Formal Lang. and Automata Th. Spring 2016 Key-Final Exam Pts 50 Time 100 min

(UG 1) Write a grammar for the following language on ∑={0,1}, L={0^n1^m | n,m≥0, n<m}. Generate the strings with your grammar: 001, 011. [6+2+2]

S->R1T
R->0R1| ϵ
T->1T| ϵ

011: S->R1T->0R11T->0ϵ11T->011T->011ϵ->011
001111: S->R1T->0R11T->00R111T->00ϵ111T->00111T->>001111T ->001111ϵ->001111

Yes

(Grad 1) Write a grammar for the following language on ∑={0,1}, L={0^n1^m | n,m≥0, n≠ m}. Generate the strings: 001, 011. Is this a *recursive language*? [5+2+2+1]

S->0S1| B|A
B->1B| 1
A->0A| 0

001: S->0S1->0A1->001
011: S->0S1->0B1->011

No

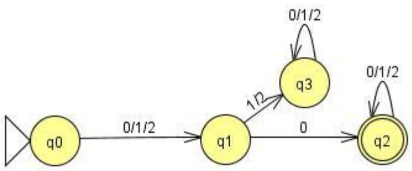
(2) L=**{***Second* symbol from the *left* must be 0} on ∑={0,1,2}. // **Grad: (2d)** Write a Turing Machine for the language. [1+2+3+4]

	0	1	2	B
->q_0	(q_1, 0, R)	(q_1, 1, R)	(q_1, 2, R)	-
q_1	(q_2, 0, R)	-	-	-
q_2^*	-	-	-	2

(2a) Write regular expressions for the language.

(0+1+2)0(0+1+2)*

(2b) Write a DFA for the language.



(2c) Write a PDA (only valid transitions in the table/diagram will do). [2+3+5]

PDA: M = ({q_0, q_1, q_2}, {0, 1}, {#}, δ, q_0, #, { })
δ:

(1) $\delta(q_0, 0, \#) = \{(q_1, \#)\}$ // the first symbol is 0
(2) $\delta(q_0, 1, \#) = \{(q_1, \#)\}$ // the first symbol is 1
(3) $\delta(q_0, 2, \#) = \{(q_1, \#)\}$ // the first symbol is 2
(4) $\delta(q_1, 0, \#) = \{(q_2, \#)\}$ // the first symbol is 0
(5) $\delta(q_2, 0, \#) = \{(q_2, \#)\}$ // more 0's
(6) $\delta(q_2, 1, \#) = \{(q_2, \#)\}$ // more 1's
(7) $\delta(q_2, 2, \#) = \{(q_2, \#)\}$ // more 2's
(8) $\delta(q_2, \varepsilon, \#) = \{(q_2, \varepsilon)\}$ // accept

(3a) Write a PDA for the language is $L=\{0^m1^m2^n \mid n,m \geq 0\}$ on $\sum=\{0,1,2\}$? // **Grad:** $L=\{0^n1^m2^n \mid n,m \geq 0\}$
(3b) Run the string 00112. [8+2]

PDA: M = ({q_1, q_2 }, {0, 1.2}, {#, L}, δ, q_1, #, { })
δ:

(1) $\delta(q_1, 0, \#) = \{(q_1, L\#)\}$ // push L when getting first 0
(2) $\delta(q_1, 0, L) = \{(q_1, LL)\}$ // push L when getting more 0's
(3) $\delta(q_1, 1, L) = \{(q_2, \varepsilon)\}$ // pop L when getting first 1
(4) $\delta(q_2, 1, L) = \{(q_2, \varepsilon)\}$ // pop L when getting more 1's
(5) $\delta(q_2, \varepsilon, \#) = \{(q_2, \varepsilon)\}$ // accept
(6) $\delta(q_1, \varepsilon, \#) = \{(q_2, \varepsilon)\}$ // null
(7) $\delta(q_1, 2, \#) = \{(q_2, \#)\}$ // processing 2's when m = 0
(8) $\delta(q_2, 2, \#) = \{(q_2, \#)\}$ // processing more 2's

00112: *(q1*, 00112*, #)* |-
(q1, 0112*, L#)* |-
(q1, 112*, LL#)* |-
(q2, 12*, L#)* |-
(q2, 2*, #)* |-
(q2, ε*, #)* |-
(q2, ε, ε*):* *accept*

Grad: L={$0^n1^m2^n$ | n,m≥0}
PDA: M = ({q_1, q_2 }, {0, 1.2}, {#, L}, δ, q_1, #, { })
δ:
(1) δ(q_1, 0, #) = {(q_1, L#)} // push L when getting first 0
(2) δ(q_1, 0, L) = {(q_1, LL)} // push L when getting more 0's
(3) δ(q_1, 1, L) = {(q_1, L)} // processing 1's
(4) δ(q_1, 2, L) = {(q_2, ε)} // pop L when getting first 1
(5) δ(q_2, 2, L) = {(q_2, ε)} // pop L when getting more 1's
(6) δ(q_2, ε, #) = {(q_2, ε)} // accept
(7) δ(q_1, 1, #) = {(q_2, #)} // processing 1's when n = 0
(8) δ(q_2, 1, #) = {(q_2, #)} // processing more 1's
(9) δ(q_1, ε, #) = {(q_2, ε)} // null

00112: *(q1*, 00112, *#)* |-
(q1, 0112, *L#)* |-
(q1, 112, *LL#)* |-
(q1, 12, *LL#)* |-
(q1, 2, *LL#)* |-
(q2, ε, *L#):* *reject*

(4a) What type of language is L={$0^n1^m2^n$ | n,m≥0} on ∑={0,1,2}? // **Grad:** L={$0^n1^m2^m0^n$ | n,m≥0}
(4b) If it is a regular language write a finite state machine, if not prove that by using the corresponding pumping lemma.
(4c) If it is a context free language write a context free grammar, if not prove that by using the corresponding pumping lemma. [2+4+4]

a) type 2 language, type 1 language, type 0 language
CFG,RL
b) **prove**

c)
S->0S2|R| ε
R->1R| ε

Grad: L={$0^n1^m2^m0^n$ | n,m≥0}
a) type 2 language, type 1 language, type 0 language
CFG,RE,RL

b) **prove**

c)
S->0S0|R| ε
R->1R2| ε

(5a) Write a *Turing machine* for $L=\{0^n1^n2^m \mid n,m\geq 0\}$ on $\sum=\{0,1,2\}$. // **Grad:** $L=\{0^n1^{2n}\ 2^n \mid n,m\geq 0\}$
(5b) Run the string 0012. // **Grad:** Universal language is a recursively enumerable language: justify this in a few lines. [8+2]

	0	1	2	X	Y	B
-> q_0	(q_1, X, R)	-	-	-	(q_3, Y, R)	(q_5, B, R)
q_1	(q_1, 0, R)	(q_2, Y, L)	-	-	(q_1, Y, R)	-
q_2	(q_2, 0, L)	-	-	(q_0, X, R)	(q_2, Y, L)	-
q_3	-	-	(q_4, 2, R)	-	(q_3, Y, R)	(q_5, B, R)
q_4	-	-	(q_4, 2, R)	-	-	(q_5, B, R)
q_5^*	-	-	-	-	-	-

$q_0$0012B |— X$q_1$012B
|— X0$q_1$12B
|— X$q_2$0Y2B
|— q_2X0Y2B
|— X$q_0$0Y2B
|— XXq_1Y2B
|— XXY$q_1$2B

Reject

Grad: $L=\{0^n 1^{2n} 2^n \mid n,m \geq 0\}$

	0	1	2	X	Y	Z	B
->q_0	(q_1, X, R)	-	-	-	(q_4, Y, R)	-	(q_5, B, R)
q_1	(q_1, 0, R)	(q_e, Y, R)	-	-	(q_1, Y, R)	-	-
q_e	-	(q_2, Y, R)	-	-	-	-	-
q_2	-	(q_2, 1, R)	(q_3, Z, L)	-	-	(q_2, Z, R)	-
q_3	(q_3, 0, L)	(q_3, 1, L)	-	(q_0, X, R)	(q_3, Y, L)	(q_3, Z, L)	-
q_4	-	-	-	-	(q_4, Y, R)	(q_5, Z, R)	(q_5, B, R)
q_5^*	-	-	-	-	-	-	-

$q_0$00112B ├— X$q_1$0112B
├— X0$q_1$112B
├— X0Yq_e12B
├— X0YY$q_2$2B
├— X0Yq_3YZB
├— X0q_3YYZB
├— X$q_3$0YYZB
├— q_3X0YYZB
├— X$q_0$0YYZB
├— XXq_1YYZB
├— XXYq_1YZB
├— XXYYq_1ZB

Reject

*The class recursively enumerable consists of all languages for which there **exists a Turing machine.**

• Define the language Lu as follows:
Lu = {x | x is in {0, 1}* and x = <M,w> where M is a TM encoding and w is in L(M)}
• Let x be in {0, 1}*. Then either:
1. x doesn't have a TM prefix, in which case x is not in Lu
2. x has a TM prefix, i.e., x = <M,w> and either:
    a) w is not in L(M), in which case x is not in Lu
    b) **w is in L(M), in which case x is in Lu**