様　式　Ｃ－１９、Ｆ－１９、Ｚ－１９　（共通）

科学研究費助成事業　　研究成果報告書

平成　２８　年　　６　月　　２　日現在

機関番号：　１２１０２
研究種目：　若手研究(B)
研究期間：　2013～2015
課題番号：　２５７３００３５

研究課題名（和文）ゲーム意味論に基づくリファインメント型の拡張とその応用

研究課題名（英文）Extensions and Applications of Refinement Types based on Game Semantics

研究代表者
海野　広志（Unno, Hiroshi）

筑波大学・システム情報系・助教

研究者番号：８０５６９５７５

交付決定額（研究期間全体）：（直接経費）　　3,200,000円

研究成果の概要（和文）：本研究では、ソフトウェアの信頼性向上のために、高レベルプログラムの形式検証手法の一つであるリファインメント型システムおよびその型検査・推論法の、表示的意味論に基づく拡張を目指した。その主な成果として、代数データ構造を扱う高階関数型プログラムの(a)停止性、(b)非停止性、(c)関係的性質の全自動・高精度検証が可能な検証ツールRCamlの開発が挙げられる。

研究成果の概要（英文）：The aim of this research project was to extend refinement type systems and their type checking and inference methods based on a denotational semantics, with applications to formal verification of high-level programs. The main result is the development of a fully-automated tool RCaml for path-sensitive verification of (a) termination, (b) non-termination, and (c) relational properties of high-order functional programs that manipulate algebraic data structures.

研究分野：　情報科学

キーワード：　プログラム検証　型システム　定理自動証明　制約解消

１．研究開始当初の背景

金融・電力・交通といった重要な社会インフラの多くがコンピュータシステムによって制御されている現代社会において、ソフトウェアの信頼性の向上が重要かつ緊急の課題となっている。このような中、プログラムが仕様通りに動作することをプログラム理論に基づき数学的に証明するための技術である形式検証が注目されており、その中でも、ユーザの負担が少ない全自動・高精度検証のための理論・ツールの研究・開発が盛んに行われている。しかし、そこで主な検証対象となっているのは、C 言語のような手続き型言語で記述された、比較的抽象度の低いデータ構造・制御構造を用いたプログラムであり、Java のようなオブジェクト指向言語や OCaml のような関数型言語で記述された抽象度の高いデータ構造・制御構造を用いた高レベルプログラムの全自動・高精度検証については、多くの理論的・実践的課題が残されている。

このような中、研究代表者は、高レベルプログラムのための形式検証手法の一つであるリファインメント型システムの研究を継続的に行ってきており、2009 年にはその型推論法を世界に先駆けて提案することによって、高レベルプログラムの部分正当性の全自動・高精度検証を可能とした。ここでリファインメント型システムとは、リファインメント型とよばれる述語論理式を備えた型として記述された仕様を、与えられた高レベルプログラムが実際に満たすことを形式的に証明もしくは反証するための論理体系であり、手続き型言語にとっての Hoare 論理に対応するものである。例えば、整数上の加算演算の「整数 x と y を受け取ってそれらの和として表される整数 z を返す」といった仕様は、リファインメント型 (x : int) -> (y : int) -> {z : int | z=x+y} として記述・検証することが可能である。

２．研究の目的

本研究では、上記のリファインメント型の研究をさらに発展させ、より広範囲の仕様を記述・検証可能とするための理論構築と、リファインメント型検査・推論に基づいた関数型言語 OCaml のための全自動・高精度検証ツールの設計・実装を目指した。

３．研究の方法

本研究では、先行研究では記述・検証できなかった以下の 4 つの仕様に焦点を当てて研究を推進した。

(1) 停止性：高階再帰関数がすべての入力について停止するという仕様。
(2) 非停止性：高階再帰関数がある入力について停止しないという仕様。
(3) 再帰関数に関する関係的仕様：複数の関数呼び出しにおける入出力値間に成り立つ関係的仕様。例えば、関数の等価性、可換性、分配性、非干渉性、単調性、単射性。
(4) データ構造に関する関係的仕様：複数の代数データ構造の要素・形状間に成り立つ関係的仕様。例えば、2 つの木構造が鏡像になっているといった仕様。

これらの仕様の全自動・高精度検証を実現するため、それぞれについて、まず仕様を型として記述・検証できるようにリファインメント型システムを拡張し、次に拡張された型システムのための型検査・推論法を設計し、最後に型推論法を実装して検証ツールに組み込んで評価する、といった手順で研究を進めた。型システムの拡張においては、場当たり的な設計を避けるために、プログラムの表示的意味の（検証対象とする仕様ごとに決まる）抽象解釈を用いて、拡張リファインメント型システムをほぼ機械的な運算によって導出するといった工夫を行った。

４．研究成果

上記研究方法で論じた仕様(1)–(4)に対して、以下の成果を得た。

(1) 停止性：2009 年に研究代表者らが提案した部分正当性検証のためのリファインメント型システムおよびその型検査・推論法を拡張し、高レベルプログラムの完全正当性（停止性＋部分正当性）の全自動・高精度検証を実現した。さらに、提案手法の実装を行い、関数型言語 OCaml のための検証ツール Refinement Caml (RCaml) を構築した。提案手法は、完全正当性検証問題を、述語変数のホーン節・整礎性制約の解消問題に帰着・解消するというアプローチを採用している。本研究では、そのような制約の解消法についても研究・開発を行い、既存手法よりも多くの制約を解けることが実験的に示された新手法や、ある条件下では有限時間内に必ず解が求まるという望ましい性質を持った新手法の提案も行った。
(2) 非停止性：非停止性のように「ある入力について～が成り立つ」といった形をした仕様を、天使的非決定性を用いて記述・検証できるように、リファインメント型システムを拡張した。また、再帰関数が停止しない入力が存在する場合に、そのような入力に関するできるだけ弱い条件を推論できるようリファインメント型推論法を拡張した。具体的には、従来のリファインメント型推論問題を、多目的最適化問題として一般化したリファインメント型最適化問題を提案し、実際にそのような最適化問題を解くための、ホ

ーン節制約最適化に基づく新手法を開発した。これによって、停止性検証においても、単に停止するかどうかだけではなく、最悪の計算ステップ数を引き起こす入力条件の推論といった新しい応用が可能になった。

(3) 再帰関数に関する関係的仕様：関係的仕様を全自動・高精度検証するためのリファインメント型推論法の設計・実装を行った。提案手法は、帰納的定理証明とホーン節制約解消という従来別々に研究されてきた技術を相補的に組み合わせた新しい検証手法であり、様々なパラダイムの言語で記述された高レベルプログラム同士の関係的仕様の検証も可能である。

(4) データ構造に関する関係的仕様：ユーザ定義の帰納的述語を用いて複数の代数データ構造の要素・形状間に成り立つ関係的仕様を記述・検証できるようリファインメント型システムを拡張した。さらに、代数データ構造上のユーザ定義再帰関数を用いた不変条件の記述・検証にも対応し、ユーザ定義関数だけでは検証に必要な不変条件を表すことができない場合に、必要な関数を自動的に合成することが可能な型推論法の提案も行った。

これらの成果をまとめた論文は、システム検証に関するトップ会議である CAV やプログラム言語理論に関するトップ会議である POPL に採録されている。また、研究代表者は国際的な研究集会での招待講演を複数依頼されている。

５．主な発表論文等

〔雑誌論文〕（計　　9 件）

1. Akihiro Murase, Tachio Terauchi, Naoki Kobayashi, Ryosuke Sato, and Hiroshi Unno. Temporal Verification of Higher-order Functional Programs. Proceedings of POPL 2016, ACM, pp.57-68, 2016 (査読有) DOI: 10.1145/2837614.2837667
2. 松本雄磨, 小林直樹, 海野広志. 高階木変換器の自動検証のための反例発見と抽象化改良. コンピュータ・ソフトウェア 32(1), pp.161-178, 2015 (査読有) DOI: 10.11309/jssst.32.1_161
3. Yuma Matsumoto, Naoki Kobayashi, and Hiroshi Unno. Automata-Based Abstraction for Automated Verification of Higher-Order Tree-Processing Programs. Proceedings of APLAS 2015, Springer, LNCS 9458, pp.295-312, 2015 (査読有) DOI: 10.1007/978-3-319-26529-2_16
4. Kodai Hashimoto and Hiroshi Unno. Refinement Type Inference via Horn Constraint Optimization. Proceedings of SAS 2015, Springer, LNCS 9291, pp.199-216, 2015 (査読有) DOI: 10.1007/978-3-662-48288-9_12
5. Takuya Kuwahara, Ryosuke Sato, Hiroshi Unno, and Naoki Kobayashi. Predicate Abstraction and CEGAR for Disproving Termination of Higher-order Functional Programs. Proceedings of CAV 2015, Springer, LNCS 9207, pp.287-303, 2015 (査読有) DOI: 10.1007/978-3-319-21668-3_17
6. Hiroshi Unno and Tachio Terauchi. Inferring Simple Solutions to Recursion-free Horn Clauses via Sampling. Proceedings of TACAS 2015, Springer, LNCS 9035, pp.149-163, 2015 (査読有) DOI: 10.1007/978-3-662-46681-0_10
7. Tachio Terauchi and Hiroshi Unno. Relaxed Stratification: A New Approach to Practical Complete Predicate Refinement. Proceedings of ESOP 2015, Springer, LNCS 9032, pp.610-633, 2015 (査読有) DOI: 10.1007/978-3-662-46669-8_25
8. Takuya Kuwahara, Tachio Terauchi, Hiroshi Unno, and Naoki Kobayashi. Automatic Termination Verification for Higher-Order Functional Programs. Proceedings of ESOP 2014, LNCS 8410, pp.392-411, 2014 (査読有) DOI:10.1007/978-3-642-54833-8_21
9. 松本雄磨, 小林直樹, 海野広志. 高階木変換器の自動検証のための反例発見と抽象化改良. 第 16 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(PPL2014)予稿集, 18 ページ, 2014 年(査読有)

〔学会発表〕（計　　9 件）

1. Hiroshi Unno. Refinement Caml: A Refinement Type Checking and Inference Tool for OCaml. Dagstuhl Seminar on “Language Based Verification Tools for Functional Programs”, Wadern, Germany, March 31, 2016
2. Hiroshi Unno. Relational Verification of Functional Programs via Induction-based Horn Constraint Solving. NII Shonan Meeting on “Higher-Order Model Checking”, 湘南国際村センター（神奈川県三浦郡）, March 19, 2016
3. Hiroshi Unno. Verification of Featherweight Java Programs via Transformation to Higher-order Functional Programs with Recursive Data Types. NII Shonan Meeting on “Semantics and Verification of

Object-Oriented Languages”，湘南国際村センター（神奈川県三浦郡），September 21，2015

4. Sho Torii and Hiroshi Unno. Automating Well-Founded Induction for Horn Clause Solving，日本ソフトウェア科学会第 32 回大会予稿集，pp.1-4，早稲田大学（東京都新宿区），2015 年 9 月 11 日
5. Hiroshi Unno. Higher-order Program Verification as Refinement Type Inference. The 3rd Workshop on Higher-Order Program Analysis (HOPA 2015)，京都大学（京都府京都市），July 4，2015（招待講演）
6. Kodai Hashimoto and Hiroshi Unno. Refinement Type Inference via Multi-Objective Optimization Subject to Horn Clauses. The 12th Asian Symposium on Programming Languages and Systems (APLAS 2014)，Singapore，November 17，2014
7. 岡本大輔，海野広志．代数的データ構造を扱う関数型プログラムの検証法，日本ソフトウェア科学会第 31 回大会予稿集，pp.1-7，名古屋大学 東山キャンパス（愛知県名古屋市），2014 年 9 月 9 日
8. Takuya Kuwahara，Tachio Terauchi，Hiroshi Unno，and Naoki Kobayashi. Automatic Termination Verification for Higher-Order Functional Programs. 第 16 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(PPL2014)，阿蘇の司 ビラパークホテル(熊本県阿蘇市)，2014 年 3 月 6 日
9. Hiroshi Unno，Tachio Terauchi，and Naoki Kobayashi. Automating Relatively Complete Verification of Higher-order Functional Programs. 日本ソフトウェア科学会創設 30 周年記念大会，東京大学 本郷キャンパス（東京都文京区），2013 年 9 月 13 日（招待講演）

６．研究組織

(1)研究代表者

海野 広志（Unno，Hiroshi）
筑波大学・システム情報系・助教
研究者番号：80569575