative objects, which is constructed from a set of atomic formulas, is used; in other words, a positive object $+p$ and a negative object $-p$ are obtained from an atomic formula p .

PCTL was introduced in [20, 24] by Kamide and Kaneiwa as an alternative to these paraconsistent computation-tree logics. An extension PCTL* of PCTL was also studied from the point of view of bisimulations for paraconsistent Kripke structures in paraconsistent model checking [13]. As explained before, another extension of PCTL was also studied in [16] for verifying student learning processes in learning support systems. These extended paraconsistent computation-tree logics are formulated as Kripke semantics, but Gentzen-type proof systems for these logics have not been proposed yet.

Acknowledgments

This work was supported by JSPS Grant-in-Aid for Scientific Research (C) JP26330263.

References

- [1] A. Almukdad and D. Nelson, Constructible falsity and inexact predicates, *Journal of Symbolic Logic* **49** (1984), 231–233.
- [2] O. Arieli and A. Zamansky, A framework for reasoning under uncertainty based on non-deterministic distance semantics, *International Journal of Approximate Reasoning* **52:2** (2011), 184–211.

- [3] S. Baratella and A. Masini, An approach to infinitary temporal proof theory, *Archive for Mathematical Logic* **43** (2004), 965–990.
- [4] D. Chen and J. Wu, Reasoning about inconsistent concurrent systems: A non-classical temporal logic, Proceedings of the 32nd Conference on Current Trends in Theory and Practice of Computer Science (SOFSEM 2006), Lecture Notes in Computer Science 3831, pp. 207–217, 2006.
- [5] E.M. Clarke and E.A. Emerson, Design and synthesis of synchronization skeletons using branching time temporal logic, *Lecture Notes in Computer Science* **131** (1981), 52–71.
- [6] E.M. Clarke, O. Grumberg, and D.A. Peled, Model checking, The MIT Press, 1999.
- [7] S. Easterbrook, and M. Chechik, A framework for multi-valued reasoning over inconsistent viewpoints, Proceedings of the 23rd International Conference on Software Engineering (ICSE 2001), pp. 411–420, 2001.
- [8] 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.
- [9] D. Gabbay, A. Pnueli, S. Shelah and J. Stavi, On the temporal analysis of fairness, Proceedings of the 7th ACM Symposium on Principles of Programming Languages, ACM Press, pp. 163–173, 1980.
- [10] Y. Gurevich, Intuitionistic logic with strong negation, *Studia Logica* **36** (1977), 49–59.
- [11] G.J. Holzmann, The SPIN model checker: Primer and reference manual, Addison-Wesley, 2006.
- [12] N. Kamide, An equivalence between sequent calculi for linear-time temporal logic, *Bulletin of the Section of the Logic* **35:4** (2006), 187–194.
- [13] N. Kamide, Extended full computation tree logics for paraconsistent model checking, *Logic and Logical Philosophy* **15:3** (2006), 251–276.
- [14] N. Kamide, A proof system for temporal reasoning with sequential information, Proceedings of the 20th Brazilian Symposium on Artificial Intelligence (SBIA 2010), Lecture Notes in Artificial Intelligence 6404, pp. 283–292, 2010.
- [15] N. Kamide, An extended LTL for inconsistency-tolerant reasoning with hierarchical information: Verifying students’ learning processes, *International Journal of e-Education, e-Business, e-Management and e-Learning* **3** (3), pp. 234–238, IACSIT Press, 2013.
- [16] N. Kamide, Modeling and verifying inconsistency-tolerant temporal reasoning with hierarchical information: Dealing with students’ learning processes, Proceedings of the 2013 IEEE International Conference on Systems, Man, and Cybernetics (IEEE SMC 2013), pp. 1859–1864, 2013.
- [17] N. Kamide, Inconsistency and sequentiality in LTL, Proceedings of the 7th International Conference on Agents and Artificial Intelligence (ICAART 2015), pp. 46–54, 2015.

- [18] N. Kamide, Inconsistency-tolerant temporal reasoning with hierarchical information, *Information Sciences* **320** (2015), 140–155.
- [19] 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.
- [20] N. Kamide and K. Kaneiwa, Paraconsistent negation and classical negation in computation tree logic, Proceedings of the 2nd International Conference on Agents and Artificial Intelligence (ICAART 2010), Vol. 1, pp. 464–469, 2010.
- [21] N. Kamide and H. Wansing, A paraconsistent linear-time temporal logic, *Fundamenta Informaticae* **106**:1 (2011), 1–23.
- [22] N. Kamide and H. Wansing, Proof theory of Nelson’s paraconsistent logic: A uniform perspective, *Theoretical Computer Science* **415** (2012), 1–38.
- [23] K. Kaneiwa and N. Kamide, Sequence-indexed linear-time temporal logic: Proof system and application, *Applied Artificial Intelligence* **24** (2010), 896–913.
- [24] K. Kaneiwa and N. Kamide, Paraconsistent computation tree logic, *New Generation Computing* **29**:4 (2011), 391–408.
- [25] K. Kaneiwa and N. Kamide, Conceptual modeling in full computation-tree logic with sequence modal operator, *International Journal of Intelligent Systems* **26**:7 (2011), 636–651.
- [26] H. Kawai, Sequential calculus for a first order infinitary temporal logic, *Zeitschrift für Mathematische Logik und Grundlagen der Mathematik* **33** (1987), 423–432.
- [27] D. Nelson, Constructible falsity, *Journal of Symbolic Logic* **14** (1949), 16–26.
- [28] A. Pnueli, The temporal logic of programs, Proceedings of the 18th IEEE Symposium on Foundations of Computer Science, pp. 46–57, 1977.
- [29] G. Priest, Paraconsistent logic, Handbook of Philosophical Logic (Second Edition), Vol. 6, D. Gabbay and F. Guenther (eds.), Kluwer Academic Publishers, Dordrecht, pp. 287–393, 2002.
- [30] W. Rautenberg, Klassische und nicht-klassische Aussagenlogik, Vieweg, Braunschweig, 1979.
- [31] N. Vorob’ev, A constructive propositional logic with strong negation, *Doklady Akademii Nauk SSSR* **85** (1952), 465–468, (in Russian).
- [32] H. Wansing, The logic of information structures, Lecture Notes in Artificial Intelligence 681, 163 pages, 1993.

Teikyo University, Faculty of Science and Engineering
Department of Information and Electronic Engineering
Toyosatodai 1-1, Utsunomiya
Tochigi 320-8551, Japan
drnkamide08@kpd.biglobe.ne.jp