Leaf Petersen

Curriculum Vitae

721 SW 199th PL
Normandy Park, WA 98116
United States of America
+1 (206) 419 6834
leaf.petersen@gmail.com
leafpetersen.com

Skills Summary

I have experience in the areas of type systems, programming language design and implementation, optimizing compilers, parallel processing, functional programming, performance optimization, and programming language runtime implementation. I have been a founding member of two small teams which designed and built highly optimizing compilers and runtimes for high-level programming languages. I have publications and patents in the areas of type theory, advanced compiler optimization, transactional memory, parallel programming, and runtime implementation. I am comfortable working at all levels of the software stack, from high-level languages to machine instructions.

Education

1996–2005 **Ph.D. in Computer Science**, *Carnegie Mellon University*, Pittsburgh, PA.

1992–1996 **B.A. in Computer Science**, *Williams College*, Williamstown, MA.
Summa Cum Laude, Phi Beta Kappa

Doctoral thesis

title *Certifying Compilation for Standard ML in a Type Analysis Framework*

supervisors Dr. Robert Harper and Dr. Karl Crary

description Built an optimizing, certifying compiler for the full Standard ML language targeting Typed Assembly Language, using type information to do runtime type dispatch optimizations.

Work and Research Experience

2004–present **Research Scientist**, *Programming Systems Lab, Intel Corporation*, Santa Clara, CA.
I am part of a small team that designed and built an aggressively optimizing compiler for an experimental high-level parallel language, subsequently retargeted to compile the Haskell programming language. Among other optimization passes, I implemented an optimizing closure converter, a global flow-analysis based optimization suite, a general simplifier, a data-flow analysis framework and several inliners. I also designed and supervised the implementation of a SIMD vectorization pass, a loop-invariant code motion pass, and a general inlining framework. I implemented significant parts of the language runtime, and helped drive the design of numerous garbage collection and runtime improvements. The compiler targeted standard Intel x86 many-core processors as well as Xeon Phi co-processor boards. We demonstrated substantial sequential and parallel performance improvement over the industry standard compiler (GHC) on numeric benchmarks, and were in some cases able to match highly tuned parallel C implementations. Prior projects at Intel included writing a graph-coloring register allocator for a Java just-in-time compiler, and implementing lightweight synchronization and scheduling mechanisms in a prototype many-core runtime.

Summer 1998 **Intern**, *Microsoft Research*, Cambridge, UK, Supervised by Dr. Luca Cardelli.
Built a distributed interpreter for an experimental process calculus (the *Ambient Calculus*), using Java and XML.

1996–2004 **Graduate Research Assistant**, *Carnegie Mellon University*, Pittsburgh, PA, Supervised by Dr. Robert Harper and Dr. Karl Crary.
I and two other graduate students designed and built the TILT/ML compiler—an optimizing compiler for the full Standard ML programming language. TILT used runtime type information to allow for non-uniform data representations, and for doing tag-free garbage collection. For my doctoral thesis, I designed, built, and measured a certifying backend for the TILT compiler, allowing the optimized code to be generated as statically checkable x86 Typed Assembly Language while preserving the benefits of the type-based optimizations. I proved soundness results for each of the main translation phases, including a general account of register allocation soundness.

1994–1996 **Undergraduate Research Assistant**, *Williams College*, Williamstown, MA, Supervised by Dr. Kim Bruce.
Worked with Dr. Bruce on the design and implementation of a strongly-typed object oriented language (LOOM) based around match-bounded polymorphism. Designed and implemented a module system for LOOM as an undergraduate honors thesis project.

Languages

English **Native**

Spanish **Limited working proficiency**

Programming Languages

I have done most of my recent programming in C and Standard ML (with a bit of Perl on the side). I have in the past worked in C++, Java, Perl, Haskell, and OCaml.

Personal Details

Citizenship United States of America

Interests Squash, Alpine Climbing, Cooking, Gardening, Guitar

Patents

Software assisted nested hardware transactions
US7730286 B2, CN101317160B, DE602006014596D1, EP1966697B1, WO2007078891A1
Leaf Petersen, Bratin Saha, Ali-Reza Adl-Tabatabai

Future scheduling by direct representation of possible dependencies
US8225326 B2
Leaf Petersen, Anwar Ghuloum, Mohan Rajagopalan

Safe code-motion of dangerous instructions during compiler optimization
US7810086 B2
Brian R. Murphy, Vijay S. Menon, Tatiana Shpeisman, Ali-Reza Adl-Tabatabai, Leaf Petersen

Service Activities

Program Committee (PC) member

- 2014 Symposium on Practical Aspects of Declarative Languages (**PADL 2014**)
- 2013 Workshop on Functional High-Performance Computing (**FHPC 2013**)
- 2012 Workshop on Declarative Aspects of Multicore Programming (**DAMP 2012**)
- 2011 Workshop on Declarative Aspects of Multicore Programming (**DAMP 2011**)
- 2010 Workshop on Declarative Aspects of Multicore Programming (**DAMP 2010**)
- 2009 ACM SIGPLAN Symposium on Principles of Programming Languages (**POPL 2009**)

Reviewer and panelist for a 2013 National Science Foundation grant program in the area of parallel processing.

Participant in 2013 National Science Foundation Workshop on High-Level Programming Models for Parallelism (invitation-only).

External reviewer for numerous conferences and workshops including PACT, PLDI, POPL, PADL, ECOOP, ICFP, IFL, and TLDI.

General Chair

- 2010 Workshop on Declarative Aspects of Multicore Programming (**DAMP 2010**)
- 2009 Workshop on Declarative Aspects of Multicore Programming (**DAMP 2009**)
- 2008 Workshop on Declarative Aspects of Multicore Programming (**DAMP 2008**)

Publications

Review copies available at `http://www.leafpetersen.com`

Conference and Workshop Publications

ICFP 2013 **Automatic SIMD Vectorization for Haskell**
In 2013 ACM SIGPLAN International Conference on Functional Programming
Leaf Petersen, Dominic Orchard and Neal Glew

DTP 2013 **A Multivalued Language with a Dependent Type System**
In 2013 ACM SIGPLAN Workshop on Dependently Typed Programming
Neal Glew, Tim Sweeney and Leaf Petersen

HS 2013 **The Intel Labs Haskell Research Compiler**
In 2013 ACM SIGPLAN Haskell Symposium
Hai Liu, Neal Glew, Leaf Petersen and Todd Anderson

IFL 2013 **Measuring the Haskell Gap**
In 2013 International Symposium on Implementation and Application of Functional Languages
Leaf Petersen, Todd Anderson, Hai Liu and Neal Glew

CC 2012 **GC-Safe Interprocedural Unboxing**
In 2012 International Conference on Compiler Construction
Leaf Petersen and Neal Glew

CUFP 2010 **Functional language compiler experiences at Intel**
In 2010 ACM SIGPLAN Commercial Users of Functional Programming
Leaf Petersen and Neal Glew

EUROSYS 2007 **Enabling scalability and performance in a large scale CMP environment**
In 2007 ACM SIGOPS/EuroSys European Conference on Computer Systems
Bratin Saha, Ali-Reza Adl-Tabatabai, Anwar M. Ghuloum, Mohan Rajagopalan, Richard L. Hudson, Leaf Petersen, Vijay Menon, Brian R. Murphy, Tatiana Shpeisman, Eric Sprangle, Anwar Rohillah, Doug Carmean and Jesse Fang

LCPC 2007 **Pillar: A Parallel Implementation Language**
In 2007 Workshop on Languages and Compilers for Parallel Computing
Todd A. Anderson, Neal Glew, Peng Guo, Brian T. Lewis, Wei Liu, Zhanglin Liu, Leaf Petersen, Mohan Rajagopalan, James M. Stichnoth, Gansha Wu and Dan Zhang

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

TLDI 2005 **Strict Bidirectional Type Checking**
In 2005 ACM SIGPLAN International Workshop on Types in Language Design and Implementation
Adam Chlipala, Leaf Petersen, and Robert Harper

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

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

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

Ph.D. Thesis

Certifying Compilation for Standard ML in a Type Analysis Framework
PhD thesis, Carnegie Mellon University, 2005
Leaf Petersen

Other Publications

Type-Preserving Flow Analysis and Interprocedural Unboxing
In 2012 Intel Technical Report
Neal Glew and Leaf Petersen

Implementing the TILT Internal Language
In 2000 Carnegie Mellon School of Computer Science Technical Report
Leaf Petersen, Perry Cheng, Robert Harper, and Chris Stone

Transparent and Opaque Interpretations of Datatypes
In 1998 Carnegie Mellon School of Computer Science Technical Report
Karl Crary, Robert Harper, Perry Cheng, Leaf Petersen, and Chris Stone

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