Eric Collom

Division of Science and Mathematics University of Minnesota, Morris Morris, Minnesota, USA

29 April '14, UMM Senior Seminar



Overview

The big picture

- Evolving whole programs is hard to do with source code.
- Evolving whole programs with bytecode and assembly is not as hard.



Outline

Background

- 2 Why Evolve Instruction-level Code
- 3 FINCH: Evolving Programs
- Using Instruction-level code to automate bug repair
- Conclusions



Background

- EC
- Java Bytecode and x86 Assembly

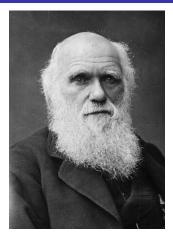


Background

•000000

What is Evolutionary Computation?

- EC is a a technique that is used to automate computer problem solving.
- Loosely emulates evolutionary biology.



Charles Darwin http://tinyurl.com/lgwj3wt

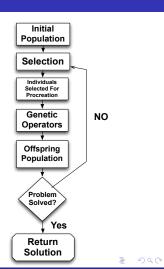


How does it work

Background

000000

- Continuous Optimization
- Selection is driven by the *fitness* of individuals
- Genetic Operators mimic sexual reproduction and mutation

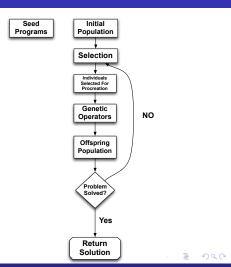


Genetic Programming

Background

0000000

- Uses the EC technique to evolve programs
- The population is programs

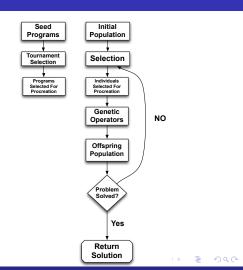


Genetic Programming

Background

0000000

Tournament Selection

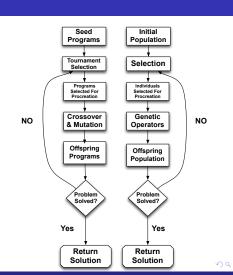


Genetic Programming

Background

0000000

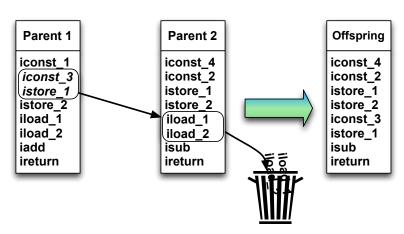
- Crossover
- Mutation



Crossover

Background

0000000



Crossover with Java Bytecode



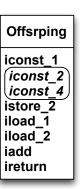
Mutation



Background

000000





Crossover with Java Bytecode



Bytecode and Assembly

Java Virtual Machine

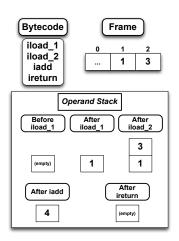
- Frames
- Array of local variables
- Operand Stack



Bytecode and Assembly

Java Bytcode and Frames

- Opcodes
- Prefix indicates type





Outline

- 1 Background
- 2 Why Evolve Instruction-level Code
- 3 FINCH: Evolving Programs
- 4 Using Instruction-level code to automate bug repair
- 5 Conclusions





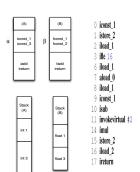


- 1 Background
- 2 Why Evolve Instruction-level Code
- 3 FINCH: Evolving Programs
 - How it Works
 - Results
- 4 Using Instruction-level code to automate bug repair
- 5 Conclusions



Selecting Offspring

- There is still a chance to produce non-compilable code
- Solution: Add restrictions to code selection.
- Stack and Frame Depth
- Variable Types
- Control Flow





How it works

Crossover



FINCH 000

How it works

Non-Halting Offspring



- 1 Background
- 2 Why Evolve Instruction-level Code
- 3 FINCH: Evolving Programs
- 4 Using Instruction-level code to automate bug repair
 - How it Works
 - Results
- 5 Conclusions



Selecting Offspring



Genetic Operators



How it Works

Non-Halting Offspring



- 1 Background
- 2 Why Evolve Instruction-level Code
- 3 FINCH: Evolving Programs
- 4 Using Instruction-level code to automate bug repair
- 5 Conclusions



Conclusions



References

