CSci 435: Formal Languages and Automata

Instructor: Dr. M. E. Kim Date: October 15th, 2020

**Midterm: 100 points + 20 points (optional)**

**Due: by the end of the day, 10/18 (Sun.)**

Name: \_\_\_\_\_\_\_\_Derek Trom\_\_\_\_\_\_\_\_\_\_\_\_\_

1. Your answer should be precise and fully described; any sloppy answer will not get a full point.
2. Your hand writing should be clear and readable.
3. Do not insert a photo copy of your handwriting but draw a figure using a graphic tool.

Mark the followings;

Difficulty:

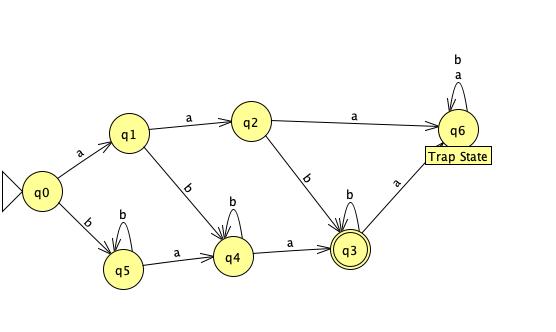
Very Easy: \_\_\_\_\_\_ Easy: \_\_\_\_\_\_ Moderate: \_\_\_X\_\_ Difficult: \_\_\_\_\_\_ Very Difficulty: \_\_\_\_\_\_

Time Taken:

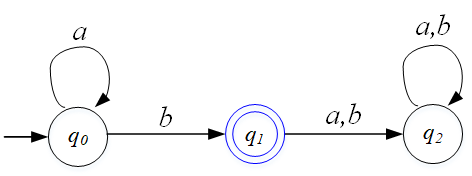
\_\_\_\_4\_\_\_ Hours and \_\_\_\_30\_\_\_\_ Minutes.

Q1. [10] Construct a minimal DFA that accepts L = {w ∈{*a, b*}\* | w has at least one *b* and exactly two *a*’s}.

Hint: the number of states are 7.



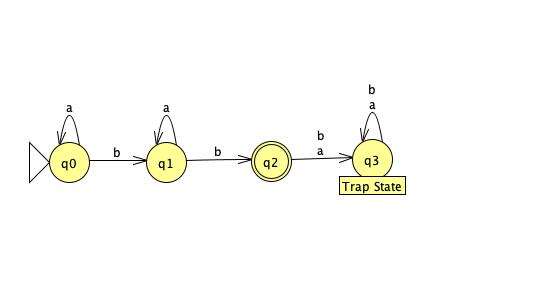
Q2. [10] Let L be the language accepted by the DFA below. Construct a DFA for the language **L2 – L**.



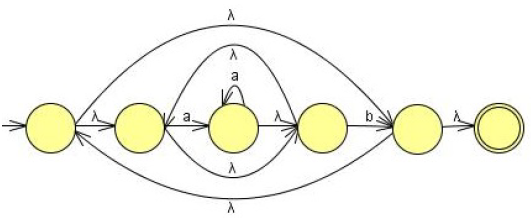
L = a\*b

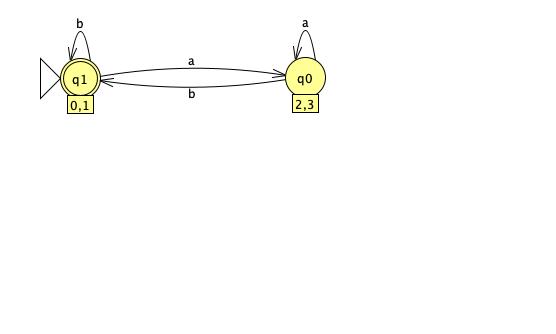
L2=a\*ba\*b

L2-L= = = L2

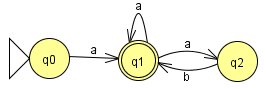


Q3. [15] In the given NFA M of the figure,



1. [5] Give the language **L(M)** that is accepted by M in the simplest ***regular expression***.
   1. R = ((aa\*)\*b)\*
2. [10] Construct a ***minimal DFA*** that is equivalent to the NFA M.
   1. 

Q4. [10] For the language L(*aa*\*(*ab*+*a*)\*)

1. [5] Construct a (minimal) NFA with a single final state that accepts L.
   1. 
2. [5] Construct a right-linear ***regular grammar*** that generates L in the simplest form

Hint: 3 variables and 6 production rules. e.g) A → BC | λ is counted to 2 rules: A → BC or A → λ

* 1. S→aA, A→aA|B, B→abB|aB|λ

Q5. [10, optional] The ***chopleft*** operationof a regular language L is removing the leftmost symbol of every string in L:

***chopleft*(L)** = { *w* | *vw* ∈ L, with |*v*| = 1}.

Prove or disprove that the family of regular languages is ***closed*** under the *chopleft* operation.

Hint: If it’s regular, give an idea of constructing an FA that accepts chopleft(L) using an FA M that accepts L.

Otherwise, give a counterexample.

1. Given L is regular
2. Chopleft(L) is made by removing the leftmost symbol of every string in L
3. Since L is regular there is an FA M that accepts the strings in L
4. If L is regular NFA than for all transitions from q0 the initial state can be replaced with a lambda transition after Chopleft(L) is preformed
   1. For example is initial is S(q0, a) = q1 and S(q0, b) = q2 the transition can be replaced by ∈ transition S(q0, a) = ∈ and S(q0, b) = ∈
   2. This removes the leftmost symbol of a string in L and will accept the string in L with the leftmost symbol removed.
   3. This is regular, thus chopleft(L) is also regular

Q6. [10] Prove or disprove that the language L = {*anblak | n=l* or *l* ≠ *k* } is regular.  
If L is regular, give a *regular grammar* that generates L. Otherwise, disprove it by *Pumping Lemma*.

1. Assume L is regular. Therefore the Pumping Lemma holds for L. Let m be the “magic number” for this language. Consider the string ambmam. We chose this string because it belongs to the language, is long enough, AND the second condition (l≠k) is true and will have to remain true even when we “pump.” This forces us to keep n=l when we pump in order to keep the pumped strings in the language. By the Pumping Lemma, this string can be represented as xyz with |xy|≤m and y non-empty, so that xyiz is in Lfor all values of i. But since the length of xy is ≤ m and there are ma’s at the start of the string, xy(and also y) must consist of only a’s. So y=ak with k> 0.The string xy2z is am+kbmam, because we have added another y to the string. But the number of a’s at the beginning here does not equal the number of b’s and the number of b’s is equal to the number of a’s at the end, so neither part of the condition holds and this string is NOT in the language. This contradicts the Pumping Lemma. Therefore, since the Pumping Lemma holds for all regular languages, the language L={anblak: n=l or l≠k} is not regular.

Q7. [10] Show that the L = { *anbmck* | *n=m* or *m* ≠ *k* } where *n, m, k* ≥ 0 is context-free, by giving the Context-Free Grammar that generates it.

1. There are two languages to start *anbmck* | *n=m, anbmck* | *m* ≠ *k*
2. L1 = {*anbmck* | *n=m}*
3. L2= {anbmck | m ≠ k}
4. S→aAb|aC, A→aAb|B, B→Bc|c, C→aC|bDc, D→bDc|b|c

Q8. [15] In the given CFG, G = ( {S}, {*a, b*}, S, P ) with productions S → SS | *a*S*b* | *b*S*a*| λ

1. [5] Give the language, L(G), that is generated by G, in a formal expression.
   1. L = { *w* | *na*(*w*) = *nb*(*w*) }
2. [5] Decide if the G is ***ambiguous*** or not. Justify your answer.
   1. I will derive abab
   2. S→ SS→ aSbS→aλbS→ aλbaSb→ aλb aλb→abab
   3. S→aSb→abSab→abλab→abab
   4. This shows that the string can be derived two different ways, thus it is ambiguous.
3. [5] If G is ambiguous, give an unambiguous grammar. Otherwise, show that G is inherently ambiguous.
   1. S->aBS|bAS|λ
   2. A->a|bAA
   3. B->b|aBB

Q9. [10] Transform the grammar with the following productions into a ***Chomsky Normal Form***

S → *ba*AB, A → *bAB* | λ, B → BA*a* |A | λ.

1. Eliminate Lambda
   1. S → baAB|baB|baA|ba, A → AB|bB, B → BAa|A|λ|Ba.
   2. S → abAB|abB, A → bAB|bB|bA|b,B→Baa|A|Ba|Aa|a
2. Remove Unit productions
   1. S→baAB|baB|baA|ba
   2. A→bAB|bB|bA|b
   3. B→BAa|bAB||bB|bA|b|Ba|Aa|a
3. Replace Terminals with variables
   1. S→CDAB|CDB|CDA|CD
   2. A→CAB|CB|CA|C
   3. B→BAD|CAB|CB|CA|C|BD|AD|D
   4. D→a
   5. C→b
4. Final form
   1. S→CU|CX|CY|CD
   2. A→CV|CB|CA|C
   3. B→BZ|CV|CB|CA|C|BD|AD|D
   4. U→DV
   5. V→AB
   6. X→DB
   7. Y→DA
   8. Z→ AD
   9. D→a
   10. C→b

Q10. [10] In the grammar G= ({S, A, B}, {*a, b*}, S, P) with the following productions,

S →*aA*B, A → *b*B*b*, B → A | λ

1. [5] Generate the simplified equivalent simplest grammar.
   * 1. S →*aA*A
     2. A → *b*B*b*
     3. B → A
2. [5] Give the language, L(G), that is generated by G, in a formal expression.
   1. L={ab2n|n>=1}

Q11. [10, optional] The reverse of a string can be defined recursively as:

for all *a*∈Σ, w∈Σ\*.

Prove that **,** for all *u, v* ∈, by Mathematical Induction.

* 1. Base Case:
     1. Let u be an arbitrary string of length 0. u = ∈ since there is only one such string. Then (uv) R = (∈v) R = v R = v R ∈ = v R ∈ R = v Ru R
  2. Inductive Step:
     1. Let u be an arbitrary string of length n>0. Assume inductive hypothesis holds for all strings w of length < n
     2. Since |u| = n>0 we have u = ay for some string y with |y| < n and a ∈
     3. Then
     4. (uv)R = ((ay)v) R
     5. = (a(yv))R -associativity of concat
     6. = (yv) R a R -definition of reversal
     7. = (v R y R )a R – induction since |y|<|u|
     8. = v R (y R a R ) -associativity of concat
     9. = v R (ay) R -definition of reverse
     10. = v Ru R -since u=ay