Falsification-aware Semantics for CTL and Its Inconsistency-tolerant Subsystem: Towards Falsification-aware Model Checking

Norihiro Kamide and Seidai Kanbe
Teikyo University, Faculty of Science and Engineering, Department of Information and Electronic Engineering, Toyosatodai 1-1, Utsunomiya-shi, Tochigi 320-8551, Japan

Keywords: Computation Tree Logic, Inconsistency-tolerant Computation Tree Logic, Falsification-aware Kripke-style Semantics, Falsification-aware Model Checking.

Abstract: This study introduces two falsification-aware Kripke-style semantics for computation tree logic (CTL). The equivalences among the proposed falsification-aware Kripke-style semantics and the standard Kripke-style semantics for CTL are proven. Furthermore, a new logic, inconsistency-tolerant CTL (ICTL) is semantically defined and obtained from the proposed falsification-aware Kripke-style semantics for CTL by deleting a characteristic condition on the labeling function of the semantics. Because ICTL is regarded as an inconsistency-tolerant and many-valued logic, the proposed semantic framework for CTL and ICTL is regarded as a unified framework for combining and generalizing the standard, inconsistency-tolerant, and many-valued semantic frameworks. This unified semantic framework is useful for generalized model checking, referred to here as falsification-aware model checking.

1 INTRODUCTION

Computation tree logic (CTL) (Clarke and Emerson, 1981) is known to be one of the most useful temporal logics for *model checking* (Clarke and Emerson, 1981; Clarke et al., 2018), which is a computer-assisted method used to verify concurrent systems that can be modeled by state-transition systems. Some inconsistency-tolerant variants of CTL, referred to as *inconsistency-tolerant* CTLs, *paraconsistent* CTLs, and *many-valued* CTLs, have been developed as a theoretical basis for *inconsistency-tolerant (paraconsistent and many-valued) model checking* (Easterbrook and Chechik, 2001; Chen and Wu, 2006). Some inconsistency-tolerant CTLs have *falsification-aware Kripke-style semantics*, which are capable of representing the explicit falsification of a given negated formula and are appropriate for specifying and verifying inconsistency-tolerant reasoning. For more information on inconsistency-tolerant temporal logics and their model checking applications, see (Easterbrook and Chechik, 2001; Chen and Wu, 2006; Kamide, 2006; Kamide and Wansing, 2011; Kamide and Kaneiwa, 2010; Kaneiwa and Kamide, 2011; Kamide, 2015; Kamide and Koizumi, 2016; Kamide and Endo, 2018).

In this study, we first introduce two new falsification-aware Kripke-style semantics for CTL: *falsification-aware normal Kripke-style semantics* and *falsification-aware dual Kripke-style semantics*. We then prove the equivalences among the proposed falsification-aware Kripke-style semantics and the standard Kripke-style semantics for CTL. Second, we also semantically define a new logic, *inconsistency-tolerant* CTL (ICTL), which is obtained from the proposed falsification-aware normal and dual Kripke-style semantics for CTL by deleting a characteristic condition on the labeling function of the semantics. Because ICTL is regarded as an inconsistency-tolerant and many-valued logic, the proposed semantic framework for CTL and ICTL is regarded as a unified framework for combining and generalizing the standard, inconsistency-tolerant, and many-valued semantic frameworks. This unified semantic framework is useful for combined and generalized model checking, referred to here as *falsification-aware model checking*.

Next, we explain the motivation for developing falsification-aware Kripke-style semantics for CTL and ICTL. An adequate representation of falsification-aware reasoning is considered a major concern in philosophical logic (Horn and Wansing, 2017). It was suggested in (Kamide, 2021) that se-

Kamide, N. and Kanbe, S.
Falsification-aware Semantics for CTL and Its Inconsistency-tolerant Subsystem: Towards Falsification-aware Model Checking.
DOI: 10.5220/0010803700003116
In *Proceedings of the 14th International Conference on Agents and Artificial Intelligence (ICAART 2022) - Volume 3*, pages 242-252
ISBN: 978-989-758-547-0; ISSN: 2184-433X

mantics and/or proof systems for a logic can be considered *falsification-aware* if they are capable of providing (or representing) the direct (or explicit) falsifications or refutations of given negated formulas (except for negated atomic formulas). Furthermore, falsification-aware Kripke-style semantics are suitable for the theoretical basis of model checking because falsification plays a critical role in obtaining counterexample traces for the underlying object specifications in model checking. In fact, a falsification-aware technique for explicitly dividing falsification (or refutation) and verification (or justification) is often used for model checking (Gurfinkel et al., 2006). A counterexample-guided abstraction and refinement technique (Clarke et al., 2003) for model checking, which is based on falsification-aware Kripke-style semantics in temporal logics, is also an example of a useful technique in model checking. However, the standard Kripke-style semantics for CTL is not falsification-aware. Thus, we intend to develop falsification-aware Kripke-style semantics for CTL to obtain a concrete logical foundation of falsification-aware model checking, which is defined as model checking based on falsification-aware semantics.

The proposed falsification-aware normal and dual Kripke-style semantics for CTL are constructed on the basis of the idea of falsification-aware settings for *Nelson's inconsistency-tolerant (or paraconsistent) four-valued logic* N4 (Almukdad and Nelson, 1984; Nelson, 1949). Moreover, they are regarded as generalizations of the previously proposed falsification-aware Kripke-style semantics for some inconsistency-tolerant CTLs. The proposed falsification-aware Kripke-style semantics for CTL have no standard condition "$\models \neg\alpha$ iff $\not\models \alpha$" with respect to the satisfaction relation $\models$ and the classical negation connective $\neg$. This standard condition represents the transformation of a negated formula to a non-negated formula (i.e., it is not falsification-aware). Instead of this condition, the falsification-aware dual Kripke-style semantics for CTL has two clearly divided satisfaction relations $\models^+$ and $\models^-$, which explicitly represent verification (or justification) and falsification (or refutation), respectively. Thus, we can obtain the appropriate falsification-aware reasoning with $\models^-$. The proposed falsification-aware Kripke-style semantics have the merit of simply defining a natural inconsistency-tolerant CTL as a subsystem of CTL. In fact, the inconsistency-tolerant temporal logic ICTL proposed here can be defined in a modular way from the falsification-aware Kripke-style semantics for CTL by deleting only one mapping condition for the labeling function of the semantics. This type of simple definition cannot be obtained using the standard Kripke-style semantics for CTL. Thus, we can generalize the framework and concept of previously proposed inconsistency-tolerant (or paraconsistent) temporal logics and inconsistency-tolerant (paraconsistent or many-valued) model checking by using the proposed falsification-aware Kripke-style semantics and the new concept of falsification-aware model checking. In what follows, we discuss existing inconsistency-tolerant temporal logics and inconsistency-tolerant model checking.

Compared to other non-classical logics, *inconsistency-tolerant (or paraconsistent) logics* including inconsistency-tolerant temporal logics have no *axiom of explosion*, $(\alpha\wedge\neg\alpha)\rightarrow\beta$. Hence, they can be appropriately used in inconsistency-tolerant reasoning (Priest and Tanaka, 2009; da Costa et al., 1995; Wansing, 1993). For example, the following undesirable scenario is considered. The formula $(s(x)\wedge\neg s(x))\rightarrow d(x)$, which is an instance of the axiom of explosion, is valid for any symptom s and disease d, where $\neg s(x)$ represents "a person x does not have a symptom s" and $d(x)$ represents "a person x suffers from a disease d." The inconsistent scenario written as $clinical\text{-}depression(Maria)\wedge\neg clinical\text{-}depression(Maria)$ will inevitably arise from the uncertain definition of clinical depression (or melancholia). The statement "Maria has clinical depression" can be judged as true or false on the basis of the perception of different pathologists. In this case, the formula $(clinical\text{-}depression(Maria)\wedge\neg clinical\text{-}depression(Maria))\rightarrow cancer(Maria)$ is valid in classical logic and standard temporal logics (as an inconsistency that has an undesirable consequence) but invalid in inconsistency-tolerant logics (as these logics are inconsistency-tolerant). ICTL is also regarded as an inconsistency-tolerant logic, although CTL is not.

Several inconsistency-tolerant temporal logics have been developed. For example, *multi-valued computation tree logic*, referred to as χCTL, was developed by Easterbrook and Chechik (Easterbrook and Chechik, 2001) to realize *multi-valued model checking*, which is regarded as the first consideration for inconsistency-tolerant model checking. *Quasiclassical temporal logic*, referred to as QCTL, was developed by Chen and Wu (Chen and Wu, 2006) to handle inconsistent concurrent systems using inconsistency-tolerant model checking. *Paraconsistent full computation tree logic*, referred to as 4CTL*, was proposed by Kamide (Kamide, 2006) for an expressive base logic for inconsistency-tolerant model checking. A *paraconsistent linear-time temporal logic*, referred to as PLTL, based on a Gentzen-style sequent calculus was developed by Kamide

and Wansing (Kamide and Wansing, 2011) to analyze philosophical issues. A *paraconsistent computation tree logic*, referred to as PCTL, was developed by Kamide and Kaneiwa (Kamide and Kaneiwa, 2010; Kaneiwa and Kamide, 2011) to realize efficient inconsistency-tolerant CTL-model checking. Another paraconsistent CTL, referred to as pCTL, and paraconsistent linear-time temporal logic, referred to as pLTL, were developed by Kamide and Endo (Kamide and Endo, 2018; Kamide and Endo, 2019) to realize simple and efficient inconsistency-tolerant model checking. We remark that the name pCTL for paraconsistent CTL is rather confusing because the same name has also been used for *probabilistic* CTL (Aziz et al., 1995; Bianco and de Alfaro, 1995), and this probabilistic temporal logic pCTL has been used for *probabilistic model checking*. Probabilistic temporal logic was developed by Aziz et al. (Aziz et al., 1995) and Bianco and de Alfaro (Bianco and de Alfaro, 1995) to verify probabilistic concurrent systems. In the present study, ICTL is regarded as or closely related to the classical negation-free subsystems of PCTL and the inconsistency-tolerant temporal logic pCTL. Namely, ICTL is regarded as a purely inconsistency-tolerant subsystem of PCTL and pCTL.

Several extensions of inconsistency-tolerant temporal logics have been developed. For example, a *sequence-indexed paraconsistent computation tree logic*, referred to as SPCTL, was developed by Kamide (Kamide, 2015) by extending CTL with the addition of both a paraconsistent negation connective and a sequence modal operator. SPCTL was used to verify clinical reasoning with inconsistent and hierarchical information. An *inconsistency-tolerant (or paraconsistent) probabilistic computation tree logic*, referred to as PpCTL, was developed by Kamide and Koizumi (Kamide and Koizumi, 2015; Kamide and Koizumi, 2016) to verify stochastic or randomized inconsistent concurrent systems. A *locative inconsistency-tolerant hierarchical probabilistic computation tree logic*, referred to as LIHpCTL, was developed by Kamide and Bernal (Kamide and Bernal J.P.A., 2019) as an extension of PpCTL and the *hierarchical (or sequential) computation tree logic*, referred to as sCTL, developed in (Kamide and Yano, 2017; Kamide, 2018). An *inconsistency-tolerant (or paraconsistent) hierarchical probabilistic computation tree logic*, referred to as IHpCTL, was developed by Kamide and Yamamoto (Kamide and Yamamoto, 2021) as a modified version of the location-operator-free subsystem of LIHpCTL. For more information on (extended) inconsistency-tolerant temporal logics and their model checking applications, see (Easterbrook and Chechik, 2001; Chen and Wu, 2006; Kamide, 2006; Kamide and Wansing, 2011; Kamide and Kaneiwa, 2010; Kaneiwa and Kamide, 2011; Kamide, 2015; Kamide and Koizumi, 2016; Kamide and Endo, 2018).

The remainder of this paper is organized as follows. In Section 2, as the preliminaries of this study, we define the standard syntax and Kripke-style semantics for CTL. In Section 3, we introduce the falsification-aware normal Kripke-style semantics for CTL and prove the equivalence between the standard Kripke-style semantics and the proposed falsification-aware normal Kripke-style semantics. In Section 4, we introduce the falsification-aware dual Kripke-style semantics for CTL and prove the equivalence between the standard Kripke-style semantics and the proposed falsification-aware dual Kripke-style semantics. Further, we obtain the equivalences among the standard and two proposed falsification-aware Kripke-style semantics for CTL. In Section 5, we first semantically define ICTL by deleting a characteristic mapping condition from the falsification-aware normal and dual Kripke-style semantics for CTL. We then prove the equivalence between the falsification-aware normal and dual Kripke-style semantics for ICTL. In Section 6, we conclude this paper and address some remarks on falsification-aware model checking based on the proposed and extended falsification-aware Kripke-style semantics.

2 NORMAL KRIPKE-STYLE SEMANTICS FOR CTL

Formulas of CTL are constructed from countably many propositional variables by the logical connectives: $\rightarrow$ (implication), $\wedge$ (conjunction), $\vee$ (disjunction), and $\neg$ (negation); the temporal operators: X (next-time), G (globally or any-time in the future), F (eventually or some-time in the future), U (until), and R (release); and the path quantifiers: A (all computation paths) and E (some computation path). We use an expression $\alpha \leftrightarrow \beta$ to denote $(\alpha \rightarrow \beta) \wedge (\beta \rightarrow \alpha)$. We use lower-case letters $p, q, \ldots$ to denote propositional variables, Greek lower-case letters $\alpha, \beta, \ldots$ to denote formulas, and lower-case letters i, j and k to denote any natural numbers. We use the symbol ω to represent the set of natural numbers, the symbol $\geq$ or $\leq$ to denote the linear order on ω, and the symbol $\equiv$ to denote the equality of symbols.

Definition 2.1. *Formulas of* CTL *are defined by the following Backus-Naur form, assuming p represents propositional variables:*

$$\alpha ::= p \mid \alpha \wedge \alpha \mid \alpha \vee \alpha \mid \alpha \rightarrow \alpha \mid \neg \alpha$$

| AXα | EXα | AGα | EGα | AFα | EFα | A(αUα) | E(αUα) | A(αRα) | E(αRα).

Definition 2.2 (Normal Kripke-style semantics for CTL)**.** *A structure (S,S_0,R,L) is called a* CTL-model *if*

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 such that $\forall s \in S\ \exists s' \in S\ [(s,s') \in R]$,*
4. *L is a mapping from S to the power set of a nonempty set Φ of propositional variables.*

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

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

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 $(M,s) \not\models \alpha$ or $(M,s) \models \beta$,*
5. *$(M,s) \models \neg\alpha$ iff $(M,s) \not\models \alpha$,*
6. *$(M,s) \models \mathrm{AX}\alpha$ iff $\forall s_1 \in S\ [(s,s_1) \notin R$ or $(M,s_1) \models \alpha]$,*
7. *$(M,s) \models \mathrm{EX}\alpha$ 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 any paths $\pi \equiv s_0,s_1,s_2,\ldots$ with $s \equiv s_0$, and any states s_i along π, we have $(M,s_i) \models \alpha$,*
9. *$(M,s) \models \mathrm{EG}\alpha$ iff there exists a path $\pi \equiv s_0,s_1,s_2,\ldots$ with $s \equiv s_0$, and for any states s_i along π, we have $(M,s_i) \models \alpha$,*
10. *$(M,s) \models \mathrm{AF}\alpha$ iff for any paths $\pi \equiv s_0,s_1,s_2,\ldots$ with $s \equiv s_0$, there exists a state s_i along π such that $(M,s_i) \models \alpha$,*
11. *$(M,s) \models \mathrm{EF}\alpha$ iff there exists a path $\pi \equiv s_0,s_1,s_2,\ldots$ with $s \equiv s_0$, 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 any paths $\pi \equiv s_0,s_1,s_2,\ldots$ with $s \equiv s_0$, there exists 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 exists a path $\pi \equiv s_0,s_1,s_2,\ldots$ with $s \equiv s_0$, 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 any paths $\pi \equiv s_0,s_1,s_2,\ldots$ with $s \equiv s_0$, and any 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 exists a path $\pi \equiv s_0,s_1,s_2,\ldots$ with $s \equiv s_0$, and for any 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 called valid *in* CTL *if $(M,s) \models \alpha$ holds for any CTL-model $M := (S,S_0,R,L)$, any $s \in S$, and any CTL-satisfaction relation $\models$ on M.*

3 FALSIFICATION-AWARE NORMAL KRIPKE-STYLE SEMANTICS FOR CTL

A falsification-aware normal Kripke-style semantics for CTL is defined as follows.

Definition 3.1 (Falsification-aware normal Kripke-style semantics for CTL)**.** *Let Φ be a non-empty set of propositional variables and $\Phi^{\neg}$ be the set $\{\neg p \mid p \in \Phi\}$ of negated propositional variables.*

A structure $(S,S_0,R,L^\star)$ is called a falsification-aware normal CTL-model *if*

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 such that $\forall s \in S\ \exists s' \in S\ [(s,s') \in R]$,*
4. *$L^\star$ is a mapping from S to the power set of $\Phi\cup\Phi^{\neg}$ such that for any $p \in \Phi$ and any $s \in S$, $p \in L^\star(s)$ iff $\neg p \notin L^\star(s)$.*

A path *in a falsification-aware normal CTL-model is defined in the same way as that for CTL-model.*

A falsification-aware normal CTL-satisfaction relation $(M,s) \models^\star \alpha$ *for any formula α, where M is a falsification-aware normal CTL-model $(S,S_0,R,L^\star)$ and s is a state in S, is defined inductively by the following clauses:*

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

14. $(M,s) \models^{\star} \mathrm{E}(\alpha \mathrm{R} \beta)$ *iff there exists a path* $\pi \equiv s_0, s_1, s_2, \ldots$ *with* $s \equiv s_0$, *and for any states* s_j *along* π, *we have* $(M,s_j) \models^{\star} \beta$ *or* $\exists 0 \leq k < j\ (M,s_k) \models^{\star} \alpha$,
15. *for any* $\neg p \in \Phi^{\neg}$, $(M,s) \models^{\star} \neg p$ *iff* $\neg p \in L^{\star}(s)$,
16. $(M,s) \models^{\star} \neg\neg\alpha$ *iff* $(M,s) \models^{\star} \alpha$,
17. $(M,s) \models^{\star} \neg(\alpha \wedge \beta)$ *iff* $(M,s) \models^{\star} \neg\alpha$ *or* $(M,s) \models^{\star} \neg\beta$,
18. $(M,s) \models^{\star} \neg(\alpha \vee \beta)$ *iff* $(M,s) \models^{\star} \neg\alpha$ *and* $(M,s) \models^{\star} \neg\beta$,
19. $(M,s) \models^{\star} \neg(\alpha \rightarrow \beta)$ *iff* $(M,s) \models^{\star} \alpha$ *and* $(M,s) \models^{\star} \neg\beta$,
20. $(M,s) \models^{\star} \neg \mathrm{AX}\alpha$ *iff* $\exists s_1 \in S\ [(s,s_1) \in R$ *and* $(M,s_1) \models^{\star} \neg\alpha]$,
21. $(M,s) \models^{\star} \neg \mathrm{EX}\alpha$ *iff* $\forall s_1 \in S\ [(s,s_1) \notin R$ *or* $(M,s_1) \models^{\star} \neg\alpha]$,
22. $(M,s) \models^{\star} \neg \mathrm{AG}\alpha$ *iff there exists a path* $\pi \equiv s_0, s_1, s_2, \ldots$ *with* $s \equiv s_0$, *and for some state* s_i *along* π, *we have* $(M,s_i) \models^{\star} \neg\alpha$,
23. $(M,s) \models^{\star} \neg \mathrm{EG}\alpha$ *iff for any paths* $\pi \equiv s_0, s_1, s_2, \ldots$ *with* $s \equiv s_0$, *there exists a state* s_i *along* π *such that* $(M,s_i) \models^{\star} \neg\alpha$,
24. $(M,s) \models^{\star} \neg \mathrm{AF}\alpha$ *iff there exists a path* $\pi \equiv s_0, s_1, s_2, \ldots$ *with* $s \equiv s_0$, *and for any states* s_i *along* π, *we have* $(M,s_i) \models^{\star} \neg\alpha$,
25. $(M,s) \models^{\star} \neg \mathrm{EF}\alpha$ *iff for any paths* $\pi \equiv s_0, s_1, s_2, \ldots$ *with* $s \equiv s_0$, *and any states* s_i *along* π, *we have* $(M,s_i) \models^{\star} \neg\alpha$,
26. $(M,s) \models^{\star} \neg \mathrm{A}(\alpha \mathrm{U} \beta)$ *iff there exists a path* $\pi \equiv s_0, s_1, s_2, \ldots$ *with* $s \equiv s_0$, *and for any states* s_j *along* π, *we have* $(M,s_j) \models^{\star} \neg\beta$ *or* $\exists 0 \leq k < j\ (M,s_k) \models^{\star} \neg\alpha$,
27. $(M,s) \models^{\star} \neg \mathrm{E}(\alpha \mathrm{U} \beta)$ *iff for any paths* $\pi \equiv s_0, s_1, s_2, \ldots$ *with* $s \equiv s_0$, *and any states* s_j *along* π, *we have* $(M,s_j) \models^{\star} \neg\beta$ *or* $\exists 0 \leq k < j\ (M,s_k) \models^{\star} \neg\alpha$,
28. $(M,s) \models^{\star} \neg \mathrm{A}(\alpha \mathrm{R} \beta)$ *iff there exists a path* $\pi \equiv s_0, s_1, s_2, \ldots$ *with* $s \equiv s_0$, *and for some state* s_j *along* π, *we have* $(M,s_j) \models^{\star} \neg\beta$ *and* $\forall 0 \leq k < j\ (M,s_k) \models^{\star} \neg\alpha$,
29. $(M,s) \models^{\star} \neg \mathrm{E}(\alpha \mathrm{R} \beta)$ *iff for any paths* $\pi \equiv s_0, s_1, s_2, \ldots$ *with* $s \equiv s_0$, *there exists a state* s_j *along* π *such that* $(M,s_j) \models^{\star} \neg\beta$ *and* $\forall 0 \leq k < j\ (M,s_k) \models^{\star} \neg\alpha$.

A formula α *is called* n-valid *in* CTL *if* $(M,s) \models^{\star} \alpha$ *holds for any falsification-aware normal CTL-model* $M := (S, S_0, R, L^{\star})$, *any* $s \in S$, *and any falsification-aware normal CTL-satisfaction relation* $\models^{\star}$ *on* M.

Theorem 3.2. *For any falsification-aware normal CTL-model* $M := (S, S_0, R, L^{\star})$, *any falsification-aware normal CTL-satisfaction relation* $\models^{\star}$ *on* M, *any formula* α, *and any* $s \in S$, *we have:* $(M,s) \models^{\star} \alpha$ *iff* $(M,s) \not\models^{\star} \neg\alpha$.

Proof. By induction on α. We show some cases.

1. Case $\alpha \equiv p \in \Phi$: We obtain: $(M,s) \models^{\star} p$ iff $p \in L^{\star}(s)$ iff $\neg p \notin L^{\star}(s)$ iff $(M,s) \not\models^{\star} \neg p$.
2. Case $\alpha \equiv \beta \rightarrow \gamma$: We obtain: $(M,s) \models^{\star} \beta \rightarrow \gamma$ iff $(M,s) \not\models^{\star} \beta$ or $(M,s) \models^{\star} \gamma$ iff $(M,s) \not\models^{\star} \beta$ or $(M,s) \not\models^{\star} \neg\gamma$ (by induction hypothesis) iff $(M,s) \not\models^{\star} \neg(\beta \rightarrow \gamma)$.
3. Case $\alpha \equiv \neg\beta$: We obtain: $(M,s) \models^{\star} \neg\beta$ iff $(M,s) \not\models^{\star} \beta$ (by induction hypothesis with contraposition) iff $(M,s) \not\models^{\star} \neg\neg\beta$.
4. Case $\alpha \equiv \mathrm{AX}\beta$: We obtain: $(M,s) \models^{\star} \mathrm{AX}\beta$ iff $\forall s_1 \in S\ [(s,s_1) \notin R$ or $(M,s_1) \models^{\star} \beta]$ iff $\forall s_1 \in S\ [(s,s_1) \notin R$ or $(M,s_1) \not\models^{\star} \neg\beta]$ (by induction hypothesis) iff not-$[\exists s_1 \in S\ [(s,s_1) \in R$ and $(M,s_1) \models^{\star} \neg\beta]]$ iff not-$[(M,s) \models^{\star} \neg \mathrm{AX}\beta]$ iff $(M,s) \not\models^{\star} \neg \mathrm{AX}\beta$.
5. Case $\alpha \equiv \mathrm{AG}\beta$: We obtain: $(M,s) \models^{\star} \mathrm{AG}\beta$ iff for any paths $\pi \equiv s_0, s_1, s_2, \ldots$ with $s \equiv s_0$, and any states s_i along π, we have $(M,s_i) \models^{\star} \beta$ iff for any paths $\pi \equiv s_0, s_1, s_2, \ldots$ with $s \equiv s_0$, and any states s_i along π, we have $(M,s_i) \not\models^{\star} \neg\beta$ (by induction hypothesis) iff not-[there exists a path $\pi \equiv s_0, s_1, s_2, \ldots$ with $s \equiv s_0$, and for some state s_i along π, we have $(M,s_i) \models^{\star} \neg\beta$] iff not-$[(M,s) \models^{\star} \neg \mathrm{AG}\beta]$ iff $(M,s) \not\models^{\star} \neg \mathrm{AG}\beta$.
6. Case $\alpha \equiv \mathrm{A}(\beta \mathrm{U} \gamma)$: We obtain: $(M,s) \models^{\star} \mathrm{A}(\beta \mathrm{U} \gamma)$ iff for any paths $\pi \equiv s_0, s_1, s_2, \ldots$ with $s \equiv s_0$, there exists a state s_j along π such that $(M,s_j) \models^{\star} \gamma$ and $\forall 0 \leq k < j\ (M,s_k) \models^{\star} \beta$ iff for any paths $\pi \equiv s_0, s_1, s_2, \ldots$ with $s \equiv s_0$, there exists a state s_j along π such that $(M,s_j) \models^{\star} \gamma$ and $\forall 0 \leq k < j\ (M,s_k) \not\models^{\star} \neg\beta$ iff not-[there exists a path $\pi \equiv s_0, s_1, s_2, \ldots$ with $s \equiv s_0$, and for any states s_j along π, we have $(M,s_j) \models^{\star} \neg\gamma$ or $\exists 0 \leq k < j\ (M,s_k) \models^{\star} \neg\beta]$ iff not-$[(M,s) \models^{\star} \neg \mathrm{A}(\beta \mathrm{U} \gamma)]$ iff $(M,s) \not\models^{\star} \neg \mathrm{A}(\beta \mathrm{U} \gamma)$.

□

Remark 3.3. *We make the following remarks.*

1. *Theorem 3.2 shows that the mapping condition "for any* $p \in \Phi$ *and any* $s \in S$, $p \in L^{\star}(s)$ *iff* $\neg p \notin L^{\star}(s)$*" in Definition 3.1 can be extended to the falsification-aware normal CTL-satisfaction relation for any formula* α.
2. *By the contraposition of the statement of Theorem 3.2, we can obtain the following standard Boolean negation condition for falsification-aware normal CTL-satisfaction relation:* $(M,s) \models^{\star} \neg\alpha$ *iff* $(M,s) \not\models^{\star} \alpha$.
3. *We use Theorem 3.2 (with the fact discussed just above) for proving the equivalence between the falsification-aware normal and normal semantics for* CTL. *Indeed, Theorem 3.2 will be used for proving the case* $\alpha \equiv \neg\beta$ *in Lemma 3.4.*

Prior to prove the equivalence between the falsification-aware normal and normal Kripke-style semantics for CTL (i.e., the equivalence between the n-validity and the validity), we need to show the following two lemmas.

Lemma 3.4. *For any CTL-model* $M := (S, S_0, R, L)$ *and any CTL-satisfaction relation* $\models$ *on* M, *we can*

construct a falsification-aware normal CTL-model $N := (S, S_0, R, L^\star)$ *and a falsification-aware normal CTL-satisfaction relation* $\models^\star$ *on* N *such that for any formula* α *and any* $s \in S$, $(M,s) \models \alpha$ *iff* $(N,s) \models^\star \alpha$.

Proof. Let M be a CTL-model (S, S_0, R, L) and $\models$ be a CTL-satisfaction relation on M. Then, we define a falsification-aware normal CTL-model $N := (S, S_0, R, L^\star)$ such that for any $s \in S$ and any $p \in \Phi$,

1. $p \in L^\star(s)$ iff $p \in L(s)$,
2. $\neg p \in L^\star(s)$ iff $p \notin L(s)$.

Then, we can obtain the mapping condition "$p \in L^\star(s)$ iff $\neg p \notin L^\star(s)$," because we have: $p \in L^\star(s)$ iff $p \in L(s)$ iff $\neg p \notin L^\star(s)$.

We now prove this lemma by induction on α. We show some cases.

1. Case $\alpha \equiv p \in \Phi$: We obtain: $(M,s) \models p$ iff $p \in L(s)$ iff $p \in L^\star(s)$ iff $(N,s) \models^\star p$.
2. Case $\alpha \equiv \beta \rightarrow \gamma$: We obtain: $(M,s) \models \beta \rightarrow \gamma$ iff $(M,s) \not\models \beta$ or $(M,s) \models \gamma$ iff $(N,s) \not\models^\star \beta$ or $(N,s) \models^\star \gamma$ (by induction hypothesis) iff $(N,s) \models^\star \beta \rightarrow \gamma$.
3. Case $\alpha \equiv \neg\beta$: We obtain: $(M,s) \models \neg\beta$ iff $(M,s) \not\models \beta$ iff $(N,s) \not\models^\star \beta$ (by induction hypothesis) iff $(N,s) \models^\star \neg\beta$ (by Theorem 3.2).
4. Case $\alpha \equiv \mathrm{AX}\beta$: We obtain: $(M,s) \models \mathrm{AX}\beta$ iff $\forall s_1 \in S\ [(s,s_1) \notin R$ or $(M,s_1) \models \beta]$ iff $\forall s_1 \in S\ [(s,s_1) \notin R$ or $(N,s_1) \models^\star \beta$ (by induction hypothesis) iff $(N,s) \models^\star \mathrm{AX}\beta$.
5. Case $\alpha \equiv \mathrm{AG}\beta$: We obtain: $(M,s) \models \mathrm{AG}\beta$ iff for any paths $\pi \equiv s_0, s_1, s_2, \ldots$ with $s \equiv s_0$, and any states s_i along π, we have $(M,s_i) \models \beta$ iff for any paths $\pi \equiv s_0, s_1, s_2, \ldots$ with $s \equiv s_0$, and any states s_i along π, we have $(N,s_i) \models^\star \beta$ (by induction hypothesis) iff $(N,s) \models^\star \mathrm{AG}\beta$.
6. Case $\alpha \equiv \mathrm{A}(\beta \mathrm{U} \gamma)$: We obtain: $(M,s) \models \mathrm{A}(\beta \mathrm{U} \gamma)$ iff for any paths $\pi \equiv s_0, s_1, s_2, \ldots$ with $s \equiv s_0$, there exists a state s_j along π such that $(M,s_j) \models \gamma$ and $\forall 0 \leq k < j\ (M,s_k) \models \beta$ iff for any paths $\pi \equiv s_0, s_1, s_2, \ldots$ with $s \equiv s_0$, there exists a state s_j along π such that $(N,s_j) \models^\star \gamma$ and $\forall 0 \leq k < j\ (N,s_k) \models^\star \beta$ (by induction hypothesis) iff $(N,s) \models^\star \mathrm{A}(\beta \mathrm{U} \gamma)$.

□

Lemma 3.5. *For any falsification-aware normal CTL-model* $M := (S, S_0, R, L^\star)$ *and any falsification-aware normal CTL-satisfaction relation* $\models^\star$ *on* M*, we can construct a CTL-model* $N := (S, S_0, R, L)$ *and a CTL-satisfaction relation* $\models$ *on* N *such that for any formula* α *and any* $s \in S$, $(M,s) \models^\star \alpha$ *iff* $(N,s) \models \alpha$.

Proof. Similar to the proof of Lemma 3.4. □

Theorem 3.6 (Equivalence between n-validity and validity in CTL)**.** *For any formula* α*, we have:* α *is n-valid in* CTL *iff* α *is valid in* CTL.

Proof. By Lemmas 3.4 and 3.5. □

4 FALSIFICATION-AWARE DUAL KRIPKE-STYLE SEMANTICS FOR CTL

A falsification-aware dual Kripke-style semantics for CTL is defined as follows.

Definition 4.1 (Falsification-aware dual Kripke-style semantics for CTL)**.** *Let* Φ *be a non-empty set of propositional variables and* $\Phi^\neg$ *be the set* $\{\neg p \mid p \in \Phi\}$ *of negated propositional variables.*

A structure (S, S_0, R, L^+, L^-) *is called a* falsification-aware dual CTL-model *if*

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 such that* $\forall s \in S\ \exists s' \in S\ [(s,s') \in R]$,
4. L^+ *and* L^- *are mappings from S to the power set of* Φ *such that for any* $p \in \Phi$ *and any* $s \in S$, $p \in L^+(s)$ *iff* $p \notin L^-(s)$.

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

Falsification-aware dual CTL-satisfaction relations $(M,s) \models^+ \alpha$ *and* $(M,s) \models^- \alpha$ *for any formula* α*, where M is a falsification-aware dual CTL-model* (S, S_0, R, L^+, L^-) *and s is a state in S, are defined inductively by the following clauses:*

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* $(M,s) \not\models^+ \alpha$ *or* $(M,s) \models^+ \beta$,
5. $(M,s) \models^+ \neg\alpha$ *iff* $(M,s) \models^- \alpha$,
6. $(M,s) \models^+ \mathrm{AX}\alpha$ *iff* $\forall s_1 \in S\ [(s,s_1) \notin R$ *or* $(M,s_1) \models^+ \alpha]$,
7. $(M,s) \models^+ \mathrm{EX}\alpha$ *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 any paths* $\pi \equiv s_0, s_1, s_2, \ldots$ *with* $s \equiv s_0$*, and any states* s_i *along* π*, we have* $(M,s_i) \models^+ \alpha$,
9. $(M,s) \models^+ \mathrm{EG}\alpha$ *iff there exists a path* $\pi \equiv s_0, s_1, s_2, \ldots$ *with* $s \equiv s_0$*, and for any states* s_i *along* π*, we have* $(M,s_i) \models^+ \alpha$,
10. $(M,s) \models^+ \mathrm{AF}\alpha$ *iff for any paths* $\pi \equiv s_0, s_1, s_2, \ldots$ *with* $s \equiv s_0$*, there exists a state* s_i *along* π *such that* $(M,s_i) \models^+ \alpha$,
11. $(M,s) \models^+ \mathrm{EF}\alpha$ *iff there exists a path* $\pi \equiv s_0, s_1, s_2, \ldots$ *with* $s \equiv s_0$*, and for some state* s_i *along* π*, we have* $(M,s_i) \models^+ \alpha$,

12. $(M,s) \models^+ A(\alpha U \beta)$ *iff for any paths* $\pi \equiv s_0,s_1,s_2,...$ *with* $s \equiv s_0$*, there exists 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. $(M,s) \models^+ E(\alpha U \beta)$ *iff there exists a path* $\pi \equiv s_0,s_1,s_2,...$ *with* $s \equiv s_0$*, 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. $(M,s) \models^+ A(\alpha R \beta)$ *iff for any paths* $\pi \equiv s_0,s_1,s_2,...$ *with* $s \equiv s_0$*, and any states* s_j *along* π*, we have* $(M,s_j) \models^+ \beta$ *or* $\exists 0 \leq k < j\ (M,s_k) \models^+ \alpha$,
15. $(M,s) \models^+ E(\alpha R \beta)$ *iff there exists a path* $\pi \equiv s_0,s_1,s_2,...$ *with* $s \equiv s_0$*, and for any states* s_j *along* π*, we have* $(M,s_j) \models^+ \beta$ *or* $\exists 0 \leq k < j\ (M,s_k) \models^+ \alpha$,
16. *for any* $p \in \Phi$, $(M,s) \models^- p$ *iff* $p \in L^-(s)$,
17. $(M,s) \models^- \alpha \wedge \beta$ *iff* $(M,s) \models^- \alpha$ *or* $(M,s) \models^- \beta$,
18. $(M,s) \models^- \alpha \vee \beta$ *iff* $(M,s) \models^- \alpha$ *and* $(M,s) \models^- \beta$,
19. $(M,s) \models^- \alpha \rightarrow \beta$ *iff* $(M,s) \models^+ \alpha$ *and* $(M,s) \models^- \beta$,
20. $(M,s) \models^- \neg\alpha$ *iff* $(M,s) \models^+ \alpha$,
21. $(M,s) \models^- AX\alpha$ *iff* $\exists s_1 \in S\ [(s,s_1) \in R$ *and* $(M,s_1) \models^- \alpha]$,
22. $(M,s) \models^- EX\alpha$ *iff* $\forall s_1 \in S\ [(s,s_1) \notin R$ *or* $(M,s_1) \models^- \alpha]$,
23. $(M,s) \models^- AG\alpha$ *iff there exists a path* $\pi \equiv s_0,s_1,s_2,...$ *with* $s \equiv s_0$*, and for some state* s_i *along* π*, we have* $(M,s_i) \models^- \alpha$,
24. $(M,s) \models^- EG\alpha$ *iff for any paths* $\pi \equiv s_0,s_1,s_2,...$ *with* $s \equiv s_0$*, there exists a state* s_i *along* π *such that* $(M,s_i) \models^- \alpha$,
25. $(M,s) \models^- AF\alpha$ *iff there exists a path* $\pi \equiv s_0,s_1,s_2,...$ *with* $s \equiv s_0$*, and for any states* s_i *along* π*, we have* $(M,s_i) \models^- \alpha$,
26. $(M,s) \models^- EF\alpha$ *iff for any paths* $\pi \equiv s_0,s_1,s_2,...$ *with* $s \equiv s_0$*, and any states* s_i *along* π*, we have* $(M,s_i) \models^- \alpha$,
27. $(M,s) \models^- A(\alpha U \beta)$ *iff there exists a path* $\pi \equiv s_0,s_1,s_2,...$ *with* $s \equiv s_0$*, and for any states* s_j *along* π*, we have* $(M,s_j) \models^- \beta$ *or* $\exists 0 \leq k < j\ (M,s_k) \models^- \alpha$,
28. $(M,s) \models^- E(\alpha U \beta)$ *iff for any paths* $\pi \equiv s_0,s_1,s_2,...$ *with* $s \equiv s_0$*, and any states* s_j *along* π*, we have* $(M,s_j) \models^- \beta$ *or* $\exists 0 \leq k < j\ (M,s_k) \models^- \alpha$,
29. $(M,s) \models^- A(\alpha R \beta)$ *iff there exists a path* $\pi \equiv s_0,s_1,s_2,...$ *with* $s \equiv s_0$*, and for some state* s_j *along* π*, we have* $(M,s_j) \models^- \beta$ *and* $\forall 0 \leq k < j\ (M,s_k) \models^- \alpha$,
30. $(M,s) \models^- E(\alpha R \beta)$ *iff for any paths* $\pi \equiv s_0,s_1,s_2,...$ *with* $s \equiv s_0$*, there exists a state* s_j *along* π *such that* $(M,s_j) \models^- \beta$ *and* $\forall 0 \leq k < j\ (M,s_k) \models^- \alpha$.

A formula α *is called* d-valid *in* CTL *if* $(M,s) \models^+ \alpha$ *holds for any falsification-aware dual CTL-model* $M := (S,S_0,R,L^+,L^-)$*, any* $s \in S$*, and any falsification-aware dual CTL-satisfaction relations* $\models^+$ *and* $\models^-$ *on* M.

Lemma 4.2. *For any CTL-model* $M := (S,S_0,R,L)$ *and any CTL-satisfaction relation* $\models$ *on* M*, we can construct a falsification-aware dual CTL-model* $N := (S,S_0,R,L^+,L^-)$ *and falsification-aware dual CTL-satisfaction relations* $\models^+$ *and* $\models^-$ *on* N *such that for any formula* α *and any* $s \in S$*,*

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

Proof. Let M be a CTL-model (S,S_0,R,L) and $\models$ be an CTL-satisfaction relation on M. Then, we define a falsification-aware dual CTL-model $N := (S,S_0,R,L^+,L^-)$ such that for any $s \in S$ and any $p \in \Phi$,

1. $p \in L^+(s)$ iff $p \in L(s)$,
2. $p \in L^-(s)$ iff $p \notin L(s)$.

Then, we can obtain the mapping condition "$p \in L^+(s)$ iff $p \notin L^-(s)$."

We now prove this lemma by simultaneous induction on α. We show some cases.

1. Case $\alpha \equiv p \in \Phi$: For 1, we obtain: $(M,s) \models p$ iff $p \in L(s)$ iff $p \in L^+(s)$ iff $(N,s) \models^+ p$. For 2, we obtain: $(M,s) \models \neg p$ iff $(M,s) \not\models p$ iff $p \notin L(s)$ iff $p \in L^-(s)$ iff $(N,s) \models^- p$.
2. Case $\alpha \equiv \beta \rightarrow \gamma$: For 1, we obtain: $(M,s) \models \beta \rightarrow \gamma$ iff $(M,s) \not\models \beta$ or $(M,s) \models \gamma$ iff $(N,s) \not\models^+ \beta$ or $(N,s) \models^+ \gamma$ (by induction hypothesis for 1) iff $(N,s) \models^+ \beta \rightarrow \gamma$. For 2, we obtain: $(M,s) \models \neg(\beta \rightarrow \gamma)$ iff $(M,s) \not\models \beta \rightarrow \gamma$ iff $(M,s) \models \beta$ and $(M,s) \not\models \gamma$ iff $(M,s) \models \beta$ and $(M,s) \models \neg\gamma$ iff $(N,s) \models^+ \beta$ and $(N,s) \models^- \gamma$ (by induction hypotheses for 1 and 2) iff $(N,s) \models^- \beta \rightarrow \gamma$.
3. Case $\alpha \equiv \neg\beta$: For 1, we obtain: $(M,s) \models \neg\beta$ iff $(N,s) \models^- \beta$ (by induction hypothesis for 2) iff $(N,s) \models^+ \neg\beta$. For 2, we obtain: $(M,s) \models \neg\neg\beta$ iff $(M,s) \models \beta$ iff $(N,s) \models^+ \beta$ (by induction hypothesis for 1) iff $(N,s) \models^- \neg\beta$.
4. Case $\alpha \equiv AX\beta$: For 1, we obtain: $(M,s) \models AX\beta$ iff $\forall s_1 \in S\ [(s,s_1) \notin R$ or $(M,s_1) \models \beta]$ iff $\forall s_1 \in S\ [(s,s_1) \notin R$ or $(N,s_1) \models^+ \beta$ (by induction hypothesis for 1) iff $(N,s) \models^+ AX\beta$. For 2, we obtain: $(M,s) \models \neg AX\beta$ iff $(M,s) \not\models AX\beta$ iff not-$[(M,s) \models AX\beta]$ iff not-$[\forall s_1 \in S\ [(s,s_1) \notin R$ or $(M,s_1) \models \beta],]$ iff $\exists s_1 \in S\ [(s,s_1) \in R$ and $(M,s_1) \not\models \beta]$ iff $\exists s_1 \in S\ [(s,s_1) \in R$ and $(M,s_1) \models \neg\beta]$ iff $\exists s_1 \in S\ [(s,s_1) \in R$ and $(N,s_1) \models^- \beta]$ (by induction hypothesis for 2) iff $(N,s) \models^- AX\beta$.
5. Case $\alpha \equiv AG\beta$: For 1, we obtain: $(M,s) \models AG\beta$ iff for any paths $\pi \equiv s_0,s_1,s_2,...$ with $s \equiv s_0$, and any states s_i along π, we have $(M,s_i) \models \beta$ iff for any paths $\pi \equiv s_0,s_1,s_2,...$ with $s \equiv s_0$, and any states s_i along π, we have $(N,s_i) \models^+ \beta$ (by induction hypothesis for 1) iff $(N,s) \models^+ AG\beta$. For 2, we obtain: $(M,s) \models \neg AG\beta$ iff $(M,s) \not\models AG\beta$ iff there exists a path $\pi \equiv s_0,s_1,s_2,...$ with $s \equiv s_0$, and for some state s_i along π, we have $(M,s_i) \not\models \beta$ iff there exists a path $\pi \equiv s_0,s_1,s_2,...$ with $s \equiv s_0$, and for some state s_i along π, we have $(M,s_i) \models \neg\beta$ iff

there exists a path $\pi \equiv s_0, s_1, s_2, ...$ with $s \equiv s_0$, and for some state s_i along π, we have $(N, s_i) \models^- \beta$ (by induction hypothesis for 2) iff $(N, s) \models^- AG\beta$.

6. Case $\alpha \equiv A(\beta U \gamma)$: For 1, we obtain: $(M, s) \models A(\beta U \gamma)$ iff for any paths $\pi \equiv s_0, s_1, s_2, ...$ with $s \equiv s_0$, there exists a state s_j along π such that $(M, s_j) \models \gamma$ and $\forall 0 \leq k < j$ $(M, s_k) \models \beta$ iff for any paths $\pi \equiv s_0, s_1, s_2, ...$ with $s \equiv s_0$, there exists a state s_j along π such that $(N, s_j) \models^+ \gamma$ and $\forall 0 \leq k < j$ $(N, s_k) \models^+ \beta$ (by induction hypothesis) iff $(N, s) \models^+ A(\beta U \gamma)$. For 2, we obtain: $(M, s) \models \neg A(\beta U \gamma)$ iff $(M, s) \not\models A(\beta U \gamma)$ iff there exists a path $\pi \equiv s_0, s_1, s_2, ...$ with $s \equiv s_0$, and for any states s_j along π, we have $(M, s_j) \not\models \gamma$ or $\exists 0 \leq k < j$ $(M, s_k) \not\models \beta$ iff there exists a path $\pi \equiv s_0, s_1, s_2, ...$ with $s \equiv s_0$, and for any states s_j along π, we have $(M, s_j) \models \neg\gamma$ or $\exists 0 \leq k < j$ $(M, s_k) \models \neg\beta$ iff there exists a path $\pi \equiv s_0, s_1, s_2, ...$ with $s \equiv s_0$, and for any states s_j along π, we have $(N, s_j) \models^- \gamma$ or $\exists 0 \leq k < j$ $(N, s_k) \models^- \beta$ (by induction hypothesis for 2) iff $(N, s) \models^- A(\beta U \gamma)$.

□

Lemma 4.3. *For any falsification-aware dual CTL-model $M := (S, S_0, R, L^+, L^-)$ and any falsification-aware dual CTL-satisfaction relations $\models^+$ and $\models^-$ on M, we can construct a CTL-model $N := (S, S_0, R, L)$ and a CTL-satisfaction relation $\models$ on N such that for any formula α and any $s \in S$,*

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

Proof. Similar to the proof of Lemma 4.2. □

Theorem 4.4 (Equivalence between d-validity and validity in CTL)**.** *For any formula α, we have: α is d-valid in* CTL *iff α is valid in* CTL.

Proof. By Lemmas 4.2 and 4.3. □

Remark 4.5. *We can obtain the following fact by Theorems 3.6 and 4.4. The following conditions are equivalent: For any formula α,*

1. *α is valid in* CTL,
2. *α is n-valid in* CTL,
3. *α is d-valid in* CTL.

5 FALSIFICATION-AWARE KRIPKE-STYLE SEMANTICS FOR ICTL

We now semantically define ICTL as a subsystem of CTL. *Formulas* of ICTL are defined in the same way as those for CTL.

Definition 5.1 (Falsification-aware normal Kripke-style semantics for ICTL)**.** *The falsification-aware normal Kripke-style semantics for* ICTL *is obtained from that for* CTL *by deleting the mapping condition "for any $p \in \Phi$ and any $s \in S$, $p \in L^\star(s)$ iff $\neg p \notin L^\star(s)$" in Definition 3.1. The notions and notations concerning this semantics for* ICTL *can be defined in a similar way as those for* CTL.

Remark 5.2. *We make the following remarks.*

1. *We cannot semantically define* ICTL *using the standard non-falsification-aware Kripke-style semantics as presented in Definition 2.2.*
2. ICTL *is regarded as or closely related to the classical negation-free subsystems of* PCTL (Kamide and Kaneiwa, 2010; Kaneiwa and Kamide, 2011) *and* pCTL (Kamide and Endo, 2018; Kamide and Endo, 2019). *Namely,* ICTL *is regarded as the purely inconsistency-tolerant subsystem of* PCTL *and* pCTL.
3. *As will be discussed in the following item 5,* ICTL *is a proper subsystem of* CTL, *because the formula of the form $(p \wedge \neg p) \rightarrow q$, where p and q are distinct propositional variables, is n-valid in* CTL, *but not n-valid in* ICTL.
4. *The same theorem as Theorem 3.2 does not hold for* ICTL *(i.e., the negation connective $\neg$ in* ICTL *is not the classical negation connective).*
5. *In general, a satisfaction relation $\models$ for a logic is called* paraconsistent *with respect to a negation connective $\neg$ if the condition "$\exists \alpha, \beta$ $(M, s) \not\models (\alpha \wedge \neg\alpha) \rightarrow \beta$" holds. This condition reflects that $(\alpha \wedge \neg\alpha) \rightarrow \beta$ is not valid in the underlying logic. Based on this definition of paraconsistency,* ICTL *is shown to be paraconsistent with respect to $\neg$, although* CTL *is not paraconsistent with respect to $\neg$. The paraconsistency of* ICTL *with respect to $\neg$ can be shown as follows. Assume a falsification-aware normal ICTL-model $M = (S, S_0, R, L^\star)$ such that $p \in L^\star(s)$, $\neg p \in L^\star(s)$ and $q \notin L^\star(s)$ for a pair of distinct propositional variables p and q. Then, we have $(M, s) \not\models^\star (p \wedge \neg p) \rightarrow q$.*
6. ICTL *is regarded as a four-valued logic. This fact is shown 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., $(M, s) \models^\star \alpha$,*
   (b) *α is falsified at s, i.e., $(M, s) \models^\star \neg\alpha$,*
   (c) *α is both verified and falsified at s,*
   (d) *α is neither verified nor falsified at s.*

7. *We can show the fact that* ICTL *is embeddable into the positive (i.e., ¬-less) fragment of* CTL. *This fact can be proved in the same way as presented in* (Kamide and Endo, 2018; Kamide and Endo, 2019). *Thus, the paraconsistent negation connective in* ICTL *can be handled in the positive fragment of* CTL.

Next, we introduce a falsification-aware dual Kripke-style semantics for ICTL.

Definition 5.3 (Falsification-aware dual Kripke-style semantics for ICTL)**.** *By deleting the mapping condition "for any $p \in \Phi$ and any $s \in S$, $p \in L^{+}(s)$ iff $p \notin L^{-}(s)$" in Definition 4.1, we can obtain a falsification-aware dual Kripke-style semantics for* ICTL.

Then, we can obtain the following theorem.

Theorem 5.4 (Equivalence between n-validity and d–validity in ICTL)**.** *For any formula α, we have: α is n-valid in* ICTL *iff α is d-valid in* ICTL.

Proof. We can prove this theorem in a similar way as the proof of Theorem 3.6. We have to prove the lemmas that are similar to Lemmas 4.2 and 4.3. But, to prove these lemmas, we do not need to use Theorem 3.2 because ICTL has no classical negation. □

6 CONCLUDING REMARKS

In this paper, we first introduced new falsification-aware normal and dual Kripke-style semantics for CTL. We then proved the equivalences among the proposed falsification-aware Kripke-style semantics and the standard Kripke-style semantics for CTL. By using the proposed falsification-aware Kripke-style semantics, we semantically defined the new inconsistency-tolerant and many-valued temporal logic ICTL, which is obtained from the proposed falsification-aware Kripke-style semantics for CTL by deleting only one characteristic condition on the labeling function of the semantics. Thus, the proposed falsification-aware semantic framework for CTL and ICTL is considered to be a unified framework for combining and generalizing the standard, inconsistency-tolerant, and many-valued semantic frameworks. The proposed framework can be used for falsification-aware model checking, which is roughly defined as that of combined and generalized model checking based on falsification-aware Kripke-style semantics. The notion of falsification-aware model checking and an illustrative example for falsification-aware model checking will be explained and addressed in the last part of this section.

We remark that the proposed falsification-aware semantic framework can be extended to other temporal logics. For example, we can obtain the falsification-aware normal and dual Kripke-style semantics for *linear-time temporal logic* (LTL) (Pnueli, 1977) and its inconsistency-tolerant subsystem ILTL in a similar manner as those for CTL and ICTL. Then, we can prove the equivalences among the falsification-aware Kripke-style semantics and the standard Kripke-style semantics for LTL as well as those between the falsification-aware normal and dual Kripke-style semantics for ILTL. We can also obtain the falsification-aware normal and dual Kripke-style semantics for *full computation tree logic*, referred to as CTL* (Emerson and Halpern, 1986; Emerson and Sistla, 1984), and its inconsistency-tolerant subsystem ICTL*. We can also prove the equivalences among the falsification-aware Kripke-style semantics and the standard Kripke-style semantics for CTL* as well as those between the falsification-aware normal and dual Kripke-style semantics for ICTL*. Thus, we can obtain a falsification-aware model checking framework based on these standard and inconsistency-tolerant temporal logics.

Furthermore, we can also obtain the falsification-aware normal and dual Kripke-style semantics for probabilistic CTL, referred to as pCTL, probabilistic CTL*, referred to as pCTL* (Aziz et al., 1995; Bianco and de Alfaro, 1995), and their inconsistency-tolerant subsystems IpCTL and IpCTL*. We can also prove the equivalences among the falsification-aware Kripke-style semantics and the standard Kripke-style semantics for pCTL and pCTL*, respectively, as well as those between the falsification-aware normal and dual Kripke-style semantics for IpCTL and IpCTL*, respectively. In addition to these results, we can also obtain the falsification-aware normal and dual Kripke-style semantics for hierarchical (or sequential) CTL, referred to as sCTL, hierarchical (or sequential) LTL, referred to as sLTL (Kamide and Yano, 2017; Kamide, 2018), and their inconsistency-tolerant subsystems IsCTL and IsLTL, respectively. Then, we can prove the equivalences among the falsification-aware Kripke-style semantics and the standard Kripke-style semantics for sCTL and sLTL, respectively, as well as those between the falsification-aware normal and dual Kripke-style semantics for IsCTL and IsLTL, respectively. Thus, we can also obtain a falsification-aware model checking framework based on these extended temporal logics.

Finally, we illustrate a small verification example based on falsification-aware model checking. We consider the disease diagnosis model shown in Figure 1 for *familial adenomatous polyposis* (FAP) (Wehbi,

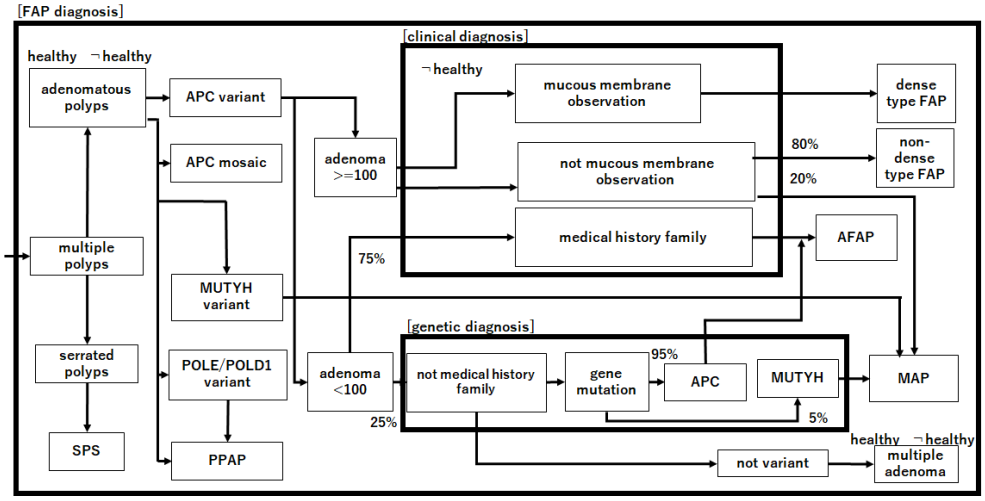

Figure 1: A diagnosis model for familial adenomatous polyposis.

2019). FAP is an inherited disorder characterized by an autosomal dominant inherited condition in which numerous adenomatous polyps form mainly in the epithelium of the large intestine. Although these polyps start out benign, malignant transformation into colon cancer occurs when they are left untreated. Three variants are known to exist: FAP, attenuated FAP (AFAP), and MUTYH-associated polyposis (MAP). Furthermore, the first variant of FAP is classified as dense- or non-dense-type FAP. For these variants, the resulting colonic polyps and cancers were initially confined to the colon wall. Detection and removal before metastasis outside the colon can greatly reduce and eliminate the spread of cancer in many cases.

For the model presented in Figure 1, we can verify the following statements:

1. "Is there a state in which a person is both healthy and unhealthy (i.e., not yet ill), has less than 100 adenomas, and had a medical examination with genetic diagnosis?"
2. "Is there a state in which a person has a family medical history, more than 100 adenomas, had a medical examination with clinical diagnosis, and has MAP?"

Although the first statement is true because there is a case in which the person has multiple benign adenomas, the second statement is false because this case does not imply MAP but implies AFAP. These statements are formally expressed as follows:

1. $\mathrm{EF}(healthy \wedge \neg healthy \wedge (adenoma < 100) \wedge geneticDiagnosis)$
2. $\mathrm{EF}(hasMedicalHistoryFamily \wedge (adenoma > 100) \wedge clinicalDiagnosis \wedge MAP)$

where the negation connective $\neg$ in the first formula is regarded as the inconsistency-tolerant negation connective in ICTL. In this example, we can simultaneously verify and falsify these formulas using the falsification-aware dual CTL- or ICTL-satisfaction relations $\models^+$ and $\models^-$ in the falsification-aware dual Kripke-style semantics. Furthermore, we can formally consider a falsification-aware model checking problem as follows. Suppose that M is a falsification-aware dual CTL- or ICTL-model (S, S_0, R, L^+, L^-) and that $\models^+$ and $\models^-$ are falsification-aware dual CTL- or ICTL-satisfaction relations on M. Then, the *falsification-aware model checking problem* for CTL or ICTL is defined as follows. For any formula α, find the *verification set* $\{s \in S \mid M, s \models^+ \alpha\}$ and *falsification set* $\{s \in S \mid M, s \models^- \alpha\}$. These sets can be simultaneously found. Finally, we remark that extended falsification-aware pCTL-, IpCTL-, sCTL-, and IsCTL-model checking frameworks, which are based on the aforementioned extended CTLs, are more suitable for the verification of the model presented in Figure 1 because probabilistic and hierarchical specifications are also required to verify this model.

ACKNOWLEDGEMENTS

This research was supported by JSPS KAKENHI Grant Numbers JP18K11171 and JP16KK0007.

REFERENCES

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

Aziz, A., Singhal, V., and Balarin, F. (1995). It usually works: The temporal logic of stochastic systems. In *Proceedings of the 7th Int. Conf. on Computer Aided Verification (CAV 1995), Lecture Notes in Computer Science 939*, pages 155–165.

Bianco, A. and de Alfaro, L. (1995). Model checking of probabilistic and nondeterministic systems. In *Proceedings of the 15th Conf. on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 1995), Lecture Notes in Computer Science 1026*, pages 499–513.

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., Jha, S., Lu, Y., and Veith, H. (2003). Counterexample-guided abstraction refinement for symbolic model checking. *Journal of the ACM*, 50 (5):752–794.

Clarke, E., Henzinger, T., Veith, H., and Bloem, R. (2018). *Handbook of Model Checking*. Springer.

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 (ICSE 2001)*, pages 411–420.

Emerson, E. and Halpern, J. (1986). 'sometimes' and 'not never' revisited: on branching versus linear time temporal logic. *Journal of the ACM*, 33 (1):151–178.

Emerson, E. and Sistla, A. (1984). Deciding full branching time logic. *Information and Control*, 61:175–201.

Gurfinkel, A., Wei, O., and Chechik, M. (2006). Yasm: A software model-checker for verification and refutation. In *Proceedings of the 18th International Conference on Computer Aided Verification (CAV 2006), Lecture Notes in Computer Science*, volume 4144, pages 170–174.

Horn, L. and Wansing, H. (2017). Negation. *The Stanford Encyclopedia of Philosophy (Spring 2017 Edition), Edward N. Zalta (editor), Last modified on January 2017*.

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. (2018). Logical foundations of hierarchical model checking. *Data Technologies and Applications*, 52 (4):539–563.

Kamide, N. (2021). Falsification-aware semantics and sequent calculi for classical logic. *Journal of Philosophical Logic*, Online first article.

Kamide, N. and Bernal J.P.A. (2019). Towards locative inconsistency-tolerant hierarchical probabilistic ctl model checking: Survey and future work. In *Proceedings of the 11th International Conference on Agents and Artificial Intelligence (ICAART 2019)*, volume 2, pages 869–878.

Kamide, N. and Endo, K. (2018). Logics and translations for inconsistency-tolerant model checking. In *Proceedings of the 10th International Conference on Agents and Artificial Intelligence (ICAART 2018)*, volume 2, pages 191–200.

Kamide, N. and Endo, K. (2019). Foundations of inconsistency-tolerant model checking: Logics, translations, and examples. In *Agents and Artificial Intelligence, the 10th International Conference ICAART 2018 Revised Selected Papers, Lecture Notes in Artificial Intelligence*, volume 11352, pages 1–31.

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)*, volume 1, pages 464–469.

Kamide, N. and Koizumi, D. (2015). Combining paraconsistency and probability in ctl. *Proceedings of the 7th International Conference on Agents and Artificial Intelligence (ICAART 2015)*, 2:285–293.

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

Kamide, N. and Yamamoto, Y. (2021). Inconsistency-tolerant hierarchical probabilistic computation tree logic and its application to model checking. *Proceedings of the 13th International Conference on Agents and Artificial Intelligence (ICAART 2021)*, 2:490–499.

Kamide, N. and Yano, R. (2017). Logics and translations for hierarchical model checking. *Proceedings of the 21st International Conference on Knowledge-Based and Intelligent Information and Engineering Systems (KES2017), Procedia Computer Science*, 112:31–40.

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.

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. and Tanaka, K. (2009). Paraconsistent logic. *Web site of the Stanford encyclopedia of philosophy (2009 Edition), Edward N. Zalta (editor)*.

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

Wehbi, M. (2019). Familial adenomatous polyposis. *Web site of eMedicine Gastroenterology*.