様　式　Ｃ－１９、Ｆ－１９－１、Ｚ－１９　（共通）

科学研究費助成事業　　研究成果報告書

令和　元 年　 ６ 月 １４ 日現在

機関番号：３２５０３
研究種目：若手研究(B)
研究期間：2016～2018
課題番号：１６Ｋ２１３３５

研究課題名（和文）メモリ一貫性モデルを考慮したプログラム検証の統一理論の構築とその検査器の実装

研究課題名（英文）A unified theory of program verision with memory consistency models and its implementations

研究代表者
安部　達也（ABE, Tatsuya）

千葉工業大学・人工知能・ソフトウェア技術研究センター・上席研究員

研究者番号：５０５４７３８８

交付決定額（研究期間全体）：（直接経費）　　3,200,000円

研究成果の概要（和文）：メモリ一貫性モデルを考慮することでおこなえる、状態の統合やリオーダリングされる命令の制御による状態爆発の回避に関する方法を提案しました。Non-MCAなメモリ一貫性モデルにセンシティブなプログラムであるIRIWをも検証できるプログラム論理を構築しました。モデル検査器SPINをメモリ一貫性モデル込みで検査できるようにライブラリを開発しました。並列プログラミング言語XcalableMPのメモリ一貫性モデルの策定をおこないました。開発しているモデル検査器で並列コピーガベージコレクションの検査をおこないました。Non-MCAなメモリ一貫性モデルでのモデル検査で有効なLDRFという概念を提唱しました。

研究成果の学術的意義や社会的意義
プログラムを世に出す前にその安全性を検証する手法を研究・開発しました。本研究で開発した手法により、既存手法に対して検査時間の短縮と扱えるプログラムの範囲の拡張が実現されました。また、並列計算環境が普及した現代において広く使われているアルゴリズムである並行コピーガベージコレクションの検査を実演することで私たちの手法の有効性を示しました。研究したことは論文というかたちで出版されていることによりその内容を公知のものとし、また、開発したツールとプログラムをすべて公開したことでこれらを広く使用可能にしました。

研究成果の概要（英文）：I did joint researches with Tomoharu Ugawa, Toshiyuki Maeda, and Kosuke Matsumoto. We proposed an optimization of model checking with memory consistency models (MCMs) that reduces the number of states by integrating multiple states and controlling reordering of instructions according to input MCMs. We constructed a program logic with MCMs that Independent Reads Independent Writes with non-multi-copy atomicity (non-MCA). We developed a SPIN library that supports MCMs. We designed an MCM for a directive-based parallel programing language XcalableMP. We demonstrated model checking of concurrent copy garbage collecton algorithms. We proposed local data race freedom (LDRF), a property of a program. We also proposed an optimization of model checking with MCMs that reduces the number of states of programs that enjoy LDRF with non-MCA MCMs.

We wrote some papers about these results. All of them were published. We also released tools and programs that we developed in this study.

研究分野：メモリ一貫性モデルを考慮したプログラム検証、特にモデル検査とプログラム論理を用いた静的検証

キーワード：プログラム検証　メモリ一貫性モデル　モデル検査　ガベージコレクション　プログラム論理　Non-multi-copy atomicity　IRIW

様　式　Ｃ－１９、Ｆ－１９－１、Ｚ－１９、ＣＫ－１９（共通）

１．研究開始当初の背景

計算機を動作させるための指示書であるプログラム中の誤り（以下、バグ）はしばしば深刻な問題を引き起こします。このようなバグをあらかじめ発見・除去するための理論の一つにプログラム検証があります。しかし、従来のプログラム検証はメモリ一貫性モデルを扱っていませんでした。メモリ一貫性モデルとは（プログラミング言語・アーキテクチャによってそれぞれ異なる）計算機内の共有データが各スレッドたちにどのように見えるかを規定したルールです。メモリ一貫性モデルはスレッドたちのコマンド実行タイミングという人間にとっては把握しづらいものに言及することから、特に制約のゆるいメモリ一貫性モデル下でプログラム検証の支援なしにバグを含まないプログラムを開発することは困難です。しかし、現存する並列計算機を効率よく利用するためには、困難であるからといって制約のゆるいメモリ一貫性モデルを無視することはできません。例えば、並行ガーベジコレクション（以下、GC）を実装したプログラムが TSO や PSO といったメモリ一貫性モデルの下で正常・不正に動作するかといったことが議論されるようになってきていました。

２．研究の目的

この問題を解決すべく、メモリ一貫性モデルを扱うプログラム検証に関する研究が見られるようになってきていました。しかし、それらが提供する理論はメモリ一貫性モデルが固定されているか、あるいは、入力として外から与えられるものであったとしても、その扱えるメモリ一貫性モデルには制限がありました。

そこで、研究代表者は並行・並列プログラミング言語の基礎理論とその検証理論・検査器の実装の経験を踏まえて、メモリ一貫性モデルを考慮したプログラム検証、特に、モデル検査とプログラム論理の統一的な理論の構築とその検査器の実装をおこなうことにしました。

３．研究の方法

プログラム論理は、プログラムが正しく書かれている時、その正しさに保証を与えるもの（証明）を書き下すための論理です。メモリ一貫性モデルの相違を特徴づけるプログラム一式を検証対象とすることで、以前に構築した理論のブラッシュアップをおこない、グローバルタイムを仮定しないプログラム意味論に対して健全な並行プログラム論理の構築を試みました。

また、比較的大きなプログラムを検査できるようにしてきたものの、さらに大きな並列分散アルゴリズムを実装したプログラムを検査したいという要望に応えるべく、メモリ一貫性モデルを考慮していないモデル検査で行われている最適化手法をメモリ一貫性モデルを考慮したモデル検査に適用できるように調節・ブラッシュアップすることで、研究代表者が開発している検査器の改良をおこないました。

GC の専門家である鵜川博士（高知工科大学）と共同研究をおこないました。GC の分野では、既存・新規の GC アルゴリズムが制約のゆるいメモリ一貫性モデル下では要求される仕様を満たしているかということが問題となっています。既存または新規の GC アルゴリズムが制約のゆるいメモリ一貫性モデルで要求される仕様を満たしているかの試行と、また、関連することを議論しました。また、メモリ一貫性モデルを考慮しないモデル検査の最適化手法を今回のメモリ一貫性モデルを考慮したモデル検査への適用をおこないました。

GC のモデル検査は制約の強いメモリ一貫性モデル下であっても状態爆発の問題により困難であることが知られています。検査を可能にするためには既存の（いくつかある）状態数削減手法をメモリ一貫性モデルを考慮しても行えるようにしなければなりません。状態数削減手法はアルゴリズムの理解も必要なため、鵜川博士と共同でこれをおこないました。

また、研究代表者は 2011 年より国産の並列プログラミング言語 XcalableMP の仕様策定ワーキンググループに属しています。ワーキンググループでは、 XcalableMP でどのようなメモリ一貫性モデルを採用したら高性能と高生産性とを両立できるかを議論し、仕様書への反映を行っています。これに記載されているメモリ一貫性モデルは前述の形式言語で記述されており、また、載っているサンプルプログラムは前述の検査器で検査済みです。本研究で得た成果を XcalableMP に反映していくことで研究成果の実世界への迅速な還元と、そのフィードバックから研究成果のブラッシュアップをおこないました。

４．研究成果

メモリ一貫性モデルを考慮することで行える、状態の統合による状態爆発の回避に関する論文を執筆・投稿し、国際会議 The 2nd International Symposium on Dependable Software

Engineering: Theories, Tools and Applications に採択されました[1-9,2-6]。また、Xu 博士（澳門大学）の提案した（メモリ一貫性モデルを考慮していない）並行プログラム論理を拡張することでメモリ一貫性モデルを考慮した新しい並行プログラム論理を構築しました。このプログラム論理は既存の全ての並行プログラム論理が扱うことができない(non‐)multiple‐copy‐atomicity についてパラメタライズされており、既存のプログラム論理で扱うことが容易でないことが知られている並行プログラム Independent Reads Independent Writes の検証を行うことに成功しました。このプログラム論理に関する論文を投稿し、Journal of Information Processing と国際会議 The 14th Asian Symposium on Programming Languages and Systems にそれぞれ採択されました[1-6,1-8,2-2]。また、共同研究を行っている鵜川博士・松元氏（高知工科大学）とメモリ一貫性モデルを考慮する SPIN ライブラリの開発に関する論文を投稿し、FOSE'16 に採択されました[1-7]。また、国産並列プログラミング言語 XcalableMP の仕様について、代表者が執筆した Appendix E (pages 147‐‐152) が 2017 年 1 月 12 日に行われた XcalableMP の仕様策定の規格部会で承認され、Version 1.3 から採用されています[3-1]。

メモリ一貫性モデルを考慮したモデル検査における状態爆発問題に対し、リオーダリングされる命令の制御という解決手法の提案を行い、それを開発しているモデル検査器 McSPIN に実装し、評価をおこないました[3-3]。それらに関して執筆した論文は、国際会議 9th Working Conference on Verified Software: Theories, Tools, and Experiments に採択されました[1-5,2-5]。Garbage Collection の代表的な教科書である Jones et al.「The Garbage Collection Handbook」に載っている並行 GC を McSPIN で検査し、知られていないバグ 2 つを発見し、それに関する論文が国際会議 The 32nd OOPSLA に採択されました[1-4]。また、[1-6] に構築したプログラム論理においてもプログラムの表現として利用されている directed acyclic graph を検査するモデル検査器 VeriDAG を開発し、これに関する論文が The 13th Haifa Verification Conference に採択されました[1-3,2-4]。メモリ一貫性モデルを考慮したモデル検査器 SPIN 用のライブラリに関する、共同研究をしている松元氏・鵜川博士との共著論文が Journal of Information Processing に採択されました[1-2,2-3]。

メモリ一貫性モデルを考慮したモデル検査を高速に実行可能にするプログラム中の全スレッドに対して定義される性質 data race freedom をスレッドのサブセットに対する概念に拡張することで local data race freedom という性質を提唱しました。この性質は、メモリ一貫性モデルを考慮したプログラム検証において特徴的なプログラムである Independent Reads Independent Writes（IRIW）や、全てのミューテータが data race free なプログラムであることを仮定した場合、ミューテータたちとコレクタからなる並行コピーガベージコレクションのモデルが満たす性質の一つです。また、local data race freedom を利用したモデル検査の状態数削減手法、memory sharing optimization を提唱しました。この最適化を、IRIW と昨年度に開発を開始したモデル検査器 VeriDAG に実装し、本研究課題で継続的に開発している McSPIN の評価のための実験に使用した並行コピーガベージコレクションのモデルを使用して実験をおこない、その有効性を確認しました。これらすべてに関する内容を記載した論文を執筆し、The 23rd International SPIN Symposium に採択されました[1-1]。モデル検査器 VeriDAG と実験に使用したモデルすべてを公開しました[3-2,2-1]。

５．主な発表論文等

〔雑誌論文〕（計９件）

[1-1] Tatsuya Abe. Local data race freedom with non-multi-copy atomicity. In Proceedings of SPIN, Vol. 10869 of LNCS, pp. 196-215, June 2018.査読あり

[1-2] Kosuke Matsumoto, Tomoharu Ugawa, and Tatsuya Abe. Improvement of a library for model checking under weakly ordered memory model with SPIN. Journal of Information Processing, Vol. 26, pp. 314-326, 2018. 査読あり

[1-3] Tatsuya Abe. A verifier of directed acyclic graphs for model checking with memory consistency models. In Proceedings of HVC, Vol. 10629 of LNCS, pp. 51-66, November 2017. 査読あり

[1-4] Tomoharu Ugawa, Tatsuya Abe, and Toshiyuki Maeda. Model checking copy phases of concurrent copying garbage collection with various memory models. Proceedings of the ACM on Programming Languages, Vol. 1, No. OOPSLA:53, pp. 1-26, 2017. 査読あり

[1-5] Tatsuya Abe, Tomoharu Ugawa, and Toshiyuki Maeda. Reordering control approaches to state explosion in model checking with memory consistency models. In Proceedings of VSTTE, Vol. 10712 of LNCS, pp. 170-190, July 2017. 査読あり

[1-6] Tatsuya Abe and Toshiyuki Maeda. Concurrent program logic for relaxed memory consistency models with dependencies across loop iterations. Journal of Information Processing, Vol. 25, pp. 244-255, 2017. 査読あり

[1-7] 松元稿如、鵜川始陽、安部達也。メモリモデルを考慮したメモリアクセス命令を提供す

る SPIN 用ライブラリ。第 23 回ソフトウェア工学の基礎ワークショップ(FOSE)、 pp. 63-72 、琴平、 12 月、 2016 年。査読あり
[1-8] Tatsuya Abe and Toshiyuki Maeda. Observation-based concurrent program logic for relaxed memory consistency models. In Proceedings of APLAS, Vol. 10017 of LNCS, pp. 63-84, November 2016. 査読あり
[1-9] Tatsuya Abe, Tomoharu Ugawa, Toshiyuki Maeda, and Kousuke Matsumoto. Reducing state explosion for software model checking with relaxed memory consistency models. In Proceedings of SETTA, Vol. 9984 of LNCS, pp. 118-135, October 2016. 査読あり

〔学会発表〕(計６件)
[2-1] 安部達也。局所的データ非競合なプログラムの観測的同値。第 29 回代数,論理,幾何と情報科学研究集会、熊本、 8 月、 2018 年。
[2-2] 安部達也。並行プログラム論理の証明の導出に関する諸問題。第 35 回記号論理と情報科学、札幌、 8 月、2018 年。
[2-3] 松元稿如、鵜川始陽、安部達也。 SPIN で弱いメモリ順序のメモリモデルでのプログラムの実行をモデル検査するためのライブラリの改良。第 115 回情報処理学会プログラミング研究会、秋田、 7 月、 2017 年。
[2-4] 安部達也。メモリ一貫性モデルを考慮したモデル検査のためのプログラムグラフ検査器の Haskell 実装。ラムダ計算と論理の早春セミナー、草津、 3 月、 2017 年。
[2-5] 安部達也。命令のリオーダリングを許すモデル検査における反復的探索。第 33 回記号論理と情報科学、名古屋、 8 月、 2016 年。
[2-6] 安部達也。メモリ一貫性モデルを考慮したモデル検査器の開発。第 8 回学際大規模情報基盤共同利用・共同研究拠点シンポジウムポスター発表、東京、 7 月、 2016。

〔その他〕
仕様書
[3-1] XcalableMP Specification Working Group. XcalableMP Language Specification. Version 1.3, August 2017. Take charge of writing Appendix E. pages 147-151.
ホームページ等
[3-2] Tatsuya Abe. VeriDAG: a verifier of program graphs for model checking with memory consistency models. 2016. https://bitbucket.org/abet/veridag
[3-3] Tatsuya Abe, Toshiyuki Maeda, and Tomoharu Ugawa. McSPIN: a model checker with memory consistency models. 2013. https://bitbucket.org/abet/mcspin