Towards Hierarchical Probabilistic CTL Model Checking: Theoretical Foundations

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

Keywords: Computation Tree Logic, Probabilistic Computation Tree Logic, Hierarchical Computation Tree Logic, Embedding Theorem, Relative Decidability.

Abstract: This study proposes a hierarchical probabilistic computation tree logic, HpCTL, which is an extension of the standard probabilistic computation tree logic pCTL, as a theoretical basis for hierarchical probabilistic CTL model checking. Hierarchical probabilistic model checking is a new paradigm that can appropriately verify hierarchical randomized (or stochastic) systems. Furthermore, a probability-measure-independent translation from HpCTL into pCTL is defined, and a theorem for embedding HpCTL into pCTL is proved using this translation. Finally, the relative decidability of HpCTL with respect to pCTL is proved using this embedding theorem. These embedding and relative decidability results allow us to reuse the standard pCTL-based probabilistic model checking algorithms to verify hierarchical randomized systems that can be described using HpCTL.

1 INTRODUCTION

1.1 Aims

In this study, we develop a new temporal logic for *hierarchical probabilistic model checking*, which is a new model checking paradigm that can appropriately verify hierarchical randomized (or stochastic) systems. *Model checking* is a formal method for automatically verifying concurrent systems (Clarke and Emerson, 1981; Cavada et al., 2015; Holzmann, 2006; Clarke et al., 2018), and has been extended to *probabilistic model checking* (Aziz et al., 1995; Bianco and de Alfaro, 1995; Kwiatkowska et al., 2011; Baier et al., 2018) and *hierarchical model checking* (Kamide and Kaneiwa, 2009; Kaneiwa and Kamide, 2011; Kamide, 2015; Kamide and Yano, 2017). Thus, the objective of this study is to integrate these extended model checking paradigms. *Computation tree logic* (CTL) (Clarke and Emerson, 1981) has been typically used as a theoretical basis for model checking. In fact, the model checker known as NuSMV (Cavada et al., 2015) was developed based on CTL. However, CTL is unsuitable for verifying hierarchical randomized systems because it lacks the constructors to represent "hierarchical randomized" systems naturally. Thus, CTL has been extended to *probabilistic computation tree logics* (Aziz et al., 1995; Bianco and de Alfaro, 1995) and *hierarchical (or sequential) computation tree logics* (Kamide and Kaneiwa, 2009; Kaneiwa and Kamide, 2011; Kamide and Yano, 2017; Kamide, 2018). The main aim of this study is to integrate these extended logics for obtaining a theoretical basis for hierarchical probabilistic model checking.

1.2 Probabilistic Computation Tree Logic

Several probabilistic computation tree logics and their variants have been studied to handle randomized (or stochastic) systems (Aziz et al., 1995; Bianco and de Alfaro, 1995; Kamide and Koizumi, 2015; Kamide and Koizumi, 2016). A probabilistic computation tree logic, pCTL, was studied by Aziz et al. (Aziz et al., 1995) and Bianco and de Alfaro (Bianco and de Alfaro, 1995). This pCTL was obtained from CTL by adding some *probabilistic* or *probability operators*, such as $P_{\geq x}$. The formulas of the form $P_{\geq x}\alpha$ are intended to read "the probability of α holding in the future evolution of the system is at least x." In (Bianco and de Alfaro, 1995), the logic pCTL was studied to verify the reliability properties and performance of the systems mod-

Kamide, N. and Yano, Y.
Towards Hierarchical Probabilistic CTL Model Checking: Theoretical Foundations.
DOI: 10.5220/0007456507620769
In *Proceedings of the 11th International Conference on Agents and Artificial Intelligence (ICAART 2019)*, pages 762-769
ISBN: 978-989-758-350-6

eled by *discrete Markov chains*, and the complexities of model-checking algorithms with regard to the logic were also clarified. In (Aziz et al., 1995), efficient model checking algorithms for various extensions of the previous settings of pCTL were proposed to verify probabilistic non-deterministic concurrent systems, wherein probabilistic behavior coexists with non-determinism. The difference between the pCTL-settings of Aziz et al. (Aziz et al., 1995) and those of Bianco and de Alfaro (Bianco and de Alfaro, 1995) is the settings of the *probability measures* within the *probabilistic Kripke structures* of pCTL. In (Kamide and Koizumi, 2015; Kamide and Koizumi, 2016), the *inconsistency-tolerant (or paraconsistent) probabilistic computation tree logic*, PpCTL, which was obtained from pCTL by adding a *paraconsistent negation connective* $\sim$, was developed on the basis of a probability-measure-independent translation of PpCTL into pCTL. The theorem for embedding PpCTL into pCTL was shown using this translation, and entailed the relative decidability of PpCTL with respect to pCTL. This result indicates that we can reuse the existing pCTL-based model checking algorithms of Aziz et al. (Aziz et al., 1995) and Bianco and de Alfaro (Bianco and de Alfaro, 1995) to the PpCTL-based model checking algorithms. Hence, the aim of this study is to progress in this direction for hierarchical probabilistic model checking.

1.3 Hierarchical Computation Tree Logic

Several hierarchical (or sequential) computation tree logics and their variants have been studied to handle hierarchical systems (Kamide and Kaneiwa, 2009; Kaneiwa and Kamide, 2011; Kamide, 2015; Kamide and Yano, 2017; Kamide, 2018). A modal operator called a *sequence modal operator*, which is denoted as $[b]$ where b is a sequence, is used in these hierarchical computation tree logics. The formulas of the form $[b_1 ; b_2 ; \cdots ; b_n]\alpha$ intuitively mean that "α is true based on a sequence $b_1 ; b_2 ; \cdots ; b_n$ of ordered pieces of information." For more information on $[b]$, see Remark 2.6 in Section 2. In (Kamide, 2015), an extension of CTL, which was called the *sequence-indexed paraconsistent computation-tree logic*, SPCTL, was introduced by adding $[b]$ and $\sim$ to CTL. This logic was used to verify clinical reasoning systems. In (Kamide and Kaneiwa, 2009; Kaneiwa and Kamide, 2011), an extension of the *full computation tree logic* (CTL*), which was called CTLS*, was developed by adding $[b]$ to CTL*. This logic was used to represent conceptual hierarchies and ontologies. In (Kaneiwa and Kamide, 2010), an extension of the *linear-time temporal logic* (LTL) (Pnueli, 1977), which was called the *sequence-indexed linear-time temporal logic*, SLTL, was introduced by adding $[b]$ to LTL. In addition, a proof system for SLTL was developed to verify certain specifications of secure authentication systems. In (Kamide and Yano, 2017; Kamide, 2018), an extension of CTL, which was called the *sequential computation tree logic*, sCTL, was introduced by adding $[b]$ to CTL. The logic sCTL has a simple single satisfaction relation, which is compatible with that of CTL. Thus, the aim of this study is to move in this direction for hierarchical probabilistic computation tree logic. In fact, the logic proposed in this study is regarded as an extension of sCTL.

1.4 Results

In this study, a simple new extended computation tree logic called *hierarchical probabilistic computation tree logic*, HpCTL, which can appropriately represent hierarchical information and probabilistic phenomena, is developed by extending pCTL and sCTL. Furthermore, a probability-measure-independent translation from HpCTL into pCTL is defined, and a theorem for embedding HpCTL into pCTL is proved using this translation. In addition, the relative decidability theorem of HpCTL with respect to pCTL is proved using this embedding theorem. This relative decidability theorem indicates that the decidability of pCTL implies the decidability of HpCTL. Moreover, these embedding and relative decidability results allow the efficient reuse of the standard pCTL-based probabilistic model checking algorithms to verify hierarchical randomized systems that can be modeled and specified using HpCTL. The previously proposed logics CTLS* (Kamide and Kaneiwa, 2009; Kaneiwa and Kamide, 2011), SLTL (Kaneiwa and Kamide, 2010), and SPCTL (Kamide, 2015) had complex multiple sequence-indexed satisfaction relations $\models^{\hat{d}}$, where $\hat{d}$ represents sequences. On the other hand, the proposed logic HpCTL has a simple single satisfaction relation $\models^\star$, which is highly compatible with the standard single satisfaction relation of CTL. By using this simple satisfaction relation, the theorem for embedding HpCTL into pCTL can be simply proved, and the sequence modal operator $[b]$ can be formalized and handled uniformly.

The remainder of this paper is organized as follows. In Section 2, we define pCTL and introduce HpCTL based on the single satisfaction relation $\models^\star$. In Section 3, we define a probability-measure-independent translation function from HpCTL into pCTL, which is considered a simplification of the

translation functions used in (Kamide and Kaneiwa, 2009; Kaneiwa and Kamide, 2010; Kaneiwa and Kamide, 2011; Kamide, 2015). The theorem for embedding HpCTL into pCTL is proved using the proposed translation function, and the relative decidability theorem for HpCTL is obtained using this embedding theorem. In Section 4, we address some illustrative examples for hierarchical probabilistic CTL model checking based on HpCTL. In Section 5, we present our concluding remarks.

2 LOGICS

Formulas of *probabilistic computation tree logic* (pCTL) are constructed from countably many propositional variables, $\rightarrow$ (implication) $\wedge$ (conjunction), $\vee$ (disjunction), $\neg$ (classical 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), and $\mathrm{P}_{>x}$ (greater than probability). 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 operators* or *probability operators*. A formula $\mathrm{P}_{\leq x}\alpha$ is intended to read "the probability of α is at least x." We use the symbol Φ to denote a non-empty set of propositional variables. We use an expression $A \equiv B$ to denote the syntactical identity between A and B.

Definition 2.1. *Formulas* α *of* pCTL *are defined by the following grammar, assuming* $p \in \Phi$ *and* $x \in [0,1]$*:*

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

In this definition, pairs of symbols like AG and EU are indivisible, and the symbols $\mathrm{X}, \mathrm{G}, \mathrm{F}, \mathrm{U}$ and R cannot occur without being preceded by an A or an E. Similarly, every A or E must have one of X, G, F, U and R to accompany it.

Definition 2.2 (pCTL)**.** *A structure* (S, S_0, R, μ_s, L) *is a* 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 (or probability distribution) concerning* $s \in S$*: a set of paths beginning at s is mapped into a real number in* $[0,1]$*,*
5. *L is a mapping from S to the power set of* Φ*.*

A path *in a model is an infinite sequence of states,* $\pi = s_0, s_1, s_2, \ldots$ *such that*

$\forall i \geq 0\ [(s_i, s_{i+1}) \in R]$.

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

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

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

Remark 2.3.

1. *The definition of* μ_s *is not precisely and explicitly given in this paper. The reasons are as follows. (1) the proposed translation from* HpCTL *into* pCTL *is independent of the setting of* μ_s*. (2) There are many possibilities for defining* μ_s*.*

2. *In* (Bianco and de Alfaro, 1995)*, two probability measures* μ_s^+ *and* μ_s^-*, called* minimal probability *and* maximal probability*, respectively, are introduced for* pCTL*. The measures* μ_s^+ *and* μ_s^- *are defined on a Borel* σ*-algebra* $\mathcal{B}_s$ $(\subseteq 2^{\Omega_s})$ *as follows: for any* $\Delta \in \mathcal{B}_s$*,* $\mu_s^+(\Delta) = sup\ \mu_{s,\eta}(\Delta)$ *and* $\mu_s^-(\Delta) = inf\ \mu_{s,\eta}(\Delta)$ *where* $\mu_{s,\eta}$ *with a strategy* η *concerning nondeterminism is a unique probability measure on* $\mathcal{B}_s$*.*
3. *In* (Aziz et al., 1995)*, a probability measure* μ^s *concerning some* discrete Markov processes *and* discrete generalized Markov processes *is introduced for* pCTL*.* μ^s *is defined as a mapping from* $\mathcal{C}^s$ *into* $[0,1]$ *where* $\mathcal{C}^s$ *is a Borel sigma field, which is the class of subsets of the set of all infinite state sequences starting at s.*

The language of *hierarchical probabilistic computation tree logic* (HpCTL) is obtained from that of pCTL by adding $[b]$ (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 $b, c, \ldots$ 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 $\Phi^{[d]}$ $(d \in \mathrm{SE})$ is used to denote the set $\{[d]p \mid p \in \Phi\}$, and the symbol Φ^d $(d \in \mathrm{SE})$ is used to denote the set $\{p^d \mid p \in \Phi\}$ of propositional variables where $p^\emptyset$ means p. Note that $\Phi^{[\emptyset]} = \Phi^\emptyset = \Phi$.

Definition 2.4. *Formulas* α *and sequences* b *of* HpCTL *are defined by the following grammar, assuming* p *and* e *represent propositional variables and atomic sequences, respectively:*

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

We use an expression $\overline{[d]}$ to represent $[d_0][d_1]\cdots[d_i]$ with $i \in \omega$, $d_i \in \mathrm{SE}$ and $d_0 \equiv \emptyset$. We remark that $\overline{[d]}$ can be the empty sequence, and that $\overline{[d]}$ 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 $[d_1\ ;\ d_2\ ;\ d_3]$. We also note that $\overline{[d]}$ can be $[d]$ (i.e., $\overline{[d]}$ includes $[d]$).

Definition 2.5 (HpCTL)**.** *A structure* $(S, S_0, R, \mu_s, L^\star)$ *is a* 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* $[0,1]$*,*
5. $L^\star$ *is a mapping from S to the power set of* $\bigcup_{d \in \mathrm{SE}} \Phi^{[d]}$*.*

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

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

1. *for any* $p \in \Phi$*,* $(M,s) \models^\star [d]p$ *iff* $[d]p \in L^\star(s)$*,*
2. $(M,s) \models^\star \overline{[d]}[b]\alpha$ *iff* $(M,s) \models^\star [d\ ;\ b]\alpha$*,*
3. $(M,s) \models^\star \overline{[d]}(\alpha \wedge \beta)$ *iff* $(M,s) \models^\star \overline{[d]}\alpha$ *and* $(M,s) \models^\star \overline{[d]}\beta$*,*
4. $(M,s) \models^\star \overline{[d]}(\alpha \vee \beta)$ *iff* $(M,s) \models^\star \overline{[d]}\alpha$ *or* $(M,s) \models^\star \overline{[d]}\beta$*,*
5. $(M,s) \models^\star \overline{[d]}(\alpha{\rightarrow}\beta)$ *iff* $(M,s) \models^\star \overline{[d]}\alpha$ *implies* $(M,s) \models^\star \overline{[d]}\beta$*,*
6. $(M,s) \models^\star \overline{[d]}\neg\alpha$ *iff* $(M,s) \not\models^\star \overline{[d]}\alpha$*,*
7. $(M,s) \models^\star \overline{[d]}\mathrm{AX}\alpha$ *iff* $\forall s_1 \in S\ [(s,s_1) \in R$ *implies* $(M,s_1) \models^\star \overline{[d]}\alpha]$*,*
8. $(M,s) \models^\star \overline{[d]}\mathrm{EX}\alpha$ *iff* $\exists s_1 \in S\ [(s,s_1) \in R$ *and* $(M,s_1) \models^\star \overline{[d]}\alpha]$*,*
9. $(M,s) \models^\star \overline{[d]}\mathrm{AG}\alpha$ *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* $(M,s_i) \models^\star \overline{[d]}\alpha$*,*
10. $(M,s) \models^\star \overline{[d]}\mathrm{EG}\alpha$ *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* $(M,s_i) \models^\star \overline{[d]}\alpha$*,*
11. $(M,s) \models^\star \overline{[d]}\mathrm{AF}\alpha$ *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* $(M,s_i) \models^\star \overline{[d]}\alpha$*,*
12. $(M,s) \models^\star \overline{[d]}\mathrm{EF}\alpha$ *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* $(M,s_i) \models^\star \overline{[d]}\alpha$*,*
13. $(M,s) \models^\star \overline{[d]}\mathrm{A}(\alpha \mathrm{U} \beta)$ *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* $\forall 0 \leq k < j\ (M,s_k) \models^\star \overline{[d]}\alpha$*,*
14. $(M,s) \models^\star \overline{[d]}\mathrm{E}(\alpha \mathrm{U} \beta)$ *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* $\forall 0 \leq k < j\ (M,s_k) \models^\star \overline{[d]}\alpha$*,*
15. $(M,s) \models^\star \overline{[d]}\mathrm{A}(\alpha \mathrm{R} \beta)$ *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* $\exists 0 \leq k < j\ (M,s_k) \models^\star \overline{[d]}\alpha$*,*

16. $(M,s) \models^\star \overline{[d]}\mathrm{E}(\alpha \mathrm{R} \beta)$ *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* $\exists 0 \leq k < j\ (M,s_k) \models^\star \overline{[d]}\alpha$,
17. *for any* $x \in [0,1]$, $M,s \models^\star \overline{[d]}\mathrm{P}_{\leq x}\alpha$ *iff* $\mu_s(\{w \in \Omega_s \mid M,s \models^\star \overline{[d]}\alpha\}) \leq x$,
18. *for any* $x \in [0,1]$, $M,s \models^\star \overline{[d]}\mathrm{P}_{\geq x}\alpha$ *iff* $\mu_s(\{w \in \Omega_s \mid M,s \models^\star \overline{[d]}\alpha\}) \geq x$,
19. *for any* $x \in [0,1]$, $M,s \models^\star \overline{[d]}\mathrm{P}_{< x}\alpha$ *iff* $\mu_s(\{w \in \Omega_s \mid M,s \models^\star \overline{[d]}\alpha\}) < x$,
20. *for any* $x \in [0,1]$, $M,s \models^\star \overline{[d]}\mathrm{P}_{> x}\alpha$ *iff* $\mu_s(\{w \in \Omega_s \mid M,s \models^\star \overline{[d]}\alpha\}) > x$.

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

Remark 2.6.

1. *The following clauses hold for any formula* α, *and any sequences* b, c *and* d,
   (a) $(M,s) \models^\star [b][c]\alpha$ *iff* $(M,s) \models^\star [b\ ;\ c]\alpha$,
   (b) $(M,s) \models^\star \overline{[d]}\alpha$ *iff* $(M,s) \models^\star [d]\alpha$.
2. *The following formulas are valid in* HpCTL*: For any formulas* α *and* β, *and any sequences* b, c *and* d,
   (a) $[b](\alpha \sharp \beta) \leftrightarrow ([b]\alpha) \sharp ([b]\beta)$ *where* $\sharp \in \{\wedge, \vee, \rightarrow\}$,
   (b) $[b]\sharp\alpha \leftrightarrow \sharp[b]\alpha$ *where* $\sharp \in \{\neg, \mathrm{AX}, \mathrm{EX}, \mathrm{AG}, \mathrm{EG}, \mathrm{AF}, \mathrm{EF}\}$,
   (c) $[b](\mathrm{A}(\alpha \mathrm{U} \beta)) \leftrightarrow \mathrm{A}(([b]\alpha)\mathrm{U}([b]\beta))$,
   (d) $[b](\mathrm{E}(\alpha \mathrm{U} \beta)) \leftrightarrow \mathrm{E}(([b]\alpha)\mathrm{U}([b]\beta))$,
   (e) $[b](\mathrm{A}(\alpha \mathrm{R} \beta)) \leftrightarrow \mathrm{A}(([b]\alpha)\mathrm{R}([b]\beta))$,
   (f) $[b](\mathrm{E}(\alpha \mathrm{R} \beta)) \leftrightarrow \mathrm{E}(([b]\alpha)\mathrm{R}([b]\beta))$,
   (g) $[b][c]\alpha \leftrightarrow [b\ ;\ c]\alpha$,
   (h) $\overline{[d]}\alpha \leftrightarrow [d]\alpha$.
3. *The operator* $[b]$ *is useful for representing informative and highly complex hierarchical systems with the concepts of hierarchical information, hierarchical trees, orders, and ontologies. This is plausible because a sequence structure gives 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), and (3)* $\emptyset$ *is an empty piece of information (i.e., an empty sequence). Then, the formulas of the form* $[b_1\ ;\ b_2\ ;\cdots;\ b_n]\alpha$ *intuitively mean that "*α *is true based on a sequence* $b_1\ ;\ b_2\ ;\cdots;\ b_n$ *of ordered pieces of information." Furthermore, the formulas of the form* $[\emptyset]\alpha$, *which coincide with* α, *intuitively mean that "*α *is true without any information (i.e., it is an eternal truth in the sense of classical logic)."*

3 EMBEDDING AND RELATIVE DECIDABILITY

Definition 3.1. *The language* $\mathcal{L}^h$ *(the set of formulas) of* HpCTL *is defined using* Φ, $\rightarrow, \wedge, \vee, \neg$, 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* pCTL *is obtained from* $\mathcal{L}^h$ *by adding* Φ^d *and deleting* $[b]$.

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

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

Proposition 3.2. *Let* f *be the mapping defined in Definition 3.1. The following conditions hold for any sequences* b, c, d *and* k:

1. $f(\overline{[d]}[b][c]\alpha) = f(\overline{[d]}[b\ ;\ c]\alpha)$,
2. $f(\overline{[d]}[k]\alpha) = f(\overline{[d]}[k]\alpha)$.

Lemma 3.3. *Let* f *be the mapping defined in Definition 3.1. For any hierarchical probabilistic model* $M := (S, S_0, R, \mu_s, L^\star)$ *of* HpCTL *and any hierarchical probabilistic satisfaction relation* $\models^\star$ *on* M, *we can construct a probabilistic model* $N := (S, S_0, R, \mu_s, L)$ *of* pCTL *and a probabilistic satisfaction relation* $\models$ *on* N *such that for any formula* α *in* $\mathcal{L}^h$, *any* $d \in \mathrm{SE}$, *and any state* s *in* S,

$(M,s) \models^\star \overline{[d]}\alpha$ *iff* $(N,s) \models f(\overline{[d]}\alpha)$.

Proof. Suppose that M is a hierarchical probabilistic model $(S, S_0, R, \mu_s, L^\star)$ such that

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

We then define a 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 $s \in S$ and any $p \in \Phi$,
   $[d]p \in L^\star(s)$ iff $p^d \in L(s)$.

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

• Base step:

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

• Induction step: We show some cases.

1. Case $\alpha \equiv [b]\beta$: $(M,s) \models^\star \overline{[d]}[b]\beta$ iff $(M,s) \models^\star [d\,;b]\beta$ iff $(N,s) \models f([d\,;b]\beta)$ (by induction hypothesis) iff $(N,s) \models f([d\,;b]\beta)$ (by the definition of f) iff $(N,s) \models f(\overline{[d]}[b]\beta)$ (by the definition of f).
2. Case $\alpha \equiv \beta \rightarrow \gamma$: We obtain: $(M,s) \models^\star \overline{[d]}(\beta \rightarrow \gamma)$ iff $(M,s) \models^\star \overline{[d]}\beta$ implies $(M,s) \models^\star \overline{[d]}\gamma$ iff $(N,s) \models f(\overline{[d]}\beta)$ implies $(N,s) \models f(\overline{[d]}\gamma)$ (by induction hypothesis) iff $(N,s) \models f(\overline{[d]}\beta) \rightarrow f(\overline{[d]}\gamma)$ iff $(N,s) \models f(\overline{[d]}(\beta \rightarrow \gamma))$ (by the definition of f).
3. Case $\alpha \equiv \neg\beta$: We obtain: $(M,s) \models^\star \overline{[d]}\neg\beta$ iff $(M,s) \not\models^\star \overline{[d]}\beta$ iff $(N,s) \not\models f(\overline{[d]}\beta)$ (by induction hypothesis) iff $(N,s) \models \neg f(\overline{[d]}\beta)$ iff $(N,s) \models f(\overline{[d]}\neg\beta)$ (by the definition of f).
4. Case $\alpha \equiv \mathrm{AX}\beta$: We obtain: $(M,s) \models^\star \overline{[d]}\mathrm{AX}\beta$ iff $\forall s_1 \in S\ [(s,s_1) \in R$ implies $(M,s_1) \models^\star \overline{[d]}\beta]$ iff $\forall s_1 \in S\ [(s,s_1) \in R$ implies $(N,s_1) \models f(\overline{[d]}\beta)]$ (by induction hypothesis) iff $(N,s) \models \mathrm{AX} f(\overline{[d]}\beta)$ iff $(N,s) \models f(\overline{[d]}\mathrm{AX}\beta)$ (by the definition of f).
5. Case $\alpha \equiv \mathrm{AG}\beta$: We obtain:
   $(M,s) \models^\star \overline{[d]}\mathrm{AG}\beta$
   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 $(M,s_i) \models^\star \overline{[d]}\beta$
   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 $(N,s_i) \models f(\overline{[d]}\beta)$ (by induction hypothesis)
   iff $(N,s) \models \mathrm{AG} f(\overline{[d]}\beta)$
   iff $(N,s) \models f(\overline{[d]}\mathrm{AG}\beta)$ (by the definition of f).
6. Case $\alpha \equiv \mathrm{A}(\beta \mathrm{U} \gamma)$: We obtain:
   $(M,s) \models^\star \overline{[d]}\mathrm{A}(\beta \mathrm{U} \gamma)$
   iff for all paths $\pi \equiv s_0, s_1, s_2, \ldots$, where $s \equiv s_0$, there is a state s_j along π such that $(M,s_j) \models^\star \overline{[d]}\gamma$ and $\forall 0 \leq k < j\ (M,s_k) \models^\star \overline{[d]}\beta$
   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 $(N,s_j) \models f(\overline{[d]}\gamma)$ and $\forall 0 \leq k < j\ (N,s_k) \models f(\overline{[d]}\beta)$ (by induction hypothesis)
   iff $(N,s) \models \mathrm{A}(f(\overline{[d]}\beta) \mathrm{U} f(\overline{[d]}\gamma))$
   iff $(N,s) \models f(\overline{[d]}\mathrm{A}(\beta \mathrm{U} \gamma))$ (by the definition of f).
7. Case $\alpha \equiv \mathrm{P}_{\leq x}\beta$: We obtain: $(M,s) \models^\star \overline{[d]}\mathrm{P}_{\leq x}\beta$ iff $\mu_s(\{w \in \Omega_s \mid (M,w) \models^\star \overline{[d]}\beta\}) \leq x$ iff $\mu_s(\{w \in \Omega_s \mid (N,w) \models f(\overline{[d]}\beta)\}) \leq x$ (by induction hypothesis) iff $(N,s) \models \mathrm{P}_{\leq x} f(\overline{[d]}\beta)$ iff $(N,s) \models f(\overline{[d]}\mathrm{P}_{\leq x}\beta)$ (by the definition of f). ∎

Lemma 3.4. *Let f be the mapping defined in Definition 3.1. For any probabilistic model $N := (S, S_0, R, \mu_s, L)$ of* pCTL *and any probabilistic satisfaction relation $\models$ on N, we can construct a hierarchical probabilistic model $M := (S, S_0, R, \mu_s, L^\star)$ of* HpCTL *and a hierarchical probabilistic satisfaction relation $\models^\star$ on M such that for any formula α in $\mathcal{L}^h$, any $d \in$ SE, and any state s in S,*

$(N,s) \models f(\overline{[d]}\alpha)$ *iff* $(M,s) \models^\star \overline{[d]}\alpha$.

Proof. Similar to the proof of Lemma 3.3. ∎

Theorem 3.5 (Embedding)**.** *Let f be the mapping defined in Definition 3.1. For any formula α,*

α *is valid in* HpCTL *iff $f(\alpha)$ is valid in* pCTL.

Proof. By Lemmas 3.3 and 3.4. ∎

Theorem 3.6 (Relative decidability)**.** *If the model-checking, validity, and satisfiability problems for* pCTL *with a probability measure are decidable, then the model-checking, validity, and satisfiability problems for* HpCTL *with the same probability measure as that of* pCTL *are also decidable.*

Proof. Suppose that the probability measure μ_s in the hierarchical probabilistic model $(S, S_0, R, \mu_s, L^\star)$ of HpCTL is the same as the probabilistic model (S, S_0, R, μ_s, L) of pCTL. Suppose also that pCTL with μ_s is decidable. Then, by the mapping f defined in Definition 3.1, a formula α of HpCTL can be transformed into the corresponding formula $f(\alpha)$ of pCTL. By Lemmas 3.3 and 3.4 and Theorem 3.5, the model checking, validity and satisfiability problems for HpCTL can be transformed into those of pCTL. Since the model checking, validity and satisfiability problems for pCTL with μ_s are decidable by the assumption, the problems for HpCTL with μ_s are also decidable. ∎

Remark 3.7.

1. *The model checking problem for* 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).
2. *The model checking problem for* pCTL *with the probability measure* μ^s *introduced by Aziz et al. was shown to be decidable in* (Aziz et al., 1995).
3. *An extension of* HpCTL *with the above-mentioned probability measures by Bianco and de Alfaro or by Aziz et al. is also decidable by Theorem 3.6.*
4. *The complexities of the decision procedures for the model checking, validity, and satisfiability problems of* HpCTL *are the same as those of* pCTL*, since the translation function* f *defined in Definition 3.1 is a polynomial time reduction.*

4 ILLUSTRATIVE EXAMPLES

We present an illustrative example for hierarchical probabilistic CTL model checking. Here, we consider the scenario presented in Figure 1, which shows a hierarchical probabilistic structure of students' learning processes in a university setting. In this figure, a student who is learning the natural sciences (e.g., physics) and engineering (e.g., electronics) will graduate from the university. In this model, the entrance examination pass rate is 60 % and the average examination pass rate varies among the courses.

In this example, we can declare the hierarchy of the academic subjects: *science*, *mathematics*, *analysis*, *vector analysis*, *electromagnetics*, *engineering*, *electronics*, *power electronics*, and *semiconductor engineering* as the sequence modal operator:

1. $[Science\,;Mathematics\,;Analysis\,;Vector\ analysis]$,
2. $[Science\,;Physics\,;Electromagnetics]$,
3. $[Engineering\,;Electronics\,;Telecoms\ engineering]$,
4. $[Engineering\,;Electronics\,;Power\ electronics]$,
5. $[Engineering\,;Electronics\,;Semiconductor\ engineering]$.

The first expression shows that the concept *Vector analysis* is a subconcept of *Analysis*, the concept of *Analysis* is a subconcept of *Mathematics*, and *Mathematics* is a subconcept of *Science*.

We can use some probabilistic operators to represent certain probabilistic phenomena concerning the learning process. As previously mentioned, the formula of the form $\mathrm{P}_{\geq x}\alpha$ is intended to read "the probability of α holding in the future evolution of the system is at least x." Thus, we can describe and verify the following statement using HpCTL:

> "If a student is learning in the second stage of the subject of "Telecoms engineering" and he or she understands the subject sufficiently, then there is approximately an 80 % chance that he or she will graduate some time in the near future."

This statement is true, and is expressed formally as:

$$[Engineering\,;Electronics\,;Telecoms\ engineering](\mathrm{AG}(stage2 \wedge learning \wedge understand \rightarrow \mathrm{EF}(\mathrm{P}_{\leq 0.85}\, graduate \wedge \mathrm{P}_{\geq 0.75}\, graduate))).$$

Moreover, if we can use the paraconsistent negation connective $\sim$, we can also express the negation of ambiguous concepts. If we cannot determine whether someone understands the underlying subject, then the ambiguous concept *understand* can be represented by asserting the following inconsistent formula: $understand \wedge \sim understand$. However, the classical negation connective $\neg$ is appropriate for describing the negation of the non-ambiguous concept *learning*.

5 CONCLUDING REMARKS

In this study, the new logic HpCTL and its translation into the standard logic pCTL were developed to obtain a theoretical foundation for hierarchical probabilistic CTL model checking. We demonstrated that the existing probabilistic model checking algorithms for pCTL can be reused for hierarchical probabilistic model checking as described using HpCTL. Moreover, we noted that the complexity of the model-checking algorithms for HpCTL is the same as that of pCTL. In addition, some illustrative examples for hierarchical probabilistic CTL model checking was presented in this study.

Prospective courses of study may involve extending the proposed logic by adding a paraconsistent negation connective. An extended hierarchical computation tree logic with a paraconsistent negation connective was studied in (Kamide, 2015). By combining our present work with that in (Kamide, 2015), we hope to establish the theoretical foundations of hierarchical inconsistency-tolerant probabilistic model checking based on such an extended logic.

ACKNOWLEDGEMENTS

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

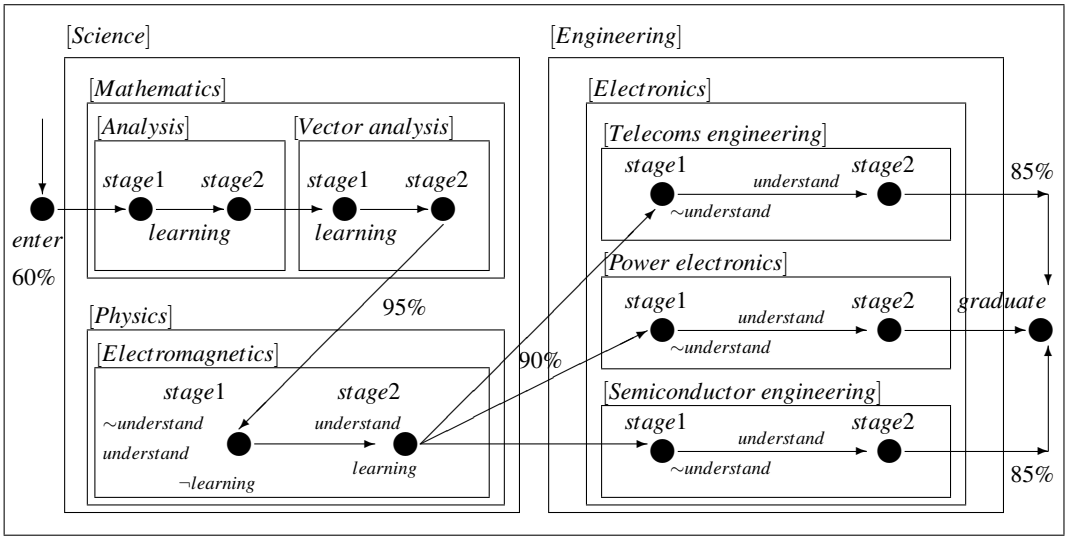

Figure 1: A hierarchical learning process model for academic subjects.

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.

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.

Cavada, R., Cimatti, A., Jochim, C., Keighren, G., Olivetti, E., Pistore, M., Roveri, M., and Tchaltsev, A. (2015). *NuSMV 2.6 user manual, 144 pages*. Online available.

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

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

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

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.

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

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