| | | | |
|---|---|---|---|
| 氏名 | Yati Phyo | | |
| 学位の種類 | 博士（情報科学） | | |
| 学位記番号 | 博情第 478 号 | | |
| 学位授与年月日 | 令和 4 年 9 月 22 日 | | |
| 論文題目 | A Divide & Conquer Approach to Leads-to and Conditional Stable Model Checking | | |
| 論文審査委員 | 主査 緒方和博 | 北陸先端科学技術大学院大学 | 教授 |
| | 平石邦彦 | 同 | 教授 |
| | 石井大輔 | 同 | 准教授 |
| | 岡野浩三 | 信州大学 | 教授 |
| | 中村正樹 | 富山県立大学 | 准教授 |

**論文の内容の要旨**

Model checking is one of the most successful computer science achievements in the last few decades. This is why Edmund M. Clarke, E. Allen Emerson and Joseph Sifakis were honored with 2007 A.M. Turing Award for their role in developing model checking into a highly effective verification technology. Model checking has been widely adopted in industries, especially in hardware ones. There are still some. issues to tackle in model checking, one of which is the notorious state explosion. Many techniques to mitigate the state explosion, such as partial order reduction and abstraction, have been devised. Because these existing techniques are not too enough to deal with the state explosion, however, it is still worth tackling it.

To mitigate the state space explosion problem to some extent, we propose a divide & conquer approach to model checking (DCA2MC) leads-to properties and conditional stable properties, where leads-to properties are expressed in $\varphi_1 \rightsquigarrow \varphi_2$ and conditional stable properties are expressed in $\varphi_1 \rightsquigarrow \Box\varphi_2$, where $\varphi_1$ and $\varphi_2$ are state propositions or classical propositional formulas. Chandy and Misra designed a temporal logic called UNITY in which the leads-to temporal connective plays an important role and demonstrated that many important systems requirements can be expressed as leads-to properties. Moreover, Dwyer et al. showed some statistics on the usage distribution of the various patterns in property specifications in which the leads-to property (or the response pattern) had the highest proportion. Conditional stable properties can be used to express core requirements in self-stabilizing systems, which were first introduced by Dijkstra and became a very important concept in fault tolerance to design robust systems. Thus, it is worth focusing on leads-to and conditional stable properties. DCA2MC divides an original leads-to (or conditional stable) model checking problem into multiple smaller model checking problems and tackles each smaller one. We prove a theorem that the multiple smaller model checking problems are equivalent to the original leads-to (or conditional stable) model checking problem. An algorithm is constructed based on the theorem to support model checking leads-to (or conditional stable) properties by DCA2MC. A support tool is developed in Maude, a rewriting logic-based specification/programming language and system, to support the technique based on the algorithm for each of leads-to and conditional stable properties. Some experiments are then conducted with the support tools to demonstrate that our tools/techniques can mitigate the state space explosion to some extent.

Both leads-to and conditional stable properties can be expressed as linear temporal logic (LTL) formulas that are very similar. However, how to deal with the two classes of properties with DCA2MC is so different that we need to prove the correctness of DCA2MC for the two classes of properties in

two different ways and come up with two different algorithms for the two classes of properties, from which the two support tools are built. Note that because the architecture of the tools is well-designed, many components (data structures and functions) are shared by the two tools. This is because it suffices to take a look at the top state of an infinite sequence $\pi$ of states so as to check if a state proposition $\varphi_2$ holds for $\pi$, while it is necessary to take a look at all states in $\pi$ in order to check if an LTL formula $\Box\varphi_2$, where $\varphi_2$ is a state proposition, holds for $\pi$. One piece of our future work is to extend DCA2MC for the other classes of LTL properties and then come up with a unified DCA2MC for all LTL properties.

**Keywords:** LTL model checking, leads-to properties, conditional stable properties, state space explosion, divide & conquer.

**論文審査の結果の要旨**

状態空間爆発は、モデル検査におけるもっとも深刻な問題である。メモリ不足によりモデル検査不能に陥る。モデル検査をソフトウェア産業における日業業務で使えるような技術にまで成熟させるには状態空間爆発の緩和は不可欠である。２つの重要な性質のクラス（leads-to 性と条件付き安定性）に対し状態空間爆発を緩和する技術を提案している。leads-to は、時相論理である UNITY の設計者である Chandy と Misra 等により多くのシステム要件を表現出来ることが知られている。条件付き安定性は、耐故障において重要である自己安定システムの性質を記述可能である。限定した性質に限った技術であるが、現場において十分に有効であることの証左である。leads-to は、あることが起こると、有限時間内に別のあることが必ず起こるということである。たとえば、相互排除プロトコルの非排斥性（際どい領域に入りたいプロセスは有限時間内にその領域に入る）を記述可能である。条件付き安定性は、あることが起こると、有限時間内に別のことが必ず起こりある状態になりその状態が永遠に継続するということである。自己安定システムの重要な要件（異常状態に陥ったら有限時間内に正常状態に戻り正常状態で永遠に安定する）を記述可能である。提案方法は、到達可能状態空間を複数の層に分割し、複数の部分空間を作り、各々の部分空間を独立に処理するということが基本アイディアである。各々の部分空間を独立に処理することと、到達可能状態空間を直接モデル検査することが等価であることが証明されている。基本アイディアに基づきアルゴリズムが設計され、アルゴリズムに基づき支援ツールが Maude の自己反映計算機能を用いて実装されている。Maude はプログラミング言語のみならず形式仕様言語としても利用可能で、モデル検査器を有している。支援ツールの実装では Maude のモデル検査器を内部で支援ツールの基本コンポーネントとして利用している。複数の事例を用いたモデル検査実験により、Maude のモデル検査では状態空間爆発によるモデル検査不能に陥るモデル検査に対して、支援ツールではモデル検査を正常に終了することが可能であることが確認されており、提案技術・支援ツールの有効性が実証されている。事例研究では Maude を形式仕様言語として用いている。

以上、本論文は、モデル検査におけるもっとも深刻な問題である状態空間爆発を、２つの重要な性質のクラス（leads-to 性と条件付き安定性）に対し、緩和する方法を提案すると共に正しさを証明し、その方法に基づきアルゴリズムを設計し、アルゴリズムに基づき支援ツールを実装し、事例研究により提案方法・ツールの有効性を確認したものであり、学術的に貢献するところが大きい。よって博士（情報科学）の学位論文として十分価値あるものと認めた。