Logics and Translations for Inconsistency-tolerant Model Checking

Norihiro Kamide[1] and Kazuki Endo[2]
[1]*Teikyo University, Faculty of Science and Engineering, Department of Information and Electronic Engineering, Toyosatodai 1-1, Utsunomiya-shi, Tochigi 320-8551, Japan*
[2]*Teikyo University, Faculty of Science and Engineering, Department of Human Information Systems, Toyosatodai 1-1, Utsunomiya-shi, Tochigi 320-8551, Japan*

Keywords: Model Checking, Paraconsistent Logic, Linear-time Temporal Logic, Computation-tree Logic, Embedding Theorem.

Abstract: In this study, we develop logics and translations for inconsistency-tolerant (or paraconsistent) model checking that can be used to verify systems with inconsistencies. Paraconsistent linear-time temporal logic (pLTL) and paraconsistent computation tree logic (pCTL) are introduced, and these are extensions of standard linear-time temporal logic (LTL) and standard computation tree logic (CTL), respectively. These novel logics can be applied when handling inconsistency-tolerant temporal reasoning. These logics are also regarded as four-valued temporal logics that extend the four-valued logic of Belnap and Dunn. Translations from pLTL into LTL and pCTL into CTL are defined, and these are used to prove the theorems for embedding pLTL into LTL and pCTL into CTL. These embedding theorems allow the standard LTL- and CTL-based model checking algorithms to be used for verifying inconsistent systems that are modeled and specified by pLTL and pCTL. A new illustrative example for inconsistency-tolerant model checking is also presented on the basis of the proposed logics and translations.

1 INTRODUCTION

Inconsistencies are frequent and inevitable when verifying and specifying large, complex, and open systems. The goal of this study is to develop simple logics and translations for *inconsistency-tolerant model checking* (or *paraconsistent model checking*) that can be used to verify systems with inconsistencies. *Model checking* is a formal and automated technique for verifying concurrent systems (Clarke and Emerson, 1981; Clarke et al., 1999; Holzmann, 2006). We develop two novel and simple versions of paraconsistent four-valued temporal logics such as *paraconsistent linear-time temporal logic* (pLTL) and *paraconsistent computation tree logic* (pCTL). These are extensions of the standard temporal logics: *linear-time temporal logic* (LTL) (Pnueli, 1977) and *computation-tree logic* (CTL) (Clarke and Emerson, 1981), typically used in model checking. pLTL and pCTL may be applied when handling inconsistency-tolerant temporal reasoning, and may also provide the base logics for inconsistency-tolerant model checking. These four-valued temporal logics are also regarded as extensions of *Belnap and Dunn's four-valued logic* (Belnap, 1977b; Belnap, 1977a; Dunn, 1976). In this paper, we define the translations of pLTL into LTL and pCTL into CTL. These translations will be used to prove the theorems for embedding pLTL into LTL and pCTL into CTL. These embedding theorems allow us to repurpose the standard LTL- and CTL-based model checking algorithms for verifying inconsistent systems that are modeled and specified by pLTL and pCTL.

LTL (Pnueli, 1977) is one of the most useful temporal logics for model checking based on the *linear-time paradigm*, which uses linear order to represent the passage of time. CTL (Clarke and Emerson, 1981) is another form of temporal logic that is widely used for model checking. It is based on the *branching-time paradigm* that uses computation trees to represent the passage of time. Since these standard temporal logics lack *paraconsistency*, they are unsuitable for specifying and verifying inconsistent systems. The satisfaction relation $\models$ of a logic is considered to be paraconsistent with respect to a negation connective $\sim$ if the following condition holds: $\exists\alpha,\beta\ (M,x) \not\models (\alpha\wedge\sim\alpha)\rightarrow\beta$, where x is a state or position in a semantic structure M of the underlying logic. This con-

Kamide, N. and Endo, K.
Logics and Translations for Inconsistency-tolerant Model Checking.
DOI: 10.5220/0006640601910200
In *Proceedings of the 10th International Conference on Agents and Artificial Intelligence (ICAART 2018) - Volume 2*, pages 191-200
ISBN: 978-989-758-275-2

dition reflects that formulas of the form $(\alpha \wedge \sim\alpha)\rightarrow\beta$ are not valid in the underlying logics.

Compared to other non-classical logics, *paraconsistent logics* such as pLTL and pCTL can be appropriately used in inconsistency-tolerant reasoning (Priest, 2002; da Costa et al., 1995; Wansing, 1993). For example, the following scenario is undesirable: $(s(x) \wedge \sim s(x))\rightarrow d(x)$ is valid for any symptom s and disease d, where $\sim s(x)$ implies that "a person x does not have a symptom s" and $d(x)$ implies that "a person x suffers from a disease d." The inconsistent scenario written as $melancholia(john) \wedge \sim melancholia(john)$ will inevitably arise from the uncertain definition of melancholia; the statement "John has melancholia" may be judged true or false based on the perception of different pathologists. In this case, the formula $(melancholia(john) \wedge \sim melancholia(john))\rightarrow cancer(john)$ is valid in classical logic (as an inconsistency that has an undesirable consequence), but invalid in paraconsistent logics (as these logics are inconsistency-tolerant). Typical examples of non-temporal paraconsistent logics are Belnap and Dunn's four-valued logic (Belnap, 1977b; Belnap, 1977a; Dunn, 1976) and *Nelson's paraconsistent four-valued logic* (Almukdad and Nelson, 1984; Nelson, 1949). The proposed logics, pLTL and pCTL, are based on these typical paraconsistent four-valued logics.

The idea of introducing paraconsistent versions of LTL and CTL is not a new one. *Multi-valued computation tree logic*, χCTL, was introduced by Easterbrook and Chechik (Easterbrook and Chechik, 2001) as the base logic for *multi-valued model checking*, which is considered to be the first framework for inconsistency-tolerant model checking. *Quasi-classical temporal logic*, QCTL, was introduced by Chen and Wu (Chen and Wu, 2006) to verify inconsistent concurrent systems using inconsistency-tolerant model checking. *Paraconsistent full computation tree logic*, PCTL*, proposed by Kamide (Kamide, 2006), applied bisimulations to inconsistency-tolerant model checking. Another paraconsistent linear-time temporal logic, PLTL, was introduced by Kamide and Wansing (Kamide and Wansing, 2011) to obtain a cut-free and complete Gentzen-type sequent calculus. Another paraconsistent computation tree logic, PCTL, was proposed by Kamide and Kaneiwa (Kamide and Kaneiwa, 2010; Kaneiwa and Kamide, 2011), providing an alternative inconsistency-tolerant model checking framework. Kamide (Kamide, 2015) also introduced *sequence-indexed paraconsistent computation tree logic*, SPCTL, which extended CTL by adding a paraconsistent negation connective and a sequence modal operator. SPCTL was used for the representation and verification of medical reasoning with hierarchical and inconsistent information. *Paraconsistent probabilistic computation tree logic*, PpCTL, was introduced by Kamide and Koizumi (Kamide and Koizumi, 2016) for the verification of randomized and stochastic inconsistent systems.

In this study, we developed pLTL and pCTL as novel versions of paraconsistent linear-time temporal logic and paraconsistent computation tree logic by extending LTL and CTL, respectively. While PLTL (Kamide and Wansing, 2011), PCTL (Kamide and Kaneiwa, 2010; Kaneiwa and Kamide, 2011), SPCTL (Kamide, 2015), and PpCTL (Kamide and Koizumi, 2016) have two types of dual satisfaction relations $\models^+$ (verification or justification) and $\models^-$ (refutation or falsification), pLTL and pCTL are simpler, having a single satisfaction relation $\models^*$ that is highly compatible with the standard single satisfaction relations of LTL and CTL. These single satisfaction relations provide simple proofs for the embedding theorems of pLTL and pCTL, and the paraconsistent negation connective $\sim$ used in pLTL and pCTL can be simply formalized and uniformly handled. pLTL is also more expressive than PLTL, since it lacks the standard until and release temporal operators found in LTL. Furthermore, pLTL and pCTL employ novel sets of axiom schemes for combining the paraconsistent negation connective $\sim$, classical negation connective $\neg$, and implication connective $\rightarrow$. The negated implication and negation axioms used in pLTL and pCTL are $\sim(\alpha\rightarrow\beta) \leftrightarrow \neg\sim\alpha \wedge \sim\beta$ and $\sim\neg\alpha \leftrightarrow \neg\sim\alpha$. These recently introduced axiom schemes by De and Omori are natural and plausible from the point of view of many-valued semantics (De and Omori, 2015). The logic BD+ (De and Omori, 2015) of these axiom schemes was shown to be essentially equivalent to *Béziau's four-valued modal logic* PM4N (Beziau, 2011) and *Zaitsev's paraconsistent logic* FDEP (Zaitsev, 2012).

The contents of this paper are organized as follows.

Section 2 discusses the linear-time case based on LTL and pLTL. The new formulation pLTL is introduced on the basis of the single satisfaction relation $\models^*$. A function translating pLTL into LTL is defined. This is a simplification of the translation functions used in (Kamide and Wansing, 2011; Kamide and Kaneiwa, 2010; Kaneiwa and Kamide, 2011; Kamide, 2015; Kamide and Koizumi, 2016). The proposed translation function is then used to prove the theorem for embedding pLTL into LTL. The present and previous versions of these translation functions are regarded as modifications or extensions

of those used by Gurevich (Gurevich, 1977), Rautenberg (Rautenberg, 1979), and Vorob'ev (Vorob'ev, 1952) to embed Nelson's constructive logic (Almukdad and Nelson, 1984; Nelson, 1949) into intuitionistic logic. Similar translations have recently been used (Kamide, 2016; Kamide and Shramko, 2017) to embed some of the paraconsistent logics into classical logic.

Section 3 discusses the branching-time case based on CTL and pCTL. Similar to the linear-time case, pCTL is introduced on the basis of the single satisfaction relation $\models^*$, a function translating pCTL to CTL is defined, and the theorem for embedding pCTL into CTL is proved. The translation function is constructed in a similar manner to that of pLTL.

Section 4 presents a new illustrative example for inconsistency-tolerant model checking on the basis of the proposed logics and translations.

Section 5 concludes the paper. It is noted in that two further alternative logics, pLTL* and pCTL*, can be respectively obtained from pLTL and pCTL by replacing the axiom schemes $\sim(\alpha\to\beta) \leftrightarrow \neg\sim\alpha \wedge \sim\beta$ and $\sim\neg\alpha \leftrightarrow \neg\sim\alpha$ with the axiom schemes $\sim(\alpha\to\beta) \leftrightarrow \alpha \wedge \sim\beta$ and $\sim\neg\alpha \leftrightarrow \alpha$ by Odintsov (Odintsov, 2005). It is further noted that, by appropriate modification of the translation functions for pLTL and pCTL, the embedding theorems for pLTL* into LTL and pCTL* into CTL can also be obtained.

2 LINEAR-TIME CASE

Formulas of *linear-time temporal logic* (LTL) are constructed from countably many propositional variables, $\to$ (implication), $\wedge$ (conjunction), $\vee$ (disjunction), $\neg$ (classical negation), X (next), G (globally), F (eventually), U (until) and R (release). An expression $\alpha \leftrightarrow \beta$ is used to denote $(\alpha\to\beta)\wedge(\beta\to\alpha)$. Lower-case letters $p, q, \ldots$ are used to denote propositional variables, and Greek lower-case letters $\alpha, \beta, \ldots$ are used to denote formulas. 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 the linear order on ω. An expression $A \equiv B$ is used to indicate the syntactical identity between A and B.

Definition 2.1. *Formulas of* LTL *are defined by the following grammar, assuming p represents propositional variables:*

$$\alpha ::= p \mid \alpha\wedge\alpha \mid \alpha\vee\alpha \mid \alpha\to\alpha \mid \neg\alpha \mid \mathrm{X}\alpha \mid \mathrm{G}\alpha \mid \mathrm{F}\alpha \mid \alpha\mathrm{U}\alpha \mid \alpha\mathrm{R}\alpha.$$

Definition 2.2 (LTL)**.** *Let S be a non-empty set of states, and Φ be the set of propositional variables.*

A structure $M := (\sigma, I)$ 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 Φ 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. *$(M,i) \models \alpha\to\beta$ iff $(M,i) \models \alpha$ implies $(M,i) \models \beta$,*
5. *$(M,i) \models \neg\alpha$ iff $(M,i) \not\models \alpha$,*
6. *$(M,i) \models \mathrm{X}\alpha$ iff $(M,i+1) \models \alpha$,*
7. *$(M,i) \models \mathrm{G}\alpha$ iff $\forall j \geq i\ [(M,j) \models \alpha]$,*
8. *$(M,i) \models \mathrm{F}\alpha$ iff $\exists j \geq i\ [(M,j) \models \alpha]$,*
9. *$(M,i) \models \alpha\mathrm{U}\beta$ iff $\exists j \geq i\ [(M,j) \models \beta$ and $\forall i \leq k < j\ (M,k) \models \alpha]$,*
10. *$(M,i) \models \alpha\mathrm{R}\beta$ iff $\forall j \geq i\ [(M,j) \models \beta$ or $\exists i \leq k < j\ (M,k) \models \alpha]$.*

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

The language of *paraconsistent linear-time temporal logic* (pLTL) is obtained from that of LTL by adding $\sim$ (paraconsistent negation).

Definition 2.3. *Formulas of* pLTL *are defined by the following grammar, assuming p represents propositional variables:*

$$\alpha ::= p \mid \alpha\wedge\alpha \mid \alpha\vee\alpha \mid \alpha\to\alpha \mid \neg\alpha \mid \sim\alpha \mid \mathrm{X}\alpha \mid \mathrm{G}\alpha \mid \mathrm{F}\alpha \mid \alpha\mathrm{U}\alpha \mid \alpha\mathrm{R}\alpha.$$

Definition 2.4 (pLTL)**.** *Let S be a non-empty set of states, Φ be the set of propositional variables and $\Phi^\sim$ be the set $\{\sim p \mid p \in \Phi\}$ of negated propositional variables.*

A structure $M := (\sigma, I^*)$ is a paraconsistent model *iff*

1. *σ is an infinite sequence $s_0, s_1, s_2, \ldots$ of states in S,*
2. *I^* is a mapping from $\Phi \cup \Phi^\sim$ to the power set of S.*

A paraconsistent satisfaction relation $(M,i) \models^* \alpha$ for any formula α, where M is a paraconsistent 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. *for any $\sim p \in \Phi^\sim$, $(M,i) \models^* \sim p$ iff $s_i \in I^*(\sim p)$,*
3. *$(M,i) \models^* \alpha\wedge\beta$ iff $(M,i) \models^* \alpha$ and $(M,i) \models^* \beta$,*
4. *$(M,i) \models^* \alpha\vee\beta$ iff $(M,i) \models^* \alpha$ or $(M,i) \models^* \beta$,*
5. *$(M,i) \models^* \alpha\to\beta$ iff $(M,i) \models^* \alpha$ implies $(M,i) \models^* \beta$,*
6. *$(M,i) \models^* \neg\alpha$ iff $(M,i) \not\models^* \alpha$,*

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. $(M,i) \models^* \alpha U\beta$ *iff* $\exists j \geq i\ [(M,j) \models^* \beta$ *and* $\forall i \leq k < j\ (M,k) \models^* \alpha]$,
11. $(M,i) \models^* \alpha R\beta$ *iff* $\forall j \geq i\ [(M,j) \models^* \beta$ *or* $\exists i \leq k < j\ (M,k) \models^* \alpha]$,
12. $(M,i) \models^* {\sim}{\sim}\alpha$ *iff* $(M,i) \models^* \alpha$,
13. $(M,i) \models^* {\sim}(\alpha \wedge \beta)$ *iff* $(M,i) \models^* {\sim}\alpha$ *or* $(M,i) \models^* {\sim}\beta$,
14. $(M,i) \models^* {\sim}(\alpha \vee \beta)$ *iff* $(M,i) \models^* {\sim}\alpha$ *and* $(M,i) \models^* {\sim}\beta$,
15. $(M,i) \models^* {\sim}(\alpha \rightarrow \beta)$ *iff* $(M,i) \not\models^* {\sim}\alpha$ *and* $(M,i) \models^* {\sim}\beta$,
16. $(M,i) \models^* {\sim}\neg\alpha$ *iff* $(M,i) \not\models^* {\sim}\alpha$,
17. $(M,i) \models^* {\sim}X\alpha$ *iff* $(M,i+1) \models^* {\sim}\alpha$,
18. $(M,i) \models^* {\sim}G\alpha$ *iff* $\exists j \geq i\ [(M,j) \models^* {\sim}\alpha]$,
19. $(M,i) \models^* {\sim}F\alpha$ *iff* $\forall j \geq i\ [(M,j) \models^* {\sim}\alpha]$,
20. $(M,i) \models^* {\sim}(\alpha U\beta)$ *iff* $\forall j \geq i\ [(M,j) \models^* {\sim}\beta$ *or* $\exists i \leq k < j\ (M,k) \models^* {\sim}\alpha]$,
21. $(M,i) \models^* {\sim}(\alpha R\beta)$ *iff* $\exists j \geq i\ [(M,j) \models^* {\sim}\beta$ *and* $\forall i \leq k < j\ (M,k) \models^* {\sim}\alpha]$.

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

We make some remarks.

1. pLTL is *paraconsistent* with respect to $\sim$. The reason is explained as follows. Assume a paraconsistent model $M := (\sigma, I^*)$ such that $s_i \in I^*(p)$, $s_i \in I^*({\sim}p)$ and $s_i \notin I^*(q)$ for a pair of distinct propositional variables p and q. Then, $(M,i) \models^* (p \wedge {\sim}p) \rightarrow q$ does not hold.
2. pLTL is regarded as a four-valued logic. The reason is explained as follows. For each $i \in \sigma$ and each formula α, we can take one of the following four cases:
   (a) α is verified at i, i.e., $(M,i) \models^* \alpha$,
   (b) α is falsified at i, i.e., $(M,i) \models^* {\sim}\alpha$,
   (c) α is both verified and falsified at i,
   (d) α is neither verified nor falsified at i.

Next, we define a translation function f from pLTL into LTL.

Definition 2.5. *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* $\mathcal{L}^p$ *(the set of formulas) of* pLTL *is defined using* Φ, $\wedge, \vee, \rightarrow, \neg$, X, G, F, U, R *and* $\sim$. *The language* $\mathcal{L}$ *of* LTL *is obtained from* $\mathcal{L}^p$ *by adding* Φ' *and deleting* $\sim$.

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

1. *for any* $p \in \Phi$, $f(p) := p$ *and* $f({\sim}p) := p' \in \Phi'$,
2. $f(\alpha \sharp \beta) := f(\alpha) \sharp f(\beta)$ *where* $\sharp \in \{\wedge, \vee, \rightarrow, U, R\}$,
3. $f(\sharp\alpha) := \sharp f(\alpha)$ *where* $\sharp \in \{\neg, X, F, G\}$,
4. $f({\sim}{\sim}\alpha) := f(\alpha)$,
5. $f({\sim}(\alpha \wedge \beta)) := f({\sim}\alpha) \vee f({\sim}\beta)$,
6. $f({\sim}(\alpha \vee \beta)) := f({\sim}\alpha) \wedge f({\sim}\beta)$,
7. $f({\sim}(\alpha \rightarrow \beta)) := \neg f({\sim}\alpha) \wedge f({\sim}\beta)$,
8. $f({\sim}\sharp\alpha) := \sharp f({\sim}\alpha)$ *where* $\sharp \in \{\neg, X\}$,
9. $f({\sim}F\alpha) := Gf({\sim}\alpha)$,
10. $f({\sim}G\alpha) := Ff({\sim}\alpha)$,
11. $f({\sim}(\alpha U\beta)) := f({\sim}\alpha) R f({\sim}\beta)$,
12. $f({\sim}(\alpha R\beta)) := f({\sim}\alpha) U f({\sim}\beta)$.

In order to obtain the theorem for embedding pLTL into LTL, we need to show some lemmas.

Lemma 2.6. *Let* f *be the mapping defined in Definition 2.5, and* S *be a non-empty set of states. For any paraconsistent model* $M := (\sigma, I^*)$ *of* pLTL, *any paraconsistent satisfaction relation* $\models^*$ *on* M, *and any state* s_i *in* σ, *we can construct a model* $N := (\sigma, I)$ *of* LTL *and a satisfaction relation* $\models$ *on* N *such that for any formula* α *in* $\mathcal{L}^p$, $(M,i) \models^* \alpha$ *iff* $(N,i) \models f(\alpha)$.

Proof. Let Φ be a non-empty set of propositional variables, $\Phi^{\sim}$ be $\{{\sim}p \mid p \in \Phi\}$, and Φ' be $\{p' \mid p \in \Phi\}$. Suppose that M is a paraconsisitent model (σ, I^*) where I^* is a mapping from $\Phi \cup \Phi^{\sim}$ to the power set of S. We then define a model $N := (\sigma, I)$ such that

1. I is a mapping from $\Phi \cup \Phi'$ to the power set of S,
2. for any s_i in σ,
   (a) $s_i \in I^*(p)$ iff $s_i \in I(p)$,
   (b) $s_i \in I^*({\sim}p)$ iff $s_i \in I(p')$,

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

• Base step:

1. Case $\alpha \equiv p \in \Phi$: We obtain: $(M,i) \models^* p$ iff $s_i \in I^*(p)$ iff $s_i \in I(p)$ iff $(N,i) \models p$ iff $(N,i) \models f(p)$ (by the definition of f).
2. Case $\alpha \equiv {\sim}p \in \Phi^{\sim}$: We obtain: $(M,i) \models^* {\sim}p$ iff $s_i \in I^*({\sim}p)$ iff $s_i \in I(p')$ iff $(N,i) \models p'$ iff $(N,i) \models f({\sim}p)$ (by the definition of f).

• Induction step: We show some cases.

1. Case $\alpha \equiv \beta U\gamma$: We obtain:
   $(M,i) \models^* \beta U\gamma$
   iff $\exists j \geq i\ [(M,j) \models^* \gamma$ and $\forall i \leq k < j\ (M,k) \models^* \beta]$
   iff $\exists j \geq i\ [(N,j) \models f(\gamma)$ and $\forall i \leq k < j\ (N,k) \models f(\beta)]$ (by induction hypothesis)
   iff $(N,i) \models f(\beta) U f(\gamma)$
   iff $(N,i) \models f(\beta U\gamma)$ (by the definition of f).

2. Case $\alpha \equiv \sim(\beta \wedge \gamma)$: We obtain: $(M,i) \models^* \sim(\beta \wedge \gamma)$ iff $(M,i) \models^* \sim\beta$ or $(M,i) \models^* \sim\gamma$ iff $(N,i) \models f(\sim\beta)$ or $(N,i) \models f(\sim\gamma)$ (by induction hypothesis) iff $(N,i) \models f(\sim\beta) \vee f(\sim\gamma)$ iff $(N,i) \models f(\sim(\beta \wedge \gamma))$ (by the definition of f).
3. Case $\alpha \equiv \sim(\beta \rightarrow \gamma)$: We obtain: $(M,i) \models^* \sim(\beta \rightarrow \gamma)$ iff $(M,i) \not\models^* \sim\beta$ and $(M,i) \models^* \sim\gamma$ iff $(N,i) \not\models f(\sim\beta)$ and $(N,i) \models f(\sim\gamma)$ (by induction hypothesis) iff $(N,i) \models \neg f(\sim\beta) \wedge f(\sim\gamma)$ iff $(N,i) \models f(\sim(\beta \rightarrow \gamma))$ (by the definition of f).
4. Case $\alpha \equiv \sim\neg\beta$: We obtain: $(M,i) \models^* \sim\neg\beta$ iff $(M,i) \not\models^* \sim\beta$ iff $(N,i) \not\models f(\sim\beta)$ (by induction hypothesis) iff $(N,i) \models \neg f(\sim\beta)$ iff $(N,i) \models f(\sim\neg\beta)$ (by the definition of f).
5. Case $\alpha \equiv \sim\sim\beta$: We obtain: $(M,i) \models^* \sim\sim\beta$ iff $(M,i) \models^* \beta$ iff $(N,i) \models f(\beta)$ (by induction hypothesis) iff $(N,i) \models f(\sim\sim\beta)$ (by the definition of f).
6. Case $\alpha \equiv \sim X\beta$: We obtain: $(M,i) \models^* \sim X\beta$ iff $(M,i+1) \models^* \sim\beta$ iff $(N,i+1) \models f(\sim\beta)$ (by induction hypothesis) iff $(N,i) \models Xf(\sim\beta)$ iff $(N,i) \models f(\sim X\beta)$ (by the definition of f).
7. Case $\alpha \equiv \sim G\beta$: We obtain: $(M,i) \models^* \sim G\beta$ iff $\exists j \geq i\ [(M,j) \models^* \sim\beta]$ iff $\exists j \geq i\ [(N,j) \models f(\sim\beta)]$ (by induction hypothesis) iff $(N,i) \models Ff(\sim\beta)$ iff $(N,i) \models f(\sim G\beta)$ (by the definition of f).
8. Case $\alpha \equiv \sim(\beta U \gamma)$: We obtain:
   $(M,i) \models^* \sim(\beta U \gamma)$
   iff $\forall j \geq i\ [(M,j) \models^* \sim\gamma$ or $\exists i \leq k < j\ (M,k) \models^* \sim\beta]$
   iff $\forall j \geq i\ [(N,j) \models f(\sim\gamma)$ or $\exists i \leq k < j\ (N,k) \models f(\sim\beta)]$ (by induction hypothesis)
   iff $(N,i) \models f(\sim\beta) R f(\sim\gamma)$
   iff $(N,i) \models f(\sim(\beta U \gamma))$ (by the definition of f).
9. Case $\alpha \equiv \sim(\beta R \gamma)$: We obtain:
   $(M,i) \models^* \sim(\beta R \gamma)$
   iff $\exists j \geq i\ [(M,j) \models^* \sim\gamma$ and $\forall i \leq k < j\ (M,k) \models^* \sim\beta]$
   iff $\exists j \geq i\ [(N,j) \models f(\sim\gamma)$ and $\forall i \leq k < j\ (N,k) \models f(\sim\beta)]$ (by induction hypothesis)
   iff $(N,i) \models f(\sim\beta) U f(\sim\gamma)$
   iff $(N,i) \models f(\sim(\beta R \gamma))$ (by the definition of f).

Q.E.D.

Lemma 2.7. *Let f be the mapping defined in Definition 2.5, and S be a non-empty set of states. For any model $N := (\sigma, I)$ of* LTL*, any satisfaction relation $\models$ on N, and any state s_i in σ, we can construct a paraconsisitent model $M := (\sigma, I^*)$ of* pLTL *and a satisfaction relation $\models^*$ on M such that for any formula α in $\mathcal{L}^p$, $(N,i) \models f(\alpha)$ iff $(M,i) \models^* \alpha$.*

Proof. Similar to the proof of Lemma 2.6. **Q.E.D.**

Theorem 2.8 (Embedding from pLTL into LTL)**.** *Let f be the mapping defined in Definition 2.5. For any formula α, α is valid in* pLTL *iff $f(\alpha)$ is valid in* LTL*.*

Proof. By Lemmas 2.6 and 2.7. **Q.E.D.**

3 BRANCHING-TIME CASE

Formulas of *computation tree logic* (CTL) are constructed from countably many propositional variables, $\rightarrow$, $\wedge$, $\vee$, $\neg$, X, G, F, U, R, A (all computation paths), and E (some computation path). The same notions and notations as those in the previous sections are also used in the following.

Definition 3.1. *Formulas of* CTL *are defined by the following grammar, assuming p represents propositional variables:*

$$\begin{aligned}\alpha ::= {} & p \mid \alpha \wedge \alpha \mid \alpha \vee \alpha \mid \alpha \rightarrow \alpha \mid \neg\alpha \mid \\ & AX\alpha \mid EX\alpha \mid AG\alpha \mid EG\alpha \mid AF\alpha \mid EF\alpha \mid \\ & A(\alpha U \alpha) \mid E(\alpha U \alpha) \mid A(\alpha R \alpha) \mid E(\alpha R \alpha).\end{aligned}$$

Note that pairs of symbols like AX and EU are indivisible, and that the symbols X, G, F, U, and R cannot occur without being preceded by an A or an E. Similarly, every A or E must have one of X, G, F, U, and R to accompany it.

Definition 3.2 (CTL)**.** *A structure (S, S_0, R, L) is a* model *iff*

1. *S is the set of states,*
2. *S_0 is a set of initial states and $S_0 \subseteq S$,*
3. *R is a binary relation on S which satisfies the condition: $\forall s \in S\ \exists s' \in S\ [(s,s') \in R]$,*
4. *L is a mapping from S to the power set of a nonempty set Φ of propositional variables.*

A path *in a model is an infinite sequence of states, $\pi = s_0, s_1, s_2, \ldots$ such that $\forall i \geq 0\ [(s_i, s_{i+1}) \in R]$.*

A satisfaction relation $(M,s) \models \alpha$ *for any formula α, where M is a model (S, S_0, R, L) and s represents a state in S, is defined inductively by:*

1. *for any $p \in \Phi$, $(M,s) \models p$ iff $p \in L(s)$,*
2. *$(M,s) \models \alpha \wedge \beta$ iff $(M,s) \models \alpha$ and $(M,s) \models \beta$,*
3. *$(M,s) \models \alpha \vee \beta$ iff $(M,s) \models \alpha$ or $(M,s) \models \beta$,*
4. *$(M,s) \models \alpha \rightarrow \beta$ iff $(M,s) \models \alpha$ implies $(M,s) \models \beta$,*
5. *$(M,s) \models \neg\alpha$ iff $(M,s) \not\models \alpha$,*
6. *$(M,s) \models AX\alpha$ iff $\forall s_1 \in S\ [(s,s_1) \in R$ implies $(M,s_1) \models \alpha]$,*
7. *$(M,s) \models EX\alpha$ iff $\exists s_1 \in S\ [(s,s_1) \in R$ and $(M,s_1) \models \alpha]$,*

8. $(M,s) \models \mathrm{AG}\alpha$ *iff for all paths* $\pi \equiv s_0,s_1,s_2,...,$ *where* $s \equiv s_0$*, and all states* s_i *along* π*, we have* $(M,s_i) \models \alpha$,
9. $(M,s) \models \mathrm{EG}\alpha$ *iff there is a path* $\pi \equiv s_0,s_1,s_2,...,$ *where* $s \equiv s_0$*, and for all states* s_i *along* π*, we have* $(M,s_i) \models \alpha$,
10. $(M,s) \models \mathrm{AF}\alpha$ *iff for all paths* $\pi \equiv s_0,s_1,s_2,...,$ *where* $s \equiv s_0$*, there is a state* s_i *along* π *such that* $(M,s_i) \models \alpha$,
11. $(M,s) \models \mathrm{EF}\alpha$ *iff there is a path* $\pi \equiv s_0,s_1,s_2,...,$ *where* $s \equiv s_0$*, and for some state* s_i *along* π*, we have* $(M,s_i) \models \alpha$,
12. $(M,s) \models \mathrm{A}(\alpha \mathrm{U} \beta)$ *iff for all paths* $\pi \equiv s_0,s_1,s_2,...,$ *where* $s \equiv s_0$*, there is a state* s_j *along* π *such that* $(M,s_j) \models \beta$ *and* $\forall 0 \leq k < j$ $(M,s_k) \models \alpha$,
13. $(M,s) \models \mathrm{E}(\alpha \mathrm{U} \beta)$ *iff there is a path* $\pi \equiv s_0,s_1,s_2,...,$ *where* $s \equiv s_0$*, and for some state* s_j *along* π*, we have* $(M,s_j) \models \beta$ *and* $\forall 0 \leq k < j$ $(M,s_k) \models \alpha$,
14. $(M,s) \models \mathrm{A}(\alpha \mathrm{R} \beta)$ *iff for all paths* $\pi \equiv s_0,s_1,s_2,...,$ *where* $s \equiv s_0$*, and all states* s_j *along* π*, we have* $(M,s_j) \models \beta$ *or* $\exists 0 \leq k < j$ $(M,s_k) \models \alpha$,
15. $(M,s) \models \mathrm{E}(\alpha \mathrm{R} \beta)$ *iff there is a path* $\pi \equiv s_0,s_1,s_2,...,$ *where* $s \equiv s_0$*, and for all states* s_j *along* π*, we have* $(M,s_j) \models \beta$ *or* $\exists 0 \leq k < j$ $(M,s_k) \models \alpha$.

A formula α *is* valid *in* CTL *iff* $(M,s) \models \alpha$ *holds for any model* $M := (S,S_0,R,L)$*, any* $s \in S$*, and any satisfaction relation* $\models$ *on* M.

The language of *paraconsistent computation tree logic* (pCTL) is obtained from that of CTL by adding $\sim$.

Definition 3.3. *Formulas of* pCTL *are defined by the following grammar, assuming* p *represents propositional variables:*

$$\alpha ::= p \mid \alpha \wedge \alpha \mid \alpha \vee \alpha \mid \alpha \rightarrow \alpha \mid \neg\alpha \mid \sim\alpha \mid \mathrm{AX}\alpha \mid \mathrm{EX}\alpha \mid \mathrm{AG}\alpha \mid \mathrm{EG}\alpha \mid \mathrm{AF}\alpha \mid \mathrm{EF}\alpha \mid \mathrm{A}(\alpha \mathrm{U} \alpha) \mid \mathrm{E}(\alpha \mathrm{U} \alpha) \mid \mathrm{A}(\alpha \mathrm{R} \alpha) \mid \mathrm{E}(\alpha \mathrm{R} \alpha).$$

Definition 3.4 (pCTL)**.** *Let* Φ *be a non-empty set of propositional variables, and* $\Phi^{\sim}$ *be the set* $\{\sim p \mid p \in \Phi\}$ *of negated propositional variables.*

A structure (S,S_0,R,L^*) *is a* paraconsistent model *iff*

1. S *is the set of states,*
2. S_0 *is a set of initial states and* $S_0 \subseteq S$,
3. R *is a binary relation on* S *which satisfies the condition:* $\forall s \in S\ \exists s' \in S\ [(s,s') \in R]$,
4. L^* *is a mapping from* S *to the power set of* $\Phi \cup \Phi^{\sim}$.

A path *in a paraconsistent model is an infinite sequence of states,* $\pi = s_0,s_1,s_2,...$ *such that* $\forall i \geq 0$ $[(s_i,s_{i+1}) \in R]$.

A paraconsistent satisfaction relation $(M,s) \models^* \alpha$ *for any formula* α*, where* M *is a paraconsistent model* (S,S_0,R,L^*) *and* s *represents a state in* S*, is defined inductively by:*

1. *for any* $p \in \Phi$, $(M,s) \models^* p$ *iff* $p \in L^*(s)$,
2. *for any* $\sim p \in \Phi^{\sim}$, $(M,s) \models^* \sim p$ *iff* $\sim p \in L^*(s)$,
3. $(M,s) \models^* \alpha \wedge \beta$ *iff* $(M,s) \models^* \alpha$ *and* $(M,s) \models^* \beta$,
4. $(M,s) \models^* \alpha \vee \beta$ *iff* $(M,s) \models^* \alpha$ *or* $(M,s) \models^* \beta$,
5. $(M,s) \models^* \alpha \rightarrow \beta$ *iff* $(M,s) \models^* \alpha$ *implies* $(M,s) \models^* \beta$,
6. $(M,s) \models^* \neg\alpha$ *iff* $(M,s) \not\models^* \alpha$,
7. $(M,s) \models^* \mathrm{AX}\alpha$ *iff* $\forall s_1 \in S\ [(s,s_1) \in R$ *implies* $(M,s_1) \models^* \alpha]$,
8. $(M,s) \models^* \mathrm{EX}\alpha$ *iff* $\exists s_1 \in S\ [(s,s_1) \in R$ *and* $(M,s_1) \models^* \alpha]$,
9. $(M,s) \models^* \mathrm{AG}\alpha$ *iff for all paths* $\pi \equiv s_0,s_1,s_2,...,$ *where* $s \equiv s_0$*, and all states* s_i *along* π*, we have* $(M,s_i) \models^* \alpha$,
10. $(M,s) \models^* \mathrm{EG}\alpha$ *iff there is a path* $\pi \equiv s_0,s_1,s_2,...,$ *where* $s \equiv s_0$*, and for all states* s_i *along* π*, we have* $(M,s_i) \models^* \alpha$,
11. $(M,s) \models^* \mathrm{AF}\alpha$ *iff for all paths* $\pi \equiv s_0,s_1,s_2,...,$ *where* $s \equiv s_0$*, there is a state* s_i *along* π *such that* $(M,s_i) \models^* \alpha$,
12. $(M,s) \models^* \mathrm{EF}\alpha$ *iff there is a path* $\pi \equiv s_0,s_1,s_2,...,$ *where* $s \equiv s_0$*, and for some state* s_i *along* π*, we have* $(M,s_i) \models^* \alpha$,
13. $(M,s) \models^* \mathrm{A}(\alpha \mathrm{U} \beta)$ *iff for all paths* $\pi \equiv s_0,s_1,s_2,...,$ *where* $s \equiv s_0$*, there is a state* s_j *along* π *such that* $(M,s_j) \models^* \beta$ *and* $\forall 0 \leq k < j$ $(M,s_k) \models^* \alpha$,
14. $(M,s) \models^* \mathrm{E}(\alpha \mathrm{U} \beta)$ *iff there is a path* $\pi \equiv s_0,s_1,s_2,...,$ *where* $s \equiv s_0$*, and for some state* s_j *along* π*, we have* $(M,s_j) \models^* \beta$ *and* $\forall 0 \leq k < j$ $(M,s_k) \models^* \alpha$,
15. $(M,s) \models^* \mathrm{A}(\alpha \mathrm{R} \beta)$ *iff for all paths* $\pi \equiv s_0,s_1,s_2,...,$ *where* $s \equiv s_0$*, and all states* s_j *along* π*, we have* $(M,s_j) \models^* \beta$ *or* $\exists 0 \leq k < j$ $(M,s_k) \models^* \alpha$,
16. $(M,s) \models^* \mathrm{E}(\alpha \mathrm{R} \beta)$ *iff there is a path* $\pi \equiv s_0,s_1,s_2,...,$ *where* $s \equiv s_0$*, and for all states* s_j *along* π*, we have* $(M,s_j) \models^* \beta$ *or* $\exists 0 \leq k < j$ $(M,s_k) \models^* \alpha$,
17. $(M,s) \models^* \sim\sim\alpha$ *iff* $(M,s) \models^* \alpha$,
18. $(M,s) \models^* \sim(\alpha \wedge \beta)$ *iff* $(M,s) \models^* \sim\alpha$ *or* $(M,s) \models^* \sim\beta$,
19. $(M,s) \models^* \sim(\alpha \vee \beta)$ *iff* $(M,s) \models^* \sim\alpha$ *and* $(M,s) \models^* \sim\beta$,
20. $(M,s) \models^* \sim(\alpha \rightarrow \beta)$ *iff* $(M,s) \not\models^* \sim\alpha$ *and* $(M,s) \models^* \sim\beta$,
21. $(M,s) \models^* \sim\neg\alpha$ *iff* $(M,s) \not\models^* \sim\alpha$,

22. $(M,s) \models^* \sim\mathrm{AX}\alpha$ *iff* $\exists s_1 \in S\ [(s,s_1) \in R$ *and* $(M,s_1) \models^* \sim\alpha]$,
23. $(M,s) \models^* \sim\mathrm{EX}\alpha$ *iff* $\forall s_1 \in S\ [(s,s_1) \in R$ *implies* $(M,s_1) \models^* \sim\alpha]$,
24. $(M,s) \models^* \sim\mathrm{AG}\alpha$ *iff there is a path* $\pi \equiv s_0,s_1,s_2,...,$ *where* $s \equiv s_0$, *and for some state* s_i *along* π, *we have* $(M,s_i) \models^* \sim\alpha$,
25. $(M,s) \models^* \sim\mathrm{EG}\alpha$ *iff for all paths* $\pi \equiv s_0,s_1,s_2,...,$ *where* $s \equiv s_0$, *there is a state* s_i *along* π *such that* $(M,s_i) \models^* \sim\alpha$,
26. $(M,s) \models^* \sim\mathrm{AF}\alpha$ *iff there is a path* $\pi \equiv s_0,s_1,s_2,...,$ *where* $s \equiv s_0$, *and for all states* s_i *along* π, *we have* $(M,s_i) \models^* \sim\alpha$,
27. $(M,s) \models^* \sim\mathrm{EF}\alpha$ *iff for all paths* $\pi \equiv s_0,s_1,s_2,...,$ *where* $s \equiv s_0$, *and all states* s_i *along* π, *we have* $(M,s_i) \models^* \sim\alpha$,
28. $(M,s) \models^* \sim\mathrm{A}(\alpha\mathrm{U}\beta)$ *iff there is a path* $\pi \equiv s_0,s_1,s_2,...,$ *where* $s \equiv s_0$, *and for all states* s_j *along* π, *we have* $(M,s_j) \models^* \sim\beta$ *or* $\exists 0 \leq k < j$ $(M,s_k) \models^* \sim\alpha$,
29. $(M,s) \models^* \sim\mathrm{E}(\alpha\mathrm{U}\beta)$ *iff for all paths* $\pi \equiv s_0,s_1,s_2,...,$ *where* $s \equiv s_0$, *and all states* s_j *along* π, *we have* $(M,s_j) \models^* \sim\beta$ *or* $\exists 0 \leq k < j$ $(M,s_k) \models^* \sim\alpha$,
30. $(M,s) \models^* \sim\mathrm{A}(\alpha\mathrm{R}\beta)$ *iff there is a path* $\pi \equiv s_0,s_1,s_2,...,$ *where* $s \equiv s_0$, *and for some state* s_j *along* π, *we have* $(M,s_j) \models^* \sim\beta$ *and* $\forall 0 \leq k < j$ $(M,s_k) \models^* \sim\alpha$,
31. $(M,s) \models^* \sim\mathrm{E}(\alpha\mathrm{R}\beta)$ *iff for all paths* $\pi \equiv s_0,s_1,s_2,...,$ *where* $s \equiv s_0$, *there is a state* s_j *along* π *such that* $(M,s_j) \models^* \sim\beta$ *and* $\forall 0 \leq k < j$ $(M,s_k) \models^* \sim\alpha$.

A formula α *is* valid *in* pCTL *iff* $(M,s) \models^* \alpha$ *holds for any paraconsistent model* $M := (S,S_0,R,L^*)$, *any* $s \in S$, *and any paraconsistent satisfaction relation* $\models^*$ *on* M.

We make some remarks.

1. pCTL is paraconsistent with respect to $\sim$. The reason is explained as follows. Assume a paraconsistent model $M = (S,S_0,R,L^*)$ such that $p \in L^*(s)$, $\sim p \in L^*(s)$ and $q \notin L^*(s)$ for a pair of distinct propositional variables p and q. Then, $(M,s) \models^* (p \wedge \sim p) \rightarrow q$ does not hold.
2. pCTL is regarded as a four-valued logic. The reason is explained as follows. For each $s \in S$ and each formula α, we can take one of the following four cases:
   (a) α is verified at s, i.e., $(M,s) \models^* \alpha$,
   (b) α is falsified at s, i.e., $(M,s) \models^* \sim\alpha$,
   (c) α is both verified and falsified at s,
   (d) α is neither verified nor falsified at s.

Definition 3.5. *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* $\mathcal{L}^p$ *(the set of formulas) of* pCTL *is defined using* Φ, $\wedge,\vee,\rightarrow,\neg$, X, F, G, U, R, A, E *and* $\sim$. *The language* $\mathcal{L}$ *of* CTL *is obtained from* $\mathcal{L}^p$ *by adding* Φ' *and deleting* $\sim$.

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

1. *for any* $p \in \Phi$, $f(p) := p$ *and* $f(\sim p) := p' \in \Phi'$,
2. $f(\alpha \sharp \beta) := f(\alpha) \sharp f(\beta)$ *where* $\sharp \in \{\wedge,\vee,\rightarrow\}$,
3. $f(\sharp\alpha) := \sharp f(\alpha)$ *where* $\sharp \in \{\neg,\mathrm{AX},\mathrm{EX},\mathrm{AG},\mathrm{EG},\mathrm{AF},\mathrm{EF}\}$,
4. $f(\mathrm{A}(\alpha\mathrm{U}\beta))) := \mathrm{A}(f(\alpha)\mathrm{U}f(\beta))$,
5. $f(\mathrm{E}(\alpha\mathrm{U}\beta))) := \mathrm{E}(f(\alpha)\mathrm{U}f(\beta))$,
6. $f(\mathrm{A}(\alpha\mathrm{R}\beta))) := \mathrm{A}(f(\alpha)\mathrm{R}f(\beta))$,
7. $f(\mathrm{E}(\alpha\mathrm{R}\beta))) := \mathrm{E}(f(\alpha)\mathrm{R}f(\beta))$,
8. $f(\sim\sim\alpha) := f(\alpha)$,
9. $f(\sim(\alpha \wedge \beta)) := f(\sim\alpha) \vee f(\sim\beta)$,
10. $f(\sim(\alpha \vee \beta)) := f(\sim\alpha) \wedge f(\sim\beta)$,
11. $f(\sim(\alpha \rightarrow \beta)) := \neg f(\sim\alpha) \wedge f(\sim\beta)$,
12. $f(\sim\neg\alpha) := \neg f(\sim\alpha)$,
13. $f(\sim\mathrm{AX}\alpha) := \mathrm{EX}f(\sim\alpha)$,
14. $f(\sim\mathrm{EX}\alpha) := \mathrm{AX}f(\sim\alpha)$,
15. $f(\sim\mathrm{AG}\alpha) := \mathrm{EF}f(\sim\alpha)$,
16. $f(\sim\mathrm{EG}\alpha) := \mathrm{AF}f(\sim\alpha)$,
17. $f(\sim\mathrm{AF}\alpha) := \mathrm{EG}f(\sim\alpha)$,
18. $f(\sim\mathrm{EF}\alpha) := \mathrm{AG}f(\sim\alpha)$,
19. $f(\sim(\mathrm{A}(\alpha\mathrm{U}\beta))) := \mathrm{E}(f(\sim\alpha)\mathrm{R}f(\sim\beta))$,
20. $f(\sim(\mathrm{E}(\alpha\mathrm{U}\beta))) := \mathrm{A}(f(\sim\alpha)\mathrm{R}f(\sim\beta))$,
21. $f(\sim(\mathrm{A}(\alpha\mathrm{R}\beta))) := \mathrm{E}(f(\sim\alpha)\mathrm{U}f(\sim\beta))$,
22. $f(\sim(\mathrm{E}(\alpha\mathrm{R}\beta))) := \mathrm{A}(f(\sim\alpha)\mathrm{U}f(\sim\beta))$.

Lemma 3.6. *Let* f *be the mapping defined in Definition 3.5. For any paraconsistent model* $M := (S,S_0,R,L^*)$ *of* pCTL, *and any paraconsistent satisfaction relation* $\models^*$ *on* M, *we can construct a model* $N := (S,S_0,R,L)$ *of* CTL *and a satisfaction relation* $\models$ *on* N *such that for any formula* α *in* $\mathcal{L}^p$ *and any state* s *in* S, $(M,s) \models^* \alpha$ *iff* $(N,s) \models f(\alpha)$.

Proof. Let Φ be a nonempty set of propositional variables, $\Phi^\sim$ be $\{\sim p \mid p \in \Phi\}$, and Φ' be $\{p' \mid p \in \Phi\}$. Suppose that M is a paraconsistent model (S,S_0,R,L^*) such that L^* is a mapping from S to the power set of $\Phi \cup \Phi^\sim$. We then define a model $N := (S,S_0,R,L)$ such that

1. L is a mapping from S to the power set of $\Phi \cup \Phi'$,
2. for any $s \in S$ and any $p \in \Phi$,
   (a) $p \in L^*(s)$ iff $p \in L(s)$,
   (b) $\sim p \in L^*(s)$ iff $p' \in L(s)$.

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

• Base step:

1. Case $\alpha \equiv p \in \Phi$: We obtain: $(M,s) \models^* p$ iff $p \in L^*(s)$ iff $p \in L(s)$ iff $(N,s) \models p$ iff $(N,s) \models f(p)$ (by the definition of f).
2. We obtain: $(M,s) \models^* \sim p$ iff $\sim p \in L^*(s)$ iff $p' \in L(s)$ iff $(N,s) \models p'$ iff $(N,s) \models f(\sim p)$ (by the definition of f).

• Induction step: We show some cases.

1. Case $\alpha \equiv \sim \mathrm{AX}\beta$: We obtain: $(M,s) \models^* \sim \mathrm{AX}\beta$ iff $\exists s_1 \in S\ [(s,s_1) \in R$ and $(M,s_1) \models^* \sim\beta]$ iff $\exists s_1 \in S\ [(s,s_1) \in R$ and $(N,s_1) \models f(\sim\beta)]$ (by induction hypothesis) iff $(N,s) \models \mathrm{EX} f(\sim\beta)$ iff $(N,s) \models f(\sim \mathrm{AX}\beta)$ (by the definition of f).
2. Case $\alpha \equiv \sim \mathrm{AG}\beta$: We obtain:

   $(M,s) \models^* \sim \mathrm{AG}\beta$

   iff there is a path $\pi \equiv s_0, s_1, s_2, \ldots$, where $s \equiv s_0$, for some state s_i along π, we have $(M,s_i) \models^* \sim\beta$

   iff there is a path $\pi \equiv s_0, s_1, s_2, \ldots$, where $s \equiv s_0$, for some state s_i along π, we have $(N,s_i) \models f(\sim\beta)$ (by induction hypothesis)

   iff $(N,s) \models \mathrm{EF} f(\sim\beta)$

   iff $(N,s) \models f(\sim \mathrm{AG}\beta))$ (by the definition of f).
3. Case $\alpha \equiv \sim \mathrm{A}(\beta \mathrm{U} \gamma)$: We obtain:

   $(M,s) \models^* \sim \mathrm{A}(\beta \mathrm{U} \gamma)$

   iff there is a path $\pi \equiv s_0, s_1, s_2, \ldots$, where $s \equiv s_0$, and for all states s_j along π, we have $(M,s_j) \models^* \sim\gamma$ or $\exists 0 \leq k < j\ (M,s_k) \models^* \sim\beta$

   iff there is a path $\pi \equiv s_0, s_1, s_2, \ldots$, where $s \equiv s_0$, and for all states s_j along π, we have $(N,s_j) \models f(\sim\gamma)$ or $\exists 0 \leq k < j\ (N,s_k) \models f(\sim\beta)$ (by induction hypothesis)

   iff $(N,s) \models \mathrm{E}(f(\sim\beta) \mathrm{R} f(\sim\gamma))$

   iff $(N,s) \models f(\sim \mathrm{A}(\beta \mathrm{U} \gamma))$ (by the definition of f).
4. Case $\alpha \equiv \sim \mathrm{A}(\beta \mathrm{R} \gamma)$: We obtain:

   $(M,s) \models^* \sim \mathrm{A}(\beta \mathrm{R} \gamma)$

   iff there is a path $\pi \equiv s_0, s_1, s_2, \ldots$, where $s \equiv s_0$, and for some state s_j along π, we have $(M,s_j) \models^* \sim\gamma$ and $\forall 0 \leq k < j\ (M,s_k) \models^* \sim\beta$

   iff there is a path $\pi \equiv s_0, s_1, s_2, \ldots$, where $s \equiv s_0$, and for some state s_j along π, we have $(N,s_j) \models f(\sim\gamma)$ or $\forall 0 \leq k < j\ (N,s_k) \models f(\sim\beta)$ (by induction hypothesis)

   iff $(N,s) \models \mathrm{E}(f(\sim\beta) \mathrm{U} f(\sim\gamma))$

   iff $(N,s) \models f(\sim \mathrm{A}(\beta \mathrm{R} \gamma))$ (by the definition of f).

Q.E.D.

Lemma 3.7. *Let f be the mapping defined in Definition 3.5. For any model $N := (S, S_0, R, L)$ of* CTL*, and any satisfaction relation $\models$ on N, we can construct a paraconsistent model $M := (S, S_0, R, L^*)$ of* pCTL *and a paraconsistent satisfaction relation $\models^*$ on M such that for any formula α in $\mathcal{L}^p$ and any state s in S, $(N,s) \models f(\alpha)$ iff $(M,s) \models^* \alpha$,*

Proof. Similar to the proof of Lemma 3.6. **Q.E.D.**

Theorem 3.8 (Embedding from pCTL into CTL)**.** *Let f be the mapping defined in Definition 3.5. For any formula α, α is valid in* pCTL *iff $f(\alpha)$ is valid in* CTL*.*

Proof. By Lemmas 3.6 and 3.7. **Q.E.D.**

4 ILLUSTRATIVE EXAMPLE

We present a new illustrative example for inconsistency-tolerant model checking, as shown in Figure 1 for representing the health of a person who has a tumor. The proposed example is regarded as a modification of the example presented in (Kamide and Kaneiwa, 2010; Kaneiwa and Kamide, 2011).

In this example, a paraconsistent negation connective $\sim$ is used to express the negation of ambiguous concepts. If we cannot determine whether someone is healthy, then the ambiguous concept *healthy* can be represented by asserting the inconsistent formula $\mathit{healthy} \wedge \sim \mathit{healthy}$. This is well-formalized because $(\mathit{healthy} \wedge \sim \mathit{healthy}) \rightarrow \bot$ is not valid in pLTL and pCTL. On the other hand, we can decide whether someone has a tumor. The decision is represented by *hasTumor* or $\neg$*hasTumor*, where $(\mathit{hasTumor} \wedge \neg \mathit{hasToumor}) \rightarrow \bot$ is valid in pLTL and pCTL.

In the model of Figure 1, the initial state implies that a person is healthy. When a person undergoes a medical checkup, his or her state changes to one of the two states. If a tumor is detected in a person by the medical checkup, he or she is both healthy and not healthy, i.e., both *healthy* and $\sim$*healthy* are true, because it is unknown if the tumor is malignant (i.e., cancer) or not. If cancer is detected in a person (i.e., the tumor is diagnosed with cancer), then $\sim$*healthy* is true. This means that the person is not healthy, but he or she may return to good health if the cancer is completely removed by surgical operation. Moreover, when the cancer increases, the diagnosis reveals worse cancer. If the cancer is cured, the person will be healthy. Otherwise, if the cancer is not controlled, the person will die.

We can verify the statement "Is there a state in which a person is both healthy and not healthy?" This statement is true and expressed as: $\mathrm{EF}(\mathit{healthy} \wedge \sim \mathit{healthy})$. We can also verify the statement "Is

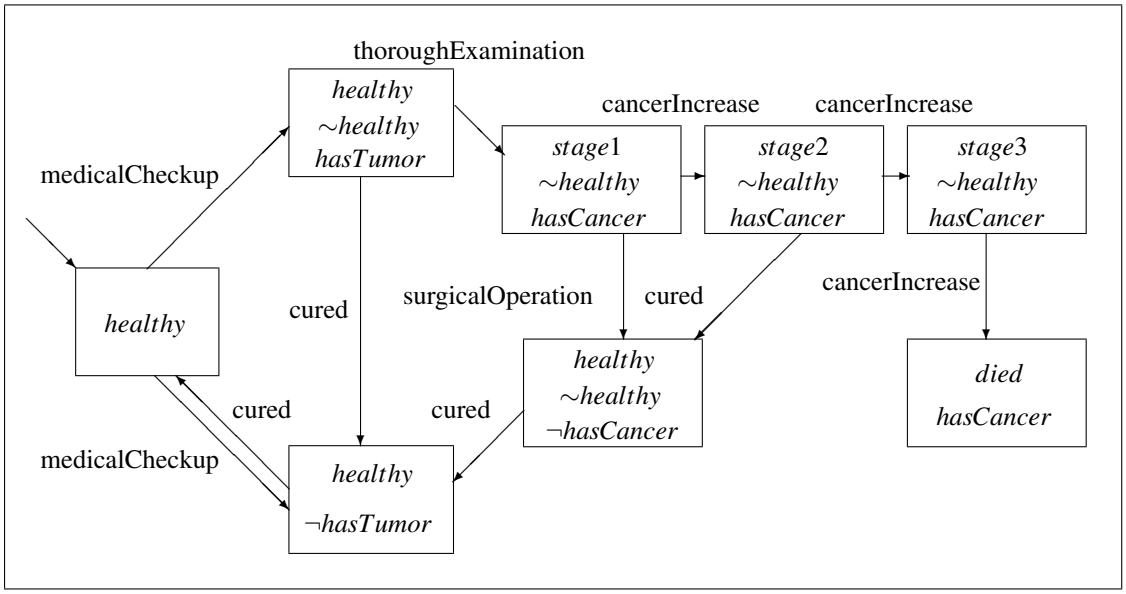

Figure 1: A clinical reasoning process model.

there a state in which a dead person will not be alive again?" This statement is true and expressed as: EF($died \wedge \neg$EF$\neg died$).

As already pointed out in (Kamide and Kaneiwa, 2010; Kaneiwa and Kamide, 2011), two negative expressions can be differently interpreted as $\neg healthy$ (definitely unhealthy) and $\sim healthy$ (not healthy). The first statement indicates that a person is definitely unhealthy that is inconsistent with his or her health. The second statement means that we can say that a person is not healthy but he or she may be healthy. The interpretation of the two negations leads to some useful verification examples. For example, the statement "Is there a state in which a person is not definitely unhealthy?" can be expressed as EF$\neg\neg healthy$. Moreover, the statement "Is there a state in which it is not true that a person is not healthy?" can be expressed as: EF$\neg\sim healthy$.

5 CONCLUDING REMARKS

In this paper, we proposed pLTL and pCTL as novel versions of paraconsistent linear-time temporal logic and paraconsistent computation tree logic, respectively. These provided a logical basis for inconsistency-tolerant model checking, and were developed by extending the standard temporal logics LTL and CTL. These are also regarded as extensions of Belnap and Dunn's four-valued logics. The translations from pLTL into LTL and pCTL into CTL were defined, and were used to prove the theorems for embedding pLTL into LTL and pCTL into CTL. It was thus demonstrated that the standard LTL- and CTL-based model checking algorithms can be repurposed for verifying inconsistent systems that are modeled and specified using pLTL and pCTL. A new illustrative example for verifying clinical reasoning process was presented on the basis of the proposed logics and translations.

Finally, we note that the proposed framework is applicable to other new variants pLTL* and pCTL* of pLTL and pCTL, respectively. The proposed logics pLTL and pCTL have the axiom schemes $\sim(\alpha\rightarrow\beta) \leftrightarrow \neg\sim\alpha \wedge \sim\beta$ and $\sim\neg\alpha \leftrightarrow \neg\sim\alpha$ by De and Omori (De and Omori, 2015), using the paraconsistent negation connective $\sim$ and the classical negation connective $\neg$. These axiom schemes are known to be plausible candidates for combining $\sim$ and $\neg$ within a logic (De and Omori, 2015). Our framework is equally applicable to the logics pLTL* and pCTL*. These are obtained from pLTL and pCTL by replacing the following clauses for $x \in \{i,s\}$:

1. $(M,x) \models^* \sim(\alpha\rightarrow\beta)$
   iff $(M,x) \not\models^* \sim\alpha$ and $(M,x) \models^* \sim\beta$,
2. $(M,x) \models^* \sim\neg\alpha$ iff $(M,x) \not\models^* \sim\alpha$,

with the following clauses for $x \in \{i,s\}$, which just correspond to the axiom schemes $\sim(\alpha\rightarrow\beta) \leftrightarrow \alpha \wedge \sim\beta$ and $\sim\neg\alpha \leftrightarrow \alpha$ by Odintsov (Odintsov, 2005):

1. $(M,x) \models^* \sim(\alpha\rightarrow\beta)$
   iff $(M,x) \models^* \alpha$ and $(M,x) \models^* \sim\beta$,

2. $(M,x) \models^* \sim\neg\alpha$ iff $(M,x) \models^* \alpha$.

By applying appropriate modifications to the translation functions of pLTL and pCTL, we obtain the embedding theorems of pLTL* into LTL and pCTL* into CTL, in the same way as with pLTL and pCTL.

ACKNOWLEDGEMENTS

We would like to thank the anonymous referees for their valuable comments. We would also like to thank Yosuke Matsuo and Ryu Yano for their assistance of this research. This research has been supported by the Kayamori Foundation of Informational Science Advancement. This research was partially supported by JSPS KAKENHI Grant (C) JP26330263.

REFERENCES

Almukdad, A. and Nelson, D. (1984). Constructible falsity and inexact predicates. *Journal of Symbolic Logic*, 49:231–233.

Belnap, N. (1977a). How a computer should think. *Contemporary Aspects of Philosophy, (G. Ryle ed.), Oriel Press, Stocksfield*, pages 30–56.

Belnap, N. (1977b). A useful four-valued logic. *Modern Uses of Multiple-Valued Logic, G. Epstein and J. M. Dunn, eds. Dordrecht: Reidel*, pages 5–37.

Beziau, J.-Y. (2011). A new four-valued approach to modal logic. *Logique et Analyse*, 54 (213):109–121.

Chen, D. and Wu, J. (2006). Reasoning about inconsistent concurrent systems: A non-classical temporal logic. In *Lecture Notes in Computer Science*, volume 3831, pages 207–217.

Clarke, E. and Emerson, E. (1981). Design and synthesis of synchronization skeletons using branching time temporal logic. In *Lecture Notes in Computer Science*, volume 131, pages 52–71.

Clarke, E., Grumberg, O., and Peled, D. (1999). *Model checking*. The MIT Press.

da Costa, N., Beziau, J., and Bueno, O. (1995). Aspects of paraconsistent logic. *Bulletin of the IGPL*, 3 (4):597–614.

De, M. and Omori, H. (2015). Classical negation and expansions of belnap-dunn logic. *Studia Logica*, 103 (4):825–851.

Dunn, J. (1976). Intuitive semantics for first-degree entailment and 'coupled trees'. *Philosophical Studies*, 29 (3):146–168.

Easterbrook, S. and Chechik, M. (2001). A framework for multi-valued reasoning over inconsistent viewpoints. In *Proceedings of the 23rd International Conference on Software Engineering*, pages 411–420.

Gurevich, Y. (1977). Intuitionistic logic with strong negation. *Studia Logica*, 36:49–59.

Holzmann, G. (2006). *The SPIN model checker: Primer and reference manual*. Addison-Wesley.

Kamide, N. (2006). Extended full computation tree logics for paraconsistent model checking. *Logic and Logical Philosophy*, 15 (3):251–276.

Kamide, N. (2015). Inconsistency-tolerant temporal reasoning with hierarchical information. *Information Sciences*, 320:140–155.

Kamide, N. (2016). Paraconsistent double negation that can simulate classical negation. In *Proceedings of the 46th IEEE International Symposium on Multiple-Valued Logic (ISMVL 2016)*, pages 131–136.

Kamide, N. and Kaneiwa, K. (2010). Paraconsistent negation and classical negation in computation tree logic. In *Proceedings of the 2nd International Conference on Agents and Artificial Intelligence (ICAART 2010), Vol.1*, pages 464–469.

Kamide, N. and Koizumi, D. (2016). Method for combining paraconsistency and probability in temporal reasoning. *Journal of Advanced Computational Intelligence and Intelligent Informatics*, 20:813–827.

Kamide, N. and Shramko, Y. (2017). Embedding from multilattice logic into classical logic and vice versa. *Journal of Logic and Computation*, 25 (5):1549–1575.

Kamide, N. and Wansing, H. (2011). A paraconsistent linear-time temporal logic. *Fundamenta Informaticae*, 106 (1):1–23.

Kaneiwa, K. and Kamide, N. (2011). Paraconsistent computation tree logic. *New Generation Computing*, 29 (4):391–408.

Nelson, D. (1949). Constructible falsity. *Journal of Symbolic Logic*, 14:16–26.

Odintsov, S. (2005). The class of extensions of nelson paraconsistent logic. *Studia Logica*, 80:291–320.

Pnueli, A. (1977). The temporal logic of programs. In *Proceedings of the 18th IEEE Symposium on Foundations of Computer Science*, pages 46–57.

Priest, G. (2002). Paraconsistent logic, handbook of philosophical logic (second edition), d. gabbay and f. guenthner (eds.). *Handbook of Philosophical Logic (Second Edition), D. Gabbay and F. Guenthner (eds.)*, 6:287–393.

Rautenberg, W. (1979). *Klassische und nicht-klassische Aussagenlogik*. Vieweg, Braunschweig.

Vorob'ev, N. (1952). A constructive propositional calculus with strong negation (in Russian). *Doklady Akademii Nauk SSR*, 85:465–468.

Wansing, H. (1993). *The logic of information structures*. Springer.

Zaitsev, D. (2012). *Generalized relevant logic and models of reasoning*. Moscow State Lomonosov University (Doctoral Dissertation).