

# 矛盾許容階層確率モデル検査技術を用いた 臨床推論・知識検証技法の提案

山本 ノリコ<sup>1,a)</sup> Khoo Nee Kwan<sup>2,b)</sup> 楊 奕偉<sup>2,c)</sup> 石野 寛人<sup>2,d)</sup> 神戸 勢大<sup>2,e)</sup> 上出 哲広<sup>2,f)</sup>

**概要:** 本研究では、矛盾許容階層確率モデル検査技術を用いた臨床推論・知識検証技法を提案する。矛盾許容階層確率モデル検査技術は、矛盾許容性、階層性および確率性を扱えるように標準的なモデル検査技術を拡張したものである。本研究で扱う臨床推論および臨床知識は、それぞれ医師が疾患を診断するための思考過程および臨床医学オントロジーなどを基盤として構造化された臨床情報である。本研究では、矛盾許容階層確率モデル検査技術を標準的なモデル検査技術に変換するアルゴリズムを使用する。これにより、矛盾許容性、階層性および確率性を考慮した臨床推論・知識のモデルや対応する検査式を標準的なモデル検査技術で使用されるモデルや検査式に変換することができる。この変換アルゴリズムを用いることにより、臨床推論・知識を標準的なモデル検査器である SPIN, NuSMV および PRISM を組み合わせて検証することが可能になる。本研究では、臨床推論・知識に対するモデル化の具体例を与え、それらを検証する方法を示す。そのような検証の具体例として、簡単な疾患モデルに対する検証例を示す。

## Clinical Reasoning/Knowledge Verification Method Using Inconsistency-tolerant Hierarchical Probabilistic Model Checking

NORIKO YAMAMOTO<sup>1,a)</sup> KHOO NEE KWAN<sup>2,b)</sup> YIWEI YANG<sup>2,c)</sup> HIROTO ISHINO<sup>2,d)</sup> SEIDAI KANBE<sup>2,e)</sup>  
NORIHIRO KAMIDE<sup>2,f)</sup>

### 1. 序論

#### 1.1 矛盾許容階層確率モデル検査技術の概要

本研究の目的は、矛盾許容階層確率モデル検査技術を使用することによる臨床推論・知識検証技法を提案することである。矛盾許容階層確率モデル検査技術は、最近著者らによって導入された矛盾許容階層確率計算木論理 (IHpCTL) をベースにして拡張されたモデル検査技術である [26]。モデル検査技術は、状態遷移系によってモデル化される並行システムを検証するために広く使用されているコンピュータ支援技術である [7], [8], [9], [13]。矛盾許容階層確率モデ

ル検査技術のベースとなっている論理である IHpCTL は、確率推論 [4]、矛盾許容推論 [6] および階層的推論 [30] を適切に扱うことができる拡張時間論理である。IHpCTL を使用することにより、確率モデル検査 [1], [2], [3], [4], [32]、矛盾許容モデル検査 [6], [10], [18], [19], [29] および階層モデル検査 [15], [16], [20], [25], [30] の枠組みを統合することが可能になる。IHpCTL は標準的な確率時間論理である確率計算木論理 (pCTL) [1], [4] の拡張であり、pCTL は標準的な時間論理である計算木論理 (CTL) [8] の拡張である。最近著者らによって提案された IHpCTL から pCTL への変換アルゴリズムを使用することにより、IHpCTL をベースにした矛盾許容階層確率モデル検査アルゴリズムを pCTL をベースにした標準的な確率モデル検査アルゴリズム [1], [4] に還元することが可能になる。

#### 1.2 提案する臨床推論・知識検証技法の概要

本研究では、矛盾許容性、階層性および確率性を同時に扱

<sup>1</sup> 帝京大学大学院理工学研究科情報科学専攻

<sup>2</sup> 帝京大学理工学部情報電子工学科

a) noriko@while.jp

b) nkkhoo555@gmail.com

c) 183146u5@stu.teikyo-u.ac.jp

d) 183204vm@stu.teikyo-u.ac.jp

e) 1832143r@stu.teikyo-u.ac.jp

f) drnkamide08@kpd.biglobe.ne.jp

うことが可能な IHPCTL をベースにした矛盾許容階層確率モデル検査技術を使用した新たな臨床推論・知識検証技法を提案する。提案法では、IHPCTL とその部分体系に対する pCTL や CTL への変換アルゴリズムを用いる。これにより、標準的なモデル検査器である NuSMV[5], SPIN[13] および PRISM[32] を組み合わせて用いることが可能になる。本研究では、このような変換アルゴリズムとして [26] で提案されたものを使用する。

本研究で扱う臨床推論および臨床知識は、それぞれ医師が疾患を診断するための思考過程および臨床医学オントロジー等を基盤として構造化された臨床データ・情報である。これら臨床推論・知識に現れる「病気」「健康」の概念は曖昧な概念であり矛盾した内容を含んでいる。例えば、健康という概念については、「健康でかつ健康でない」という矛盾した状態（「未病」に対応する状態）が存在し得る。このような矛盾状態を適切に扱うためには、矛盾許容モデル検査技術が有効である [6], [10], [18], [29]。また、実際の臨床推論・知識を扱うには上記のような矛盾許容性だけではなく、階層性および確率性も考慮しなければならない。例えば、病気などの概念の階層性を適切に扱うためには、階層モデル検査技術が有効である [15], [16], [20], [25], [30]。病気の治癒の可能性や患者の死亡率などを適切に扱うためには、確率モデル検査技術が有効である [1], [2], [3], [4], [32]。従って、本研究では、これら矛盾許容モデル検査技術、階層モデル検査技術および確率モデル検査技術を統合した矛盾許容階層確率モデル検査技術が必要となる。

## 2. 矛盾許容階層確率モデル検査技術の基礎

### 2.1 本研究で使用する論理体系と基本定理

本研究では、論文 [26] で提案された矛盾許容階層確率計算木論理 (Inconsistency-tolerant Hierarchical Probabilistic Computation Tree Logic) IHPCTL を使用する。以下、IH-pCTL の概要と基本的性質について説明する。IHPCTL の定義の詳細については論文 [26] を参照されたい。IHPCTL の論理式では以下の論理結合子および演算子を使用する。→ (implication), ∧ (conjunction), ∨ (disjunction), ¬ (classical negation), ∼ (paraconsistent negation), X (next time), G (globally in the future), F (eventually in the future), U (until), R (release), A (all computation paths), E (some computation path),  $P_{\leq x}$  (less than or equal probability),  $P_{\geq x}$  (greater than or equal probability),  $P_{< x}$  (less than probability),  $P_{> x}$  (greater than probability), [b] (hierarchical or sequence modal operator)。上記 [b] において b は列 (sequence) を表している。IHPCTL は、矛盾許容階層確率モデル  $(S, S_0, R, \mu_s, L^*)$  上の充足可能関係  $(M, s) \models^* \alpha$  により定義される。

IHPCTL に対して以下の結果が知られている。これらの結果により、IHPCTL に対するモデル検査アルゴリズムを

pCTL や CTL に対するモデル検査アルゴリズムに変換することに対する正当性が保証される。また、そのような変換に対する具体的なアルゴリズムを得ることができる。そのような変換アルゴリズムを用いることにより、IHPCTL をベースにした矛盾許容階層確率モデル検査技術を標準的なモデル検査技術に還元することが可能になる。これにより、標準的なモデル検査器である NuSMV, SPIN および PRISM を組み合わせた検証に対する正当性が保証されることになる。

**定理 (埋め込み可能性 [26])** IHPCTL は pCTL に埋め込み可能である。すなわち、IHPCTL から pCTL へのトランスレーション関数 (トランスレーションアルゴリズム)  $f$  が存在して以下が成立する。任意の論理式  $\alpha$  に対して、 $\alpha$  が IHPCTL で恒真であることの必要十分条件は  $f(\alpha)$  が pCTL で恒真であることである。

**定理 (相対決定可能性 [26])** もしある確率メジャーを持つ pCTL が決定可能ならば、同じ確率メジャーを持つ IH-pCTL も決定可能である。すなわち、ある確率メジャーを持つ pCTL におけるモデル検査問題、充足可能性問題および恒真性問題が決定可能ならば、同じ確率メジャーを持つ IHPCTL におけるモデル検査問題、充足可能性問題および恒真性問題も決定可能である。

### 2.2 これまで提案されてきた本研究に関連する論理体系

以下、IHPCTL が提案される以前からすでに提案されていた密接に関連する論理について簡単に説明する。矛盾許容確率計算木論理 (PpCTL) が論文 [22], [23] で提案された。PpCTL は pCTL に矛盾許容否定結合子  $\sim$  を加えることによって定義される。同論文 [22], [23] では、PpCTL から pCTL への変換アルゴリズムを用いた埋め込み定理が証明された。この埋め込み定理を用いることによって、PpCTL の pCTL に対する相対決定可能性定理が証明された。この相対決定可能性定理は「もし適切な確率メジャーを用いることにより pCTL が決定可能ならば、同じ確率メジャーを用いることによる PpCTL も決定可能である」ということを示した定理である。階層確率計算木論理 (HpCTL) が論文 [27] で提案された。HpCTL は pCTL に階層演算子 [b] を加えることによって定義される。同論文 [27] では、HpCTL から pCTL への変換アルゴリズムを用いた埋め込み定理が証明された。この埋め込み定理を用いることによって、HpCTL の pCTL に対する相対決定可能性定理が証明された。階層演算子については論文 [15], [20], [28], [30] などでも研究されてきた。空間矛盾許容階層確率計算木論理 (LIHPCTL) が論文 [17] で提案された。LIHPCTL は、PpCTL と HpCTL 両方の拡張であり、 $\sim$  と [b] に加えてさらに、[31] で考案された空間演算子  $[i]$  を加えること

によって pCTL の拡張として定義される。しかしながら、LIHpCTL に対しては、技術的な困難さにより、埋め込み定理も相対決定可能性定理も証明されていない。本研究で使用する IHpCTL は、LIHpCTL の空間演算子  $[l_i]$  を持たない部分体系の変種である。IHpCTL では、LIHpCTL で用いられていた複雑な充足可能関係  $\models^d$  をシンプルな充足可能関係  $\models^*$  を用いることにより再定義したことにより、pCTL への埋め込み定理を証明することが可能になった。

### 3. 臨床推論・知識の表現

#### 3.1 階層の表現

以下、IHpCTL を基にした臨床推論・知識の表現について説明していく。まず、論文 [30] を基にして、IHpCTL の階層演算子  $[b]$  を用いた表現について説明する。オントロジーの表現においては、is-a 関係が使用される。これはある概念とその概念の部分概念間の関係を表すものである。 $c_1, c_2, \dots, c_n$  を概念を表す記号とする。このとき、概念記号の列は階層演算子を使用して  $[c_1 ; c_2 ; \dots ; c_n]$  と記述される。ここで、 $[c_1 ; c_2 ; \dots ; c_n]$  の中の各々の順序  $(c_i, c_j)$  ( $1 \leq i < j \leq n$ ) は、 $c_i$  と  $c_j$  の間の is-a 関係を表している。例えば、 $[disease ; cancer]$  は、*cancer* が *disease* の部分概念であることを示している。これは、「すべての癌は病気である」ということを表している。

まず、図 1 に示されているある種の疾患オントロジーを考える。この図において、*schizophrenia* (統合失調症) と *anorexia nervosa* (拒食症) は *mental diseases* (精神的病気) の一種であると考えられる。例えば、この図のオントロジー表現の一部は階層演算子を用いて以下のように記述される。

- (1)  $[disease ; cancer]$ .
- (2)  $[disease ; diseaseOfMentalHealth]$ .
- (3)  $[disease ; cancer ; lungCancer]$ .
- (4)  $[disease ; cancer ; brainCancer]$ .
- (5)  $[disease ; diseaseOfMentalHealth ; schizophrenia]$ .

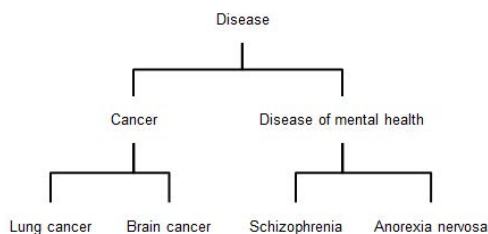


図 1 疾患の階層

今、 $c$  を概念とし、 $p$  を性質とする。このとき、論理式  $[c]p$  は概念  $c$  が性質  $p$  を持つことを表す。例えば、論理式  $[c_1 ; c_2]p$  は、概念  $c_2$  が概念  $c_1$  の部分概念でありかつ性質  $p$  を持つことを表す。概念の複雑な性質は論理結合子を使用して記述される。例えば、 $p_1$  および  $p_2$  が性質の

とき、これらを組み合わせた複雑な性質は、論理結合子を用いて  $[c](p_1 \wedge p_2)$ ,  $[c](p_1 \vee p_2)$ ,  $[c](p_1 \rightarrow p_2)$  および  $[c](\neg p_1)$  のように記述される。例えば、以下の表現は、肺癌は癌であるとともに病気でもあり、死に至る可能性もあるということを表している。

$[disease ; cancer ; lungCancer](death \vee \neg death)$ .

以下に、概念/性質および列/論理式の間の対応を表したリストを示す。

- (1) “A concept  $c_j$  is a subconcept  $c_i$  ( $1 \leq i < j \leq n$ )” は階層演算子  $[c_1 ; c_2 ; \dots ; c_n]$  で表現される。
- (2) “A concept  $c$  has a property  $p$ ” は論理式  $[c]p$  で表現される。
- (3) “A concept  $c$  has properties  $p_1$  and  $p_2$ ” は論理式  $[c](p_1 \wedge p_2)$  で表現される。
- (4) “A concept  $c$  has a property  $p_1$  or  $p_2$ ” は論理式  $[c](p_1 \vee p_2)$  で表現される。
- (5) “A concept  $c$  does not have a property  $p_1$  or does have a property  $p_2$ ” は論理式  $[c](p_1 \rightarrow p_2)$  で表現される。
- (6) “A concept  $c$  does not have a property  $p$ ” は論理式  $[c](\neg p)$  で表現される。

#### 3.2 時間の表現

次に、時間的性質を持つ概念を表現することを考える。これには IHpCTL の時間演算子を使用する。例えば、以下は *lung cancer* (肺癌) の概念における時間的性質を時間演算子を用いて表現したものである。

$[disease ; cancer ; lungCancer]EF$

$(\neg hasBenignTumor \wedge \neg hasMalignantTumor)$ .

この論理式は「一般に肺癌の初期のステージでは、医者は、肺癌患者の持つ良性の腫瘍も悪性の腫瘍も見つけることができない場合がある」ということを意味している。

以下に、時間的性質と時間論理式の間の対応を表したリストを示す。

- (1) “A concept  $c$  has a property  $p$  eventually in all paths” は論理式  $[c]AFp$  で表現される。
- (2) “A concept  $c$  has a property  $p$  eventually in some path” は論理式  $[c]EFp$  で表現される。
- (3) “A concept  $c$  always has a property  $p$  in all paths” は論理式  $[c]AGp$  で表現される。
- (4) “A concept  $c$  always has a property  $p$  in some path” は論理式  $[c]EGp$  で表現される。
- (5) “A concept  $c$  has a property  $p$  eventually in the near future (i.e., next time) in all paths” は論理式  $[c]AXp$  で表現される。
- (6) “A concept  $c$  has a property  $p$  eventually in the near future (i.e., next time) in some path” は論理式  $[c]EXp$  で表現される。
- (7) “A concept  $c$  has a property  $p_1$  until it has a property

$p_2$  in all paths” は論理式  $[c]A(p_1Up_2)$  で表現される。  
 (8) “A concept  $c$  has a property  $p_1$  until it has a property  $p_2$  in some path” は論理式  $[c]E(p_1Up_2)$  で表現される。

### 3.3 確率の表現

次に、確率的性質を持つ概念を表現することを考える。これには IHPCTL の確率演算子を使用する。例えば、以下は *lung cancer* の概念における確率的性質を確率演算子を用いて表現したものである。

$[disease ; cancer ; lungCancer]AG$   
 $(stage4 \wedge hasMalignantTumor \rightarrow P_{\geq 0.88} EFdeath)$ .

この論理式は「患者が肺癌のステージ 4 でありかつ悪性腫瘍を持っているならば、同患者は近い将来 88 パーセントの確率で死に至る可能性がある」ということを意味している。

以下に、確率的性質と確率論理式の間の対応を表したリストを示す。

- (1) “A concept  $c$  has a property  $p$  with a probability less than or equal to  $x$ ” は論理式  $[c]P_{\leq x}p$  で表現される。
- (2) “A concept  $c$  has a property  $p$  with a probability greater than or equal to  $x$ ” は論理式  $[c]P_{\geq x}p$  で表現される。
- (3) “A concept  $c$  has a property  $p$  with a probability less than  $x$ ” は論理式  $[c]P_{< x}p$  で表現される。
- (4) “A concept  $c$  has a property  $p$  with a probability greater than  $x$ ” は論理式  $[c]P_{> x}p$  で表現される。

### 3.4 矛盾許容性の表現

次に、矛盾許容性を持つ概念の扱いを考える。そのようなタイプの概念は IHPCTL の矛盾許容否定結合子  $\sim$  を使用することにより表現される。IHPCTL は、 $(\alpha \wedge \sim \alpha) \rightarrow \beta$  の形の公理を排除した論理である。これは IHPCTL が矛盾許容論理（矛盾を許容する論理）[33] の一種であることを意味する。標準的にモデル検査に使用されている時間論理は矛盾許容論理ではない。例えば、矛盾許容論理ではない論理においては、任意の病状  $s$  と病気  $d$  に対して、 $(s(x) \wedge \sim s(x)) \rightarrow d(x)$  の形の論理式が妥当になってしまい、問題が生じる。この論理式は「患者  $x$  の病状が  $s$  であるか否かがあいまいなとき（すなわち  $s$  がかつ  $s$  ではないとき） $x$  は病気  $d$  である」ということを表している。従って、当該論理は適切な臨床推論を表せないことになる。

例えば、 $schizophrenia(maria) \wedge \sim schizophrenia(maria)$  で表されるような状況（矛盾した状況）は頻繁に起こり得る。この論理式は、「マリアが統合失調症でありかつ統合失調症ではない」ということを表している。統合失調症の診断は難しく、医者によって診断が異なる可能性がある。そのため、このような「ある医者は統合失調症と診断したが別の医者は統合失調症ではないと診断した状態」が起こり得る。すると、矛盾許容論理以外の論理を使用する

と、 $(schizophrenia(maria) \wedge \sim schizophrenia(maria)) \rightarrow cancer(maria)$  のような望ましくない論理式が真となる。この論理式は「マリアが上記のような状態ならば癌である」ということを意味している。

## 4. 臨床推論・知識の検証例

まず、IHPCTL を用いた疾患モデルについて説明する。図 2、図 3 および図 4 はそれぞれ全体の疾患モデル、肺癌モデルおよび統合失調症モデルである。ここで、肺癌モデルと統合失調症モデルは全体の疾患モデルの部分モデルである。すなわち、図 2 中の [Lung cancer] および [Schizophrenia] の詳細がそれぞれ図 3 と図 4 の中で記述されている。肺癌および統合失調症のモデルに関する内容については、それぞれ [35] および [36] の内容を参考にした。

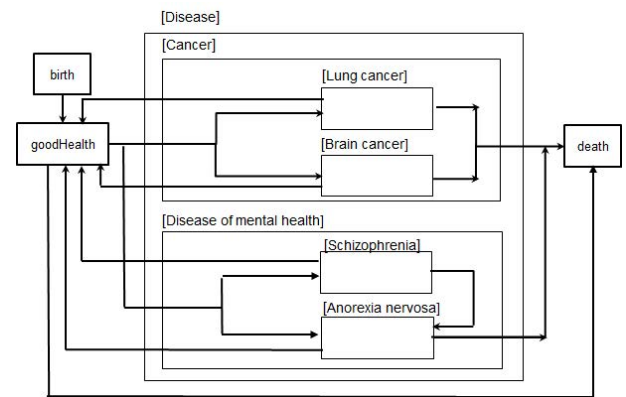


図 2 疾患モデル

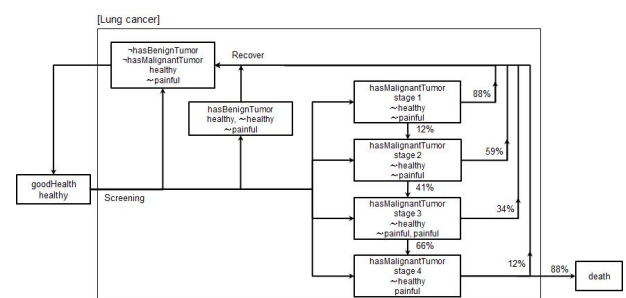


図 3 肺癌モデル

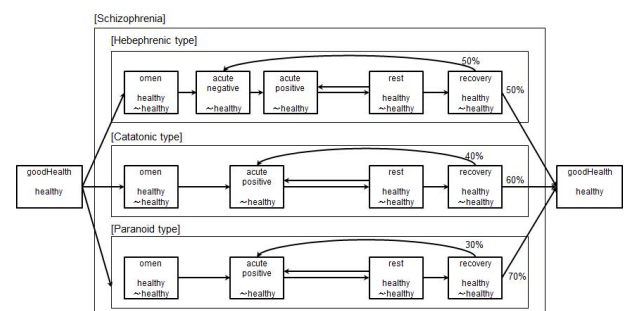


図 4 統合失調症モデル

次に, IHpCTL を用いた図 3 の肺癌モデルの検証例を示す. IHpCTL を用いることにより, 以下のステートメントを検証することができる.

(1) “Is there a state in which a person experiences both pain and the absence of pain during Stage 4 lung cancer?”

(2) “Is there a state in which a person is both healthy and unhealthy in the previous stage of lung cancer?”

第 1 のステートメントは偽であり, 第 2 のステートメントは真である. これらステートメントは IHpCTL 論理式としてそれぞれ以下のように記述できる.

(1)  $[Disease ; Cancer ; LungCancer]EF(painful \wedge \sim painful)$ .

(2)  $EF(healthy \wedge \sim healthy)$ .

また, 以下のステートメントも検証することができる.

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

このステートメントは真であり, IHpCTL 論理式として以下のように記述できる.

$[Disease ; Cancer ; LungCancer]AG$   
 $(stage4 \wedge hasMalignantTumor \wedge painful \wedge \sim healthy$   
 $\rightarrow EF(P_{\leq 0.89} death \wedge P_{\geq 0.87} death))$ .

上記のステートメントは, IHpCTL から pCTL への変換アルゴリズム [26] を使用して検証することができる. すなわち, IHpCTL 論理式は対応する pCTL 論理式に変換アルゴリズムを使用して変換され, 変換された pCTL 論理式は, 標準的な確率モデル検査技術によって検証することができる. 以下, そのような変換の例を, 最後に示した IHpCTL 論理式を例にとって説明する. まず, 以下のように当該 IHpCTL 論理式を略記して, これに対する変換結果を示す.

$[d ; c ; l](AG(s \wedge t \wedge p \wedge \sim h \rightarrow EF(P_{\leq 0.89} d \wedge P_{\geq 0.87} d)))$

ここで,  $d, c, l, t, p, h$  および  $h$  はそれぞれ, *Disease*, *Cancer*, *lungCancer*, *hasMalignantTumor*, *painful*, *healthy* および *death* の略記である. このとき, 対応する pCTL 論理式は以下ようになる.

$AG(q_1 \wedge q_2 \wedge q_3 \wedge q_4 \rightarrow EF(P_{\leq 0.89} q_5 \wedge P_{\geq 0.87} q_5))$

ここで,  $q_1, q_2, q_3, q_4$  および  $q_5$  は, それぞれ異なる命題変数である. この変換結果において, 元の略記版 IHpCTL 論理式中の  $\sim$  と  $[d ; c ; l]$  はそれぞれこれら新しい命題変数で表現される (命題変数中に埋め込まれる) ことになる.

## 5. 結論

本研究では, 矛盾許容階層確率モデル検査技術を用いた臨床推論・知識検証技法を提案した. 矛盾許容階層確率モデル検査技術は, 矛盾許容性, 階層性および確率性を扱えるように標準的なモデル検査技術を拡張したものである. 本

研究では, 論文 [26] で提案されている「矛盾許容階層確率モデル検査技術を標準的なモデル検査技術に変換するアルゴリズム」を使用した. これにより, 矛盾許容性, 階層性および確率性を考慮した臨床推論・知識のモデルや対応する検査式を標準的なモデル検査技術で使用されるモデルや検査式に変換することができた. この変換アルゴリズムを用いることにより, 臨床推論・知識を標準的なモデル検査器である SPIN, NuSMV および PRISM を組み合わせ検証することが可能になった. 本研究では, 臨床推論・知識モデルの簡単な具体例として肺癌モデルや統合失調症モデルを含む疾患モデルに対する検証例を示した.

本研究で扱った上記疾患モデルに対する検証はモデル検査器 NuSMV, SPIN および PRISM を組み合わせ実行することができる. NuSMV は CTL モデル検査技術と LTL モデル検査技術を, SPIN は LTL モデル検査技術, そして PRISM は pCTL モデル検査技術をサポートしている. ここで LTL モデル検査技術とは線形時間論理 LTL をベースにしたモデル検査技術である. 本研究で扱った疾患モデルに関連するこれらモデル検査器を組み合わせた実際の検証については論文 [34] で報告されている. なお, 標準的なモデル検査器ではモデル合成の機能があるため, 個々の疾患モデルをそのような合成機能を用いて自動的に合成することができる. 例えば, 統合失調症と肺癌を併発している場合には, それら個々のモデルの非同期合成モデルをモデル検査器により自動的に生成することができる. あるいは, 2つの密接に関連している疾患に対するモデルの合成を考える場合には, 同期合成モデルや通信合成モデルをモデル検査器により自動的に生成することができる. 同期合成は, 2つの疾患においてある症状が同期して起こる場合などの表現に有効である. 通信合成は, 1つの疾患におけるある症状が引き金となって別の疾患の症状を引き起こす場合などの記述に有効であろう. これらの実装問題も含めて, 本研究の今後の課題は, 実的な臨床推論・知識モデルを生成して検証するための自動化ツールを開発することである.

最後に, 本研究に関連する参考文献リストを示す. 矛盾許容時間論理および矛盾許容モデル検査技術に関する情報については文献 [6], [10], [14], [15], [18], [19], [21], [23], [24], [29], 階層時間論理および階層モデル検査技術に関する情報については文献 [15], [16], [20], [25], [27], [28], [30], そして確率時間論理および確率モデル検査技術に関する情報については文献 [1], [2], [3], [4], [11], [12], [17], [23], [27], [32] が参考になる. また, 矛盾許容モデル検査技術, 階層モデル検査技術, 確率モデル検査技術およびこれらの組み合わせに関する簡単なサーベイとしては [17] を参照されたい.

謝辞 本研究は一般財団法人日本産業科学研究所研究助成金および JSPS 科研費 (JP18K11171, JP16KK0007) の援助を受けた.

## 参考文献

- [1] A. Aziz, et. al., It usually works: The temporal logic of stochastic systems, Proceedings of the 7th International Conference on Computer Aided Verification, Lecture Notes in Computer Science 939, pp. 155-165, 1995.
- [2] C. Baier, et. al., Model Checking Probabilistic Systems, In: Handbook of Model Checking, pp. 963-999, 2018.
- [3] C. Baier and M. Kwiatkowska, Model checking for a probabilistic branching time logic with fairness, Distributed Computing 11, pp. 125-155, 1998.
- [4] A. Bianco and L. de Alfaro, Model checking of probabilistic and nondeterministic systems, Proceedings of the 15th Conference on Foundations of Software Technology and Theoretical Computer Science, Lecture Notes in Computer Science 1026, pp. 499-513, 1995.
- [5] R. Cavada, et. al., NuSMV 2.6 user manual, 144 pages, 2015.
- [6] D. Chen and J. Wu, Reasoning about inconsistent concurrent systems: A non-classical temporal logic, Proceedings of the 32nd Conference on Current Trends in Theory and Practice of Computer Science, Lecture Notes in Computer Science 3831, pp. 207-217, 2006.
- [7] E.M. Clarke, O. Grumberg, and D.A. Peled, Model checking, The MIT Press, 1999.
- [8] E.M. Clarke and E.A. Emerson, Design and synthesis of synchronization skeletons using branching time temporal logic, Lecture Notes in Computer Science 131, pp. 52-71, 1981.
- [9] E.M. Clarke, T.A. Henzinger, H. Veith, and R. Bloem (editors), Handbook of Model Checking, Springer, 2018.
- [10] S. Easterbrook and M. Chechik, A framework for multi-valued reasoning over inconsistent viewpoints, Proceedings of the 23rd International Conference on Software Engineering, pp. 411-420, 2001.
- [11] H. A. Hanson, Time and Probability in Formal Design of Distributed Systems (Real-Time Safety Critical Systems, Volume 1), Amsterdam, Elsevier, 1994.
- [12] H. Hansson and B. Jonsson, A logic for reasoning about time and reliability, Formal Aspects of Computing 6 (5), pp. 512-535, 1994.
- [13] G.J. Holzmann, The SPIN Model Checker: Primer and Reference Manual, Addison-Wesley, 2006.
- [14] N. Kamide, Extended full computation-tree logics for paraconsistent model checking, Logic and Logical Philosophy 15 (3), pp. 251-276, 2006.
- [15] N. Kamide, Inconsistency-tolerant temporal reasoning with hierarchical information, Information Sciences 320, pp. 140-155, 2015.
- [16] N. Kamide, Logical foundations of hierarchical model checking, Data Technologies and Applications 52 (4), pp. 539-563, 2018.
- [17] N. Kamide and J.P.A. Bernal, Towards locative inconsistency-tolerant hierarchical probabilistic CTL model checking: Survey and future work, Proceedings of the 11th International Conference on Agents and Artificial Intelligence, Volume 2, pp. 869-878, 2019.
- [18] N. Kamide and K. Endo, Logics and translations for inconsistency-tolerant model checking, Proceedings of the 10th International Conference on Agents and Artificial Intelligence, Volume 2, pp. 191-200, 2018.
- [19] N. Kamide and K. Endo, Foundations of inconsistency-tolerant model checking: Logics, translations, and examples, Lecture Notes in Computer Science 11352, pp. 312-342, 2018.
- [20] N. Kamide and K. Kaneiwa, Extended full computation-tree logic with sequence modal operator: Representing hierarchical tree structures, Proceedings of the 22nd Australasian Joint Conference on Artificial Intelligence, Lecture Notes in Artificial Intelligence 5866, pp. 485-494, 2009.
- [21] N. Kamide and K. Kaneiwa, Paraconsistent negation and classical negation in computation tree logic, Proceedings of the 2nd International Conference on Agents and Artificial Intelligence, Volume 1, pp. 464-469, 2010.
- [22] N. Kamide and D. Koizumi, Combining paraconsistency and probability in CTL, Proceedings of the 7th International Conference on Agents and Artificial Intelligence, Volume 2, pp. 285-293, 2015.
- [23] N. Kamide and D. Koizumi, Method for combining paraconsistency and probability in temporal reasoning, Journal of Advanced Computational Intelligence and Intelligent Informatics 20, pp. 813-827, 2016.
- [24] N. Kamide and H. Wansing, A paraconsistent linear-time temporal logic, Fundamenta Informaticae 106 (1), pp. 1-23, 2011.
- [25] N. Kamide and R. Yano, Logics and translations for hierarchical model checking, Proceedings of the 21st International Conference on Knowledge-Based and Intelligent Information & Engineering Systems, Procedia Computer Science 112, pp. 31-40, 2017.
- [26] N. Kamide and N. Yamamoto, Inconsistency-tolerant hierarchical probabilistic computation tree logic and its application to model checking, Proceedings of the 13rd International Conference on Agents and Artificial Intelligence, To appear, 2021.
- [27] N. Kamide and Y. Yano, Towards hierarchical probabilistic CTL model checking: Theoretical foundation, Proceedings of the 11th International Conference on Agents and Artificial Intelligence, Volume 2, pp. 762-769, 2019.
- [28] K. Kaneiwa and N. Kamide, Sequence-indexed linear-time temporal logic: Proof system and application, Applied Artificial Intelligence 24 (10), pp. 896-913, 2010.
- [29] K. Kaneiwa and N. Kamide, Paraconsistent computation tree logic, New Generation Computing 29 (4), pp. 391-408, 2011.
- [30] K. Kaneiwa and N. Kamide, Conceptual modeling in full computation-tree logic with sequence modal operator, International Journal of Intelligent Systems 26 (7), pp. 636-651, 2011.
- [31] N. Kobayashi, T. Shimizu and A. Yonezawa, Distributed concurrent linear logic programming, Theoretical Computer Science 227, pp. 185-220, 1999.
- [32] M. Kwiatkowska, G. Norman, and D. Parker, PRISM 4.0: Verification of probabilistic real-time systems, Proceedings of the 23rd International Conference on Computer Aided Verification, Lecture Notes in Computer Science 6806, pp. 585-591, 2011.
- [33] G. Priest, Paraconsistent logic, Handbook of Philosophical Logic (Second Edition), Vol. 6, D. Gabbay and F. Guenther (eds.), Kluwer Academic Publishers, Dordrecht, pp. 287-393, 2002.
- [34] Y. Yano, A proposal for formal clinical reasoning verification techniques using model checkers (in Japanese), Bachelor thesis, Teikyo University, Faculty of Science and Engineering, Department of Information and Electronic Engineering, 55 pages, 2020.
- [35] Lung cancer, Wikipedia, 2020.  
[https://en.wikipedia.org/wiki/Lung\\_cancer](https://en.wikipedia.org/wiki/Lung_cancer)
- [36] Schizophrenia, Wikipedia, 2020.  
<https://en.wikipedia.org/wiki/Schizophrenia>