# Yin Wang

PhD student in Computer Science at Indiana University Bloomington
yw21@cs.indiana.edu

## SPECIALTIES

programming language semantics, lambda calculus, type systems, compilation, logic programming, abstract interpretation

## RESEARCH EXPERIENCE

Intersection types, session types and delimited continuations

Spring 2012 - current (with Amr Sabry)

*Implemented a polar intersection type inference system*

Non-graph-coloring register allocation methods

Fall 2011 (with R. Kent Dybvig)

*Designed a semantic-based register allocation method, implemented in a Scheme compiler*

Reversible computing

Spring 2009 (with Amr Sabry)

*Designed a reversible CEK abstract machine*

Logic programming language extension

Fall 2008

*Reimplemented the logic language miniKanren, implemented universal quantification and constraint-based negation*

## PUBLICATIONS

- *Yin Wang, R. Kent Dybvig, Register Allocation By Model Transformer Semantics, draft, 2011. (arxiv:1202.5539)*
- I write quite some technical blog posts here*: http://yinwang0.wordpress.com*

## TALKS

*"Towards Structural Version Control" (Feb 2012)*

*"The Little Register Allocator" (Nov 2011)*

*"Separation Logic, Faulty Logic and Monads" (April 2010)*

## TEACHING EXPERIENCE

B501: Graduate Theory of Computation (Indiana, spring 2012)

B461: Database Concepts (Indiana, fall 2011)

C212: Introduction to Software Systems (Indiana, fall 2008, fall 2009, spring 2010)

CS100H: Honors course for Java programming (Cornell, spring 2008)

CS682: Graduate Analysis of Algorithms (Cornell, fall 2007)

CS110: Introduction to Computing using Java (Cornell, spring 2007)

## WORK EXPERIENCE

Software Engineer Intern, Google Inc. (May 2010 – Aug 2010)
*type inference and control-flow analysis for Python*

Software Engineer Intern, Google Inc. (May 2009 – Aug 2009)
*semantic indexing for Python*

Software Engineer, UOneNet (Feb 2006 – Aug 2006)
*worked on a large-scale distributed online game engine*

## SKILLS

- Programming Languages*: Java, C, C++, Python, JavaScript, Pascal, Scheme, Common Lisp, Haskell, OCaml, Coq, Agda, SQL, Perl, shell, assembly*
- Operating Systems*: Linux(15 years of experience), Windows*

## SIGNIFICANT PROGRAM IMPLEMENTATIONS

1. *deep static analyzer* for Python which infers types and finds semantic bugs (in daily internal use by Google)
2. *parser combinator library* and *parsers* for C++, JavaScript, Scheme
3. physically-based *3D renderer*
4. physically-based *animation engine*
5. *Linux kernel device driver* and *Windows VxD device driver*
6. *optimizing compiler* from the Scheme programming language into X86_64 assembly code (with advanced transformation techniques and highly sophisticated register allocator)
7. *structural differencer* which compare program by parse trees and not text (open-source project)
8. various *type systems*:
    - Hindley-Milner polymorphic type system (as in SML, OCaml and Haskell)
    - $ML^F$ style type systems with first-class polymorphism
    - Intersection type systems
9. *logic programming language* miniKanren with constraint logic programming, universal quantification and negation operator
10. *interpreters* for various small programming languages with different semantic features (call-by-name, call-by-value, call-by-need, combinatory logic, delimited continuations etc.)
11. *online partial evaluator* for a subset of Scheme

12. one-pass *CPS and ANF transformers* with no administrative redexes (key concepts of JIT compilers)

## PORTFOLIO

Some of my code can be found at: *http://github.com/yinwang0*

## EDUCATION

**Indiana University**

PhD student, Computer Science, Sep 2008 – current (with one year on leave)

Research area: *programming languages*

**Cornell University**

M.S., Computer Science, 2006 – 2008

**Tsinghua University**

Computer Science, 2001 – 2006

**Sichuan University**

B.E., Computer Science, 1997 - 2001

**Oregon Programming Languages Summer School**

Participation, Logic, Languages, Compilation, and Verification, 2010 – 2010

## RELATED COURSES

B629: Language-Based Approaches to Security (Indiana, Amal Ahmed)
B629: Syntactic Abstraction and Source-level Optimization (Indiana, Kent Dybvig)
B621: Advanced Concepts in Programming Languages (Indiana, Dan Friedman, A+)
P523: Programming Language Implementation (compilers) (Indiana, Kent Dybvig, A+)
B522: Programming Language Foundations (Indiana, Amr Sabry)
B521: Programming Language Principles (Indiana, Dan Friedman, A+)
CS611: Advanced Programming Languages (Cornell, Dexter Kozen)

## AWARDS

Fellowship, Indiana University, 2006 (declined)

## LANGUAGES

*English (proficient)*

*Mandarin Chinese (native speaker)*

## HOBBIES

*biking, skateboarding, Frisbee, reading, running, cooking*