CISC 672: ADVANCED COMPILER CONSTRUCTION
Fall 2010 Midterm Exam Study Guide
In-class individual exam

References
- Lectures notes and slides from start of course through October 19, 2010.
- Readings listed on schedule through October 19, 2010.
- Assignments PA1 (cool programming), PA2 (scanner) and PA3 (parser).
- Handouts from class
- Parsing homework
- Quizzes

Topic Coverage
- overall compiler and compiler-related tool constructions.
- lexical specification: regular expressions - reading and writing.
- implementation of a lexical analyzer: manually and using JLex.
- dfa and nfa construction from regular expressions.
- error detection and recovery in lexical analysis.
- syntax specification: context free grammars.
- problems with grammars: ambiguity.
- grammar rewriting to attempt to remove ambiguity.
- top-down parsing: getting the grammar in the right form (left factoring, eliminating left recursion, recursive-descent parsing, FIRST and FOLLOW computation, LL(1) parsing method and and parse table construction, determining whether a grammar is LL(1).
- bottom-up parsing: issues in parsing, shift-reduce parsing method, use of yacc/JavaCup, LR(0), SLR(1), LR(1), LALR(1) DFA and parse table construction, determining whether a grammar is one of these classes.
- symbol tables and their design, construction and use
- static semantic checking
- inheritance graphs

Format of Exam
The exam is closed book, closed neighbor and you will have the full class period to work. In general, the exam will be a combination of testing your basic knowledge and understanding of the concepts covered in class and application of the concepts. The questions will most likely be of the form:
- Short answer.
- Writing regular expressions.
- Drawing DFA's/NFA's.
- Understanding of a flex-like specification.
- Writing context-free grammars.
- Rewriting context-free grammars.
- Identifying problems in context-free grammars.
- Deriving strings and constructing parse trees.
- Top-down parsing methods (recursive-descent, LL(1)):
(writing part of a recursive-descent parser, constructing FIRST and FOLLOW sets, constructing an LL(1) parse table from given FIRST and FOLLOW sets, determining whether a grammar is LL(1).
- Bottom-up parsing: constructing LR(0) and LR(1) DFA's, constructing SLR(1), LR(1) parse tables from DFA's, issues in the shift-reduce parsing methods.
- symbol tables: organizing, implementing operations, tradeoffs

The questions are NOT multiple choice. Instead, partial credit will be given when possible on any question in the exam.

How to Study
Review your lecture notes, handouts, labs, and textbook chapters, parsing homework. Use the textbook as a resource for added understanding of the topics covered in class and on assignments.