

Peng Fu

Email: peng-fu@uiowa.edu
Homepage: <https://fermat.github.io/>

Education

- Ph.D. Department of Computer Science, the University of Iowa, Iowa City, USA. August 2014.
- B.Eng. School of Computer Science, Huazhong University of Science and Technology, China. July 2009.

Research Interests

- Lambda calculus, type theory and their applications.
- Theorem proving and language-based verification.
- Type systems for functional programming languages.

Conference Publications

1. **Proof Relevant Corecursive Resolution.**

Peng Fu, Ekaterina Komendantskaya, Tom Schrijvers, Andrew Pond. International Symposium on Functional and Logic Programming, FLOPS 2016.

2. **A Type-Theoretic Approach to Resolution.**

Peng Fu, Ekaterina Komendantskaya. International Symposium on Logic-Based Program Synthesis and Transformation, LOPSTR 2015.

3. **Self Types for Dependently Typed Lambda Encodings.**

Peng Fu, Aaron Stump. Joint 25th International Conference on Rewriting Techniques and Applications and 12th International Conference on Typed Lambda Calculi and Applications, RTA-TLCA 2014.

Journal and Workshop Publications

1. **Efficiency of Lambda-Encodings in Total Type Theory.**

Aaron Stump, **Peng Fu**. Journal of Functional Programming, 2016.

2. **A Framework for Internalizing Relations into Type Theory.**

Peng Fu, Aaron Stump, Jeff Vaughan. International Workshop on Proof-Search in Axiomatic Theories and Type Theories, PSATTT 2011.

Dissertation

- Title: **Lambda Encodings in Type Theory.**
- Summary: The dissertation explores the reasoning of Scott-encoded programs using the comprehension principle.
- Committee: Aaron Stump, Cesare Tinelli, Kasturi Varadarajan, Ted Herman, Douglas Jones.

Research Projects

- **Functional Certification of Rewriting (FCR)**¹. A prototype type checker for analyzing and proving the nontermination of term rewriting system. Main features:
 - The certification of nonterminating rewriting is reduced to type checking.
 - The type checking algorithm is based on resolution with second-order matching.
- **Corecursive Type Class**². A prototype interpreter and type checker that implements the type class mechanism based on corecursive resolution. Main features:
 - It supports dictionary construction for nonterminating type class resolution.
 - It uses goal directed automated proof construction to construct type class evidence.
 - It provides a heuristic for generating intermediate lemma for proof construction.
- **The Gottlob System**³. A prototype interpreter for typed functional programming and theorem proving. Main features:
 - The functional programming fragment is equipped with Hindley-Miller type inference. The core language is based entirely on Scott encoding, without build-in data-type and pattern matching.
 - The theorem proving fragment can reason about the program with general recursion.
 - It can automatically synthesize an induction principle from an algebraic data type declaration, induction principle is not primitive in Gottlob.

Academic Positions

- Postdoctoral Research Assistant, Heriot-Watt University, UK. March 2016 – August 2016.
- Postdoctoral Research Assistant, University of Dundee, UK. October 2014 – March 2016.
- Teaching Assistant, “Algorithm and AI”, 2015 Spring. Computer Science, The University of Dundee.
 - Taught basic functional programming in Haskell, the first Haskell class taught in University of Dundee.
 - Delivered one lecture per week (total 13 lectures, class size: around 50).
 - Ran one lab session per week.
 - Developed class materials, homeworks and part of the final exam.
- Graduate Teaching Assistant, “Programming Language Concepts”, 2013 Spring, 2014 Spring. Department of Computer Science, The University of Iowa.
 - Graded assignments (Class size: around 70 both times).
 - Ran weekly office hours.
- Graduate Teaching Assistant, “Object-Oriented Software Development ”, 2013 Fall. Department of Computer Science, The University of Iowa.
 - Graded assignments (Class size: around 70).
 - Ran one lab session per week.

¹Source code available from: <https://github.com/fermat/fcr>

²Source code available from: <https://github.com/fermat/corecursive-type-class>

³Source code available from: <https://github.com/fermat/gottlob>

- Ran weekly office hours.
- Graduate Teaching Assistant, “Computer Networking”, 2009 Fall. Department of Computer Science, The University of Iowa.
 - Graded assignments (Class size: around 30).
 - Ran weekly office hours.

Professional Service

- External Reviewer. 24th International Conference on Rewriting Techniques and Applications (RTA 2013). Reviewed: 1 paper.
- External Reviewer. 19th International Conference on Foundations of Software Science and Computation Structures (FoSSaCS 2016). Reviewed: 1 paper.
- External Reviewer. 32nd International Conference on Logic Programming (ICLP 2016). Reviewed: 1 paper.