Karl B. Naden

5510 Wilkins Ave.
Pittsburgh, PA 15217

kbn@cs.cmu.edu
(206) 375-1266

Education:

Carnegie Mellon University, Pittsburgh, PA
PhD Candidate in **Computer Science,** enrolled August 2009
- Advised by Dr. Jonathan Aldrich

Williams College, Williamstown, MA
Bachelor of Arts in **Math** and **History**, Magna Cum Laude, June 2006
- GPA – 3.83, elected to **Phi Beta Kappa,** Deans List all 7 semesters in residence
- Received **Highest Honors** and the **Bostert prize** for best American history thesis

Current Research:

Plaid Programming Language - a typestate and permission-based language
- Contributed to a **compiler** for the dynamic subset of Plaid
- Designed the **dynamic semantics** of the state change operation to work in a modular way
- Implemented a basic **typechecker** for a subset of Plaid in the dynamic subset of the Plaid language
- Currently formalizing the precise **static semantics** of the core language

Published Work:

- ***Modular Composition and State Update in Plaid***. Jonathan Aldrich, Karl Naden, and Éric Tanter. In Proceedings of the Workshop on Mechanisms for Specialization, Generalization, and Inheritance (MASPEGHI), 2010.
- ***Permission-Based Programming Languages***. Jonathan Aldrich, et. al. To appear in the New Ideas and Emerging Results Track at the International Conference on Software Engineering (NIER-ICSE), 2011.

Related Coursework:

Type Systems for Programming Languages - taught by Bob Harper at CMU, Fall 2009
- Thorough introduction to the logical underpinnings of types

Data, Code, Decisions - taught by Andre Platzer at CMU, Fall 2009
- Learned the **theory** and **practice** of proving programs correct using the KeY theorem prover
- Class Project: analyzed methods for **complete-program verification** using JML and KeY

Program Analysis - taught by Jonathan Aldrich at CMU, Spring 2010
- Explored cutting edge approaches to program **verification**
- Class Project: developed a Boogie-style **static checker** for a C-like language

Teaching:

TA for Principles of Imperative Computation - taught by Frank Pfenning at CMU, Spring 2011
- Course focuses on the power of **specifications** (pre/post-conditions, loop-invariants) forunderstanding, correctness, and testing imperative programs
- Investigating how to adapt existing static analysis tools to provide clear, useful, and unmbiguous feedback to beginning programmers

Other Experience:

Epic Systems Corporation, Madison, WI
EDI Interface Analyst – September 2006 to July 2009
- **Owner** of interfaces to communicate with external medication auto-dispense systems
- **Designed** and **Programmed** major enhancement to interface codebase
- **Managed** Epic interface account for a major healthcare provider

Activities:

Singing
- Member of the **Mendelssohn Choir of Pittsburgh**, performing with the **Pittsburgh Symphony**

Personal: avid player of strategy board games, enthusiastic runner, cook, and dancer, reader of The Economist