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

平成　２９　年　　６　月　１３　日現在

機関番号：　１３７０１
研究種目：　基盤研究(C)（一般）
研究期間：　2012～2016
課題番号：　２４５０００１２

研究課題名（和文）静的再帰構造解析に基づく関数プログラムの停止性自動証明

研究課題名（英文）On Proving Termination of Functional Programs by Static Recursion Analysis

研究代表者
草刈　圭一朗（Kusakari, Keiichirou）

岐阜大学・工学部・教授

研究者番号：９０３２３１１２

交付決定額（研究期間全体）：（直接経費）　　3,900,000円

研究成果の概要（和文）：我々はSML系の関数型言語そのものと呼んでもよい構文を持つ書換えモデルを定式化した．理論的には，高階関数，多相型，代数的データ型，パターンを用いた関数抽象の４つの拡張を行っている．このモデル上に，パターンを用いた関数抽象への対応が不十分ではあるものの，静的な再帰構造解析に基づく停止性証明法である静的依存対法と，その補助理論となる簡約化順序，引数切り落とし法，実効規則の3つの理論構築を行い，得られた理論成果とSMTソルバとの連携に基づいた関数型プログラム停止性証明ツールの作成を行った．これらの拡張により実用性が劇的に向上し，実在のプログラム検証へ多大な貢献が期待できることになる．

研究成果の概要（英文）：We formalize a rewriting model for functional programs, which can represent the notions of higher-order function, polymorphic type, algebraic data type, and functional abstraction with pattern. We extend the static dependency pair method, which proves the termination by analyzing a static recursive structure, onto this rewriting model. In order that this method gives full play to its ability, we also extend reduction orders, the argument filtering method, and the notion of usable rules, although present results do not handle functional abstraction with pattern. Since the syntax of our rewriting model is very close to SML-like functional programs, our result can expect the effective applicability to verification for existing functional programs.

研究分野：　情報基礎

キーワード：　関数型プログラム　停止性　再帰定義

１．研究開始当初の背景

関数プログラミング言語を用いてプログラムする際には停止性を保証するために，多くのプログラマーは関数の再帰定義を行う際に引数の重みが減少するようにプログラムする．逆に言うと，再帰構造を適切に設計することにより停止性が保証されるという”常識”が存在する．しかし，この”常識″が成立しない例が存在することが知られている．例えば，次の２つの規則からなる組合せ子論理と呼ばれる系がそのような例である．

$S\ f\ g\ x \rightarrow f\ x\ (g\ x)$　$K\ x\ y \rightarrow x$

この体系は再帰構造を持たない(関数定義の右辺にS もK も出現しない）にも関わらず，任意の計算可能(プログラミング可能）な関数を記述でき，それゆえに停止性を持たない．この例は，上述の”常識″が一般には成立しないことを意味する．これは非常に深刻な問題である．何故ならば，”常識″として信じられていたことが成立しないと言うことは，プログラマーが停止性を持つと確信していたプログラムが停止性を持たない可能性を含んでしまうからである．

2007 年に我々は静的な再帰構造解析に基づく停止性証明法である静的依存対法を提案し，関数型言語のモデルである単純型付きの書換え系上でこの”常識″の成立を保証する具体的な条件を世界で初めて与えた．この成果の鍵は，型付きλ計算の停止性証明のために導入された強計算性の概念に基づく再帰構造という非常に抽象的な概念の定式化に成功したことにある．さらに，2008-2011年度に科研費の補助を受け，静的依存対法に基づく停止性証明法を確立していった．これらの研究は非常に成功を収めた．実際，我々の静的依存対法は高階関数を直接取り扱える現状では最も強力な停止性証明法である．また，これらの成果により，上述の”常識”が成立するためには，高階変数が左辺の引数またはそれに準じた位置に出現すれば良いことが判明した．これは，ほぼ全ての実用的な関数型プログラムが満たす性質であるので，上述の”常識”を破壊する不具合は稀にしか発生し得ない．この事実は，上述の”常識” が常識として残り続けていたことを説明するという意味において，我々の成果が非常に興味深い成果である事を主張する．一方，我々の成果を実用的な関数プログラミング言語
(SML#,SML/NJ,OCaml,Haskell 等）に適用するには未だ多くのギャップがある．特に問題となってくるのが多相型に対応していないと言うことである．このため，我々の成果を現実のソフトウェア検証に応用するために，なお一層の拡張が必要となっている．逆に言うと，これらのギャップを埋めることができたならば，実用的な関数プログラミング言語で作成されたソフトウェアの検証への劇的な貢献が期待できる．よって，社会に対する非常に大きな貢献となることが期待できる．

２．研究の目的

本研究の目的は，静的再帰構造解析に基づく関数プログラムの停止性証明法の理論整備と，実用的な関数プログラミング言語で作成された実在するソフトウェアの検証へ直接適用可能な停止性自動証明システムの構築である．この目標が達成されたならば，これまで理論研究として蓄積されてきた成果を現実世界のソフトウェア検証に適用できる．この非常に野心的な目標に我々は後一歩と言うところまで届いている．我々の目的はこの最後の一歩を埋めるという目的であり，それゆえに社会に対する非常に大きな貢献となることが期待できる目的である．

３．研究の方法

数学的に厳密な議論を行うための計算モデルの形式化が必要となる．また，本研究の実用性を高めるために，実在の関数型言語との親和性が高い計算モデルとする必要がある．我々は項書換え系と呼ばれる計算モデルに，高階関数，多相型，代数的データ型，パターンを用いた関数抽象の４点を表現する能力が付加できるように拡張する．

提案した計算モデル上に，静的な再帰構造解

析に基づく停止性証明法である静的依存対法を多相型，代数的データ型，パターンを用いた関数抽象の３点に対応できるように拡張する．
静的依存対法が有効に機能するために必要な簡約化順序，引数切り落とし法，実効規則の３つの補助理論を設計する．簡約化順序としてはすでに提案済みの高階辞書式経路順序を多相型，代数的データ型，パターンを用いた関数抽象の３点に対応できるように拡張する．引数切り落とし法と実効規則に関しても，多相型，代数的データ型，パターンを用いた関数抽象の３点に対応できるように拡張する．
提案した理論と，近年飛躍的に能力を向上させている SMT ソルバ(Satisfiability Modulo Theories Solver)との連携に基づく停止性自動証明システムの作成が最終目標となる．

４．研究成果
実用的な関数型言語(SML/NJ, SML#)そのものと呼んでもよい構文を持つ計算モデルの形式化を行った．パターンを用いた関数抽象以外に対応した計算モデルは公開済みである．その後，パターンを用いた関数抽象に対応した計算モデルの形式化にも成功し現在論文執筆中である．
提案した計算モデル上に，静的な再帰構造解析に基づく停止性証明法である静的依存対法をパターンを用いた関数抽象以外に対応できるように拡張した．また，パターンを用いた関数抽象に対応できるようにも拡張し現在論文執筆中である．
静的依存対法が有効に機能するために必要な簡約化順序，引数切り落とし法，実効規則の３つの補助理論に関しては発表できた成果としては不十分ではある．一方，静的依存対法の拡張の際に提案した技術の応用で代表的な簡約化順序である高階辞書式経路順序の設計が可能であるという感触は得ている．また，引数切り落とし法と実効規則に関しては，土台となる静的依存対法が完成していないため論文執筆ができないものの，すでにほぼ研究は完成している．
提案した理論と近年飛躍的に能力を向上させている SMT ソルバとの連携に基づく停止性自動証明システムの作成は，基礎となる理論の構築が不完全であるため未完成である．一方，単純型に限定されるものの高階関数に対応し SMT ソルバと連携できるツールを，理論の完成と共に容易に拡張できるように拡張性を持たせた形で作成済みである．

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２件）
① YAMADA Akihisa, KUSAKARI Keiichirou, SAKABE Toshiki, A Unified Ordering for Termination Proving, Science of Computer Programming, In press, manuscript is available as CoRR abs/1404.6245, 2014
② KUSAKARI Keiichirou, Static Dependecy Pair Method in Rewriting Systems for Functional Programs with Product, Algebraic Data, and ML-Polymorphic Types, IEICE Transactions on Information and Systems, Vol. E96-D, 2013, 472-480

〔学会発表〕（計４件）
① YAMADA Akihisa, Christian Sternagel, René Thiemann, KUSAKARI Keiichirou, AC Dependency Pairs Revisited, 25th EACSL Annual Conference on Computer Science Logic (CSL 2016), 2016 年 8 月 29 日, Marseille (France)
② YAMADA Akihisa, KUSAKARI Keiichirou, SAKABE Toshiki, Nagoya Termination Tool, Joint 25th International Conference on Rewriting Techniques and Applications and 12th International Conference on Typed Lambda Calculi and Applications (RTA-TLCA 2014), 2014 年 7 月 15 日, Vienna (Austria)
③ YAMADA Akihisa, KUSAKARI Keiichirou, SAKABE Toshiki, Unifying the Knuth-Bendix, Recursive Path and Polynomial Orders, 15th International Symposium on Principles and Practice of Declarative Programming (PPDP2013), 2013 年 9 月 17 日, Madrid (Spain)
④ YAMADA Akihisa, KUSAKARI Keiichirou, SAKABE Toshiki, Partial Status for KBO, 13th International Workshop on Termination (WST2013), 2013 年 8 月 29 日, Bertinoro (Italy)

〔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：
出願年月日：
国内外の別：

○取得状況（計０件）

名称：
発明者：
権利者：
種類：

番号：
取得年月日：
国内外の別：

〔その他〕
ホームページ等

６．研究組織
(1)研究代表者
草刈 圭一朗（KUSAKARI, Keiichirou）
岐阜大学・工学部・教授
研究者番号：90323112

(2)研究分担者
坂部　俊樹（SAKABE, Toshiki）
名古屋大学・情報科学研究科・教授
研究者番号： 60111829

(3)連携研究者
酒井 正彦（SAKAI, Masahiko）
名古屋大学・情報科学研究科・教授
研究者番号：50215597

西田 直樹（NISHIDA, Naoki）
名古屋大学・情報科学研究科・准教授
研究者番号：00397449

(4)研究協力者
（　　）