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

令和　　元 年　　５ 月 ２８ 日現在

機関番号：１２６０１
研究種目：若手研究(B)
研究期間：2016～2018
課題番号：１６Ｋ１６００４

研究課題名（和文）ゲーム意味論と交差型システムによるプログラム検証

研究課題名（英文）Game semantics and intersection type systems for program verification

研究代表者
塚田　武志（Tsukada, Takeshi）

東京大学・大学院情報理工学系研究科・助教

研究者番号：５０７５８９５１

交付決定額（研究期間全体）：（直接経費）　　2,200,000円

研究成果の概要（和文）：ソフトウェアの信頼性の確保は重要な社会的課題であり、この観点から強力な型システムを持つ関数型プログラミング言語が注目を集めている。型システムごとに検出できるエラーの種類や検出精度は異なるため、目的に応じて適切な型システムを設計する必要がある。本研究では交差型システムと呼ばれる種類の型システムについて研究し、従来では研究者の経験と試行錯誤に依存していた交差型システムの設計プロセスに、ゲーム意味論と呼ばれる分野の数理が利用できることを示した。

研究成果の学術的意義や社会的意義
本研究成果は、経験と試行錯誤が頼りであった交差型システムの設計に、数理的な背景を見出したことである。数理的な背景とは、具体的には、ゲーム意味論と組合せ論である。ゲーム意味論は多くの研究があるものの、成果を理解し利用できる研究者の数が少ないという問題があったが、本研究成果によってゲーム意味論の知見を交差型システムの形で幅広い研究者に理解できる形で提供できる道が拓けた。また組合せ論との関わりを示したことで、交差型システムに新しい見方を提供し、例えば母関数などといった組合せ論の概念を交差型システムについて考えることを可能にした。

研究成果の概要（英文）：With growing concern about reliability of software systems, functional programming languages and their strong type systems are drawing attention. Each type system describes and ensures different properties of different programming languages, and one needs to design a type system that fits one's purpose. This research project focuses on a special class of type system, called intersection type system, and develops a general method for designing intersection types for a variety of languages. The method is based on game semantics, a mathematical framework for giving interactive semantics of programs.

研究分野：プログラム意味論

キーワード：ゲーム意味論　交差型システム　generalised species　線形論理

様　式　Ｃ－１９，Ｆ－１９－１，Ｚ－１９，ＣＫ－１９（共通）

１．研究開始当初の背景

ソフトウェアの信頼性の確保は重要な課題である．ソフトウェアに潜むバグをコードレビューやテストのみで無くすことは困難であり，プログラミング言語機構やプログラム解析器・検証器などによるサポートが不可欠である．

こうした背景から,近年,強力な型システムを持つプログラミング言語が注目を集めている．どのようなエラーを検出できるかは型システムによって異なるため，型システムによるプログラム検証手法を開発するには,「プログラミング言語」と「検出したいエラー」に応じた型システムを設計し，その正しさを証明する必要がある．

本研究では交差型システムというクラスの型システムに注目した．交差型システムの面白さは，プログラムの詳細な振る舞いを完全に特徴付けられる点にある．通常の型システムは近似の解析であって,「型が付けばエラーがない（健全性）」ことだけを保障し，その逆「エラーがないなら型が付く（完全性）」は成り立たない. 一方で、多くの交差型システムは完全性も持つ。交差型を理論的基礎とした全自動のプログラム検証が提案されており，目的に応じた交差型システムの設計は実用上の興味のある課題となっていた．

しかしながら，交差型システムの設計は研究者の経験と試行錯誤に頼るところが大きく，設計の指針となる原理がなかった．さらに悪いことに，交差型システムはプログラミング言語の変化に繊細であり，少しの変化が型システムの正しさを壊してしまいかねない．そのため，実用上の興味はあるが適切な交差型システムが知られていないケースも少なくない．例えば並行計算のモデルに対する交差型システムは知られていなかった．

２．研究の目的

本研究の目的は，交差型システムの背後に潜む数理を明らかにすることで，これまで経験と試行錯誤に頼っていた交差型システムの設計を，機械的に行えるようにすることである．具体的には次の２つを目的とした．

(1) 既知の交差型システムの数理的分析
関数型プログラミング言語については，多くの交差型システムが既に提案されていた．研究の第一のステップは，それらの既知の交差型システムの数理的な背景を明らかにすることである．

(2) 新しい交差型システムの開発
(1)で行った分析を元に，これまで交差型システムの知られていない計算体系に対する交差型システムの設計手法を開発する．具体的な対象の例としては，並行計算の体系であるπ計算が挙げられる．

３．研究の方法

「ゲーム意味論」が交差型システムの背後に潜む数理であると予想し，この予想を軸として研究を行う．

ゲーム意味論とはプログラム意味論（＝プログラムの意味の数学的な取り扱いを研究する分野）の分野のひとつであり，そのアプローチは直感的には次のように説明できる．分析対象のプログラムとそれを利用するクライアントプログラムが異なるマシンにあり，一定のプロトコルに従って通信することで計算を行っていると考える．このとき「分析対象のプログラムの意味」をクライアントを様々に変えたときの通信のログの集合だと考えるのがゲーム意味論である．もちろん「プロトコル」や「ログ」というのは比喩である．これらに相当する適切な数学的構造をプログラミング言語ごとに設計することがゲーム意味論の分野では行われてきた．

本研究の基礎を成すアイデアは，ログの構造が交差型システムの構造と一致するということである．上記の目的に照らし，以下の２つを行うことが研究の基本計画である．

(1) 関数型プログラミング言語について，交差型システムとゲーム意味論の対応の確立
「ログの構造が交差型システムの構造と一致する」という非形式的なアイデアを，数学に厳密な命題として表現し，その命題を証明する．

(2) ゲーム意味論に基づく交差型システムの開発
ゲーム意味論は知られているが，交差型システムが知られていない計算体系は多い．並行計算のモデルであるπ計算も，そうした計算体系の例である．ゲーム意味論のログの構造に対応するように交差型を設計することで，新しい交差型システムの提案を行う．

４．研究成果

(1) 関数型プログラミング言語の交差型システムの分析

以下に述べるように，当初の想定を超えた結果を得ることができた．

はじめに，交差型システムの構造とゲーム意味論のログの構造の対応について，研究計画段階の予想を証明することに成功した．具体的には，交差型システムの「型」とゲーム意味論の「ログ」の間に多対一の対応が存在することを示した．

さらに，当初の想定を超えて，微分λ計算という別の分野との関わりも明らかとなった．この結果は，大雑把に言えば，交差型システム，微分λ計算，そしてゲーム意味論の３つが本質的に同じ対象を記述していることを示しており，例えば微分λ計算の問題を解くのにゲーム意味論の伝統的なアイデアが使えることを示した．この結果はプログラム意味論分野のトップ国際会議である ACM/IEEE Symposium on Logic in Computer Science 2016 に採録された（雑誌論文⑤）．

そして，この結果を発展させることで，ゲーム意味論と交差型システムが組合せ論的なアイデアと繋がっていることを示した．得られた結果は，当初想定していたものに比べ，次の２つの点で深い．第一に，当初想定していた対応は「証明可能か否か」だけに注目する proof-irrelevant と呼ばれる種類のものであったが，今回得られたものは「何種類の証明が存在するか」まで保存する強い対応関係である（proof-relevant な対応関係と言われる）．第二に，組み合わせ論という別の分野との繋がりをも与える．具体的には，組み合わせ論の概念である Joyal's species を拡張した generalised species というものがあり，これが交差型システムの proof-relevant なものであるが，ゲーム意味論によるλ計算の解釈と generalised species による解釈が一致することを示した．そして，この対応関係を利用することで，非決定性プログラム・確率的プログラム・量子的プログラムなどを一様に扱うことのできる枠組みを与えることに成功した. この結果はトップ国際会議である ACM/IEEE Symposium on Logic in Computer Science 2017 および 2018 に採録された（雑誌論文③および②）．

(2) 並行計算の交差型システムの設計

はじめに，並行計算のモデルである非同期π計算にゲーム意味論を与え，そこから交差型システムを抽出した．得られた交差型システムは完全であったが，健全性は持たないという点で不満の残るものであった．健全性を持たないのは，並行計算に特有の現象であるデッドロックが原因であることを突き止め，（強い意味で）デッドロックの起きないプログラムについては，型システムが完全かつ健全であることを示した．以上の結果は論文にまとめられ，International Conference on Foundations of Software Science and Computation Structures 2017 に採録された（雑誌論文④）．

次いで，当初計画になかったデッドロックを扱えないという問題点の克服のため，π計算そのものについての研究を行った．具体的には，π計算の論理的な側面（＝カリー・ハワード・ランベック対応）を明らかにし，“論理的に素直な”π計算の断片を提案した．特に注目すべき点としては，“論理的に素直な”π計算は線形論理という論理体系の特殊な形であると理解できる点である．線形論理と交差型システムの関連は良く知られており，本結果はπ計算の交差型システムへの指針を与えることが期待できる．この結果は European Symposium on Programming 2019 で発表した（雑誌論文①）．

５．主な発表論文等

〔雑誌論文〕（計 10 件）

① Ken Sakayori and Takeshi Tsukada. A Categorical Model of an i/o-typed pi-calculus. In Proceedings of the 28th European Symposium on Programming, Springer, LNCS 11423, pp.640-667, 2019, 査読付き
doi:10.1007/978-3-030-17184-1_23

② Naoki Kobayashi, Takeshi Tsukada and Keiichi Watanabe. Higher-Order Program Verification via HFL Model Checking, In Proceedings of the 27th European Symposium on Programming, Springer, LNCS 10801, pp.711-738, 2018, 査読付き
doi:10.1007/978-3-319-89884-1_25

③ Takeshi Tsukada, Kazuyuki Asada and C.-H. Luke Ong. Species, Profunctors and Taylor Expansion Weighted by SMCC: A Unified Framework for Modelling Nondeterministic, Probabilistic and Quantum Programs. In proceedings of the 33rd Annual ACM/IEEE Symposium on Logic in Computer, ACM, pp.889-898, 2018, 査読付き
doi:10.1145/3209108.3209157

④ Takeshi Tsukada, Kazuyuki Asada and C.-H. Luke Ong. Generalised species of rigid resource terms, In Proceedings of the 32nd Annual ACM/IEEE Symposium on Logic in Computer Science, IEEE Computer Society, pp.1-12, 2017, 査読付き
doi:10.1109/LICS.2017.8005093

⑤ Ryota Suzuki, Koichi Fujima, Naoki Kobayashi and Takeshi Tsukada. Streett Automata Model Checking of Higher-Order Recursion Schemes, In Proceedings of the 2nd International Conference on Formal Structures for Computation and Deduction, Schloss Dagstuhl, LIPIcs

84, pp.32:1-32:18, 2017, 査読付き
doi:10.4230/LIPIcs.FSCD.2017.32

⑥ Ryoma Sin'ya, Kazuyuki Asada, Naoki Kobayashi and Takeshi Tsukada. Almost Every Simply Typed Lambda-Term Has a Long Beta-Reduction Sequence, In Proceedings of the 20th International Conference on Foundations of Software Science and Computation Structures, Springer, LNCS 10203, pp.53-68, 2017, 査読付き
doi:10.1007/978-3-662-54458-7_4

⑦ Takashi Suwa, Takeshi Tsukada, Naoki Kobayashi and Atsushi Igarashi. Verification of code generators via higher-order model checking, In Proceedings of the 2017 ACM SIGPLAN Workshop on Partial Evaluation and Program Manipulation, ACM, pp.59-70, 2017, 査読付き

⑧ Ken Sakayori and Takeshi Tsukada. A Truly Concurrent Game Model of the Asynchronous pi-Calculus, In Proceedings of the 20th International Conference on Foundations of Software Science and Computation Structures, Springer, LNCS 10203, pp.389-406, 2017, 査読付き
doi:10.1007/978-3-662-54458-7_23

⑨ Takeshi Tsukada and C.-H. Luke Ong. Plays as Resource Terms via Non-idempotent Intersection Types, In Proceedings of the 31st Annual ACM/IEEE Symposium on Logic in Computer, ACM, pp.237-246, 2016, 査読付き
doi: 10.1145/2933575.2934553

⑩ Kazuhide Yasukata, Takeshi Tsukada and Naoki Kobayashi. Verification of Higher-Order Concurrent Programs with Dynamic Resource Creation, In Proceedings of the 14th Asian Symposium on Programming Languages and Systems, Springer, LNCS 10017, pp.335-353, 2016, 査読付き
doi:10.1007/978-3-319-47958-3_18

〔学会発表〕（計 1 件）

① Takeshi Tsukada. Strategies in HO/N games as profunctors. Games for Logic and Programming Languages XI, 2016

〔図書〕（計 0 件）

〔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
なし

６．研究組織

(1)研究分担者
なし

(2)研究協力者
なし