

Steven E. Ganz

<http://steven.ganz.googlepages.com>
steven.ganz@genetius.com

514 Live Oak Lane
Emerald Hills, CA 94062
(408) 591-9872

Research Interests

Fundamentally, I am interested in **semantics**: ways of expressing meaning in language and ways of drawing meaning out of language. With **programming languages**, we have the advantage of being able to design the rules ourselves. Various **paradigms** (functional, logical, object-oriented, multithreaded) can be applied, together or separately, in expressing problems and their solutions. **Transformations** between paradigms can be used both to define and to implement the intended semantics. Indeed, abstractions appropriate for defining program transformations comprise important language features. I am interested in **modular** representations of language (by feature) and in **encapsulation** constructs that aid in the static analysis of programs.

Education

Indiana University

Bloomington, IN

9/95 – 12/06

School of Informatics and Computing

Ph. D., Computer Science

Advisor: Daniel P. Friedman

Committee: Christopher T. Haynes, David C. McCarty, Paul W. Purdom

College of Arts and Sciences

M. S., Computer Science, 9/97

University of Pennsylvania

Philadelphia, PA

9/84 – 5/89

Magna Cum Laude

Dual-degree Management & Technology Program

School of Engineering and Applied Science

B. S. Engineering, Computer Science

Wharton School of Business and Finance

B. S. Economics, Operations and Information Management/Accounting

Work Experience

Independent Consultant

9/09 – pres

Assignments have included:

- For a prominent automobile manufacturer's research unit, I conducted a survey of relevant projects at various labs in North America.
- For a SF Bay area startup, I designed and coded in Python the core of a system to match local businesses and application service providers for mobile devices. This work involved the development and implementation of a domain-specific query language.
- For a small company designing a next-generation semantic web platform, I developed a comparison parallel processing example in NVIDIA's CUDA. I also attend and help prepare for meetings directed towards the generation of partnerships or sales.

Ritsumeikan Asia Pacific University

Visiting Lecturer, 8/10

I taught an accelerated summer course in C++ through the School of Information, Computer, and Communication Technology (ICT).

Blue Vector Systems

Senior Software Architect, 1/08 – 8/09

I maintained, enhanced, and packaged software that was key to this dynamic, small firm's mission of providing an efficient, reliable, and powerful platform to support processing on the edge for large sense-and-respond networks. Reporting to the founder, I generalized this "weaver" software from a component in particular solutions to form the basis of an aspect-oriented programming-based development platform, and helped to redesign the platform for scalability. My accomplishments also included putting an advanced modeling and testing infrastructure into place.

Elo Systems, Inc.

Software Architect, 4/07 – 1/08

Senior Member of Technical Staff, 7/05 – 4/07

I designed and developed the core functionality of a recommendation engine that aids cross-selling by modeling configurations of related products for this small Silicon Valley startup. I worked directly with the CEO and founder to help him to refine and realize his vision with a solid mathematical foundation. I also devised and communicated robust interfaces to protect those responsible for developing downstream components.

Department of Computer Science, Indiana University

Instructor, Independent Study Program, 10/04 – 5/05

I taught Intro. to Programming with Recursion over the internet.

Research Assistant, 1/99 – 12/01

I helped draft a successful NSF grant proposal on structuring compilers.

Assistant Instructor, various 9/95 – 5/04

Undergraduate Courses Taught:

Discrete Structures for Computer Science (Summer 1997)

Graduate Courses Assisted:

Computational Complexity, Programming Language Principles

Undergraduate Courses Assisted:

Intro. to Programming, Intro. to Programming with Recursion, Software Systems, Compilers, Software Verification

Articles

Trampolining Architectures

with Daniel P. Friedman

presented at the Festschrift and Symposium in Honor of Mitchell Wand,
August 23, 2009, Boston, MA

Analyzes the extent to which the monadic underpinnings of trampolining, a translation that supports multithreaded applications, are able to withstand a more diverse set of systems offering additional power and efficiency.

Macros as Multi-Stage Computations:

Type-Safe, Generative, Binding Macros in MacroML

with Amr Sabry and Walid Taha

presented at the International Conference on Functional Programming
September 3–5, 2001, Florence, Italy

Demonstrates a significantly expressive macro system that allows for static checking of type and stage safety. Macro applications take the syntactic form of existing language constructs (including binding constructs); their semantics is modified.

Trampolined Style

with Daniel P. Friedman and Mitchell Wand

presented at the International Conference on Functional Programming
September 27–29, 1999, Paris, France

Formalizes a technique for writing multithreaded programs so as to clarify its connection with continuation-passing style.

An Object Encoding For SelfType

with Daniel P. Friedman

presented at the Midwest Soc. for Programming Languages and Systems

December 9, 2000, Chicago, IL
Technical Report #554, 12/00, Indiana Univ., Dept. of Computer Science

Encodes in standard type theory an object-oriented programming construct that allows types as well as programs to refer to *this*.

A Modular Monadic Interpreter In Scheme With Objects

with Daniel P. Friedman

Technical Report #553, 10/00, Indiana Univ., Dept. of Computer Science

Demonstrates an interpreter that uses monad transformers to provide a modular structure based on features of the interpreted language. In the process it provides an interesting use of metaobjects.

Interleaving is Possible with Refined Abstract Machines:

A New Approach to Engineering a Compiler

with Daniel P. Friedman

Technical Report #547, 6/99, Indiana Univ., Dept. of Computer Science

Presents a multi-stage compiler, each stage corresponding to an abstract machine, such that high-level code can be run on any lower-level machine. This is similar to the notion of just-in-time compilation.

Books

The Shameless Schemer (forthcoming)

with Daniel P. Friedman

This is to be the next installment in the “Little Schemer” series, which explores ideas in computer science through the Scheme programming language. With Daniel P. Friedman.

Encapsulation of State with Monad Transformers

Ph.D. Thesis, 12/06, Indiana Univ., Dept. of Computer Science, UMI

Models the containment of state operations within parts of a program using constructs from category theory. Both the dynamic monadic language and its type system use relative addressing of regions.

Technical Skills

Programming language semantics using category theory.

Programming language design and implementation.

Software verification with PVS, SMV.

Data modeling with RDBMS DDL (MySQL), JSON, XML/XSD,
(familiarity with RDF/RDF-S, OWL).

Object-oriented design with design patterns, UML.

Programming with C++, Java, Scheme, Python, RDBMS DML (MySQL),
(familiarity with Erlang, Scala, XQuery, SPARQL).

Development in Emacs and Eclipse environments, with Git and Subversion version control.

Professional Associations

Bay Area Artificial Intelligence Meetup Group

International Association of Software Architects

Silicon Valley Patterns Group

Association for Computing Machinery

Honors and Service

First Lego League

Design Judge, Los Altos Student Robotics Tournament
2007 – present

Sigma Xi Scientific Research Society

Associate Member

Graduate and Professional Student Organization

Indiana University

Constitutional Committee and Computer Science Representative
various 9/98 – 5/05

Graduate Education Committee

Department of Computer Science, Indiana University
9/98 – 5/00

Admissions Committee

Department of Computer Science, Indiana University
9/97 – 5/98