

## CS653 Static and Dynamic Program Analysis

---

**Instructor:** **Michelle Strout**  
[mstrout@cs.colostate.edu](mailto:mstrout@cs.colostate.edu)  
USC 227  
Office hours: 3-4 Tuesday, 2-3 Thursday

**URL:** <http://www.cs.colostate.edu/~cs653>

These notes are loosely based on Amer Diwan's CSCI 7135 Notes

CS653 1/17/2006

Introduction

1

## Plan for Today

---

### Introductions

### Announcements

- OpenAnalysis research group meeting: Monday 1-2, Monday 2-3, Friday 1-2, Friday 2-3

### Course Goals and Outline

### Administrivia

- You are responsible for what is on the syllabus, on the schedule, and sent out to the class mailing list

### Paper Reviews

- reviews are due before class on the day the paper is being discussed

### Projects

CS653 1/17/2006

Introduction

2

## Course Goals

---

### Understand two big picture problems

- program optimization
- security

### In depth knowledge of program analysis ranging

- from entirely at compile-time
- to run-time
- to post run-time
- to barely any analysis at all.

### How these program analyses are applied to program optimization and security

### Approach

- read and critique recent and influential papers
- implement a program analysis idea

CS653 1/17/2006

Introduction

3

## Other Important Course Goals

---

### Research Exam Preparation

- paper reviewing skills will be critical for the research exam
- presenting the crucial points in a paper
- synthesizing ideas between papers
- suggesting future research questions

### Thesis Project Preparation

- how to break projects into manageable pieces
- how to leverage existing research tools
- presenting research in a succinct fashion

CS653 1/17/2006

Introduction

4

## Compile-time Program Analysis

---

### Discover properties of programs by looking at its source

- local (a few lines of straight-line code)
- global or intraprocedural (full procedure)
- interprocedural (several procedures)
  - special case: whole program analysis

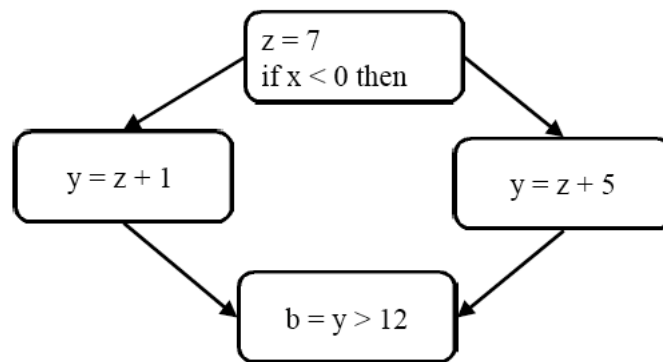
CS653 1/17/2006

Introduction

5

## Example of Compile-time Program Analysis

---



What are the possible values of  $y$ ?

Does answering this question require local, global, or interprocedural analysis?

CS653 1/17/2006

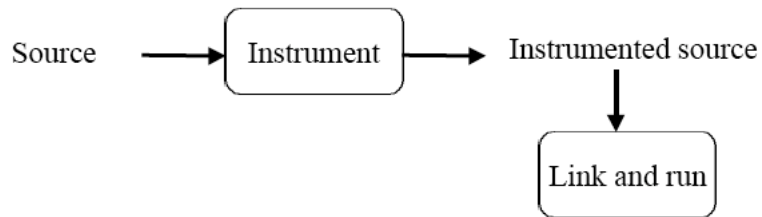
Introduction

6

## Run-time Program Analysis

---

Discovers properties of programs by examining its runs



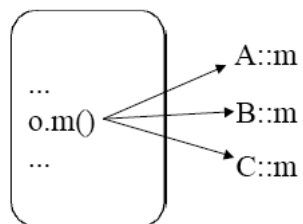
CS653 1/17/2006

Introduction

7

## Example of Run-time Program Analysis

---



**What is the most common target of `o.m()`?**  
**What can we do with that information?**

CS653 1/17/2006

Introduction

8

## Outline

---

**Read papers that use a range of static and analysis techniques to solve the problems that arise in program optimization and program security**

- data-flow analysis as a static analysis framework
- relationship between data-flow analysis and abstract interpretation
- applying data-flow analysis to security
- dealing with pointers
- other ways to improve analysis precision (context, flow, and path sensitivity)
- staged and dynamic analysis
- profiling to feed back into static analysis
- when data-flow analysis is not enough
- static and dynamic data dependence analysis
- don't analyze, just generate code

## Administrative Matters

---

**Turn to your syllabus**

## **Tips for Reading a Research Paper (by Tia Newhall)**

---

### **Read the paper three times**

- First read the abstract, introduction, related work, and conclusion
- Next read the entire paper writing down questions and vocabulary words you need to look up
- Finally re-read the paper critically, answer the paper review questions

### **Questions to answer in a paper review**

- What problem did the paper address?
- Is it important/interesting? What was the context for the paper?
- What is the approach used to solve the problem?
- How does the paper support or otherwise justify the conclusions it reaches?
- What problems are explicitly or implicitly left as future research questions?

## **Selecting a Paper to Present**

---

**Read the abstract, introduction, and conclusion for all papers you are seriously considering**

**Consider selecting a paper related to your course project**

**Schedule around your deadlines in this course and other courses**

**Paper selection will be first-come-first-serve**

- I will start posting assignments as soon as they are sent
- You must indicate which paper you would like to present by January 26
- There are 21 possible papers, don't panic!

## Project

---

### Five deliverables

- proposed tool example with preliminary project proposal
- proposal and tool example
- verbal status report (15 minutes)
- intermediate report
- final report and elevator speech

### Possible projects accessible from colostate.edu

## Tool Example

---

### C-breeze

- last semester wrote a walker that counted operations in the program

### LLVM compiler infrastructure

- <http://llvm.cs.uiuc.edu/docs/Projects.html>, describes how to create a project that uses LLVM
- could count operations or memory references or if statements or ...

### TAU: Tuning and Analysis Utilities

- <http://www.cs.uoregon.edu/research/tau/home.php>
- could profile one of your benchmarks from last semester

### OpenAnalysis

- could count the different types of memory reference expressions

## Possible Projects

---

<http://www.cs.colostate.edu/~cs653/Project/possible-projects.html>

## Next Time

---

### Reading

- “Quantifying Behavioral Differences Between C and C++ Programs” by Brad Calder, Dirk Grunwald, and Benjamin Zorn

### Class

- Example paper presentation

### Due

- Paper review for the Calder, Grunwald, and Zorn paper