Evolutionary Software Repair

Stephanie Forrest
Univ. of New Mexico
Albuquerque, NM
forrest@cs.unm.edu

Claire LeGoues
Univ. of Virginia
Charlottesville, VA
legoues@virginia.edu

http://www.sigevo.org/gecco-2012/

GECCO'12 Companion, July 7–11, 2012, Philadelphia, PA, USA.
ACM 978-1-4503-1178-6/12/07.

Instructors/Presenters

- **Stephanie Forrest** is Professor of Computer Science at the University of New Mexico in Albuquerque. She is also a member of the Santa Fe Institute External Faculty and co-Chair of its Science Board. She received the B.A. from St. John's College in 1977, and the M.S. and Ph.D. degrees in Computer Science from the University of Michigan (1982, 1985). Her research focuses on biocomputing, including evolutionary computation, immunology, complex systems, computer security, evolutionary diseases, and software engineering.
- **Claire LeGoues** is a graduate student in Computer Science at the University of Virginia. She received the A.B. degree in Computer Science from Harvard University in 2006. Her research develops methods for automated software repair.

Course Agenda

- Introduction
- Background information
- GenProg framework
- Benchmarks
- Experimental results
- Analysis of results
- Mutational robustness
- Future prospects and conclusions

Background Information and History

- Genetic Programming
- Search-based Software Engineering
- Evolutionary software repair
- Automated Program Repair

GenProg Framework

- Motivating Example
- Representations
  - Abstract Syntax Tree
  - Patches
- Fault Localization
- Operators
  - Mutation
  - Crossover
- Fitness functions

5

Benchmarks and Data Sets

- General requirements
  - Open source
  - Test suites
  - Documented bugs
- ICSE '09 Datset
- ICSE '12 Dataset
- Dataset challenges

6

Experimental Results

- Source-level repairs
  - Experimental setup---parameter settings
  - Experiments on ICSE '09 data set
  - Experiments on ICSE '12 Dataset
- Repairs on Assembly code and ELF binaries
  - Statistical fault localization
  - Constraining mutation operators
  - Additional data sets
  - Parameter settings
  - Experimental results

7

Analysis of Results

- How does performance scale with problem size?
- Most bugs are small
- Landscape analysis

8

Mutational Robustness

- Metrics
  - First-order mutations
  - Random Walks
- Experiments
  - Data sets
  - Is mutational robustness correlated with test suite size?
  - Does representation level affect mutational robustness?
  - Comparative robustness across languages
- Proactive diversity

Evolving software for other reasons

- Optimizing performance
  - Graphics shaders
- Optimizing energy

Conclusions

- Summary
- Future prospects
  - Iterated repairs
  - Closing the loop
  - Concurrency bugs
  - Evolving software from scratch
- Conclusions

References