

反証的 CTL モデル検査の拡張とその応用

神戸勢大[†] 石野寛人[†] 楊奕偉[†] Khoo Nee Kwan[†] 上出哲広[†][†] 帝京大学理工学部情報電子工学科

概要

本研究では反証的 CTL モデル検査の拡張とその応用を提案した。反証的 CTL モデル検査は、従来の標準的な CTL モデル検査および矛盾許容 CTL モデル検査を統合・一般化したモデル検査である。本研究では反証的 CTL モデル検査の拡張として、矛盾許容性、階層性および不確実性を同時に扱える拡張を提案した。そして、この拡張反証的 CTL モデル検査の応用例として、臨床推論検証の具体例を示した。本稿の内容は、著者らによる論文 [12, 10] の内容を発展させたものである。また、論文 [12] の内容を一部含んでいる。

1 反証的 CTL モデル検査

モデル検査 (model checking) は、システムの数理的なモデルが期待する性質を持っている否かを判定する技術である。モデル検査は、主にソフトウェアを検証するために広く使用されている [5, 6]。CTL モデル検査 (CTL model checking) は、モデル検査の基盤となる論理として計算木論理 (CTL: Computation Tree Logic) を使用するモデル検査である [5]。矛盾許容 CTL モデル検査 (inconsistency-tolerant CTL model checking) は、CTL モデル検査の拡張であり、矛盾許容性を適切に扱うことができるモデル検査である [7, 4, 11]。反証的 CTL モデル検査 (falsification-aware CTL model checking) は、CTL モデル検査および矛盾許容 CTL モデル検査を統合・一般化したモデル検査である [12]。また、反証的 CTL モデル検査は、古典論理に対して Kamide によって導入された反証的意味論 [9] の考え方を取り入れた CTL モデル検査である。反証的 CTL モデル検査では、反証的 Kripke 意味論を使用したモデル検査を実行できる。反証的 Kripke 意味論を使用したモデル検査では、標準的なモデル検査で使用している充足関係 (ここでは \models と表示する) に加えて反証的充足関係 (ここでは \models^- と表示する) を使用することが可能である。反証的充足関係を使用することにより、否定を伴う検査式 (否定論理式) に対する検証と反証を同時に実行することができる。これにより、否

定を扱う検証・反証をより適切に実行できる可能性がある。

2 反証的 CTL モデル検査の拡張

本研究では反証的 CTL モデル検査の拡張として、さらに階層 CTL モデル検査 (hierarchical CTL model checking) [8] および確率 CTL モデル検査 (probabilistic CTL model checking) [1, 3, 2] を組み合わせた拡張を提案した。これにより、矛盾許容 CTL モデル検査、階層 CTL モデル検査、確率 CTL モデル検査および反証的 CTL モデル検査を統合した拡張反証的 CTL モデル検査が実現可能になる。このような拡張反証的 CTL モデル検査では、矛盾許容 CTL モデル検査で使用されている論理である ICTL (Inconsistency-tolerant Computation Tree Logic) [11]、階層 CTL モデル検査で使用されている論理である sCTL (Sequential Computation Tree Logic) [8] および確率 CTL モデル検査で使用されている論理である pCTL (Probabilistic Computation Tree Logic) [1, 3] を組み合わせた拡張論理に対する反証的 Kripke 意味論を使用する。ここで、ICTL、sCTL および pCTL はそれぞれ CTL の拡張である。このような拡張論理は、[10] で提案されている拡張論理である IHPCTL (Inconsistency-tolerant Hierarchical Probabilistic Computation Tree Logic) の部分体系となる。IHPCTL および拡張反証的 CTL のベースとなる拡張論理では、矛盾許容性を表現するための矛盾許容否定結合子 \sim 、階層構造を表現するための階層演算子 $[b]$ および不確実性を表現するための確率演算子 $P_{\leq x}$ 、 $P_{\geq x}$ 、 $P_{< x}$ および $P_{> x}$ を使用することができる。また、これら論理は、 $(\alpha \wedge \sim \alpha) \rightarrow \beta$ の形の公理を排除した論理である。このことは、これら論理が矛盾許容論理の一種であることを意味する。拡張反証的 CTL モデル検査で使用する反証的 Kripke 意味論は、矛盾許容論理で使用されている Kripke 意味論の考え方を取り入れた意味論である。そのため、この意味論では、矛盾許容性を適切に扱うことができる。

3 反証的 CTL モデル検査の拡張の応用

本研究では、拡張反証的 CTL モデル検査の応用例として、図 1 に示すような家族性大腸線種症診断モデルに対する臨床推論検証の具体例を示した。家族性大腸腺腫症 (FAP: Familial Adenomatous Polyposis) は大腸に 100 個以上のポリープが発生する遺伝的な疾

Extended falsification-aware CTL model checking and its applications

[†] Seidai Kanbe (1832143r@stu.teikyo-u.ac.jp)

[†] Hiroto Ishino (183204vm@stu.teikyo-u.ac.jp)

[†] Yiwei Yang (183146u5@stu.teikyo-u.ac.jp)

[†] Khoo Nee Kwan (nkkhoo555@gmail.com)

[†] Norihiro Kamide (drnkamide08@kpd.biglobe.ne.jp)

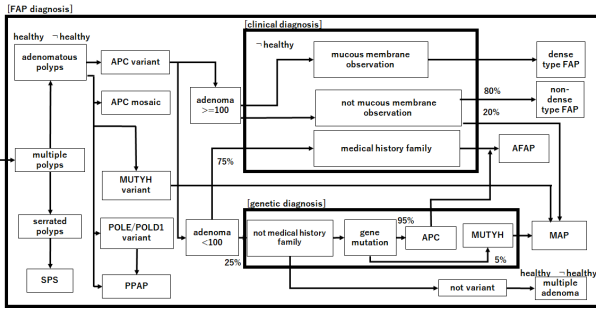


図 1: 家族性大腸線種症診断モデル

患である。ポリープが発生し始めるのは 10 歳前後であり、以降は時間の経過とともに数と大きさが增大する。このポリープから大腸癌が発生する。家族性大腸腺腫症は、常染色体優性遺伝の遺伝疾患であり、原因遺伝子は APC 遺伝子であることが判明している。家族性大腸腺腫症は、以下の 3 つに分類される。古典的 FAP、軽症型 FAP (AFAP) および MUTYH 関連型 FAP (MAP)。

図 1 に示されるような家族性大腸線種症診断モデルにおいては、「健康でかつ健康でない」というような「未病」の状態を表すために矛盾許容否定結合子が必要となる。また、病気の階層や分類を表現するための階層演算子が必要となる。さらに、病状進行の可能性を示すための不確実性を表現するために確率演算子が必要となる。これらの演算子を、提案した拡張反証的 CTL モデル検査で扱うことができる。以下では、矛盾許容性に関連する検証の例を示す。拡張反証的 CTL モデル検査では、矛盾許容否定結合子 \sim や反証的充足関係 \models を用いることにより、矛盾許容性を適切に扱うことができる。

以下のステートメントを拡張反証的 CTL モデル検査によって検証することができる。なお、このステートメントは当該モデルにおいて真である。

“Is there a state in which a person is both healthy and unhealthy (i.e., not yet ill), has less than 100 adenomas, and had a medical examination with genetic diagnosis?”

このステートメントは拡張反証的 CTL モデル検査の論理式を用いて以下のように記述することができる。

$$EF(\text{healthy} \wedge \sim \text{healthy} \wedge (\text{adenoma} < 100) \wedge \text{geneticDiagnosis})$$

ここで、E は CTL のパス量化演算子であり、F は「eventually in the future」を表す時間演算子である。

謝辞 本研は JSPS 科研費 (番号: JP18K11171 および JP16KK0007) の援助を受けた。

参考文献

- [1] A. Aziz, V. Singhal, and F. Balarin, 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, L. de Alfaro, V. Forejt, and M. Kwiatkowska, Model Checking Probabilistic Systems, In: Handbook of Model Checking, pp. 963-999, 2018.
- [3] 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.
- [4] 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.
- [5] 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.
- [6] E.M. Clarke, T.A. Henzinger, H. Veith, and R. Bloem (editors), Handbook of Model Checking, Springer, 2018.
- [7] 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.
- [8] N. Kamide, Logical foundations of hierarchical model checking, Data Technologies and Applications 52 (4), pp. 539-563, 2018.
- [9] N. Kamide, Falsification-aware semantics and sequent calculi for classical logic, Journal of Philosophical Logic, Published as an online first article, 2021.
- [10] N. Kamide, Inconsistency-tolerant hierarchical probabilistic CTL model checking: Logical foundations and illustrative examples, International Journal of Software Engineering and Knowledge Engineering, To appear, 2022.
- [11] 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.
- [12] N. Kamide and S. Kanbe, Falsification-aware semantics for CTL and its inconsistency-tolerant subsystem: Towards falsification-aware model checking, Proceedings of the 14th International Conference on Agents and Artificial Intelligence, To appear, 2022.