Alessandro Warth

Address:	10910 Wellworth Ave, Apt. 407 Los Angeles, CA 90024	**E-mail:**	awarth@cs.ucla.edu
Phone:	(310) 709-6562	**Homepage:**	http://www.cs.ucla.edu/~awarth

Objective To work on exciting research projects in Computer Science, preferably in the area of Programming Languages or Search Engines.

Education

SEP 2004 —PRESENT *University of California, Los Angeles.* Computer Science Ph.D. Student. Research Area: Programming Languages. Advisor: Dr. Todd Millstein. GPA: 3.933

AUG 1996 —DEC 2000 *University of Miami.* Bachelor of Science in Computer Engineering. May 2000. Cummulative GPA: 3.7. Dual major in Computer Science (GPA: 4.0). Graduated with Honors.

Patents

AUG 29, 2004 **Name:** Method of Authenticating a Payment Account User
International Publication Number: WO 02/067091 A3

NOV 29, 2001 **Name:** Method for Simplifying and/or Securing Transactions Over a Network
International Publication Number: WO 01/90994 A2

Research Experience

SEP 2004 —PRESENT **LazyJ: Lazy Evalution in Java**
Independently designed and implemented LazyJ, a backward compatible extension to the Java programming language that allows programmers to seamlessly tap into the power and expressiveness of lazy evaluation. LazyJ's technical contribution is its novel type system which includes lazy and eager types, and allows coercions between these types. Thanks to these coercions, programmers are able to write lazy code without ever having to worry about explicitly `delay`ing and `force`ing expressions. The LazyJ compiler is available at http://www.cs.ucla.edu/~awarth/lazyj.

JUN 2000 —DEC 2003 **Independent Research Project** with Dr. Stephen Murrell
In the early 80's, functional languages were implemented almost exclusively as *SECD (Stack, Environment, Control, Dump) machines*. In the late 80's, the research community's increased interest in *graph reduction machines* resulted in the the "disappearance" of the *SECD machine*. But did this happen because *graph reduction machines* are strictly better than *SECD machines*, or was it just an unfortunate coincidence? In order to answer that question, we produced a high-performance implementation of a functional programming language with lazy semantics and pattern-matching (similar to *Haskell*) using an *SECD machine*. Our interpreter runs between 1.5 and 3 times faster than *Hugs*, a popular *graph reduction*-based *Haskell* interpreter.

JUN 1999 —AUG 1999 **University of California, Berkeley**, *Undergraduate Researcher*
Duties: Extended Berkeley's implementation of the Virtual Interface Architecture (VIA). Worked on the Linux kernel and device drivers for high-performance networking hardware to add new functionality to the user library. Wrote a technical paper entitled "Event Support and RDMA Transfers in the Virtual Interface Architecture". Under the direction of Dr. David Culler.

Teaching Experience

SEPT 2005 —PRESENT
University of California, Los Angeles, *Teaching Assistant*
Course: *Programming Languages (ML, Prolog, Java, Smalltalk, Python).* Taught by Dr. Paul Eggert and Dr. Todd Millstein.

AUG 2000 —DEC 2000
University of Miami, *Teaching Assistant*
Course: *Introduction to Software Engineering (in C++).* Taught by Dr. Moiez Tapia and Dr. Stephen Murrell

AUG 1998 —MAY 2000
University of Miami, *Tutor / Grader*
Courses: *Data Structures and Algorithms, Object-Oriented Software Engineering, Operating Systems, Databases* (all taught by Dr. Stephen Murrell).

Work Experience

AUG 2000 —OCT 2003
Verid, Inc. (formerly **iShopSecure, Inc.**), *Senior Software Engineer*
Duties: Designed and implemented *Transact-Secure*, the company's flagship product. *TransactSecure* prevents credit card fraud and increases the security of both customers and merchants by verifying the identity of consumers during online and MOTO transactions using data provided by Equifax and Experian. To communicate with the data providers, *Transact-Secure* sends XML messages over TCP/IP sockets.

JUN 1998 —AUG 1998
AT&T RAPID Development Group, *Intern*
Duties: Enhanced AT&T's *Restoration And Provisioning Integrated Design* (*RAPID*) application, responsible for the self-healing characteristic of AT&T's Network, by writing an interface through which *RAPID* engineers can monitor *Virtual Network Links*, an important abstraction used by the routing algorithm. The work included writing C++ code for accessing persistent database objects through an *Object Request Broker* (*ORB*), and extending the existing *RAPID GUI* (written in Java) to provide access to this new functionality.

Fun Stuff

- Designed and implemented a multi-threaded web spider and search engine. Used my own index file format instead of using an off-the-shelf database system in order to optimize search performance.
- Wrote a *BCPL* compiler from scratch (without using lex and yacc, just C's standard library functions), complete with a Pentium assembler and linker/loader.

Skills

Programming language implementation (parsers, compilers, interpreters),
Java, C/C++, Smalltalk, Python, Haskell, OCaml, ML, Scheme, Prolog, Pentium Assembly Language, PHP, Perl, HTML, CGI, SQL, XML, XQuery, XSLT, Linux, FreeBSD, Mac OS X

Honors

- Recipient of the U.S. Departent of Education's GAANN Fellowship, 2004
- Honorable Mention—NSF Graduate Research Fellowship, 2004
- Recipient of the GEM Fellowship, 2004 (declined)
- Honor Societies: Tau Beta Pi, Eta Kappa Nu, Golden Key, Phi Kappa Phi
- 1999 Southeastern ACM Intercollegiate Programming Contest, 4th place
- George E. Merrick, Ann Bacheller, and Florida Undergraduate Scholarships
- National Dean's List, Provost's Honor Roll, Dean's Honor Roll (several times)

Languages Fluent in Portuguese and Spanish