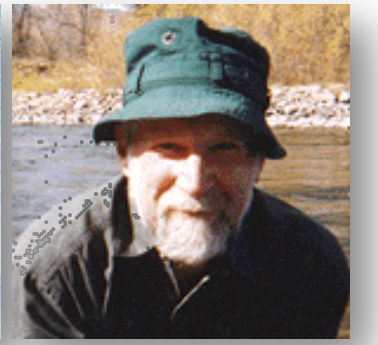
Classic, Hot, and Unrecognized Problems in Programming Languages

Kathryn S McKinley
Google

About me





Mentors

Research



Congressional Testimony

life

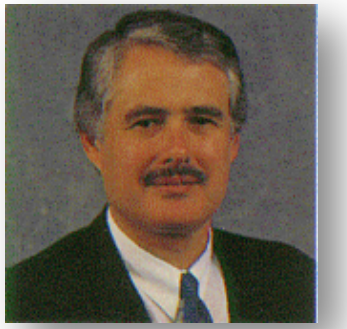
Family

Researchers solve problems & ask questions

My undergraduate & graduate research

1983 What is the bottleneck in a local area network?

1984 TED A Text EDitor to help Fortran programmers produce correct programs



1992 Interactive and automatic parallelization

Performance

Programmer Tools

Optimization

Programming language researchers

Help people make computers do stuff

Classic topics I

Design programming languages

- Correctness, expressiveness, efficiency

Specify semantics for languages & programs

Prove properties about languages & programs

Programming language implementation

Programming languages timeline

https://www.levenez.com/lang/lang.pdf

Classic topics II

Compilers & interpreters

- map high level to low level languages
- correct, secure, etc.

Finding bugs in programs

Performance of programs

- Profiling, benchmarks, measurements

Classic topics III

Classic topics III

Tools for programmers, users, & PL researchers

- writing programs
- writing programming languages
- proving things about programs
- compiling & optimizing programs
- debugging programs
- profiling programs

Techniques PL researchers use

Language grammars & tools
Lambda calculus
Type theory
Data flow analysis
Static analysis
Dynamic analysis
Simulation

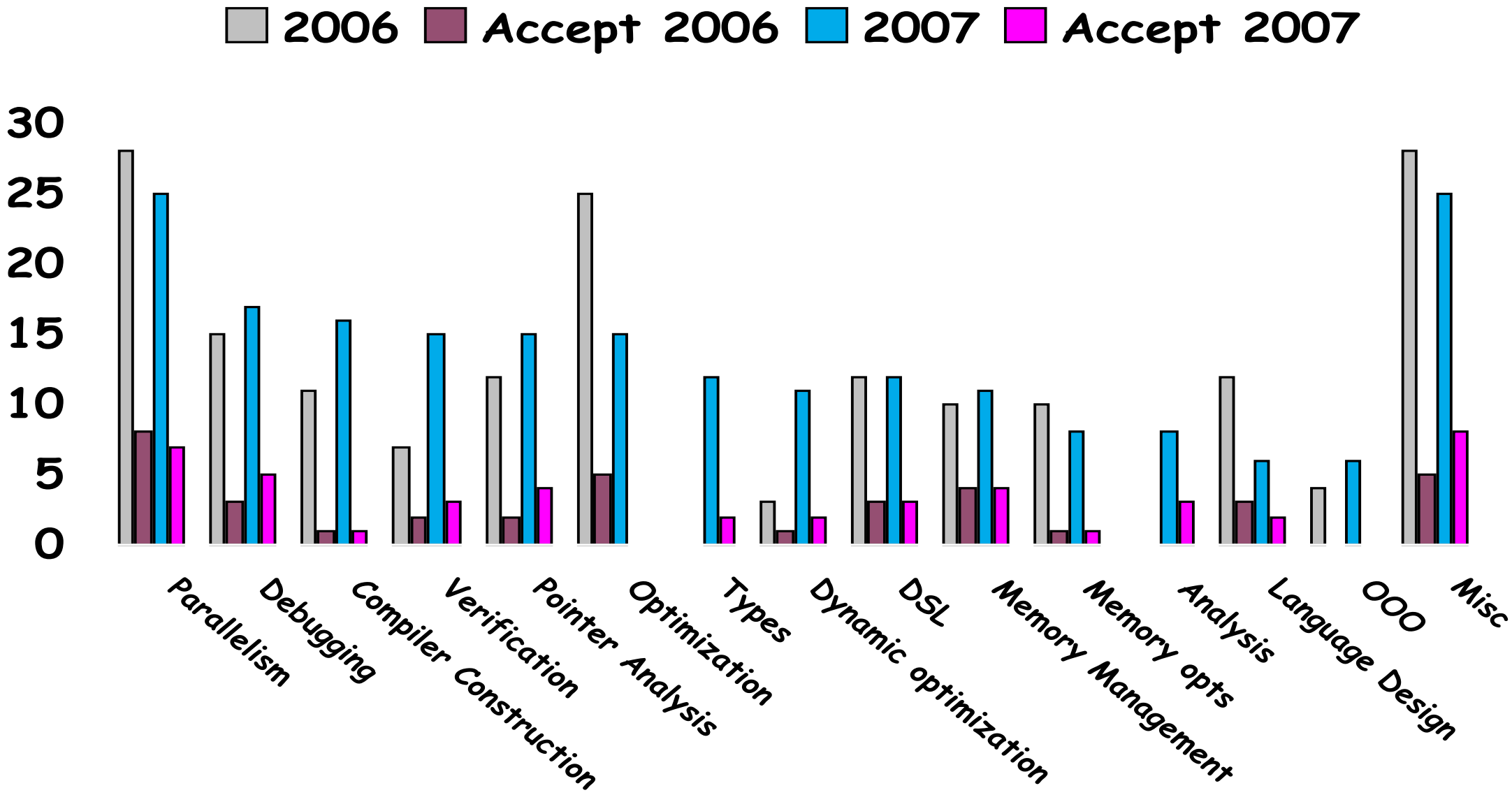
Proof languages
Model checking
Proof assistants
SMT solvers
Synthesis

Applications & hardware change

but… they always need a

programming language and implementation

PLDI 06 & 07 Topics Submitted/Accepted



Hot Topics in CS become PL topics

Big data & streaming

Approximation

Machine learning

Probabilistic programming

Human computation

Spreadsheets as programs

Hardware

- Multicore
- GPUs, accelerators
- Non volatile memory

Exploring new topics

Enhancing Server Availability and Security Through Failure-Oblivious Computing

OSDI 2004

Martin Rinard, Cristian Cadar, Daniel Dumi
Tudor Leu, and William S. Bee
Computer Science and Artificial Intelligen
Massachusetts Institute of Technol
Cambridge, MA 02139

Abstract

We present a new technique, *failure-oblivious comput-ing*, that enables servers to execute through memory er-rors without memory corruption. Our safe compiler for

1 Introduction

Memory errors such as out
invalid pointer accesses are a
failures. Safe languages such

ses and
of program
and Java use dy-

Rejected for 3 years from PL venues

First paper

failure oblivious ➔ approximate computing

Sample reviews

… damage the field ….

This is utter nonsense.

It seems heavily based on Uncertain<T>

Exploring new topics

Uncertain<*T*>: A First-Order Type for Uncertain D

ASPLOS 2014

James Bornholt Todd Mytkowicz

Expressing and Verifying Probabilistic Ass

PLDI 2014

Adrian Sampson Pavel Panchekha Todd Mytkowicz Kathryn S. McKinley Dan G

University of Washington Microsoft Research Univ

Rejected next papers for 3 years from PL & ML venues

Two papers

probabilistic programming for the rest of us

Researcher characteristics

thick skin

growth mind set

persistence

PL is fundamental to CS