CSE 501: Implementation of Programming Languages

Instructor: Craig Chambers

- office hours: Tu 12-1, Th 11-12 (Sieg 309)

TAs: Dave Grove

- office hours: W 10-11 (Chateau 110)

Jon Nowitz

- office hours: M 11-12 (Sieg 233)

Goals:

- Understand how languages get implemented efficiently
- Understand what can and can't be done, with how much effort
- Understand state of current research in efficient language implementation

Prerequisites:

- CSE 401 or equivalent
- CSE 505 or equivalent

Readings:

- Appel's *Modern Compiler Implementation*
- important papers from the literature

Grading:

- Midterm: 25%
- Final: 25%
- Homework: 25%
- Project: 25%

Course Outline

Week	Topic
1/5	Intro; structure of compilers; standard optimizations Standard intermediate representations; control flow, data flow; dependence Dataflow analysis; reaching constants, constant propagation
1/12	Lattice-theoretic data flow analysis framework; integer range analysis Data flow analyzer generators, frameworks
1/19	***Holiday*** Advanced intermediate representations: def/use chains, control dependence tree, SSA form, VDG CSE; loop-invariant code motion
1/26	Inlining Interprocedural analysis; analysis with first-class functions, dynamically dispatched messages
2/2	Procedure specialization, partial evaluation Alias and pointer analysis
2/9	Dependence analysis; automatic parallelization
2/16	***Holiday*** Global register allocation Calling conventions; callee-save vs. caller-save
2/23	Link-time register allocation Instruction scheduling
3/2	Garbage collection
3/9	Implementing functional and object-oriented languages