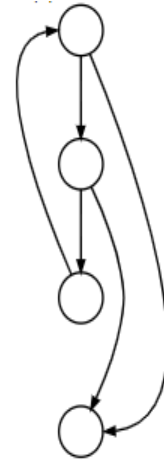


Program Analysis

Venkatesh Vinayakarao

venkateshv@cmi.ac.in
Mar – Apr, 2018
Chennai Mathematical Institute



“The supply of grand challenges ... shows little sign of drying up.”

– Harman and O’Hearn in “Opportunities and Open Problems for Static and Dynamic Program Analysis”, Madrid, Spain, 2018.

Agenda

- What is Program Analysis?
- Why Study Program Analysis?
- In this Course...
- Course Logistics

What is Program Analysis?

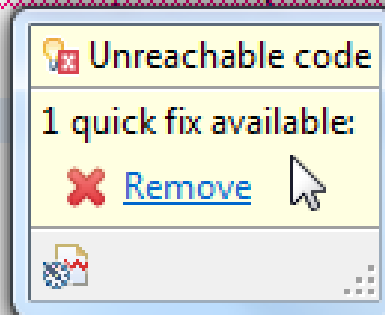
Quiz

```
class Immortal {  
    public static void main(String[] args) {  
  
        int x;  
  
        x = 1;  
        while (true) {  
            x = -x;  
        }  
  
        System.out.println("Result = " + x);  
    }  
}
```

Any problem in
this code?

Built into Eclipse

```
1  
2 public class Immortal {  
3     public static void main(String[] args) {  
4         int x;  
5  
6         x = 1;  
7         while(true) {  
8             x = -x;  
9         }  
10  
11     System.out.println("Result = " + x);  
12 }  
13 }  
14
```



Quiz

```
private static int test() {  
    int x;  
    int y;  
    y = x;  
    return x;  
}
```

Any problem in
this code?

IDE Catches Some!

```
13 private static int test() {  
14     int x;  
15     int y;  
16     y = x;  
17  
18 }
```

The local variable x may not have been initialized

Source Code Optimization

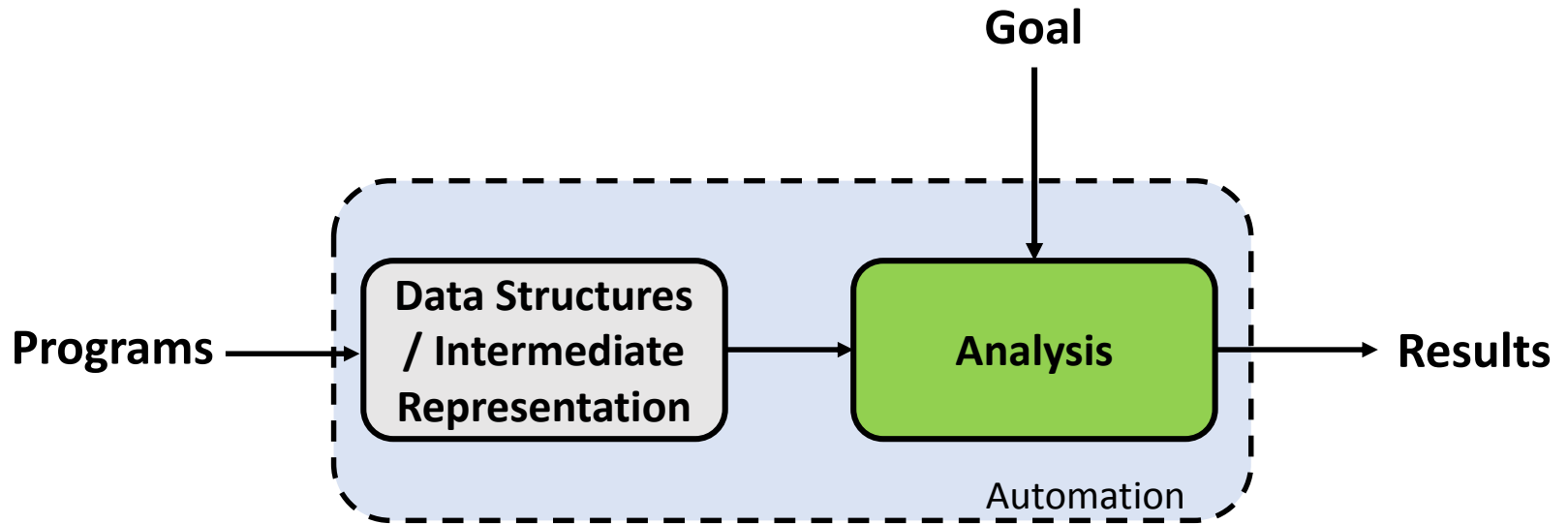
- How to optimize this?

```
if (x != 5) x = 5;
```

- Simply,

```
x = 5;
```


Program Analysis



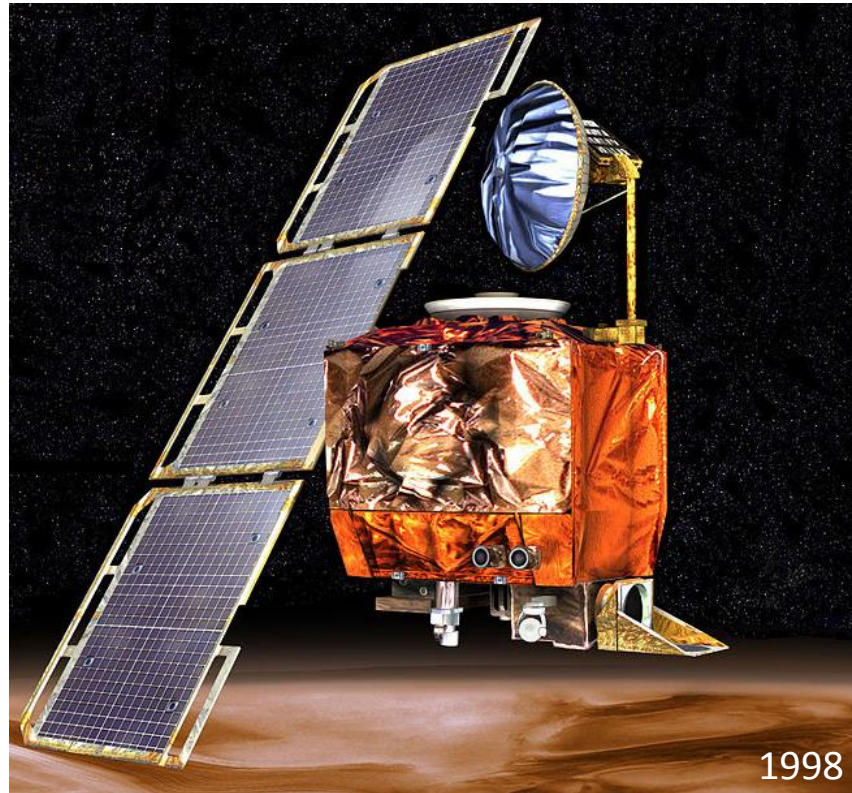
Why Study Program Analysis?

Everyone is in a hurry...



Picture Courtesy: Web.

Software Reliability: An Issue



For more, visit <http://www.cs.tau.ac.il/~nachumd/horror.html>

Mars Orbiter Crash

- Primary Cause: Results reported in wrong units
- "Various officials at NASA have stated that NASA itself was at fault **for failing to make the appropriate checks and tests** that would have caught the discrepancy."

Security Breaches

Aadhaar details leaked after TRAI chief throws breach challenge

Alleged personal details of Indias telecom watchdog chief R.S. Sharma were leaked on Saturday after the TRAI chairman threw a challenge and tweeted his 12-digit Aadhaar asking if it had made him vulnerable to any security risk.

IANS | Updated: July 29, 2018, 08:47 IST



Elliot Alderson @fs0c131y · Jan 30

With more than 100,000,000 downloads @ESFileExplorer is on famous Android file manager. Bonus: the list of applications in victim's phone is stored in an unsecured way.



ES File Explorer is leaking the apps installed on yo...

With more than 100,000,000 downloads ES File Explorer is one of the most famous Android file manager. Bonus: the list of applications installed on the tel is...

[youtube.com](https://www.youtube.com)

10 34 87

2:59 AM - 12 Feb 2019

Pinned Tweet



Elliot Alderson @fs0c131y · Feb 12

A new #Aadhaar breach is coming cc @UIDAI 🤔

101 462 1.2K

Microsoft Research

[Working at Microsoft](#) ▾ [Students and graduates](#) ▾ [Find a job](#) ▾ [Things to do](#) ▾

[← Back to search results](#)

RSDE

Looking for an individual that can apply programming language techniques to improve the performance and correctness of software executing in the cloud. The cloud is a major investment for Microsoft, costing us large sums of capital investment and requiring high quality of service guarantees for our customers.

We have already demonstrated through several RiSE projects that applying PL techniques to these problems (including model checking, symbolic execution, semantic abstractions, etc.) we can significantly improve cloud software over the current state of the art. Existing projects in this area include P and P#, Parasail, Uncertain, Network Verification, and Retro. RiSE has been successful in applying foundational reasoning to cloud software problems and leveraging our deep understanding and tool investment (e.g., Z3 SMT solver, Zing model checker, etc.) to create unique and effective solutions. A candidate should have both

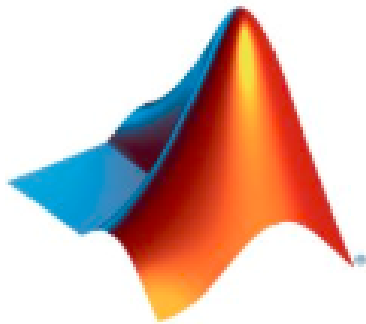
Job # 981039
Locations United States, Redmond (WA)
Job families Research
Teams Research

[Apply now](#)

[Add to job watch list](#)

Empower your future

MathWorks



Software Engineer - Dynamic Program Analysis

MathWorks · Bangalore, IN

Posted 2 weeks ago · 64 views

Save

Apply

who is good at abstract thinking
is a plus. You will join a dynamic
and debugging capabilities to
to learn many of our core

technologies and apply your design and implementation skills to build parts of our product from ground up.

- Design data-structures and algorithms for data-flow analysis of Simulink/Stateflow models and generated code
- Build customer visible UIs for configuring and invoking analysis and transformation engines
- Participate in architecture and design reviews

Oracle Labs

Oracle Labs Australia

Overview

External Presence

Careers

Visitor Information

▲ WORKING WITH US

Interested in working for Oracle Labs, Australia?

- ➔ Watch [this](#) to find out more about the projects we work on.
- ➔ Watch [this](#) to hear some testimonials from previous students.

▲ CURRENTLY ADVERTISING

The following [CEED](#) internships are currently available for students.

- ➔ [Compiling MySQL to LLVM for Static Program Analysis](#)
- ➔ ~~[Analysis of Software Defined Networks for the Cloud](#)~~
- ➔ [Verifying Cloud Security using Attack Graphs](#)
- ➔ [Security Analysis of Open Source Java Enterprise Applications](#)
- ➔ [PDF Malware Detection Tool](#)
- ➔ [Bug Finding Metrics Visualisation](#)

IBM IRL

Productive Parallel Programming

Current object-oriented languages have revealed several drawbacks with respect to parallel/concurrent programming at the level of unstructured threads with lock-based synchronization. IBM Research is developing X10, a modern object-oriented programming language designed for high performance with explicit programmer defined parallelism for realizing high productivity programming of parallel computer systems. The key features of X10 include explicit reification of locality in the form of places, support for a partitioned global address space (PGAS) across places, and lightweight activities embodied in `async`, `future`, `foreach`, and `ateach` constructs which subsume communication and multithreading operations in other languages. Our current focus is on static program analysis (for example, May-Happen-in-Parallel analysis, Bad Place Analysis), compilation for C/C++, debugging for X10, and assessment and semi-automated migration of domain-specific serial code to emerging multi-core architectures, leveraging productive programming models and their variants (such as OpenMP, OpenCL).

More...



CodeSearch - Senior Software Engineer - Program Analysis

Elastic · Yokohama-shi, JP

Posted 2 weeks ago · 207 views

Save

Apply




Principal Researcher - Phd/Program Analysis

Oracle · Brisbane, Australia

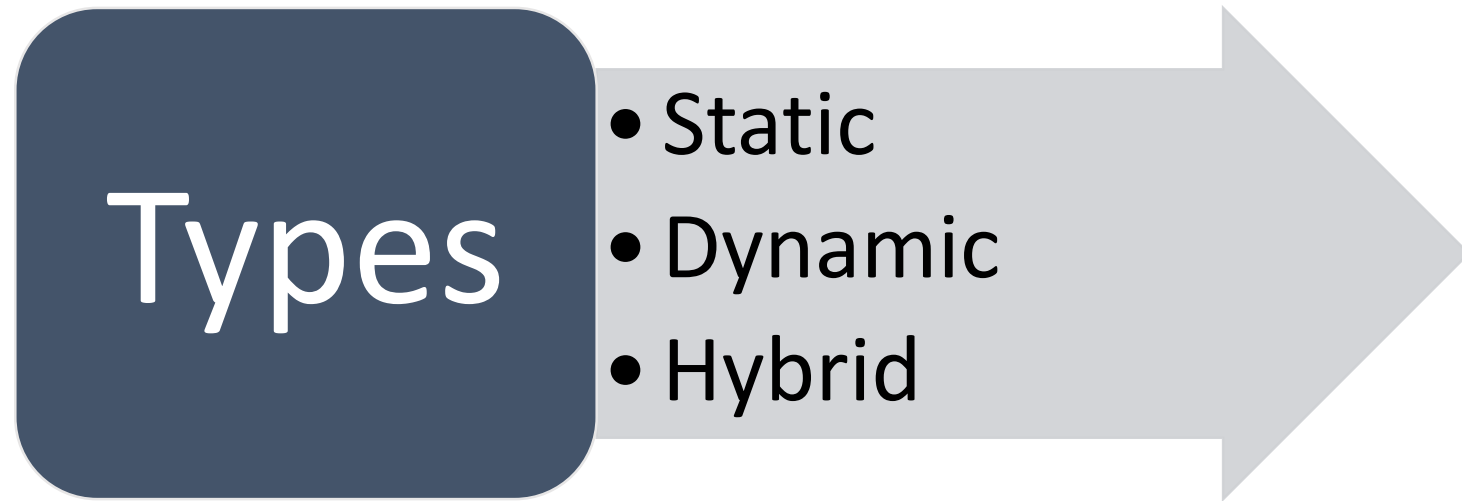
Posted 6 days ago · 79 views

Save

 Easy Apply

In This Course...

Analysis



Static Analysis

Analysis without executing the program

In this course, we will learn...

Reaching
Definitions
Analysis

Live Variable
Analysis

Available
Expressions
Analysis

Very Busy
Expressions
Analysis

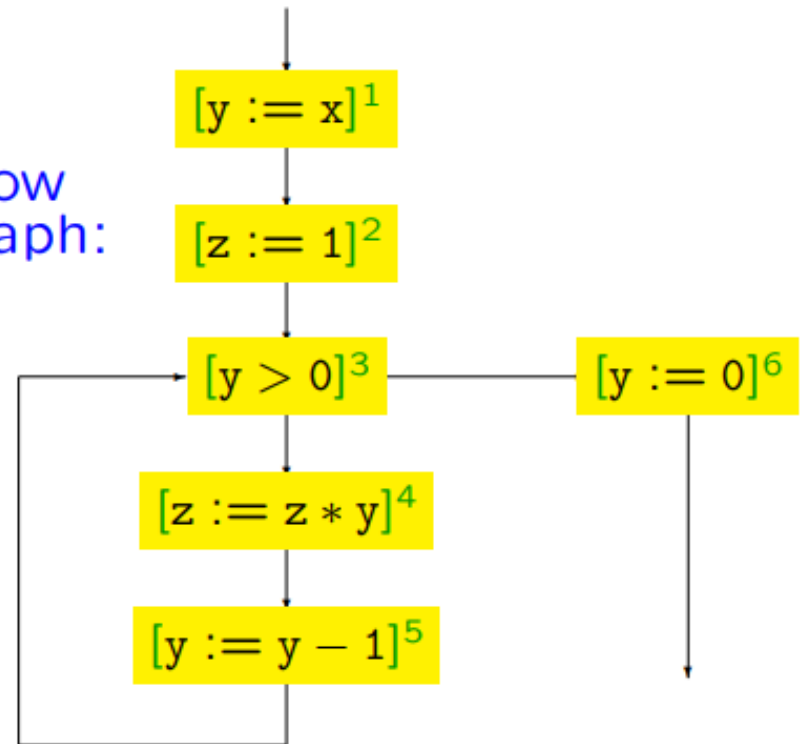
Generalized
Data Flow
Analysis
Framework

Implement using
Soot

Reaching Definitions (RD) Analysis

```
y = x;  
z = 1;  
while (y > 0) {  
    z = z * y;  
    y = y - 1;  
}  
y = 0
```

Flow
graph:



The assignment $[x := a]^{\ell}$ reaches ℓ' if there is an execution where x was last assigned at ℓ . Does $[z := 1]^2$ reach 5?

RD Analysis

ℓ	$RD_{\text{entry}}(\ell)$	$RD_{\text{exit}}(\ell)$
1	$(x, ?), (y, ?), (z, ?)$	$(x, ?), (y, 1), (z, ?)$
2		
3		
4		
5		
6		

$(y, ?) \rightarrow y$ is undefined. $(y, 1) \rightarrow y$ is defined in line 1.

Labeled Input Program

```
[y = x]1;  
[z = 1]2;  
while [(y > 0)]3 {  
    [z = z * y]4;  
    [y = y - 1]5;  
}  
[y = 0]6
```


RD Analysis

ℓ	$RD_{\text{entry}}(\ell)$	$RD_{\text{exit}}(\ell)$
1	$(x, ?), (y, ?), (z, ?)$	$(x, ?), (y, 1), (z, ?)$
2	$(x, ?), (y, 1), (z, ?)$	$(x, ?), (y, 1), (z, 2)$
3		
4		
5		
6		

Labeled Input Program

```
[y = x]1;  
[z = 1]2;  
while [(y > 0)]3 {  
    [z = z * y]4;  
    [y = y - 1]5;  
}  
[y = 0]6
```

RD Analysis

ℓ	$RD_{\text{entry}}(\ell)$	$RD_{\text{exit}}(\ell)$
1	$(x, ?), (y, ?), (z, ?)$	$(x, ?), (y, 1), (z, ?)$
2	$(x, ?), (y, 1), (z, ?)$	$(x, ?), (y, 1), (z, 2)$
3	$(x, ?), (y, 1), (z, 2), (z, 4), (y, 5)$	$(x, ?), (y, 1), (z, 2), (z, 4), (y, 5)$
4		
5		
6		

Labeled Input Program

```
[y = x]1;  
[z = 1]2;  
while [(y > 0)]3 {  
    [z = z * y]4;  
    [y = y - 1]5;  
}  
[y = 0]6
```

RD Analysis

ℓ	$RD_{\text{entry}}(\ell)$	$RD_{\text{exit}}(\ell)$
1	$(x, ?), (y, ?), (z, ?)$	$(x, ?), (y, 1), (z, ?)$
2	$(x, ?), (y, 1), (z, ?)$	$(x, ?), (y, 1), (z, 2)$
3	$(x, ?), (y, 1), (z, 2)(z, 4), (y, 5)$	$(x, ?), (y, 1), (z, 2), (z, 4), (y, 5)$
4	$(x, ?), (y, 1), (z, 2)(z, 4), (y, 5)$	$(x, ?), (y, 1), (z, 4), (y, 5)$
5		
6		

Labeled Input Program

```
[y = x]1;  
[z = 1]2;  
while [(y > 0)]3 {  
    [z = z * y]4;  
    [y = y - 1]5;  
}  
[y = 0]6
```

RD Analysis

ℓ	$RD_{\text{entry}}(\ell)$	$RD_{\text{exit}}(\ell)$
1	$(x, ?), (y, ?), (z, ?)$	$(x, ?), (y, 1), (z, ?)$
2	$(x, ?), (y, 1), (z, ?)$	$(x, ?), (y, 1), (z, 2)$
3	$(x, ?), (y, 1), (z, 2)(z, 4), (y, 5)$	$(x, ?), (y, 1), (z, 2), (z, 4), (y, 5)$
4	$(x, ?), (y, 1), (z, 2)(z, 4), (y, 5)$	$(x, ?), (y, 1), (z, 4), (y, 5)$
5	$(x, ?), (y, 1), (z, 4), (y, 5)$	$(x, ?), (y, 5), (z, 4)$
6		

Labeled Input Program

```

[y = x]1;
[z = 1]2;
while [(y > 0)]3 {
    [z = z * y]4;
    [y = y - 1]5;
}
[y = 0]6
    
```

RD Analysis

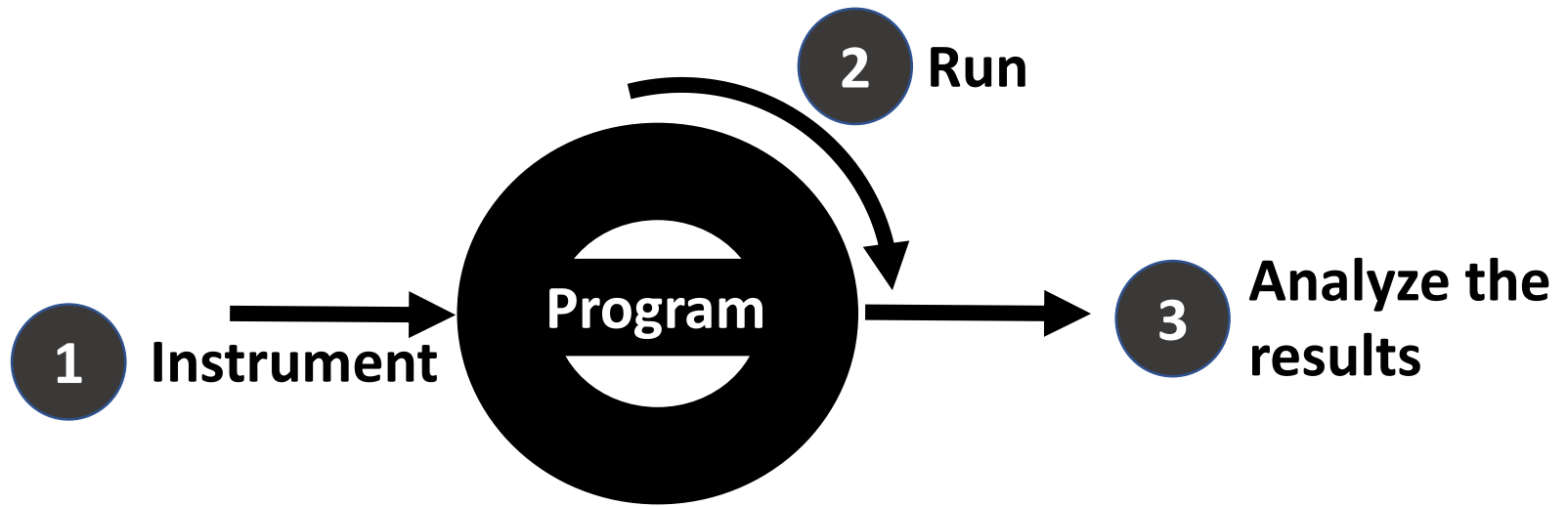
ℓ	$RD_{\text{entry}}(\ell)$	$RD_{\text{exit}}(\ell)$
1	$(x, ?), (y, ?), (z, ?)$	$(x, ?), (y, 1), (z, ?)$
2	$(x, ?), (y, 1), (z, ?)$	$(x, ?), (y, 1), (z, 2)$
3	$(x, ?), (y, 1), (z, 2)(z, 4), (y, 5)$	$(x, ?), (y, 1), (z, 2), (z, 4), (y, 5)$
4	$(x, ?), (y, 1), (z, 2)(z, 4), (y, 5)$	$(x, ?), (y, 1), (z, 4), (y, 5)$
5	$(x, ?), (y, 1), (z, 4), (y, 5)$	$(x, ?), (y, 5), (z, 4)$
6	$(x, ?), (y, 1), (z, 2), (z, 4), (y, 5)$	$(x, ?), (y, 6), (z, 2), (z, 4)$

This table answers all RD questions.

Labeled Input Program

```
[y = x]1;  
[z = 1]2;  
while [(y > 0)]3 {  
    [z = z * y]4;  
    [y = y - 1]5;  
}  
[y = 0]6
```

Dynamic Analysis



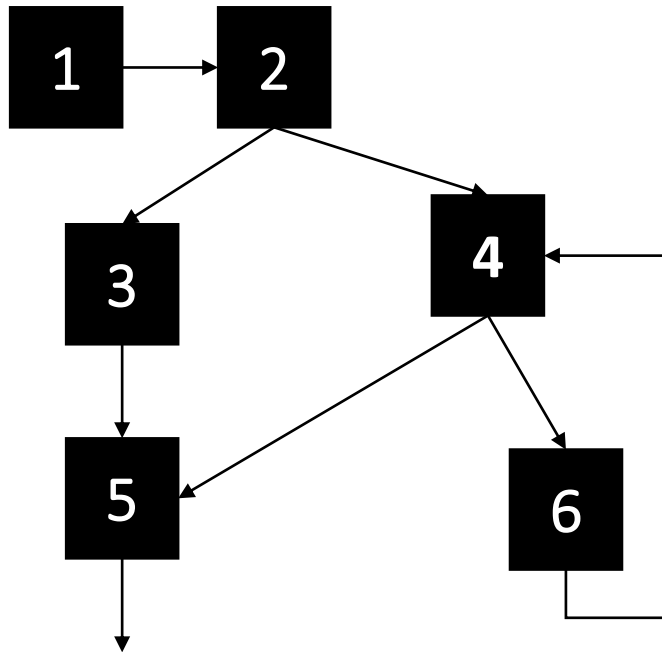
In this course, we will learn...

Introduction to
Runtime
Verification

Monitor Oriented
Programming
(Using JavaMOP)

Example: Path Profiling

- Count the paths taken during actual execution
 - consider them for optimization, distribution (with better hardware support) and test coverage



Path	Frequency
1 2 3 5	100
1 2 4 6 4 5	2000
1 2 4 6 4 6 4 5	10
1 2 4 5	10

Hybrid Analysis

- Often, we hit limitations with pure static or dynamic analysis.
- Hybrid = Static + Dynamic

In this course, we will learn...

Symbolic Execution and
Concolic Execution
(Overview of KLEE and
JavaPathFinder if time
permits)

Course Overview

**Program
Representation**
(SSA, TAC, AST,
CFG, PDG, IR...)

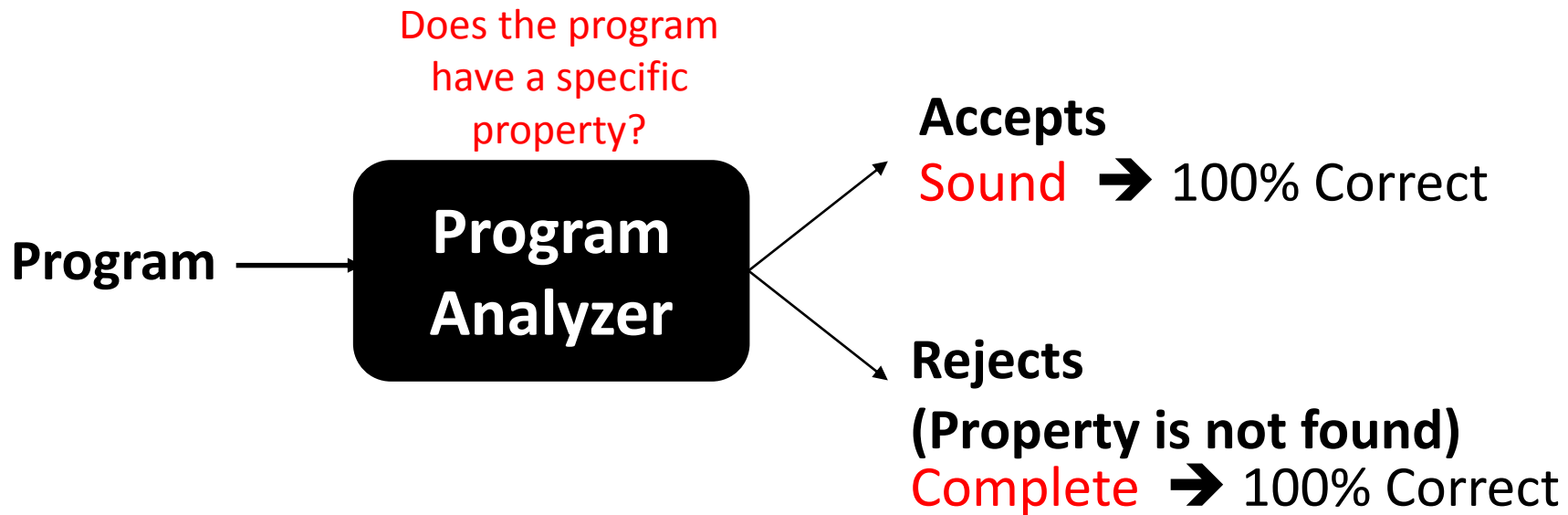
Classic Analyses
(Static - Data
Flow, Pointer, ...,
Dynamic, Hybrid)

**Modern
Analyses**
(Program Slicing,
Security
Automata, ...)

Tools
(Eclipse JDT, Soot,
JavaMOP,
AspectJ, ...)

Is Our Analysis Good?

Soundness and Completeness



May not halt as well! The **Don't Know** result.

Safe Approximation and Precision

```
if (...) {  
    x = 2;  
} else {  
    x = 3;  
}
```



$x = \{2,3\}$

i.e., x may be 2 or 3

Imprecise but safe.

```
if (...) {  
    x = 2;  
} else {  
    x = 3;  
}
```



$x = \{2,3,4\}$

i.e., x may be 2, 3 or 4

Much more Imprecise but still safe.

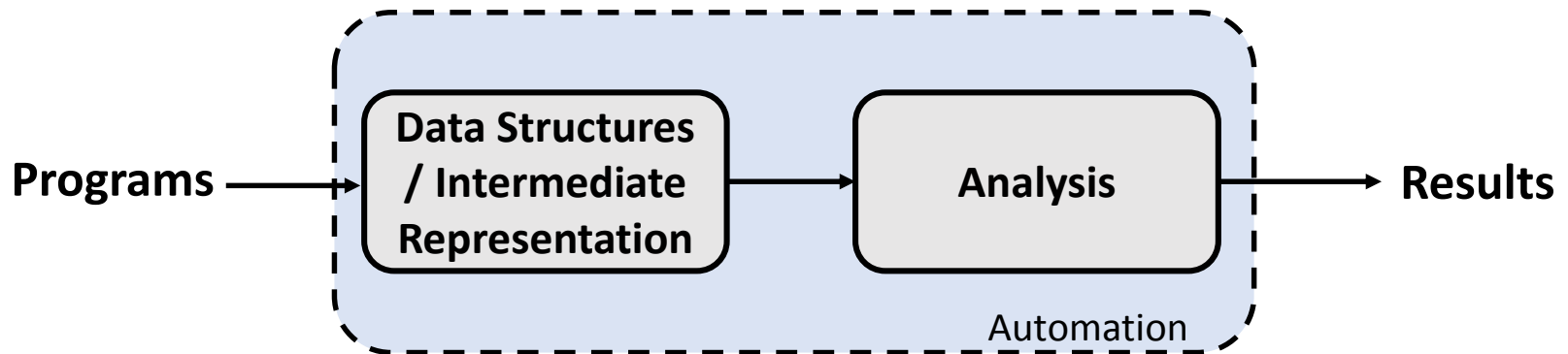
Analysis Quality

- Soundness
- Completeness
- Scalability
- Applicability
- ...

Note: Many other terminologies can be found in the literature such as Correctness and Completeness, Precision and Recall, Safety and Precision, ...

Summary

- What is Program Analysis?



- Why Study Program Analysis?
 - Analyzing programs is important for various reasons including optimization and verification.
- In this Course
 - **Static**, Dynamic and Hybrid Analyses