CSC2/455 Software Analysis and Improvement

Conclusion

Sreepathi Pai
Apr 26, 2023

URCS
Outline

What we’ve covered

What we’ve not covered

Where to go from here?
Outline

What we’ve covered

What we’ve not covered

Where to go from here?
Data Flow Analysis

- 3-address code, CFG
  - Value Numbering
- Global Iterative Data Flow Analyses
  - Live, Reaching Definitions, Available Expressions, Very busy Expressions
  - Constant Propagation
  - Dominators
- Foundations of DFA
  - Lattices, Monotonicity, Distributivity
  - Proofs of termination, etc.
- Static single assignment (SSA) form
Optimizations

- Partial Redundancy Elimination
- Instruction Scheduling
- Instruction Selection
- Register Allocation
Type checking and inference

- Typing rules
- Type equations
- Unification
- Handling polymorphism
Interprocedural Analysis

- Context-sensitivity
- Flow-sensitivity
- Points-to analysis
Region-based Analysis

- Reducible Graphs
  - Region-building
- Creating summaries of analyses
Loop Analysis

- How do different iterations of a loop access data?
- Most useful in automatic parallelization of programs
  - If data accesses don’t overlap, can execute the iterations at the same time on different processors
- A difficult problem, with the Integer Linear Programming (ILP) at the core
  - ILP is NP-complete
- Lecture slides from my 2018 edition of this course are available
  - Newer techniques used “polyhedral compilation”
Abstract Interpretation

- Abstraction
- Denotational Semantics
- Abstract Interpretation
  - Signs domain
  - Intervals domain
Model Checking and Symbolic Execution

- Bounded Model Checking
- Kripke Structures
- Liveness and Safety Properties
- CBMC
- KLEE
- Angr
Program Verification using Hoare Logic

- Basic ideas of verification
- Loop Invariants
- Axiomatic Semantics
  - \{P\} C \{Q\}
- Generating verification conditions
- Using program provers
  - Dafny
- Using theorem provers?
Outline

What we’ve covered

What we’ve not covered

Where to go from here?
Backend Optimizations

- Software Pipelining
Outline

What we’ve covered

What we’ve not covered

Where to go from here?
- Compilers
  - Diffuse: LLVM, GCC
  - Companies: Intel (icc), Microsoft (Roslyn, open-source), IBM (XL)
  - Specialist: Cray/HPE, PGI/NVIDIA, etc.
  - Every company at large enough scale has a compiler group (Google, Facebook, Amazon)
- Program Analysis
  - Facebook, Google, Amazon, etc.
  - Lots more (search for “static analysis tools”)
- Matthew Gaudet’s Compiler Jobs list
Graduate school

- Verification of Parallel and Concurrent Programs
- Verification of very large code bases
- Lots of theoretical and engineering problems in this field
Summer Suggestions

- Read all the CACM articles I’ve linked to!
- Get familiar with a real-world compiler (e.g., LLVM)
- Identify and start using program analysis tools for your favorite programming language
- Use and learn Dafny
- Model checking using TLA+ or Alloy
  - CACM article on Alloy, Alloy: A Language and Tool for Exploring Software Designs
  - Great topics for an independent study