Terence John Parr

terence@parr.us (415) 577-3138
https://www.linkedin.com/in/terence-parr/

http://parrt.cs.usfca.edu http://github.com/parrt
http://explained.ai http://github.com/antlr
Twitter: @the_antlr_guy

Interests

For 30 years, I have developed, maintained, and built online communities for open-source packages and tools, such as the ANTLR parser generator. Coding is still my passion and represents a significant chunk of my activity. While I am currently focused on creating libraries to interpret, visualize, and debug machine learning models, my expertise lies in programming language implementation, translation, and construction of developer tools.

Education

Ph.D., Computer Engineering, Purdue University; 1993
Dissertation: "Obtaining Practical Variants of $LL(k)$ and $LR(k)$ for $k > 1$ By Splitting the Atomic k-Tuple." Invented new parsing strategy that reduces parser time complexity from $O(n^k)$ to $O(n \times k)$ by approximating $k > 1$ lookahead.

MS Engineering, School of Electrical Engineering, Purdue University; 1990

BS Computer Science, School of Science, Purdue University; 1987

Significant Open-Source Projects

TensorSensor. This library clarifies exceptions from Python tensor libraries by augmenting messages and visualizing code to indicate tensor variable shape; works with JAX, Tensorflow, PyTorch, and Numpy, as well as higher-level libraries like Keras and fastai. https://github.com/parrt/tensor-sensor

dtreeviz. A Python library for visualizing decision trees and interpreting related models (scikit-learn, XGBoost, Spark MLlib, and LightGBM trees). dtreeviz also provides feature and classifier probability space illustrations. https://github.com/parrt/dtreeviz

lolviz. A simple Python data-structure visualization library that started out as a List Of Lists (lol) visualizer but now handles arbitrary object graphs, including function call stacks. lolviz tries to identify and nicely format common data structures such as lists, dictionaries, linked lists, binary trees, numpy arrays, and pandas data frames. https://github.com/parrt/lolviz

rfpimp. Identifying features with predictive power in random forests and other machine learning models is an important interpretation technique and this Python library provides permutation and drop-column importances for any scikit-learn model. As part of this work, we showed that random forest gini-drop feature importances can be highly biased. https://github.com/parrt/random-forest-importances

CodeBuff. Given a grammar and a code sample, this tool was an early attempt to use machine learning to format a new corpus in the same style. https://github.com/antlr/codebuff

ANTLR. Designer and project lead. ANTLR is a popular and widely-used parser generator that introduced a variety of parsing/translation techniques. http://antlr.org and https://github.com/antlr

ANTLR 4 Intellij IDE Plugin. Among other features, the plugin provides a "live programming" pane showing parse trees for input as programmers type and reflects changes to the grammar immediately without generating code. The plug-in has a sophisticated profiler that identifies ambiguous phrases, deep lookahead requirements, and grammar hotspots. https://github.com/antlr/intellij-plugin-v4

ANTLR 4 grammar repository. A collection of over 250 ANTLR contributed grammars. https://github.com/antlr/grammars-v4

StringTemplate. A Java template engine (with ports for C# and Python) for generating source code, web pages, emails, or any other structured text output. It is well-suited to building retargetable source code generators. http://stringtemplate.org

Employment

University of San Francisco; **Full professor of computer science**; 2014-present.

University of San Francisco; **Associate professor of computer science**; 2008-2014. Graduate program director in *computer science* Summer 2004-2014. Co-founder and founding director of *MS in Data Science* 2011-2014.

University of San Francisco; **Assistant professor of computer science**; 2003-2008.

jGuru.com. Cofounder and Chief Scientist San Francisco, CA; 1995-2004 jGuru.com was a well-respected and large independent site for Java developers. Solicited and received $5M private investment, managed 20+ people (10 Ph.D.s) when doing business as MageLang Institute, and implemented 110k-line jGuru server using Java/XML/RDBMS. *Sold in 2004 to Jupiter Media.*

Parr Research Corporation; President and Founder Minneapolis, Minnesota; 1994 - 1995 Software developement and consulting firm. Clients included NeXT Computer, Army Research Lab (Aberdeen Proving Grounds), Tandem, Computing Devices International, Berkeley Systems, Pencom.

Army High-Performance Computing Research Center; Postdoctoral Research Fellow Minneapolis, Minnesota; 1993 - 1994 Research interests: language translation tools and their role in parallel supercomputing.

Army High-Performance Computing Research Center; Predoctoral Fellow Minneapolis, Minnesota; September 1991 - August 1993 Involved in the formulation of portable, application-specific programming language (Fortran-P) and compiler for supercomputers (e.g. MasPar MP-1 and Thinking Machines CM-200, CM-5).

IBM; Software Engineer Lexington, Kentucky; June 1990 - December 1990 Developed translator that generated a proprietary IBM language from C++

Renault Automation; Engineer Paris, France; Direction des Techniques Avancees; January - June 1988. Completed work on compiler, interpreter, and debugger for KAREL (robot-control language), ported to industrial robot controller; continuation of work from Cybotech.

Cybotech Corporation; Software Engineer West Lafayette, Indiana; May 1986 - December 1987 Principle developer of compiler, debugger and environment for KAREL, a robot control language; supervised work of two other employees.

Lockheed Missiles and Space Company; Summer Technical Hire Sunnyvale, California; May - August 1984, 1985 Assistant system administrator for network of 45 Apollo workstations. Developed program to schedule calibration of fleet ballistic missile test consoles.

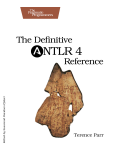
Purdue University Psychology Department; Software developer West Lafayette, Indiana; January - April 1984; September 1984 - May 1985 Created library of routines to control and monitor hardware functions required for psychological experimentation.

Kaman Sciences Corporation; Junior Programmer Colorado Springs, Colorado; May - August 1983 Developed graphics package for representation of data from nuclear tests.

Bio-Analytical Systems; Software Engineer West Lafayette, Indiana; September 1982 - May 1983 Developed software to collect and display data from chemical analysis hardware.

Books

"The Definitive ANTLR 4 Reference", Terence Parr, Pragmatic Bookshelf, Dallas Texas, January 2013. ISBN 978-1-93435-699-9. Sold 15,631 copies as of Sept 10, 2021. http://amzn.com/1934356999.



"Language Implementation Patterns", Terence Parr, Pragmatic Bookshelf, Dallas Texas, 2009. ISBN 978-1-93435-645-6. Sold 21,149 copies as of Sept 10, 2021. http://amzn.com/193435645X.

"The Definitive ANTLR Reference: Building Domain-Specific-Languages," Terence Parr, Pragmatic Bookshelf, Dallas Texas, May 2007. ISBN 0-9787392-5-6. Sold 17,777 copies as of Sept 10, 2021. http://amzn.com/0978739256.

Section in "Lucene in Action", Erik Hatcher and Otis Gospodnetic. Manning 2005.

"Language Translation Using PCCTS AND C++", Terence John Parr, Automata Publishing; San Jose, CA 1997 ISBN 0-9627488-5-4.

Papers in Refereed Journals

"Partial dependence through stratification," Terence Parr and James D. Wilson. Machine Learning with Applications. Volume 6 (Dec. 2021), https://doi.org/10.1016/j.mlwa.2021.100146

"ANTLRWorks: an ANTLR grammar development environment," Jean Bovet and Terence Parr. Software Practice and Experience. Volume 38, No. 12 (Oct. 2008), pp 1305-1332.

"The Fortran-P Translator: Automatic Translation of Fortran 77 Programs for Massively Parallel Processors," Matthew O'Keefe, Terence Parr, B. Kevin Edgar, Steve Anderson, Paul Woodward, and Hank Dietz; Journal of Scientific Programming, Vol. 4, pp 1-21, 1995.

"ANTLR: A Predicated-LL(k) Parser Generator," T.J. Parr and R.W. Quong; Journal of Software Practice & Experience, Vol. 25, No. 7; July, 1995.

Papers at Refereed Conferences

"*Towards a Universal Code Formatter through Machine Learning*," Terence Parr, Jurgen Vinju, Software Language Engineering (SLE) 2016; Amsterdam, NL 2016. *Awarded the Distinguished Paper Award.*

"*Adaptive LL(*)* Parsing: The Power of Dynamic Analysis," Terence Parr, Sam Harwell, Kathleen Fisher, OOPSLA; Portland, OR 2014.

"*LL(*)*: The foundation of the ANTLR parser generator," Terence Parr, Kathleen Fisher, Programming language design and implementation (PLDI), San Jose, CA 2011.

"Web Application Internationalization and Localization in Action," Terence Parr, International Conference on Web Engineering, Palo Alto, CA July 2006.

"Chronica: A Temporal Web Search Engine," Deniz Efendioglu, Chris Fraschetti, and Terence Parr, Poster paper, International Conference on Web Engineering, Palo Alto, CA July 2006. *Written with two USF graduate students.*

"Enforcing Strict Model-View Separation in Template Engines", WWW2004 conference, NYC May 2004. *Nominated for best paper (acceptance rate for WWW2004 was 14%).*

"A Language for Creating and Manipulating VRML", Terence Parr and Tim Rohaly, First Annual Symposium on the Virtual Reality Modeling Language, San Diego, 1995.

"Adding Semantic and Syntactic Predicates to LL(k): pred-LL(k)," Terence J Parr and Russell W. Quong; International Conference on Compiler Construction 1994; Edinburgh, Scotland; April 1994.

"An Overview of SORCERER-A Simple Tree-Parser Generator," Terence John Parr; Poster paper; International Conference on Compiler Construction 1994; Edinburgh, Scotland; April 1994.

Non-peer-reviewed Publications

"The Reuse of Grammars with Embedded Semantic Actions," Terence Parr, **Keynote presentation** at International Conference on Program Comprehension 2008. Amsterdam, Netherlands.

"LL and LR Translators Need $k > 1$ Lookahead," Terence J. Parr and Russell W. Quong; SIGNPLAN Notices, Vol. 31, No. 2, February 1996.

"PCCTS 1.00: The Purdue Compiler Construction Tool Set," T.J. Parr, H.G. Dietz, W.E. Cohen; SIGPLAN Notices, February 1992.

Web Publications

"Clarifying exceptions and visualizing tensor operations in deep learning code" Terence Parr, 2020 https://explained.ai/tensor-sensor/index.html

"Explaining RNNs without neural networks" Terence Parr, 2020 https://explained.ai/rnn/index.html

"A visual explanation for regularization of linear models" Terence Parr, 2020 https://explained.ai/regularization/index.html

"Nonparametric Feature Impact and Importance" Terence Parr and James D. Wilson https://arxiv.org/abs/2006.04750

"Technical Report: Partial Dependence through Stratification" Terence Parr and James D. Wilson https://arxiv.org/abs/1907.06698

"How to visualize decision trees" Terence Parr and Prince Grover, 2018 http://explained.ai/decision-tree-viz/index.html

"How to explain gradient boosting" Terence Parr and Jeremy Howard, 2018 http://explained.ai/gradient-boosting/index.html

"The Matrix Calculus You Need For Deep Learning" Terence Parr and Jeremy Howard, 2018 http://explained.ai/matrix-calculus/index.html

"Beware Default Random Forest Importances" Terence Parr, Kerem Turgutlu, Christopher Csiszar, and Jeremy Howard, 2018 http://explained.ai/rf-importance/index.html

"The Importance of Model-View Separation", Terence Parr and Bill Venners, 2008 http://www.artima.com/lejava/articles/stringtemplate.html

"Learn the essentials of debugging," Terence Parr, IBM DeveloperWorks, 2004 http://www-128.ibm.com/developerworks/web/library/wa-debug.html

"Humans should not have to grok XML," Terence Parr, IBM DeveloperWorks, 2001 http://www-128.ibm.com/developerworks/xml/library/x-sbxml.html

"Why we care about Java," Terence Parr, JavaWorld Magazine, 1997 http://www.javaworld.com/javaworld/jw-11-1997/jw-11-portability.html

Workshops

"ALL(*) model of parsing in ANTLR," Parsing @ SLE, Nov 2013; http://www.sleconf.org/blog/11-20-2013-parsing-at-sle-2013/

"Implementing parsers and state machines in Java," Java VM Summit, Sept, 2009; http://wiki.jvmlangsummit.com/images/c/c3/Parr_Java_Parsers.pdf

ANTLR2009; co-organizer and presenter; USF, June 6-7, 2009.

ANTLR2005; co-organizer and presenter; BEA Systems, San Francisco, October, 2005.

"The Role of Template Engines in Translation", Source-to-source 2004 workshop co-located with OOPSLA 2004; Vancouver, Canada; October 25, 2004.

ANTLR2004 (in cooperation with ACM); co-organizer and presenter; University of San Francisco, October 7-8, 2004.

PCCTS workshops; organizer and presenter at NeXT Computer July 1994, SGI July 1995, and Sun Microsystems August 1997.

"An Overview of SORCERER," SGI Compiler Summit; San Jose CA; June 26-28, 1994.

"Object-Oriented ANTLR Parsers," (Presented by R.W. Quong) "OO Compilation–What are the Objects?" workshop at OOPSLA 94; Portland OR.

Invited Presentations

"The Quest for the One True Parser," QCon San Francisco 2014, Nov 2014; http://qconsf.com/presentation/quest-one-true-parser

"ANTLR 4, Honey Badger," Boundary, February 13, 2013. http://www.youtube.com/watch?v=q8p1voEiu8Q.

"ANTLR 4, Honey Badger," Adobe, January 12, 2012.

"Why program by hand in 5 days what you can spend 5 years of your life automating?", Keynote presentation at Code Generation conference 2011, Cambridge, England, June 2011. http://www.infoq.com/presentations/Automation-DSL

"A Taste of StringTemplate," Netflix Inc. August 2009.

"The Reuse of Grammars with Embedded Semantic Actions," Aachen Institute for Advanced Study in Computational Engineering Science (AICES); Aachen, Germany; June 16, 2008.

"The Reuse of Grammars with Embedded Semantic Actions," Centrum Wiskunde & Informatica Amsterdam, Netherlands; June 10, 2008.

"ANTLR v3, ANTLRWorks, and StringTemplate", BEA Systems; April, 2005. *With partial presentation by USF grad student Jean Bovet*

"The Evolution of The StringTemplate Engine", Harmonia Research group, UC Berkeley; Dec. 2004.

"The ANTLR Parser Generator, Present and Future", University of Quebec at Montreal; Nov. 12, 2004.

"The Role Of Template Engines in Code Generation", Microsoft Research; Seattle, Washington; 2004.

"Language Translation, Domain Specific Languages, and ANTLR" with Loring Craymer, NASA JPL IT Symposium, October 2002.

"The ANTLR Parser Generator," Apple Computer; Cupertino, CA; February 1995.

"Language Translation with ANTLR and SORCERER," Sun Laboratories; Mountain View, CA; November 1994.

"PCCTS and It's Application to C++ Parsing," Lawrence Livermore National Lab; April 1994.

"An Overview of SORCERER," Argonne National Laboratories, Chicago Illinois; November 1994.

"An Introduction to PCCTS," IBM; Rochester, MN; April 1994.

"Parsing and Translation with ANTLR and SORCERER," Xerox Design Research Institute; Cornell University, Ithaca, NY; November 1993.

"Linear Approximation to Exponential LL(k) and LR(k) Lookahead," SUNY Albany; Albany, NY; November 1993.

"Translation with SORCERER," NeXT, Inc.; Redwood City, California; October 1993.

"Language Tools and Their Role in Scientific Computing," Konrad Zuse Institute of Berlin (ZIB); Berlin, Germany; September 1993.

"PCCTS," Technical University of Dresden; Dresden, Germany; September 1993.

"Advanced Parsing Strategies Using PCCTS," Argonne National Lab; Chicago, Illinois; July 1993.

"Advanced Parsing Strategies Using PCCTS: The ANTLR Parser Generator," Cray Research Inc.; Eagan, Minnesota; March 1993.

"The Role of Language Tools in Supercomputing," Army High-Performance Computing Research Center; Minneapolis, Minnesota; March, 1993.

Teaching

MSDS501 Computation for Analytics
MSDS692 Data Acquisition
MSDS621 Introduction to Machine Learning
MSDS689 Data Structures

CS110 Introduction to programming
CS245 Data Structures and Algorithms
CS345 Programming Language Paradigms
CS385 Special Lecture Series
CS342 Software Engineering
CS414 Compilers
CS601 Object-Oriented Software Development
CS652 Programming Languages
CS680 Web Systems and Algorithms
CS690 Masters Project

Expert Witness Activity

Enterprise Systems Technologies S.a.r.l. v Google. October 2014 - June 2015. Defended Google against infringement allegations related to Android and US patent 6691302 in jurisdiction of the *International Trade Commission*. Case was settled post deposition.

Oracle v Google. July 2011 - May 2012. Defended Google on 2 of 7 patent infringement allegations, 1 of which went to trial (US patent 6,061,520). Testified in *federal court* May 11, 2012. Jury found in favor of Google.
Testimony: http://www.groklaw.net/article.php?story=20120511165908331

Ajaxo, Inc. v Bank Of America Corporation. July-August, 2008. Defended Bank Of America against copyright infringement allegations. Settled in favor of Bank Of America.