Goals: learn about

- Foundations of static program analysis
- Approaches for formulating analyses
- Approaches to inter-procedural analysis
- Implementation of analyses for Java via SOOT

Topics:

- *Survey of Data Flow Analyses:* Descriptions of a wide-variety of program analyses used in optimization, specialization, ...
- *Data Flow Frameworks:* A style of mathematical framework for defining a data flow analysis.
- *Constraint-based Analysis:* A style of defining (and implementing) flow analyses that is based on expressing direct relationships between syntactic program entities.
- *Inter-procedural Analysis:* Approaches for analyzing programs that exploit the procedural structure to improve the speed of analysis and its precision.

Topics (continued):

- *Survey of Java Analyses:* Descriptions of analyses that have been developed directly for Java or have been customized for Java.
- *Implementation Techniques:* ranging from basic iterative algorithms to large systems designed to solve analysis related problems.
- *Applications:* Ways that analysis results can be used, for example, to transform the program to *improve* it in some way.
- *Relationship to Model Checking:* Data flow analysis is model checking of abstract interpretations, to quote Dave Schmidt, we'll see why that is so.

The *Twist*:

Traditional courses in optimization look at improving a programs

- *Space:* code distribution, embedded systems
- *Speed:* high-performance applications
- *Predictability:* real-time

One could also look at analysis results as an end

- *Anomaly detection:* deadlocks, race conditions
- *Test adequacy:* def-use coverage
- *Style guides:* potentially uninitialized data

We will *optimize* programs for

- *State-Space Reduction:* improve the performance of subsequent model checks

Schedule:

- *Lectures:* 3 hours/week. MW 3:30-4:45pm

Prerequisites:

- CIS 606/706, solid Java programming skills

Marking Scheme:

- 30% final exam
- 40% project presentations
- 30% final project report

Lecturer:

- Professor Matthew Dwyer, Nichols 324, Office Hours (by appt)
- Professor John Hatcliff, Nichols 324, Office Hours (by appt)

Course material:

- textbook: Principles of Program Analysis, Neilsen, Neilsen & Hankin
- other books: Advanced Compiler Design and Implementation, Muchnick
- papers (I'll distribute copies)
- slides for the lectures; and
- tool documentation linked off course WWW pages.

Analysis Frameworks:

- PAG : analyzer generator
- SOOT : Java compiler environment
- BANE : constraint solution engine
- BANDERA : for applying analysis results

Projects:

- infrastructure (starts now)
- escape analysis
- side-effects analysis
- safe lock analysis
- copy propogation and constant folding
- byte-code reordering
- others are possible