CS 5900 Compiler Final Exam Review: Topics and Examples

Exercise 3.1, 3.2, 3.3, 3.4, a5 (10 pts) Derivation, parse tree, and ambiguity.
Given the grammar A → AA | (A) | ε. Show it is ambiguous.

Exercise 3.20

(a) (10 pts) Write a regular expression that generates the same language as the following CF grammar:
A → aA | B | ε
B → bB | A

(b) (10 pts) Write a CF grammar that generates the same language as the following regular expression: **(a | c | ba | bc)*(b | ε)**.

Exercise 4.5, 4.6 (10 pts)
Show the action of LL(1) parse that uses the table below to recognize string: (())().

M[N, T]	(	)	$
S	S → (S)S	S → ε	S → ε

Exercise 4.8 (10 pts) Remove left recursion**.**

…
lexp-seq → lexp-seq lexp | lexp

From A → Aα | β

To A → βA'
A' → αA' | ε

Textbook Example 4.3 on P.159 (10 pts) Remove left recursion**.** Be careful with the case that a substitution is needed first.
A → Ba | Aa | c
B → Bb | Ab | d

α = a
β = Ba | c

α = b | aA'b
β = cA'b | d

Exercise 4.9 (10 pts) Left factor the grammar:
lexp → *atom* | *list*
atom → ***number*** | ***identifier***
list → ***(*** *lexp-seq* ***)***
lexp-seq → *lexp* **,** *lexp-seq* | *lexp*

SLR(1) Parsing. Exercise 5.1, 5.3, a6, a7
Given the CF grammar: S → S (S) | *ε*

(1) (10 pts) Convert the grammar to a augmented grammar with a new start symbol and list all LR(0) items.
(2) (10 pts) Construct NFA of LR(0) items for the grammar.
(3) (10 pts) Construct the DFA from the above NFA
(4) (10 pts) Show the parsing stack and the action (shift and reduce) of an SLR(1) parser for the input ***(()())*** using the DFA constructed above.
OR Show the parsing stack and the action (shift and reduce) of an SLR(1) parser for the input ***()())***.