Towards Locative Inconsistency-tolerant Hierarchical Probabilistic CTL Model Checking: Survey and Future Work

Norihiro Kamide[1] and Juan Pedro Altamirano Bernal[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]*Panamerican University, Faculty of Engineering, Department of Electronic Engineering and Digital Systems, Josemaria Escriva de Balaguer No. 101, CP. 20190 Aguascalientes, Aguascalientes, Mexico*

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

Abstract: A locative inconsistency-tolerant hierarchical probabilistic computation tree logic (LIHpCTL) is introduced in this paper to establish the logical foundation of a new model checking paradigm. This logic is an extension of several previously proposed extensions of the standard temporal logic known as CTL, which is widely used for model checking. The extended model checking paradigm proposed is intended to appropriately verify locative (spatial), inconsistent, hierarchical, probabilistic (randomized), and time-dependent concurrent systems. Additionally, a survey of various studies on probabilistic, inconsistency-tolerant, and hierarchical temporal logics and their applications in model checking is conducted.

1 INTRODUCTION

In this paper, we introduce a *locative inconsistency-tolerant hierarchical probabilistic computation tree logic* (LIHpCTL), which is designed to form the logical foundation of a new model checking paradigm. This paradigm is intended to appropriately verify locative (spatial), inconsistent, hierarchical, probabilistic (randomized), and time-dependent concurrent systems. LIHpCTL is an extension of several previously proposed locative, inconsistency-tolerant, hierarchical, and probabilistic extensions of the standard temporal logic known as CTL (Clarke and Emerson, 1981), which is widely used for *model checking* (Clarke and Emerson, 1981; Clarke et al., 1999; Holzmann, 2006; Clarke et al., 2018). Model checking is a well-known formal and automated technique for verifying concurrent systems. Although the decidability of model checking based on LIHpCTL has not been determined yet, we present an illustrative example of the novel LIHpCTL-based model checking paradigm.

In what follows, we present a brief explanation of several standard and extended temporal logics and their applications to model checking. The following are well-known standard temporal logics typically used in model checking: *computation tree logic* (CTL) (Clarke and Emerson, 1981), *linear-time temporal logic* (LTL) (Pnueli, 1977), and *full computation tree logic* (CTL^*) (Emerson and Sistla, 1984; Emerson and Halpern, 1986). The logic CTL (Clarke and Emerson, 1981) is one of the most useful temporal logics for model checking; it is based on the branching-time paradigm, which uses computation trees to represent the passage of time. The logic LTL (Pnueli, 1977) is another form of temporal logic widely used for model checking; it is based on the linear-time paradigm, which uses linear order to represent the passage of time. The logic CTL^* (Emerson and Sistla, 1984; Emerson and Halpern, 1986) is more expressive than LTL and CTL; it is based on the branching-time paradigm with path quantifiers to represent the passage of time.

To extend the above-mentioned temporal logics, other useful non-classical logics must be combined and integrated in a natural way. This is also an important issue in mathematical logic (Carnielli et al., 2008). The following logics can be utilized for this purpose: *probabilistic (probability) logics*, *inconsistency-tolerant (paraconsistent) logics*, *hierarchical (sequential) logics*, and *spatial (locative) logics*. By combining and integrating these logics, we can extend and refine the existing standard model-checking framework. Model checking has been extended to *probabilistic model checking* (Aziz

Kamide, N. and Bernal, J.
Towards Locative Inconsistency-tolerant Hierarchical Probabilistic CTL Model Checking: Survey and Future Work.
DOI: 10.5220/0007683808690878
In *Proceedings of the 11th International Conference on Agents and Artificial Intelligence (ICAART 2019)*, pages 869-878
ISBN: 978-989-758-350-6

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).

Probabilistic temporal, inconsistency-tolerant (paraconsistent) temporal, and hierarchical (sequential) temporal logics and their applications to probabilistic, inconsistency-tolerant, and hierarchical model checking, respectively, have been studied by many researchers. In the following section, we present a survey of these previously proposed logics and their applications in model checking. However, this survey is not comprehensive; information on probabilistic temporal logics and their model checking applications in greater detail can be found in (Hanson, 1994; Hansson and Jonsson, 1994; Aziz et al., 1995; Bianco and de Alfaro, 1995; Baier and Kwiatkowska, 1998; Ognjanovic et al., 2011; Kwiatkowska et al., 2011; Ognjanovic et al., 2012; Kamide and Koizumi, 2016; Baier et al., 2018). For inconsistency-tolerant temporal logics and their model checking applications, information can be found in greater detail in (Easterbrook and Chechik, 2001; Chen and Wu, 2006; Kamide, 2006a; Kamide and Wansing, 2011; Kamide and Kaneiwa, 2010; Kaneiwa and Kamide, 2011b; Kamide, 2015; Kamide and Koizumi, 2016; Kamide and Endo, 2018). For hierarchical temporal logics and their model checking applications, information can be found in greater detail in (Kamide and Kaneiwa, 2009; Kaneiwa and Kamide, 2010; Kaneiwa and Kamide, 2011a; Kamide, 2015; Kamide and Yano, 2017; Kamide, 2018).

The contents of this paper are then summarized as follows. In Section 2, a survey of previously proposed typical extended temporal logics and their applications to model checking is performed. In Section 3, the logic LIHpCTL is developed by extending several existing typical extensions of CTL. In Section 4, an illustrative example of model checking based on LIHpCTL is presented. Section 5 presents the conclusion of this study with some remarks.

2 EXISTING LOGICS

2.1 Probabilistic Temporal Logics

In comparison with the standard non-probabilistic temporal logics CTL*, CTL, and LTL, probabilistic temporal logics can be effectively used in randomized and stochastic situations. Thus, many studies regarding probabilistic temporal logics and their applications, including probabilistic model checking, have been performed as discussed below. More information can be found in (Hanson, 1994; Hansson and Jonsson, 1994; Aziz et al., 1995; Bianco and de Alfaro, 1995; Baier and Kwiatkowska, 1998; Ognjanovic et al., 2011; Kwiatkowska et al., 2011; Ognjanovic et al., 2012; Kamide and Koizumi, 2016; Baier et al., 2018).

Probabilistic full computation tree logic (pCTL*) and its subsystem, *probabilistic computation tree logic* (pCTL), have been investigated by Aziz et al. (Aziz et al., 1995) and Bianco and de Alfaro (Bianco and de Alfaro, 1995). The logics pCTL* and pCTL are obtained from CTL* and CTL, respectively, by adding the probabilistic or probability operator, $P_{\geq x}$. Formulas with the form $P_{\geq x}\alpha$ can be interpreted as follows: the probability that α holds in the future evolution of the system is at least x. In (Bianco and de Alfaro, 1995), pCTL* and pCTL are introduced to verify the reliability and performance of systems modeled by discrete Markov chains. These logics can appropriately express the quantitative bounds on the probability of system evolutions. The complexities of model checking algorithms with respect to the logics are clarified in (Bianco and de Alfaro, 1995). In (Aziz et al., 1995), model-checking algorithms for various extensions of the previous settings of the logics are proposed to verify the probabilistic non-deterministic concurrent systems in which probabilistic behavior coexists with non-determinism. Further, these algorithms are shown to exhibit polynomial-time complexity depending on the sizes of the systems. The main difference between the pCTL* settings by Aziz et al. (Aziz et al., 1995) and those by Bianco and de Alfaro (Bianco and de Alfaro, 1995) is the probability measure settings in the probabilistic Kripke structure of pCTL*.

In (Hansson and Jonsson, 1994), PCTL, a probabilistic and real-time extension of CTL, is investigated based on an interpretation of discrete time Markov chains. In contrast to the probabilistic frameworks of pCTL and pCTL*, the notion of probability in PCTL is assigned to all of its temporal operators. For example, a PCTL formula with the form $G^{\leq t}_{\geq p}\alpha$ implies that α holds continuously for t time units with a probability of at least p. In (Hanson, 1994), a *timed probabilistic concurrent computation tree logic* (TPCTL) is introduced and investigated. In (Baier and Kwiatkowska, 1998), *probabilistic branching time logics* (PBTL and PBTL*) are introduced based on the probabilistic temporal logics TPCTL, PCTL, pCTL, and pCTL*, and model-checking algorithms based on PBTL and PBTL* are proposed to automatically verify randomized distributed systems. In (Ognjanovic et al., 2011), a *propositional discrete probabilistic branching temporal logic* (pBTL) is developed by extending CTL*. There are two types of novel probability operators in pBTL: $\mathrm{P}^{p}_{\geq r}$ and $\mathrm{P}^{s}_{\geq r}$, where $\mathrm{P}^{p}_{\geq r}\alpha$ implies that the probability that α holds on a randomly chosen branch is at least γ, and $\mathrm{P}^{s}_{\geq r}\alpha$ means that the probability that α holds on a particular branch is at least γ. In (Ognjanovic et al., 2012), a propositional probabilistic logic with discrete linear time is developed to handle the clinical reasoning with respect to evidence. More recent developments in probabilistic model checking based on probabilistic temporal logics are reported in (Baier et al., 2018).

2.2 Inconsistency-tolerant Temporal Logics

In comparison with the standard non-paraconsistent temporal logics CTL*, CTL, and LTL, inconsistency-tolerant (paraconsistent) logics can be appropriately used in inconsistency-tolerant situations (Priest, 2002; da Costa et al., 1995; Wansing, 1993). Typical examples of non-temporal paraconsistent logics include *Belnap and Dunn's useful four-valued logic* (Belnap, 1977b; Belnap, 1977a; Dunn, 1976) and *Nelson's paraconsistent four-valued logic* (Almukdad and Nelson, 1984; Nelson, 1949). Combining these logics with CTL*, CTL, and LTL has led to the introduction of various inconsistency-tolerant temporal logics, and inconsistency-tolerant versions (extensions) of CTL, CTL*, and LTL have been developed by many researchers.

The *multi-valued computation tree logic* (χCTL) was introduced by Easterbrook and Chechik (Easterbrook and Chechik, 2001) as the base logic for *multi-valued model checking*, the first framework for inconsistency-tolerant model checking. The *quasi-classical temporal logic* (QCTL) was introduced by Chen and Wu (Chen and Wu, 2006) to verify inconsistent concurrent systems using inconsistency-tolerant model checking. The *paraconsistent full computation tree logic* (4CTL*) was proposed by Kamide (Kamide, 2006a) to obtain a logical foundation for inconsistency-tolerant model checking. The *paraconsistent linear-time temporal logic* (PLTL) was introduced by Kamide and Wansing (Kamide and Wansing, 2011) to obtain a cut-free and complete Gentzen-type sequent calculus. The alternative paraconsistent computation tree logic (PCTL) was introduced by Kamide and Kaneiwa (Kamide and Kaneiwa, 2010; Kaneiwa and Kamide, 2011b) and offered an alternative inconsistency-tolerant model checking framework. Kamide (Kamide, 2015) introduced the *sequence-indexed paraconsistent computation tree logic* (SPCTL), which can be obtained from the CTL by adding the paraconsistent negation connective, $\sim$, and a sequence modal operator, $[b]$. This logic was used to verify clinical reasoning through inconsistency-tolerant hierarchical model checking.

Kamide and Endo (Kamide and Endo, 2018) recently developed pCTL and pLTL, which are alternative refined versions of the paraconsistent temporal logics PCTL and PLTL, respectively, to obtain various simple and efficient translation methods for inconsistency-tolerant model checking. These alternative refined logics adopt a simple single-satisfaction relation. The alternative version of PCTL is pCTL, which has the same name as the probabilistic computation tree logic. Whereas PLTL (Kamide and Wansing, 2011), PCTL (Kamide and Kaneiwa, 2010; Kaneiwa and Kamide, 2011b), 4CTL* (Kamide, 2006a), and SPCTL (Kamide, 2015) have two types of dual-satisfaction relations, $\models^+$ (verification or justification) and $\models^-$ (refutation or falsification), pCTL and pLTL (Kamide and Endo, 2018) have a simple single-satisfaction relation, $\models^*$, which is highly compatible with the standard single-satisfaction relations of LTL, CTL, and CTL*. The single-satisfaction relation can provide simple proofs for embedding theorems, and the connective $\sim$ can be formalized simply and handled uniformly.

To deal with open, large, and complex concurrent systems, such as cloud-based systems and web application systems, it is necessary to have extended logics with model checking frameworks that can also simultaneously handle both inconsistency-tolerant and probabilistic reasoning. To this end, in (Kamide and Koizumi, 2015; Kamide and Koizumi, 2016), a partial solution was obtained (i.e., an inconsistency-tolerant extension of the probabilistic computation tree logic, pCTL, was developed). The *paraconsistent probabilistic computation tree logic* (PpCTL), which can be obtained from the probabilistic computation tree logic, pCTL, by adding a paraconsistent negation connective $\sim$, was constructed in (Kamide and Koizumi, 2015; Kamide and Koizumi, 2016) based on a probability-measure-independent translation of PpCTL into pCTL. The theorem for embedding PpCTL into pCTL was proven using this translation and entails the relative decidability of PpCTL with respect to pCTL (i.e., the decidability of pCTL implies the decidability of PpCTL). This result indicates that we can reuse the existing pCTL-based model-checking algorithms by Aziz et al. (Aziz et al., 1995) and Bianco and de Alfaro (Bianco and de Alfaro, 1995) for the PpCTL-based model-checking algorithms.

2.3 Hierarchical Temporal Logics

In what follows, we present a survey of various hierarchical (sequential) temporal logics and their applications. Various extended temporal logics employing the sequence (hierarchical) modal operator $[b]$, where b represents a sequence of symbols, have been investigated to handle hierarchical information (Kamide and Kaneiwa, 2009; Kaneiwa and Kamide, 2010; Kaneiwa and Kamide, 2011a; Kamide, 2015; Kamide and Yano, 2017; Kamide, 2018).

The sequence 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 with 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).

Kamide and Kaneiwa (Kamide and Kaneiwa, 2009; Kaneiwa and Kamide, 2011a) introduced CTLS*, which is an extension of CTL*, by adding $[b]$ to CTL*. Similarly, the *sequence-indexed linear-time temporal logic* (SLTL), which is an extension of LTL, was introduced in (Kaneiwa and Kamide, 2010) by adding $[b]$ to LTL. A proof system for SLTL was developed to verify certain specifications of secure authentication systems. The *sequence-indexed paraconsistent computation-tree logic* (SPCTL), which is an extension of CTL, was introduced by Kamide in (Kamide, 2015) by adding $[b]$ and $\sim$ to CTL. As explained above, SPCTL was used to verify clinical reasoning through inconsistency-tolerant hierarchical model checking. In (Kamide and Yano, 2017; Kamide, 2018), the*sequential linear-time temporal logic* (sLTL) and *sequential computation-tree logic* (sCTL) were developed by extending LTL and CTL, respectively, to obtain the logical foundation of hierarchical model checking.

The hierarchical (sequential) temporal logics sCTL and sLTL proposed by Kamide and Yano in (Kamide and Yano, 2017; Kamide, 2018) adopted a simple single-satisfaction relation to obtain various simple and efficient translation methods for hierarchical model checking. Although the previously proposed logics, CTLS* (Kamide and Kaneiwa, 2009; Kaneiwa and Kamide, 2011a), SLTL (Kaneiwa and Kamide, 2010), and SPCTL (Kamide, 2015), have complex multiple sequence-indexed satisfaction relations, $\models^{\hat{d}}$, sLTL and sCTL (Kamide and Yano, 2017; Kamide, 2018) have a simple single-satisfaction relation, $\models^{\star}$, which is highly compatible with the standard single-satisfaction relations of LTL, CTL, and CTL*. Using this simple satisfaction relation, embedding theorems can be easily proven, and the operator $[b]$ can be formalized simply and handled uniformly.

2.4 Locative Temporal Logics

Future studies should aim to extend the previously proposed standard and extended temporal logics by adding a location (locative) operator, $[l_i]$. By adding this operator to previously proposed logics, locative (spatial) properties within the model checking framework can be appropriately handled. The location operator was originally introduced by Kobayashi et al. in (N. Kobayashi and Yonezawa, 1999) using a structural congruence relation in the formalization of a distributed concurrent linear logic programming language. The location operator was also reformulated by Kamide in (Kamide, 2005; Kamide, 2006b) and the reformulated operator was a variant of the original setting by Kobayashi et al. (N. Kobayashi and Yonezawa, 1999). In fact, the framework of the original operator was improved in (Kamide, 2005; Kamide, 2006b) as a purely logical formulation without any structural congruence relations. Formulas with the form $[l_i]\alpha$ can be interpreted as follows: proposition α holds at location l_i.

Various locative (spatial) temporal logics employing the location operator $[l_i]$ have been investigated to handle distributed concurrent systems. For example, the *locative paraconsistent probabilistic computation tree logic* (LPpCTL) was introduced by Kamide and Koizumi in (Kamide and Koizumi, 2016) by adding $[l_i]$ to the inconsistency-tolerant (paraconsistent) probabilistic computation tree logic, PpCTL. By integrating the findings of the above-mentioned studies with the findings in (Kamide and Koizumi, 2016), we intend to establish a logical foundation for the proposed framework of locative inconsistency-tolerant hierarchical probabilistic model checking based on LIHpCTL employing $[l_i]$.

Assuming a space domain *Loc* and operator $[l]$ with $l \in Loc$, the satisfaction relation $(s,l) \models^\star \alpha$ of LPpCTL can be interpreted as follows: proposition α holds at time state s and location l. Then, various properties and situations with space and time can be expressed using LPpCTL formulas. The following liveness property is an example: "If we input the login password of host computer Comp3 at one of the mobile computers Comp1 and Comp2, then we will eventually be able to log in to Comp3." This can be formally expressed as $\mathrm{AG}([comp1]password \vee [comp2]password \to \mathrm{EF}[comp3]login)$, where the space domain, *Loc*, is $\{comp1, comp2, comp3\}$.

3 PROPOSED LOGIC

Formulas of *locative inconsistency-tolerant hierarchical probabilistic computation tree logic* (LIHpCTL) are constructed from countably many propositional variables, $\to$ (implication) $\wedge$ (conjunction), $\vee$ (disjunction), $\neg$ (classical negation), $\sim$ (paraconsistent negation), X (next), G (globally), F (eventually), U (until), R (release), A (all computation paths), 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), $[l_i]$ (location operator), and $[b]$ (hierarchical operator or sequence modal operator) where b is a sequence. *Sequences* are constructed from atomic sequences, $\emptyset$ (empty sequence) and ; (composition).

The set of sequences (including the empty sequence $\emptyset$) is denoted as SE. Lower-case letters $b, c, ...$ are used to denote sequences. An expression $[\emptyset]\alpha$ means α, and expressions $[\emptyset\ ;\ b]\alpha$ and $[b\ ;\ \emptyset]\alpha$ mean $[b]\alpha$. The symbol Φ is used to denote a non-empty set of propositional variables, the symbol $\Phi^\sim$ is used to denote the set $\{\sim p \mid p \in \Phi\}$, and the symbol $\Phi^{\sim[d]}$ is used to denote the set $\{[d]\gamma \mid \gamma \in \Phi \cup \Phi^\sim\}$.

Definition 3.1. *Let Loc be a finite non-empty set of locations, and assume $l \in Loc$, and $x \in [0,1]$. Formulas α and sequences b of* LIHpCTL *are defined by the following grammar, assuming p and e represent propositional variables and atomic sequences, respectively:*

$$\begin{aligned}\alpha ::= {} & p \mid \alpha \wedge \alpha \mid \alpha \vee \alpha \mid \alpha \to \alpha \mid \neg\alpha \mid \sim\alpha \mid \\ & \mathrm{AX}\alpha \mid \mathrm{EX}\alpha \mid \mathrm{AG}\alpha \mid \mathrm{EG}\alpha \mid \mathrm{AF}\alpha \mid \mathrm{EF}\alpha \mid \\ & \mathrm{A}(\alpha \mathrm{U} \alpha) \mid \mathrm{E}(\alpha \mathrm{U} \alpha) \mid \mathrm{A}(\alpha \mathrm{R} \alpha) \mid \mathrm{E}(\alpha \mathrm{R} \alpha) \mid \\ & \mathrm{P}_{\leq x}\alpha \mid \mathrm{P}_{\geq x}\alpha \mid \mathrm{P}_{<x}\alpha \mid \mathrm{P}_{>x}\alpha \mid [l]\alpha \mid [b]\alpha. \\ b ::= {} & e \mid \emptyset \mid b\ ;\ b.\end{aligned}$$

An expression $\overline{[d]}$ is used to represent $[d_0][d_1]\cdots[d_i]$ 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.

Definition 3.2. *A structure $(Loc, S, S_0, R, \mu_s, L^\star)$ is a* locative inconsistency-tolerant hierarchical probabilistic model *iff*

1. *Loc is a finite non-empty set of locations,*
2. *S is the set of states,*
3. *S_0 is a set of initial states and $S_0 \subseteq S$,*
4. *R is a binary relation on S which satisfies the condition: $\forall s \in S\ \exists s' \in S\ [(s,s') \in R]$,*
5. *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 $[0,1]$,*
6. *$L^\star$ is a mapping from S to the power set of $\bigcup_{d \in \mathrm{SE}} \Phi^{\sim[d]}$.*

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

Definition 3.3. (LIHpCTL). *A* locative inconsistency-tolerant hierarchical probabilistic satisfaction relation $(M, \langle s,l \rangle) \models^\star \alpha$ *for any formula α, where M is a locative inconsistency-tolerant hierarchical probabilistic model $(Loc, S, S_0, R, \mu_s, L^\star)$, s represents a state in S, and l represents a location in Loc, is defined by:*

1. *for any $\gamma \in \Phi^{\sim[d]}$ $(d \in \mathrm{SE})$, $(M, \langle s,l \rangle) \models^\star \gamma$ iff $\gamma \in L^\star(s)$,*
2. *for any $p \in \Phi$, $(M, \langle s,l \rangle) \models^\star \overline{[d]}{\sim}p$ iff $(M, \langle s,l \rangle) \models^\star {\sim}\overline{[d]}p$,*
3. *$(M, \langle s,l \rangle) \models^\star \overline{[d]}[b]\alpha$ iff $(M, \langle s,l \rangle) \models^\star [d\ ;\ b]\alpha$,*
4. *$(M, \langle s,l \rangle) \models^\star \overline{[d]}(\alpha \wedge \beta)$ iff $(M, \langle s,l \rangle) \models^\star \overline{[d]}\alpha$ and $(M, \langle s,l \rangle) \models^\star \overline{[d]}\beta$,*
5. *$(M, \langle s,l \rangle) \models^\star \overline{[d]}(\alpha \vee \beta)$ iff $(M, \langle s,l \rangle) \models^\star \overline{[d]}\alpha$ or $(M, \langle s,l \rangle) \models^\star \overline{[d]}\beta$,*
6. *$(M, \langle s,l \rangle) \models^\star \overline{[d]}(\alpha \to \beta)$ iff $(M, \langle s,l \rangle) \models^\star \overline{[d]}\alpha$ implies $(M, \langle s,l \rangle) \models^\star \overline{[d]}\beta$,*
7. *$(M, \langle s,l \rangle) \models^\star \overline{[d]}\neg\alpha$ iff $(M, \langle s,l \rangle) \not\models^\star \overline{[d]}\alpha$,*
8. *for any $k \in Loc$, $(M, \langle s,l \rangle) \models^\star \overline{[d]}[k]\alpha$ iff $(M, \langle s,k \rangle) \models^\star \overline{[d]}\alpha$,*
9. *for any $x \in [0,1]$, $(M, \langle s,l \rangle) \models^\star \overline{[d]}\mathrm{P}_{\leq x}\alpha$ iff $\mu_s(\{w \in \Omega_s \mid (M, \langle s,l \rangle) \models^\star \overline{[d]}\alpha\}) \leq x$,*
10. *for any $x \in [0,1]$, $(M, \langle s,l \rangle) \models^\star \overline{[d]}\mathrm{P}_{\geq x}\alpha$ iff $\mu_s(\{w \in \Omega_s \mid (M, \langle s,l \rangle) \models^\star \overline{[d]}\alpha\}) \geq x$,*
11. *for any $x \in [0,1]$, $(M, \langle s,l \rangle) \models^\star \overline{[d]}\mathrm{P}_{<x}\alpha$ iff $\mu_s(\{w \in \Omega_s \mid (M, \langle s,l \rangle) \models^\star \overline{[d]}\alpha\}) < x$,*
12. *for any $x \in [0,1]$, $(M, \langle s,l \rangle) \models^\star \overline{[d]}\mathrm{P}_{>x}\alpha$ iff $\mu_s(\{w \in \Omega_s \mid (M, \langle s,l \rangle) \models^\star \overline{[d]}\alpha\}) > x$,*
13. *$(M, \langle s,l \rangle) \models^\star \overline{[d]}\mathrm{AX}\alpha$ iff $\forall s_1 \in S\ [(s,s_1) \in R$ implies $(M, \langle s_1,l \rangle) \models^\star \overline{[d]}\alpha]$,*

14. $(M,\langle s,l\rangle) \models^\star \overline{[d]}\mathrm{EX}\alpha$ *iff* $\exists s_1 \in S\ [(s,s_1) \in R$ *and* $(M,\langle s_1,l\rangle) \models^\star \overline{[d]}\alpha]$,
15. $(M,\langle s,l\rangle) \models^\star \overline{[d]}\mathrm{AG}\alpha$ *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* $(M,\langle s_i,l\rangle) \models^\star \overline{[d]}\alpha$,
16. $(M,\langle s,l\rangle) \models^\star \overline{[d]}\mathrm{EG}\alpha$ *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* $(M,\langle s_i,l\rangle) \models^\star \overline{[d]}\alpha$,
17. $(M,\langle s,l\rangle) \models^\star \overline{[d]}\mathrm{AF}\alpha$ *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* $(M,\langle s_i,l\rangle) \models^\star \overline{[d]}\alpha$,
18. $(M,\langle s,l\rangle) \models^\star \overline{[d]}\mathrm{EF}\alpha$ *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* $(M,\langle s_i,l\rangle) \models^\star \overline{[d]}\alpha$,
19. $(M,\langle s,l\rangle) \models^\star \overline{[d]}\mathrm{A}(\alpha\mathrm{U}\beta)$ *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* $(M,\langle s_j,l\rangle) \models^\star \overline{[d]}\beta$ *and* $\forall 0 \leq k < j$ $(M,\langle s_k,l\rangle) \models^\star \overline{[d]}\alpha$,
20. $(M,\langle s,l\rangle) \models^\star \overline{[d]}\mathrm{E}(\alpha\mathrm{U}\beta)$ *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* $(M,\langle s_j,l\rangle) \models^\star \overline{[d]}\beta$ *and* $\forall 0 \leq k < j$ $(M,\langle s_k,l\rangle) \models^\star \overline{[d]}\alpha$,
21. $(M,\langle s,l\rangle) \models^\star \overline{[d]}\mathrm{A}(\alpha\mathrm{R}\beta)$ *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* $(M,\langle s_j,l\rangle) \models^\star \overline{[d]}\beta$ *or* $\exists 0 \leq k < j$ $(M,\langle s_k,l\rangle) \models^\star \overline{[d]}\alpha$,
22. $(M,\langle s,l\rangle) \models^\star \overline{[d]}\mathrm{E}(\alpha\mathrm{R}\beta)$ *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* $(M,\langle s_j,l\rangle) \models^\star \overline{[d]}\beta$ *or* $\exists 0 \leq k < j$ $(M,\langle s_k,l\rangle) \models^\star \overline{[d]}\alpha$,
23. $(M,\langle s,l\rangle) \models^\star \overline{[d]}{\sim}{\sim}\alpha$ *iff* $(M,\langle s,l\rangle) \models^\star \overline{[d]}\alpha$,
24. $(M,\langle s,l\rangle) \models^\star \overline{[d]}{\sim}[b]\alpha$ *iff* $(M,\langle s,l\rangle) \models^\star [d\ ;b]{\sim}\alpha$,
25. $(M,\langle s,l\rangle) \models^\star \overline{[d]}{\sim}(\alpha\wedge\beta)$ *iff* $(M,\langle s,l\rangle) \models^\star \overline{[d]}{\sim}\alpha$ *or* $(M,\langle s,l\rangle) \models^\star \overline{[d]}{\sim}\beta$,
26. $(M,\langle s,l\rangle) \models^\star \overline{[d]}{\sim}(\alpha\vee\beta)$ *iff* $(M,\langle s,l\rangle) \models^\star \overline{[d]}{\sim}\alpha$ *and* $(M,\langle s,l\rangle) \models^\star \overline{[d]}{\sim}\beta$,
27. $(M,\langle s,l\rangle) \models^\star \overline{[d]}{\sim}(\alpha{\to}\beta)$ *iff* $(M,\langle s,l\rangle) \not\models^\star \overline{[d]}{\sim}\alpha$ *and* $(M,\langle s,l\rangle) \models^\star \overline{[d]}{\sim}\beta$,
28. $(M,\langle s,l\rangle) \models^\star \overline{[d]}{\sim}\neg\alpha$ *iff* $(M,\langle s,l\rangle) \not\models^\star \overline{[d]}{\sim}\alpha$,
29. *for any* $k \in Loc$, $(M,\langle s,l\rangle) \models^\star \overline{[d]}{\sim}[k]\alpha$ *iff* $(M,\langle s,k\rangle) \models^\star \overline{[d]}{\sim}\alpha$,
30. *for any* $x \in [0,1]$, $(M,\langle s,l\rangle) \models^\star \overline{[d]}{\sim}\mathrm{P}_{\leq x}\alpha$ *iff* $\mu_s(\{w \in \Omega_s \mid (M,\langle s,l\rangle) \models^\star \overline{[d]}{\sim}\alpha\}) > x$,
31. *for any* $x \in [0,1]$, $(M,\langle s,l\rangle) \models^\star \overline{[d]}{\sim}\mathrm{P}_{\geq x}\alpha$ *iff* $\mu_s(\{w \in \Omega_s \mid (M,\langle s,l\rangle) \models^\star \overline{[d]}{\sim}\alpha\}) < x$,
32. *for any* $x \in [0,1]$, $(M,\langle s,l\rangle) \models^\star \overline{[d]}{\sim}\mathrm{P}_{<x}\alpha$ *iff* $\mu_s(\{w \in \Omega_s \mid (M,\langle s,l\rangle) \models^\star \overline{[d]}{\sim}\alpha\}) \geq x$,
33. *for any* $x \in [0,1]$, $(M,\langle s,l\rangle) \models^\star \overline{[d]}{\sim}\mathrm{P}_{>x}\alpha$ *iff* $\mu_s(\{w \in \Omega_s \mid (M,\langle s,l\rangle) \models^\star \overline{[d]}{\sim}\alpha\}) \leq x$,
34. $(M,\langle s,l\rangle) \models^\star \overline{[d]}{\sim}\mathrm{AX}\alpha$ *iff* $\exists s_1 \in S\ [(s,s_1) \in R$ *and* $(M,\langle s_1,l\rangle) \models^\star \overline{[d]}{\sim}\alpha]$,
35. $(M,\langle s,l\rangle) \models^\star \overline{[d]}{\sim}\mathrm{EX}\alpha$ *iff* $\forall s_1 \in S\ [(s,s_1) \in R$ *implies* $(M,\langle s_1,l\rangle) \models^\star \overline{[d]}{\sim}\alpha]$,
36. $(M,\langle s,l\rangle) \models^\star \overline{[d]}{\sim}\mathrm{AG}\alpha$ *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* $(M,\langle s_i,l\rangle) \models^\star \overline{[d]}{\sim}\alpha$,
37. $(M,\langle s,l\rangle) \models^\star \overline{[d]}{\sim}\mathrm{EG}\alpha$ *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* $(M,\langle s_i,l\rangle) \models^\star \overline{[d]}{\sim}\alpha$,
38. $(M,\langle s,l\rangle) \models^\star \overline{[d]}{\sim}\mathrm{AF}\alpha$ *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* $(M,\langle s_i,l\rangle) \models^\star \overline{[d]}{\sim}\alpha$,
39. $(M,\langle s,l\rangle) \models^\star \overline{[d]}{\sim}\mathrm{EF}\alpha$ *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* $(M,\langle s_i,l\rangle) \models^\star \overline{[d]}{\sim}\alpha$,
40. $(M,\langle s,l\rangle) \models^\star \overline{[d]}{\sim}\mathrm{A}(\alpha\mathrm{U}\beta)$ *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* $(M,\langle s = j,l\rangle) \models^\star \overline{[d]}{\sim}\beta$ *or* $\exists 0 \leq k < j$ $(M,\langle s_k,l\rangle) \models^\star \overline{[d]}{\sim}\alpha$,
41. $(M,\langle s,l\rangle) \models^\star \overline{[d]}{\sim}\mathrm{E}(\alpha\mathrm{U}\beta)$ *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* $(M,\langle s_j,l\rangle) \models^\star \overline{[d]}{\sim}\beta$ *or* $\exists 0 \leq k < j$ $(M,\langle s_k,l\rangle) \models^\star \overline{[d]}{\sim}\alpha$,
42. $(M,\langle s,l\rangle) \models^\star \overline{[d]}{\sim}\mathrm{A}(\alpha\mathrm{R}\beta)$ *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* $(M,\langle s_j,l\rangle) \models^\star \overline{[d]}{\sim}\beta$ *and* $\forall 0 \leq k < j$ $(M,\langle s_k,l\rangle) \models^\star \overline{[d]}{\sim}\alpha$,
43. $(M,\langle s,l\rangle) \models^\star \overline{[d]}{\sim}\mathrm{E}(\alpha\mathrm{R}\beta)$ *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* $(M,\langle s_j,l\rangle) \models^\star \overline{[d]}{\sim}\beta$ *and* $\forall 0 \leq k < j$ $(M,\langle s_k,l\rangle) \models^\star \overline{[d]}{\sim}\alpha$,
44. $(M,\langle s,l\rangle) \models^\star {\sim}\overline{[d]}{\sim}\alpha$ *iff* $(M,\langle s,l\rangle) \models^\star \overline{[d]}\alpha$,
45. $(M,\langle s,l\rangle) \models^\star {\sim}\overline{[d]}[b]\alpha$ *iff* $(M,\langle s,l\rangle) \models^\star {\sim}[d\ ;b]\alpha$,
46. $(M,\langle s,l\rangle) \models^\star {\sim}\overline{[d]}(\alpha\wedge\beta)$ *iff* $(M,\langle s,l\rangle) \models^\star {\sim}\overline{[d]}\alpha$ *or* $(M,\langle s,l\rangle) \models^\star {\sim}\overline{[d]}\beta$,
47. $(M,\langle s,l\rangle) \models^\star {\sim}\overline{[d]}(\alpha\vee\beta)$ *iff* $(M,\langle s,l\rangle) \models^\star {\sim}\overline{[d]}\alpha$ *and* $(M,\langle s,l\rangle) \models^\star {\sim}\overline{[d]}\beta$,
48. $(M,\langle s,l\rangle) \models^\star {\sim}\overline{[d]}(\alpha{\to}\beta)$ *iff* $(M,\langle s,l\rangle) \not\models^\star {\sim}\overline{[d]}\alpha$ *and* $(M,\langle s,l\rangle) \models^\star {\sim}\overline{[d]}\beta$,
49. $(M,\langle s,l\rangle) \models^\star {\sim}\overline{[d]}\neg\alpha$ *iff* $(M,\langle s,l\rangle) \not\models^\star {\sim}\overline{[d]}\alpha$,
50. *for any* $k \in Loc$, $(M,\langle s,l\rangle) \models^\star {\sim}\overline{[d]}[k]\alpha$ *iff* $(M,\langle s,k\rangle) \models^\star {\sim}\overline{[d]}\alpha$,
51. *for any* $x \in [0,1]$, $(M,\langle s,l\rangle) \models^\star {\sim}\overline{[d]}\mathrm{P}_{\leq x}\alpha$ *iff* $\mu_s(\{w \in \Omega_s \mid (M,\langle s,l\rangle) \models^\star {\sim}\overline{[d]}\alpha\}) > x$,
52. *for any* $x \in [0,1]$, $(M,\langle s,l\rangle) \models^\star {\sim}\overline{[d]}\mathrm{P}_{\geq x}\alpha$ *iff* $\mu_s(\{w \in \Omega_s \mid (M,\langle s,l\rangle) \models^\star {\sim}\overline{[d]}\alpha\}) < x$,
53. *for any* $x \in [0,1]$, $(M,\langle s,l\rangle) \models^\star {\sim}\overline{[d]}\mathrm{P}_{<x}\alpha$ *iff* $\mu_s(\{w \in \Omega_s \mid (M,\langle s,l\rangle) \models^\star {\sim}\overline{[d]}\alpha\}) \geq x$,
54. *for any* $x \in [0,1]$, $(M,\langle s,l\rangle) \models^\star {\sim}\overline{[d]}\mathrm{P}_{>x}\alpha$ *iff* $\mu_s(\{w \in \Omega_s \mid (M,\langle s,l\rangle) \models^\star {\sim}\overline{[d]}\alpha\}) \leq x$,
55. $(M,\langle s,l\rangle) \models^\star {\sim}\overline{[d]}\mathrm{AX}\alpha$ *iff* $\exists s_1 \in S\ [(s,s_1) \in R$ *and* $(M,\langle s_1,l\rangle) \models^\star {\sim}\overline{[d]}\alpha]$,

56. $(M,\langle s,l\rangle) \models^\star \sim\overline{[d]}\mathrm{EX}\alpha$ *iff* $\forall s_1 \in S\ [(s,s_1) \in R$ *implies* $(M,\langle s_1,l\rangle) \models^\star \sim\overline{[d]}\alpha]$,
57. $(M,\langle s,l\rangle) \models^\star \sim\overline{[d]}\mathrm{AG}\alpha$ *iff there is a path* $\pi \equiv s_0,s_1,s_2,...,$ *where* $s \equiv s_0$, *and for some state* s_i *along* π, *we have* $(M,\langle s_i,l\rangle) \models^\star \sim\overline{[d]}\alpha$,
58. $(M,\langle s,l\rangle) \models^\star \sim\overline{[d]}\mathrm{EG}\alpha$ *iff for all paths* $\pi \equiv s_0,s_1,s_2,...,$ *where* $s \equiv s_0$, *there is a state* s_i *along* π *such that* $(M,\langle s_i,l\rangle) \models^\star \sim\overline{[d]}\alpha$,
59. $(M,\langle s,l\rangle) \models^\star \sim\overline{[d]}\mathrm{AF}\alpha$ *iff there is a path* $\pi \equiv s_0,s_1,s_2,...,$ *where* $s \equiv s_0$, *and for all states* s_i *along* π, *we have* $(M,\langle s_i,l\rangle) \models^\star \sim\overline{[d]}\alpha$,
60. $(M,\langle s,l\rangle) \models^\star \sim\overline{[d]}\mathrm{EF}\alpha$ *iff for all paths* $\pi \equiv s_0,s_1,s_2,...,$ *where* $s \equiv s_0$, *and all states* s_i *along* π, *we have* $(M,\langle s_i,l\rangle) \models^\star \sim\overline{[d]}\alpha$,
61. $(M,\langle s,l\rangle) \models^\star \sim\overline{[d]}\mathrm{A}(\alpha\mathrm{U}\beta)$ *iff there is a path* $\pi \equiv s_0,s_1,s_2,...,$ *where* $s \equiv s_0$, *and for all states* s_j *along* π, *we have* $(M,\langle s_j,l\rangle) \models^\star \sim\overline{[d]}\beta$ *or* $\exists 0 \le k < j$ $(M,\langle s_k,l\rangle) \models^\star \sim\overline{[d]}\alpha$,
62. $(M,\langle s,l\rangle) \models^\star \sim\overline{[d]}\mathrm{E}(\alpha\mathrm{U}\beta)$ *iff for all paths* $\pi \equiv s_0,s_1,s_2,...,$ *where* $s \equiv s_0$, *and all states* s_j *along* π, *we have* $(M,\langle s_j,l\rangle) \models^\star \sim\overline{[d]}\beta$ *or* $\exists 0 \le k < j$ $(M,\langle s_k,l\rangle) \models^\star \sim\overline{[d]}\alpha$,
63. $(M,\langle s,l\rangle) \models^\star \sim\overline{[d]}\mathrm{A}(\alpha\mathrm{R}\beta)$ *iff there is a path* $\pi \equiv s_0,s_1,s_2,...,$ *where* $s \equiv s_0$, *and for some state* s_j *along* π, *we have* $(M,\langle s_j,l\rangle) \models^\star \sim\overline{[d]}\beta$ *and* $\forall 0 \le k < j$ $(M,\langle s_k,l\rangle) \models^\star \sim\overline{[d]}\alpha$,
64. $(M,\langle s,l\rangle) \models^\star \sim\overline{[d]}\mathrm{E}(\alpha\mathrm{R}\beta)$ *iff for all paths* $\pi \equiv s_0,s_1,s_2,...,$ *where* $s \equiv s_0$, *there is a state* s_j *along* π *such that* $(M,\langle s_j,l\rangle) \models^\star \sim\overline{[d]}\beta$ *and* $\forall 0 \le k < j$ $(M,\langle s_k,l\rangle) \models^\star \sim\overline{[d]}\alpha$.

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

Remark 3.4.

1. LIHpCTL *is regarded as an extension of the following temporal logics: probabilistic CTL (called* pCTL*) studied in* (Aziz et al., 1995; Bianco and de Alfaro, 1995), *inconsistency-tolerant CTL (also called* pCTL, *but different from the above-mentioned probabilistic one) proposed in* (Kamide and Endo, 2018)*, and hierarchical CTL (called* sCTL*) proposed in* (Kamide and Yano, 2017; Kamide, 2018).
2. *There are some possibilities 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* $\mathcal{B}_s$*, were proposed in* (Bianco and de Alfaro, 1995). *A probability measure* μ^s*, which is concerned with some* discrete Markov processes*, was proposed in* (Aziz et al., 1995).
3. *The model checking problem for the probabilistic CTL with the probability measures* μ_s^+ *and* μ_s^- *was shown to be decidable in* (Bianco and de Alfaro, 1995)*. The model checking problem for the probabilistic CTL with the probability measure* μ^s *was shown to be decidable in* (Aziz et al., 1995).
4. *The setting of the conditions concerning the negated implication and negated negation in* LIHpCTL *is based on the axiom schemes* $\sim(\alpha\to\beta) \leftrightarrow \neg\sim\alpha \wedge \sim\beta$ *and* $\sim\neg\alpha \leftrightarrow \neg\sim\alpha$*. These axiom schemes were originally introduced by De and Omori in* (De and Omori, 2015) *wherein these are shown to be natural and plausible from the point of view of many-valued semantics.*
5. *The setting of the location operator is the same as that of the locative paraconsistent probabilistic computation tree logic (called* LPpCTL*) introduced in* (Kamide and Koizumi, 2016)*. The decidability of the model checking problem for* LPpCTL *has not yet been proved.*
6. *The single-satisfaction relation* $\models^\star$ *of* LIHpCTL *is highly compatible with the standard single-satisfaction relations of* CTL*. By using this satisfaction relation, we can simply formalize and uniformly handle* $\sim$ *and* $[b]$.

We then have the following conjecture.

Conjecture 3.5 (Decidability)**.**

1. *(Relative decidability for* LIHpCTL*): If the model-checking, validity, and satisfiability problems for the purely probabilistic fragment of* LIHpCTL *(i.e., it is obtained from* CTL *by adding the probability operators) with a certain probability measure are decidable, then the model-checking, validity, and satisfiability problems for* LIHpCTL *with the same probability measure as that of the fragment are also decidable.*
2. *(Decidability for the probability-free fragment of* LIHpCTL*): The model-checking, validity, and satisfiability problems for the probability-operator-free fragment of* LIHpCTL *(i.e., it is obtained from* LIHpCTL *by deleting the probability operators) is decidable.*

4 ILLUSTRATIVE EXAMPLE

In what follows, we present an illustrative example of verifying the reasoning process behind diagnosing multiple sclerosis (MS), a rare disease, using the proposed LIHpCTL-based model checking. The aim is to verify mission-critical clinical reasoning with disease ontology using an extended formal method, such

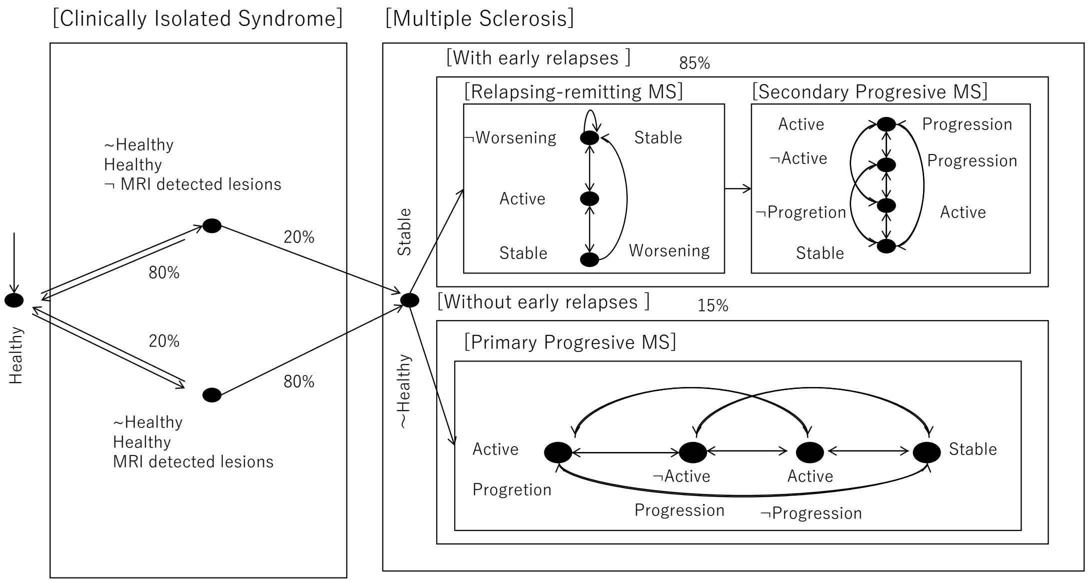

Figure 1: An ontological reasoning process model for multiple sclerosis.

as that in our proposal. We consider the scenario presented in Figure 1, in which a person suffers from MS. The cause of this disease is not clear and there is no known cure for it. For more information about MS, see (Wikipedia, 2018) and the references therein.

MS is more common in people who live farther from the equator; in particular, it is more common in regions with northern European populations. Thus, the location operator $[l_i]$ in LIHpCTL can be effectively used to describe this fact. Furthermore, MS is typically diagnosed based on presenting signs and symptoms in conjunction with supporting medical imaging based on Magnetic Resonance Imaging (MRI) results and laboratory testing. It can be difficult to confirm, especially in early stages because the signs and symptoms can be similar to those of other medical conditions. Thus, the inconsistency-tolerant (paraconsistent) negation connective $\sim$ in LIHpCTL can be effectively used to describe such symptoms and a patient's health. For example, if it cannot be determined whether someone is healthy, then an ambiguous concept, *healthy*, can be represented by asserting the inconsistent formula $healthy \wedge \sim healthy$. This is well formalized because $(healthy \wedge \sim healthy) \rightarrow \perp$ is not valid in LHIpCTL.

The United States National Multiple Sclerosis Society and the Multiple Sclerosis International Federation describe four types of MS as follows: clinically isolated syndrome (CIS), relapsing-remitting MS (RRMS), primary progressive MS (PPMS), and secondary progressive MS (SPMS). The hierarchy of these types can be effectively described by the sequence modal operator $[b]$ in LIHpCTL.

RRMS is characterized by unpredictable relapses followed by periods of months to years of remission with no new signs of disease activity. Deficits that occur during attacks may either resolve or leave permanent damage, the latter being the case in approximately 40% of attacks and being more common the longer a person has the disease. This process describes the initial course of 80% of individuals with MS, and it usually begins with CIS. In CIS, a person has an attack suggestive of demyelination but does not fulfill the criteria for multiple sclerosis. Of the individuals experiencing CIS, 30–70% later develop MS. PPMS occurs in approximately 10–20% of individuals, with no remission after the initial symptoms. SPMS occurs in approximately 65% of individuals with initial RRMS and involves a progressive neurologic decline between acute attacks without any definite periods of remission. These probabilistic phenomena concerning MS can be effectively expressed by the probabilistic operators in LIHpCTL.

We can verify the following statements using LIHpCTL:

1. "Is there a state in which a person is both healthy and unhealthy with CIS?"
2. "Is there a state in which a person is both healthy and unhealthy with RRMS or PPMS?"

The first statement is true, while the second statement is not. These statements are expressed as follows:

1. $[Disease\ ;\ CIS]\mathrm{EF}(healthy \wedge \sim healthy)$,
2. $[Disease\ ;\ MS\ ;\ RRMS]\mathrm{EF}(healthy \wedge \sim healthy) \vee$

[*Disease* ; *MS* ; *PPMS*]EF(*healthy* $\wedge \sim$*healthy*).

We can also verify the following statement using LIHpCTL:

> "If a person living in the USA has CIS and no lesion has been detected by the MRI, then there is an approximately 80% chance that the person will be cured in the near future."

This statement is true and is expressed as follows:

$[USA][Disease\ ;\ CIS]\ (\mathrm{AG}(\sim healthy \wedge \neg MRIdetectedLesion) \rightarrow \mathrm{EF}(\mathrm{P}_{\leq 0.85}\ cure \wedge \mathrm{P}_{\geq 0.75}\ cure))$.

5 CONCLUSION

In this paper, we have introduced a new extended temporal logic, LIHpCTL, to establish the logical foundation of a new model checking paradigm. This model checking paradigm is called locative inconsistency-tolerant hierarchical probabilistic CTL model checking, and is intended to effectively verify locative, inconsistent, hierarchical, probabilistic (randomized), and time-dependent concurrent systems. LIHpCTL is an extension of previously proposed locative, inconsistency-tolerant, hierarchical, and probabilistic extensions of the standard temporal logic known as CTL. We have also presented a survey of various studies on probabilistic, inconsistency-tolerant, and hierarchical temporal logics and their applications to model checking. Although the decidability of the model checking problem for LIHpCTL has not yet been determined, we have presented an illustrative example for verifying the clinical reasoning process for a disease, MS, using the LIHpCTL-based model checking paradigm. We have thus shown that LIHpCTL and its model checking framework, which is an extension of existing frameworks, are useful for a variety of existing and novel applications in computer science and artificial intelligence.

ACKNOWLEDGEMENTS

This research was supported by the Kayamori Foundation of Informational Science Advancement. This research has been partially supported by JSPS KAKENHI Grant Numbers JP18K11171, JP16KK0007 and JSPS Core-to-Core Program (A. Advanced Research Networks).

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.

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.

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

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

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.

Carnielli, W., Coniglio, M., Gabbay, D., Gouveia, P., and Sernadas, C. (2008). *Analysis and synthesis of logics: How to cut and paste reasoning systems, Applied Logic Series, Vol. 35.* Springer.

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.

Dunn, J. (1976). Intuitive semantics for first-degree entailment and 'coupled trees'. *Philosophical Studies*, 29 (3):149–168.

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

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. (2005). A spatial modal logic with a location interpretation. *Mathematical Logic Quarterly*, 51 (4):331–341.

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

Kamide, N. (2006b). Linear and affine logics with temporal, spatial and epistemic operators. *Theoretical Computer Science*, 353 (1-3):165–207.

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 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.

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.

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

Ognjanovic, Z., Doder, D., and Markovic, Z. (2011). A branching time logic with two types of probability operators. *Proceedings of the 5th International Conference on Scalable Uncertainty Management, Lecture Notes in Artificial Intelligence*, 6929:219–232.

Ognjanovic, Z., Markovic, Z., Raskovic, M., Doder, D., and Perovic, A. (2012). A propositional probabilistic logic with discrete linear time for reasoning about evidence. *Annals of Mathematics and Artificial Intelligence*, 65 (2-3):217–243.

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

Priest, G. (2002). *Paraconsistent logic, Handbook of Philosophical Logic (Second Edition), 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 (2018). Multiple sclerosis. *Wikipedia*, `https://en.wikipedia.org/wiki/Multiple_sclerosis`.