様式Ｃ－１９

科学研究費助成事業（科学研究費補助金）研究成果報告書

平成 25 年　6 月　5 日現在

機関番号：14301
研究種目：若手研究(A)
研究期間：2009～2012
課題番号：21680002
研究課題名（和文）静的・動的型付けの融合による安全かつ柔軟なプログラミング言語の理論と設計
研究課題名（英文）Theory and Design of a Safe and Flexible Programming Language Based On the Integration of Static and Dynamic Typing
研究代表者
五十嵐 淳　（IGARASHI ATSUSHI）
京都大学・大学院情報学研究科・教授
研究者番号：40323456

研究成果の概要（和文）：プログラミング言語には、実行前にプログラム中の誤り検査を行う静的型付言語と実行中に検査を行う動的型付言語があり一長一短があることが知られている。本研究では両方の機能を兼ね備えたプログラミング言語を設計し、その理論と実装手法を研究した。理論的成果として、部分的に型付けされるプログラムの実行前検査のためのアルゴリズムと実行時検査手法を与え、アルゴリズムの正当性や、その検査で検出される誤りは全て実行前検査を行わなかった部分に起因することを証明した。また実行時検査の実装手法を考案した。

研究成果の概要（英文）：There are two kinds of programming languages: statically typed languages, in which a program is subject to error-checking before running it; and dynamically typed languages, in which errors in a program are found only at its run time. They have both advantages and disadvantages. In this research we designed a new language which has both functionalities and studied its theory and implementation. As theoretical results, we obtained a type-checking algorithm for partially-typed programs, a method to perform run-time checks, and their correctness proofs, which show errors found at run-time can be attributed to a portion of programs that are not checked before running. We also devised an implementation scheme.

交付決定額

（金額単位：円）

	直接経費	間接経費	合　計
2009 年度	5,200,000	1,560,000	6,760,000
2010 年度	4,200,000	1,260,000	5,460,000
2011 年度	4,200,000	1,260,000	5,460,000
2012 年度	3,800,000	1,140,000	4,940,000
総　計	17,400,000	5,220,000	22,620,000

研究分野：総合領域
科研費の分科・細目：情報学・ソフトウェア
キーワード：オブジェクト指向言語、漸進的型付け、プログラミング言語、Java 言語、ジェネリクス、型安全性、コントラクト

１．研究開始当初の背景

ソフトウェアの安全性向上のための技術として，静的解析・モデル検査など、コンパイル時情報を使って実行前に安全性を確保する技術と、モニタリングなど、実行時情報を使う技術とが盛んに研究されている。また、これに対応して、プログラミング言語も、プログラ

ム中のエラー検出手段によって、Java や ML などの、型を使ってコンパイル時にエラーを検出する静的型付け言語と、Lisp やスクリプト言語などの、実行時に検出する動的型付け言語とに大別できる。
プログラム実行前に安全性を保証する理論は重要である一方で、理論の高度化によるプログラミング言語の複雑化、という問題がある。これに対し、Perl、 Python、 Ruby など、いわゆるスクリプト言語と呼ばれる動的型付け言語は、全てのエラー検出を実行時に行うためプログラムの誤りの発見が遅れる・大規模プログラムを書くための機能が貧弱である、などの欠点がありながらも、豊富なライブラリなどのおかげで素早いプロトタイプ作成に威力を発揮することで幅広く普及している。

２．研究の目的
上記経緯・背景を踏まえ、本研究では、近年のソフトウェア開発において実際に中心的に使われているクラスに基づくオブジェクト指向プログラミング言語を対象とし、静的型付けと動的型付けの技術を融合し、静的型付けの安全性と動的型付けの柔軟性・敏捷性を兼ね備えたプログラミング言語の理論を構築し、その理論に基づく言語の設計・実装を行うことを目的とした。具体的には以下のような技術的課題に取り組んだ。

(1) 静的・動的型付け融合の理論: 素朴に静的型付け言語に動的型付けを持ち込むだけでは「ある種のエラーは実行時に発生しない」という静的型付け言語の利点が失われてしまう。これの解決策として「静的に検査されたコードの責任で 実行時にエラーが発生することはない」という (部分的) 安全性を考え、これを保証するような理論的枠組みを確立する。
(2) 動的検査の理論: 既存の静的型付け言語に見られる典型的な動的検査はキャストという機能である。これは、あるオブジェクトが実行時に別の型の部分型に属しているかを実行時に検査する機能である。しかし、Java のような表現力の高い型システムにおいては、その動的検査問題が決定可能であるかも実は明らかにされていない。(そのため実際の Java の実装では、検査の範囲を制限している。) そこで、動的検査問題のアルゴリズム的側面、すなわち決定可能性・計算量などを研究し、効率的な実行時検査の実装に応用する。
(3) 静的型付け技術の向上: 静的・動的型付けを融合した言語を使った開発においても, 最終的な目標は動的検査に頼らないプログラムを作成することである。そのために、不足している静的型システムの表現力、特に再利用性の向上に取り組む。
(4) プログラミング言語の設計と実装: それらの結果を総合して、Java 言語を拡張した新しいプログラミング言語の設計を行う。

３．研究の方法
研究目的の(1)〜(4)に対応して、以下の方法で研究を行った。
(1) 静的・動的型付け融合の理論: 代表者がこれまで研究してきた Java 言語のための計算モデル Featherweight Java [3] の上に構築し、実際に部分的安全性が成立することを数学的に証明する。この性質は、実行時エラーが発生した時にも、エラー原因の範囲が限定できるので、プログラムの誤りが発見しやすいなど、実用上重要である。
(2) 動的検査の理論: 動的検査の一例であるキャストに伴なう部分型検査が、そもそも決定可能であるかという問題の解決をはかる。
(3) 静的型付け技術の向上: 自己参照・相互参照を持つ総称(ジェネリック)クラス定義や、動的にクラス定義が合成される動的レイヤー合成といった機能を持つ言語のための型システムを構築する。
(4) プログラミング言語の設計と実装: 既存のコンパイラを利用した処理系構築を行い、特に動的検査の効率的な実装方法を開発する。

４．研究成果
研究成果は ACM OOPSLA を中心とするトップ国際会議で発表されるとともに、代表者は、本研究成果を含めてこれまでの業績に関してオブジェクト指向分野での最高権威である Dahl-Nygaard 若手研究者賞を受賞([発表論文 4; 研究発表 7])するなど国際的にインパクトのある成果をあげることに成功した。
(1) ジェネリクスを持つオブジェクト指向言語のモデル Featherweight GJ に対し、部分的型付けを行う型理論を与え部分的安全性が成立することの証明を与えることができた[発表論文 7,5, 研究発表 6,10]。また、顕在的契約計算と呼ばれる、関数型言語における静的/動的型付けの融合の理論においても、既存の理論を多相型による拡張に成功した[発表論文 3,6; 研究発表 2,5,8]。これらの成果は ACM OOPSLA, ESOP などのトップレベル国際会議で発表された。
(2)部分型検査が決定可能であることの証明を与えることができた。この成果が得られたのが最終年度ということもあり、未発表となっている。
(3) 自己参照・相互参照を持つ総称(ジェネリック)クラス定義や、動的にクラス定義が合成

される動的レイヤー合成といった機能を持つ言語のための型システムとその正当性の証明を与えることに成功した。これらの成果はACM OOPSLA などトップ国際会議で発表されるとともに、代表者が国際ワークショップで招待講演を行うなどの顕著な成果をあげている[発表論文1, 2, 8; 成果発表1, 3, 4, 9]。

(4) 実装については、OpenJDK コンパイラを拡張することで、動的検査の基本的な実装手法を研究したが、コンパイラを完成するために必要な部分型検査の決定可能性の解決が最終年度になったために実装の完成にはこぎつけられなかった。

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 9 件）

[1] Kohei Suenaga, Ryota Fukuda, Atsushi Igarashi, Type-based Safe Resource Deallocation for Shared-Memory Concurrency, Proc. of ACM OOPSLA, 査読有, 2012, 1-20, 10.1145/2384616.2384618

[2] Atsushi Igarashi, Robert Hirschfeld, Hidehiko Masuhara, A Type System for Dynamic Layer Composition, Proc. of the International Workshop on Foundations of Object-Oriented Languages, 査読有, 2012, 13—24, http://www.cs.uwm.edu/~boyland/fool2012/papers/fool2012_submission_9.pdf

[3] 関山 太朗、 五十嵐 淳、顕在的契約計算におけるアップキャスト除去、第 14 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(PPL2012)論文集、査読有、2012

[4] Atsushi Igarashi, Featherweight Approach to FOOL, Proc. Of European Conference on Object-Oriented Programming, 査読無, LNCS 6813, 2011, 433-433, 10.1007/978-3-642-22655-7_20

[5] Lintaro Ina, Atsushi Igarashi, Gradual Typing for Generics, Proc. Of ACM OOPSLA, 査読有, 2011, 609—624, 10.1145/2048066.2048114

[6] Joao Filipe Belo, Michael Greenberg, Atsushi Igarashi, Benjamin C. Pierce, Polymorphic Contracts, Proc. Of European Symposium on Programming, 査読有, Springer LNCS 6602, 2011, 18-37, 10.1007/978-3-642-19718-5_2

[7] Lintaro Ina, Atsushi Igarashi, Towards Gradual Typing for Generics, Proc. Of Workshop on Script to Program Evolution, 査読有, 2009, 17—29, 10.1145/1570506.1570509

[8] Chieri Saito, Atsushi Igarashi, Self Type Constructors, Proc. Of ACM OOPSLA, 査読有, 2009, 263—282, 10.1145/1640089.1640109

[9] 伊奈 林太郎、五十嵐 淳、Featherweight Java のための漸進的型付け、コンピュータソフトウェア、査読有、26、２００９、18—40

〔学会発表〕（計 10 件）

[1] Atsushi Igarashi, Type Systems for Context-Oriented Programming, International Workshop on Foundations of Aspect-Oriented Languages（招待講演）, 2013 年 03 月, 日本・福岡

[2] Taro Sekiyama, Atsushi Igarashi, Logical Relations for a Manifest Contract Calculus, Fixed, ACM SIGPLAN Workshop on Higher-Order Programming with Effects, 2012 年 09 月, デンマーク・コペンハーゲン

[3] Ryota Fukuda, Kohei Suenaga, Atsushi Igarashi, Type-based Safe Resource Deallocation for Shared-Memory Concurrency, ACM OOPSLA, 2012 年 10 月, アメリカ・アリゾナ州・ツーソン

[4] Atsushi Igarashi, Robert Hirschfeld, Hidehiko Masuhara, A Type System for Dynamic Layer Composition, International Workshop on Foundations of Object-Oriented Languages, 2012 年 10 月, アメリカ・アリゾナ州・ツーソン

[5] 関山 太朗、 五十嵐 淳、顕在的契約計算におけるアップキャスト除去、第 14 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、2012 年 3 月、和歌山県西牟婁郡白浜町

[6] Lintaro Ina, Atsushi Igarashi, Gradual Typing for Generics, ACM OOPSLA, アメリカ・オレゴン州ポートランド、2011 年 10 月

[7] Atsushi Igarashi, Featherweight Approach to FOOL, European Conference on Object-Oriented Programming, 招待講演, イギリス・ランカスター, 2011 年 7 月

[8] Joao Filipe Belo, Michael Greenberg, Atsushi Igarashi, Benjamin C. Pierce, Polymorphic Contracts, European Symposium on Programming, ドイツ・ザールブルッケン, 2011 年 3 月

[9] Chieri Saito, Atsushi Igarashi, Self Type Constructors, ACM Conference on Object-Oriented Programming, Systems, Languages, and Applications, アメリカ・フロリダ州オーランド, 2009 年 10 月

[10] Lintaro Ina, Atsushi Igarashi, Towards Gradual Typing for Generics, International Workshop on Script to

Program Evolution (STOP),イタリア・ジェノヴァ, 2009 年 7 月

〔図書〕（計 1 件）
[1] 五十嵐 淳、プログラミング言語の基礎概念、サイエンス社、2011、192 頁

〔産業財産権〕
○出願状況（計 0 件）
○取得状況（計 0 件）

〔その他〕

ホームページ等
http://www.fos.kuis.kyoto-u.ac.jp/~igarashi/

６．研究組織
(1)研究代表者
五十嵐 淳（IGARASHI ATSUSHI）
京都大学・大学院情報学研究科・教授
研究者番号： 40323456