

令和元年6月14日現在

機関番号：17104

研究種目：若手研究(B)

研究期間：2015～2018

課題番号：15K15974

研究課題名(和文) 正しさと効率に関する機械的証明を備えた並列プログラミング環境の実現に関する研究

研究課題名(英文) Towards parallel programming environment with certified correctness and complexity

研究代表者

江本 健斗 (Emoto, Kento)

九州工業大学・大学院情報工学研究院・准教授

研究者番号：00587470

交付決定額(研究期間全体)：(直接経費) 3,400,000円

研究成果の概要(和文)：本研究では、正しさのみでなく、その並列計算量(実行時間)をも形式的に保証できるような並列プログラミング環境の構築を目指し、そのための基盤技術として、大きく次の二点についての成果を得た：(1) 局所計算量をプログラム中に自然に併記できるモナドに基づく逐次計算量の形式的記述手法の並列計算量への拡張、(2) 定理証明支援系における不等式などの扱いを簡潔にし、計算量証明を行いやすくする技術。これらにより、並列実行の仕組みに制限はあるものの、機械により速さと正しさが保証される並列プログラムの開発が容易になった。

研究成果の学術的意義や社会的意義

身の回りにあふれるコンピュータが安全かつ効率的に仕事をしているのは、その仕事の指示をしているプログラムが「正しく・速く」動くように作られているからである。しかし、どのような専門家であっても、人である以上、見落としなく「正しさ」と「速さ」を保証したプログラムを作ることはできない。そのため、人ではなく、見落としをしないコンピュータ自身にその「正しさ」と「速さ」を保証させることが理想である。残念なことに、コンピュータにその保証をさせるにも煩雑な手順が必要であり、コストが掛かってしまう。本研究は、その手順を単純化する手法を開発し、そのコストを下げることでより安全かつ効率的な社会の実現に貢献する。

研究成果の概要(英文)：Programs have to be correct and efficient. We have obtained the following two results for parallel programming with certified correctness and complexity: (1) a concise notation in a proof assistant Coq for writing local complexity naively within the program code, to carry out a formal proof of its parallel complexity; and (2) a method for easy and natural manipulation of inequalities in formal proofs. These results have made it easier to build parallel programs with certified correctness and complexity.

研究分野：計算機科学

キーワード：定理証明支援系 並列プログラミング

## 様式 C - 19、F - 19 - 1、Z - 19、CK - 19 (共通)

### 1. 研究開始当初の背景

近年、計算機性能の向上は主にその並列度に依存する状況にある。しかしながら、その性能を有効活用するために必要な並列プログラムの開発は未だ困難である。一般に、高い効率を求めたプログラムはその構造が複雑になり正しさの保証が難しくなるが、複数の計算主体を同時に動作させる並列プログラムはその複雑度が逐次プログラムの比ではなく、並列度を最大限利用して効率を確保しつつその正しさを保つことは困難である。そのため、正しさと効率とが保証された並列プログラムを容易に開発するための手法が必要とされている。

正しい並列プログラムを容易に開発できる枠組みとして、スケルトン並列プログラミング (Skeletal Parallel Programming) が考えられている。この枠組では、プログラマは「スケルトン」と呼ばれる既成の並列計算パターンを組み合わせて並列プログラム (以下、スケルトン並列プログラムと呼ぶ) を作成する。スケルトンが並列性に起因する複雑度を隠蔽するため、正しい並列プログラムの開発が容易となる。しかし、目的の計算に対してスケルトンを最適に組み合わせると効率のよいスケルトン並列プログラムを得ることは、一般に自明ではない。

効率のよいスケルトン並列プログラムを容易に得る手法として、プログラム変換を用いた手法が提案されている。この手法では、プログラマは目的の計算の愚直なプログラム記述を正しく作成する。その後、プログラミング環境が、演算規則 (プログラム間の等式) を用いてプログラムの意味を保ちつつそれを効率的なスケルトン並列プログラムへと変換する。この流れにより、プログラマは効率のよい正しい並列プログラムを容易に得ることが期待できる。

これまで、スケルトン並列プログラミングにプログラム変換を組み合わせた枠組みに関し、多くの研究が行われてきた。しかしながら、いずれの研究成果も最終的に得られる並列プログラムの正しさと効率を保証するものではない。すなわち、並列スケルトン及び演算規則の実装に間違いがあれば、最終的に得られたプログラムは正しくない。そして、複数の演算規則が適用可能な場合に変換後のプログラムの効率の最適性が保証されない。

一方、正しさの保証されたプログラムを得るための手法として、定理証明支援系を用いた正しさの機械的に保証された証明と、その証明からの実プログラムコード抽出が考えられている。定理証明支援系による機械的な保証は強力であり、それを用いて並列スケルトンや演算規則の正しさや効率を機械的に保証できれば、上記の問題は解決される。しかし、その実現にはその証明に関連する理論すべてを定理証明支援系上に定式化する必要があり、残念なことに、スケルトン並列プログラムの正しさと効率とを保証するために必要な理論の定理証明支援系での定式化が欠けている状況にある。プログラムの効率 (計算量) に関する定式化や最適性の証明手法、演算規則 (プログラム変換) に関する性質の証明手法等の不在が大きな問題となっている。

以上をまとめ、正しさと効率とが保証された並列プログラムを容易に開発するための手法実現にあたっては、大きく以下の二点の問題が存在した：

- (1) 並列プログラムの正しさと効率の機械的保証のある証明のための定式化の不在
- (2) 正しさと効率が機械的に保証される、並列プログラミング環境の不在

### 2. 研究の目的

上記の背景のもと、本研究は、正しさと効率とが保証された並列プログラムを容易に開発できる並列プログラミング環境の実現を目的とした。具体的には、2点の実現を目的とした。

- (1) 並列プログラムの正しさと効率の機械的保証のある証明のための定式化

既存の定理証明支援系でのプログラムの効率に関する証明の研究では、逐次プログラムの計算量が特定の式で押さえられることの証明に関する定式化は存在するものの、並列プログラムの計算量に関する定式化とプログラム変換による計算量の変化に関する定式化は行われていない。これは、一般の並列プログラムの計算量の定式化が困難であることによると考えられる。本研究では、スケルトン等による並列プログラムの構造化を軸に据えることで、並列計算量見積に関する定式化を試みる。

- (2) 正しさと効率が機械的に保証される並列プログラミング環境

上記の定式化を元に、記述したプログラムの効率と正しさを機械的に保証できる環境の構築を行う。そのために、定理証明支援系上にスケルトン並列プログラミング環境の構築を試みる。すなわち、本研究項目は、実際のスケルトン集合の定式化や、各種の証明を簡略化するための手法の構築が目的となる。

### 3. 研究の方法

上記ふたつの目的に対し、それぞれ以下のように研究を進める。

- (1) 並列プログラムの正しさと効率の機械的証明のための定式化

本研究項目では、逐次プログラムの簡潔な計算量証明のための既存研究の成果を拡張し、よく知られた並列計算実行モデルである fork-join モデルに対する計算量証明手法の構築を行う。具体的には、既存研究の (拡張) モナドを用いた記述方法に対して並列計算による分配性を考慮する拡張を施し、fork-join モデルの計算量の定式化を行う。また、それを具体例に対して適用し、定式化の有用性を確認する。

- (2) 正しさと効率が機械的に保証される並列プログラミング環境

上記の定式化のもと、並列実行モデルを fork-join モデルとした並列スケルトン集合を定義し、それらの計算量証明とともにそのアプリケーションについての計算量証明を行うための支援ライブラリを作成する。これらの枠組みは、広く用いられている定理証明支援系である Coq 上に実現する。これらの実現に関する成果はオープンソースとして公開する。

#### 4. 研究成果

上のあげたふたつの研究目的それぞれについて、次のような成果を得た。(2)については、当初想定していなかった問題点が浮上したため、主にその解決手法の開発に注力することとなった。

##### (1) 並列プログラムの正しさと効率の機械的証明のための定式化

予定通り、既存の逐次プログラムに関する研究成果を拡張し、単純な並列実行モデルである fork-join モデルに対する簡潔な計算量証明手法を構築した。具体的には、既存研究の用いていた(拡張)モナドが計算量の加算だけを一直線に行っていたのに対し、本研究は、fork-join のポイントにおいて二股に分配できる仕組みを拡張として加えた。分岐点における計算量のマージは、加算が分配する max 演算子を用いてシンプルに実現されている。本手法は Coq 上のライブラリとして構築された。そして、並列プログラミング環境の評価によく用いられる例であるフィボナッチ数の計算プログラムに対して適用され、その計算量証明における有用性が確認された。これにより、全ての並列実行モデルに対してではないものの、並列プログラムの実行性能に関する機械的な保証が可能となった。

##### (2) 正しさと効率が機械的に保証される並列プログラミング環境

本研究項目は、当初にあげた目的のうち、簡潔な証明を支援する手法について主に注力することとなった。これは、上記の定式化のもと複雑なプログラムの証明に取り掛かったものの、定理証明支援系上において不等式等を扱うための支援が貧弱であり、本質的でない努力が多に必要になることが判明したからである。そのため、本項目では、スケルトン並列プログラミングのための並列スケルトンに基づく言語の設計と、一般的な不等式等に基づく定理証明支援系上での証明を支援するライブラリ等の構築についての成果を得ることとなった。前者については、近年重要視されている大規模グラフ処理に関し、グラフ全体を操作する感覚で並列プログラムを容易に記述できる言語が提案・実装された。提案言語により、近年広く用いられている頂点主体な記述を用いるプログラミング言語に比べ、複雑な計算を直観的にかつコンパクトに記述できることが確認された。今後、この言語で書かれたプログラムの計算量保証の仕組みが望まれる。後者については、等式・不等式を用いた証明の自然な形である、等式や不等式を連鎖させた形で定理証明支援系上の証明を記述できるライブラリが提案・実装された。一般には定理証明支援系上での証明はシステムへのコマンド列であり非直感的であるが、この提案ライブラリにより、計算量証明に頻出する不等式の証明を直観的に記述できるようになった。これにより、定理証明支援を用いた機械的な保証の仕組みがより身近なものとなった。

#### 5. 主な発表論文等

##### [雑誌論文](計 2 件)

Kento Emoto, Fumihisa Sadahira: A DSL for graph parallel programming with vertex subsets. The Journal of Supercomputing, 査読有, Vol. 75, 2019, 18 pages (Online first) DOI: 10.1007/s11227-019-02821-w

村田 康佑, 江本 健斗: 定理証明支援系 Coq における不等式変形記法. 情報処理学会論文誌プログラミング (PRO), 査読有, Vol. 11(4), 2018, pp. 1-12

##### [学会発表](計 9 件)

村田 康佑, 江本 健斗: Coq を用いた高度なプログラム演算定理の検証に向けて. 第 21 回プログラミングおよびプログラミング言語ワークショップ PPL 2019, 2019.

村田 康佑, 江本 健斗: Coq における検証されたプログラム演算の拡張. 日本ソフトウェア科学会 第 35 回大会, 2018.

村田 康佑, 江本 健斗: Coq における可読性の高い形式的証明に向けて. 第 20 回プログラミングおよびプログラミング言語ワークショップ PPL 2018, 2018.

白水 駿, 江本 健斗: 並列プログラム計算量の系統的機械証明手法の開発. 日本ソフトウェア科学会第 34 回大会, 2017.

定平 典久, 江本 健斗: 頂点部分集合変数を備えた大規模グラフ計算用領域特化言語. 第 19 回プログラミングおよびプログラミング言語ワークショップ PPL 2017, 2017.

江本 健斗, 松崎 公紀, 胡 振江, 森畑 明昌, 岩崎 英哉: 大規模グラフ並列処理のための関数型領域特化言語 Fregel とその評価. 日本ソフトウェア科学会第 33 回大会, 2016.

Kento Emoto, Kiminori Matsuzaki, Zhenjiang Hu, Akimasa Morihata, Hideya Iwasaki: Think like a vertex, behave like a function! a functional DSL for vertex-centric big graph processing. 21st ACM SIGPLAN International Conference on Functional Programming, ICFP2016, (国際学会), 査読有, 2016, pp. 200-213.

Kento Emoto, Kiminori Matsuzaki, Zhenjiang Hu, Akimasa Morihata, Hideya Iwasaki: A Functional DSL for Large Scale Graph Processing. Thirteenth International Symposium on Functional and Logic Programming (FLOPS 2016) (国際学会), 2016.

江本 健斗, 松崎 公紀, 胡 振江, 森畑 明昌, 岩崎 英哉: 大規模グラフ並列処理のための関数型領域特化言語. 第 18 回プログラミングおよびプログラミング言語ワークショップ PPL 2016, 2016.

〔図書〕(計 0 件)

〔産業財産権〕

出願状況(計 0 件)

名称:  
発明者:  
権利者:  
種類:  
番号:  
出願年:  
国内外の別:

取得状況(計 0 件)

名称:  
発明者:  
権利者:  
種類:  
番号:  
取得年:  
国内外の別:

〔その他〕

ホームページ等

## 6. 研究組織

(1)研究分担者  
該当なし

(2)研究協力者  
該当なし

科研費による研究は、研究者の自覚と責任において実施するものです。そのため、研究の実施や研究成果の公表等については、国の要請等に基づくものではなく、その研究成果に関する見解や責任は、研究者個人に帰属されます。