Automatic Testing of Software Libraries

Marko Dimjašević

Supervisor: Zvonimir Rakamarić

October 21, 2013
Take-Home Message

- We developed a software testing technique that combines random and concolic testing approaches
Main Goal

- Improve code coverage in automatic software testing

```java
public int indexOf(Object o) {
    int index = 0;
    if (o == null) {
        for (Entry e = header.next; e != header; e = e.next) {
            if (e.element == null)
                return index;
            index++;
        }
    } else {
        for (Entry e = header.next; e != header; e = e.next) {
            if (o.equals(e.element))
                return index;
            index++;
        }
    }
    return -1;
}
```

- Preliminary results for a network library:
  - Instruction coverage: from 31% to 42%
  - Branch coverage: from 14% to 23%
Combination of Two Approaches

- Feedback-directed random testing
  - Used for a global search
  - Randoop
- Concolic testing
  - Used for a local search
  - Java PathFinder’s jDART
- Implementation: JPF-Doop

Diagram:

- Random values generator
- Random input values
- Random testing - Randoop
- Unit tests
- Refine unit tests?
  - No
  - Code coverage report
  - Concolic execution - JDart
  - Yes
Conclusion

- Multipronged technique for improving code coverage in automatic software testing
- Interested in details? Let’s talk during the poster session!