Eleftherios "Lef" Ioannidis

Info

Address: 235 Albany St, Cambridge, MA
Phone: +1 857 294 6849
email: elefthei@mit.edu

Education

Dec 2017 - Jan 2019 (Expected) Master's of Engineering in Computer Science, *CSAIL MIT*, Cambridge, MA

Thesis: *An implementation of Coq's Gallina in C++17*

Advisors: Frans Kaashoek, Nickolai Zeldovich, Adam Chlipala, CSAIL

Sep 2011 - Jun 2015 Bachelor's in Computer Science, *MIT*, Cambridge, MA

Major: Electrical Engineering and Computer Science

Thesis: *Parallel Instructions for the LLVM Compiler*

Advisors: Saman Amarasinghe, Charles Leiserson, CSAIL

Academic Work

Today Teaching Assistant, MIT, Cambridge, MA

6.828: Operating Systems. Graduate level course, teaching students to build operating systems based on xv6. Taught by Frans Kaashoek, Adam Belay.

6.858: Computer Systems Security. Graduate level course, focused on attacking and defending Computer Systems. Buffer overflows, Return-to-libc, Symbolic Execution, POSIX security, Cryptography, Web Security, Distributed Systems, Blockchain. Taught by Nickolai Zeldovich, Frans Kaashoek.

Today Graduate Researcher, PDOS, CSAIL, Cambridge, MA

Formal Verification of Computer Systems. Building a compiler for the Functional, Verified Language of the Coq Proof Assistant, while formally verifying a Mail Server and a Payment Application.

Sep 2015 - Jan 2016 Teaching Assistant, NTUA, Athens, Greece

Helped develop problem sets for the Cryptography class, taught in the National Technical University of Athens.

Sep 2014 - Jan 2015 Undergraduate Researcher, PL Group, CSAIL, Cambridge, MA

SuperUROP Program for Advanced Undergraduate Research. Year-long research program, worked on developing new parallel extensions for polyhedral Optimizations in LLVM, targeted at the Halide Domain-specific Language
http://www.halide-lang.org/

Jan 2012 - Jan 2015 Class Instructor, SIPB IAP, Cambridge, MA

Taught the *Secure Programming in C* class in MIT, organized by the Student Information Processing Board every January.
https://github.com/elefthei/secure-C

Nov 2010 - Jun 2011 Researcher, Aristotle University of Thessaloniki, Greece

FLOW Walkers, a simulation of human behavior in groups. Using flocking, finite and stochastic State Machines and more, automata walk in a room based on deterministic rules, as well as rules of attraction and repulsion. Generated interesting 3D graphics of human-like walkers in using Matlab and DarkBasic.

Conference Talks

July 2018	Making the Rooster fly, DeepSpec 2018, Princeton, NJ.
September 2017	Data aware Nginx for Load Balancing ML data, Nginxconf 2017, Portland, OR.
April 2017	Scalable ML microservices on GPUs, Dockercon 2017, Austin, TX.
February 2017	Secure, Real-time data collection on mobiles, MadCon 2017, Austin, TX.
April 2015	Parallel and Distributed Extensions to LLVM, MIT Undergraduate Research Conference and Journal 2015, Cambridge, MA.

Industry Work

May 2016 - Oct 2017 — Principal Engineer, UnifyID, San Francisco, CA

First Engineer of UnifyID. While working there, the startup won the 2^{nd} place at TechCrunch Disrupt Battlefield 2016, 1^{st} place at the RSA Innovation Sandbox 2017, 1^{st} place in the Security and Privacy Category, SXSW 2017. Designed and implemented the UnifyID back-end for secure, implicit authenticaton. Designed services as a scalable, Distributed microservice back-end, for multiple User & Device authentication. Designed and implemented Data Collection clients and real-time Machine Learning algorithms, designed and implemented all end-to-end encryption protocols. Managed a team of 10 engineers and handled all technical interviews.

Aug 2015 - May 2016 — Security Engineer, Apple, Cupertino, CA

FairPlay and DRM group. Worked on Application Security, Compilers, Reverse Engineering. Ensured the FairPlay daemon and FairPlay DRM suite are immune to static and runtime attacks.

Jun 2014 - Sep 2014 — Graduate Firmware Engineer Intern, Intel, Hillsboro, OR

Internet of Things (IoT) research branch of Intel. Designed and engineered firmware for Intel embedded microprocessors in a wireless mesh network configuration for large-scale Datacenter monitoring. Published a whitepaper in the Intel internal library.

Mar 2013 - Sep 2015 — Systems and Network Administrator, MIT MediaLab, Cambridge, MA

Mobile Experience Lab, Comparative Media Studies Group, MIT. Responsible for the uptime and maintenance of 5 physical servers and over 100 websites. Software stack was Debian and Xen, running Apache, Django, Kerberos, LDAP, SSH, LDAP and more.

Jun 2013 - Aug 2013 — Software Engineer Intern, MokaFive, Redwood City, CA

LiveCloud^{TM}, a cross-platform, secure cloud storage system. Developed a blacklisting system and Peer-to-Peer file transfers.

Nov 2011 - Jun 2012 — Mobile Security Analyst Intern, Securigin, Cambridge, MA

Implemented an automated iOS app penetration testing framework, pen-tested thousands of apps and found hundreds of generic network and memory management vulnerabilities. Selected for MIT100K entrepreneurship competition.

Patents

March 2018 — Privacy preserving System for ML Training Data (us UFID18-1001)

Privacy-preserving system based on Intel SGX Secure Enclaves. Enables high-performance and GPU computing on anonymized user data for Machine Learning, without exposing the identity of the user to the operator, cloud owner or engineer.

Open Source Projects

October 2017 — WebTorch: A Deep Learning web server based on Nginx

WebTorch takes advantage of the versatile LuaJIT compiler to bring together the Nginx web serve and Torch, creating an equivalent system to Tensorflow serve, based on a REST API. Multiple data formats are supported and WebTorch was built with performance and ease-of-use in mind. Presented at Nginxconf 2017.
https://github.com/elefthei/WebTorch

April 2017 — SLOB: Sticky Load Balancer

A Sticky-session Load Balancer using DynDNS for Machine Learning data, proof-of-concept. Presented at Dockercon 2017.
https://github.com/elefthei/slob-poc

July 2016 — Krypto: OpenSSL bindings for Elixir

Krypto is an easy to use, light-weight, crypto implementation for Elixir with native OpenSSL bindings for common crytpographic operations.
https://github.com/elefthei/krypto

October 2016 — Lua Machine Learning Microservice

A waffle based web server written in Lua, that integrates with S3 for storage and Torch for Deep Learning on real-time data. Dokcer container available for fast deployment. https://github.com/elefthei/lua-ml-microservice

September 2016 — Python Machine Learning Microservice

A flask based web server written in Python, that integrates with S3 for storage and Keras/TensorFlow for Deep Learning on real-time data. Docker container available for fast deployment. Presented at Dockercon 2017.
https://github.com/elefthei/python-ml-microservice

May 2015 — Eclipse Orion, an Open-source cloud IDE by Eclipse

Added GIT support for the NodeJS implementation of Eclipse Orion. Worked in a team of four students, with technologies like NodeJS, ConnectJS, NodeGit, Javascript, Mocha. Final project for 6.S194 Open Source Lab, Spring 2015.
https://github.com/eclipse/orion.client

December 2012 — PAL: Progressive Authentication for Linux

The PAL system sets up a progressive authentication scheme for Linux systems. Upon login, users are given a fresh, anonymous user with no privileges, and users can request additional privileges to be given to them on a per-need basis. Final project for 6.858 Computer Systems Security, Fall 2012.
https://github.com/elefthei/PAL--Progressive-Authentication-for-Linux

March 2011 — Logistic Map Cipher

An encryption standard based on the chaotic properties of the logistic map function. Code available for both Matlab and C/C++.
https://github.com/elefthei/lmcipher

Software Engineering

Skills:	Programming Languages, Computer Security, Formal Verification, Functional Programming, Systems, Architecture, Compilers, Performance Engineering.
Interests:	Category Theory, Dependent Types, Hardware Architecture, HPC.
Languages (expert):	Haskell, C/C++, Go, Javascript, Python, SQL, LLVM, x86 ASM, Bash.
Languages (intermediate):	Coq, Elixir, Lua, Swift, LaTeX.
Software:	Linux, Docker, Kubernetes, MongoDB, PostgreSQL, GIT, LLVM, GDB.

Extracurricular Activities

Feb 2018 - Jan 2019 Office Commissar, SIPB, MIT

Office Commissar for the Student Information Processing Board of MIT. The SIPB is the oldest volunteer computing group in the US. Our services include but not limited to; the Athena project, the Scripts Compute cluster, the Debathena Linux distribution and the Hyades compute cluster, all of which are free software and available to MIT students and faculty. As Office Czar I helped renovate the SIPB office and repurpose space for use by current student projects. Contributed code in the development of the Scripts compute cluster, the Debathena Linux distribution and most recently the Hyades compute cluster.

Feb 2016 - Jun 2016 Electric Guitar player, Blue Bear School of Music, San Francisco, CA

Guitar Player in the Blue Bear school of music in San Francisco, CA. Completed all intermediate classes and performed live with a full band in *The Boom Boom Room* in Filmore St.

Sep 2014 - Jan 2015 Social Chair, The No6 Club, MIT

Social chair for the No6 club, a co-ed Independent Living Group in MIT. Organized dinners and social events such as Coffee Hours, where a renowned speaker was invited to give a lecture and discuss with the students. The most notable lecturer was Noam Chomsky.

Dear ACT school student selection committee,

I'm writing to recommend Lef Ioannidis to the ACT school. He's taking my "7 Sketches" course and is enthusiastic and quite intelligent. He comes from a functional programming background, working with Adam Chlipala in the MIT computer science department.

I've already recommended Jabari King, who would be my first choice for a variety of reasons. But Lef appears to be equally good, though I haven't worked with him directly. He certainly would make a great addition to the program.

Best,
David

ACT 2019 Application

Eleftherios "Lef" Ioannidis

January 25, 2019

1 Relevant background in category theory or any of the specific projects areas

As part of my research in programming languages and formal verification I have had the pleasure of working with categorical abstractions every day. For my Master's thesis in CSAIL, MIT I developed a compiler for the Coq proof-assistant language. The target program Abstract Syntax Tree (AST) is represented as a traversable in Haskell and optimization passes as endofunctors in the category of Hindley-Milner types and expressions. Haskell lenses are particularly useful for structured term rewriting, which raised my attention to Bartosz Milewski's problem of optical traversals, my first project of choice. Structured term rewriting is vital in programming language development and profunctor optics appear like the ideal candidate for its implementation.

Apart from functional programming I enjoy category theory in itself. I am currently taking David Spivak's *"Seven sketches in compositionality class"* in MIT, which teaches category theory applied to science and engineering. Before that, I was a grader for Ronald Rivest's cryptography class which offers an algebraic number-theoretical approach to cryptography. Before that I was a teaching assistant for Albert Mayer's Discrete Mathematics class. Category theory has given me the theoretical depth necessary to understand structure in a variety of fields and how it is expressed in terms of Categories, Functors and Monads.

Another familiar problem that would be a good candidate for me to work on is Tobias Fritz proposal for a categorical model for partial evaluation. Coq's language and standard library depends heavily on partial evaluation, partial and total maps and their respective proofs and my compiler offers partial evaluation support at compile-time using zero-cost abstractions in

C++17. My initial approach to a categorical description for partial evaluation would be a mapping of n-ary morphisms to objects in the category of Cartesian Closures (CC), for which composition is straightforward. Then it would be interesting to explore adjunctions between n-arry functions and their partial evaluation results.

I expect to graduate from my Master's in MIT this June and begin a doctoral program in September 2018, for which I am currently awaiting the application decisions from US universities. Participation in the ACT program would help me solidify my knowledge of Category Theory both theoretically and in an applied context. My research goal is to make the development and execution of formally verified software practical, by reimagining software abstractions from a categorical perspective. Using categorical abstractions for designing and implementing large systems with formal guarantees build-in seems like the best approach towards achieving that goal and attending ACT would help be get a few steps closer.

2 Order of project preference

1. Bartosz Milewski
2. Tobias Fritz
3. Pieter Hofstra
4. David Spivak
5. Miriam Backens
6. Mehrnoosh Sadrzadeh

2.1 Coming to Oxford

I can pay for transportation from and to Oxford.

3 Statement of Interest

From my participation in the ACT 2019 summer school I expect to make significant contributions to an open problem in Category theory, solidify my theoretical foundations of the subject and connect with other young researchers in the field. I hope to contribute significantly to what will be a paper and/or a functional programming library but more importantly, I

hope to gain fluency in categorical abstractions. I believe Category theory is akin to a swiss-army-knife of mathematical formalism that can be used for software engineering, modeling, designing and building large systems.

It is no fad that every year formally verified systems are featured in computer systems conferences (OSDI, Usenix etc). While formal verification has shown great advances, making it possible to prove operating systems, distributed applications and more, traditional software engineering is failing to keep up with side-channel attacks. A common example is the Spectre attack in Intel CPUs, a side-channel attack on branch prediction, embedded into Intel CPUs for so long it is now impossible to remove without a big performance regression. I believe Category Theory can help in the systematic formalization of resource use and side-effects in the hardware/software interface and has some potential for mitigating side-channel attacks such as Spectre. It is thus my research goal to use categories for bridging the gap between theoretical formalism and practical computer systems engineering.

Finally, in terms of career development, participation in the ACT 2019 summer school will allow me to start publishing in category theoretical conferences such as ICFP. I have been able to participate in Systems conferences (OSDI), Formal Verification conferences (NFM), however, I have not been able to publish in ICFP yet, which is one of my goals going into my doctoral program. I believe ACT will ease my transition into a PhD program and help me meet future collaborators, who care about programming abstractions and their application to building computer systems as much as I do.