Exame de Época Normal / First Exam (2020/01/21)

Duration: 2h30 **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!

Group I: [4.5 Points] Finite Automata and Regular Expressions

Consider the language L_1 = L(a*b):

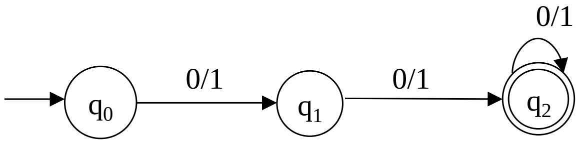
a) Draw a Finite Automata (FA) for L_1 in a methodologically way and present the templates used and all the necessary steps;

b) From the previous FA draw a minimized DFA for L_1;

c) Write a regular expression for the complement of L_1;

d) Given two regular languages L_x and L_y represented by regular expressions, describe how to check if $L_x = L_y$;

e) Given the regular expression RE=(0+1)(0+1)(0+1)* and the DFA bellow, prove by induction that L(RE) = L(D);



Group II: [2 Pts] Properties of Regular Languages

a) Knowing that $L_1 = \{a^n b^n \mid n \geq 0\}$ is a non-regular language, prove, without using the Pumping Lemma for regular languages, that $L_2 = \{0^{n+m}2^n \mid n \geq 3 \text{ and } m \geq 3\}$ is also a non-regular language;

b) Suppose that given a DFA A for a language L_1, that is L(A) = L_1, we perform the following transformations on A to obtain a new DFA A':

- Change all the accepting states into non-accepting states;
- Change all the non-accepting states into accepting states.

Prove that L(A`) = L_1^C (with L_1^C representing the complement of L_1).

Group III: [4.5 Pts] Context-Free Grammars (CFG) and Push-Down Automata (PDA)

Consider the following CFG:

$S \rightarrow AB \mid aaB$
$A \rightarrow a \mid Aa$
$B \rightarrow b$

a) Draw the PDA, which accepts by empty stack, directly obtained from the CFG;
b) Indicate a sequence of instantaneous descriptions that result in the PDA acceptance of the string: "aaab";
c) Draw, for the CFG, a PDA that accepts by final state;
d) Show a leftmost derivation for the string "aaab" and draw the respective syntax tree;
e) Is this CFG ambiguous? Justify your answer and provide a non-ambiguous grammar in the case it is ambiguous;

Group IV: [4 Pts] Turing Machine

We intend to propose a Turing Machine (TM) to implement the language L={w | w $\in$ {0,1}* and $n_0(w)=n_1(w)$}, where n_0 and n_1 give the number of 0's and the number of 1's in a string, respectively.

a) Describe a strategy to implement a possible TM for L;

b) Draw a TM related to the described strategy;

c) Indicate the computing trace when the input to the TM is: 0110;

d) What modifications to the TM you need, to identify, for any given w $\in$ {0,1}*, the three possible cases: $n_0(w)= n_1(w)$; $n_0(w)> n_1(w)$; and $n_0(w)< n_1(w)$.

Group V: [5 Pts] Statements about Languages (T/F: 20%, justification: 80%; wrong answer = reduction of 50%)

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) In a DFA with n states representing L, if there is a string of length > n in L, then there is a string of length between n and 2n-1;

b) $L=\{a^nb^m \mid n \neq 2m \text{ and } n \geq 0 \text{ and } m \geq 0\}$ is a CFL;

c) The language represented by the CFG, $S \rightarrow 00S \mid 11S \mid \varepsilon$, is a regular language;

d) There are CFGs that can be automatically translated to DFAs;

e) If L is a CFL then its complement L^C is also a CFL;

f) A language of palindromes can be a regular language as long as those palindromes are formed using the alphabet {0,1};

g) The pumping lemma for regular languages can be used to prove that a language is regular;

h) There is a systematic way to provide a deterministic Turing Machine for any given regular expression;

i) An algorithm requiring recursivity cannot be implemented with a Turing Machine;