15-819 M: Program Analysis

Jonathan Aldrich

Assignment 4 (Programming): Alias or Interprocedural Analysis

Due Monday, March 1, 1:30pm

Turn in a zip file named electronically in the Blackboard drop box. The zip file should contain the following files:

overview.xxx (in txt, pdf, or Word (doc/docx) format). At the top of the document, state your name and Andrew id.
code/ (zipped-up source code with your analysis implementation)
test/ (test cases for your analysis)
MyAnalysisTestOutput.xxx (either graphical or text output for the test case(s))

200 points

Assignment objectives:

- Implement a larger-scale, more realistic program analysis
- Solidify and demonstrate understanding of the course material on alias and/or interprocedural analysis

Analysis Implementation

Choose any alias analysis or interprocedural analysis discussed in class or described in the literature. Any interprocedural analysis in which the analysis of one function affects the analysis of another is OK; this includes any context-sensitive analysis but also analyses based on an interprocedural call graph. Any nontrivial alias analysis is OK; flow-sensitive alias analyses may be implemented intra-procedurally, while flow-insensitive analyses like Steensgaard's and Andersen's analysis should be interprocedural. You may develop experimental adaptations or extensions of the analysis if you wish, but coordinate with the instructor in this case.

Implement your analysis in Crystal or some other program analysis infrastructure. It is OK to handle a subset of features in your target language as long as you define what they are, but if you choose alias analysis you must handle fields in some way. Write appropriate test cases to ensure your analysis is working properly.

Write an overview of what you did. Describe what analysis you implemented, and if it differs from or extends the description in class or in the research literature, describe those differences. If you chose interprocedural analysis explain what context-sensitivity strategy you used, if any. Describe any limitations of your implementation, e.g. what language features are not fully supported. Describe how to compile your analysis and how to run the tests, in enough detail that the instructor should be able to compile and test it without contacting you.