CE 160 Problem Sets
First Semester, 2004-2005

Exercise and page numbers are from the textbook unless indicated otherwise.
Study Chapter 1 as needed.

Problem Set 1.1.
(Study sections 2.1 and 2.2 on Deterministic Finite Automata)

1. Exercise 2.2.3, p. 54
2. Exercise 2.2.4 (b), p. 54
3. Exercise 2.2.4 (c), p. 54
4. Exercise 2.2.5 (a), p. 54
5. Exercise 2.2.5 (c), p. 54

Problem Set 1.2.
(Study sections 2.1 and 2.2 on Deterministic Finite Automata)

1. Exercise 2.2.5 (d), p. 54
2. Exercise 2.2.6 (b), p. 54
3. Exercise 2.2.7, p. 54
4. Exercise 2.2.8 (a), p. 54
5. Exercise 2.2.11, p. 55

Problem Set 1.3.
(Study section 3.1 on Regular Expressions)

1. Exercise 3.1.1 (b), p. 89
2. Exercise 3.1.1 (c), p. 89
3. Exercise 3.1.2 (b), p. 89
4. Exercise 3.1.3 (a), p. 90
5. Exercise 3.1.4 (c), p. 90

Problem Set 1.4.
(Study sections 5.1 and 5.2 on Context-Free Grammars and Parse Trees)

1. Exercise 5.1.1 (c), p. 179-180
2. Exercise 5.1.2, p. 180
3. Design a CFG for the set of all strings of balanced parentheses (a string of parentheses is "balanced" if each left parenthesis has a matching right parenthesis, and pairs of matching parentheses are properly nested).
4. Exercise 5.1.4 (b), p. 180
5. Design a CFG for the set of all palindromes over {0,1,2} that have odd length.

Problem Set 1.5.
(Study sections 5.1 and 5.2 on Context-Free Grammars and Parse Trees)

1. Exercise 5.1.7, p. 181
2. Exercise 5.1.8, p. 181
3. Exercise 5.2.2, p. 191
4. Exercise 5.2.3, p. 191
5. Exercise 5.2.4, p. 191

Problem Set 1.6.
(Study sections 6.1 and 6.2 on Pushdown Automata)

1. Exercise 6.1.1 (b), p. 228
2. Exercise 6.1.1 (c), p. 228
3. Design a PDA whose final state language is the set of all strings over {0,1} such that no prefix has more 1's than 0's.
4. Design a PDA whose empty stack language is the set of all strings over {0,1} such that no prefix has more 1's than 0's.
5. Design a PDA whose empty stack language is the set of all strings over {0,1} with an equal number of 0's and 1's.

Problem Set 1.7.
(Study sections 8.1 and 8.2 on Turing Machines)

1. Exercise 8.2.1 (b), p. 328
2. Exercise 8.2.1 (c), p. 328
3. Exercise 8.2.2 (b), p. 328
4. Exercise 8.2.2 (c), p. 328
5. Exercise 8.2.3, p. 328

Problem Set 1.8.
(Study sections 8.1 and 8.2 on Turing Machines)

1. Exercise 8.2.5 (b), p. 329
2. Exercise 8.2.5 (c), p. 329
3. Design a Turing Machine for the set of all strings of balanced parentheses (a string of parentheses is “balanced” if each left parenthesis has a matching right parenthesis, and pairs of matching parentheses are properly nested).
4. Design a Turing Machine for the set of all palindromes over {0,1} that have odd length.
5. Design a Turing Machine for the set of all strings over {0,1} such that no prefix has more 1's than 0's.

Problem Set 1.9.
(Study section 2.3 on Nondeterministic Finite Automata)

1. Exercise 2.3.2, p. 66
2. Exercise 2.3.3, p. 67
3. Exercise 2.3.4 (b), p. 67
4. Exercise 2.3.4 (c), p. 67
5. Design an NFA who language is the set of all strings over {0,1} such that the 3rd symbol from the end is a 1. Convert this NFA to a DFA.

Problem Set 1.10.
(Study section 2.5 on Finite Automata with Epsilon Transitions)

1. Exercise 5.1.4 (a), p. 180
2. Exercise 2.5.2, p. 80
3. Exercise 2.5.3 (a), p. 80
4. Exercise 2.5.3 (b), p. 80
5. Exercise 2.5.3 (c), p. 80

Problem Set 1.11.
(Study section 3.2 on Finite Automata and Regular Expressions)

1. Exercise 3.2.1 (c), p. 106
2. Exercise 3.2.3, p. 106
3. Exercise 3.2.4 (c), p. 106-107
4. Exercise 3.2.6 (c), p. 107
5. Exercise 3.2.6 (d), p. 107

Problem Set 1.12.
(Study section 3.4 on Algebraic Laws for Regular Expressions)

1. Exercise 3.4.2 (b), p. 121
2. Exercise 3.4.2 (d), p. 121
3. Exercise 3.4.2 (e), p. 121
4. Exercise 3.4.3, p. 121
5. Exercise 3.4.5, p. 122

Problem Set 1.13.
(Study section 4.4 on Equivalence and Minimization of Automata)

1. Exercise 4.4.2, p. 164-165
2. Draw the table of distinguishabilities of the following DFA and construct its minimum-state equivalent:

	0	1
→*A	B	D
B	E	C
*C	B	F
D	A	E
E	E	E
F	C	E

3. Draw the table of distinguishabilities of the following DFA and construct its minimum-state equivalent:

	0	1
→A	B	C
*B	D	E
*C	F	G
D	D	D
E	H	I
F	H	C
G	D	G
*H	D	E
*I	A	G

4. Draw the table of distinguishabilities of the following DFA and construct its minimum-state equivalent:

	0	1
→A	B	C
*B	D	E
C	E	F
D	B	G
*E	G	H
*F	H	J
G	E	I
*H	F	E
*I	H	G
J	E	F

5. Draw the table of distinguishabilities of the following DFA and construct its minimum-state equivalent:

	0	1
→A	B	A
*B	C	B
C	D	C
*D	E	D
E	F	E
*F	G	F
G	H	G
*H	I	H
I	J	I
*J	A	J

Problem Set 1.14.
(Study section 5.4 on Ambiguity in Grammars and Languages)

1. Exercise 5.4.2, p. 214
2. Exercise 5.4.5 (a), p. 214-215
3. Exercise 5.4.5 (b), p. 214-215
4. Exercise 5.4.7 (a), p. 215
5. Exercise 5.4.7 (b), p. 215

Problem Set 1.15.
(Study section 6.2 on The Languages of a PDA)

1. Exercise 6.2.3 (a), p. 236
2. Exercise 6.2.5, p. 236
3. Exercise 6.2.6 (a), p. 236-237
4. Exercise 6.2.6 (b), p. 236-237
5. Exercise 6.2.7, p. 237

Problem Set 1.16.
(Study section 6.3 on Equivalence of PDA's and CFG's)

1. Exercise 6.3.2, p. 245-246
2. Exercise 6.3.4, p. 246
3. Exercise 6.3.5 (a), p. 246
4. Exercise 6.3.5 (b), p. 246
5. Exercise 6.3.5 (c), p. 246

Problem Set 1.17.
(Study section 6.4 on Deterministic Pushdown Automata)

1. Exercise 6.4.1 (a), p. 251
2. Exercise 6.4.1 (c), p. 251
3. Exercise 6.4.2 (a), p. 251
4. Exercise 6.4.2 (b), p. 251
5. Exercise 6.4.2 (c), p. 251

Problem Set 1.18.
(Study section 7.1 on Normal Forms for CFG's)

1. Exercise 7.1.3, p. 271
2. Exercise 7.1.4, p. 271-272
3. Exercise 7.1.5, p. 272
4. Exercise 7.1.6, p. 272
5. Exercise 7.1.8, p. 272