

平成22年5月26日現在

研究種目：若手研究（スタートアップ）

研究期間：2008～2009

課題番号：20800022

研究課題名（和文） 項書き換えに基づく計算量解析の自動化

研究課題名（英文） Automated Complexity Analysis for Term Rewriting

研究代表者

廣川 直（HIROKAWA NAO）

北陸先端科学技術大学院大学・情報科学研究科・助教

研究者番号：50467122

研究成果の概要（和文）： 探索やソートアルゴリズムを開発したとき、「入力の大きさに対しどれくらいの速度(ステップ数)で動作するか」という自然な疑問が生じる。現在に至るまで時間的計算量は、プログラムごとに手作業で解析するものと認識されていた。本研究では、関数型プログラムの計算モデルである項書き換え系に対して、解析を自動化する強力な理論を構築した。さらにそれに基づく計算量自動解析ツールを実装した。既存手法との比較実験を行った結果、解析精度の劇的な向上が確認された。

研究成果の概要（英文）： Time complexity is one of the most fundamental properties of programs. So far time complexity of programs has been analyzed manually. In this project we developed a powerful theoretical framework for automating runtime complexity analysis of term rewrite systems, which are underlying computational models for declarative programs. Moreover, we have implemented a complexity analyzer based on our framework. Experiments showed that our method outperforms existing ones in its analytical precision.

交付決定額

(金額単位：円)

| | 直接経費 | 間接経費 | 合計 |
|--------|-----------|---------|-----------|
| 2008年度 | 790,000 | 237,000 | 1,027,000 |
| 2009年度 | 1,160,000 | 348,000 | 1,508,000 |
| 年度 | | | |
| 年度 | | | |
| 年度 | | | |
| 総計 | 1,950,000 | 585,000 | 2,535,000 |

研究分野：総合領域

科研費の分科・細目：情報学・情報学基礎

キーワード：項書き換え、計算モデル、計算量

1. 研究開始当初の背景

プログラムの時間効率がどれくらいなのかを解析することは、プログラム解析における中心的なテーマである。仮に探索アルゴリズムやソートアルゴリズムを開発したとする。そのとき、私たちは時間的計算量、すなわち「入力の大きさに対しどれくらい速度(ステップ数)で動作するか」という自然な疑問を持つ。これまで計算量は、プログラムごとに手作業で解析するものと認識されていた。

本研究のねらいは、この解析を自動化することである。本研究では特に項書き換え系と呼ばれる宣言型プログラミング言語のためのシンプルな計算モデルを対象にする。

たとえば、以下は自然数の減算と除算を計算する項書き換え系である。

$$\begin{aligned}x - 0 &\rightarrow x \\s(x) - s(y) &\rightarrow x - y \\0 / s(y) &\rightarrow 0 \\s(x) / s(y) &\rightarrow s((x - y) / s(y))\end{aligned}$$

ここで定義される関数は、入力の項のサイズに対して、線形時間 $O(n)$ で計算を終える。

研究開始時、このような時間的計算量の上限を精度良く調べる理論は確立しておらず、また強力な自動解析手法・ツールが存在する停止性解析に比べ、計算量の自動解析に必要な理論・技法は多くが欠けている状態であった。

2. 研究の目的

これまでに項書き換え系の計算量解析のために開発された手法は、Match-Bound (Geser et al, 2007) のような一部の例外を除けば、すべて順序に基づく解析手法であった。POP* (Avanzini & Moser, 2008) や SMC (Bonfante et al, 2001) と呼ばれる順序は、非常に特殊な順序で、項書き換え系の全ての規則 $\ell \rightarrow r$ が $\ell > r$ を満たすかを調べるだけで、項書き換え系の計算量を結論でき、さらにこのチェックは自動化も可能である。

しかしながら、これらの順序は適応能力が極めて乏しく、ユークリッドの互除法やソートアルゴリズムの解析は不可能であった。また多項式時間以下の計算量を証明できるPOP*, SMC は特に弱いため、低い計算量の自動解析は事実上不可能であった。

事前研究において、研究協力者 Georg Moser (University of Innsbruck) とともに、重差原理 (Hirokawa & Moser, 2008) という計算量の原理を発見した。この発見により、依存対という元々は停止性証明のために開発された手法を、計算量解析へ導入することに成

功した。この依存対は既存の順序に基づく計算量解析手法を劇的に強化する手法で、これにより、例えば先にあげた除算や典型的なリスト操作関数の線形時間性を自動解析できるようになった。

本プロジェクトでの目標は以下の通り。

- (1) この重差原理と依存対に基づく手法を発展させることが第一の目標である。理想としては、ソートアルゴリズムの計算量を自動的に示せるくらいまで発展させたいと考えていた。
- (2) さらに関数型プログラムの計算量を解析するため、プログラムを項書き換え系へ変換する手法を開発する。
- (3) 項書き換えにおいて「計算量の自動解析」は新しい研究テーマであり、これを項書き換えの一分野としての確立させる。特に、計算量解析ツールの解析精度と速度を競う国際大会の開催を目指す。

3. 研究の方法

前節の目標達成のため、以下の事項に取り組んだ。

- (1) 計算量理論の研究。
 - ① 依存グラフ。重差原理の研究を行い、停止性証明のために導入された強力な解析手法である依存グラフ (Giesl et al, 2002) を計算量解析へ適応させる。
 - ② 重差原理の拡張。重差原理は強線形解釈と呼ばれる、極めて制約の強い多項式解釈がベースとなっている。この制約の緩和を試みる。
 - ③ Uncurrying 変換。関数型プログラムの項構造は Applicative Term と呼ばれる直接の解析が極めて難しい形をしている。これを通常の項の形に戻す Uncurrying 変換を開発する。
- (2) 自動化のための理論
 - ① 高速なグラフ解析アルゴリズムの開発。停止性解析では、依存グラフの循環解析 recursive SCC algorithm (Hirokawa & Middeldorp, 2005) があり、ナイーブなアプローチでは指数時間かかる解析を線形時間で行う高効率なアルゴリズムがある。この計算量解析のためのアルゴリズムを開発する。

- ② 高速な順序探索手法。前述の通り、計算量解析は基本的に計算量を保障する順序を見つけることにより実現されている。この順序を高速に見つける手法を見つける。
 - (3) 計算量自動解析ツール実装。研究代表は強力な停止性解析ツールとして知られている TTT (Hirokawa & Middeldorp, 2005) の開発者であり、これをベースに開発を行う。実装は、上記(1), (2)の進捗に応じて進める。
 - (4) 評価実験。TPDB (Termination Problem Database) と呼ばれる大規模な問題集を用いて、(1), (2)で得られた手法を実装した(3)のツールで解く。その解けた問題数、その種類、また解くためにかかった時間が評価の基準になる。
 - (5) 計算量解析ツールの国際大会発足のため研究者らとルール策定の協議を行う。
4. 研究成果

(1) 計算量理論の研究成果。

- ① 依存グラフの導入に成功した。停止性では潜在的なループを検出することが目的であったため、依存グラフにおける循環成分の停止性解析を行えばよかった。一方、計算量解析では、潜在的な最長の書き換え列を同定することがグラフの利用の目的になるため、強連結成分の列の解析を行えばよいという着想に至った。これにより解析精度が劇的に向上する。
- ② 重差原理の拡張に成功した。Moserら (2008) が導入した上三角行列の係数を持つ行列解釈 (Endrullis, 2006) が、重差原理の新たなベースになることが判明した。プロトタイプツールを用いた評価実験では上記①のインパクトを凌ぐ結果が得られた。
- ③ Uncurrying 変換の開発に成功した。変換後のプログラムの簡約列を保存する変換であり、またこの変換は依存対の手法と組み合わせることもできる。Giesl ら (2005) の Uncurrying 変換と違い、map のような高階関数 (head variable を含む書き換え規則) も扱える。

(2) 自動化のための理論

- ① 循環解析のためのアルゴリズムを開発した。停止性解析では $O(n)$ の解析アルゴリズムが存在したが、計算量解析でもグラフ上の全パスに対応する解析を $O(n^2)$ の時間で実現するアルゴリズムを開発することに成功した。
- ② SAT/SMT ベースの探索。計算量を決定する適切な順序を発見するには、広大な探索空間を廻らねばならない。Fuhs ら (2006) の SAT ソルバを探索に用いるアプローチを採用することにより、これを解決できた。
- (3) 計算量解析ツール。以上の成果を取り込んだ計算量自動解析ツール Hydra を実装した。公開されている停止性問題集 TPDB の 1739 個の項書き換え系に対する実験結果は下表の通り (30 秒タイムアウト)。速度は「本研究手法」がやや遅い。

戦略制限なしの場合 (full rewriting)

| 計算量 | 既存 | 事前研究 | 本研究 |
|----------|-------|-------|--------------|
| $O(n)$ | 139 個 | 147 個 | 162 個 |
| $O(n^2)$ | 182 個 | 186 個 | 202 個 |

最内戦略の場合

| 計算量 | 既存 | 事前研究 | 本研究 |
|----------|-------|-------|--------------|
| $O(n)$ | 144 個 | 166 個 | 181 個 |
| $O(n^2)$ | 183 個 | 193 個 | 208 個 |

以上は当初の計画に沿った成果である。これらに加え、研究当初、予期しなかった停止性解析・定理証明システムにとって重要な成果が得られた：

- (4) Knuth-Bendix 順序の有限性の証明。重差原理の研究から、1970 年に導入された Knuth-Bendix 順序が要求する探索空間が実は有限であることが判明した。この順序は定理証明システムに広く採用されている停止性証明の手法であり、この順序を用いるには関数記号から自然数への適切な写像を見つける必要がある。本研究により入力項書き換え系のサイズを n とした場合、 $\{0, \dots, 2^{(2^n)}\}$ への写像で全ての場合が網羅されることが証明された。

本研究の位置づけとインパクトについて。

- (1) 上述の成果は、(1)-(2)を除き、すべてAランク(格付け機関 CORE による)の国際会議・国際論文誌に掲載された。
- (2) 計算量解析ツールの国際大会開催が国際ワークショップ WST で決まり、2008年12月に第一回大会が開かれ、それ以来、毎年開かれている。

<http://www.termination-portal.org/wiki/Complexity>

- (3) 他の計算量解析ツール TCT, CaT もまた本プロジェクトによって得られた(ほぼ全ての)手法を採用している。

今後の展望について。

- (1) 項書き換えのコミュニティにおいて、計算量の自動解析は分野として注目を集めるようになった。解析精度、速度ともに現状不十分であり、さらなる手法の開発が今後の焦点になるであろう。特に計算量解析のためのモジュラ性が鍵である。
- (2) ML や Haskell のような言語で書かれたプログラムの計算量解析を実現に向けた課題はまだ山積している。特に Call by Value や Call by Need のような簡約戦略や、遅延評価の実装で用いられる項グラフを考慮した解析方法の研究が必要である。
- (3) 計算量の自動解析の実現によって、コンパイラなどのプログラム最適化のための新たな枠組みが可能になると期待している。

5. 主な発表論文等

(研究代表者、研究分担者及び連携研究者には下線)

[雑誌論文] (計1件)

- ① Harald Zankl, Nao Hirokawa, and Aart Middeldorp.
KBO Orientability.
Journal of Automated Reasoning 43(2), pp. 173-201, 2009. 査読有

[学会発表] (計4件)

- ① Nao Hirokawa and Georg Moser.
The Subterm Criterion in Complexity Analysis.
In Proceedings of the 10th International Workshop on Termination,

Leipzig, Germany, 2009.
2009年6月4日発表。

- ② Nao Hirokawa, Aart Middeldorp, and Harald Zankl.
Uncurrying for Termination.
In Proceedings of the 15th International Conferences on Logic for Programming, Artificial Intelligence and Reasoning, Sydney, Australia, Lecture Notes in Computer Science 5330, pp. 667-681, 2008. 査読有.
2008年11月27日発表
- ③ Nao Hirokawa and Georg Moser.
Complexity, Graphs, and the Dependency Pair Method.
In Proceedings of the 15th International Conferences on Logic for Programming, Artificial Intelligence and Reasoning, Sydney, Australia, Lecture Notes in Computer Science 5330, pp. 652-666, 2008. 査読有
2008年11月27日発表

[図書] (計1件)

- ① 廣川 直.
電子情報通信学会 知識ベース 7群1編「ソフトウェア基礎」2章「定理証明と検証法」3節「SAT」社団法人 電子情報通信学会, 2010年

[その他]

ホームページ等

研究代表者のホームページ:

<http://www.jaist.ac.jp/~hirokawa/>

6. 研究組織

(1) 研究代表者

廣川 直 (HIROKAWA NAO)

北陸先端科学技術大学院大学・情報科学研究科・助教

研究者番号: 50467122

(2) 研究分担者

()

研究者番号:

(3) 連携研究者

()

研究者番号:

(4) 研究協力者

MOSER GEORG (モージー ゲーオグ)

University of Innsbruck・Institute for Computer Science・Research Assistant