Alexander G. Bakst

Contact Information

Computer Science and Engineering
University of California, San Diego
9500 Gilman Drive, Mail Code 0404
La Jolla, CA 92093-0404

(858) 354-4091
abakst@cs.ucsd.edu
http://cse.ucsd.edu/~abakst

Research Interests

Programming language design, type systems, program analysis, program verification

Education

University of California, San Diego, La Jolla, CA

Ph.D., Computer Science, fifth year

Massachusetts Institute of Technology, Cambridge, Massachusetts

M.Eng., Computer Science, May, 2009
Thesis: "Enabling Diagnostics in User Interfaces for CAD Applications"
Web: http://hdl.handle.net/1721.1/53142

S.B., Computer Science, May, 2008

Peer-reviewed Publications

Predicate Abstraction for Linked Data Structures
VMCAI 2016: International Conference on Verification, Model Checking, and Abstract Interpretation
Alexander Bakst, and Ranjit Jhala

Bounded Refinement Types
ICFP 2015: ACM SIGPLAN International Conference on Functional Programming
Niki Vazou, Alexander Bakst, and Ranjit Jhala

Deterministic Parallelism via Liquid Effects
PLDI 2012: ACM SIGPLAN Conference on Programming Language Design and Implementation
Ming Kawaguchi, Patrick Rondon, Alexander Bakst, and Ranjit Jhala

CSolve: Verifying C Programs with Liquid Types (tool description)
CAV 2012: Computer Aided Verification
Patrick Rondon, Alexander Bakst, Ming Kawaguchi, and Ranjit Jhala

Academic Experience

University of California, San Diego, La Jolla, CA
Graduate Student **September 2011 - Present**
I am current working on developing refinement type systems for stateful programming. I am interested in expressive refinement types that can inferred, and thus automatically prove the absence of implementation errors. In particular I am interested in both low-level and high-level programs that are effectful.

Microsoft Research, Redmond, WA
Research Intern **June 2012 - September 2012**
I worked with Chris Hawblitzel at Microsoft Research on *V*erve, a computer-verified memory-safe operating system. We used several language-based techniques in order to specify and verify the memory-safety of Verve on multicore processors.

Massachusetts Institute of Technology, Cambridge, Massachusetts

Graduate Student **September 2008 - June 2009**
Master of Engineering research done as an intern at Autodesk. I augmented geometric solvers in Autodesk Civil 3D in order to enable the development of user interfaces that would be able to guide users through the design process. I developed a method to allow the solvers to explore the solution space of the problem. The user is then presented with various corrections to infeasible designs, or valid ranges for unspecified parameters.

Undergraduate Research - MIT Media Lab **February - May, 2007**
Worked in the Cognitive Machines Group in the MIT Media Lab. I was responsible for contributing to a puzzle game in Second Life to study human collaboration.

MASLAB Robotics Competition **January 2007**
As part of a four person team, developed a robot to navigate an unknown maze, find scattered red balls, and deposit them into yellow goals. Our team won the award for software engineering.

Teaching Experience

UCSD - CSE 130 Programming Languages **January - March, 2013 & 2014**
Teaching assistant duties included leading a discussion section, holding office hours, and grading student work.

MIT - 6.005 Elements of Software Construction **February - June, 2009**
Teaching assistant duties included leading a discussion section, holding office hours, and grading student work.

Professional Experience

Oracle Corporation, Nashua, New Hampshire USA

Software Developer **June 2009 - May 2011**
Developed a cluster filesystem and a dynamic volume manager (Oracle ACFS and Oracle ADVM). I worked on adding filesystem replication suport to the Solaris and AIX ports of Oracle ACFS. I contributed to both products on Linux, IBM AIX, Solaris, and Windows 2003 and 2008.

Autodesk, Manchester, New Hampshire USA

Master of Engineering Intern **June - December 2008**
I explored different ways of augmenting geometric solvers to enable user interfaces that could guide the user through a design. In particular, these user interfaces would attempt to describe the space of input parameters that result in valid geometry.