

平成 2 8 年 5 月 3 1 日現在

機関番号 : 8 2 4 0 1

研究種目 : 若手研究(B)

研究期間 : 2013 ~ 2015

課題番号 : 2 5 7 3 0 0 5 0

研究課題名 (和文) ゆるいメモリー貫性モデルのための簡潔で柔軟な型システムに関する研究

研究課題名 (英文) Study on a simple and flexible type system for relaxed memory consistency models

研究代表者

前田 俊行 (Maeda, Toshiyuki)

国立研究開発法人理化学研究所・計算科学研究機構・チームリーダー

研究者番号 : 5 0 4 3 6 5 5 7

交付決定額 (研究期間全体) : (直接経費) 1,400,000 円

研究成果の概要 (和文) : 並列プログラミングにおいてミスが生じやすくなる原因の一つである「メモリー貫性モデル」、特に「ゆるいメモリー貫性モデル」(複数のプログラムを逐次的に交互に実行した場合には起こりえない結果が生じるモデル) におけるプログラムの動作の正しさ (本研究ではプログラムが不正なメモリー操作をしないこと) を検証する手法を設計した。より具体的には、CPUやプログラミング言語によって異なる様々なメモリー貫性モデルを簡潔かつ統一的に扱うことができるようなプログラミング言語の型システムを設計した (従来の手法では、異なる種類のメモリー貫性モデルに対してそれぞれ検証手法を設計する必要があって煩雑であり、実用的でなかった)。

研究成果の概要 (英文) : We designed a program verification method that is able to verify correctness of program behavior (in this research, the memory safety that ensures that programs perform no illegal memory operation) under various memory consistency models, especially under the relaxed ones (where the results of program execution may not match any result that can be obtained from executing multiple programs in a sequential and interleaved manner), which are one of the causes of mistakes in parallel programming. More specifically, we designed a simple and unified type system of a programming language that is able to handle various memory consistency models that may vary from CPU to CPU, and programming language to programming language (the conventional approaches were impractical in the sense that different verification methods have to be designed for each memory consistency model).

研究分野 : プログラミング言語・システムソフトウェア

キーワード : メモリー貫性モデル 型システム プログラム検証 プログラミング言語

1. 研究開始当初の背景

「メモリー貫性モデル」とは、並列に実行される複数のプログラムが共有のメモリーを同時に操作した場合に、そのメモリー操作の結果・影響が各プログラムからどのように観測されるかを表すモデルのことである。このうち「ゆるい」メモリー貫性モデルとは、複数のプログラムを逐次的に実行した場合には観測されないはずの結果・影響が観測されるモデルのことである。

例えば以下のような二つのプログラムが同時に実行された場合について考える(ただし、変数 x および y は二つのプログラムで共有されているものとし、また初期値は両変数とも 0 と仮定する)：

プログラム 1	プログラム 2
$x = 1;$	$y = 1;$
$t1 = y;$	$t2 = x;$

逐次的な(すなわちゆるくない)メモリー貫性モデルにおいては、実行結果は $\{t1 = 1, t2 = 1\}$ もしくは $\{t1 = 0, t2 = 1\}$ もしくは $\{t1 = 1, t2 = 0\}$ のいずれかとなる。ところが、(ある種の)ゆるいメモリー貫性モデルにおいては $\{t1 = 0, t2 = 0\}$ という結果が生じ得る。なぜなら、プログラム 1 による変数 x への書き込み操作の結果がプログラム 2 によって観測されるのが、プログラム 2 による変数 x への読み込み操作より後になり得るため、またプログラム 2 による変数 y への書き込み操作の結果がプログラム 1 によって観測されるのが、プログラム 1 による変数 y への読み込み操作より後になり得るためである。

上述のように、ゆるいメモリー貫性モデルにおいては、従来の逐次的なメモリーの挙動からは予想できない結果が生じ得るため、プログラミングにおいてミスをしやすい。なぜなら、逐次的なモデルにおいては正しく動作するプログラムが、ゆるいモデルでは誤った振舞いをするところがあり得るからである。

このため、プログラムの正しさを検証するプログラム解析・検証、とりわけ、ゆるいメモリー貫性モデルに対応した並列プログラム解析・検証が重要となる。従来の並列プログラム解析・検証においては、ゆるいメモリー貫性モデルを考慮した研究は多くなかったが、近年、急速に研究が進展している。

しかし、このようなゆるいメモリー貫性モデルを考慮した従来のプログラム解析・検証の研究では、モデルの表現が複雑でプログラム開発者には理解しづらい、また特定のモデルにのみ対応していて他のモデルに応用が可能な自明ではなく、柔軟性に欠ける等といった問題があった。

2. 研究の目的

本研究は、上記研究背景に述べた従来のプログラム解析・検証の研究の問題点に対応した、プログラムのメモリー安全性(プログラム

が不正なメモリー操作を行わないこと)の検証手法を確立することを目的とした。より具体的には、様々な種類のゆるいメモリー貫性モデルを簡潔に表現できる、プログラミング言語の柔軟な型システムの理論を構築することを目的とした。

3. 研究の方法

本研究は、以下の四項目についてそれぞれ順番に、また相互にフィードバックを行いつつ進められた：(1)既存のゆるいメモリー貫性モデルについての調査、(2)様々なゆるいメモリー貫性モデルを簡潔に表現できる型システムの設計、(3)設計した型システムの型安全性の証明、(4)設計した型システムの実用性の評価。以下、それぞれの項目について説明する：

(1) 既存のゆるいメモリー貫性モデルについての調査

新旧各種プロセッサのマニュアル・仕様書等を調査し、それぞれが採用しているメモリー貫性モデルを明らかにした。また、各種分散並列プログラミング言語・フレームワークについても同様にマニュアル・仕様書等を調査し、それぞれのメモリー貫性モデルを明らかにした。

(2) 様々なゆるいメモリー貫性モデルを簡潔に表現できる型システムの設計

まず、項目(1)によって調査した各種のゆるいメモリー貫性モデルの比較を行い、共通化できる部分・できない部分を明らかにした。次にこれにもとづいてモデルの表現のベースとなる抽象機械の設計を行い、更にモデルの表現に必要な(抽象機械上の)制約条件を表現できるような型システムを検討・設計した

(3) 設計した型システムの型安全性の証明

項目(2)で設計した型システムが、実際に型安全性を保証できるかどうかの証明を行った。具体的には、型システム上で表現された制約条件をメモリー貫性モデルの定義とみなしたときに、その制約条件の下での型検査にプログラムがパスしたならば、プログラムのメモリー安全性が(そのメモリー貫性モデルの下で)保証されることを証明した。

(4) 設計した型システムの実用性の評価

項目(2)で設計した型システムを用いて、項目(1)で調査した複数のゆるいメモリー貫性モデルを記述した。またこの記述を利用して、実際に複数のプロセスからなる幾つかのプログラムに対して型検査を行い、プログラムの型安全性の検証や、異なるメモリー貫性モデルの下でどのように型検査の結果が変化するかと比較等を行った。

4. 研究成果

本研究の成果は以下の三項目であり、上記研究目的に述べた当初の目的を概ね達成することができた：(1) 様々なメモリー貫性モデルを統一的に表現できる抽象機械モデル、(2) ベースとなる抽象機械モデル上での様々なメモリー貫性モデルの表現、(3) 様々なメモリー貫性モデルに対応した型システム。以下、それぞれの項目について説明する：

(1) 様々なメモリー貫性モデルを統一的に表現できる抽象機械モデル

上記3の(1)で調査した様々なメモリー貫性モデルを表現できるようなベースとなる抽象機械モデルを設計した。具体的にはこのベースモデルは、複数プロセスそれぞれに対応した状態を持ち、それらのプロセスが独立にプログラムを実行(状態遷移)するもので、プロセス間で共有されるメモリは、個々のプロセスに付随する局所的(すなわち他のプロセスからは直接アクセスできない)メモリによって仮想的に再現される。つまり、ベースモデルとしては共有メモリを持たず、各プロセスの局所的メモリを仮想的な共有メモリのキャッシュとし、プロセス間のメッセージ通信によってこのキャッシュを更新する。

また、様々なメモリー貫性モデルをカバーするために、全てのプロセスの全ての命令(プログラム)実行、メモリ操作、プロセス間通信等が、ベースとなる最低限の制約を除き、任意の順序で入れ替えることを許し、更に、複数のプロセスが異なる入れ替え順序を観測するような並行世界的振舞いも許している。

(2) ベースとなる抽象機械モデル上での様々なメモリー貫性モデルの表現

上記(1)のベースモデルにおいて、3の(1)で調査した様々なメモリー貫性モデルを表現することができた。具体的には、ベースモデルにおいてメモリー貫性モデルは、命令実行・メモリ操作・プロセス間通信等の順序入れ替えに対して何らかの制約を課すことで表現される。例えば、メモリ操作の順序入れ替えは許すが、全てのプロセスで入れ替えの順序は一致させる、また、同一アドレスへの複数のメモリ操作、特定の命令発行に伴うメモリ操作については入れ替えを許さない等の制約によって様々なメモリー貫性モデルを表現することができる。

(3) 様々なメモリー貫性モデルに対応した型システム

上記(1)のベースモデルにおける順序入れ替え、および入れ替えに関する制約条件を考慮したプログラミング言語の型システムの設計を行った。具体的には、並列実行に対応した別名型(alias types)ベースの型付アセンブリ言語をベースとし、存在型

(existential types)をpack/unpackする際、またそれに関連するメモリ操作をする際の型付けにおいて、制約条件を考慮した型付けを行うように設計した。

また、ループや条件分岐に関連するメモリ操作、またそれらを跨るようなメモリ操作において、未執行の可能性のある命令の副作用を考慮するために、ジャンプ命令・分岐命令および存在型のpack/unpackの型付けにおいてこの副作用を考慮するように事前条件等进行检查するようにした。これにより、ループや条件分岐に関するメモリ操作、またそれらを跨るようなメモリ操作も柔軟に型付けすることができるようになった。

上記のとおり、本研究で得られた型システムは本研究の当初の目的を概ね達成したものといえるが、今後の課題としては、得られた型システムでは、同一のプログラムであっても異なるメモリー貫性モデル下においては異なる型情報を与えて型検査をしなければいけないという点があるため、これを改善するということが考えられる(型操作を行う特別な擬似命令を導入して型情報の修正の必要性を少なくする手法の検討を既に行ったが、この場合、異なるメモリー貫性モデルに応じてプログラムを変更して型検査を行う必要が生じるため、更なる改良が必要と考えられる。)

5. 主な発表論文等

[雑誌論文](計1件: 査読あり)

1. Tatsuya Abe and Toshiyuki Maeda: “A General Model Checking Framework for Various Memory Consistency Models”, International Journal on Software Tools for Technology Transfer. In Press. 2016.

[学会発表](計4件)

1. Tatsuya Abe and Toshiyuki Maeda: “Towards a Unified Verification Theory for Various Memory Consistency Models”, In Proc. of the 6th Workshop on Syntax and Semantics of Low-Level Languages. Jul. 7, 2015. Grand Prince Hotel Kyoto, Kyoto, Japan.
2. Tatsuya Abe and Toshiyuki Maeda: “Optimization of a General Model Checking Framework for Various Memory Consistency Models”, In Proc. of the 8th International Conference on PGAS Programming Models. Oct. 7 – 10, 2014. Eugene, Oregon, USA.
3. Tatsuya Abe and Toshiyuki Maeda: “A General Model Checking Framework for Various Memory Consistency Models”, In Proc. of International Workshop on High-Level Parallel Programming Models and Supportive Environments. May. 9, 2014. Phoenix, Arizona, USA.

4. Tatsuya Abe and Toshiyuki Maeda:
“Model Checking with User-Definable
Memory Consistency Models”, In Proc.
of the 7th International Conference on
PGAS Programming Models. Oct. 3 – 4,
2013. Edinburgh, Scotland, UK.

6 . 研究組織

(1)研究代表者

前田 俊行 (Maeda, Toshiyuki)

国立研究開発法人理化学研究所

計算科学研究機構

チームリーダー

研究者番号 : 50436557