Automated Student Code Assessment with Test Generation

Karl Bell
Advisor: John Clements

Outline

- The Problem
- Solutions
- Background
- My Solution
- Validation

The Problem

How can an instructor test student submissions for correctness?

The Solutions

- Output comparison?

The Solutions

- Output comparison?
- Unit tests?

The Solutions

- Output comparison?
- Unit tests?
- Student tests?

My Solution

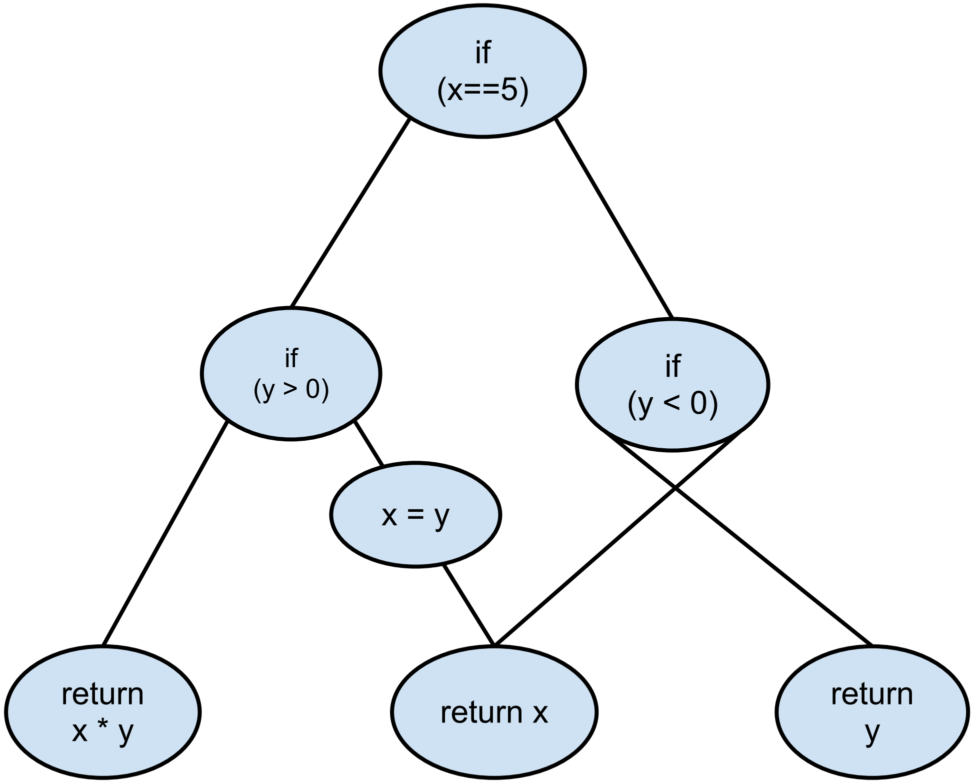
Automatically generate tests!

Background

- Symbolic Execution
- Concolic Testing
- Java PathFinder
- WebIDE

Symbolic Execution

```
if (x == 5) {
  if (y > 0) {
    return x * y;
  } else {
    x = y;
  }
} else if (y < 0) {
  return y;
}
return x;
```



Concolic Testing

1. Give the program an input

2. Gather symbolic constraints

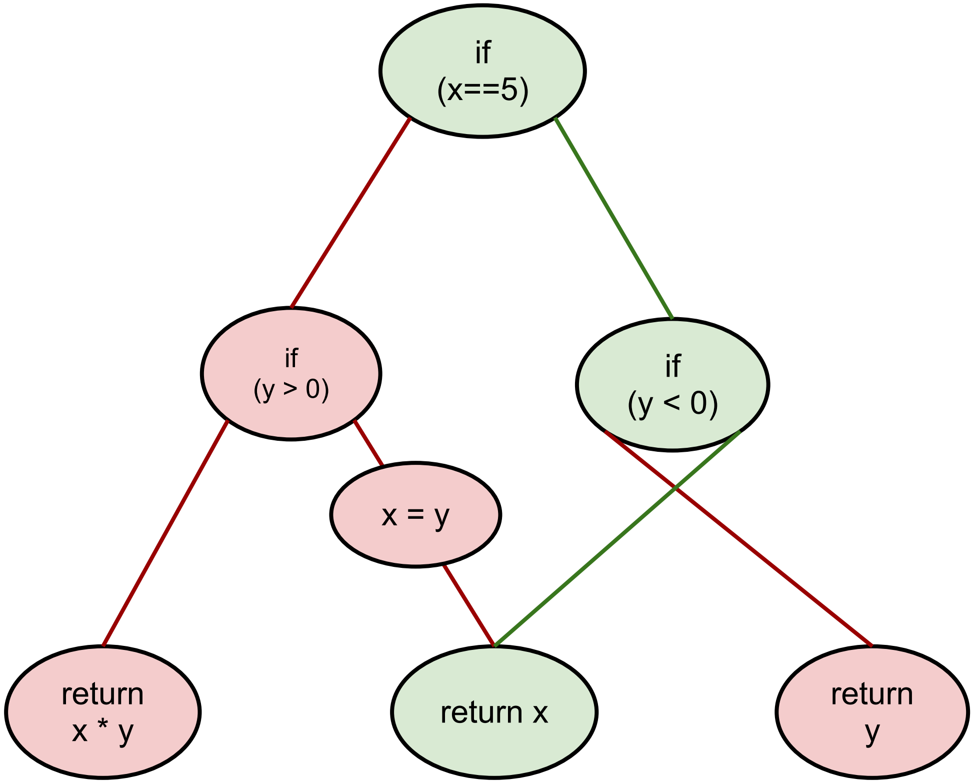
3. Falsify last constraint

4. Repeat

Path Constraints

```
x = 0, y = 0

if (x == 5) {
  if (y > 0) {
    return x * y;
  } else {
    x = y;
  }
} else if (y < 0) {
  return y;
}
return x;
```

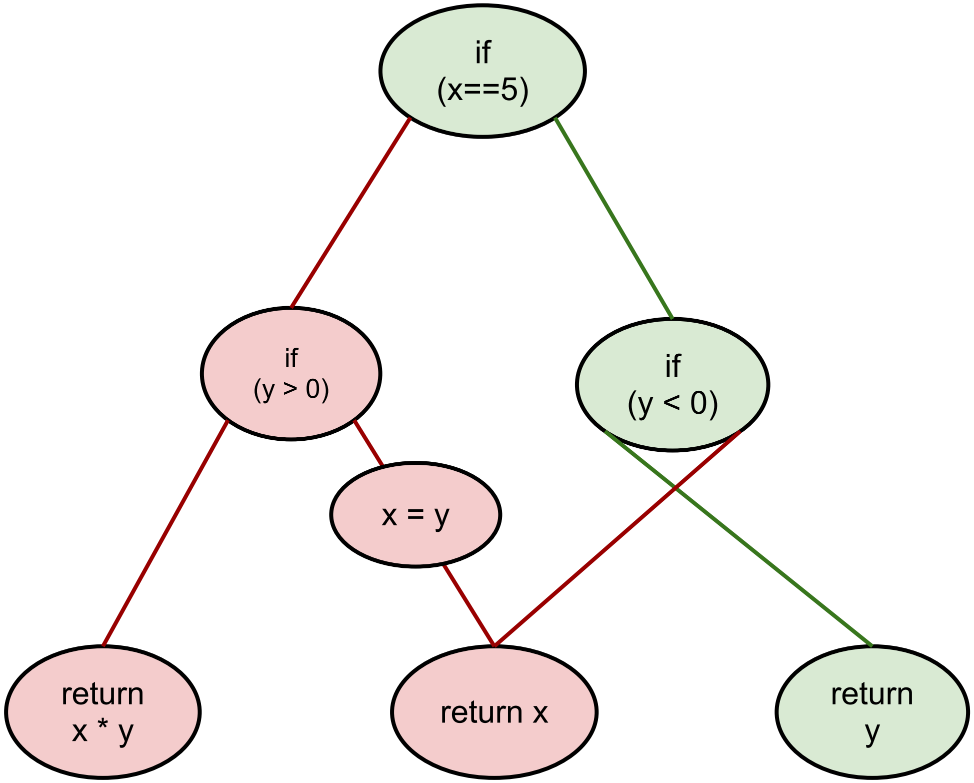


PC: (x != 5, y >= 0), ret == x

Path Constraints

```
x = 0, y = -2

if (x == 5) {
  if (y > 0) {
    return x * y;
  } else {
    x = y;
  }
} else if (y < 0) {
  return y;
}
return x;
```

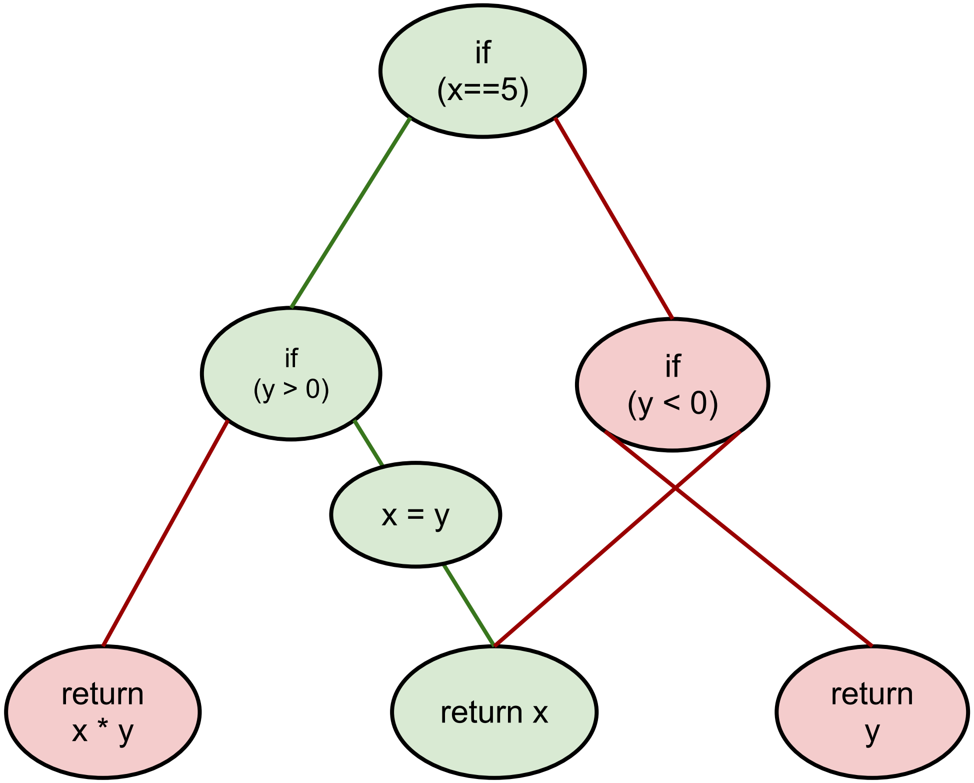


PC: (x != 5, y < 0), ret == y

Path Constraints

```
x = 5, y = -2

if (x == 5) {
  if (y > 0) {
    return x * y;
  } else {
    x = y;
  }
} else if (y < 0) {
  return y;
}
return x;
```



PC: (x == 5, y <= 0), ret == y

Java PathFinder

- Open-source NASA tool

Java PathFinder

- Open-source NASA tool
- Program verification

Java PathFinder

- Open-source NASA tool
- Program verification
- Bytecode instrumentation

Java PathFinder

- Open-source NASA tool
- Program verification
- Bytecode instrumentation
- Symbolic/Concolic test generation

WebIDE

- Developed here

WebIDE

- Developed here
- Guided learning labs

WebIDE

- Developed here
- Guided learning labs
- Driven by evaluation of student code

WebIDE

- Developed here
- Guided learning labs
- Driven by evaluation of student code
- But what if we didn't have to write the tests?

My Solution

- Instructor provides interface and reference implementation

My Solution

- Instructor provides interface and reference implementation

- Student submits attempt

My Solution

- Instructor provides interface and reference implementation
- Student submits attempt
- Generate tests based on student submission
    - Reference implementation as a test oracle

My Solution

- Instructor provides interface and reference implementation
- Student submits attempt
- Generate tests based on student submission
  - Reference implementation as a test oracle
- Provide feedback to student

Validation

- Code Coverage

Validation

- Code Coverage

- Evaluation of Code Correctness

Validation

- Code Coverage

- Evaluation of Code Correctness

- Helpfulness of Feedback

Questions?