

# Leaf Petersen

# Curriculum Vitae

---

<http://www.cs.cmu.edu/~leaf>  
leaf.petersen@gmail.com

## Research Interests

---

Type theory. Advanced programming languages - theory, design, and efficient implementation. Compiling with types. Language based security and certified code. Module systems. Data representation languages. Object oriented languages.

## Employment

---

I work for Intel Corporation as a Software Researcher in the Programming Systems Laboratory. My current research is focused on making it easier to write scalable code for future generations of many-core processors.

## Education

---

**1996 - present:** Carnegie Mellon University. Pittsburgh, PA

- Doctoral candidate supervised by Dr. Robert Harper and Dr. Karl Crary.
- Degree expected spring 2004.

**1992 - 1996:** Williams College. Williamstown, MA.

- BA in Computer Science.
- Summa Cum Laude with Highest Honors.

## Research history

---

**2001-present:** Thesis work, Carnegie Mellon University.

As part of my doctoral thesis, I am currently working on a certifying back end for the TILT compiler which will compile Standard ML into a form of typed assembly language. TILT is an optimizing compiler that uses type information for optimization purposes, both statically and at runtime. Typed assembly code is ordinary machine code with additional

annotations that allow the code to be statically checked for type and memory safety. My work is to extend the TILT compilation model to push types all the way down to the assembly level without sacrificing the benefits of the TILT optimizations.

**1996-present:** TILT compiler, Carnegie Mellon University.

Worked on the TILT/ML compiler - a type based compiler for the full SML '98 language. TILT preserves types throughout the compilation process, which allows for various data representation optimizations and for doing almost tag-free garbage collection. TILT is aimed at showing that advanced languages can be compiled efficiently using type information.

**Summer 1998:** Microsoft Research, Cambridge UK.

Worked with Dr. Luca Cardelli on an interpreter for the *Ambient Calculus* - a calculus for describing mobility and mobile computations. The interpreter, written in Java, was extended to allow for processes running in remote locations. Process mobility was implemented using XML as an intermediate representation form for transmission.

**1994-1996** Williams College, Williamstown, MA.

Undergraduate research with Dr. Kim Bruce. Design and implementation of strongly typed object-oriented languages. Designed LOOM - an object-oriented programming language using "matching", and "match" based parametric polymorphism. Designed a module system for LOOM, as described in my Honors Thesis.

## Publications

---

Vijay S. Menon, Neal Glew, Brian R. Murphy Andrew McCreight, Tatiana Shpeisman, Ali-Reza Adl-Tabatabai and Leaf Petersen. *A Verifiable SSA Program Representation for Aggressive Compiler Optimization*. To appear in POPL 2006: ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages.

Leaf Petersen. *Certifying Compilation for Standard ML in a Type Analysis Framework*. PhD thesis, Carnegie Mellon University, 2005. Published as CMU Technical Report CMU-CS-05-135.

Leaf Petersen, Robert Harper, Karl Crary and Frank Pfenning. *A Type Theory for Memory Allocation and Data Layout*. POPL 2003: ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages.

Joseph C. Vanderwaart, Derek R. Dreyer, Leaf Petersen, Karl Crary, and Robert Harper. *Typed Compilation of Recursive Datatypes*. TLDI 2003: ACM SIGPLAN International Workshop on Types in Language Design and Implementation.

Kim B. Bruce, Adrian Fiech, and Leaf Petersen. *Subtyping is not a good 'Match' for object-oriented languages*. ECOOP 1997: European Conference for Object-Oriented Programming.

Leaf Petersen, Perry Cheng, Robert Harper, and Chris Stone. *Implementing the TILT Internal Language*. CMU Technical Report #CMU-CS-00-180, December 2000.

Karl Crary, Robert Harper, Perry Cheng, Leaf Petersen, and Chris Stone. *Transparent and Opaque Interpretations of Datatypes*. CMU Technical Report #CMU-CS-98-177, November 1998.

Leaf Petersen. *Certifying Compilation for Standard ML in a Type Analysis Framework*. Doctoral thesis proposal, Carnegie Mellon University, 2001.

Leaf Petersen. *A Module System for LOOM*. Undergraduate thesis, Williams College, 1996.

## Teaching

---

### Spring 1999: Carnegie Mellon University

Teaching assistant for the undergraduate course *Great Ideas in Theoretical Computer Science*, taught by Professor Steven Rudich. This course is the advanced discrete math course for first year CS students at CMU. I prepared and graded assignments, held office hours and taught recitation.

### Fall 1997: Carnegie Mellon University

Teaching assistant for the graduate level *Programming Languages Core Course*, assisting Professor Robert Harper. This course was a one semester required course for School of Computer Science graduate students. I wrote and graded assignments, helped write and grade the final exam, held office hours, and gave a lecture on programming in ML.

### Spring 1997: Carnegie Mellon University

Teaching assistant for the undergraduate course *Fundamental Structures of Computer Science II*. This was a required course, intended to teach advanced programming using the programming language ML, as well as introducing formal reasoning about programs and some theory of computation. I wrote and graded assignments, taught weekly recitation sections, and helped write and grade exams.

**Spring 1996:** Williams College

Teaching assistant for CS109: *The Art and Science of Computer Graphics*. Aimed specifically at non-computer majors, this course used a powerful rendering package implemented in Scheme to allow students to access much of the power of computer graphics without requiring any prior programming experience. I graded assignments, helped with labs, and held office hours.

**Fall 1994,1995:** Williams College

Teaching assistant for CS137: *Data Structures*. This course was an advanced version of the required second level programming course, taught in ML and Object Pascal (1/2 semester each). Formal reasoning about programs was also introduced. I was responsible for grading assignments, teaching lab sessions, and for holding office hours.

**Spring 1994:** Williams College

Teaching assistant for CS134: *Introduction to Computer Science*. This course was an introductory course for CS majors, taught in Pascal. I graded assignments, helped with labs, and held office hours.

**Awards**

---

- NSF Fellowship Finalist, 1996.
- Summa Cum Laude from Williams College, 1996.
- Sigma Xi, 1996.
- Elected Phi Beta Kappa, 1995.
- ARCS Scholar, 1994-95

**Other**

---

Moderator for the newsgroup comp.lang.ml - December 1998 to present.

**Citizenship**

---

United States Citizen

**Personal Interests**

---

Squash, climbing, backpacking, guitar, cooking.

**References**

---

Upon Request