CS 410 Compiler Construction

(formerly CS 236 or 230)
Syllabus
Fall Semester, 2002
19 August 2002 version

Class meetings
2:00-3:15 Tuesdays and Thursdays
room 107 MER Building (aka COMER Building)

Text
Alfred V. Aho, Ravi Sethi, Jeffrey D. Ullman
Compilers: Principles, Techniques, and Tools
Addison-Wesley

Instructor
Frances L. Van Scoy
fvanscoy@wvu.edu
304-293-LANEx4181
G11B CERC/CRRB (Concurrent Engineering Research Center / Chestnut Ridge Research Building 886 Chestnut Ridge Road)
Office hours: 7th floor student lounge ESB 1:00-1:45 Tuesdays and Thursdays
lobby of MER building 3:15-3:45 Thursdays
or by appointment

Course Objectives

describe Chomsky hierarchy, identify the class to which specific languages belong, convert language definitions to equivalent language- or automata-based definitions

use lex and yacc to implement translation software

describe the block diagram of a compiler and the major algorithms for each component

Grading

description	due date	grade %
lex exercise	September 24	10
yacc exercise	October 29	10
term project	November 21	25
oral presentation	December 3 and 5	5
2 tests	September 19, October 24	30 (15 each)
final exam	December 12	20

Class Meetings

	Date	Topic	Reading Assignment	Assignment Due
1.	August 20	Introduction to compilers		
2.	August 22	Introduction to formal language theory	1, 2	
3.	August 27	Scanner; left and right linear grammars	3	
4.	August 29	Finite state automata; regular expressions		
5.	September 3	Lex - part 1		
6.	September 5	Lex - part 2		
7.	September 10	Parser; context free grammars	4	
8.	September 12	Some parsing techniques	5	
9.	September 17	LL parsing		
10.	September 19	Test 1		
11.	September 24	LR parsing		lex assignment
12.	September 26	LR parsing		
13.	October 1	Semantics		
14.	October 3	Yacc - part 1		
15.	October 8	Yacc - part 2		
16.	October 10	Symbol table; type checking	6	
17.	October 15	Run-time environments	7	
18.	October 17	Intermediate code	8	
19.	October 22	Expressions and data structures; assignment statements		
20.	October 24	Test 2		
21.	October 29	Control structures		yacc assignment
22.	October 31	Procedures and functions		
	November 5	WVU closed for Election Day		
23.	November 7	Code generation	9	
24.	November 12	Local optimization	10	
25.	November 14	Global optimization		
26.	November 19	Error recovery		
27.	November 21	Practical issues	11, 12	term project
	November 26	WVU closed for Thanksgiving Break		
	November 28			
28.	December 3	Student oral presentations		
29.	December 5			
30.	December 12	8am Final Exam		

Reading assignments

You should plan on reading each chapter of the text 3 times:
skimming it before the lecture on the indicated day,
reading it carefully to determine whether you now understand it well within a day or two after the corresponding lecture,
reviewing it before the first test on the material

Also, you are being given a CD-ROM with old notes from CS 136 and 236. You should review old lecture notes and read the lex and yacc documentation on that CD. Other files from that CD may be specifically assigned as well.

One ore more additional reference CD-ROMs may be distributed during the semester and additional reading material may be posted to the course MIX web pages.

Yacc assignment

Using yacc (or ayacc or bison or similar),

implement a postfix notation calculator. Your calculator should include the arithmetic operators (addition, subtraction, multiplication, division, and exponentiation) and selected trigonometric functions (sine, cosine, and tangent). Your calculator should also provide 26 registers or variables, labeled A through Z, and an assignment operator.

or

construct a program which generates an English narrative from a GEDCOM data base. (Sample GEDCOMS for selected U.S. Presidents are on the CD-ROM.)

Term project

The recommended term project is a compiler which will translate VRML files to OpenGL or to programs using the GHOST library.

Other term project topics may be approved by discussion with the instruction.

Lex assignment

Use lex (or alex or flex or similar) to implement a scanner for the following language based vaguely on COBOL.

The language has seven "sentences":

INPUT THE VALUE OF []. or INPUT A VALUE OF [].
OUTPUT THE VALUE OF []. or OUTPUT A VALUE OF [].
ADD [] TO [] GIVING [].
SUBTRACT [] FROM [] GIVING [].
MULTIPLY [] BY [] GIVING [].
DIVIDE [] BY [] GIVING [].

REPEAT THIS PARAGRAPH [] TIMES.
BEGIN PARAGRAPH.
<>
END PARAGRAPH.
or
REPEAT THE FOLLOWING PARAGRAPH [] TIMES.
BEGIN PARAGRAPH.
<>
END PARAGRAPH.

where
<> is a list of one or more sentences
[] is an identifier or a numeric literal
an identifier begins with a letter,
and may contain letters, digits, and spaces
a numeric literal is
digits
. digits
digits .
digits . digits
and may be preceded by a + or -

Notice that every sentence ends with a period.

Produce three classes of tokens:
identifiers
numeric literals
operators.
Define as *few* operators as possible, within the limits of the power of finite state automata and the needs of the parser.

West Virginia University is committed to social justice.

If you are a person with a disability and anticipate needing any type of accommodation in order to participate in this class, please advise me and make appropriate arrangements with Disability Services (293-6700).

Plagiarism involves claiming as your own work the work done by another. Plagiarism can result in a variety of penalties including a failing grade in the course.