

Lausanne, 1007, Switzerland

fengyun.liu.cs@gmail.com

fengy.me

+41 (0)78 793 1094

ABOUT ME I value critical thinking, careful analysis, and the usage of simple abstractions to solve complex problems. I handle knowledge explosion by [coming back to basics](#) and reading papers by the need of projects.

WORK EXPERIENCE *Senior Researcher* *2021.6 – present*
Oracle Labs, Switzerland

Compiler Developer for Scala *2016.3 – 2021.6*
EPFL, Switzerland

CONTRIBUTIONS **The Scala Programming Language** ¹
I work extensively on the Scala 3 compiler, [Dotty](#), as a core contributor. I tinker with almost every part of the compiler, from parser, typer, pattern matcher to backend.

A Regressional Benchmarking Framework ²
I conceptualized and implemented a low-maintenance, secure and flexible regressional benchmarking framework for Dotty. (1) Zero databases, all data is stored on github; (2) UI is Github static pages; (3) support test on open PRs via comments; (4) support test groups.

Exhaustivity Check
I propose a novel algorithm for checking exhaustivity of pattern matches, which is integrated in the Scala 3 compiler. The algorithm is later adopted in Swift ³ by Apple and in Dart ⁴ by Google.

Safe Initialization of Objects
I conduct original research on the theory and algorithms for safe object initialization in programming languages. The work produces several publications at OOPSLA and is recognized by experts as advancing the design and implementation of object-oriented programming languages.

Implicit State Machines
I propose *implicit state machines*, which is a new theoretical foundation for programming digital circuits and embedded systems. The work is accepted for publication at LCTES, recognizing its potential for innovations and broad applications.

¹<https://scala-lang.org>

²<https://github.com/lampepfl/bench>

³<https://github.com/apple/swift/pull/8908>

⁴See exhaustiveness.md in <https://github.com/dart-lang/language/pull/2616>

EDUCATION	PhD in Computer Science <i>EPFL, Switzerland</i> Advisor: Prof. Martin Odersky	<i>2016.6 – 2020.6</i>
	Master in Computer Science <i>EPFL, Switzerland</i>	<i>2014.9 – 2016.2</i>
	Master in Western Philosophy <i>Nanjing University, China</i>	<i>2007.9 – 2010.6</i>
	Bachelor in Software Engineering <i>Nanjing University, China</i>	<i>2003.9 – 2007.6</i>

PUBLICATION	A Conceptual Framework for Safe Object Initialization <i>C. Blauddau, F. Liu, OOPSLA, 2022</i>
	Implicit State Machines <i>F. Liu, A. Prokopec, LCTES, 2022</i>
	Safe Object Initialization, Abstractly <i>F. Liu, O. Lhotak, E. Xing, and N. C. Pham, Scala Symposium, 2021</i>
	A Type-and-Effect System for Safe Initialization <i>F. Liu, O. Lhotak, A. Biboudis, P. Giarrusso, M. Odersky, OOPSLA, 2020</i>
	Theory and Practice of Coroutines with Snapshots <i>A. Prokopec, F. Liu, ECOOP, 2018</i>
	Simplicity: Foundations and Applications of Implicit Function Types <i>M. Odersky, O. Blanvillain, F. Liu, A. Biboudis, H. Miller et al, POPL, 2017</i>
	A Generic Algorithm for Checking Exhaustivity of Pattern Matching <i>F. Liu, Scala Symposium, 2016</i>

SERVICE	LCTES 2023, Review Committee Member
	CC 2023, Review Committee Member
	Scala 2022, Review Committee Member
	Scala 2020, Review Committee Member
	ICFP 2019, Artefact Evaluation Committee Member
	ICFP 2018, Artefact Evaluation Committee Member
	PLDI 2018, Artefact Evaluation Committee Member
Reference: https://conf.researchr.org/profile/fengyunliu	