MA/CSSE 474 Exam 2 Winter 2015-16 Name____**Solution**_______ Section: 01 (2^{nd}) ____ 02 (3^{rd})___ 03(4^{th}) ___

No books or papers except the "notation and formula" sheet supplied by the instructor.

You may not use any devices with WI-FI/bluetooth, headphones, or earbuds.

No communication about this exam's contents with anyone besides the instructor before 11:45 AM.

If a problem is marked with a †, it means that there is information on the "notation and formulas" sheet that gives some helpful details for this problem.

Scores:

Problem	Possible	Score
1	30	
2	5	
3	4	
4	21	
5	10	
6	15	
7	10	
8	5	
Total	**100**	

1. (30 points) Circle T or F to indicate whether it is *True* or *False*. IDK means *I don't know*
If the statement can ever be False, then False is the correct answer. **You do not need to give proofs or counterexamples.**
For each part, you earn **1 point** for circling IDK, **2** for circling the correct answer, **-1** for circling the incorrect answer, and **0** if you leave it blank. Leaving it blank is silly, since you get more points for IDK.

a) T (F) IDK The set of all languages over alphabet {a} is countably infinite
b) (T) F IDK The set of all regular languages over alphabet {a, b} is countably infinite.
c) (T) F IDK The complement of a non-regular language must be non-regular.
d) T (F) IDK The complement of an infinite language must be infinite.

The next parts deal with the function *pref*(L) = {w∈Σ* : ∃x∈Σ* (wx ∈ L) }

e) (T) F IDK The set of regular languages is closed under *pref*.
f) T (F) IDK The set of non-regular languages is closed under *pref*.
g) (T) F IDK The set of finite languages is closed under *pref*.
h) (T) F IDK The set of infinite languages is closed under *pref*.
i) (T) F IDK If the transition graph of a DFSM M has no loops or cycles, then L(M) is finite.
j) T (F) IDK † If maxstring(L) is regular, then L must be regular.
k) (T) F IDK † If L is a regular language with alphabet {a, b}, then *chop*(*L*) ∩ {a, aba} = ∅ .
l) T (F) IDK If neither of the languages L_1, L_2 is regular, then $L_1 \cup L_2$ is not regular.
m) (T) F IDK Every non-regular language is the intersection of a countably infinite set of regular languages.
n) (T) F IDK Every non-regular language is the intersection of a countably infinite set of non-regular languages.
o) T (F) IDK Every regular language is the intersection of a countably infinite set of regular languages.

2. (5 points) † Consider the contrapositive to the Pumping Theorem (it is written out on the "Notations" handout). If we are using this to prove that a language L is non-regular, which of the values mentioned in the theorem do we get to choose? Circle them (and do not circle the ones that we are not allowed to choose).

k (w) x y z (q)

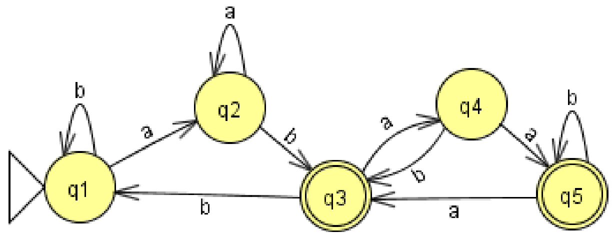
3. (4 points) Write the English name for each of these Greek letters that we frequently use in the course.

ε epsilon Σ sigma δ delta κ kappa

Explanations for Problem 1 answers

a) The set of all languages over any non-empty alphabet is uncountable.
b) If L is regular, L = L(α) for some regular expression α. The set of regular expressions over {a, b} is countable (we saw how to enumerate this set in class last week).
c) Let L be a non-regular language. What if its complement L' were regular. Then the complement of L' would have to be regular. But L is the complement of L'.
d) If Σ={a}, then Σ*-{a} is infinite. Its complement is {a}, which is certainly finite.
e) We did the construction in class.
f) Consider the non-regular language L = {a^p : p is prime}. Pref(L) is {a}*, which is regular.
g) Taking all prefixes of a finite set of strings gives us a finite set of strings.
h) Taking all prefixes of an infinite set of strings gives us an infinite set of strings.
i) The transition graph of *every* DFSM has at least one loop or cycle, since there is a transition out of every state on every alphabet symbol.
j) Consider the non-regular language L = {a^p : p is prime}. *Maxstring*(L) is ∅, which is regular.
k) For any L, chop(L) does not contain any odd-length strings.
l) Consider the non-regular languages L_1 = {a^p : p is prime} and L_2 = {a^p : p is not prime}. $L_1 \cup L_2$ = {a}*.
m) This was on HW 9. See the HW9 solution.
n) Let L be a non-regular language. L' = ¬L is also non-regular, so infinite. Let {w_0, w_1, w_2, …} be an enumeration of L'. Let T be { L∪ {w_i} : i ≥ 0}; this is an infinite set of languages. Adding one element to a non-regular language does not make it regular. L is the intersection of all of the languages in T.
o) Let Σ = {a}. L = Σ* is a regular language. If an intersection results in L, it can be the intersection of only one set. If any pother set is included it will not contain some element of L, so the intersection cannot be L.

4. (21 points) † Consider the following transition diagram for a DFSM M, and the algorithm from class to find a regular expression α such that $L(\alpha) = L(M)$.



(6) What the value of

r_{220}? $a \cup \epsilon$

r_{350}? $\emptyset$

r_{321}? b^+a Credit for each: 2 for same or equivalent answer, 0 otherwise

(6) For each the following two r_{ijk} values, fill in the four sets of three subscripts that tell how to calculate the given r value using the recursive formula. -1 for each incorrect subscript number, up to -3 for each of the problem parts

$r_{225} = r_{224} \cup r_{254}\ r_{554}{}^*\ r_{524}$ $\qquad$ $r_{414} = r_{413} \cup r_{443}\ r_{443}{}^*\ r_{413}$

(3) Write a regular expression that defines L(M), in terms of a union of some of the r_{ijk}. You do not have to evaluate that/those r_{ijk}.

$L(M) = r_{135} \cup r_{155}$ No partial credit for this part

(3) List all pairs (i, j) such that R_{ij1} is different than R_{ij0}. -1 point for each missing or incorrect pair (minimum of zero)

(1, 1) $b \cup \epsilon \rightarrow b^*$ (1, 2) $a \rightarrow b^*a$ (3, 1) $b \rightarrow b^+$ (3, 2) $\emptyset \rightarrow b^+a$

(3) List all pairs (j, j) such that R_{ij2} is different than R_{ij1}. -1 for one or two missing or incorrect pairs, -2 for three or four, -3 for more than four

(1, 2) $b^*a \rightarrow b^*a^+$ (1, 3) $\emptyset \rightarrow b^*a^+b$ + (2, 2) $a \cup \epsilon \rightarrow a^*$ (2, 3) $b \rightarrow a^*b$ (3, 2) $b^+a \rightarrow b^+a$ (3, 3) $\emptyset \rightarrow b^+a^+b$

5. (10 points) This problem is the same as a problem in HW5, but you do not have to prove your answer here. Let $M_1=(K_1, \Sigma, \delta_1, s_1, A_1)$ and $M_2=(K_2, \Sigma, \delta_2, s_2, A_2)$ be DFSMs that accept the regular languages $L_1 = L(M_1)$ and $L_2 = L(M_2)$. Let $L = L_1 \cap L_2$. Show that L is regular by carefully constructing a DFSM $M=(K, \Sigma, \delta, s, A)$ such that $L=L(M)$. I do not expect you to have memorized the solution; you should be able to figure out the details again.

Answer: Full or no credit for each part

(2) $K = K_1 \times K_2$

(4) δ is defined as $\delta((q_1, q_2), a) = (\delta_1(q_1, a), \delta_2(q_2, a))$

(2) $s = (s_1, s_2)$

(2) $A = A_1 \times A_2$ Could also be written {(q1, q2) :

6. (15 points) † Use the contrapositive of the pumping theorem directly (no closure properties or equivalence classes) to show that L = {w∈{a,b}* : ∃x,y∈{a,b}(∃n≥0 (w = xa^nb^ny))} is not regular.

Given k ≥ 0, choose w = $a^{k+1}b^{k+1}$, which is an element of L, because it is aa^kb^kb.
If w=xyz (meeting the requirements |xy| ≤ k and |y|≥1), then y=a^p for some p≥1 and p≤k.
Choose q=0 (the textbook calls this *pumping out*).
Then xz is aa^{k-p}bkb, which is not in L.

Of course, there are other w's and q's that can work. Examples of point values to take off.
(-10) Choosing a w that does not depend on k, or not making it specific (for example ba^2b^2a or xa^kb^ky)
(-5) Assuming that y is a specific string that the student chooses (for example y=a^3)
(-5) Not choosing a specific value of q; not pumping correctly, or =not getting outside L after pumping.
(?) Other errors on a case-by-case basis.

7. (10 points) Write a decision procedure that decides the following problem:
Given regular expressions α and β over an alphabet Σ, is it true that L(α) is the complement of L(β)?

a) From α, construct a DFSM M such that L(M) = L(α)
b) From β, construct a DFSM M' such that L(M') = L(β)
c) From M', construct a DFSM for ¬L(M')
[students do not have to say this, but it is done by swapping accepting and non-accepting states]
d) From M, construct an equivalent minimal state, canonical form machine M''
e) From M', construct an equivalent minimal state, canonical form machine M'''
f) If M' is the same as M''' return "yes"
else return "no"

8. (5 points) What is the relationship between
the lexicographic ordering of all strings over alphabet Σ
and
the ordering of the states in our textbook's canonical form of a minimal DFSM over Σ?

This could be expressed in many different ways. Here is one:

For any state q in M, define *firstString*(q) to be the first string w in the lexicographic ordering of Σ* such that (s, w) $\vdash_M^*$ (q, ε).
If M is in canonical form, then for any two states numbered q_i and q_j, i < j iff *firstString*(q_i) comes before *firstString*(q_j) in lexicographic order.