Lecture 1: Introduction, UML and OO basics

Overview

- Research area: software engineering
  - Software model checking
  - Analysis of application security
- In a nutshell:

  Complex Software System

  Does this software have ... correctness property?

  yes

  no

Very expensive, error-prone
Lecture 1: Introduction, UML and OO basics

Overview

- Research area: software engineering
  - Software model checking
  - Analysis of application security
- In a nutshell:

```
Complex Software System
```

Does this software have ... correctness property?

```
Automated checker
```

- yes
- maybe
- no

- Cannot always be fully automated
- Potential for inconclusive results

Challenges in model checking

- High complexity of the analysis
- Have to trade-off precision for complexity

```
Automated checker
```

analyze

state model of the system
Analysis of application security

- Distributed programs consist of a number of components
  - Components reside on different machines
  - A machine can be compromised
  - A machine may not be trusted
- Distributed software has to be designed in a way to protect other components if one is compromised

Analysis of visibility and modifiability of information in a component

A component (e.g. class)

```java
private long password
public long getPassword()
{
    return this.password;
}
```

Value of password escapes the component

- Compute escape information for Java classes
  - Fields values of (or references to) which can be obtained by other classes
- Compute mutability information
  - Whether values of fields may be modified via public method calls
Lecture 1: Introduction, UML and OO basics

Analysis of access control policies

- Security policy
  - role Student: getGrades
  - role Professor: add

- Remote component
  - private List grades;
  - public void add(Grade);
  - public void modify(Grade);

- Compute dependency information for methods and fields of a class
  - Check if methods assigned to different roles access the same fields in similar ways
  - Allow specifying access control policies in terms of fields

Other

- Automated tools for protection of intellectual property on software
  - Obfuscation
  - Watermarking
  - Tamper-proofing
- Combining testing and static analysis
- Parallel implementations of analysis algorithms