

# 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

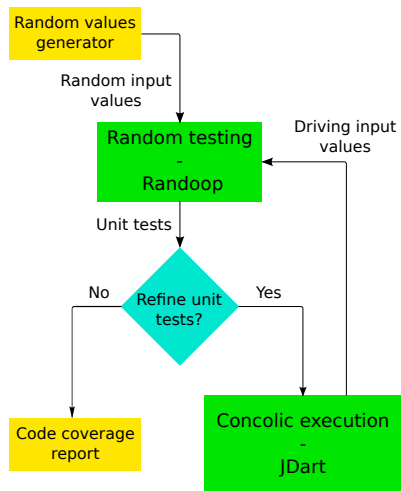
- ▶ Improve code coverage in automatic software testing

```
411.     public int indexOf(Object o) {
412.         int index = 0;
413.         ◆ if (o==null) {
414.             ◆ for (Entry e = header.next; e != header; e = e.next) {
415.                 ◆ if (e.element==null)
416.                     return index;
417.                 index++;
418.             }
419.         } else {
420.             ◆ for (Entry e = header.next; e != header; e = e.next) {
421.                 ◆ if (o.equals(e.element))
422.                     return index;
423.                 index++;
424.             }
425.         }
426.         return -1;
427.     }
```

- ▶ 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



# Conclusion

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