Carnegie Mellon University
Computer Science Department

PRINCIPLES OF PROGRAMMING

Umut Acar

Stephanie Balzer

Guy Blelloch

Stephen Brookes

Karl Crary

Matt Fredrikson

Robert Harper

Marijn Heule

Jan Hoffmann

Dilsun Kaynar

Ruben Martins

Frank Pfenning

André Platzer

Affiliated Faculty

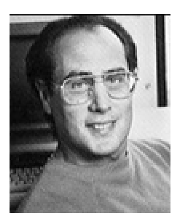
Jonathan Aldrich

Jeremy Avigad

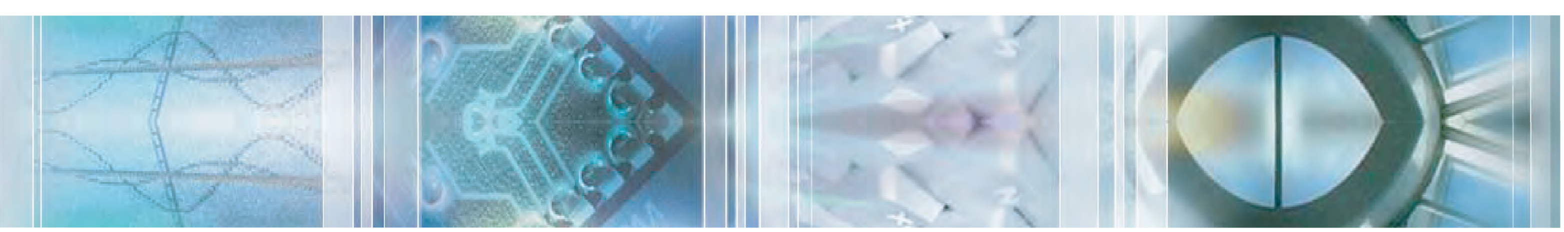
Steve Awodey

Claire Le Goues

Wilfried Sieg



Richard Statman



Carnegie Mellon University
Computer Science Department

PRINCIPLES OF PROGRAMMING

Umut Acar, Associate Professor (CS)
umut@cs.cmu.edu
http://www.cs.cmu.edu/~umut
http://www.csd.cs.cmu.edu/people/faculty/umut-acar

Scalable Parallelism, Dynamic Parallelism, Self-adjusting Computation

Broadly construed, I research problems of scalability. Current foci include the design and development of abstractions, algorithms, languages, and systems for scalable parallel, dynamic, and interactive computation.

Jonathan Aldrich, Professor (ISR)
aldrich@cs.cmu.edu
http://www.cs.cmu.edu/~aldrich/
http://www.csd.cs.cmu.edu/people/faculty/jonathan-aldrich

Programming Languages and Software Engineering

My research improves the quality of software and the productivity of engineers by expressing and enforcing structural and behavioral aspects of software design within source code, typically through language design and type systems. I have contributed to object-oriented typestate verification, techniques for modular reasoning about state, and new object-oriented language models.

Stephanie Balzer, Assistant Research Faculty (CS)
balzers@cs.cmu.edu
http://www.cs.cmu.edu/~balzers/
https://csd.cmu.edu/people/research-faculty/stephanie-balzer

Programming Languages, Type Systems, Program Verification, Security

My current research focuses on the development of type systems and logics for verifying properties of interest of concurrent programs.

Guy Blelloch, Professor (CS)
Associate Dean of Undergraduate Programs
guyb@cs.cmu.edu
http://www.cs.cmu.edu/~guyb/
http://www.csd.cs.cmu.edu/people/faculty/guy-blelloch

Parallelism, Data Structures and Algorithms, Cache Efficient Parallel Algorithms

My research has largely been in the interaction of Algorithms and Programming Languages, much of it in the area of parallel computing.

Carnegie Mellon University
Computer Science Department

PRINCIPLES OF PROGRAMMING

Stephen Brookes, Professor (CS)
brookes@cs.cmu.edu
http://www.cs.cmu.edu/~brookes/
http://www.csd.cs.cmu.edu/people/faculty/stephen-brookes

Semantic models for programming languages and logics for reasoning about the behavior of concurrent programs

I have been involved in the development of Concurrent Separation Logic, in collaboration with Peter O'Hearn and the late John Reynolds. I am currently investigating further extensions to this logic, and working out the semantic underpinnings needed to validate logics that combine concurrency with procedures and communication.

Karl Crary, Associate Professor (CS)
crary@cs.cmu.edu
http://www.cs.cmu.edu/~crary/
http://www.csd.cs.cmu.edu/people/faculty/karl-crary

Programming languages

My research interests are in applying programming language technology to improve the development, maintenance, and performance of software systems. I am particularly interested in the application of type theory to systems programming, in mechanization of the metatheory of programming languages, in type-oriented compilation strategies, in type-based certification of machine code, and in the design of practical, high- or low-level programming languages.

Matthew Fredrikson, Assistant Professor (CS)
mfredrik@cs.cmu.edu
http://www.cs.cmu.edu/~mfredrik/
http://www.csd.cs.cmu.edu/people/faculty/matthew-fredrikson

Security & Privacy

My research focuses on security and privacy issues that lead to failures in real systems. Some of the key outstanding challenges in this area lie in figuring out why promising theoretical approaches oftentimes do not translate into effective defenses. Much of my work is concerned with developing formal analysis techniques that provide insight into the problems that might exist in a system, building countermeasures that give provable guarantees, and measuring the effectiveness of these solutions in real settings.

Robert Harper, Professor (CS)
rwh@cs.cmu.edu
http://www.cs.cmu.edu/~rwh
http://www.csd.cs.cmu.edu/people/faculty/robert-harper

Mathematical principles of programming

My current research projects are in computational higher type theory, and in the development of cost semantics for higher-level programming languages. More generally I am interested in constructive mathematics, programming language semantics, program verification, and mechanized proof. All of these subjects are unified in the setting of computational type theory.

Carnegie Mellon University
Computer Science Department

PRINCIPLES OF PROGRAMMING

Marijn Heule, Associate Professor (CS)
marijn@cmu.edu
https://www.cs.cmu.edu/~mheule/
http://www.csd.cs.cmu.edu/people/faculty/marijn-heule

Automated Reasoning, Formal Verification, Parallel Computing

My research contributions to automated reasoning have enabled me and others to solve hard problems in formal verification and mathematics.I have developed award-winning automated reasoning tools and my preprocessing and proof producing techniques are used in many state-of-the-art solvers.

Jan Hoffmann, Assistant Professor (CS)
janh@cs.cmu.edu
http://www.cs.cmu.edu/~janh
http://www.csd.cs.cmu.edu/people/faculty/jan-hoffmann

Programming languages and formal methods

My research areas are programming languages and formal methods. I am specifically interested in quantitative verification, type systems, static resource analysis of programs, proof assistants, and decision procedures.

Dilsun Kaynar, Assistant Teaching Professor (CS)
dilsun@cs.cmu.edu
http://www.cs.cmu.edu/~dilsun

Foundations of distributed computing, formal methods, security

I am interested in developing modeling and verification methods for a wide range of distributed systems including those that exhibit timing-dependent behavior and possibly interact with the physical world. I have also had continued interest in security. My current topic of interest in this area is accountability.

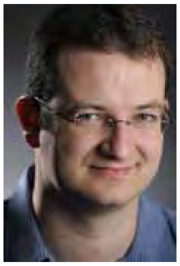
Ruben Martins, System Scientist (CS)
rubenm@cs.cmu.edu
http://www.cs.cmu.edu/~rubenm
https://sat-group.github.io/ruben/

Constraint Programming, Boolean Satisfiability and Optimization, Software Verification, Program Synthesis

The goal of my research is to improve constraint solvers and broaden their applicability in program analysis, synthesis, and security. I have developed several award winning MaxSAT solvers that are widely used in software package upgradeability, computational biology, and course timetabling. My most recent work focuses on program synthesis for data-science-related tasks. Specifically, I am interested in automating a variety of cumbersome data preparation tasks and making the life of data scientists simpler.

Carnegie Mellon University
Computer Science Department

PRINCIPLES OF PROGRAMMING



André Platzer, Associate Professor (CS)
aplatzer@cs.cmu.edu
http://www.cs.cmu.edu/~aplatzer
http://www.csd.cs.cmu.edu/people/faculty/andre-platzer

Logic in Computer Science, Cyber-Physical Systems, Programming Languages, Formal Methods

My research develops the logical foundations of cyber-physical systems to charachterize their fundamental principles and to show how we can design computers that are guaranteed to interact correctly with the physical world. The solution to this challenge is the key to enabling computer assistance that we can bet our lives on. I pursue this challenge with the principled design of programming languages with logics that can provide proofs as correctness guarantees.

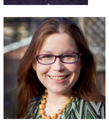
Frank Pfenning, Professor (CS)
fp@cs.cmu.edu
http://www.cs.cmu.edu/~fp
http://www.csd.cs.cmu.edu/people/faculty/frank-pfenning

Formal Reasoning about Languages for Distributed Computation

At the heart of my research lies the desire to understand the principles of programming languages. Programming languages are the key to the programming process and therefore of fundamental importance to computer science. Well-designed programming languages allow fast program development, ease software maintenance, and increase confidence in the correctness of implementations.

Jeremy Avigad, Professor (Philosophy)
avigad@andrew.cmu.edu
http://www.andrew.cmu.edu/user/avigad/

Steve Awodey, Professor (Philosophy)
awodey@andrew.cmu.edu
http://www.andrew.cmu.edu/user/awodey/



Claire Le Goues, Assistant Professor (ISR)
clegoues@cs.cmu.edu
http://www.cs.cmu.edu/~clegoues/

Wilfried Sieg, Patrick Suppes Professor (Philosophy)
sieg@cmu.edu
http://www.cmu.edu/dietrich/philosophy/people/faculty/core-faculty/sieg.html

Richard Statman, Professor (Mathematics)
statman@cs.cmu.edu
http://www.math.cmu.edu/math/faculty/Statman