COMP 615 (Fall 2021): HW 1
Name: Hossein Alishah
Student ID: 202048619
Section: Mon (7:00 pm – 9:45 pm)

COMP 615 HW 1: Review of Undergraduate Material – Regular

Problem 1.(10 points)
A={a,1, b,2}, B={a, b, c, d}, C={1,2,3,4}.
Determine each of the following:
•A∩(B∪C) = { a, 1, b, 2} = A
•(A∩B)∪C: {a, b} ∪ C = {a, b, 1, 2, 3, 4}
•(A∩B)×C: {a, b} × {1, 2, 3, 4} = { (a, 1), (a, 2), (a, 3), (a, 4), (b, 1), (b, 2), (b, 3), (b, 4)}
•(AΔB)\C: ((A\ B) ∪ (B\ A)) \ C = {1, 2, c, d} \ C = {1, 2, c, d} \ {1, 2, 3, 4} = { c, d}
•$2^{A\cap C}$= {{ }, {1}, {2}, {1, 2}}
Notice: AΔB := (A \ B) ∪ (B \ A)
(AΔB)ΔC = AΔ(BΔC)
(A\B)∩C=(A∩C)\(B∩C)
A Δ B = (A - B) U (B - A)

Problem 2. (10 points)
Let R= {(1,1), (1,2), (1,3), (2,1), (2,3), (3,1)} be a relation over X= {1,2,3}.
State and briefly justify whether R is:
•Reflexive?
It is not reflexive, because in a relation on a set if all ordered pairs (a, a) for every a∈ A appears in a relation, R is called reflexive, here we don't have this pairs { (2,2), (3,3), (4, 4)} to be a reflexive relation.
•Symmetric?
It is not Symmetric, because a relation is symmetric if, for every (a, b) ∈ R, then (b, a) ∈ R, in this question, we have (2, 3) but we don't have (3, 2)
•Transitive?
It is not Transitive, because relation R on a set X is transitive if (a, b) ∈ R and (b, c) ∈ R then (a, bc) ∈ R, here we have (2, 1) ∈ R and (1, 2) ∈ R but (2, 2) ∉ R
•Antisymmetric?
It is not antisymmetric, because a relation is antisymmetric if, for every (a, b) ∈ R, then (b, a) ∈ R is true only when a = b, in this question we have (2, 1) ∈ R and (1, 2) ∈ R but 2 ≠ 1.

Problem 3. (10 points)
Use induction to prove $5n\le 3^n + 2$ for all $n\ge 1$.
Proof: $5n\le 3^n + 2$ for all $n\ge 1$

1) Base Case: show P(1): n=1 ➔ $5(1) \le 3^{(1)} + 2$ ➔ $5 \le 5$

2) Inductive Hypothesis: Assume that P(k) is true, $\forall$ k$\geq$1 ➔ $5k \leq 3^k + 2$

3) Inductive step (prove): Show P(k+1) is true: $5(k+1) \leq 3^{k+1} + 2$

By assumption:

$$5k \leq 3^k + 2$$

Add +5 to both side:

$$5k + 5 \leq 3^k + 2 + 5 \qquad \text{(I)}$$

$3^{k+1} + 2 = 3.3^k + 2 = 3^k + 3^k + 3^K + 2$

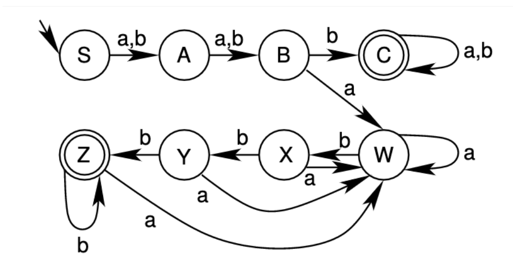
$\forall$ k$\geq$1: $\quad 3^k + 2 + 2(3^K) \geq 3^k + 2 + 5$

$$3^{k+1} + 2 \geq 3^k + 2 + 5 \qquad \text{(II)}$$

(I), (II) : $5k + 5 \leq 3^k + 2 + 5 \leq 3^{k+1} + 2$

$\therefore \quad 5(k+1) \leq 3^{k+1} + 2$

Problem 4. (10 points)
For the DFA below:



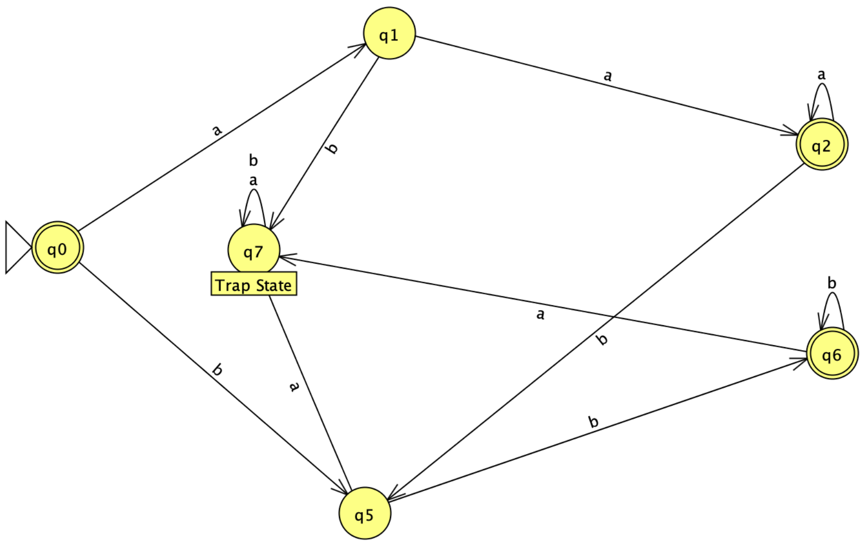
- list 3 words accepted: **bbb, aaaabbb, bbbaaaaa, baba**
- list 3 words rejected: **lambda, aaa, aaaaaabba, abab**
- give a simple description of the language (i.e., the set of words that are accepted):

RE: (a+b)(a+b)b(a+b)*+(a+b)(a+b)aa*b(aa*b)*b(aa*b(aa*b)*b)*b(b+aa*b(aa*b)*b(aa*b(aa*b)*b)*b)*

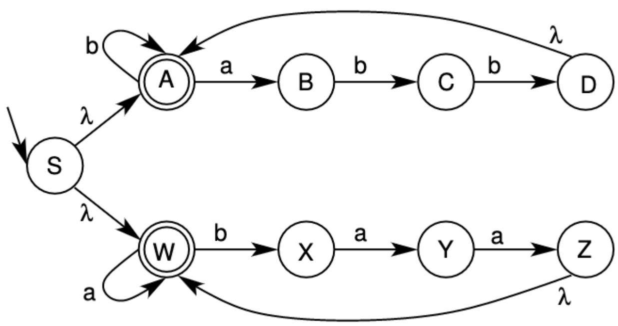
Description:

The string size must be greater than two and it ends to at least three b's if the third symbol is "a" otherwise it could be any number of a's and b's.

Problem 5. (10 points)
Create (draw or give a mathematical description) of a DFA that accepts the language $L=\{a^i b^j : i \neq 1 \text{ and } j \neq 1\}$

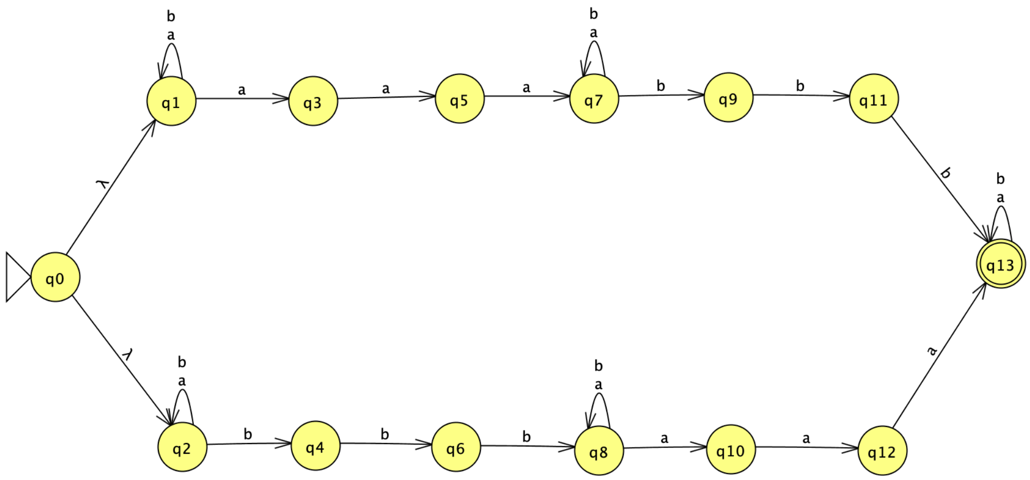


Problem 6. (10 points)
For the NFA below:

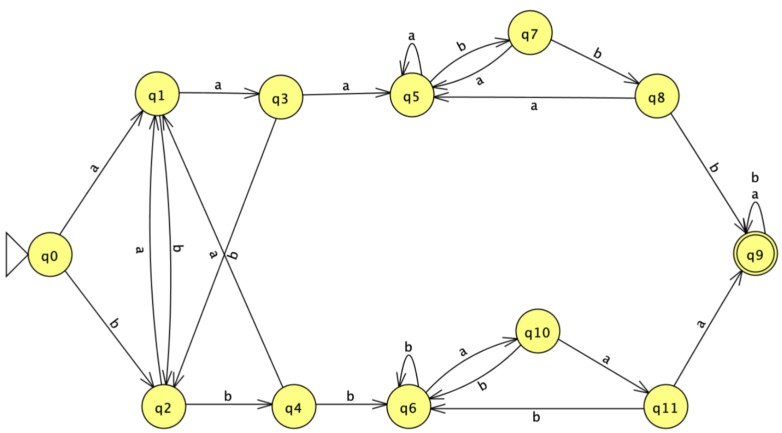


list 3 words accepted: lambda, a, b, abbabbabb, baabaabaa, aaaa, bbbb
list 3 words rejected: aaaab, bbbbaa, ab, ba, aaaba, bbab
give a simple description of the language (i.e., the set of words that are accepted).
RE: b*+b*abb(b*abb)*b*+a*+a*baa(a*baa)*a*
Description:
This language accepts any number of a's or any number of b's or lambda or any a is followed by two b's or any b is followed by two a's.

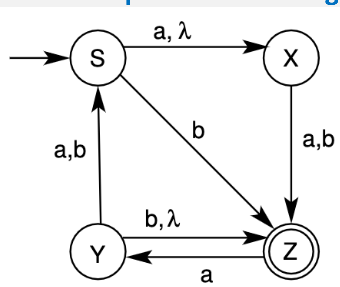
Problem 7. (10 points)
Create (draw or give a mathematical description) of an NFA that accepts the language $\{w\in \{a, b\}^*$: w contains aaa and bbb} (note the 2 substrings could occur in either order).
NFA:

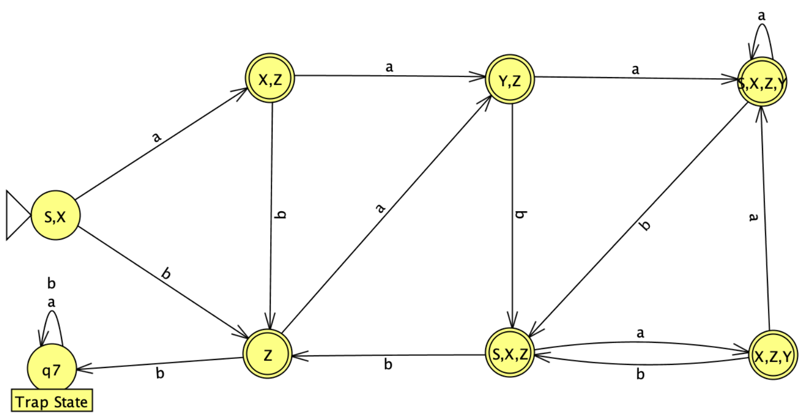


DFA:



Problem 8. (10 points)
Convert the following NFA into a DFA that accepts the same language.



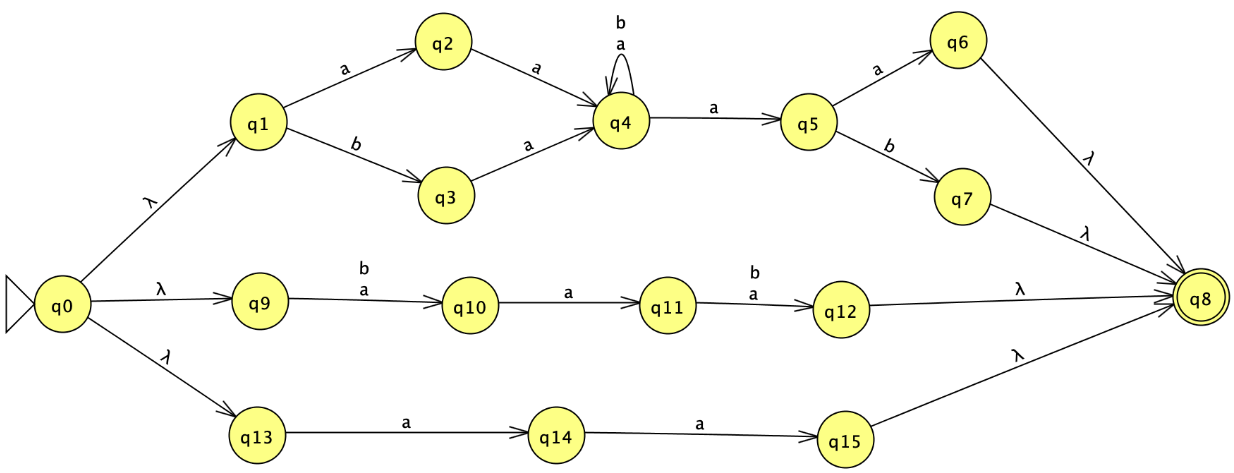


Problem 9. (10 points)
Give regular expressions for both languages below:

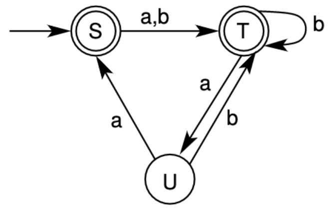
{w∈ {a, b}*: w contains aaa and bbb}
regular expression:
(a+b)*aaa(a+b)*bbb(a+b)* + (a+b)*bbb(a+b)*aaa(a+b)*

{a^{i}b^j: i·j≥5}
regular expression:
aaa*bbbb* + aaaa*bbb* + aa*bbbbbb* + aaaaaa*bb*

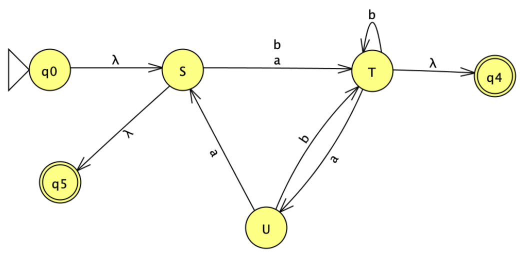
Problem 10. (10 points)
Draw an NFA that accepts the language described by the regular expression (aa+ba)(a+b)*(aa+ab) + (a+b)a(a+b) +aa (ie convert the reg exp into an NFA).



Problem 11.(10 points)
Create a regular expression that describes the language accepted by the DFA below (ie convert the DFA into a regular expression).



Step 1: add three states:



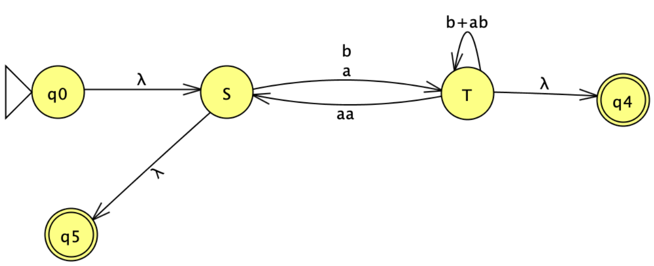
Step 2: omit the state U:

Not U = 1 × 2 = 2

1) TUT:

2) TUS:

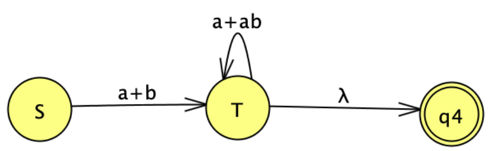
After omitting state U, we have the following DFA:

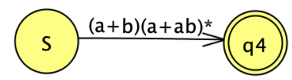


Step 3: not T:

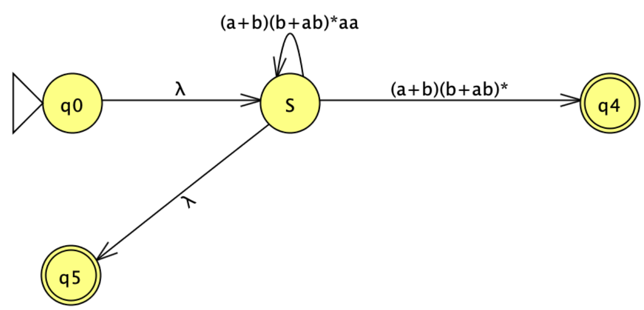
1) STS:

2)STq$_4$:





After omitting state T, we have the following DFA:



Then, the Regular Expression is:

((a+b)(b+ab)*aa)*+((a+b)(b+ab)*aa)*(a+b)(b+ab)*

Problem 12. (10 points)
Give a simple description for the language generated by the following regular grammar.

$$\begin{aligned} S &\rightarrow aaB \mid baB \mid abA \mid bbA \\ A &\rightarrow aA \mid bA \mid aa \mid ab \\ B &\rightarrow aB \mid bB \mid ba \mid bb \end{aligned}$$

This is a Right Linear Grammar, so it is Regular.

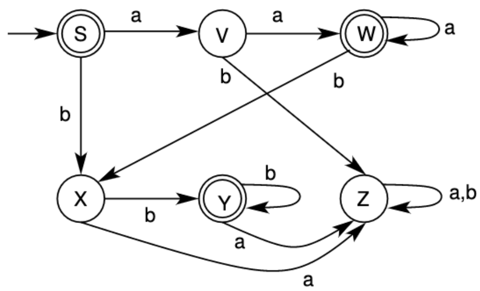
RE: (ab+bb)(a+b)*(ab+aa)+(aa+ba)(b+a)*(bb+ba)

Description:

The size of the string must be more than four and the first character and the last character of the string can be the <u>same</u> or <u>different</u> but the second character and the second from the end character always are <u>different</u>.

(EX: abab, abaa, bbab, bbaa, aabb, aaba, babb, baba}

Problem 13.(10 points)
Give a regular grammar for the language generated by the DFA below.



This is a right-linear Grammar:

S → λ | aA | aB

A → aC | bE

B → bD | aE

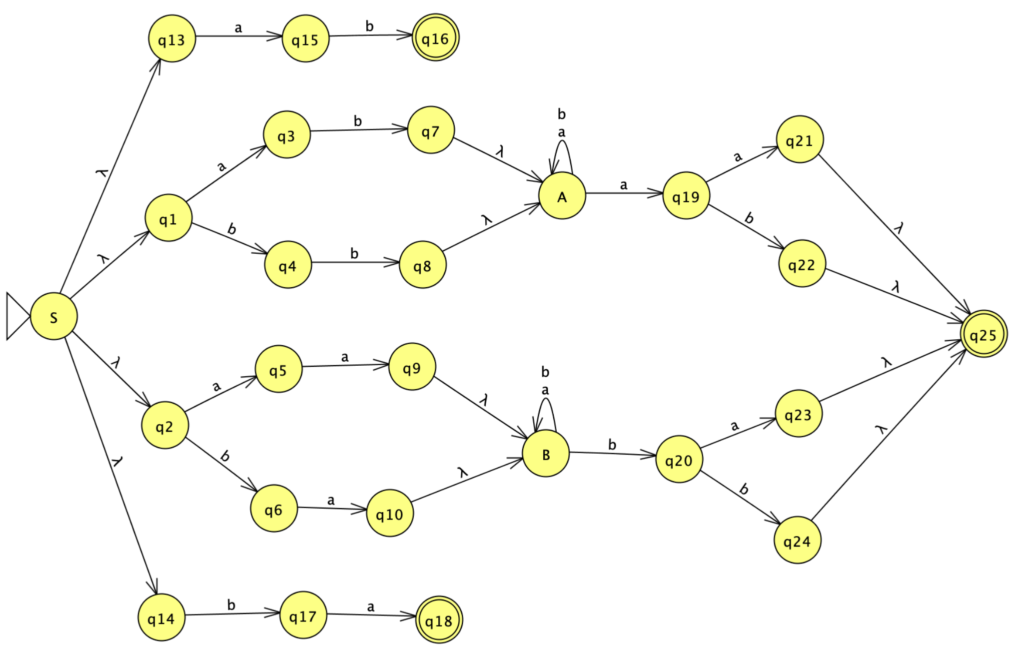
C → λ | aC | bB

D → λ | bD | aE

E → aE | bE

Problem 14. (10 points)
Draw an NFA that accepts the language generated by the regular grammar below.

$$\begin{aligned} S &\rightarrow aaB \mid baB \mid abA \mid bbA \mid ab \mid ba \\ A &\rightarrow aA \mid bA \mid aa \mid ab \\ B &\rightarrow aB \mid bB \mid ba \mid bb \end{aligned}$$



Problem 15. (10 points)

Use the pumping lemma (for regular languages) to show that $\{w \in \{a, b\}^*: \#a > \#b\}$ is not a regular language.

Assume $\{w \in \{a, b\}^*: \#a > \#b\}$ is regular.

Pumping lemma says there is a $N \geq 0$ for this language.

We pick a string $w = b^N a^{N+1}$ in the language with at least N character long and #a's is more than #b's.

Let's look at all the decompositions of w into three pieces xyz according to the following rulls:

- **$N \geq 0$**
- **$W = xyz$**
- **$|xy| \leq N$**
- **$|y| \neq 0$**
- **$x = b^k$**
- **$y = b^l$**
- **$z = b^{N-K-L} a^{N+1}$**
- **$K + L \leq N$**
- **$L \neq 0$**

look at xy^tz and see what value of t allow us to leave the language:

$$xy^tz = b^K b^{tL} b^{N-K-L} a^{N+1}$$

$$= b^{N+tL-L} a^{N+1} \qquad \text{(i)}$$

(i) is only going to be in the language if #a's $\geq$ #b's:

$$\text{So, in L iff } N+tL-L < N+1 \quad \Leftrightarrow \quad tL - L < 1$$

$$\Longrightarrow \quad L(t-1) < 1$$

We know the $L \geq 0$, so the only way that the above inequality to be true is t =1, but we pick t= 3 because t can be any number of more than 0, so in this case this inequality would be false. Therefore, this language is not Regular.

Problem 16. (10 points)

For any language L define $Two(L) = \{w_1w_2: w_1, w_2 \in L\}$. In other words, Two (L) is the language consisting of all strings created by concatenating two words in L. Show that if L is regular then Two(L) is regular.

We are going to show that if L is regular then Two(L) is regular, that any words in Two(L) is created by concatenating two words in L.

We have two DFA's (M_1, M_2) that accept w_1 and w_2:

$M_1 = (Q_1, \Sigma, \delta_1, q0_1, F_1)$

$M_2 = (Q_2, \Sigma, \delta_2, q0_2, F_2)$

The product DFA for concatenating w_1w_2 is

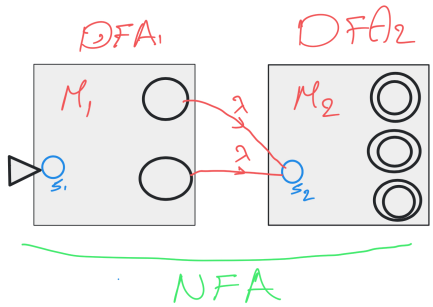
$M = (Q, \Sigma, \delta, q0, F)$

- $Q = Q_1 \times Q_2$
- $q0 = (q0_1, q0_2)$

- $\delta = ((q_1, q_2), a) = (\delta_1 (q_1, a), \delta_2 (q_2, a))$
  for all $q_1 \in Q_1$ and $q_2 \in Q_2$

$$\underbrace{W_1 \ldots\ldots}_{\in M_1} \mid \underbrace{W_2 \ldots\ldots}_{\in M_2}$$

Important: we know that each DFA has an equivalence NFA.



For example, M_1 has two final states and M_2 has three final states, as you see in the above picture, we mark the original final sates in M_1 and connect them with lambda transition to the start state of M_2.
So, the only possible way that the machine accepts a string is starting at the start state of the M_1 and go through the M_1 and land in original final states, then take the lambda transitions and finally read the string that M_2 accepts.