Exame de Época Normal / First Exam Period (2022/07/01)

Duration: 2 hours **Version A**

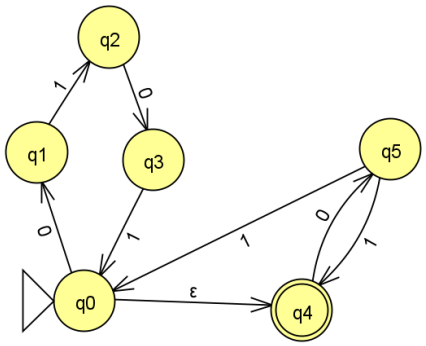
No consultation is allowed, other than the supplied document.
No electronic means are allowed (computer, cellphone, …).
Fraud attempts lead to the annulment of the exam for all students involved.

Answer each group in separate sheets!
Write your full name and exam version in all sheets!

POSSIBLE ANSWERS

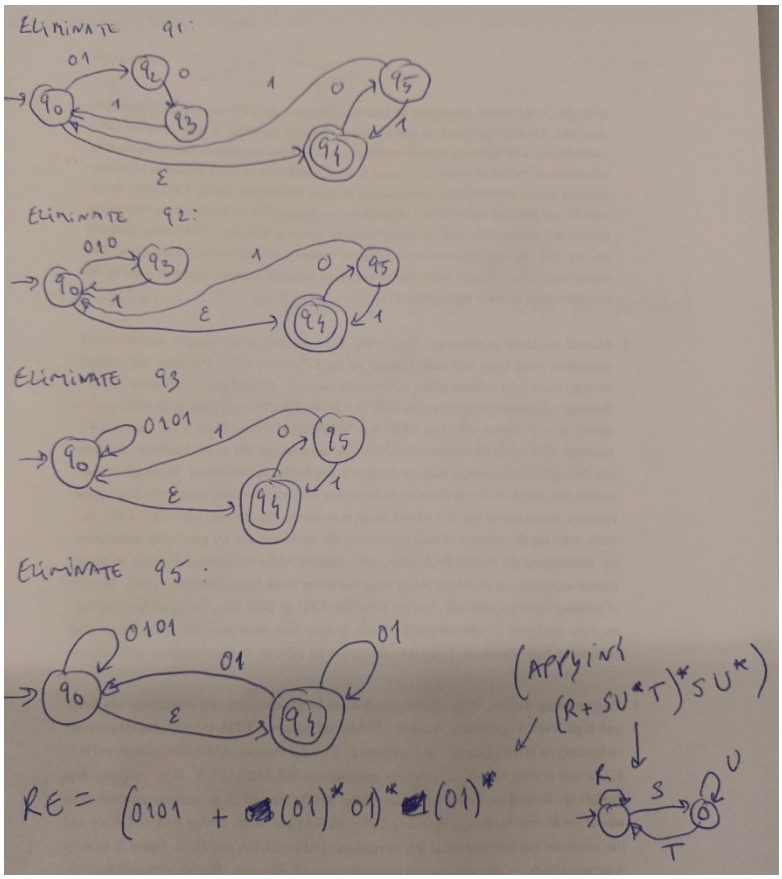
Group I: Construct [5 Pts] Finite Automata, Regular Expressions, and Regular Languages (RLs)

Consider the following ε-NFA N1:

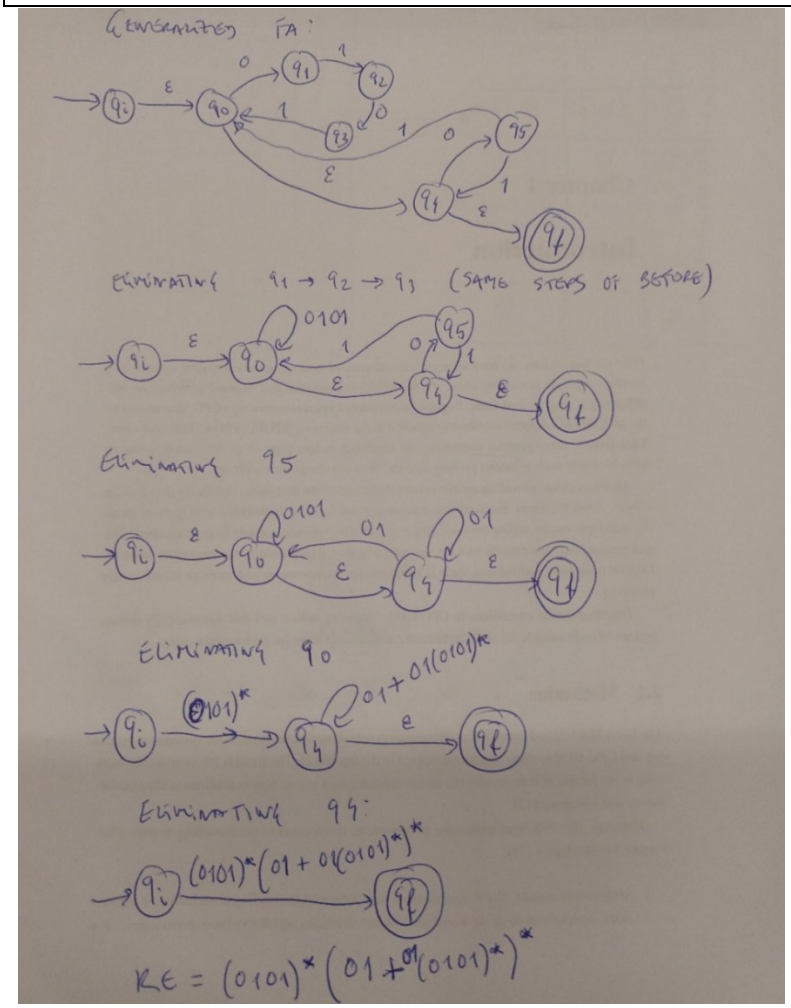


a) Show the regular expression that represents L(N1) and obtained by using state elimination with the ordering of elimination: q1→q2→q3→q5. [note: show all the steps needed]

Answer:

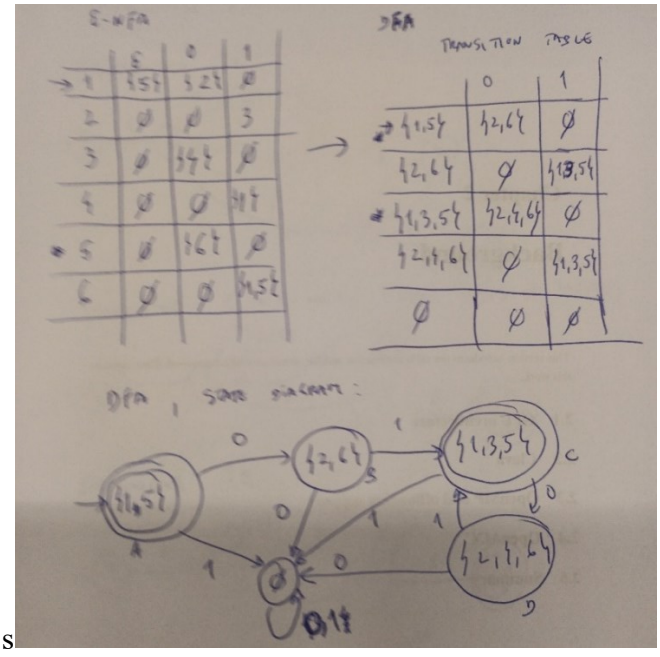


Or:



b) Using the subset construction technique convert N1 to a DFA and present the transition table and the state diagram of the resultant DFA. [note: while applying the technique consider only the states reached from the start state]

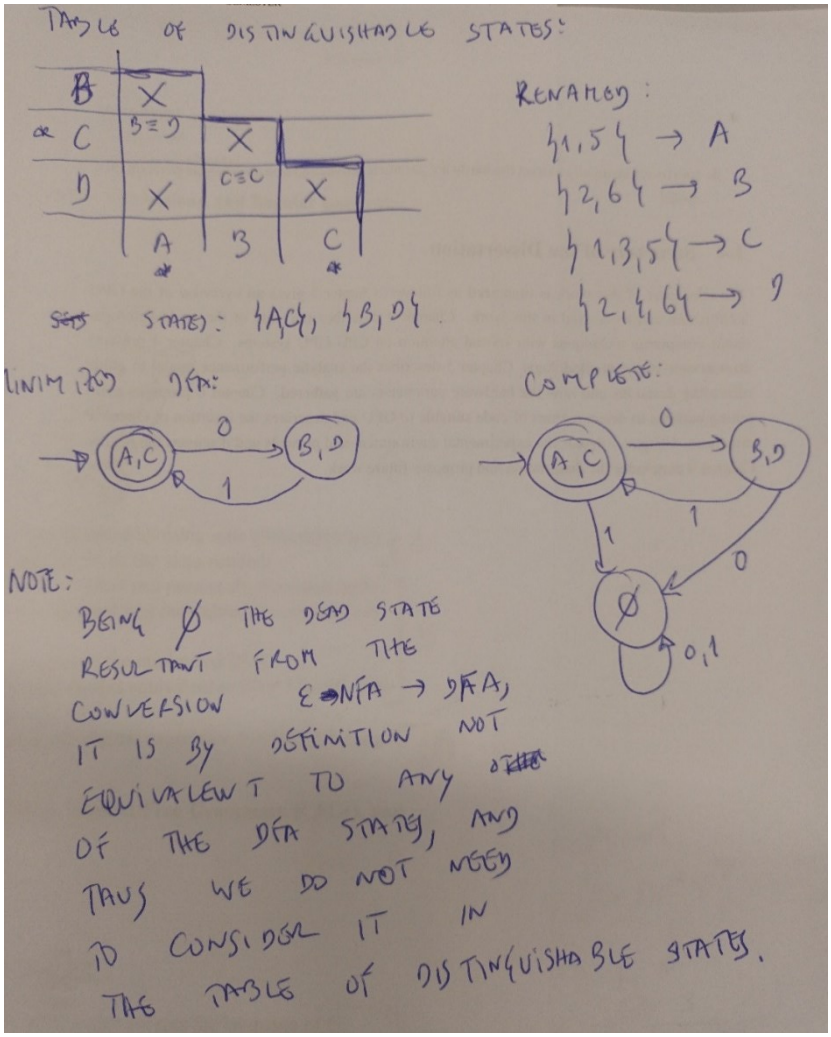
Answer:



s

c) Minimize the previous DFA and show the state diagram of the minimized DFA.

Answer:



d) Can you conclude that the regular expression obtained in a) is equivalent to (01)*? Justify your answer.

Answer:
Yes. The regular expression representing the minimized DFA is (01)*.
As the minimized DFA is obtained from proved techniques that transform any ε-NFA in a DFA and then in a minimized DFA, and the regular expression of N1 was obtained by also a proved technique (state elimination), the two regular expressions are equivalent (i.e., they represent the same language).

e) Prove by induction that all the strings represented by the regular expression 0101(01)* belong to L(N1).

Answer:
Hypothesis:
S_n: The strings given by $0101(01)^n$, with $n \geq 0$,are accepted by N1, i.e., belong to L(N1)
Basis:
Consider the shortest string represented by $0101(01)^n$, with $n \geq 0$, which is 0101
$\hat{\delta}(q0, 0101)=\{q0,q4\}$ and as q4 is a final state then N1 accepts 0101.
Induction step:

S_{n+1}: $0101(01)^{n+1}$, with n≥0,are accepted by N1, i.e., belong to L(N1)
$0101(01)^{n+1} = 0101(01)^n(01)$
By hypothesis the strings $0101(01)^n$, with n≥0, are accepted by N1, and thus in the end of any of those strings N1 is at q4.
Since $\hat{\delta}(q4,01)=\{q0,q4\}$ and as q4 is a final state then N1 accepts $0101(01)^{n+1}$, with n≥0. *qed*

Group II: [5 Pts] Context-Free Languages (CFLs), Context-Free Grammars (CFGs), and Push-Down Automata (PDAs)

Consider the following context-free grammar G:

S → aSc | B
B → bBc | ε

a) Formally define the set of strings of the language of G.

Answer:
$L(G) = \{a^k b^l c^m \mid m=k+l \wedge m,l,k \geq 0\}$

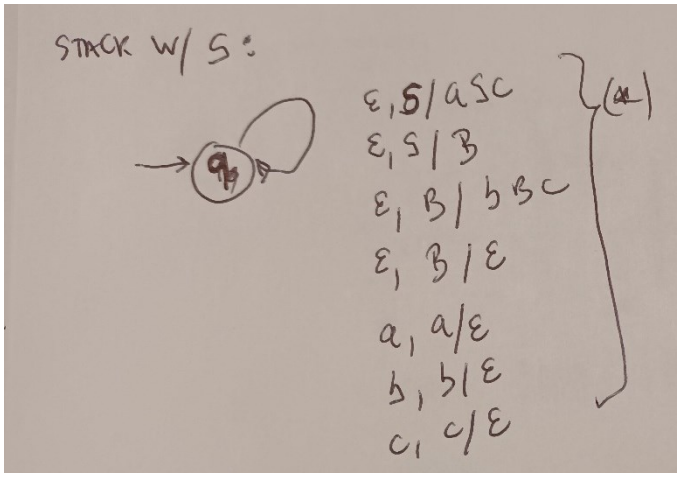
b) Is G ambiguous? Justify your answer.

Answer:
The grammar is not ambiguous as there is not possible to derive two different syntax trees for at least one string of L(G).

Requires a justification more specific to the grammar:
The first production of the start variable of the grammar (S) is able to recursively derive one 'a' to the left and one 'c' to the right and to derive B for the middle substring, and when in B the first production recursively derives one 'b' to the left and one 'c' to the right or finishes by deriving ε. As there is only one possible derivation to add the 'a' and 'c' and then the 'b' and 'c' there will not be more than one syntax tree for each string of L(G).

c) Using the technique presented in the lectures, draw a PDA that accepts the language of G.

Answer:

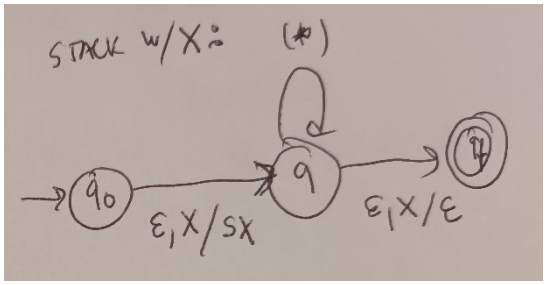


Or:

P = ({q}, {a, b, c}, {a, b, c, S, B}, δ, q, S, {})
δ (q, ε, S) = {(q, aSc), (q, B)}
δ (q, ε, B) = {(q, bBc), (q, ε)}
δ (q, a, a) = {(q, ε)}
δ (q, b, b) = {(q, ε)}
δ (q, c, c) = {(q, ε)}

d) If the PDA you just presented accepts by empty stack, transform it into one that accepts by final state. If the PDA accepts by final state, then transform it to accept by empty stack. Use the conversion presented in classes for this transformation.

Answer:



Or:

P2 = ({q, **q0, qf**}, {a, b, c}, {a, b, c, S, B, **X**}, δ, **q0**, **X**, {**pf**})
δ (q0, ε, X) = {(q, SX)}
δ (q, ε, S) = {(q, aSc), (q, B)}
δ (q, ε, B) = {(q, bBc)}
δ (q, a, a) = {(q, ε)}
δ (q, b, b) = {(q, ε)}
δ (q, c, c) = {(q, ε)}
δ (q, ε, X) = {(qf, ε)}

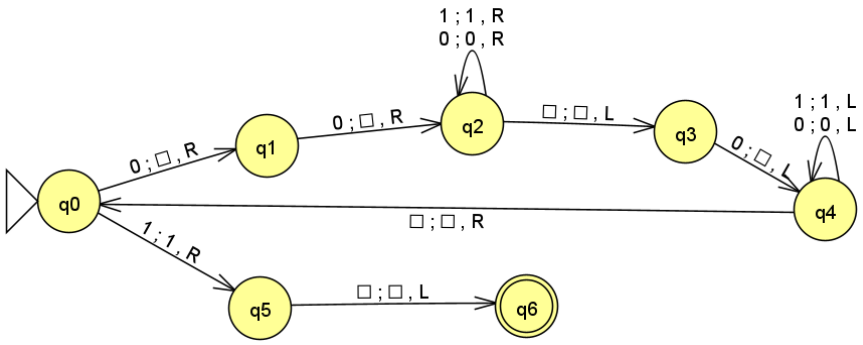
Group III: [5 Pts] Turing Machine

We want to implement a Turing Machine (TM) able to accept the language L={$0^{2n}10^{n}$ | n ≥0 }. For example: 0010 and 1 are accepted; 00010 is rejected.

a) Draw a possible TM for L.

Answer:

Using the state/transition diagram:



(labels used for the states should be renamed to q6->sf, and qi->si in order to be consistent with the labels used below)

Or:
M=({s0,s1,s2,s3,s4,s5,sf},{0,1},{0,1,B},δ,s0,B,{sf}) that decides L, where δ is given by the following table:

	0	1	B
s0	(s_1,B,R)	(s_5,1,R)	-
s1	(s_2,B,R)	-	-
s2	(s_2,0,R)	(s_2,1,R)	(s_3,B,L)
s3	(s_4,B,L)	-	-
s4	(s_4,0,L)	(s_4,1,L)	(s_0,B,R)
s_5	-	-	(s_f,B,L)
s_f	-	-	-

Another solution: This solution uses additional symbols in the tape, more close to what they have done for CA3.

	0	1	X	Y	B
s0	(s_1,X,R)	(s_5,1,R)	-	-	-
s1	(s_2,X,R)	-	-	-	-
s2	(s_2,0,R)	(s_3,1,R)	-	-	-
s3	(s_4,Y,L)	-	-	(s_4,Y,R)	-
s4	(s_4,0,L)	(s_4,1,L)	(s_0,X,R)	(s_4,Y,L)	-
s_5	-	-	-	(s_5,Y,R)	(s_f,B,L)
sf	-	-	-	-	-

b) Indicate the first 10 steps of the computing trace when the input to the TM is: 0010.
Answer:

Using the first machine:

$s_0 0010 \vdash s_1 010 \vdash s_2 10 \vdash 1 s_2 0 \vdash 10 s_2 B \vdash 1 s_3 0 \vdash s_4 1 \vdash s_4 B 1 \vdash s_0 1 \vdash 1 s_5 B \vdash s_f 1.$

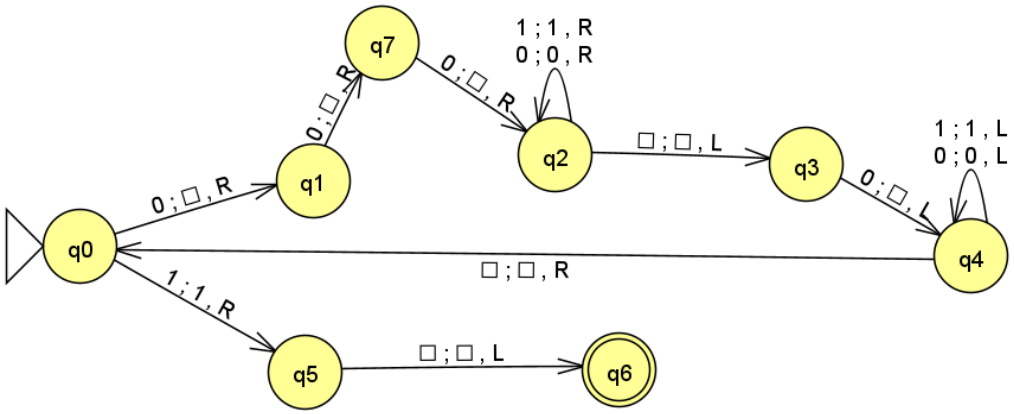
With the second machine:

$s_0 0010 \vdash X s_1 010 \vdash XX s_2 10 \vdash XX1 s_3 0 \vdash XX s_4 1Y \vdash X s_4 X1Y \vdash XX s_0 1Y \vdash XX1 s_5 Y \vdash XX1Y s_5 B \vdash XX1 s_f Y$

c) Reusing the TM in a) with as few as possible modifications, draw a possible TM to accept the language L'=$\{0^{3n}10^n \mid n \geq 0\}$.

Answer:
For the first machine: change $(s_1,0)$ to (s_6,B,R) and add $(s_6,0) = (s_2,B,R)$.

Using the state/transition diagram and including the equivalente changes:



(labels used for the states should be renamed to q6->sf, q7->q6, and qi->si in order to be consistent with the labels used below)

For the second machine: change $(s_1,0)$ to (s_6,X,R) and add $(s_6,0) = (s_2,X,R)$.

Group IV: [5 Pts] Statements about Languages (T/F: 20%, justification: 80%; wrong answer = -10%)

Indicate, justifying succinctly (with a couple of sentences or a counter example), whether each of the following statements is True (T) or False (F).

a) The language obtained by a subtraction of a regular language (RL) by a non-regular language (NRL) that includes strings belonging to the RL is always an NRL.

Answer: FALSE
Counter-example: Let´s assume that the RL is a finite language. Then the subtraction always results in another finite language and any finite language is an RL.

Another counter-example: Let's assume the RL, L1, given by the regular expression ab+(0+1)* and the NRL, L2 = $\{c^n d^n + ab \mid n \geq 0\}$. L1-L2 = (0+1)*, which is an RL.

Another counter-example: Let's assume the RL, L1, given by the regular expression (0+1)* and the NRL, L2 = $\{0^n 1^n \mid n \geq 0\}$. L1-L2 = $\{\varepsilon\}$, which is an RL.

b) The language L = $\{w^{2n+3m} \mid w \in \{0,1\}^* \wedge n, m \geq 0\}$ is an RL.

Answer: FALSE

As the strings w can be any string over {0,1}, and as 2n+3m (n, m ≥ 0) can be any natural value ≠ 1 (in fact 2n+3m = 0, 2, 3, 4,…), L can be represented by the following:

$\{\varepsilon\} \cup \{ww \mid w \in \{0,1\}^*\} \cup \{www \mid w \in \{0,1\}^*\} \cup \{wwww \mid w \in \{0,1\}^*\} \cup \ldots \cup \{w^n \mid w \in \{0,1\}^*\}$

This language cannot be represented by an RL as it is an infinite language and needs to recognize 2 or more occurrences of a substring.

Note1: One could also use the fact that $L = \{w^{2n}w^{3m} \mid w \in \{0,1\}^* \wedge n, m \geq 0\}$.

Note that if it was $w \in \{0,1\}$:
TRUE. As 2n+3m, n, m ≥ 0, represents all the natural numbers ≥ 0, excepting the number 1, the language would be $\{\varepsilon\} \cup \{w^n \mid w \in \{0,1\} \wedge n \geq 2\}$, which is an RL (can be represented by the regular expression ε + 000* + 111*).

c) Given a context-free grammar G, if it is possible to define two different derivations for the same word, we can say G is ambiguous.

Answer: FALSE
For a grammar to be ambiguous, the two different derivations must be or leftmost or rightmost. For a non-ambiguous grammar, there can exist two different derivations, one leftmost and another rightmost.

[note 1: this will provide two distinct syntax trees]
[note 2: if there are two different leftmost derivations there are two different rightmost derivations]

d) Any non-deterministic push-down automaton (PDA) obtained from a context-free grammar, using the conversion presented in the lectures, can be transformed to a deterministic PDA.

Answer: FALSE
There are context-free languages that require a non-deterministic PDA. Those languages can be represented by a CFG that, using the conversion presented in the lectures, is translated to a non-deterministic PDA. However, this PDA cannot be translated to a deterministic PDA. If that was possible, deterministic and non-deterministic PDAs would be equivalent!

e) If $L_1 \cap L_2$ is decidable (recursive) then L_1 and L_2 are both decidable.

Answer: FALSE
L1 or/and L2 can be recursive enumerable languages (i.e., not the two decidable) and the intersection of both results in a decidable language. An example is when the intersection is the empty set, which is a decidable language.