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: Sam Dressler

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;

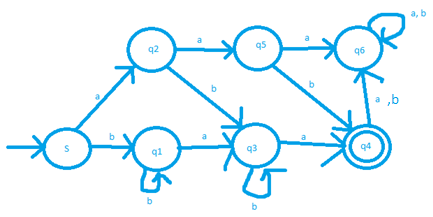
Difficulty:

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

Time Taken:

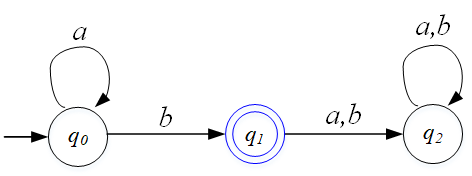
\_\_~7\_\_Hours and \_\_\_\_00\_\_\_\_\_ Minutes.

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 is 7.

i.e L = baa, aba, aab, baba,abbbba bbbaa, babbbbbbbba, ….

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

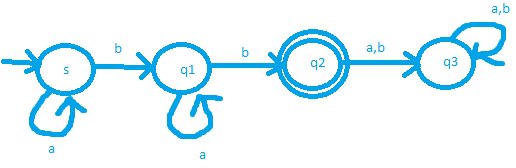


L = L(a\*b)

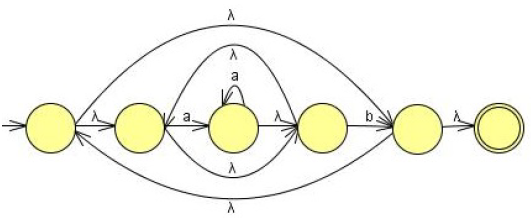
L2 = L(a\*ba\*b)

Lc = L(a\*b(a+b)\*)

L2- L = L2 ⋂ Lc



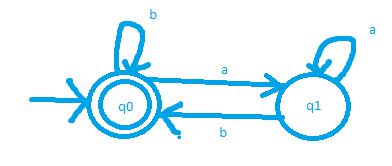
Q3. [15] In the given NFA M of the figure, ty dr.



1. [5] Give the language **L(M)** that is accepted by M in the simplest ***regular expression***.

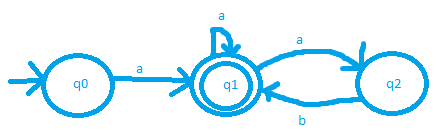
L = ((aa\*)b)\*)

1. [10] Construct a ***minimal DFA*** that is equivalent to the NFA M.



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

1. [5] Construct a (minimal) NFA with a *single final* state that accepts L.



1. [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

S -> A

A -> aA |B | λ

B -> abA | aA

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.

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*.

L is not regular.

Disprove using pumping lemma

*A language is regular if the following conditions are satisfied:*

*1. |xy| <= m*

*2. |y| >= 1*

*3. xyiz is included in the language for i >= 0*

Let n = 1, l = 1, k=2

String w = a1b1a2 = abaa

Let m = |w| = 4

Split w into 3 parts: x, y, z

x = a

y = b

z = aa

**Case 1**:

m = 4

|xy| = 2 < m, so condition is satisfied.

**Case 2**:

|y| = |b| = 1 which is >= 1, so this condition is satisfied.

**Case** 3:

xyiz is included in the language for i>= 0

let i =2

so xy2z = abbbbaa

|x| = a = 1

|y| = |y2| = bbbb = 4

|z| = |aa| = 2

since |y| and |z| must be equal for this language to be regular, this condition is not satisfied, and the language 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.

*Split L into two parts, La and Lb*

La = {anbmck | n=m}

Lb = {anbmck | m != k}

*The CFG for each can be given by:*

La = L(Ga) and Lb = L(Gb) for Grammars Ga and Gb

Derivations for Ga and Gb can be given by:

S1 -> S1 | S2

Set of productions for Ga:

S1 -> aS1b | λ,

Set of productions for Gb:

S2 -> bBS2C| BS2Cc | λ

Set of Productions for G where L = L(G)

S -> S1C | AS2

S1 -> aS1b | λ,

S2 -> bBS2C| BS2Cc | λ

A -> aA | λ

B -> bB| λ

C -> cC | λ

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.

L(G) = { s∈ (a,b) | na(s) = nb(s) = m for some m >= 0}

[5] Decide if the G is ***ambiguous*** or not. Justify your answer.

Let w = baba

S -> SS -> bSabSa -> baba

S -> bSa -> baSba -> baba

Since we can form the string w using two different derivation trees, the language is **ambiguous**.

1. [5] If G is ambiguous, give an unambiguous grammar. Otherwise, show that G is inherently ambiguous.

G is *not inherently ambiguous* since there exists a grammar that produces the language with a single derivation tree.

S -> bAS | aBS | λ

A -> a | bAA

B -> b | aBB

i.e. w = baba

S -> bAS -> baS -> babAS -> babaS -> baba

i.e. w = aaabbabb

S -> aBS ->aaBBS -> aaaBBBS -> aaabBBS -> aaabbBS -> aaabbaBBS -> aaabbabBS -> aaabbabbS -> aaabbabb

The grammar provided is unambiguous therefore the language is inherently ambiguous.

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.

***Eliminate λ-transitions:*** *A-> λ & B ->* *λ*

S -> baAB | baA | baB | ba

A -> bAB | bA | bB | b

B -> BAa | A| Ba | Aa | a

***Eliminate unit-productions:*** *B -> A by replacing A in B with all the productions from A*

S-> baAB | baA| baB | ba

A-> bAB | bA | bB | b

B -> BAa| bAB | bA | bB| b | Ba | Aa | a

***Convert to Chomsky Normal Form***

S -> ZR | ZA | ZB | YX

A -> WB | YA | YB | b

B -> BQ | WB | YA | YB | b | BX | AX | a

Q -> AX

R -> XB

T -> XA

W -> YA

X->a

Y->b

Z -> YX

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.

S -> aA

A -> bAb | bb

1. [5] Give the language, L(G), that is generated by G, in a formal expression.

L (G)= {a(bb)nbb | n >=0}

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.

**Base case:**

Let |u| ,|v| = 1

Then (uv)R = (uv)R = vRuR = vu

**Induction step:**

Induction on |u|

Let u be some string of length n >0

Since |u| = n > 0

we have u = ay for some string y with |y| < n and a∈Σ

then (uv)R = ((ay)v)R = (a(yv))R = (yv)R aR = (vRyR)aR = vR(yRaR) = vR(ay)R = vRuR