ICA: Formal Verification

ICA - Overview

Goal: Formally verify control flow behavior in servlets of a web application

1. Build control flow graphs of servlets
2. Reason about ordering of invocations

In class on Wednesday 11/11.

CSCI 512 Formal Verification

Ask path-based questions:

- Model <- Control flow graph
- Properties <- Provided in class, based on invocations
- Model checking <- graph reachability

ICA - Details

1. Finish implementation of control flow graph builder
2. Test it on example and your own test cases
3. Be able to automatically iterate over the generated graphs
4. Verify reachability properties over the graph

ICA – Hints and Tips

1. Adding labels to your graph's edges may help readability
2. Don't forget about select statements
3. Use the Instruction API, don't parse strings
4. Only analyze the body of one method at a time
5. No need to handle exceptions
6. Create your own Node and Edge classes for easy traversal and calculating reachability

Blank

Implement

1. Generate control-flow graph for the class in string or dotty format
2. For each question
   - Give the answer (e.g., “Yes” or “No”)
   - Write a **brief** explanation of how you used the tool or results to calculate the answer

Grading

100 total points

- 20 points for the correct CFG
- 10 points for the quiz
- 5 questions at 14 points each
  - 4 points for automation
  - 3 points for explanation
  - 3 points for correctness
  - 4 points in-class bonus

Submit

Submit the following:

1. CFG builder code and any other automated tools built
2. Answers/explanation for each question
3. Control-flow graph in string or dotty format

Checkoff of code by 11/24 EOH

Instructions:

- Only one zipped file with name format "Last name_first name_ICA#“
- Submit all reports in one submission. TA will only grade the last submission.

Questions

1. Does MJ reach MC?
2. Does MB reach ME?
3. Does MA reach MG?
4. Does MI reach MF?
5. Is a call to MD() reachable from a different call to MD? Itself?