An Extended LTL for Inconsistency-Tolerant Reasoning with Hierarchical Information: Verifying Students' Learning Processes

Norihiro Kamide

***Abstract*—In this paper, a new extended linear-time temporal logic (LTL), called sequential paraconsistent LTL (SPLTL), is introduced for formalizing inconsistency-tolerant reasoning with hierarchical information. A theorem for embedding SPLTL into LTL is proved, and SPLTL is shown to be decidable. Some illustrative examples for verifying Students' learning processes are presented using SPLTL.**

***Index Terms*—Decidability, linear-time temporal logic, paraconsistent logic, students' learning processes.**

I. Introduction

In this paper, a new extended *linear-time temporal logic* (LTL) [1], called *sequential paraconsistent LTL* (SPLTL), is introduced as a semantics with a paraconsistent negation connective [2]-[4] and some sequence modal operators [5], [6].

The logic SPLTL can appropriately represent both, inconsistency-tolerant reasoning by the paraconsistent negation connective, and hierarchical information by the sequence modal operators. Some illustrative examples for verifying Students' learning processes are presented using SPLTL. Some theorems for embedding SPLTL into a paraconsistent version PLTL of LTL and into LTL are proved. By using these embedding theorems, SPLTL is shown to be decidable.

A motivation of this paper is to formalize students' learning processes in SPLTL. Formalizing students' learning process in an appropriate logic is useful for implementing verification algorithms in some learning support systems such as intelligent tutoring and e-learning systems. A model of students in such a system should be inconsistency-tolerant since student's understanding is uncertain and vague in general. Moreover, detailed information on students should be well-structured with hierarchical information. In order to represent such a student model, we need a paraconsistent negation connective, which can suitably represent inconsistency-tolerant reasoning, and some sequence modal operators, which can suitably represent hierarchical information.

From the point of view of logic, SPLTL is a combination of LTL and *Nelson's paraconsistent four-valued logic with strong negation*, N4. LTL is known to be one of the most useful temporal logics for verifying and specifying concurrent systems [1], [7]. On the other hand, N4 is known to be one of the most important base logics for inconsistency-tolerant reasoning [2], [4], [8], [9]. Combining the logics LTL and N4 was studied in [8], and such a combined logic was called *paraconsistent LTL* (PLTL). Roughly speaking, SPLTL is obtained from PLTL by adding some sequence modal operators.

Combining LTL with some sequence modal operators was studied in [5], and such a combined logic was called *sequence-indexed LTL* (SLTL). SPLTL is regarded as a modified paraconsistent extension of SLTL, and hence SPLTL is a modified extension of both PLTL [3] and SLTL [5]. In the following, we explain an important property of paraconsistent negation and a plausible interpretation of sequence modal operators.

A paraconsistent negation connective $\sim$ is used in SPLTL. One reason why $\sim$ is considered is that it may be added in such a way that the extended logics satisfy the property of *paraconsistency*. A semantic consequence relation $\models$ is called paraconsistent with respect to a negation connective $\sim$ if there are formulas α, β such that $\{\alpha, \sim\alpha\} \not\models \beta$. In the case of LTL, this means that there is a model M and a position i of a sequence $\sigma = t_0, t_1, t_2, \ldots$ of time-points in M with $(M, i) \not\models (\alpha \land \sim\alpha) \to \beta$. It is known that logical systems with paraconsistency can deal with inconsistency-tolerant and uncertainty reasoning more appropriately than systems which are non-paraconsistent. For example, we do not desire that $(s(x) \land \sim s(x)) \to d(x)$ is 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). For more information on paraconsistency, see e.g., [10].

Some sequence modal operators [5], [6] are used in SPLTL. A sequence modal operator [b] represents a sequence b of symbols. The notion of sequences is useful to represent the notions of "information," "trees," "orders," and "ontologies." Thus, "hierarchical information" can be represented by sequences. This is plausible because a sequence structure gives a *monoid* (M, ;, $\emptyset$) with *informational interpretation* [9]:

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

Manuscript received March 11, 2013; revised June 20, 2013.

Norihiro Kamide is with Faculty of Information Technology and Business, Cyber University, Japan Cyber Educational Institute, Ltd. Izumishibakouen Building 4F, 1-6-8 Shibakouen, Minato-ku, Tokyo 105-0011, Japan (e-mail: drnkamide08@kpd.biglobe.ne.jp).

DOI: 10.7763/IJEEEE.2013.V3.230

A formula of the form $[b_1 ; b_2 ; \ldots ; b_n]\,\alpha$ in SPLTL intuitively means that "α is true based on a sequence $b_1 ; b_2 ; \ldots ; b_n$ of (ordered or prioritized) information pieces."Further, a formula of the form $[\emptyset]\alpha$ in SPLTL, 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 this paper is then addressed as follows. In Section II, SPLTL is introduced as a semantics by extending LTL with a paraconsistent negation connective and some sequence modal operators. In Section III, a verification example for students' learning processes is presented using SPLTL. In Section IV, the decidability of SPLTL is shown using a theorem for embedding SPLTL into PLTL. Firstly in this section, LTL and PLTL are introduced, and then the embedding theorems for SPLTL into PLTL and into LTL are proved. In Section V, this paper is concluded.

II. Sequential Paraconsistent LTL

Formulas of SPLTL are constructed from countably many propositional variables, → (implication), ∧ (conjunction), ∨ (disjunction), ¬ (classical negation), X (next), G (globally), F (eventually), ~ (paraconsistent negation) and [b] (sequence modal operator) where b is a sequence. *Sequences* are constructed from countably many atomic sequences, ∅ (empty sequence) and ; (composition). Lower-case letters b, c, ... are used for sequences, lower-case letters p, q, ... are used to denote propositional variables, and Greek lower-case letters α, β, ... are used to denote formulas. An expression $[\emptyset]$ means α, and expressions $[\emptyset ; b]\alpha$ and $[b ; \emptyset]\alpha$ mean $[b]\alpha$. 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 ≤ or ≥ is used to represent the linear order on ω. We write $A \equiv B$ to indicate the syntactical identity between A and B.

Definition 2.1: Formulas and sequences are defined by the following grammar, assuming p and e represent propositional variables and atomic sequences, respectively:

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

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

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

The formulation of SPLTL uses two kinds of satisfaction relations $\models^{+\underline{d}}$ and $\models^{-\underline{d}}$. The intuitive interpretations of $\models^{+\underline{d}}$ and $\models^{-\underline{d}}$ are "verification with sequential information"(or "support of truth with sequential information") and "refutation with sequential information"(or "falsification with sequential information", "support of falsity with sequential information"), respectively.

Definition 2.2 (SPLTL): Let S be a non-empty set of states. A structure $M := (\sigma, I^{+\underline{d}}, I^{-\underline{d}})$ with $\underline{d} \in SE$ is a *sequential paraconsistent model* iff

1. σ is an infinite sequence $s_0, s_1, s_2, \ldots$ of states in S,
2. $I^{+\underline{d}}$ and $I^{-\underline{d}}$ are mappings from the set Φ of propositional variables to the power set of S.

Satisfaction relations $(M, i) \models^{*\underline{d}} \alpha$ ($* \in \{+, -\}$, $\underline{d} \in SE$) for any formula α, where M is a sequential paraconsistent model $(\sigma, I^{+\underline{d}}, I^{-\underline{d}})$ with $\underline{d} \in SE$ and i (∈ ω) represents some position within σ, are defined by

1. for any $p \in \Phi$, $(M, i) \models^{+\underline{d}} p$ iff $s_i \in I^{+\underline{d}}(p)$,
2. $(M, i) \models^{+\underline{d}} \alpha \wedge \beta$ iff $(M, i) \models^{+\underline{d}} \alpha$ and $(M, i) \models^{+\underline{d}} \beta$,
3. $(M, i) \models^{+\underline{d}} \alpha \vee \beta$ iff $(M, i) \models^{+\underline{d}} \alpha$ or $(M, i) \models^{+\underline{d}} \beta$,
4. $(M, i) \models^{+\underline{d}} \alpha \rightarrow \beta$ iff $(M, i) \models^{+\underline{d}} \alpha$ implies $(M, i) \models^{+\underline{d}} \beta$,
5. $(M, i) \models^{+\underline{d}} \neg\alpha$ iff $(M, i) \not\models^{+\underline{d}} \alpha$,
6. $(M, i) \models^{+\underline{d}} {\sim}\alpha$ iff $(M, i) \models^{-\underline{d}} \alpha$,
7. $(M, i) \models^{+\underline{d}} X\alpha$ iff $(M, i+1) \models^{+\underline{d}} \alpha$,
8. $(M, i) \models^{+\underline{d}} G\alpha$ iff $\forall j \geq i\ [(M, j) \models^{+\underline{d}} \alpha]$,
9. $(M, i) \models^{+\underline{d}} F\alpha$ iff $\exists j \geq i\ [(M, j) \models^{+\underline{d}} \alpha]$,
10. for any $p \in \Phi$, $(M, i) \models^{-\underline{d}} p$ iff $s_i \in I^{-\underline{d}}(p)$,
11. $(M, i) \models^{-\underline{d}} \alpha \wedge \beta$ iff $(M, i) \models^{-\underline{d}} \alpha$ or $(M, i) \models^{-\underline{d}} \beta$,
12. $(M, i) \models^{-\underline{d}} \alpha \vee \beta$ iff $(M, i) \models^{-\underline{d}} \alpha$ and $(M, i) \models^{-\underline{d}} \beta$,
13. $(M, i) \models^{-\underline{d}} \alpha \rightarrow \beta$ iff $(M, i) \models^{+\underline{d}} \alpha$ and $(M, i) \models^{-\underline{d}} \beta$,
14. $(M, i) \models^{-\underline{d}} \neg\alpha$ iff $(M, i) \not\models^{-\underline{d}} \alpha$,
15. $(M, i) \models^{-\underline{d}} {\sim}\alpha$ iff $(M, i) \models^{+\underline{d}} \alpha$,
16. $(M, i) \models^{-\underline{d}} X\alpha$ iff $(M, i+1) \models^{-\underline{d}} \alpha$,
17. $(M, i) \models^{-\underline{d}} G\alpha$ iff $\exists j \geq i\ [(M, j) \models^{-\underline{d}} \alpha]$,
18. $(M, i) \models^{-\underline{d}} F\alpha$ iff $\forall j \geq i\ [(M, j) \models^{-\underline{d}} \alpha]$,
19. for any atomic sequence e and any $* \in \{+, -\}$,
    $(M, i) \models^{*\underline{d}} [e]\alpha$ iff $(M, i) \models^{*\underline{d};e} \alpha$,
20. $(M, i) \models^{*\underline{d}} [b ; c]\alpha$ iff $(M, i) \models^{*\underline{d}} [b][c]\alpha$.

A formula α is *valid* (*satisfiable*) in SPLTL iff $(M, 0) \models^{+\emptyset} \alpha$ for any (some, resp.) sequential paraconsistent model $M := (\sigma, I^{+\underline{d}}, I^{-\underline{d}})$ with $\underline{d} \in SE$.

Proposition 2.3: The following clauses hold for any formula α, any sequences c, $\underline{d}$, and any $* \in \{+, -\}$,

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

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

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 $\models^{*\underline{d}}$.

Case $c \equiv a ; b$: $(M, i) \models^{*\underline{d}} [a ; b]\alpha$ iff $(M, i) \models^{*\underline{d}} [a][b]\alpha$ iff $(M, i) \models^{*\underline{d} ; a} [b]\alpha$ (by induction hypothesis) iff
$(M, i) \models^{*\underline{d} ; a ; b} \alpha$ (by induction hypothesis).

An expression $\alpha \leftrightarrow \beta$ means $(\alpha \rightarrow \beta) \wedge (\beta \rightarrow \alpha)$. The following formulas are valid in SPLTL: 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. $[b](\alpha \# \beta) \leftrightarrow ([b]\alpha) \# ([b]\beta)$ where $\# \in \{\rightarrow, \wedge, \vee\}$,
10. $[b]\#\alpha \leftrightarrow \#[b]\alpha$ where $\# \in \{\neg, {\sim}, X, G, F\}$,
11. $[b ; c]\alpha \leftrightarrow [b][c]\alpha$.

The falsification conditions for ¬ 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 ¬*p* is indeed falsified at a state i in model M: $(M, i) \models^{-\underline{d}} \neg p$. 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.

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

1. α is verified at i, i.e., $(M, i) \models^{+\underline{d}} \alpha$,
2. α is falsified at i, i.e., $(M, i) \models^{-\underline{d}} \alpha$,
3. α is both verified and falsified at i,
4. α is neither verified nor falsified at i.

SPLTL is paraconsistent with respect to $\sim$. The reason is presented as follows. Assume a sequential paraconsistent model $M := (\sigma, I^{+\underline{d}}, I^{-\underline{d}})$ with $\underline{d} \in SE$ such that $s_i \in I^{+\underline{d}}(p)$, $s_i \in I^{-\underline{d}}(p)$ and not-$[s_i \in I^{+\underline{d}}(q)]$ for a pair of distinct propositional variables p and q. Then, $(M, i) \models^{+\underline{d}} (p \wedge \sim p) \rightarrow q$ does not hold.

III. Verifying Students' Learning Processes

A model of students should be inconsistency-tolerant since student's understanding is uncertain and vague in general. SPLTL can be used to express the negation of uncertain concepts such as *understand* (or *understanding*). For instance, if we cannot determine whether someone understands, the uncertain concept *understand* can be represented by asserting the inconsistent formula:

understand $\wedge \sim$understand.

This is well formalized because the formula:

(*understand* $\wedge \sim$*understand*) $\rightarrow \bot$

is not valid in paraconsistent logic. On the other hand, we can decide whether someone is learning: The decision is represented by ~*learning*, where

(*learning* $\wedge \sim$*learning*) $\rightarrow \bot$

is valid in classical logic.

It is remarked that the following negative expressions can be differently interpreted:

$\neg$*understand* (not understand),

$\sim$*understand* (not deeply understand).

The first statement indicates that a person is not understand that is inconsistent with his or her understanding. The second statement means that we can say that a person is not deeply understand, but he or she may be shallowly understand. We thus allow the situation: *understand* $\wedge \sim$*understand*.

In ontology representation, a concept hierarchy is constructed by *ISA-relations* between concepts, i.e., a concept is a subconcept of another concept. In this study, we use sequence modal operators to represent ISA-relations between concepts. Let $c_1, c_2, \ldots, c_n$ be concept symbols. Then, we write a sequence of concept names by $[c_1; c_2; \ldots ; c_n]$. Each order (c_i, c_j) $(1 \leq i < j \leq n)$ of concepts in the sequence modal operator $[c_1; c_2; \ldots ; c_n]$ can be used to represent the 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 subconcept of the concept “human."

The sequence modal operators in SPLTL 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 students' learning process in a high school. A typical student in a high school graduates three years from the entry. In Fig. 1, $\sim$*ustd* (an abbreviation of $\sim$*understand*) represents uncertain negative information that can be at the same time as *ustd* (an abbreviation of *understand*), which represents positive information.

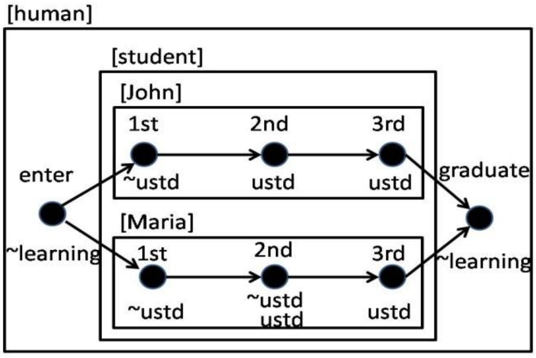

Fig. 1. Students’ learning process.

We can show a sequential paraconsistent model $M = (\sigma, I^{+\underline{d}}, I^{-\underline{d}})$ with $\underline{d} \in SE$ that corresponds to a model of students' learning processes as shown in Figure 1. For any $* \in \{+, -\}$,

1. $S = \{s_0, s_1, s_2, s_3, s_4, s_5, s_6, s_7\}$,
2. $\sigma_1 = s_0\, s_1\, s_2\, s_3\, s_7\, s_7\, s_7 \ldots$,
3. $\sigma_2 = s_0\, s_4\, s_5\, s_6\, s_7\, s_7\, s_7 \ldots$,
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 can verify: “Is there a student who is difficult to understand the lectures in the first year?” This statement is expressed as:

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

The above statement is true because we have a path $s_0 \rightarrow s_1$ with $s_1 \in I^{-John}(understand)$, $s_1 \in I^{*John}(learning)$ and $s_1 \in I^{*John}(1st)$ with $* \in \{+, -\}$. Namely, the lectures in the first year are difficult for John.

We can also verify: “Is there a student who is confusing to understand lectures?” This statement is expressed as:

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

The above statement is true because we have a path $s_0 \rightarrow s_4 \rightarrow s_5$ with $s_5 \in L^{*Maria}(learning)$, $s_5 \in I^{+Maria}(understand)$ and $s_5 \in I^{-Maria}(understand)$. Namely, To understand some lectures in the 2nd year is confusing for Maria.

IV. Embedding and Decidability

In the following, the logics LTL and PLTL are introduced. The language of LTL is obtained from that of SPLTL by deleting [b] and $\sim$, and the language of PLTL is obtained from that of LTL by adding $\sim$. A theorem for embedding PLTL into LTL is presented. The decidability of PLTL is derived from this theorem. These embedding and decidability results were originally proved in [3], and these results will be used to show the embedding and decidability results of SPLTL.

Definition 4.1 (LTL): Let S be a non-empty set of states. A structure M := (σ, I) is a *model* iff

1. σ is an infinite sequence $s_0, s_1, s_2, \ldots$ 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) ⊨ α for any formula α, where M is a model (σ, I) and i (∈ ω) represents some position within σ, is defined by

1. for any p ∈ Φ, (M, i) ⊨ p iff $s_i \in I(p)$,
2. (M, i) ⊨ α ∧ β iff (M, i) ⊨ α and (M, i) ⊨ β,
3. (M, i) ⊨ α ∨ β iff (M, i) ⊨ α or (M, i) ⊨ β,
4. (M, i) ⊨ α → β iff (M, i) ⊨ α implies (M, i) ⊨ β,
5. (M, i) ⊨ ¬α iff (M, i) ⊭ α,
6. (M, i) ⊨ Xα iff (M, i+1) ⊨ α,
7. (M, i) ⊨ Gα iff ∀j ≥i [(M, j) ⊨ α],
8. (M, i) ⊨ Fα iff ∃j ≥i [(M, j) ⊨ α].

A formula α is *valid* (*satisfiable*) in LTL iff (M, 0) ⊨ α for any (some, reps.) model M := (σ, I).

Definition 4.2 (PLTL): Let S be a non-empty set of states. A structure $M := (\sigma, I^+, I^-)$ is a *paraconsistent model* iff

1. σ is an infinite sequence $s_0, s_1, s_2, \ldots$ of states in S,
2. I^+ and I^- are mappings from the set Φ of propositional variables to the power set of S.

Satisfaction relations $(M, i) \models^+ \alpha$ and $(M, i) \models^- \alpha$ for any formula α, where M is a paraconsistent model (σ, I^+, I^-) and i (∈ ω) represents some position within σ, are defined by

1. for any p ∈ Φ, $(M, i) \models^+ p$ 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. $(M, i) \models^+ {\sim}\alpha$ iff $(M, i) \models^- \alpha$,
7. $(M, i) \models^+ X\alpha$ iff $(M, i+1) \models^+ \alpha$,
8. $(M, i) \models^+ G\alpha$ iff $\forall j \geq i\, [(M, j) \models^+ \alpha]$,
9. $(M, i) \models^+ F\alpha$ iff $\exists j \geq i\, [(M, j) \models^+ \alpha]$,
10. for any p ∈ Φ, $(M, i) \models^- p$ iff $s_i \in I^-(p)$,
11. $(M, i) \models^- \alpha \wedge \beta$ iff $(M, i) \models^- \alpha$ or $(M, i) \models^- \beta$,
12. $(M, i) \models^- \alpha \vee \beta$ iff $(M, i) \models^- \alpha$ and $(M, i) \models^- \beta$,
13. $(M, i) \models^- \alpha \rightarrow \beta$ iff $(M, i) \models^+ \alpha$ and $(M, i) \models^- \beta$,
14. $(M, i) \models^- \neg\alpha$ iff $(M, i) \not\models^- \alpha$,
15. $(M, i) \models^- {\sim}\alpha$ iff $(M, i) \models^+ \alpha$,
16. $(M, i) \models^- X\alpha$ iff $(M, i+1) \models^- \alpha$,
17. $(M, i) \models^- G\alpha$ iff $\exists j \geq i\, [(M, j) \models^- \alpha]$,
18. $(M, i) \models^- F\alpha$ iff $\forall j \geq i\, [(M, j) \models^- \alpha]$.

A formula α is *valid* (*satisfiable*) in PLTL iff $(M, 0) \models^+ \alpha$ for any (some, reps.) paraconsistent model $M := (\sigma, I^+, I^-)$.

Next, we define a translation function *g* from PLTL into LTL.

Definition 4.3: Let Φ be a non-empty set of propositional variables and Φ′ be the set {p' | p ∈ Φ} of propositional variables. The language L^p (the set of formulas) of PLTL is defined using Φ, ∼, ¬, →, ∧, ∨, F, G and X. The language *L* of LTL is obtained from L^p by adding Φ′ and deleting ∼.

A mapping *g* from L^p to *L* is defined by

1. for any p ∈ Φ, *g*(p) := p and *g*(∼p) := p' ∈ Φ′,
2. *g*(α # β) := *g*(α) # *g*(β) where # ∈ {→, ∧, ∨},
3. *g*(#α) := #*g*(α) where # ∈ {¬, X, F, G},
4. *g*(∼∼α) := *g*(α),
5. *g*(∼#α) := #*g*(∼α) where # ∈ {¬, X},
6. *g*(∼(α ∧ β)) := *g*(∼α) ∨ *g*(∼β),
7. *g*(∼(α ∨ β)) := *g*(∼α) ∧ *g*(∼β),
8. *g*(∼(α → β)) := *g*(α) ∧ *g*(∼β),
9. *g*(∼Fα) := G *g*(∼α),
10. *g*(∼Gα) := F *g*(∼α).

We can obtain the following theorems.

Theorem 4.4 (Embedding from PLTL into LTL): Let *g* be the mapping defined in Definition 4.3. For any formula α,

α is valid in PLTL iff *g*(α) is valid in LTL.

Proof: See [3].

By using this theorem, we can show the following theorem.

Theorem 4.5 (Decidability of PLTL): PLTL is decidable.

Proof: See [3].

Next, we define a translation function *f* from SPLTL into PLTL.

Definition 4.6: Let Φ be a non-empty set of propositional variables and $\Phi^{\underline{d}}$ be the set $\{p^{\underline{d}} \mid p \in \Phi\}$ ($\underline{d}$ ∈ SE) of propositional variables where $p^{\emptyset} := p$. The language L^{sp} (the set of formulas) of SPLTL is defined using Φ, ∼, ¬, →, ∧, ∨, F, G, X and [b] by the same way as in Definition 2.1. The language L^p of PLTL is obtained from L^{sp} by adding $\Phi^{\underline{d}}$ and deleting [b].

A mapping f from L^{sp} to L^p is defined by:

1. for any p ∈ Φ, $f([\underline{d}]p) := p^{\underline{d}} \in \Phi^{\underline{d}}$, especially, *f*(p) = p,
2. $f([\underline{d}](\alpha \# \beta)) := f([\underline{d}]\alpha) \# f([\underline{d}]\beta)$ where # ∈ {→, ∧, ∨},
3. $f([\underline{d}]\#\alpha) := \# f([\underline{d}]\alpha)$ where # ∈ {∼, ¬, X, F, G},
4. $f([\underline{d}][b ; c]\alpha) := f([\underline{d}][b][c]\alpha)$.

Lemma 4.7: Let *f* be the mapping defined in Definition 4.6, and S be a non-empty set of states. For any sequential paraconsistent model $M := (\sigma, I^{+\underline{d}}, I^{-\underline{d}})$ with $\underline{d}$ ∈ SE of SPLTL, any satisfaction relations $\models^{*\underline{d}}$ (* ∈ {+, -}) on M, and any state $s_i \in \sigma$, we can construct a paraconsistent model $N := (\sigma, I^+, I^-)$ of PLTL and satisfaction relations $\models^*$ on N such that for any formula α in L^{sp},

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

Proof: Let Φ be a non-empty set of propositional variables and $\Phi^{\underline{d}}$ be the set $\{p^{\underline{d}} \mid p \in \Phi\}$. Suppose that M is a sequential paraconsisitent model $(\sigma, I^{+\underline{d}}, I^{-\underline{d}})$ with $\underline{d}$ ∈ SE where

$I^{*\underline{d}}$ (* ∈{+, -}) are mappings from Φ to the power set of S.

Suppose that N is a paraconsisitent model (σ, I^+, I^-) where

I^* (* ∈ {+, -}) are mappings from $\bigcup_{\underline{d} \in SE} \Phi^{\underline{d}}$ to the power set of S.

Suppose moreover that M and N satisfy the following condition: For any $s_i \in \sigma$, any p ∈ Φ and any * ∈ {+, -},

$$s_i \in I^{*\underline{d}}(p) \ \text{ iff } \ s_i \in I^*(p^{\underline{d}}).$$

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

Base step:

Case α ≡ p ∈ Φ:

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

Induction step: We show some cases.

Case α ≡ β ∧ γ:

For the case $* \equiv +$, we obtain: $(M, i) \models^{+\underline{d}} \beta \wedge \gamma$ iff $(M, i) \models^{+\underline{d}} \beta$ and $(M, i) \models^{+\underline{d}} \gamma$ iff $(N, i) \models^{+} f([\underline{d}]\beta)$ and $(N, i) \models^{+} f([\underline{d}]\gamma)$ (by induction hypothesis) iff $(N, i) \models^{+} f([\underline{d}]\beta) \wedge f([\underline{d}]\gamma)$ iff $(N, i) \models^{+} f([d]\beta \wedge \gamma)$ (by the definition of f).

For the case $* \equiv -$, we obtain: $(M, i) \models^{-\underline{d}} \beta \wedge \gamma$ iff $(M, i) \models^{-\underline{d}} \beta$ or $(M, i) \models^{-\underline{d}} \gamma$ iff $(N, i) \models^{-} f([\underline{d}]\beta)$ or $(N, i) \models^{-} f([\underline{d}]\gamma)$ (by induction hypothesis) iff $(N,i) \models^{-} f([\underline{d}]\beta) \wedge f([\underline{d}]\gamma)$ iff $(N, i) \models^{-} f([\underline{d}](\beta \wedge \gamma))$ (by the definition of f).

Case $\alpha \equiv \beta \rightarrow \gamma$:

For the case $* \equiv +$, we obtain: $(M, i) \models^{+\underline{d}} \beta \rightarrow \gamma$ iff $(M, i) \models^{+\underline{d}} \beta$ implies $(M, i) \models^{+\underline{d}} \gamma$ iff $(N, i) \models^{+} f([\underline{d}]\beta)$ implies $(N, i) \models^{+} f([\underline{d}]\gamma)$ (by induction hypothesis) iff $(N, i) \models^{+} f([\underline{d}]\beta) \rightarrow f([\underline{d}]\gamma)$ iff $(N, i) \models^{+} f([\underline{d}](\beta \rightarrow \gamma)$ (by the definition of f).

For the case $* \equiv -$, we obtain: $(M, i) \models^{-\underline{d}} \beta \rightarrow \gamma$ iff $(M, i) \models^{+\underline{d}} \beta$ and $(M, i) \models^{-\underline{d}} \gamma$ iff $(N, i) \models^{+} f([\underline{d}]\beta)$ and $(N, i) \models^{-} f([\underline{d}]\gamma)$ (by induction hypothesis) iff $(N, i) \models^{-} f([\underline{d}]\beta) \rightarrow f([\underline{d}]\gamma)$ iff $(N, i) \models^{-} f([\underline{d}](\beta \rightarrow \gamma))$ (by the definition of f).

Case $\alpha \equiv \neg\beta$:

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

Case $\alpha \equiv \sim\beta$:

For the case $* \equiv +$, we obtain: $(M, i) \models^{+\underline{d}} \sim\beta$ iff $(M, i) \models^{-\underline{d}} \beta$ iff $(N, i) \models^{-} f([\underline{d}]\beta)$ (by induction hypothesis) iff $(N, i) \models^{+} \sim f([\underline{d}]\beta)$ iff $(N, i) \models^{+} f([\underline{d}]\sim\beta)$ (by the definition of f).

For the case $* \equiv -$, we obtain: $(M,i) \models^{-\underline{d}} \sim\beta$ iff $(M, i) \models^{+\underline{d}} \beta$ iff $(N, i) \models^{+} f([\underline{d}]\beta)$ (by induction hypothesis) iff $(N, i) \models^{-} \sim f([\underline{d}]\beta)$ iff $(N, i) \models^{-} f([\underline{d}]\sim\beta)$ (by the definition of f).

Case $\alpha \equiv X\beta$:

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

Case $\alpha \equiv G\beta$:

For the case $* \equiv +$, we obtain: $(M, i) \models^{+\underline{d}} G\beta$ iff $\forall j \geq i\ [(M, j) \models^{+\underline{d}} \beta]$ iff $\forall j \geq i\ [(N, j) \models^{+} f([\underline{d}]\beta)]$ (by induction hypothesis) iff $(N, i) \models^{+} G\ f([\underline{d}]\beta)$ iff $(N, i) \models^{+} f([\underline{d}]G\beta)$ (by the definition of f).

For the case $* \equiv -$, we obtain: $(M, i) \models^{-\underline{d}} G\beta$ iff $\exists j \geq i\ [(M, j) \models^{-\underline{d}} \beta]$ iff $\exists j \geq i\ [(N, j) \models^{-} f([\underline{d}]\beta)]$ (by induction hypothesis) iff $(N, i) \models^{-} G\ f([\underline{d}]\beta)$ iff $(N, i) \models^{-} f([\underline{d}]G\beta)$ (by the definition of f).

Case $\alpha \equiv [b]\beta$:

We obtain: $(M, i) \models^{*\underline{d}} [b]\beta$ iff $(M, i) \models^{*\underline{d}\,;\,b} \beta$ (by Proposition 2.3) iff $(N, i) \models^{*} f([\underline{d}\,;\,b]\beta)$ (by induction hypothesis) iff $(N, i) \models^{*} f([\underline{d}][b]\beta)$ by the definition of f.

Lemma 4.8: Let f be the mapping defined in Definition 4.6, and S be a non-empty set of states. For any paraconsisitent model $N := (\sigma, I^{+}, I^{-})$ of PLTL, any satisfaction relations $\models^{*}$ ($* \in \{+, -\}$) on N, and any state $s_i \in \sigma$, we can construct a sequential paraconsisitent model $M := (\sigma, I^{+\underline{d}}, I^{-\underline{d}})$ with $\underline{d} \in SE$ of SPLTL and satisfaction relations $\models^{*\underline{d}}$ ($* \in \{+, -\}$) on M such that for any formula $\alpha \in L^{sp}$,

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

Proof: Similar to the proof of Lemma 4.7.

Theorem 4.9 (Embedding from SPLTL into PLTL): Let f be the mapping defined in Definition 4.6. For any formula α, α is valid in SPLTL iff $f(\alpha)$ is valid in PLTL.

Proof: By Lemmas 4.7 and 4.8.

Theorem 4.10 (Embedding from SPLTL into LTL): Let f and g be the mappings defined in Definitions 4.6 and 4.3, respectively. For any formula α, α is valid in SPLTL iff $g\,f(\alpha)$ is valid in LTL.

Proof: By Theorems 4.4 and 4.9.

Theorem 4.11 (Decidability of SPLTL): SPLTL is decidable.

Proof: By decidability of PLTL, for each α, it is possible to decide if $f(\alpha)$ is valid in PLTL. Then, by Theorem 4.9, SPLTL is decidable.

Theorem 4.11 shows that the validity problem of SPLTL is decidable. Similarly, we can also show that both the satisfiability and model checking problems of SPLTL are decidable.

V. Conclusions

This paper introduced a new extended linear-time temporal logic (LTL), called sequential paraconsistent LTL (SPLTL), for formalizing inconsistency-tolerant reasoning with hierarchical information. Some theorems for embedding SPLTL into a paraconsistent subsystem PLTL of SPLTL and into LTL were proved, and SPLTL was shown to be decidable. The embedding and decidability results allow us to use the existing LTL-based algorithms to test the satisfiability. Thus it was shown in this paper that SPLTL can be used as an executable logic to represent inconsistency-tolerant reasoning with hierarchical information. It was also shown that SPLTL is useful for formalizing and verifying students' learning processes. In such a learning process, students' understanding, which is an inconsistent and uncertain concept, was presented using the paraconsistent negation connective in SPLTL, and certain hierarchical information on students were presented using the sequence modal operators in SPLTL.

References

[1] A. Pnueli, "The temporal logic of programs," in *Proc. the 18th IEEE Symposium on Foundations of Computer Science,* pp. 46-57, 1977.

[2] A. Almukdad and D. Nelson, "Constructible falsity and inexact predicates," *Journal of Symbolic Logic* 49, pp. 231-233, 1984.

[3] N. Kamide and H. Wansing, "A paraconsistent linear-time temporal logic,"*Fundamenta Informaticae,* vol. 106, no. 1, pp. 1-23, 2011.

[4] D. Nelson, "Constructible falsity,"*Journal of Symbolic Logic,* vol. 14, pp. 16-26, 1949.

[5] K. Kaneiwa and N. Kamide, "Sequence-indexed linear-time temporal logic: Proof system and application,"*Applied Artificial Intelligence,* vol. 24, no. 10, pp. 896-913, 2010.

[6] K. Kaneiwa and N. Kamide, "Conceptual modeling in full computation-tree logic with sequence modal operator,"*International Journal of Intelligent Systems,* vol. 26, no. 7, pp. 636-651, 2011.

[7] G. J. Holzmann, *The SPIN model checker: Primer and reference manual*, Addison-Wesley, 2006.

[8] N. Kamide and H. Wansing, "Proof theory of Nelson's paraconsistent logic: A uniform perspective," *Theoretical Computer Science,* vol. 415, pp. 1-38, 2012.

[9] H. Wansing, "The logic of information structures," *Lecture Notes in Artificial Intelligence,* vol. 681, pp. 1-163, 1993.

[10] G. Priest, "Paraconsistent logic," in *Handbook of Philosophical Logic* (Second Edition), vol. 6, D. Gabbay and F. Guenthner (eds.), Kluwer Academic Publishers, Dordrecht, pp. 287-393, 2002.

Norihiro Kamide received his Ph.D. in Information Science from Japan Advanced Institute of Science and Technology in 2000. His research interests include Theoretical Computer Science, Logics in Computer Science and in Artificial Intelligence, Mathematical Logic, and Philosophical Logic. He is a member of Japanese Society for Artificial Intelligence, Japan Society for Software Science and Technology, and Mathematical Society of Japan.