Seminar on Program Analysis

Winter semester 2014
TU Darmstadt

Dr. Michael Pradel

Plan for Today

1. Organization

2. Topics

3. Recent research from the Software Lab

Why Have a Seminar?

- **Learn fundamentals of doing research**
  - Read and digest papers
  - Present complex ideas to others
  - Scientific writing
  - Reviewing
- **Learn about program analysis**
  - Maybe your future thesis topic
  - Opportunities for HiWis

Organization

- **Weekly meetings**
- **1 topic per week**
  - Each topic: 3 papers
- **1 to 3 talks per week**

- **Your task:**
  - Talk
  - Term paper
  - Reviews

Organization

- **Weekly meetings**
- **1 topic per week**
  - Each topic: 3 papers
- **1 to 3 talks per week**

- **Your task:**
  - Talk ← 40%
  - Term paper ← 40%
  - Reviews ← 20%

Grading:

Talk

- **20 minutes + questions**
- **English**
- **Present 1 of the 3 papers**

- **Get feedback afterwards**
  - Don’t repeat mistakes made by others
  - Expectations will increase during the semester

Talk: Some Advice

Content:

- No need to explain all technical details
- But: Must contain some ”meat”

Presentation:

- Examples are your secret weapon
- Stick to the time limit
- Practice, practice, practice

Search for *How to give a good research talk* by Simon Peyton Jones

Talk: Rules

- You may use parts of existing slides
- You may use examples from the paper

Term Paper

- 6 pages
- English
- LaTeX template on course web site
- Summarize all 3 papers and relate them to each other
- Must be self-containing

Term Paper: Some Advice

- Don’t waste space on basics
- Examples are your secret weapon (yes, again)
- Most important part: Comparison of the three papers
- Bad English distracts from good content
- Revise, revise, revise

Term Paper: Rules

- No verbatim copying of text (exception: quotes)
- You may copy figures (e.g., result graphs)
- You must use your own example(s)

Reviews

- **Imitates peer reviewing process**
- **Each participant reviews three term papers**
- **Revise your term paper after getting reviews**
  - Grade will be for final term paper

- **Plain text format**
- **About 1 page, English**

Reviews: Some Advice

- Be constructive
- Be polite
- Your reviews contribute to your grade, not to the reviewee's grade

Dates

- **Regular meetings:**
  Oct 27, 2014 – Feb 9, 2015

- **Deadlines:**
  - Oct 15, 2014: Topic selection
  - Jan 12, 2015: Term papers
  - Jan 29, 2015: Reviews
  - Feb 15, 2015: Revised term papers

- **Details on submission: See web site**

Plan for Today

1. Organization ✓

2. Topics

3. Recent research from the Software Lab

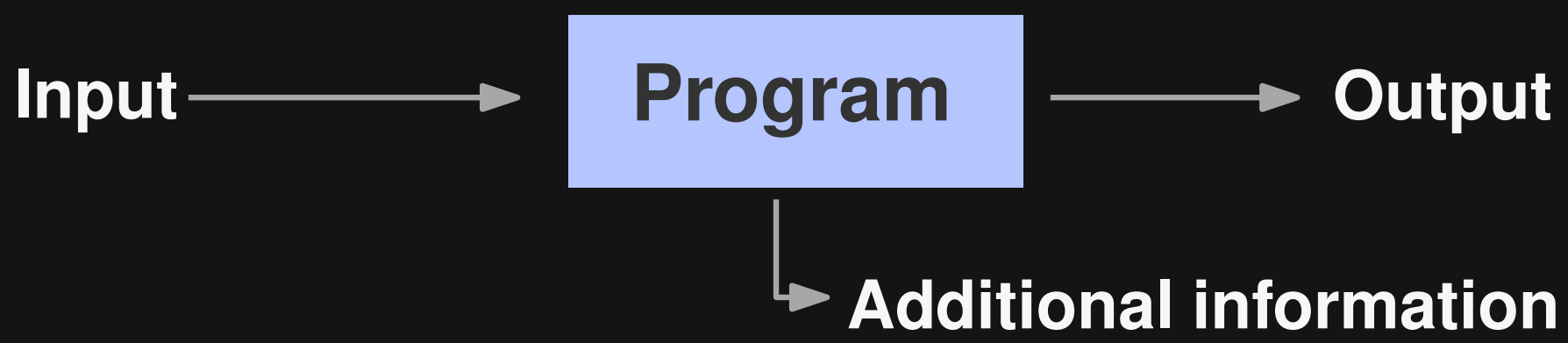
What is Program Analysis?

What is Program Analysis?

- **Automated analysis of the behavior of a program**
- **Common goals:**
  - Find programming errors
  - Optimization
- **Static vs. dynamic**

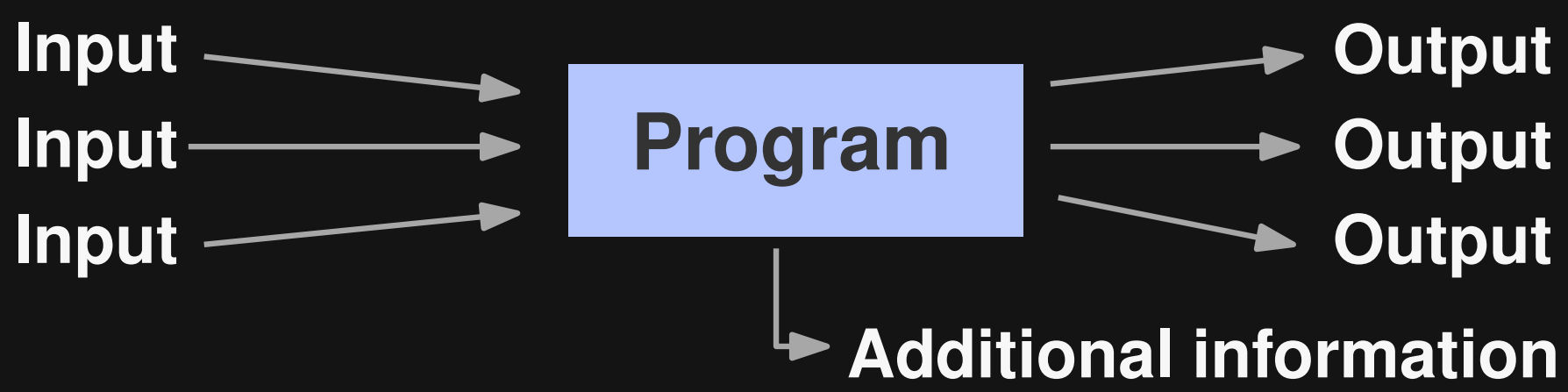
What is Program Analysis?

- **Automated analysis of the behavior of a program**
- **Common goals:**
  - Find programming errors
  - Optimization
- **Static vs. dynamic**



What is Program Analysis?

- **Automated analysis of the behavior of a program**
- **Common goals:**
  - Find programming errors
  - Optimization
- **Static vs. dynamic**



Topic: Anomaly Detection

Find bugs without any specification,
by inferring knowledge from the program

Example 1:

```
if (x == null) {
  x.foo();
}
```

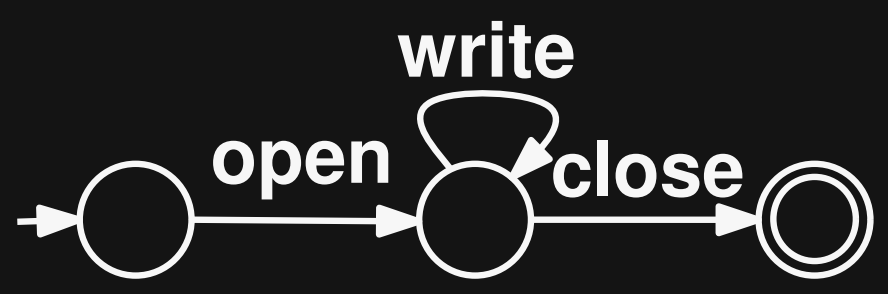
Topic: Anomaly Detection

Find bugs without any specification, by inferring knowledge from the program

Example 1:

```
if (x == null) {
  x.foo();
}
```

Example 2:



Topic: Concurrency Bugs

Analyze thread interleavings to find non-deterministic bugs

Example:

Thread 1

```
...
if (x != null) {
  x.foo();
}
...
```

Thread 2

```
...
x = null;
...
```

Topic: Concurrency Bugs

Analyze thread interleavings to find non-deterministic bugs

Example:

Thread 1

```
...
if (x != null) {
  x.foo();
}
...
```

Thread 2

```
...
x = null;
...
```

Error

Topic: Web and GUI Applications

UI-level test generation that automatically triggers sequences of input events

Example:

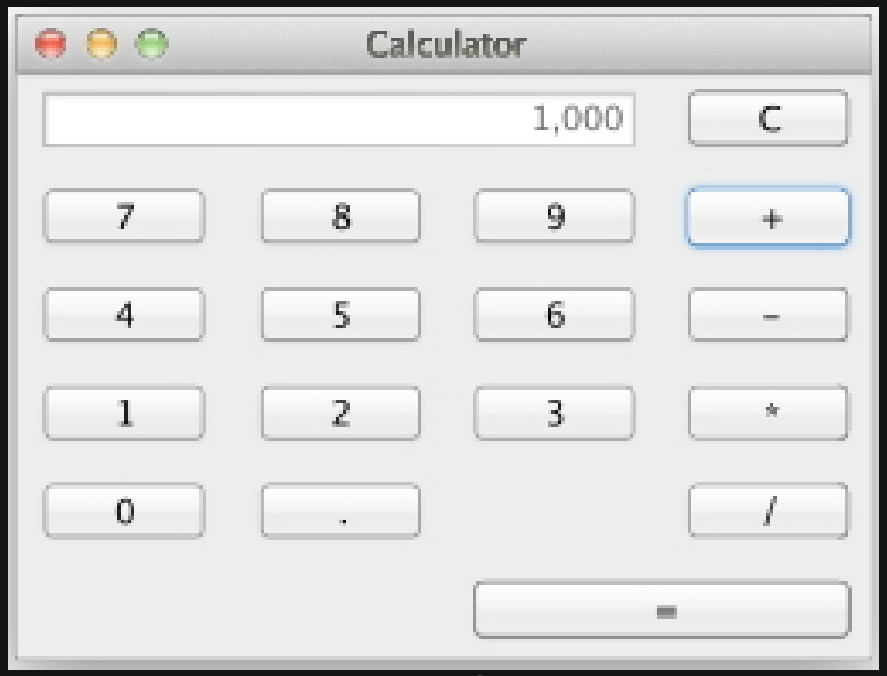


Image from Gross et al., 2012

Topic: Type Analysis f. Dyn. Lang.

Infer types for languages without type annotations and report type errors

Example:

```
// JavaScript
var obj = {x: "abc", y: 23};
var nb = 42 - obj.x;
```

Topic: Type Analysis f. Dyn. Lang.

Infer types for languages without type annotations and report type errors

Example:

```
// JavaScript
var obj = {x: "abc", y: 23};
var nb = 42 - obj.x;
```

NaN

Topic: Performance Bugs

Detect code with unnecessarily high CPU or memory requirements

Example:

```
for (Item item : items) {
  for (int i = 0; i < max; i++) {
    var x = expensive();
    ...
  }
}
```

Topic: Performance Bugs

Detect code with unnecessarily high CPU or memory requirements

Example:

```
for (Item item : items) {
  for (int i = 0; i < max; i++) {
    var x = expensive();
    ...
  }
}
```

Hoist out of loop if `expensive` is independent of `i`

Topic: Concolic Execution

Systematically explore paths by combining symbolic and concrete execution

Example:

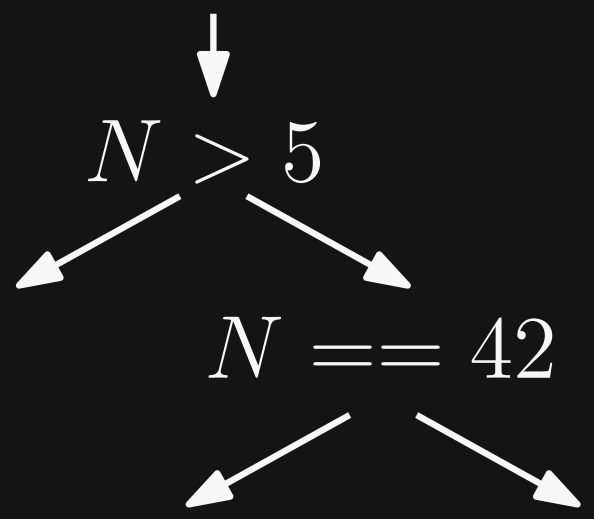
```
// Input: N
if (N > 5) {
  if (N == 42) {
    // error
  }
}
```

Topic: Concolic Execution

Systematically explore paths by combining symbolic and concrete execution

Example:

```
// Input: N
if (N > 5) {
  if (N == 42) {
    // error
  }
}
```

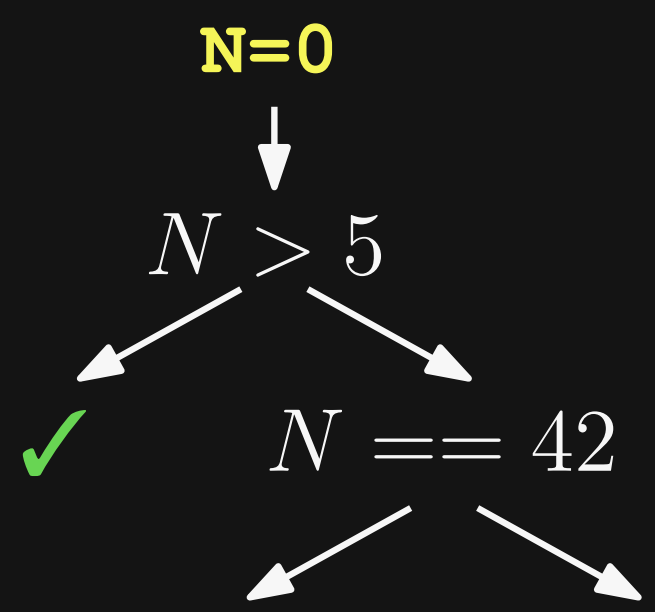


Topic: Concolic Execution

Systematically explore paths by combining symbolic and concrete execution

Example:

```
// Input: N
if (N > 5) {
  if (N == 42) {
    // error
  }
}
```

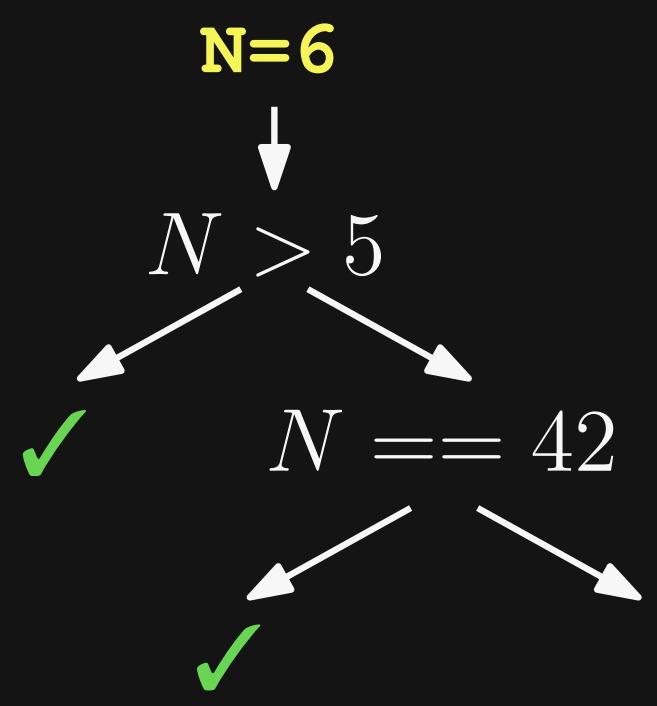


Topic: Concolic Execution

Systematically explore paths by combining symbolic and concrete execution

Example:

```
// Input: N
if (N > 5) {
  if (N == 42) {
    // error
  }
}
```

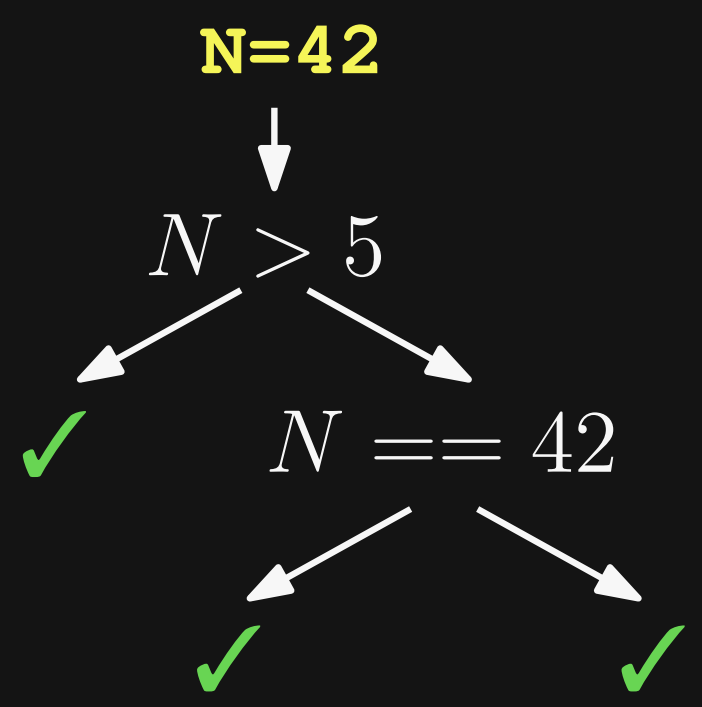


Topic: Concolic Execution

Systematically explore paths by combining symbolic and concrete execution

Example:

```
// Input: N
if (N > 5) {
  if (N == 42) {
    // error
  }
}
```



Topic: Random Test Generation

Generate test cases by creating random inputs (e.g., random method calls)

Example:

```
LinkedList l1 = new LinkedList();
Object o1 = new Object();
l1.addFirst(o1);
TreeSet t1 = new TreeSet(l1);
Set s1 = Collections.unmodifiableSet(t1);
assert(s1.equals(s1));
```

Fails (Sun's JDK 1.5)

Example from Pacheco et al. , 2007

Topic: Random Test Generation

Generate test cases by creating random inputs (e.g., random method calls)

Example:

```
LinkedList l1 = new LinkedList();
```

Example from Pacheco et al. , 2007

Topic: Random Test Generation

Generate test cases by creating random inputs (e.g., random method calls)

Example:

```
LinkedList l1 = new LinkedList();
Object o1 = new Object();
```

Example from Pacheco et al. , 2007

Topic: Random Test Generation

Generate test cases by creating random inputs (e.g., random method calls)

Example:

```
LinkedList l1 = new LinkedList();
Object o1 = new Object();
l1.addFirst(o1);
TreeSet t1 = new TreeSet(l1);
Set s1 = Collections.unmodifiableSet(t1);
```

Example from Pacheco et al. , 2007

Topic: Random Test Generation

Generate test cases by creating random inputs (e.g., random method calls)

Example:

```
LinkedList l1 = new LinkedList();
Object o1 = new Object();
l1.addFirst(o1);
TreeSet t1 = new TreeSet(l1);
Set s1 = Collections.unmodifiableSet(t1);
assert(s1.equals(s1));
```

Fails (Sun's JDK 1.5)

Example from Pacheco et al. , 2007

Topic: Security Vulnerabilities

Find security-critical bugs, e.g., potential SQL-injection attacks

Example:

```
$id = $_GET['id'];
//construct SQL statement
$sqlStmt = "SELECT␣*␣FROM␣Users
␣␣␣␣␣␣␣␣␣␣␣␣WHERE␣UserId␣=␣$id";
```

Topic: Security Vulnerabilities

Find security-critical bugs, e.g., potential SQL-injection attacks

Example:

What if `id` is:

`"23; DROP TABLE Users"`?

```
$id = $_GET['id'];
//construct SQL statement
$sqlStmt = "SELECT * FROM Users
            WHERE UserId = $id";
```

Topic: Differential Testing

Compare two supposedly equivalent programs with each other

Example:

```
int foo (void) {
  signed char x = 1;
  unsigned char y = 255;
  return x > y;
}
```

Example from Yang et al., 2011

Topic: Differential Testing

Compare two supposedly equivalent programs with each other

Example:

```
int foo (void) {
  signed char x = 1;
  unsigned char y = 255;
  return x > y;
}
```

Returns `1` when compiled with Ubuntu's GCC, but `0` with the GCC base version

Example from Yang et al., 2011

Topic Selection

- **Full list of topics and papers: Course web site**

- **Send your preferences by Oct 15, 2014**
  - Pick 3 topics (sorted by priority)
  - Optionally, indicate which paper you want to present

$\rightarrow$ **Counts as course registration**

Plan for Today

1. Organization ✓

2. Topics ✓

3. Recent research from the Software Lab