PARACONSISTENT NEGATION AND CLASSICAL NEGATION IN COMPUTATION TREE LOGIC

Norihiro Kamide
Waseda Institute for Advanced Study, 1-6-1 Nishi Waseda, Shinjuku-ku, Tokyo 169-8050, Japan

Ken Kaneiwa
National Institute of Information and Communications Technology, 3-5 Hikaridai, Seika, Soraku, Kyoto 619-0289, Japan

Keywords: Computation tree logic, Paraconsistent logic, Decidability, Medical reasoning.

Abstract: A paraconsistent computation tree logic, PCTL, is obtained by adding paraconsistent negation to the standard computation tree logic CTL. PCTL can be used to appropriately formalize inconsistency-tolerant temporal reasoning. A theorem for embedding PCTL into CTL is proved. The validity, satisfiability, and model-checking problems of PCTL are shown to be decidable. The embedding and decidability results indicate that we can reuse the existing CTL-based algorithms for validity, satisfiability and model-checking. An illustrative example of medical reasoning involving the use of PCTL is presented.

1 INTRODUCTION

Computation tree logic (CTL) (Clarke and Emerson, 1981) is known to be one of the most useful temporal logics for verifying concurrent systems by *model checking* (Clarke et al., 1999), since some CTL-based model checking algorithms are more efficient than other types of algorithms. However, the use of CTL is not suitable for verifying "inconsistent" concurrent systems since CTL is based on classical logic. Handling inconsistencies in concurrent systems requires the use of a *paraconsistent logic* (Beziau, 1999; Priest and Routley, 1982) as a base logic for CTL.

One of the most useful paraconsistent logics is *Nelson's four-valued paraconsistent logic* N4 (or also called N^-) (Almukdad and Nelson, 1984; Nelson, 1949), which includes a paraconsistent negation connective. The logic N4 and its variants have been studied by many researchers (see, e.g., (Wagner, 1991; Wansing, 1993) and the references therein). N4 has been extensively studied since it has the property of *paraconsistency* (Beziau, 1999; da Costa et al., 1995; Priest and Routley, 1982). Roughly, a satisfaction relation $\models$ is said to be paraconsistent with respect to a negation connective $\sim$ if the following condition holds: $\exists\alpha,\beta$, not-$[M,s \models (\alpha\wedge{\sim}\alpha)\rightarrow\beta]$, where s is a state of a Kripke structure M. In contrast to N4, classical logic has no paraconsistency because the formula of the form $(\alpha\wedge{\sim}\alpha)\rightarrow\beta$ is valid in classical logic.

It is known that paraconsistent logical systems are more appropriate for inconsistency-tolerant and uncertainty reasoning than other types of logical systems (Beziau, 1999; da Costa et al., 1995; Priest and Routley, 1982; Wagner, 1991; Wansing, 1993). For example, the following scenario is undesirable $(s(x)\wedge{\sim}s(x))\rightarrow d(x)$ is satisfied for any symptom s and disease d where ${\sim}s(x)$ means "a person x does not have a symptom s" and $d(x)$ means "a person x suffers from a disease d." An inconsistent scenario expressed, for example, as $melancholia(john)\wedge{\sim}melancholia(john)$ will inevitably occur, because melancholia is an uncertain concept and the fact "John has melancholia" may be determined to be true or false by different pathologists with different perspectives. In this case, the undesirable formula $(melancholia(john)\wedge{\sim}melancholia(john))\rightarrow cancer(john)$ is valid in classical logic (i.e., an inconsistency has undesirable consequences), while it is not valid in paraconsistent logics (i.e., these logics are inconsistency-tolerant).

Inconsistencies often appear and are inevitable when specifying large, complex systems in some CTL-based frameworks. N4 is then useful and appropriate as a base logic for CTL. Moreover, N4 has notable two consequence relations $\models^+$ (verification) and $\models^-$ (refutation) in the Kripke semantics. By us-

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 - Artificial Intelligence*, pages 464-469
DOI: 10.5220/0002699504640469
Copyright © SciTePress

ing these consequence relations, the ideas of "verification (or justification)" and "refutation (or falsification)" can be simultaneously incorporated into the system. Therefore, the combination of CTL and N4 is regarded as a natural candidate for obtaining a useful paraconsistent temporal logic.

In this paper, a new paraconsistent computation tree logic called PCTL is introduced by combining CTL and N4. While the idea of combining CTL and N4 is new, the idea of introducing a paraconsistent computation tree logic is not. For example, a *multi-valued computation tree logic* χCTL was introduced by Easterbrook and Chechik (Easterbrook and Chechik, 2001), and a *quasi-classical temporal logic* QCTL was developed by Chen and Wu (Chen and Wu, 2006). Thus, PCTL is introduced as an alternative to these logics, and N4 replaces the base paraconsistent logic.

As mentioned above, the application for which paraconsistent logics show the greatest promise may be medical informatics. Indeed, it has been pointed out that paraconsistent logics are useful for medical reasoning (see, e.g., (da Costa et al., 1995; Murata et al., 1991) and the references therein). Some paraconsistent computation tree logics, including PCTL, may be more useful in medical informatics because the notion of time is necessary in order to appropriately formalize realistic medical reasoning. Against this background, we present an illustrative example of medical reasoning. The proposed illustrative example can also be adapted to other paraconsistent computation tree logics such as χCTL and QCTL.

2 PARACONSISTENT COMPUTATION TREE LOGIC

Formulas of PCTL are constructed from countable atomic formulas, $\rightarrow$ (implication) $\wedge$ (conjunction), $\vee$ (disjunction), $\neg$ (classical negation), $\sim$ (paraconsistent negation), X (next), G (globally), F (eventually), U (until), R (release), A (all computation paths) and E (some computation path). The symbols X, G, F, U and R are called *temporal operators*, and the symbols A and E are called *path quantifiers*. The symbol ATOM is used to denote the set of atomic formulas. An expression $A \equiv B$ is used to denote the syntactical identity between A and B.

Definition 2.1. Formulas α are defined by the following grammar, assuming $p \in$ ATOM:

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

Note that pairs of symbols like AG 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. Remark that all the connectives displayed above are needed to obtain an embedding theorem of PCTL into CTL.

Definition 2.2. A *paraconsistent Kripke structure* is a structure $\langle S, S_0, R, L^+, L^- \rangle$ such that

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^+ and L^- are functions from S to the power set of a nonempty subset AT of ATOM.

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

The logic PCTL is then defined as a paraconsistent Kripke structure with two satisfaction relations $\models^+$ and $\models^-$. The intuitive meanings of $\models^+$ and $\models^-$ are "verification (or justification)" and "refutation (or falsification)", respectively (Wansing, 1993).

Definition 2.3. Let AT be a nonempty subset of ATOM. *Satisfaction relations* $\models^+$ and $\models^-$ on a paraconsistent Kripke structure $M = \langle S, S_0, R, L^+, L^- \rangle$ are defined inductively as follows (s represents a state in S):

1. for any $p \in$ AT, $M, s \models^+ p$ iff $p \in L^+(s)$,
2. $M, s \models^+ \alpha_1{\rightarrow}\alpha_2$ iff $M, s \models^+ \alpha_1$ implies $M, s \models^+ \alpha_2$,
3. $M, s \models^+ \alpha_1 \wedge \alpha_2$ iff $M, s \models^+ \alpha_1$ and $M, s \models^+ \alpha_2$,
4. $M, s \models^+ \alpha_1 \vee \alpha_2$ iff $M, s \models^+ \alpha_1$ or $M, s \models^+ \alpha_2$,
5. $M, s \models^+ \neg\alpha_1$ iff not-$[M, s \models^+ \alpha_1]$,
6. $M, s \models^+ {\sim}\alpha$ iff $M, s \models^- \alpha$,
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 $\pi \equiv s_0, s_1, s_2, \ldots$, 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 $\pi \equiv s_0, s_1, s_2, \ldots$, 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 $\pi \equiv s_0, s_1, s_2, \ldots$, 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 $\pi \equiv s_0, s_1, s_2, \ldots$, where $s \equiv s_0$, and for some state s_i along π, we have $M, s_i \models^+ \alpha$,

13. $M,s \models^+ \mathrm{A}(\alpha_1 \mathrm{U} \alpha_2)$ iff for all paths $\pi \equiv s_0, s_1, s_2, ...$, where $s \equiv s_0$, there is a state s_k along π such that $[(M,s_k \models^+ \alpha_2)$ and $\forall j\ (0 \leq j < k$ implies $M,s_j \models^+ \alpha_1)]$,
14. $M,s \models^+ \mathrm{E}(\alpha_1 \mathrm{U} \alpha_2)$ iff there is a path $\pi \equiv s_0, s_1, s_2, ...$, where $s \equiv s_0$, and for some state s_k along π, we have $[(M,s_k \models^+ \alpha_2)$ and $\forall j\ (0 \leq j < k$ implies $M,s_j \models^+ \alpha_1)]$,
15. $M,s \models^+ \mathrm{A}(\alpha_1 \mathrm{R} \alpha_2)$ 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 $[\forall i < j$ not-$[M,s_i \models^+ \alpha_1]$ implies $M,s_j \models^+ \alpha_2]$,
16. $M,s \models^+ \mathrm{E}(\alpha_1 \mathrm{R} \alpha_2)$ 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 $[\forall i < j$ not-$[M,s_i \models^+ \alpha_1]$ implies $M,s_j \models^+ \alpha_2]$,
17. for any $p \in \mathrm{AT}$, $M,s \models^- p$ iff $p \in L^-(s)$,
18. $M,s \models^- \alpha_1 \rightarrow \alpha_2$ iff $M,s \models^+ \alpha_1$ and $M,s \models^- \alpha_2$,
19. $M,s \models^- \alpha_1 \wedge \alpha_2$ iff $M,s \models^- \alpha_1$ or $M,s \models^- \alpha_2$,
20. $M,s \models^- \alpha_1 \vee \alpha_2$ iff $M,s \models^- \alpha_1$ and $M,s \models^- \alpha_2$,
21. $M,s \models^- \neg\alpha_1$ iff $M,s \models^+ \alpha_1$,
22. $M,s \models^- {\sim}\alpha_1$ iff $M,s \models^+ \alpha_1$,
23. $M,s \models^- \mathrm{AX}\alpha$ iff $\exists s_1 \in S\ \ [(s,s_1) \in R$ and $M,s_1 \models^- \alpha]$,
24. $M,s \models^- \mathrm{EX}\alpha$ iff $\forall s_1 \in S\ \ [(s,s_1) \in R$ implies $M,s_1 \models^- \alpha]$,
25. $M,s \models^- \mathrm{AG}\alpha$ 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 $M,s_i \models^- \alpha$,
26. $M,s \models^- \mathrm{EG}\alpha$ 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 $M,s_i \models^- \alpha$,
27. $M,s \models^- \mathrm{AF}\alpha$ 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 $M,s_i \models^- \alpha$,
28. $M,s \models^- \mathrm{EF}\alpha$ 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 $M,s_i \models^- \alpha$,
29. $M,s \models^- \mathrm{A}(\alpha_1 \mathrm{U} \alpha_2)$ 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 $[\forall i < j$ not-$[M,s_i \models^- \alpha_1]$ implies $M,s_j \models^- \alpha_2]$,
30. $M,s \models^- \mathrm{E}(\alpha_1 \mathrm{U} \alpha_2)$ iff for all paths $\pi \equiv s_0, s_1, s_2, ...$, where $s \equiv s_0$, and for all states s_j along π, we have $[\forall i < j$ not-$[M,s_i \models^- \alpha_1]$ implies $M,s_j \models^- \alpha_2]$,
31. $M,s \models^- \mathrm{A}(\alpha_1 \mathrm{R} \alpha_2)$ iff there is a path $\pi \equiv s_0, s_1, s_2, ...$, where $s \equiv s_0$, and for some state s_k along π, we have $[(M,s_k \models^- \alpha_2)$ and $\forall j\ (0 \leq j < k$ implies $M,s_j \models^- \alpha_1)]$,
32. $M,s \models^- \mathrm{E}(\alpha_1 \mathrm{R} \alpha_2)$ iff for all paths $\pi \equiv s_0, s_1, s_2, ...$, where $s \equiv s_0$, there is a state s_k along π such that $[(M,s_k \models^- \alpha_2)$ and $\forall j\ (0 \leq j < k$ implies $M,s_j \models^- \alpha_1)]$.

Definition 2.4. A formula α is *valid* (satisfiable) in PCTL if and only if $M,s \models^+ \alpha$ holds for any (some) paraconsistent Kripke structure $M = \langle S, S_0, R, L^+, L^- \rangle$, any (some) $s \in S$, and any (some) satisfaction relations $\models^+$ and $\models^-$ on M.

Definition 2.5. Let M be a paraconsistent Kripke structure $\langle S, S_0, R, L^+, L^- \rangle$ for PCTL, and $\models^+$ and $\models^-$ be satisfaction relations on M. Then, the *positive and negative model checking problems* for PCTL are respectively defined by: for any formula α, find the sets $\{s \in S \mid M,s \models^+ \alpha\}$ and $\{s \in S \mid M,s \models^- \alpha\}$.

An expression $\alpha \leftrightarrow \beta$ is used to represent $(\alpha \rightarrow \beta) \wedge (\beta \rightarrow \alpha)$.

Proposition 2.6. The following formulas concerning paraconsistent negation are valid in PCTL: for any formulas α and β,

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. ${\sim}\neg\alpha \leftrightarrow \alpha$,
6. ${\sim}\mathrm{AX}\alpha \leftrightarrow \mathrm{EX}{\sim}\alpha$,
7. ${\sim}\mathrm{EX}\alpha \leftrightarrow \mathrm{AX}{\sim}\alpha$,
8. ${\sim}\mathrm{AG}\alpha \leftrightarrow \mathrm{EF}{\sim}\alpha$,
9. ${\sim}\mathrm{EG}\alpha \leftrightarrow \mathrm{AF}{\sim}\alpha$,
10. ${\sim}\mathrm{AF}\alpha \leftrightarrow \mathrm{EG}{\sim}\alpha$,
11. ${\sim}\mathrm{EF}\alpha \leftrightarrow \mathrm{AG}{\sim}\alpha$,
12. ${\sim}\mathrm{A}(\alpha \mathrm{U} \beta) \leftrightarrow \mathrm{E}(({\sim}\alpha)\mathrm{R}({\sim}\beta))$,
13. ${\sim}\mathrm{E}(\alpha \mathrm{U} \beta) \leftrightarrow \mathrm{A}(({\sim}\alpha)\mathrm{R}({\sim}\beta))$,
14. ${\sim}\mathrm{A}(\alpha \mathrm{R} \beta) \leftrightarrow \mathrm{E}(({\sim}\alpha)\mathrm{U}({\sim}\beta))$,
15. ${\sim}\mathrm{E}(\alpha \mathrm{R} \beta) \leftrightarrow \mathrm{A}(({\sim}\alpha)\mathrm{U}({\sim}\beta))$.

For each $s \in S$ and each formula α, we can take one of the following four cases: (1) α is verified at s, i.e., $M,s \models^+ \alpha$, (2) α is falsified at s, i.e., $M,s \models^- \alpha$, (3) α is both verified and falsified at s, and (4) α is neither verified nor falsified at s. Thus, PCTL is regarded as a four-valued logic.

Assume a paraconsistent Kripke structure $M = \langle S, S_0, R, L^+, L^- \rangle$ such that $p \in L^+(s)$, $p \in L^-(s)$ and $q \notin L^+(s)$ for any distinct atomic formulas p and q. Then, $M,s \models^+ (p \wedge {\sim}p) \rightarrow q$ does not hold, and hence $\models^+$ in PCTL is paraconsistent with respect to ${\sim}$.

In order to define a translation of PCTL into CTL, CTL is defined below.

Definition 2.7 (CTL). A *Kripke structure* for CTL is a structure $\langle S, S_0, R, L \rangle$ such that

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 function from S to the power set of a nonempty subset AT of ATOM.

A *satisfaction relation* $\models$ on a Kripke structure $M = \langle S, S_0, R, L \rangle$ for CTL is defined by the same conditions 1–5 and 7–16 as in Definition 2.3 (by deleting the superscript $+$). The validity, satisfiability and model-checking problems for CTL are defined similarly as those for PCTL.

Remark that $\models^+$ of PCTL includes $\models$ of CTL, and hence PCTL is an extension of CTL.

3 EMBEDDING AND DECIDABILITY

In the following, we introduce a translation of PCTL into CTL, and by using this translation, we show an embedding theorem of PCTL into CTL. A similar translation has been used by Gurevich (Gurevich, 1977), Rautenberg (Rautenberg, 1979) and Vorob'ev (Vorob'ev, 1952) to embed Nelson's three-valued constructive logic (Almukdad and Nelson, 1984; Nelson, 1949) into intuitionistic logic.

Definition 3.1. Let AT be a non-empty subset of ATOM, and AT$'$ be the set $\{p' \mid p \in \text{AT}\}$ of atomic formulas. The language $\mathcal{L}^{\sim}$ (the set of formulas) of PCTL is defined using AT, $\sim, \neg, \rightarrow, \wedge, \vee$, X, F, G, U, R, A and E. The language $\mathcal{L}$ of CTL is obtained from $\mathcal{L}^{\sim}$ by adding AT$'$ and deleting $\sim$.

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

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

Lemma 3.2. Let f be the mapping defined in Definition 3.1. For any paraconsistent Kripke structure $M := \langle S, S_0, R, L^+, L^- \rangle$ for PCTL, and any satisfaction relations $\models^+$ and $\models^-$ on M, there exist a Kripke structure $N := \langle S, S_0, R, L \rangle$ for CTL and a satisfaction relation $\models$ on N such that for any formula α in $\mathcal{L}^{\sim}$ and any state s in S,

1. $M, s \models^+ \alpha$ iff $N, s \models f(\alpha)$,
2. $M, s \models^- \alpha$ iff $N, s \models f(\sim\alpha)$.

Proof. Suppose that M is a paraconsistent Kripke structure $\langle S, S_0, R, L^+, L^- \rangle$ such that L^+ and L^- are functions from S to the power set of AT. Suppose that N is a Kripke structure $M := \langle S, S_0, R, L \rangle$ such that L is a function from S to the power set of AT$\cup$AT$'$. Suppose moreover that for any $s \in S$ and any $p \in \text{AT}$, (1): $p \in L^+(s)$ iff $p \in L(s)$ and (2): $p \in L^-(s)$ iff $p' \in L(s)$.

The lemma is then proved by (simultaneous) induction on the complexity of α. The base step is obvious. We show some cases for the induction step.

Case $\alpha \equiv \sim\beta$: For (1), we obtain: $M, s \models^+ \sim\beta$ iff $M, s \models^- \beta$ iff $N, s \models f(\sim\beta)$ (by induction hypothesis for 2). For (2), we obtain: $M, s \models^- \sim\beta$ iff $M, s \models^+ \beta$ iff $N, s \models f(\beta)$ (by induction hypothesis for 1) iff $N, s \models f(\sim\sim\beta)$ (by the definition of f).

Case $\alpha \equiv \text{A}(\beta \text{U} \gamma)$: For (1), we obtain:

$M, s \models^+ \text{A}(\beta \text{U} \gamma)$

iff for all paths $\pi \equiv s_0, s_1, s_2, \ldots$, where $s \equiv s_0$, there is a state s_k along π such that $[M, s_k \models^+ \gamma$ and $\forall j [i \leq j < k$ implies $M, s_j \models^+ \beta]$

iff for all paths $\pi \equiv s_0, s_1, s_2, \ldots$, where $s \equiv s_0$, there is a state s_k along π such that $[N, s_k \models f(\gamma)$ and $\forall j [i \leq j < k$ implies $N, s_j \models f(\beta)]$ (by induction hypothesis for 1)

iff $N, s \models \text{A}(f(\beta) \text{U} f(\gamma))$

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

For (2), we obtain:

$M, s \models^- \text{A}(\beta \text{U} \gamma)$

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 $[\forall i < j$ not-$[M, s_i \models^- \beta]$ implies $M, s_j \models^- \gamma]$

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 $[\forall i < j$ not-$[N, s_i \models f(\sim\beta)]$ implies $N, s_j \models f(\sim\gamma)]$ (by induction hypothesis for 2)

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). ∎

Lemma 3.3. Let f be the mapping defined in Definition 3.1. For any Kripke structure $N := \langle S, S_0, R, L\rangle$ for CTL, and any satisfaction relation $\models$ on N, there exist a paraconsistent Kripke structure $M := \langle S, S_0, R, L^+, L^-\rangle$ for PCTL and satisfaction relations $\models^+$ and $\models^-$ on M such that for any formula α in $\mathcal{L}^\sim$ and any state s in S,

1. $N, s \models f(\alpha)$ iff $M, s \models^+ \alpha$,
2. $N, s \models f(\sim\alpha)$ iff $M, s \models^- \alpha$.

Proof. Similar to the proof of Lemma 3.2. ∎

Theorem 3.4 (Embedding). Let f be the mapping defined in Definition 3.1. For any formula α, α is valid in PCTL iff $f(\alpha)$ is valid in CTL.

Proof. By Lemmas 3.2 and 3.3. ∎

Theorem 3.5 (Decidability). The model-checking, validity and satisfiability problems for PCTL are decidable.

Proof. By f in Definition 3.1, a formula α of PCTL can finitely be transformed into the corresponding formula $f(\alpha)$ of CTL. By Lemmas 3.2 and 3.3 and Theorem 3.4, the model checking, validity and satisfiability problems for PCTL can be transformed into those of CTL. Since the model checking, validity and satisfiability problems for CTL are decidable, the problems for PCTL are also decidable. ∎

4 ILLUSTRATIVE EXAMPLE

We now consider examples of state structures for representing the health of non-smokers and smokers, as shown in Figure 1. In the state structure, the medical state of a person is described in a decision diagram where branching-tree structures and negative connectives from PCTL are employed. In this example, a paraconsistent negation $\sim\alpha$ in PCTL is used to express the negation of ambiguous concepts. For instance, if we cannot determine whether someone is healthy, the ambiguous concept *healthy* can be represented by asserting the inconsistent formula $healthy \wedge \sim healthy$. This is well formalized because $(healthy \wedge \sim healthy) \rightarrow \perp$ is not valid in paraconsistent logic. On the other hand, we can decide whether someone is smoking; the decision is represented by *smoking* or $\neg smoking$, where $(smoking \wedge \neg smoking) \rightarrow \perp$ is valid in classical logic.

In Figure 1, the initial state implies that a person is not smoking ($\neg smoking$ is true). The system can move to the other state to indicate that the person is smoking (*smoking* is true). When a person undergoes a medical checkup, his or her state changes to one of the two states. Even if no cancer is detected in a smoker during the medical checkup, he or she is both healthy and not healthy, i.e., both *healthy* and $\sim healthy$ are true because smoking is detrimental to health. If cancer is detected (*hasCancer* is true) in a non-smoker (or smoker), 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 does not increase. In these states, $\sim healthy$ represents ambiguous negative information that can be true at the same time as *healthy*, which represents positive information

Moreover, when the cancer increases, the diagnosis reveals worse cancer. If the cancer is cured, the patient will be healthy. Otherwise, if the cancer is not controlled, the patient will die.

We define a Kripke structure $M = \langle S, S_0, R, L^+, L^-\rangle$ that corresponds to the medical state structure as follows:

1. $S = \{s_0, s_1, s_2, s_3, s_4, s_5, s_6\}$,
2. $S_0 = \{s_0\}$,
3. $R = \{(s_0,s_1),(s_0,s_2),(s_0,s_3),(s_1,s_0),(s_1,s_3),(s_1,s_4),(s_2,s_3),(s_3,s_2),(s_3,s_4),(s_3,s_5),(s_4,s_3),(s_5,s_2),(s_5,s_6)\}$,
4. $L^+(s_0) = \emptyset$,
5. $L^+(s_1) = \{smoking\}$,
6. $L^+(s_2) = \{healthy\}$,
7. $L^+(s_3) = \{hasCancer\}$,
8. $L^+(s_4) = \{healthy\}$,
9. $L^+(s_5) = \{cancerIncrease, hasCancer\}$,
10. $L^+(s_6) = \{died, hasCancer\}$,
11. $L^-(s_0) = L^-(s_1) = L^-(s_2) = L^-(s_5) = L^-(s_6) = \emptyset$,
12. $L^-(s_3) = L^-(s_4) = \{healthy\}$.

We can verify the existence of a path that represents the required information in the structure M. For example, we can verify the following statement: "Is there a state in which a person is both healthy and not healthy?" This statement is expressed as: $\mathrm{EF}(healthy \wedge \sim healthy)$. The above statement is true because we have a path $s_0 \rightarrow s_1 \rightarrow s_4$ where $healthy \in L^+(s_4)$ and $healthy \in L^-(s_4)$.

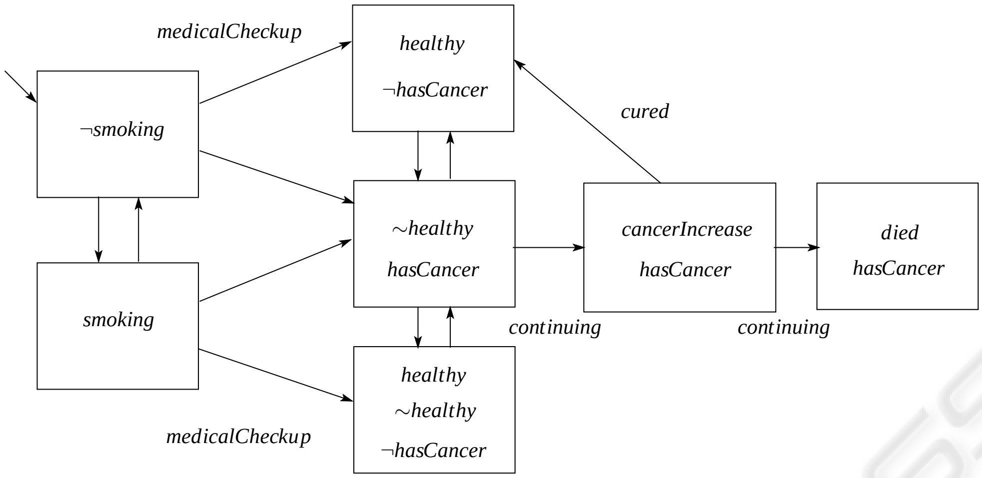

Figure 1: State structure for representing the health of smokers and non-smokers.

5 CONCLUSIONS

A new paraconsistent computation tree logic, PCTL, was introduced by combining CTL and Nelson's paraconsistent logic N4. This logic could be used appropriately in medical reasoning to deal with inconsistent data and uncertain concepts. The theorem for embedding PCTL into CTL was proved. The validity, satisfiability, and model-checking problems of PCTL were shown to be decidable. The embedding and decidability results indicate that we can reuse the existing CTL-based algorithms to test the validity, satisfiability, and model-checking. Thus, it was shown that PCTL can be used as an executable logic to represent temporal reasoning on paraconsistency. We believe that PCTL can be extensively used for inconsistency-tolerant and uncertainty reasoning, since N4 and its variants are known to be very useful for a wide range of applications such as logic programming and knowledge representations (see, e.g., (Odintsov and Wansing, 2003; Wagner, 1991) and the references therein).

REFERENCES

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

Beziau, J.-Y. (1999). The future of paraconsistent logic. *Logical Studies*, 2:Online.

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.

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.

Murata, T., Subrahmanian, V., and Wakayama, T. (1991). A petri net model for reasoning in the presence of inconsistency. *IEEE Transactions on Knowledge and Data Engineering*, 3 (3):281–292.

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

Odintsov, S. and Wansing, H. (2003). Inconsistency-tolerant description logic: Motivation and basic systems. In *Trends in Logic: 50 Years of Studia Logica*, pages 301–335.

Priest, G. and Routley, R. (1982). Introduction: paraconsistent logics. *Studia Logica*, 43:3–16.

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.

Wagner, G. (1991). Logic programming with strong negation and inexact predicates. *Journal of Logic and Computation*, 1 (6):835–859.

Wansing, H. (1993). The logic of information structures. In *Lecture Notes in Computer Science*, volume 681, pages 1–163.