

# Nathan Lilienthal

nathan@nixpulvis.com ◦ <https://nixpulvis.com> ◦ +1-202-701-4368

---

**Languages:** Ruby, Rust, C/C++, LUA, Racket, Shell. ECMAScript (JS), Python, Java, and more...

**Systems:** UNIX, Linux, Git + Hub/Lab, Rails, Heroku, Postgres (PostGIS), SQLite, AVR/ARM, WoW (ask me about it).

---

## Professional Experience

- **Northeastern University** **Boston, MA**  
*Research Programmer, Intelligence Advanced Research Projects Activity, HECTOR* *Aug. 2019 ~ May 2021*
  - Collaboratively developed a hybrid-mode secure programming language design for multi-party computation (MPC)
  - Represented my team at both remote and in-person technical exchange meetings with other researchers
  - Built a prototype implementation of our language, which is forked from the Rust programming language
  - Began a formalism for our language(s), which will include sound typing rules, and reductions
- **Forward Financing Inc.** **Boston, MA**  
*Sr. Software Engineer* *May 2018 – Aug. 2019*
  - Developed a client wrapper for an Algolia search implementation
  - Quickly performed various application performance improvements, often caused by unacceptable response times
  - Planned architecture refactoring, including object model improvements and a new data permissions system
  - Led efforts to create an orchestration CLI for managing a complex Heroku + Salesforce microservice system
  - Mentored co-op university students by providing deep code reviews and pair programming
- **HOMER Energy** **Boulder, CO**  
*Software Developer, Summer Intern* *Jul. 2017 - Nov. 2017, Summer 2012*
  - Built API integrations for the HOMER C# application, including REST and CSV file APIs which involved a general refactoring of the code which imports data, complete with added tests
  - Developed an internal tool to view the Google Protocol Buffer used to pass values between all parts of the application allowing developers to quickly see inputs and outputs
  - Created a web-based frontend for HOMER in Rails, which served as the starting point for another version and provided a proof of concept for how to integrate the HOMER API with a webserver
- **Apple Inc.** **Cupertino, CA**  
*Software Engineer* *Jan. 2015 – Aug. 2015, Jul. 2016 – Jul. 2017*
  - Built a Ruby library (**radic**) and CLI (**radish**) for interacting with Apple's bug management system (aka Radar)
  - Participated in a cross-cutting web design work group to help create common components for the hardware teams
  - Contributed to an internal tool for managing hardware validation, inspired in part by Travis CI
  - Contributed to an internal tool for analysing large amounts of pre-production device test data
- **Americas Test Kitchen** **Boston, MA**  
*Web Developer* *Jan. 2014 – June 2014*
  - Pushed code to the frontend and backend for all four Americas Test Kitchen websites, including bug fixes and technical infrastructure upgrades
  - Built modularized components to abstract functionalities found common throughout the company's codebase
- **Bluesocket - Adtran** **Burlington, MA**  
*Software Developer* *Jan. 2013 – June 2013*
  - Developed an automated build system, which reduced turnaround time by allowing anyone to easily run a build
  - Addressed user reported issues in Ruby/Rails and LUA, including hardening validations and updating database migrations for old versions of the software
  - Designed a class/model structure for users and accesspoints, which allowed the backend to represent clients of individual accesspoints

---

**Other Interests:** Microelectronics, Music, Woodworking, Billiards, Environmentalism, Travel & Culture, Gaming, Skiing, Frisbee, Cats, and much more ...

---

## Education & Projects

### • Northeastern University

Boston, MA

College of Computer and Information Science

2011 – 2016

Bachelor of Science in Computer Science

- **Relevant Courses:** Programming Languages, Special Topics in Programming Languages, Compilers, GPU Programming & Architecture, Systems and Networks, Computer Organization, Software Development (aka HELL), Theory of Computation, Algorithms and Data Structures, Fundamentals of Computer Science 1 & 2, Object Oriented Design, Artificial Intelligence, Logic and Computation, Combinatorics.
- **Clubs & Extracurriculars:** Association for Computing Machinery, NU Hacks, Hack Beanpot.

Teaching Assistant, Fundamentals of Computer Science 1

Fall of 2012, 2013, 2014, and 2015

- Conducted short lectures before each lab and then lead the lab's students and tutors in the weekly assignments
- Discovered new ways to present concepts that facilitated student understanding
- Assisted students at established office hours and online
- Participated and lead course administrative tasks, including grading, meta-grading, testing, rubric development, and weekly teaching staff meetings

Independent Study

Spring 2014

A Heterogeneous System Simulator

<http://www.multi2sim.org>

- Investigated caching protocols, including MOSI and MOESI, along with our own data dependency analysis
- Developed needed base classes for use with the LLVM to Southern Islands backend
- Fixed a register release bug in the OpenCL to LLVM pipeline

### • Notable Projects

<https://github.com/nixpulvis>

- **alacrity**, a cross-platform, GPU-accelerated terminal emulator (contributor)
- **oursh**, a multi-language shell which aims to be POSIX compatible, written in Rust
- **lalrpop-lambda**, parser and reductions for the lambda calculus with a minimal webapp
- **parser-combinator**, a Racket implementation of a recursive descent parser, used by students for a JSON lab
- **galos**, an Elite: Dangerous EDDN subscriber, (PostGIS) database, CLI, and GUI
- **nrf24l01**, basic working AVR firmware for the Nordic Semiconductor's nRF24L01+ radio transceiver
- **maze\_gl**, a small maze generation and first person OpenGL game

---

```
(λ(f)(λ
(x)((λ(λ(
x)(λ(x y)y)
)(λ(x y)x))(
x(λ(x)(λ(x y)
y))(λ(x y
y)x)((λ(
x(λ(p)(
λ(s)(s
(p (λ(
x y)y))
x(λ(f x
)(f((p(
λ(x y)
y))f x
)))))(
λ(s)(s(
λ(f x)x)
(λ(f x)x)
)))((λ(x y)
x))(λ(x)(λ(
x y)y)) (λ(
x y) x))(λ(
f x)(f x))(f
((x(λ(p)(λ(s
)(s(p( λ(x y)
y))(λ( f x)(f
(p (λ( x y)y)
)f x)))))((λ(
y s)(s (λ(f x
)x)(λ( f x)x
)))((λ(
x y)x))
)(λ(n) (λ(f
x)(f (n
f x)))
)(f(((
λ(s)(s
x y y
x)(f((
y)) f
p(λ(x y
x)))))
)(λ(s)(
x)(λ(
f x)x)
)) (λ
)(λ(p) (λ(s)(
s(p(λ(
x y)y)
)(λ(f
p(λ(x
y)))))
λ(s)(
s(λ(f
x)x)(λ
))((λ(
x y)x)
))))))
```