Inconsistency-tolerant Hierarchical Probabilistic Computation Tree Logic and Its Application to Model Checking

Norihiro Kamide[1] and Noriko Yamamoto[2]
[1]*Teikyo University, Faculty of Science and Engineering, Department of Information and Electronic Engineering, Toyosatodai 1-1, Utsunomiya-shi, Tochigi 320-8551, Japan*
[2]*Teikyo University, Graduate School of Science and Engineering, Division of Information Science, Toyosatodai 1-1, Utsunomiya, Tochigi 320-8551, Japan*

Keywords: Probabilistic Temporal Logic, Inconsistency-tolerant Temporal Logic, Hierarchical Temporal Logic, Probabilistic Model Checking, Inconsistency-tolerant Model Checking, Hierarchical Model Checking.

Abstract: An inconsistency-tolerant hierarchical probabilistic computation tree logic (IHpCTL) is developed to establish a new extended model checking paradigm referred to as IHpCTL model checking, which is intended to verify randomized, open, large, and complex concurrent systems. The proposed IHpCTL is constructed based on several previously established extensions of the standard probabilistic temporal logic known as probabilistic computation tree logic (pCTL), which is widely used for probabilistic model checking. IHpCTL is shown to be embeddable into pCTL and is relatively decidable with respect to pCTL. This means that the decidability of pCTL with certain probability measures implies the decidability of IHpCTL. The results indicate that we can effectively reuse the previously proposed pCTL model-checking algorithms for IHpCTL model checking.

1 INTRODUCTION

Model Checking: is a computer-assisted method used to verify concurrent systems that can be modeled by state-transition systems (Clarke and Emerson, 1981; Clarke et al., 1999; Holzmann, 2006; Clarke et al., 2018). The aim of this study is to develop a new temporal logic that can establish a logical foundation of extended model checking to verify randomized, open, large, and complex concurrent systems. To develop this type of temporal logic, combining and integrating probabilistic, inconsistency-tolerant, and hierarchical reasoning mechanisms into a single logic are required. The reasons for these requirements are as follows: (1) verifying randomized concurrent systems (e.g., fault-tolerant communication systems over unreliable channels) requires the handling of probabilistic reasoning (Bianco and de Alfaro, 1995), (2) verifying open and large concurrent systems (e.g., web and cloud application systems) requires the handling of inconsistency-tolerant reasoning (Chen and Wu, 2006), and (3) verifying complex concurrent systems (e.g., web systems with wide tree structures) requires the handling of hierarchical reasoning (Kaneiwa and Kamide, 2011a).

To develop this type of logic , we combine and integrate the following useful non-classical logics: *temporal logics*, *probabilistic (or probability) logics*, *inconsistency-tolerant (or paraconsistent) logics*, and *hierarchical (or sequential) logics*. By combining and integrating these non-classical logics, we can extend and refine the previously established model-checking frameworks (Clarke and Emerson, 1981; Clarke et al., 1999; Holzmann, 2006; Clarke et al., 2018), which are well-known as formal and automated techniques for verifying concurrent systems. Model checking has been extended to *probabilistic model checking* (Aziz et al., 1995; Bianco and de Alfaro, 1995; Baier and Kwiatkowska, 1998; Kwiatkowska et al., 2011; Baier et al., 2018), *inconsistency-tolerant model checking* (Easterbrook and Chechik, 2001; Chen and Wu, 2006; Kaneiwa and Kamide, 2011b; Kamide and Endo, 2018), and *hierarchical model checking* (Kamide and Kaneiwa, 2009; Kaneiwa and Kamide, 2011a; Kamide, 2015; Kamide and Yano, 2017; Kamide, 2018). Therefore, by developing this combined and integrated logic, we can combine and integrate these extended model-checking frameworks.

In this study, we develop a new combined and integrated computation tree logic called *inconsistency-tolerant hierarchical probabilistic computation tree logic* (IHpCTL). This IHpCTL is developed to establish a new extended model checking paradigm referred to as IHpCTL model checking, which is in-

Kamide, N. and Yamamoto, N.
Inconsistency-tolerant Hierarchical Probabilistic Computation Tree Logic and Its Application to Model Checking.
DOI: 10.5220/0010181604900499
In *Proceedings of the 13th International Conference on Agents and Artificial Intelligence (ICAART 2021) - Volume 2*, pages 490-499
ISBN: 978-989-758-484-8

tended to verify randomized, open, large, and complex concurrent systems, including clinical reasoning systems. We construct IHpCTL by combining and integrating several previously established extensions of the standard probabilistic temporal logic known as *probabilistic computation tree logic* (pCTL) (Aziz et al., 1995; Bianco and de Alfaro, 1995), which is widely used for probabilistic model checking. As a main contribution of this study, IHpCTL is shown to be embeddable into pCTL and is relatively decidable with respect to pCTL. This means that the decidability of pCTL with certain probability measures implies the decidability of IHpCTL. These results indicate that we can effectively reuse the previously proposed pCTL model-checking algorithms (Aziz et al., 1995; Bianco and de Alfaro, 1995) for IHpCTL model checking.

We next explain pCTL and its probabilistic model-checking framework. pCTL is an extension of the standard temporal logic known as *computation tree logic* (CTL) (Clarke and Emerson, 1981) for model checking. It is obtained from CTL by adding the *probabilistic* or *probability operator* $P_{\geq x}$. The formulas in the form of $P_{\geq x}\alpha$ are intended to be read as "the probability of α holding in the future evolution of the system is at least x." pCTL was previously investigated by Aziz et al. (Aziz et al., 1995) and Bianco and de Alfaro (Bianco and de Alfaro, 1995). In (Bianco and de Alfaro, 1995), pCTL was introduced to verify the reliability properties and performances of the systems modeled by *discrete Markov chains*. In (Bianco and de Alfaro, 1995), the complexities of model-checking algorithms with respect to this logic were clarified. In (Aziz et al., 1995), model-checking algorithms for various extensions of the previous settings of pCTL were proposed to verify probabilistic nondeterministic concurrent systems. These algorithms were shown to exhibit polynomial-time complexity depending on the different sizes of the systems. The main difference between the approaches of Aziz et al. (Aziz et al., 1995) and Bianco and de Alfaro (Bianco and de Alfaro, 1995) is the settings of the *probability measures* in the *probabilistic Kripke models* of pCTL.

Although, as previously mentioned, pCTL and its probabilistic model-checking framework are useful, they are insufficient for handling open, large, and complex concurrent systems such as very large and complex cloud-based systems. Verifying these systems requires the handling of inconsistency-tolerant reasoning. This is because in open and large concurrent systems, inconsistencies are inevitable and appear often (Chen and Wu, 2006). Verifying these systems also requires the handling of hierarchical reasoning, as complex concurrent systems are constructed based on certain hierarchies (Kaneiwa and Kamide, 2011a). In addition, verifying clinical reasoning systems with complex disease ontologies, for example, requires the handling of both inconsistency-tolerant and hierarchical reasoning, as these types of systems consist of both open data related to vague concepts of symptoms and complex hierarchical structures of disease ontologies (Kamide and Bernal J.P.A., 2019). Thus, an extended logic with an extended model-checking framework is needed that can also simultaneously handle inconsistency-tolerant, hierarchical, and probabilistic reasoning.

For this direction, a few partial solutions were obtained in some previous studies (Kamide and Koizumi, 2015; Kamide and Koizumi, 2016; Kamide and Yano, 2019; Kamide and Bernal J.P.A., 2019). An *inconsistency-tolerant (or paraconsistent) probabilistic computation tree logic* (PpCTL), which was obtained from pCTL by adding the *paraconsistent negation connective* $\sim$, was developed in (Kamide and Koizumi, 2015; Kamide and Koizumi, 2016) based on a probability-measure-independent translation of PpCTL to pCTL. A theorem for embedding PpCTL into pCTL was proved using this translation and entailed the relative decidability of PpCTL with respect to pCTL. A *hierarchical probabilistic computation tree logic* (HpCTL), which was obtained from pCTL by adding the *hierarchical (or sequence) modal operator* $[b]$, was developed in (Kamide and Yano, 2019) based on a probability-measure-independent translation of HpCTL to pCTL. The same theorems as those for PpCTL were obtained for HpCTL. A *locative inconsistency-tolerant hierarchical probabilistic computation tree logic* (LIHpCTL), which is regarded as an extension of both PpCTL and HpCTL with the addition of the *location operator* $[l_i]$ introduced in (N. Kobayashi and Yonezawa, 1999), was considered in (Kamide and Bernal J.P.A., 2019).

However, the embedding and relative decidability theorems for LIHpCTL proposed in (Kamide and Bernal J.P.A., 2019) have not yet been proved, as some technical difficulties remain. Thus, the objective of this study is to make progress in this direction. The current study proves the embedding and relative decidability theorems for the proposed logic IHpCTL, which is considered to be a modified version of the location-operator-free subsystem of LIHpCTL. To prove these theorems, we need to overcome some technical difficulties in formalizing and defining a satisfaction relation and proving some key lemmas for the embedding theorem. For example, some previously proposed extended CTLs with the hierarchical modal operator $[b]$ (see, for example, (Kamide and Kaneiwa, 2009; Kaneiwa and Kamide, 2011a;

Kaneiwa and Kamide, 2010; Kamide, 2015)) were shown to have complex multiple sequence-indexed satisfaction relations $\models^{\hat{d}}$, where $\hat{d}$ represents sequences. The proposed IHpCTL has a simple single satisfaction relation $\models^{\star}$, which is compatible with the standard single satisfaction relation of CTL. Using this simple satisfaction relation, we can naturally formalize the interaction between $[b]$ and $\sim$ in IHpCTL, and through this natural formulation, we can prove the required embedding theorem. However, a careful treatment of the interaction of $[b]$ and $\sim$ is required to prove some key lemmas of the embedding theorem. This rigorous treatment represents a technical contribution of this study.

The remainder of this paper is organized as follows. In Section 2, we introduce the logic IHpCTL and introduce some basic propositions for IHpCTL. In Section 3, we prove the theorems for embedding IHpCTL into HpCTL and pCTL, and using these embedding theorems, we prove the theorems for relative decidabilities for IHpCTL with respect to HpCTL and pCTL. In Section 4, we conclude our study and address some illustrative examples.

2 LOGIC

Formulas of *inconsistency-tolerant hierarchical probabilistic computation tree logic* (IHpCTL) are constructed from countably many propositional variables by $\rightarrow$ (implication), $\wedge$ (conjunction), $\vee$ (disjunction), $\neg$ (classical negation), $\sim$ (paraconsistent negation), X (next time), G (globally in the future), F (eventually in the future), U (until), R (release), A (all computation paths), E (some computation path), $\mathrm{P}_{\leq x}$ (less than or equal probability), $\mathrm{P}_{\geq x}$ (greater than or equal probability), $\mathrm{P}_{<x}$ (less than probability), $\mathrm{P}_{>x}$ (greater than probability), and $[b]$ (hierarchical or sequence modal operator) where b is a sequence. *Sequences* are constructed from countably many atomic sequences and $\emptyset$ (empty sequence) by ; (composition). The symbols X, G, F, U, and R are called *temporal operators*, the symbols A and E are called *path quantifiers*, and the symbols $\mathrm{P}_{\leq x}$, $\mathrm{P}_{\geq x}$, $\mathrm{P}_{<x}$, and $\mathrm{P}_{>x}$ are called *probabilistic* or *probability operators*. We use lower-case letters $p, q, r, \ldots$ to denote propositional variables, Greek small letters $\alpha, \beta, \gamma, \ldots$ to denote formulas, and lower-case letters $b, c, d, \ldots$ to denote sequences. we use an expression $\alpha \leftrightarrow \beta$ to denote the formula $(\alpha{\rightarrow}\beta)\wedge(\beta{\rightarrow}\alpha)$ and an expression $A \equiv B$ to denote the syntactical identity between A and B. An expression $[\emptyset]\alpha$ means α, and expressions $[\emptyset\,;b]\alpha$ and $[b\,;\,\emptyset]\alpha$ mean $[b]\alpha$. We use the symbol SE to denote the set of all sequences (including the empty sequence $\emptyset$) and the symbol ω to denote the set of all natural numbers. Furthermore, we use the symbol Φ to denote a non-empty set of propositional variables, the symbol Φ' to denote the set $\{p' \mid p \in \Phi\}$ of new propositional variable, the symbol $\Phi^{\sim}$ is used to denote the set $\{{\sim}p \mid p \in \Phi\}$, the symbol $\Phi^{[d]}$ to denote the set $\{[d]p \mid p \in \Phi\}$, the symbol $\Phi'^{[d]}$ to denote the set $\{[d]\gamma \mid \gamma \in \Phi\cup\Phi'\}$, the symbol $\Phi^{[d]\prime}$ to denote the set $\{\gamma' \mid \gamma \in \Phi\cup\Phi^{[d]}\}$, the symbol $\Phi^{\sim[d]}$ to denote the set $\{[d]\gamma \mid \gamma \in \Phi\cup\Phi^{\sim}\}$, and the symbol $\Phi^{[d]\sim}$ to denote the set $\{{\sim}\gamma \mid \gamma \in \Phi\cup\Phi^{[d]}\}$. We assume the following commutativity condition: For any $p \in \Phi$ and any $d \in$ SE, $([d]p)' = [d](p')$ (i.e., it can simply be denoted as $[d]p'$). Then, we have the following fact by this commutativity condition: $\Phi'^{[d]} = \Phi^{[d]\prime}$.

Definition 2.1. *Let x be in $[0,1]$. Formulas α and sequences b of* IHpCTL *are defined by the following grammar, assuming p and e represent propositional variables and atomic sequences, respectively:*

$$\alpha ::= p \mid \alpha\wedge\alpha \mid \alpha\vee\alpha \mid \alpha\rightarrow\alpha \mid \neg\alpha \mid {\sim}\alpha \mid \mathrm{AX}\alpha \mid \mathrm{EX}\alpha \mid \mathrm{AG}\alpha \mid \mathrm{EG}\alpha \mid \mathrm{AF}\alpha \mid \mathrm{EF}\alpha \mid \mathrm{A}(\alpha\mathrm{U}\alpha) \mid \mathrm{E}(\alpha\mathrm{U}\alpha) \mid \mathrm{A}(\alpha\mathrm{R}\alpha) \mid \mathrm{E}(\alpha\mathrm{R}\alpha) \mid \mathrm{P}_{\leq x}\alpha \mid \mathrm{P}_{\geq x}\alpha \mid \mathrm{P}_{<x}\alpha \mid \mathrm{P}_{>x}\alpha \mid [b]\alpha.$$

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

We use an expression $\overline{[d]}$ to denote $[d_0][d_1]\cdots[d_i]$ with $i \in \omega$, $d_i \in$ SE and $d_0 \equiv \emptyset$. The expression $\overline{[d]}$ can be the empty sequence and is not uniquely determined. For example, if $d \equiv d_1\,;\,d_2\,;\,d_3$ where d_1, d_2 and d_3 are atomic sequences, then $\overline{[d]}$ means $[d_1][d_2][d_3]$, $[d_1\,;\,d_2][d_3]$, $[d_1][d_2\,;\,d_3]$ or $[d_1\,;\,d_2\,;\,d_3]$. Note that $\overline{[d]}$ can be $[d]$ (i.e., $\overline{[d]}$ includes $[d]$).

Remark 2.2. *We make the following remarks.*

1. *The inconsistency-tolerant negation connective $\sim$ characterizes* inconsistency-tolerant logics *(also referred to as* paraconsistent logics*)* (Priest, 2002; da Costa et al., 1995) *that reject the law $(\alpha\wedge{\sim}\alpha)\rightarrow\beta$ of explosion. In comparison with other logics, inconsistency-tolerant logics can be used in inconsistency-tolerant reasoning. For example, the following scenario is undesirable in a realistic situation. The formulas of the form $(s(x)\wedge{\sim}s(x))\rightarrow d(x)$ are valid for any symptom s and disease d, where ${\sim}s(x)$ means that "a person x does not have a symptom s" and $d(x)$ means that "a person x suffers from a disease d." The scenario described as $melancholia(john)\wedge{\sim}melancholia(john)$ will naturally emerge from the vague definition of melancholia (i.e., the statement "John has melancholia" may be judged as true or false based on the perception of different doctors or pathologists). In*

this case, the formula $(melancholia(john) \wedge \sim melancholia(john)) \rightarrow cancer(john)$ is valid in non-inconsistency-tolerant logics, but invalid in inconsistency-tolerant logics.

2. *The hierarchical modal operator $[b]$ can be used to represent the concepts of hierarchical information in the following manner: a sequence structure produces a monoid $\langle M, ;, \emptyset\rangle$ with the following informational interpretation* (Wansing, 1993)*: (1) M is a set of pieces of ordered information (i.e., a set of sequences); (2) ';' is a binary operator (on M) that combines two pieces of information (i.e., it is a concatenation operator on sequences); (3) $\emptyset$ is an empty piece of information (i.e., an empty sequence). Then, formulas of the form $[b_1 ; b_2 ; \cdots ; b_n]\alpha$ imply that α is true with the sequence $b_1 ; b_2 ; \cdots ; b_n$ of ordered pieces of information. In addition, formulas with the form $[\emptyset]\alpha$, which coincide with α, imply that α is true without any information (i.e., it is an eternal truth in the sense of classical logic).*

We define the logic IHpCTL as follows.

Definition 2.3 (IHpCTL)**.** *A structure $(S, S_0, R, \mu_s, L^\star)$ is an* inconsistency-tolerant hierarchical probabilistic model *iff*

1. *S is the set of states,*
2. *S_0 is a set of initial states and $S_0 \subseteq S$,*
3. *R is a binary relation on S which satisfies the condition:*
   $\forall s \in S\ \exists s' \in S\ [(s, s') \in R]$,
4. *μ_s is a certain probability measure concerning $s \in S$: a set of paths beginning at s is mapped into a real number in $[0,1]$ (i.e., any sets of paths starting from s are measurable),*
5. *$L^\star$ is a mapping from S to the power set of $\bigcup_{d \in \mathrm{SE}} \Phi^{\sim[d]}$.*

A path *in an inconsistency-tolerant hierarchical probabilistic model is an infinite sequence of states, $\pi = s_0, s_1, s_2, \ldots$ such that $\forall i \geq 0\ [(s_i, s_{i+1}) \in R]$. We use the symbol Ω_s to denote the set of all paths beginning at s.*

An inconsistency-tolerant hierarchical probabilistic satisfaction relation $(M,s) \models^\star \alpha$ *for any formula α, where M is an inconsistency-tolerant hierarchical probabilistic model $(S, S_0, R, \mu_s, L^\star)$, and s represents a state in S, is defined by the following clauses:*

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

28. *for any* $x \in [0,1]$, $(M,s) \models^{\star} \overline{[d]}{\sim}\mathrm{P}_{\leq x}\alpha$ *iff* $\mu_s(\{w \in \Omega_s \mid (M,s) \models^{\star} \overline{[d]}{\sim}\alpha\}) > x$,
29. *for any* $x \in [0,1]$, $(M,s) \models^{\star} \overline{[d]}{\sim}\mathrm{P}_{\geq x}\alpha$ *iff* $\mu_s(\{w \in \Omega_s \mid (M,s) \models^{\star} \overline{[d]}{\sim}\alpha\}) < x$,
30. *for any* $x \in [0,1]$, $(M,s) \models^{\star} \overline{[d]}{\sim}\mathrm{P}_{<x}\alpha$ *iff* $\mu_s(\{w \in \Omega_s \mid (M,s) \models^{\star} \overline{[d]}{\sim}\alpha\}) \geq x$,
31. *for any* $x \in [0,1]$, $(M,s) \models^{\star} \overline{[d]}{\sim}\mathrm{P}_{>x}\alpha$ *iff* $\mu_s(\{w \in \Omega_s \mid (M,s) \models^{\star} \overline{[d]}{\sim}\alpha\}) \leq x$,
32. $(M,s) \models^{\star} \overline{[d]}{\sim}\mathrm{AX}\alpha$ *iff* $\exists s_1 \in S$ $[(s,s_1) \in R$ *and* $(M,s_1) \models^{\star} \overline{[d]}{\sim}\alpha]$,
33. $(M,s) \models^{\star} \overline{[d]}{\sim}\mathrm{EX}\alpha$ *iff* $\forall s_1 \in S$ $[(s,s_1) \in R$ *implies* $(M,s_1) \models^{\star} \overline{[d]}{\sim}\alpha]$,
34. $(M,s) \models^{\star} \overline{[d]}{\sim}\mathrm{AG}\alpha$ *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* $(M,s_i) \models^{\star} \overline{[d]}{\sim}\alpha$,
35. $(M,s) \models^{\star} \overline{[d]}{\sim}\mathrm{EG}\alpha$ *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* $(M,s_i) \models^{\star} \overline{[d]}{\sim}\alpha$,
36. $(M,s) \models^{\star} \overline{[d]}{\sim}\mathrm{AF}\alpha$ *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* $(M,s_i) \models^{\star} \overline{[d]}{\sim}\alpha$,
37. $(M,s) \models^{\star} \overline{[d]}{\sim}\mathrm{EF}\alpha$ *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* $(M,s_i) \models^{\star} \overline{[d]}{\sim}\alpha$,
38. $(M,s) \models^{\star} \overline{[d]}{\sim}\mathrm{A}(\alpha\mathrm{U}\beta)$ *iff there is a path* $\pi \equiv s_0, s_1, s_2, \ldots$, *where* $s \equiv s_0$, *and for all states* s_j *along* π, *we have* $(M,s_j) \models^{\star} \overline{[d]}{\sim}\beta$ *or* $\exists 0 \leq k < j$ $(M,s_k) \models^{\star} \overline{[d]}{\sim}\alpha$,
39. $(M,s) \models^{\star} \overline{[d]}{\sim}\mathrm{E}(\alpha\mathrm{U}\beta)$ *iff for all paths* $\pi \equiv s_0, s_1, s_2, \ldots$, *where* $s \equiv s_0$, *and all states* s_j *along* π, *we have* $(M,s_j) \models^{\star} \overline{[d]}{\sim}\beta$ *or* $\exists 0 \leq k < j$ $(M,s_k) \models^{\star} \overline{[d]}{\sim}\alpha$,
40. $(M,s) \models^{\star} \overline{[d]}{\sim}\mathrm{A}(\alpha\mathrm{R}\beta)$ *iff there is a path* $\pi \equiv s_0, s_1, s_2, \ldots$, *where* $s \equiv s_0$, *and for some state* s_j *along* π, *we have* $(M,s_j) \models^{\star} \overline{[d]}{\sim}\beta$ *and* $\forall 0 \leq k < j$ $(M,s_k) \models^{\star} \overline{[d]}{\sim}\alpha$,
41. $(M,s) \models^{\star} \overline{[d]}{\sim}\mathrm{E}(\alpha\mathrm{R}\beta)$ *iff for all paths* $\pi \equiv s_0, s_1, s_2, \ldots$, *where* $s \equiv s_0$, *there is a state* s_j *along* π *such that* $(M,s_j) \models^{\star} \overline{[d]}{\sim}\beta$ *and* $\forall 0 \leq k < j$ $(M,s_k) \models^{\star} \overline{[d]}{\sim}\alpha$,
42. $(M,s) \models^{\star} {\sim}\overline{[d]}{\sim}\alpha$ *iff* $(M,s) \models^{\star} \overline{[d]}\alpha$,
43. $(M,s) \models^{\star} {\sim}\overline{[d]}[b]\alpha$ *iff* $(M,s) \models^{\star} {\sim}[d\,;b]\alpha$,
44. $(M,s) \models^{\star} {\sim}\overline{[d]}(\alpha\wedge\beta)$ *iff* $(M,s) \models^{\star} {\sim}\overline{[d]}\alpha$ *or* $(M,s) \models^{\star} {\sim}\overline{[d]}\beta$,
45. $(M,s) \models^{\star} {\sim}\overline{[d]}(\alpha\vee\beta)$ *iff* $(M,s) \models^{\star} {\sim}\overline{[d]}\alpha$ *and* $(M,s) \models^{\star} {\sim}\overline{[d]}\beta$,
46. $(M,s) \models^{\star} {\sim}\overline{[d]}(\alpha\rightarrow\beta)$ *iff* $(M,s) \not\models^{\star} {\sim}\overline{[d]}\alpha$ *and* $(M,s) \models^{\star} {\sim}\overline{[d]}\beta$,
47. $(M,s) \models^{\star} {\sim}\overline{[d]}\neg\alpha$ *iff* $(M,s) \not\models^{\star} {\sim}\overline{[d]}\alpha$,
48. *for any* $x \in [0,1]$, $(M,s) \models^{\star} {\sim}\overline{[d]}\mathrm{P}_{\leq x}\alpha$ *iff* $\mu_s(\{w \in \Omega_s \mid (M,s) \models^{\star} {\sim}\overline{[d]}\alpha\}) > x$,
49. *for any* $x \in [0,1]$, $(M,s) \models^{\star} {\sim}\overline{[d]}\mathrm{P}_{\geq x}\alpha$ *iff* $\mu_s(\{w \in \Omega_s \mid (M,s) \models^{\star} {\sim}\overline{[d]}\alpha\}) < x$,
50. *for any* $x \in [0,1]$, $(M,s) \models^{\star} {\sim}\overline{[d]}\mathrm{P}_{<x}\alpha$ *iff* $\mu_s(\{w \in \Omega_s \mid (M,s) \models^{\star} {\sim}\overline{[d]}\alpha\}) \geq x$,
51. *for any* $x \in [0,1]$, $(M,s) \models^{\star} {\sim}\overline{[d]}\mathrm{P}_{>x}\alpha$ *iff* $\mu_s(\{w \in \Omega_s \mid (M,s) \models^{\star} {\sim}\overline{[d]}\alpha\}) \leq x$,
52. $(M,s) \models^{\star} {\sim}\overline{[d]}\mathrm{AX}\alpha$ *iff* $\exists s_1 \in S$ $[(s,s_1) \in R$ *and* $(M,s_1) \models^{\star} {\sim}\overline{[d]}\alpha]$,
53. $(M,s) \models^{\star} {\sim}\overline{[d]}\mathrm{EX}\alpha$ *iff* $\forall s_1 \in S$ $[(s,s_1) \in R$ *implies* $(M,s_1) \models^{\star} {\sim}\overline{[d]}\alpha]$,
54. $(M,s) \models^{\star} {\sim}\overline{[d]}\mathrm{AG}\alpha$ *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* $(M,s_i) \models^{\star} {\sim}\overline{[d]}\alpha$,
55. $(M,s) \models^{\star} {\sim}\overline{[d]}\mathrm{EG}\alpha$ *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* $(M,s_i) \models^{\star} {\sim}\overline{[d]}\alpha$,
56. $(M,s) \models^{\star} {\sim}\overline{[d]}\mathrm{AF}\alpha$ *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* $(M,s_i) \models^{\star} {\sim}\overline{[d]}\alpha$,
57. $(M,s) \models^{\star} {\sim}\overline{[d]}\mathrm{EF}\alpha$ *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* $(M,s_i) \models^{\star} {\sim}\overline{[d]}\alpha$,
58. $(M,s) \models^{\star} {\sim}\overline{[d]}\mathrm{A}(\alpha\mathrm{U}\beta)$ *iff there is a path* $\pi \equiv s_0, s_1, s_2, \ldots$, *where* $s \equiv s_0$, *and for all states* s_j *along* π, *we have* $(M,s_j) \models^{\star} {\sim}\overline{[d]}\beta$ *or* $\exists 0 \leq k < j$ $(M,s_k) \models^{\star} {\sim}\overline{[d]}\alpha$,
59. $(M,s) \models^{\star} {\sim}\overline{[d]}\mathrm{E}(\alpha\mathrm{U}\beta)$ *iff for all paths* $\pi \equiv s_0, s_1, s_2, \ldots$, *where* $s \equiv s_0$, *and all states* s_j *along* π, *we have* $(M,s_j) \models^{\star} {\sim}\overline{[d]}\beta$ *or* $\exists 0 \leq k < j$ $(M,s_k) \models^{\star} {\sim}\overline{[d]}\alpha$,
60. $(M,s) \models^{\star} {\sim}\overline{[d]}\mathrm{A}(\alpha\mathrm{R}\beta)$ *iff there is a path* $\pi \equiv s_0, s_1, s_2, \ldots$, *where* $s \equiv s_0$, *and for some state* s_j *along* π, *we have* $(M,s_j) \models^{\star} {\sim}\overline{[d]}\beta$ *and* $\forall 0 \leq k < j$ $(M,s_k) \models^{\star} {\sim}\overline{[d]}\alpha$,
61. $(M,s) \models^{\star} {\sim}\overline{[d]}\mathrm{E}(\alpha\mathrm{R}\beta)$ *iff for all paths* $\pi \equiv s_0, s_1, s_2, \ldots$, *where* $s \equiv s_0$, *there is a state* s_j *along* π *such that* $(M,s_j) \models^{\star} {\sim}\overline{[d]}\beta$ *and* $\forall 0 \leq k < j$ $(M,s_k) \models^{\star} {\sim}\overline{[d]}\alpha$.

A formula α *is* valid *in* IHpCTL *iff* $(M,s) \models^{\star} \alpha$ *holds for any inconsistency-tolerant hierarchical probabilistic model* $M := (S, S_0, R, \mu_s, L^{\star})$, *any* $s \in S$, *and any inconsistency-tolerant hierarchical probabilistic satisfaction relation* $\models^{\star}$ *on* M.

Definition 2.4. *Let* M *be an inconsistency-tolerant hierarchical probabilistic model* $M := (S, S_0, R, \mu_s, L^{\star})$ *for* IHpCTL, *and* $\models^{\star}$ *be an inconsistency-tolerant hierarchical probabilistic satisfaction relation on* M. *Then, the* model checking problem *for* IHpCTL *is defined as follows. For any formula* α, *find the set* $\{s \in S \mid M, s \models^{\star} \alpha\}$.

Remark 2.5. *We make the following remarks.*

1. *The logic* IHpCTL *is an extension of the following temporal logics: Probabilistic computation tree logic (*pCTL*) studied in* (Aziz et al., 1995; Bianco and de Alfaro, 1995), *inconsistency-tolerant computation tree logic (also referred to as* pCTL,

but different from the aforementioned probabilistic one) proposed in (Kamide and Endo, 2018), *and hierarchical computation tree logic (*sCTL*) proposed in* (Kamide and Yano, 2017; Kamide, 2018).

2. *The definition of μ_s is not precisely and explicitly given in this study, because (1) the proposed translation from* IHpCTL *to* HpCTL *is independent of the setting of μ_s and (2) there are some different ways of defining μ_s.*
3. *There are some ways of defining a probability measure μ_s. For example, two probability measures μ_s^+ and μ_s^-, which were defined on a Borel σ-algebra $\mathcal{B}_s$ ($\subseteq 2^{\Omega_s}$), were proposed in* (Bianco and de Alfaro, 1995) *for* pCTL. *A probability measure μ^s, which is concerned with certain* discrete Markov processes, *was proposed in* (Aziz et al., 1995) *for* pCTL.
4. *The setting of the conditions concerning the negated implication and negated negation in* IHpCTL *adopts the axiom schemes $\sim(\alpha{\rightarrow}\beta) \leftrightarrow \neg{\sim}\alpha{\wedge}{\sim}\beta$ and ${\sim}\neg\alpha \leftrightarrow \neg{\sim}\alpha$. These axiom schemes were originally introduced by De and Omori in* (De and Omori, 2015).
5. *The single-satisfaction relation $\models^\star$ of* IHpCTL *is compatible with the standard single-satisfaction relation of* CTL. *By using this satisfaction relation, we can simply and uniformly handle both $\sim$ and $[b]$.*

Proposition 2.6. *Let M be an inconsistency-tolerant hierarchical probabilistic model $(S,S_0,R,\mu_s,L^\star)$, and let s be a state in S, Then, the following clauses hold for* IHpCTL*: For any formula α and any $b,c,d \in$ SE,*

1. $(M,s) \models^\star [b][c]\alpha$ *iff* $(M,s) \models^\star [b\,;\,c]\alpha$,
2. $(M,s) \models^\star \overline{[d]}\alpha$ *iff* $(M,s) \models^\star [d]\alpha$,
3. $(M,s) \models^\star \overline{[d]}{\sim}\alpha$ *iff* $(M,s) \models^\star {\sim}\overline{[d]}\alpha$.

Proof. By induction on α. □

3 EMBEDDABILITY AND RELATIVE DECIDABILITY

In order to prove the relative decidability theorem for IHpCTL with respect to pCTL, we need a theorem for embedding IHpCTL into the $\sim$-free part HpCTL of IHpCTL. By combining this theorem for embedding IHpCTL into HpCTL and the previously proved theorem in (Kamide and Yano, 2019) for embedding HpCTL into pCTL, we can obtain the relative decidability of IHpCTL with respect to pCTL. Thus, we introduce HpCTL below. The language and formulas of HpCTL are respectively obtained from those of IHpCTL by deleting $\sim$.

Definition 3.1 (HpCTL). *A structure (S,S_0,R,μ_s,L) is a* hierarchical probabilistic model *iff S, S_0, R, and μ_s are the same as those in Definition 2.3, and*

$$L \text{ is a mapping from } S \text{ to the power set of } \bigcup_{d\in \mathrm{SE}} \Phi^{[d]}.$$

A path *in a hierarchical probabilistic model is defined in a similar way as in Definition 2.3.*

A hierarchical probabilistic satisfaction relation $(M,s) \models \alpha$ *for any formula α, where M is a hierarchical probabilistic model (S,S_0,R,μ_s,L) and s represents a state in S, is defined inductively by the same clauses 3 – 21 in Definition 2.3 (but $\models^\star$ in Definition 2.3 should be replaced with $\models$) and the following clause:*

For any $p \in \Phi$, $(M,s) \models [d]p$ iff $[d]p \in L(s)$.

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

We define the logic pCTL, which was originally studied in (Aziz et al., 1995; Bianco and de Alfaro, 1995).

Definition 3.2 (pCTL). *The logic* pCTL *is defined as the $[b]$-free part of* HpCTL *(i.e., it is obtained from* HpCTL *by replacing the sequences d and b with the empty sequence $\emptyset$).*

We define a translation from IHpCTL to HpCTL.

Definition 3.3. *The language $\mathcal{L}^i$ (the set of formulas) of* IHpCTL *is defined using Φ, $\wedge,\vee,\rightarrow,\neg$, $\sim$, X, G, F, U, R, A, E, $\mathrm{P}_{\leq x},\mathrm{P}_{\geq x},\mathrm{P}_{<x},\mathrm{P}_{>x}$, and $[b]$. The language $\mathcal{L}$ of* HpCTL *is obtained from $\mathcal{L}^i$ by adding Φ' and deleting $\sim$. A mapping f from $\mathcal{L}^i$ to $\mathcal{L}$ is defined inductively by:*

1. *for any $p \in \Phi$, $f(p) := p$ and $f({\sim}p) := p' \in \Phi'$,*
2. $f(\alpha \sharp \beta) := f(\alpha) \sharp f(\beta)$ *where* $\sharp \in \{\wedge,\vee,\rightarrow\}$,
3. $f(\sharp\alpha) := \sharp f(\alpha)$ *where* $\sharp \in \{\neg,\mathrm{AX},\mathrm{EX},\mathrm{AG},\mathrm{EG},\mathrm{AF},\mathrm{EF},\mathrm{P}_{\leq x},\mathrm{P}_{\geq x},\mathrm{P}_{<x},\mathrm{P}_{>x},[b]\}$,
4. $f(\mathrm{A}(\alpha \sharp \beta)) := \mathrm{A}(f(\alpha) \sharp f(\beta))$ *where* $\sharp \in \{\mathrm{U},\mathrm{R}\}$,
5. $f(\mathrm{E}(\alpha \sharp \beta)) := \mathrm{E}(f(\alpha) \sharp f(\beta))$ *where* $\sharp \in \{\mathrm{U},\mathrm{R}\}$,
6. $f({\sim}{\sim}\alpha) := f(\alpha)$,
7. $f({\sim}(\alpha{\wedge}\beta)) := f({\sim}\alpha){\vee}f({\sim}\beta)$,
8. $f({\sim}(\alpha{\vee}\beta)) := f({\sim}\alpha){\wedge}f({\sim}\beta)$,
9. $f({\sim}(\alpha{\rightarrow}\beta)) := \neg f({\sim}\alpha){\wedge}f({\sim}\beta)$,
10. $f({\sim}\sharp\alpha) := \sharp f({\sim}\alpha)$ *where* $\sharp \in \{\neg,[b]\}$,
11. $f({\sim}\mathrm{AX}\alpha) := \mathrm{EX}f({\sim}\alpha)$,
12. $f({\sim}\mathrm{EX}\alpha) := \mathrm{AX}f({\sim}\alpha)$,

13. $f(\sim\mathrm{AG}\alpha) := \mathrm{EF}f(\sim\alpha)$,

14. $f(\sim\mathrm{EG}\alpha) := \mathrm{AF}f(\sim\alpha)$,

15. $f(\sim\mathrm{AF}\alpha) := \mathrm{EG}f(\sim\alpha)$,

16. $f(\sim\mathrm{EF}\alpha) := \mathrm{AG}f(\sim\alpha)$,

17. $f(\sim\mathrm{A}(\alpha\mathrm{U}\beta)) := \mathrm{E}(f(\sim\alpha)\mathrm{R}f(\sim\beta))$,

18. $f(\sim\mathrm{E}(\alpha\mathrm{U}\beta)) := \mathrm{A}(f(\sim\alpha)\mathrm{R}f(\sim\beta))$,

19. $f(\sim\mathrm{A}(\alpha\mathrm{R}\beta)) := \mathrm{E}(f(\sim\alpha)\mathrm{U}f(\sim\beta))$,

20. $f(\sim\mathrm{E}(\alpha\mathrm{R}\beta)) := \mathrm{A}(f(\sim\alpha)\mathrm{U}f(\sim\beta))$,

21. $f(\sim\mathrm{P}_{\leq x}\alpha) := \mathrm{P}_{>x}f(\sim\alpha)$,

22. $f(\sim\mathrm{P}_{\geq x}\alpha) := \mathrm{P}_{<x}f(\sim\alpha)$,

23. $f(\sim\mathrm{P}_{<x}\alpha) := \mathrm{P}_{\geq x}f(\sim\alpha)$,

24. $f(\sim\mathrm{P}_{>x}\alpha) := \mathrm{P}_{\leq x}f(\sim\alpha)$.

Lemma 3.4. *Let f be the mapping defined in Definition 3.3. For any inconsistency-tolerant hierarchical probabilistic model $M := (S, S_0, R, \mu_s, L^\star)$ of* IHpCTL, *and any inconsistency-tolerant hierarchical probabilistic satisfaction relation $\models^\star$ on M, we can construct a hierarchical probabilistic model $N := (S, S_0, R, \mu_s, L)$ of* HpCTL *and a hierarchical probabilistic satisfaction relation $\models$ on N such that for any formula α in $\mathcal{L}^i$, any sequence d in $\mathcal{L}^i$, and any state s in S, $(M,s) \models^\star [d]\alpha$ iff $(N,s) \models [d]f(\alpha)$.*

Proof. Suppose that M is an inconsistency-tolerant hierarchical probabilistic model $(S, S_0, R, \mu_s, L^\star)$ s.t.

$L^\star$ is a mapping from S to the power set of $\bigcup_{d\in\mathrm{SE}} \Phi^{\sim[d]}$.

We then define a hierarchical probabilistic model $N := (S, S_0, R, \mu_s, L)$ such that

1. L is a mapping from S to the power set of $\bigcup_{d\in\mathrm{SE}} \Phi'^{[d]}$,
2. for any $s \in S$, any $p \in \Phi$, and any $c \in \mathrm{SE}$,
   (a) $[c]p \in L^\star(s)$ iff $[c]p \in L(s)$,
   (b) $[c]{\sim}p \in L^\star(s)$ iff $[c]p' \in L(s)$.

This lemma is then proved by induction on α.

• Base step:

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

• Induction step: We show some cases.

1. Case $\alpha \equiv [b]\beta$ ($b \in$SE): We obtain: $(M,s) \models^\star [d][b]\beta$ iff $(M,s) \models^\star [d\ ;\ b]\beta$ iff $(N,s) \models [d\ ;\ b]f(\beta)$ (by induction hypothesis) iff $(N,s) \models [d][b]f(\beta)$ iff $(N,s) \models [d]f([b]\beta)$ (by the definition of f).
2. Case $\alpha \equiv \mathrm{P}_{\leq x}\beta$: We obtain: $(M,s) \models^\star [d]\mathrm{P}_{\leq x}\beta$ iff $\mu_s(\{w \in \Omega_s \mid (M,w) \models^\star [d]\beta\}) \leq x$ iff $\mu_s(\{w \in \Omega_s \mid (N,w) \models [d]f(\beta)\}) \leq x$ (by induction hypothesis) iff $(N,s) \models [d]\mathrm{P}_{\leq x}f(\beta)$ iff $(N,s) \models [d]f(\mathrm{P}_{\leq x}\beta)$ (by the definition of f).
3. Case $\alpha \equiv {\sim}[b]\beta$ ($b \in$SE): We obtain: $(M,s) \models^\star [d]{\sim}[b]\beta$ iff $(M,s) \models^\star {\sim}[d][b]\beta$ (by Proposition 2.6 (3)) iff $(M,s) \models^\star {\sim}[d\ ;\ b]\beta$ iff $(M,s) \models^\star [d\ ;\ b]{\sim}\beta$ (by Proposition 2.6 (3)) iff $(N,s) \models [d\ ;\ b]f(\sim\beta)$ (by induction hypothesis) iff $(N,s) \models [d][b]f(\sim\beta)$ iff $(N,s) \models [d]f(\sim[b]\beta)$ (by the definition of f).
4. Case $\alpha \equiv {\sim}{\sim}\beta$: We obtain: $(M,s) \models^\star [d]{\sim}{\sim}\beta$ iff $(M,s) \models^\star [d]\beta$ iff $(N,s) \models [d]f(\beta)$ (by induction hypothesis) $(N,s) \models [d]f(\sim\sim\beta)$ (by the definition of f).
5. Case $\alpha \equiv {\sim}\neg\beta$: We obtain: $(M,s) \models^\star [d]{\sim}\neg\beta$ iff $(M,s) \not\models^\star [d]{\sim}\beta$ iff $(N,s) \not\models [d]f(\sim\beta)$ (by induction hypothesis) iff $(N,s) \models [d]\neg f(\sim\beta)$ iff $(N,s) \models [d]f(\sim\neg\beta)$ (by the definition of f).
6. Case $\alpha \equiv {\sim}(\beta\rightarrow\gamma)$: We obtain: $(M,s) \models^\star [d]{\sim}(\beta\rightarrow\gamma)$ iff $(M,s) \not\models^\star [d]{\sim}\beta$ and $(M,s) \models^\star [d]{\sim}\gamma$ iff $(N,s) \not\models [d]f(\sim\beta)$ and $(N,s) \models [d]f(\sim\gamma)$ (by induction hypothesis) iff $(N,s) \models [d](\neg f(\sim\beta) \wedge f(\sim\gamma))$ iff $(N,s) \models [d]f(\sim(\beta\rightarrow\gamma))$ (by the definition of f).
7. Case $\alpha \equiv {\sim}\mathrm{A}(\beta\mathrm{U}\gamma)$: We obtain:
   $(M,s) \models^\star [d]{\sim}\mathrm{A}(\beta\mathrm{U}\gamma)$
   iff there is a path $\pi \equiv s_0, s_1, s_2, \ldots$, where $s \equiv s_0$, and for all states s_j along π, we have $(M,s_j) \models^\star [d]{\sim}\gamma$ or $\exists 0 \leq k < j\ (M,s_k) \models^\star [d]{\sim}\beta$
   iff there is a path $\pi \equiv s_0, s_1, s_2, \ldots$, where $s \equiv s_0$, and for all states s_j along π, we have $(N,s_j) \models [d]f(\sim\gamma)$ or $\exists 0 \leq k < j\ (N,s_k) \models [d]f(\sim\beta)$ (by induction hypothesis)
   iff $(N,s) \models [d](\mathrm{E}(f(\sim\beta)\mathrm{R}f(\sim\gamma)))$
   iff $(N,s) \models [d]f(\sim\mathrm{A}(\beta\mathrm{U}\gamma))$ (by the definition of f).
8. Case $\alpha \equiv {\sim}\mathrm{P}_{\leq x}\beta$: We obtain: $(M,s) \models^\star [d]{\sim}\mathrm{P}_{\leq x}\beta$ iff $\mu_s(\{w \in \Omega_s \mid (M,w) \models^\star [d]{\sim}\beta\}) > x$ iff $\mu_s(\{w \in \Omega_s \mid (N,w) \models [d]f(\sim\beta)\}) > x$ (by induction hypothesis) iff $(N,s) \models [d]\mathrm{P}_{>x}f(\sim\beta)$ iff $(N,s) \models [d]f(\sim\mathrm{P}_{\leq x}\beta)$ (by the definition of f).

□

Lemma 3.5. *Let f be the mapping defined in Definition 3.3. For any hierarchical probabilistic model*

$N := (S, S_0, R, \mu_s, L)$ *of* HpCTL*, and any hierarchical probabilistic satisfaction relation* $\models$ *on* N*, we can construct an inconsistency-tolerant hierarchical probabilistic model* $M := (S, S_0, R, \mu_s, L^\star)$ *of* IHpCTL *and an inconsistency-tolerant hierarchical probabilistic satisfaction relation* $\models^\star$ *on* M *such that for any formula* α *in* $\mathcal{L}^i$*, any* d *in* $\mathcal{L}^i$*, and any state* s *in* S*,* $(N, s) \models [d]f(\alpha)$ *iff* $(M, s) \models^\star [d]\alpha$*.*

Proof. Similar to the proof of Lemma 3.4. □

We then obtain the following theorem.

Theorem 3.6 (Embedding from IHpCTL into HpCTL)**.** IHpCTL *is embeddable into* HpCTL*. Namely, we have the following. Let* f *be the mapping defined in Definition 3.3. For any formula* α*,* α *is valid in* IHpCTL *iff* $f(\alpha)$ *is valid in* HpCTL*.*

Proof. By Lemmas 3.4 and 3.5 (by taking d as $\emptyset$). □

We can also obtain the following theorem.

Theorem 3.7 (Embedding from IHpCTL into pCTL)**.** IHpCTL *is embeddable into* pCTL*.*

Proof. By combining Theorem 3.6 and the theorem proved in (Kamide and Yano, 2019) for embedding HpCTL into pCTL. □

We then obtain the following theorem.

Theorem 3.8 (Relative decidability for IHpCTL with respect to HpCTL)**.** *If the model-checking, validity, and satisfiability problems for* HpCTL *with a certain probability measure are decidable, then the model-checking, validity, and satisfiability problems for* IHpCTL *with the same probability measure as that of* HpCTL *are also decidable.*

Proof. Suppose that the probability measure μ_s in the underlying inconsistency-tolerant hierarchical probabilistic model $(S, S_0, R, \mu_s, L^\star)$ of IHpCTL is the same as the underlying hierarchical probabilistic model (S, S_0, R, μ_s, L) of HpCTL. Suppose also that HpCTL with μ_s is decidable. Then, by the mapping f, a formula α of IHpCTL can be transformed into the corresponding formula $f(\alpha)$ of HpCTL. By Lemmas 3.4 and 3.5 and Theorem 3.6, the model checking, validity and satisfiability problems for IHpCTL can be transformed into those of HpCTL. Since the model checking, validity and satisfiability problems for HpCTL with μ_s are decidable by the assumption, the problems for IHpCTL with μ_s are also decidable. □

We can also obtain the following theorem.

Theorem 3.9 (Relative decidability for IHpCTL with respect to pCTL)**.** *If the model-checking, validity, and satisfiability problems for* pCTL *with a certain probability measure are decidable, then the model-checking, validity, and satisfiability problems for* IHpCTL *with the same probability measure as that of* pCTL *are also decidable.*

Proof. By combining Theorem 3.8 and the theorem proved in (Kamide and Yano, 2019) for the relative decidability of HpCTL with respect to pCTL. □

Remark 3.10. *The model checking problem for the logic* pCTL *with the probability measures* μ_s^+ *and* μ_s^- *introduced by Bianco and de Alfaro was shown to be decidable in* (Bianco and de Alfaro, 1995)*. The model checking problem for the logic* pCTL *with the probability measure* μ^s *introduced by Aziz et al. was shown to be decidable in* (Aziz et al., 1995)*. Thus, an extended* IHpCTL *with the above-mentioned probability measures by Bianco and de Alfaro or by Aziz et al. is also decidable by Theorem 3.9. If we consider a sublogic without any probability measures, the decision problems for such a logic are decidable.*

4 CONCLUSION AND REMARKS

In this study, the inconsistency-tolerant hierarchical probabilistic computation tree logic IHpCTL was introduced to establish a new extended model checking paradigm referred to as IHpCTL model checking, which can verify randomized, open, large, and complex concurrent systems. The proposed logic IHpCTL was shown to be embeddable into HpCTL and pCTL and relatively decidable with respect to HpCTL and pCTL. This means that the decidabilities of HpCTL and pCTL with certain probability measures imply the decidability of IHpCTL. This study thus showed that we can effectively reuse the previously proposed pCTL model-checking algorithms for IHpCTL model checking.

As an example of IHpCTL model checking, we next consider a lung cancer model presented as Figure 1. *Lung cancer*, also known as *lung carcinoma*, is a disease or malignant lung tumor characterized by uncontrolled cell growth in tissues of the lung (Wikipedia, 2020). This growth can spread beyond the lungs by the process of metastasis into nearby tissue or other parts of the body. In this example, we can express the following formula:

$$[\mathit{Cancer}\,;\mathit{LungCancer}]\mathrm{AG}(\mathit{stage}4 \wedge \mathit{hasMalignantTumor} \wedge \mathit{painful} \wedge {\sim}\mathit{healthy} \rightarrow \mathrm{EF}(\mathrm{P}_{\leq 0.89}\ \mathit{death} \wedge \mathrm{P}_{\geq 0.87}\ \mathit{death})$$

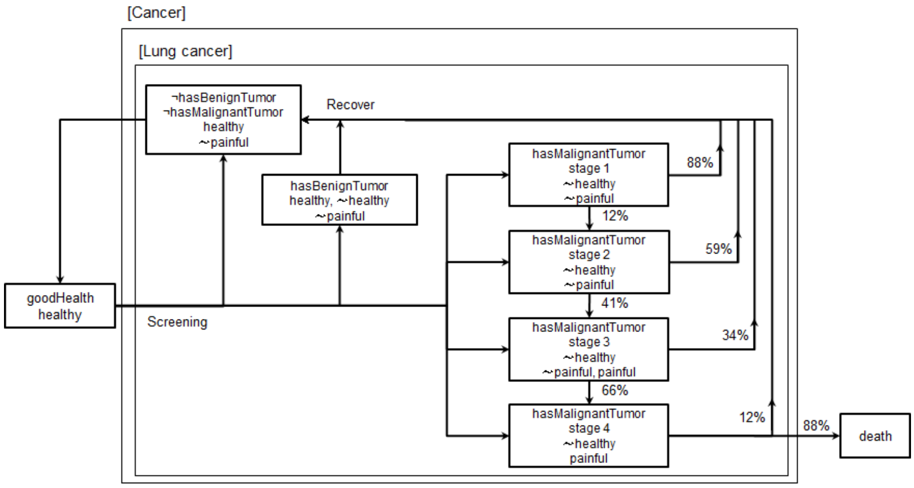

Figure 1: Lung cancer model.

which implies the following:

> "If a person with lung cancer Stage 4 has a malignant tumor (i.e., lung cancer), is not healthy, and has pain, then there is a probability of approximately 88 percent exists that this person will die in the near future."

This statement is true and we can verify it using the translation from IHpCTL to pCTL. In other words, the IHpCTL formula can be verified using the corresponding pCTL formula by the translation. Using the translation, we can obtain the corresponding pCTL formula as follows: AG($q_1 \wedge q_2 \wedge q_3 \wedge q_4 \rightarrow$ EF($P_{\leq 0.89}\ q_5 \wedge P_{\geq 0.87}\ q_5$) where q_1, q_2, q_3, q_4, and q_5 are distinct propositional variables.

Finally, we make the following remarks on related works. *Probabilistic temporal logics*, including pCTL, *inconsistency-tolerant (or paraconsistent) temporal logics*, *hierarchical (or sequential) temporal logics*, and their applications to probabilistic, inconsistency-tolerant, and hierarchical model checking, have been investigated by many researchers. For more information on probabilistic temporal logics and their model-checking applications, see (Hanson, 1994; Hansson and Jonsson, 1994; Aziz et al., 1995; Bianco and de Alfaro, 1995; Baier and Kwiatkowska, 1998; Kwiatkowska et al., 2011; Kamide and Koizumi, 2016; Baier et al., 2018; Kamide and Bernal J.P.A., 2019; Kamide and Yano, 2019). 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, 2011b; Kamide, 2015; Kamide and Koizumi, 2016; Kamide and Endo, 2018). For more information on hierarchical temporal logics and their model-checking applications, see (Kamide and Kaneiwa, 2009; Kaneiwa and Kamide, 2010; Kaneiwa and Kamide, 2011a; Kamide, 2015; Kamide and Yano, 2017; Kamide, 2018; Kamide and Yano, 2019). Finally, for a survey of a few closely related studies on probabilistic, inconsistency-tolerant, and hierarchical temporal logics and their applications, see (Kamide and Bernal J.P.A., 2019).

ACKNOWLEDGEMENTS

We would like to thank the anonymous referees for their valuable comments. This research was supported by Grant-in-Aid for the Japan Research Institute of Industrial Science and JSPS KAKENHI Grant Numbers JP18K11171 and JP16KK0007.

REFERENCES

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.

Baier, C., de Alfaro, L., Forejt, V., and Kwiatkowska, M. (2018). *Model Checking Probabilistic Systems, In: Handbook of Model Checking, pp. 963-999*. Springer.

Baier, C. and Kwiatkowska, M. (1998). Model checking for a probabilistic branching time logic with fairnes. *Distributed Computing*, 11:125–155.

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., and Peled, D. (1999). *Model checking*. The MIT Press.

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.

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

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.

Hanson, H. (1994). *Time and probability in formal design of distributed systems, real-time safety critical systems*. Elsevier.

Hansson, H. and Jonsson, B. (1994). A logic for reasoning about time and reliability. *Formal Aspects of Computing*, 6 (5):512–535.

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

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

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

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

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 Kaneiwa, K. (2009). Extended full computation-tree logic with sequence modal operator: Representing hierarchical tree structures. *Proceedings of the 22nd Australasian Joint Conference on Artificial Intelligence (AI'09), Lecture Notes in Artificial Intelligence*, 5866:485–494.

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 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.

Kamide, N. and Yano, Y. (2019). Towards hierarchical probabilistic ctl model checking: Theoretical foundation. *Proceedings of the 11th International Conference on Agents and Artificial Intelligence (ICAART 2019)*, 2:762–769.

Kaneiwa, K. and Kamide, N. (2010). Sequence-indexed linear-time temporal logic: Proof system and application. *Applied Artificial Intelligence*, 24 (10):896–913.

Kaneiwa, K. and Kamide, N. (2011a). Conceptual modeling in full computation-tree logic with sequence modal operator. *International Journal of Intelligent Systems*, 26 (7):636–651.

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

Kwiatkowska, M., Norman, G., and Parker, D. (2011). Prism 4.0: Verification of probabilistic real-time systems. *Proceedings of the 23rd International Conference on Computer Aided Verification (CAV 11), Lecture Notes in Computer Science*, 6806:585–591.

N. Kobayashi, T. S. and Yonezawa, A. (1999). Distributed concurrent linear logic programming. *Theoretical Computer Science*, 227:185–220.

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

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

Wikipedia (2020). Lung cancer. *Wikipedia, https://en.wikipedia.org/wiki/Lung_cancer*.