



研究室 多元数理科学棟 405号室 (内線番号 4661)

電子メール garrigue@math.nagoya-u.ac.jp

<http://www.math.nagoya-u.ac.jp/~garrigue/home-j.html>

所属学会 日本ソフトウェア科学会, 情報処理学会, ACM

研究テーマ

- プログラミング言語の基礎理論
- 型付ラムダ計算と型推論
- 定理証明支援系の応用

研究テーマの概要

私の研究の目的は、プログラミング言語の表現力を損わずにバグを徹底的に排除することである。プログラムが普遍的になった今、安全性に不可欠な努力だと思う。この目的に合わせて色々な方針が考えられるが、主にプログラミング言語の型システムとプログラムの証明を対象として来た。

OCamlという関数型プログラミング言語の開発を通して、表現力と安全性の両立を追求して来た。関数型プログラミング言語は型付ラムダ計算に基いており、元々論理との関係が強い。この論理との関係から健全性という重要な概念が得られる。プログラミング言語の型システムが健全であるというのは、プログラムが実行時に型エラーを起こさないという重要な性質を保証する。

こういう強い型システムを負担なくに使うために、型情報を自動的に補う型推論と関数などを複数の文脈で利用可能にする多相性が重要である。それらについていくつかの貢献をしてきた [1, 4]。

しかし、型システムの実装が本当に理論に合っているかどうかを保証しにくいので、もっと強い安全性の必要性を感じている。そのために Coq という型理論に基いた定理支援系を用いて簡単なプログラミング言語の実装と検証を行った [2]。

また、産業技術総合研究所の Reynald Affeldt や名古屋大学の学生と一緒に、Coq の数学やシステムの証明への応用も試みている。特に、符号理論における線形符号の証明や簡潔データ構造を扱ったプログラムの証明を一緒に行った [3, 5]。

上の具体的なプログラミング言語や定理証明支援系における研究以外にも、論理学や計算可能性に強い興味をもっている。

主要論文・著書

- [1] J. Garrigue and D. Rémy, Extending ML with semi-explicit first class polymorphism. *Information and Computation* **155** (1999), 134–171.
- [2] J. Garrigue, A Certified Implementation of ML with Structural Polymorphism and Recursive Types. *Mathematical Structures in Computer Science* (2014) 11:1–25.
- [3] R. Affeldt, J. Garrigue, T. Saikawa, A Library for Formalization of Linear Error-correcting Codes. *Journal of Automated Reasoning* (2020) 64:1123–1164.
- [4] J. Garrigue and J. Le Normand. GADTs and exhaustiveness: Looking for the impossible. In *Proc. ML Family / OCaml Workshops 2015*, number 241 in EPTCS (2017), 23–35.
- [5] R. Affeldt, J. Garrigue, X. Qi, K. Tanaka, Proving tree algorithms for succinct data structures. In *Proc. 10th International Conference on Interactive Theorem Proving, LIPICs* (2019).
- [6] 大堀淳・ジャックガリグ・西村進, 『コンピュータサイエンス入門・アルゴリズムとプログラミング言語』, 岩波書店, 1999.

受賞歴

- 2010年, 日本ソフトウェア科学会高橋奨励賞, 「構造的多相性をもった言語の検証つきインタープリタ」

経歴

- 1995年 東京大学大学院理学系研究科博士課程修了
- 1995年 京都大学数理解析研究所助手
- 2004年 名古屋大大学院学多元数理科学研究科助教授 (後に准教授)
- 2018年 同研究科教授

学生へのメッセージ

自分の研究テーマは関数型プログラミング言語に直接な関わりをもったものであるが, 少人数クラスではもっと論理学寄りのテーマを扱って来た. 主に計算可能性, ラムダ計算の型理論および論理学とその計算機への応用である.

計算可能性 (または計算論) はアルゴリズムが存在する関数のクラスを研究している. 数学で定義されている関数の中で, プログラムによって計算できるものと計算できないものが存在する. しかも, その違いは使っているプログラミング言語に依存しない. それを論理学に応用すると, 全ての定理を自動的に証明するプログラムがありえないという結果につながる. テキストとして以下のものを使って来た.

1. Neil D. Jones, *Computability and Complexity from a Programming Perspective*. MIT Press, 1995.
2. 高橋正子, 『計算論』. 近代科学社, 1991.

ラムダ計算は研究の概要に少し紹介したが, 論理学と計算機科学を結ぶ一つの枠組といえる. その型理論はプログラミング言語の設計や定理証明支援系の構築に応用される. 少人数クラスでは以下のようなテキストを使って来た.

1. 大堀 淳, 『プログラミング言語の基礎理論』. 共立出版, 1997.
2. Benjamin C. Pierce, *Types and Programming Languages*. MIT Press, 2002. 和訳『型システム入門—プログラミング言語と型の理論』. オウム社, 2013.
3. Mingsheng Ying, *Foundations of Quantum Programming*. Morgan Kaufmann, 2016. 和訳『量子プログラミングの基礎』共立出版, 2017.
4. Franz Baader, Tobias Nipkow, *Term Rewriting and All That*. Cambridge Univ. Press, 1998.

論理学はそれぞれの分野の理論的な基礎になると同時に, 命題の正しさ具体的な手法 (レゾリューションなど) も提供している. 少人数クラスでは以下の本を使って来た.

1. Jean Gallier, *Logic for computer science*. Online edition, 1986.
2. John Harrison, *Handbook of practical logic and automated reasoning*. Cambridge University Press, 2009.
3. Jean-Yves Girard, *The Blind Spot: Lectures on Logic*. European Mathematical Society, 2011.

そのぞれのテーマについて, 予備知識は特に要らないが, 論理学の基礎知識があると役に立つ. 修士論文では, 理論計算機科学の様々なテーマで指導できる.