Ruoyi Wang

endvroy@gmail.com | ruoyi.me | endvroy

Research

Hybrid Training NPI

Oct. 2018 - present

- Reduce supervision for Neural Programmer Interpreters (NPI) via hybrid training
- Backpropagation on traces to bootstrap the model, and reinforcement learning on I/O examples to fine-tune the model
- Investigated the numerical relationship among the data size of each type, the training time, and the accuracy

Code Generation

Sep. 2018 - present

Supervisor: Anoop Sarkar

- General-purpose code generation from natural language descriptions with a grammar-agnostic model
- Copy mechanism with Pointer Network to improve the result
- Submitted to NAACL '18 for review

STOKE for HRM

Apr. 2018 - Aug. 2018

Supervisor: Nick Sumner

- Superoptimization for the Human Resource Machine architecture via Markov Chain Monte Carlo (MCMC) sampling
- Use average edit distance on output stream as program equivalence approximator to reward partial success
- Efficiently implemented in Rust

Test Case Reduction

Jun. 2018 - Aug. 2018

Supervisor: Nick Sumner

- Test Case Reduction by leveraging fact that sometimes the grammar of the tests is known
- Implemented as a skippy Earley Parser, adding a 'SKIP' action to forward Earley items, then extract test cases from the compact representation of the parse forest
- Allow for parallel testing because the method is not adaptive

SuperHuman

Nov. 2017 - Jan. 2018

Supervisor: Nick Sumner

- Search-based synthesizer and superoptimizer for the Human Resource Machine architecture
- Used handcrafted heuristics and symbolic I/O examples for aggressive pruning over the vast program space
- Able to find the shortest program(13 instructions) in 5 seconds within a single basic block
- Also tried syntax-guided synthesis (SyGuS) and prototyped with Rosette

Projects

Efficient Path Profiler *Nov. 2017*

- An LLVM-based, dynamic path profiler
- Records which execution path in each function is actually executed
- Uses an efficient encoding of execution paths as described in *Efficient Path Profiling*

Call Graph Profiler *Sep. 2017*

- An LLVM-based, dynamic call graph profiler
- Count the number of times each call site is actually invoked while maintaining its original functionality
- When calling through function pointer, the actual called function is inferred from the address stored in the pointer

Linguist *Jul. 2017*

- An LALR(1) compiler-compiler that is built upon itself
- Implemented a custom DFA-based regular expression engine that efficiently handles negative character sets
- User-specified actions can be associated with each lexical or grammatical rule

WebTetris *Dec. 2017*

- An online Tetris match system that runs in a browser
- Two players can join a match in the lobby, competing with real-time updates of the other player's status
- Implemented with Flask and websocket, designed to be anti-cheating

miniSQL *May 2016*

- Led a team of 3 and built our own database engine
- Supports lazy deletion, B+ tree indexing and block cache management with the Least Recent Used (LRU) policy
- Adopted a Test-Driven Development (TDD) approach

Awards

2018 **Undergraduate Open Scholarship**, Simon Fraser University *Vancouver, Canada*

2017 **Undergraduate Open Scholarship**, Simon Fraser University *Vancouver, Canada*

2016 **5th Place**, Microsoft Coding Competition *Vancouver, Canada*

Education

Simon Fraser University *Sep. 2016 - present*

B.Sc. in Computer Science *Vancouver, Canada*

Zhejiang University *Sep. 2014 - Jun. 2016*

B.Sc. in Computer Science *Hangzhou, China*