CS 152: Exam-I (May 1, 2020) **Name:___Rajiv Gupta_____**

30/30

1.(30 points) Regular Expressions and Context Free Grammars

i. Given the following grammar: S → A B | ϵ A → a A | ϵ B → b B | ϵ
Identify the string (if any) that has *multiple leftmost derivations*:
a) a a a
b) a b b b
c) ϵ
d) None of the above

ii. Identify the *regular grammar* (if any) among the following:
a) S → (S) S | ϵ
b) S → a S | a | ϵ
c) S → A B | ϵ A → a A | a B → b B | b
d) None of the above

iii. Identify the grammar (if any) that is *LL(1)* among the following:
a) E → E + T | T T→ id | (E)
b) A → a B | a C B → b B | b C → c C | c
c) S → (S) S | ϵ
d) None of the above

iv. Consider the regular expression $(x^+ y)?$. x where Σ = {x,y}. Which of the following strings **can be** generated by the regular expression.
a) y x
b) x x y x x
c) x x
d) All of the above

v. Consider the regular expression [5-7] | [23][0-8]. Which of the following strings **cannot** be generated by the regular expression?
a) 6
b) 28
c) 39
d) None of the above

vi. Which of the following regular expressions is equivalent to given regular expression: (a | b) ? c where Σ = {a,b,c}.
a) a b c
b) a c | b c | c
c) a c | b c

25/25

2.(25 points) Top Down Parsing. Given the following grammar:

#1 LEXP → ATOM	#3 ATOM → num	#5 LIST → (LEXP LSEQ)
#2 LEXP → LIST	#4 ATOM → id	#6 LSEQ → LEXP LSEQ
		#7 LSEQ → ϵ

i. TABLE [LEXP, num] = #1

ii. TABLE [LEXP, ( ] = #2

iii. TABLE [ATOM, num] = #3

iv. TABLE [ATOM, ( ] = error

v. TABLE [LIST, ( ] = #5

vi. TABLE [LSEQ, num] = #6

vii. TABLE [LSEQ, id] = #6

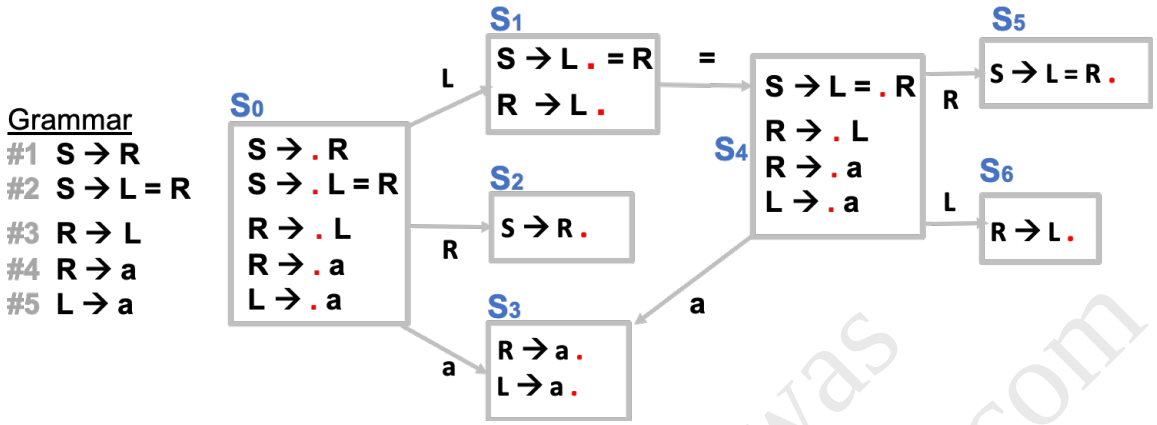
viii. TABLE [LSEQ, ( ] = #6

ix. TABLE [LSEQ,)] = #7

x. TABLE [LSEQ, $] = error

45/45

3. (45 points) Bottom Up Parsing: Given an incomplete SLR(1) state machine.



i. Provide *items missing* from state S_0 R → . L & L → . a

ii. Provide *items missing* from state S_1 R → L .

iii. Provide *items missing* from state S_3 L → a .

iv. Provide action: **ACTION [S_0, a] =** shift, S_3

v. Provide action: **ACTION [S_4, a] =** shift, S_3

vi. Provide action: **ACTION [S_1, =] =** shift, S_4

vii. Provide action: **ACTION [S_4, $] =** error

viii. Provide action: **ACTION [S_2, $] =** reduce, S→R / accept

ix. Provide action: **ACTION [S_6, $] =** reduce, R→L

x. Provide action: **ACTION [S_6, =] =** error

xi. Provide action: **ACTION [S_5, $] =** reduce, S→L=R / accept

xii. Identify a state, if any, that contain a *shift-reduce* conflict?

No (shift on = and reduce on $)

xiii. Identify a state, if any, that contains a *reduce-reduce* conflict?

Yes, in S_3 -- because $ is in Follow(R) and Follow(L).

xiv. Will conflicts found, if any, be present in the LR(1) parser? Yes