Program Analysis in Software Development

Summary of Papers

Program Analysis Application Areas

- Compilation and Optimization
  - LLVM: A Compilation Framework for Lifelong Program Analysis & Transformation
- Obfuscation Generation and Detection
  - Abstract Stack Graph to Detect Obfuscated Calls in Binaries
- Model checking and code generation
  - Model Checking and Code Generation for UML State Machines and Collaborations
- Reverse engineering
  - Static Control-Flow Analysis for Reverse Engineering of UML Sequence Diagrams
- Program understanding
  - Selecting, Refining, and Evaluating Predicates for Program Analysis
- Error detection and Testing
  - Variably Interprocedural Program Analysis for Runtime Error Detection
- Program Maintenance
  - Evaluation of Software Modernization Estimation Methods Using NIMSAD Meta Framework

Studying The Literature

- For each of the papers presented
  - What does it try to accomplish?
  - What are the main steps in the proposed approach?
  - What can it be used for?
  - What are its limitations?
- What is the role of program analysis in each of the application areas?

Example: Compilation and Optimization

- LLVM: A Compilation Framework for Lifelong Program Analysis & Transformation
  - What does it try to accomplish?
    - provide high-level information to compiler transformations at
      - compile-time, link-time, run-time, and in idle time between runs
  - What are the main steps in the proposed approach?
    - a common low-level code representation in SSA form
      - a simple, language-independent type-system
      - instruction for typed address arithmetic
      - simple mechanism for exception handling
    - A compiler design that preserves program information and support
      - Offline code generation, profiling, whole-program optimization,…

Example: Compilation and Optimization

- LLVM: A Compilation Framework for Lifelong Program Analysis & Transformation
  - What can it be used for?
    - Enable life-long optimization for whole-programs, e.g.,
      - compile-time, link-time, run-time, and in idle time between runs
  - What are its limitations? (would you use it? What are your concerns?)
    - Requires the adoption of a new IR (intermediate/internal representation) and a new virtual machine (LLVM)
    - Conversion between other languages?
      - Support offline code generation, but is that sufficient?
    - Others …
- The role of program analysis in compilation and optimization
  - Discover semantics of programs (persistent program information)
  - Discover opportunities for optimization

Other application areas?

- Obfuscation Generation and Detection
  - Abstract Stack Graph to Detect Obfuscated Calls in Binaries
- Model checking and code generation
  - Model Checking and Code Generation for UML State Machines and Collaborations
- Reverse engineering
  - Static Control-Flow Analysis for Reverse Engineering of UML Sequence Diagrams
- Program understanding
  - Selecting, Refining, and Evaluating Predicates for Program Analysis
- Error detection and Testing
  - Variably Interprocedural Program Analysis for Runtime Error Detection
- Program Maintenance
  - Evaluation of Software Modernization Estimation Methods Using NIMSAD Meta Framework