Paul Krogmeier

303 W. Green St., Urbana, IL, 61801

📱 +1 765 404 6297 • ✉ paulmk2@illinois.edu • 🌐 paulkrog.github.io

Pursuing a PhD in computer science from the University of Illinois at Urbana-Champaign.

Education

Graduate

University of Illinois at Urbana-Champaign	**Urbana**
PhD Computer Science, GPA: 3.89	*2018–present*
Purdue University	**West Lafayette**
M.Eng. Computer Engineering, GPA: 3.99	*2016–2018*

Undergraduate

Purdue University	**West Lafayette**
B.S. Computer Engineering, GPA: 4.0	*2012–2016*
EAFIT University	**Medellín, Colombia**
Study Abroad, Compilers and Operating Systems courses	*Spring 2015*

Publications

Decidable Synthesis of Programs with Uninterpreted Functions. Paul Krogmeier, P. Madhusudan, Umang Mathur, Adithya Murali, Mahesh Viswanathan. Under submission. Preprint.

Umang Mathur, Adithya Murali, Paul Krogmeier, P. Madhusudan, and Mahesh Viswanathan. 2019. Deciding memory safety for single-pass heap-manipulating programs. Proc. ACM Program. Lang. 4, POPL, Article 35 (December 2019), 29 pages. DOI:https://doi.org/10.1145/3371103

Krogmeier, P. M. and Kidd, S. and Delaware, B. Towards Context-Aware Data Refinement. Fourth International Workshop on Coq for Programming Languages, January 2018. Paper.

Experience

Illinois Programming Languages and Formal Methods	**Urbana, IL**
Research Assistant (advised by Mahesh Viswanathan)	*Aug 2018–present*

- Exploring synthesis and verification problems for entirely uninterpreted programs over infinite data domains
- Learning algorithms for first-order logic concepts: seeking tractable classes of formulae for which online learning has a polynomial mistake bound

Purdue Programming Languages Group	**West Lafayette, IN**
Research Assistant (advised by Benjamin Delaware)	*Aug 2017–Jul 2018*

- Modeled the syntax and semantics of the Fiat specification language with a deep embedding in the Coq proof assistant.
- Developed a mechanized proof of Fiat's type safety.
- Formalized a logical relations proof strategy for validity of refinement from Fiat specifications to implementations.

Purdue University ECE Department **West Lafayette, IN**
Discrete Math Teaching Assistant–ECE 369 *Aug 2017–Dec 2017*

- Verified reference exam solutions and graded student exams.
- Organized weekly office hours and help sessions.

Purdue University – Machine Learning for SAT **West Lafayette, IN**
Research Assistant *Jan 2017–May 2017*

- Studied the source code for the MiniSat SAT solver.
- Implemented online thompson sampling algorithm to learn reward function over SAT variables.
- Tested usefulness of the extension against plain MiniSat.

Purdue University – E-Lab **West Lafayette**
Student Programmer *Sep 2016–Dec 2016*

- Programmed Torch7 CNNs to solve image classification problems.
- Experimented with RNNs to study problems in speech recognition.

APOLO Scientific Computing Center **Medellín, Colombia**
Programming Internship *May 2016–Jul 2016*

- Developed software to produce client usage reports for a Linux Rocks cluster administrative team.
- Wrote and debugged Python scripts to generate reports on cluster load and usage characteristics. This involved learning the APIs for the TORQUE and SLURM resource management tools.
- Met weekly with development team to discuss progress.

Purdue University – Open Ag Data Alliance **West Lafayette**
Embedded Systems Programmer *May 2014–Jul 2014*

- Developed C code for a wireless, embedded semi-truck weight sensing application.
- Interfaced Nordic system-on-chip to air pressure sensor over I^2C.
- Programmed communication between Android app and system-on-chip using Bluetooth Low Energy stack.

Invited Workshops and Schools

2nd VMCAI Winter School **New Orleans, LA**
Student Participant *Jan 2020*

Dagstuhl Seminar on Logic and Learning **Schloss Dagstuhl, Germany**
Invited Junior Researcher *Sep 2019*

The goal of this seminar was to explore ways of combining logical knowledge with learning systems like neural networks.

SRI Formal Methods Summer School **Atherton, California**
Student Participant *May 2019*

- Experimented with EasyCrypt for Coq proofs security for cryptographic protocols.
- Experimented with the Viper verification language for proving properties of heap-manipulating programs.

Oregon Programming Languages Summer School **Eugene, Oregon**
Student Participant *Jun 2017*

- Experimented with dependently-typed Idris and with programming language semantics modelling in PLT Redex.
- Attended research lectures from experts in programming languages and formal methods.

Coursework

Graduate
CS 598 – Algorithmic Game Theory
MA 570 – Mathematical Logic
MA 511 – Linear Algebra with Applications
CS 477 – Formal Software Development
CE 642 – Information Theory and Source Coding
CE 573 – Compilers and Translator Systems
CE 608 – Computational Models and Methods
CE 600 – Probabilities and Random Processes
CS 565 – Programming Languages
CS 590 – Artificial Intelligence and Causal Inference
CS 584 – Theory of Computation and Complexity
CS 573 – Data Mining

Undergraduate
CE 368 – Algorithms and Data Structures
CE 369 – Discrete Math
CE 364 – Python and Bash Scripting Lab
CE 337 – ASIC Design Laboratory
CE 437 – Computer Architecture
CE 477 – Digital Systems Senior Design

Skills

- **Programming Languages and Tools:**
  High proficiency: Ocaml, C/C++, Python
  Medium proficiency: Haskell, Prolog, Java, Matlab, Verilog, Emacs
  Familiarity: Coq, Racket, Rosette, Idris, Lisp, x86, MIPS, Jekyll/HTML/CSS
- **Natural Languages:** Fluent in Spanish, German, and English (native)

Awards, Honors, Grants

- **ACM SIGPLAN PAC Travel Grant:** January 2020
- **UIUC Wing Kai Cheng Fellowship:** August 2018
- **Purdue Ross Fellowship:** May 2016
- **Phi Beta Kappa:** May 2016
- **Graduated “with highest distinction” (top in class, Purdue ECE):** May 2016
- **100K Strong in the Americas Scholarship:** August 2014