CE/CS/SE 3354 Software Engineering

Static bug detection

Static bug detection

- Compared to testing
  - Work for specific kinds of bugs ✗
  - Sometimes not scalable ✗
  - Generate false positives ✗

  - Easy to start (no build, no setup, no install …) ✔
  - Sometimes can guarantee the software to be free of certain kinds of bugs ✔
  - No need for debugging ✔

State-of-art: static bug detection

- Important type of bugs
  - Null pointer, memory leak, unsafe cast, injection, buffer overflow, racing, deadlock, dead loop, html error, …
- A large bunch of techniques for each kind of bugs
- Some of them have limitations preventing them from practical usage

Specification

- A description of the correct behavior of software
- We discuss two main kinds of specifications
  - Value
  - Data Flow

Value Specification

- The value (s) of one or several variable (s) must satisfy a certain constraint
- Example:
  - Final Exam Score <= 100
  - sortedlist(0) >= sortedlist(1)
  - http_url.startsWith(“http”)

Data Flow Specification

- Data from a certain source must / must not flow to a certain sink
- Example:
  - ! Contact Info -> Internet
  - Password -> encryption -> Internet

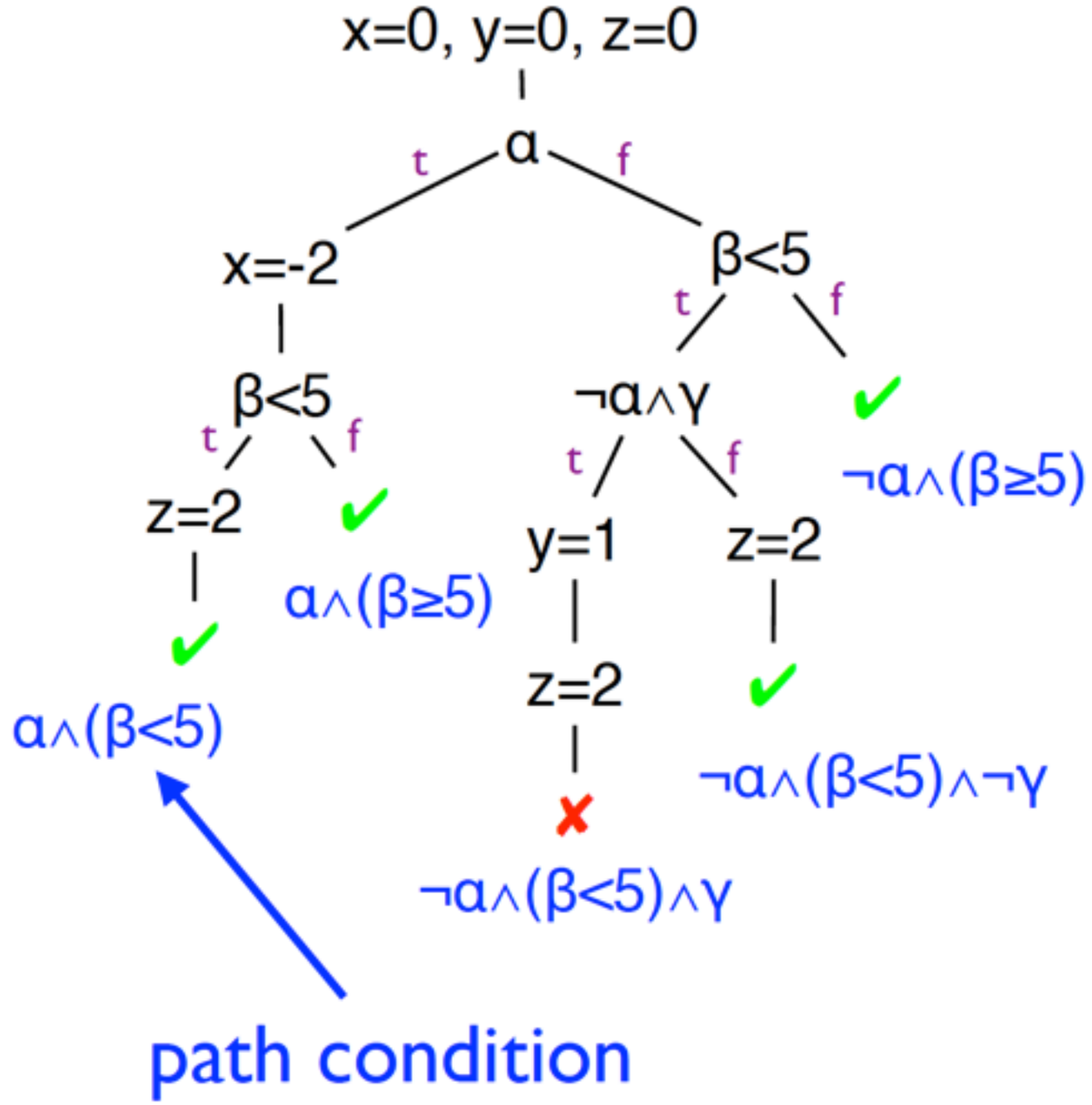
- Data Flow Specification are mainly for security usage

Checking Specifications
Basic ways

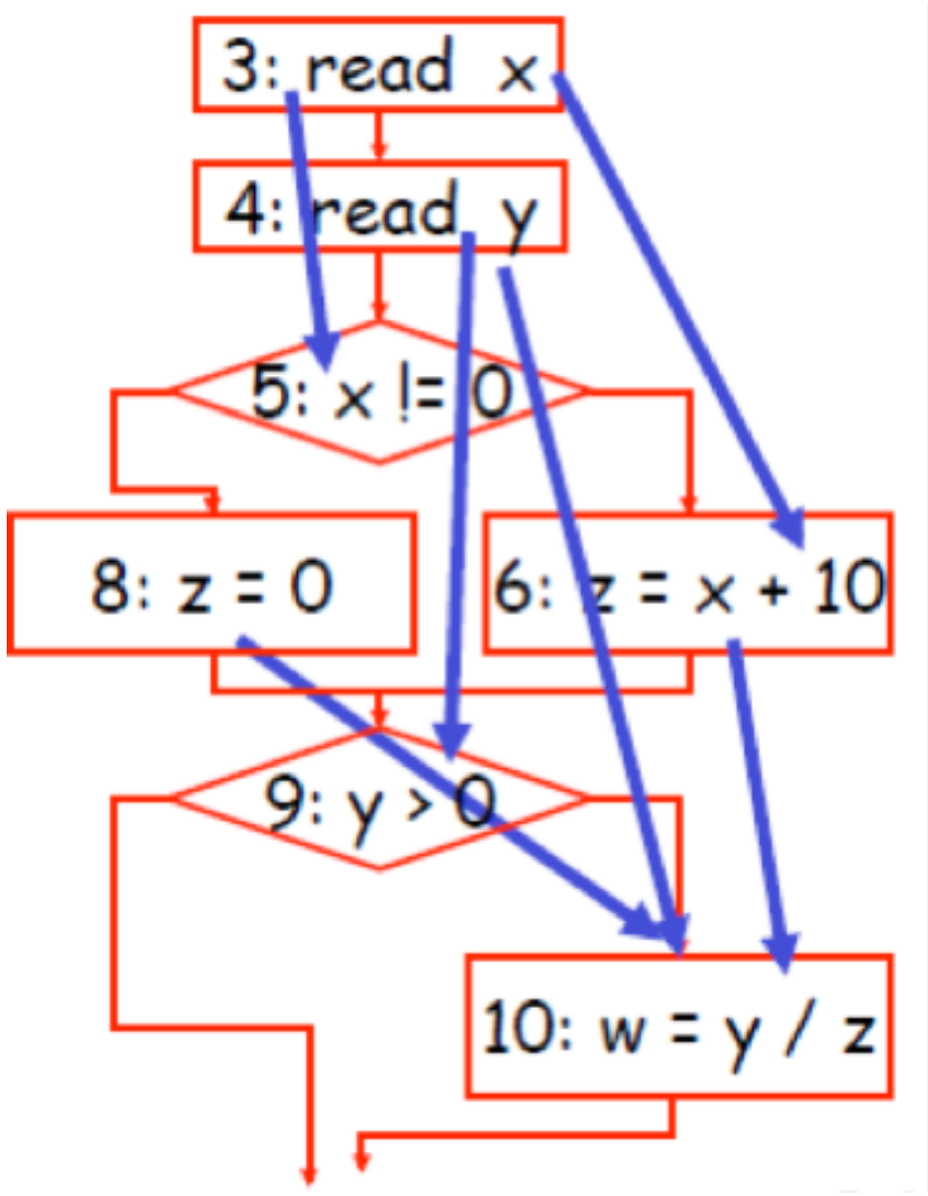
- Value Specifications
  - Symbolic evaluation
- Data Flow Specification
  - Graph traversal (Data Dependence Graph)

Symbolic Execution Example

```
a = α, b = β, c = γ;
                //symbolic
int x = 0, y = 0, z = 0;
if(a) {
  x = -2;
}
if(b<5) {
  if(!a && c) {y = 1;}
  z = 2;
}
assert(x+y+z!=3)
```



Some Simple check with Graph Traversal



Check (z is written to 0) ^ (z used as divider)

Problems of static bug detection

- False Positives vs. Efficiency
  - More precise -> higher cost

static bug detection tools

- Findbugs
  - A tool developed by researchers from UMD
  - Widely used in industry for code checking before commit
  - The idea actually comes from Lint
- Lint
  - A code style enforcing tool for C language
  - Find bad coding styles and raise warnings

    Bad naming

    Hard coded strings

FindBugs

- License: Lesser GNU Public License (LGPL)
- Author: The University of Maryland
- Statistics: downloaded more than a million times
- Homepage: http://findbugs.sourceforge.net/

Patterns to be checked

- >400 bug patterns
  - Bad Practice
  - Correctness
  - Performance
- Examples (http://findbugs.sourceforge.net/bugDescriptions.html)
  - Equals method should not assume type of object argument
  - Impossible cast
  - Should not use string.tostring()
  - ...

FindBugs: Pros and Cons

- Disadvantages
  - Cannot guarantee the software to be free of certain bugs
- Advantages
  - Easy to start
  - Relatively efficient
- Becomes the most popular and practical static bug detection technique
  - Findbugs helped Google find thousands of bugs