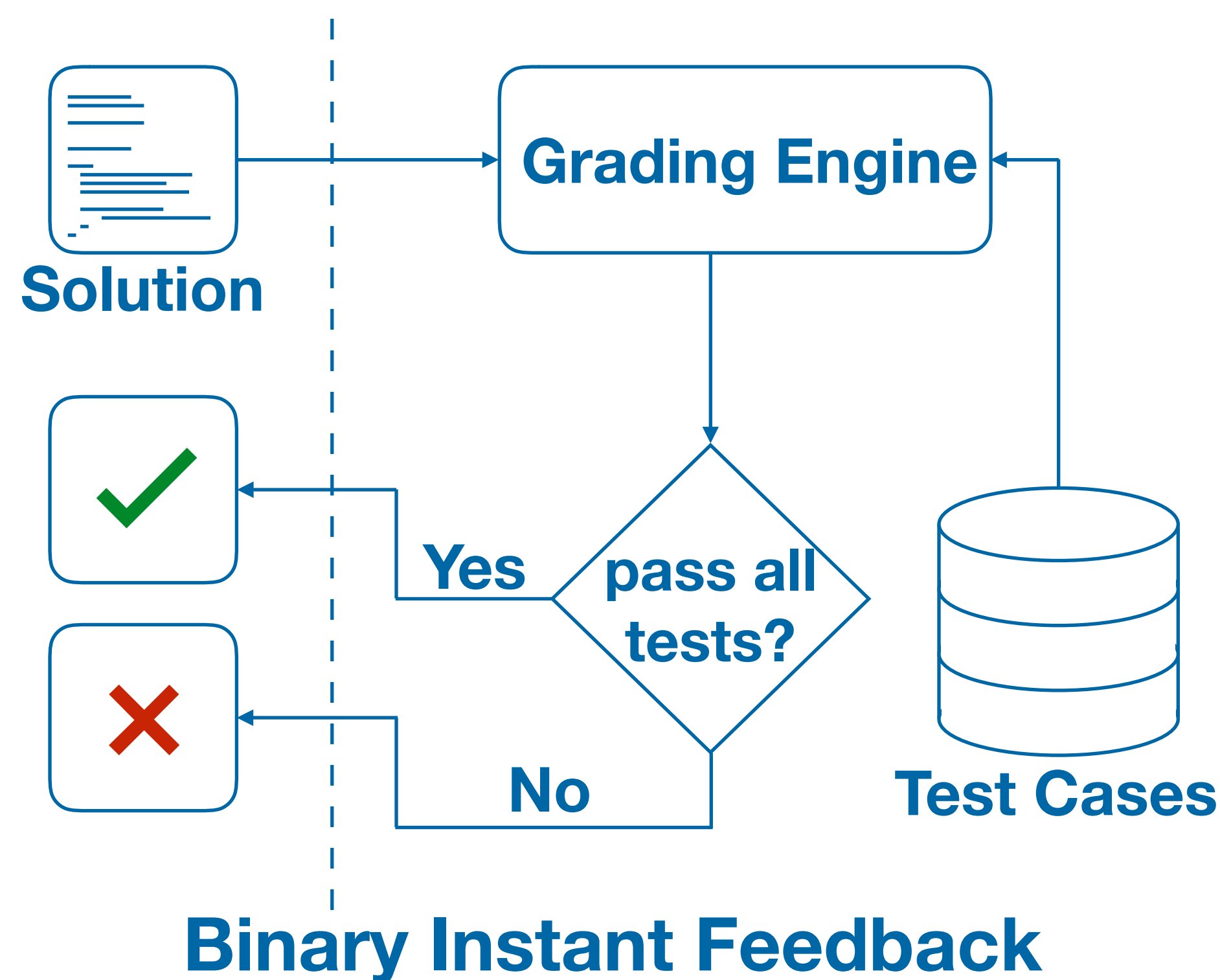


# Using Case-Based Reasoning to Automatically Generate High-Quality Feedback for Programming Exercises

Angelo Kyrilov

## Automated Assessment of Programming Exercises

### Typical Approach



### Problems with Binary Instant Feedback

- Increased **disengagement** rates
  - Lower exercise completion rates
  - Fewer exercises attempted
- Increased **plagiarism** rates
- Binary feedback offers **no guidance**
- Students see **no alternative**

## Case-Based Reasoning and High-Quality Feedback

### What is Case-Based Reasoning?

#### Retrieve

Find past cases **similar** to query

#### Reuse

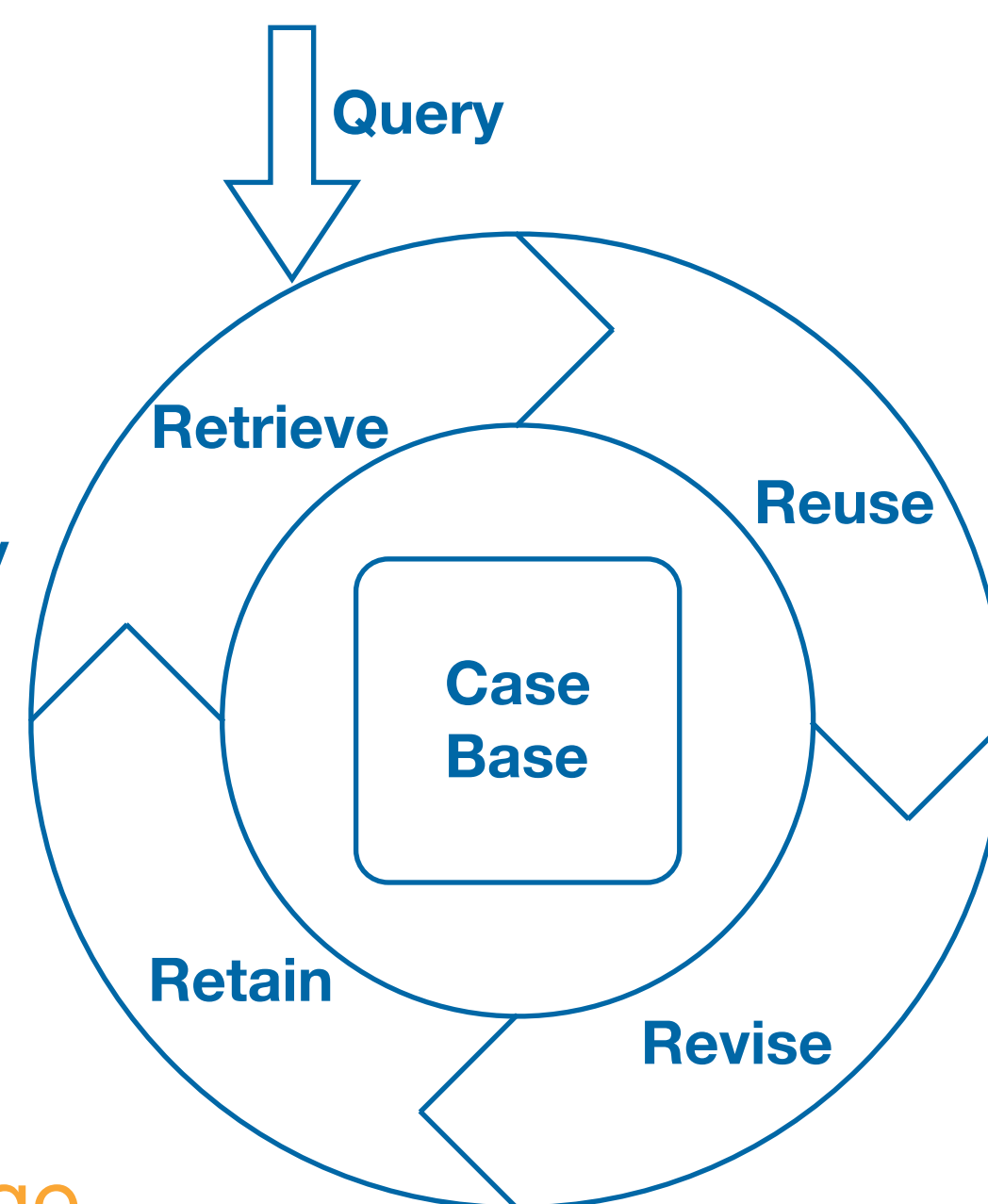
Use retrieved cases to **solve** query

#### Revise

Evaluate and **modify** solution

#### Retain

Store experience as **new knowledge**



### Proposed Feedback-Generation System

- Query: An **incorrect** student solution
- Case: Previous incorrect solution paired with **instructor-generated** feedback
- Case Base: Collection of cases
- Retrieval: **Incorrectness Similarity** metric

## Incorrectness Similarity and Applications

### Definition

Given two **incorrect** solutions to an exercise, say  $S_1$  and  $S_2$ ,  
If feedback for  $S_1$  is also **appropriate** for  $S_2$ , then  
 $S_1$  is **similarly incorrect** to  $S_2$

### Example

```
int fact(int n) {
  int result = 0;
  for (int i = 1; i <=n; i++) {
    result *= i;
  }
  return result;
}

ERROR

int fact(int n){
  if (n == 0) {
    return 0;
  }
  else {
    return n * factorial(n - 1);
  }
}
```

“What is `fact(0)`?” is **appropriate** for both solutions

### Possible Applications

- MOOCS
  - Provide **high-quality** feedback to **hundreds of thousands** of students
  - Relatively **little** instructor effort

### Future Work

- New methods for incorrectness similarity
- Testing effectiveness of system