Definitions:
RE: A regular expression is a notation that involves a combination of strings of symbols from some alphabet Σ, parentheses, and the operators +, ., and *.
DFA: A deterministic accepter has internal states, rules for transitions from one state to another, some input, and ways of making decisions.
A deterministic finite accepter or dfa is defined by the quintuple M = (Q,Σ,δ,q0 , F),
where Q is a finite set of internal states,
Σ is a finite set of symbols called the input alphabet,
δ :Q × Σ → Q is a total function called the transition function,
q0 ∈ Q is the initial state, F ⊆Q is a set of final states.
NFA: A nondeterministic finite accepter; there are three major differences between the definition of an nfa and the definition of a dfa:

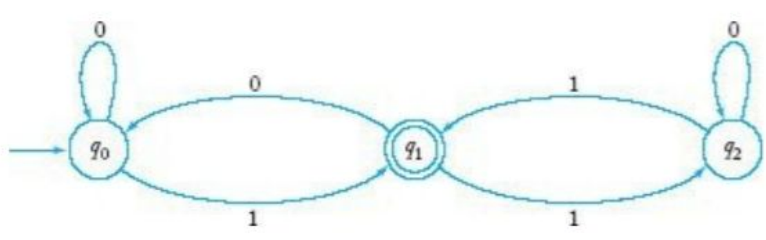
1. The range of δ is in the powerset 2Q, so that its value is not a single element of Q but a subset of it.
2. we allow λ as the second argument of δ. This allows nfa to make a transition without consuming an input symbol.
3. The set δ (qi ,a) may be empty, meaning that there is no transition defined for this specific situation.

RG: A regular grammar is one that is either right-linear or left-linear. A linear grammar is a grammar in which at most one variable can occur on the right side of any production, without restriction on the position of this variable. Clearly, a regular grammar is always linear, but not all linear grammars are regular.
RL: A language L is called regular if and only if there exists some deterministic finite accepter M such that L= L(M).

Ch.2 Linz 2.1: 1, 2. (a), (b), (c), 5. (a) (b); 2.2: 2, 3, 6, 7, 9, 10, 11; 2.3: 8

2.1

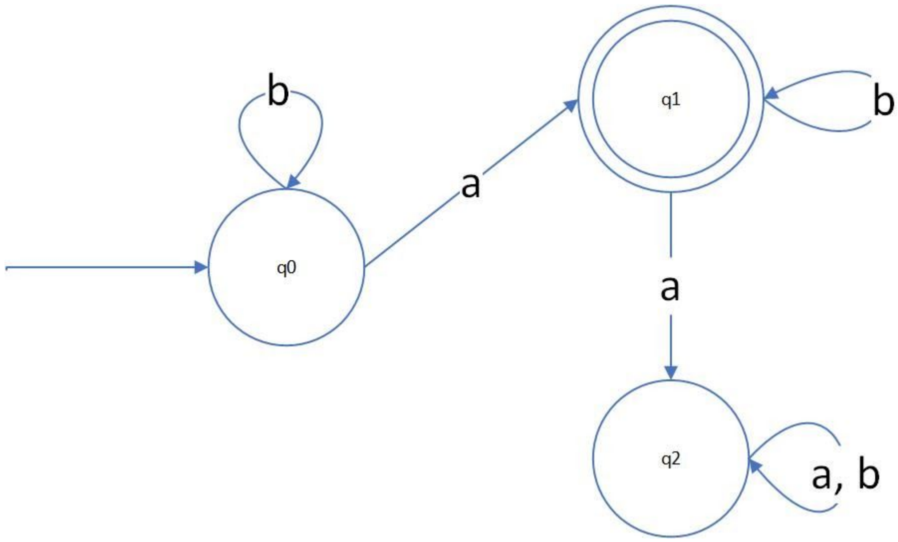
1. Which of the strings 0001, 01001, 0000110 are accepted by the dfa in Figure 2.1?



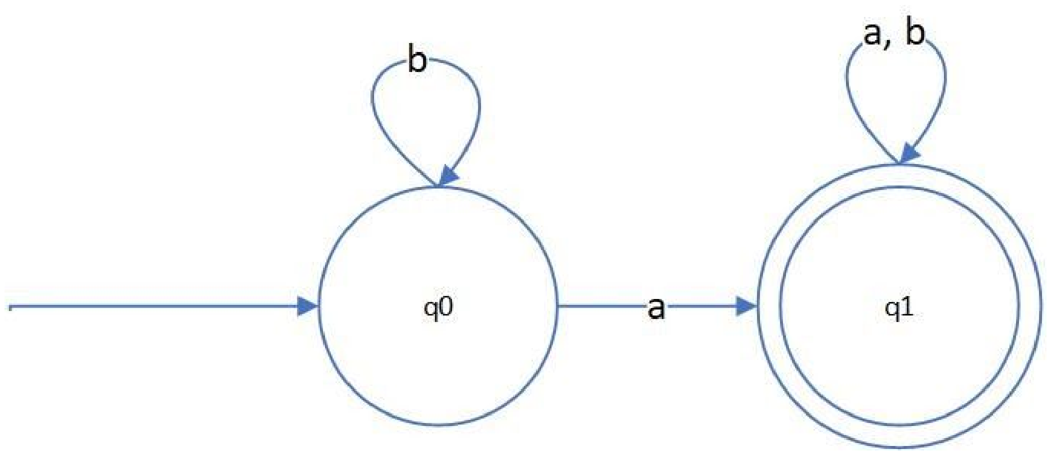
The string “0001” and “01001” are accepted by the dfa. The string has to end with one 1 or n*2 number of 1’s

2. For Σ= {a,b}, construct dfa's that accept the sets consisting of:

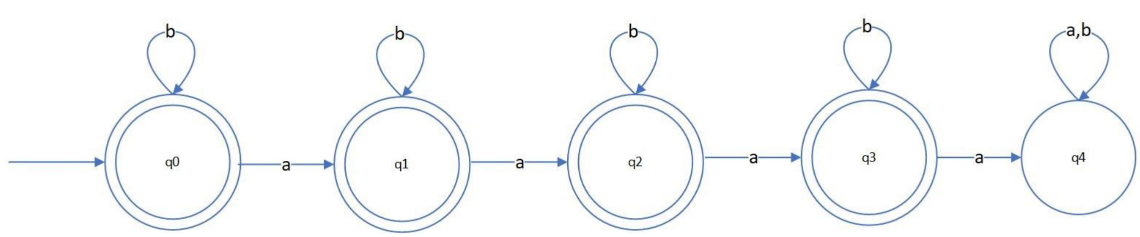
a) all strings with exactly one a



b) all strings with at least one a,

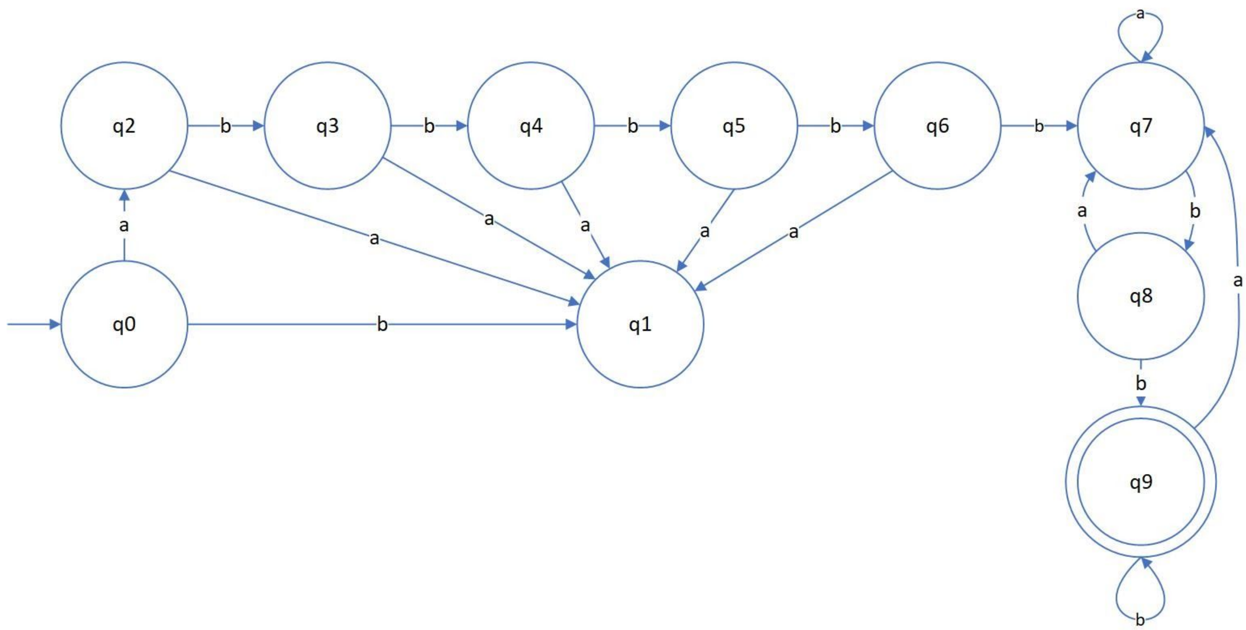


c) all strings with no more than three a's

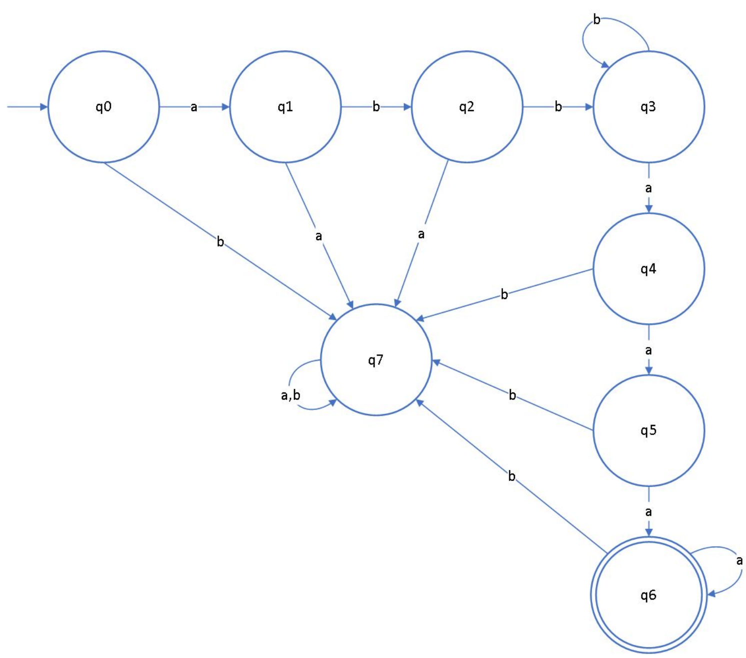


5. Give dfa's for the languages

a) L= {ab^5wb^2 : w ∈ {a,b} *},



b) $L = \{ab^na^m : n \geq 2, m \geq 3\}$

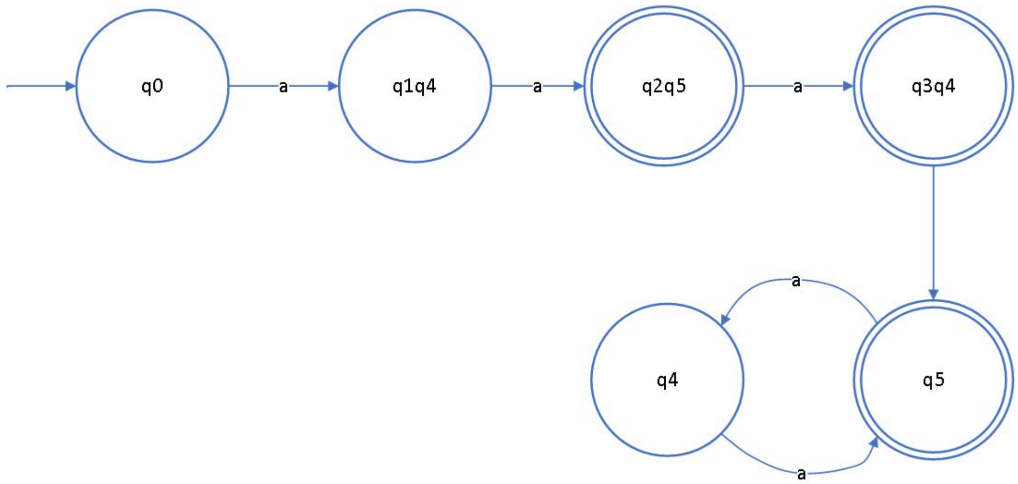


2.2

2. Find a dfa that accepts the language defined by the nfa in Figure 2.8

	a
q0	q1, q4
q1	q2
q2	q3
q3	-
q4	q5
q5	q4

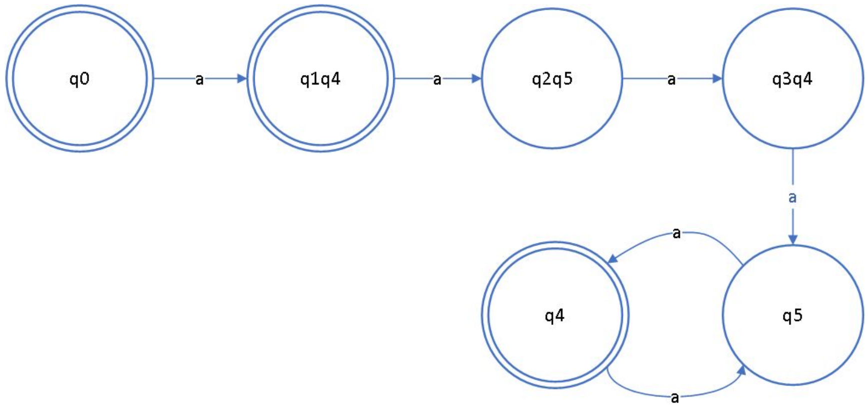
	a
q0	q1q4
q1q4	q2q5
q2q5	q3q4
q3q4	q5
q5	q4
q4	q5



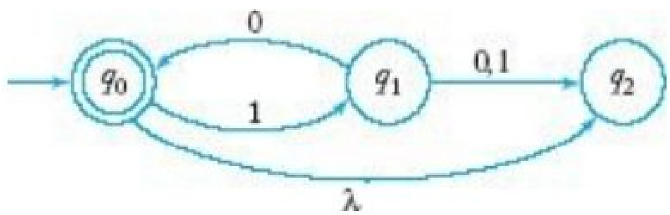
3. Find a dfa that accepts the complement of the language defined by the nfa in Figure 2.8.

	a
q0	q1, q4
q1	q2
q2	q3
q3	-
q4	q5
q5	q4

	a
q0	q1q4
q1q4	q2q5
q2q5	q3q4
q3q4	q5
q5	q4
q4	q5

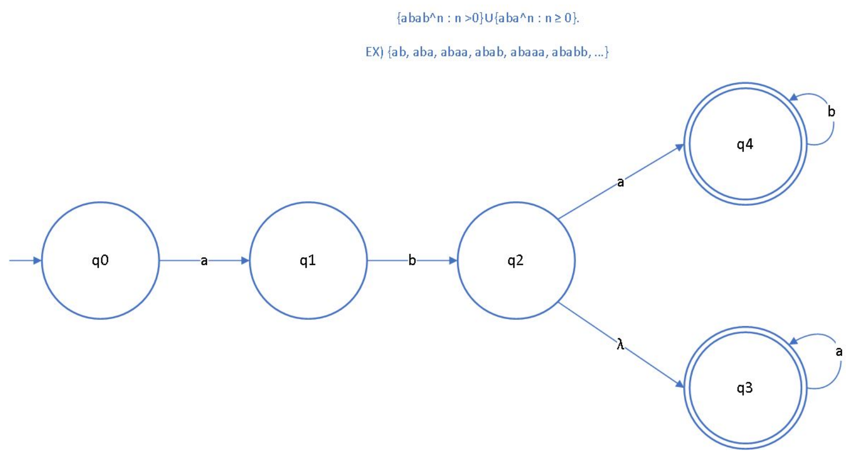


6. For the nfa in Figure 2.9, find δ * (q0 , 1010) and δ * (q1 ,00)



δ * (q0 , 1010)	δ * (q1 ,00)
(q0,q1,q2)	(q0,q1,q2)

7. Design an nfa with no more than five states for the set $\{abab^n : n > 0\} \cup \{aba^n : n \geq 0\}$.

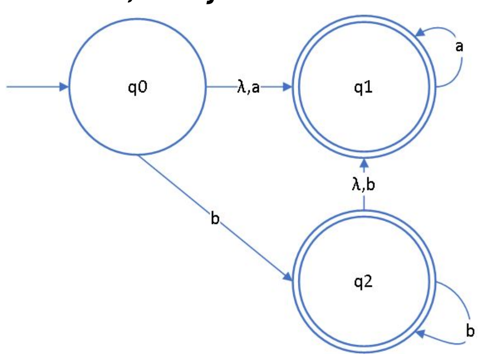


9. Do you think Exercise 8 can be solved with fewer than three states?

No, it is not possible to solve with fewer than three states because there are three letters.

10.(a) Find an nfa with three states that accepts the language:

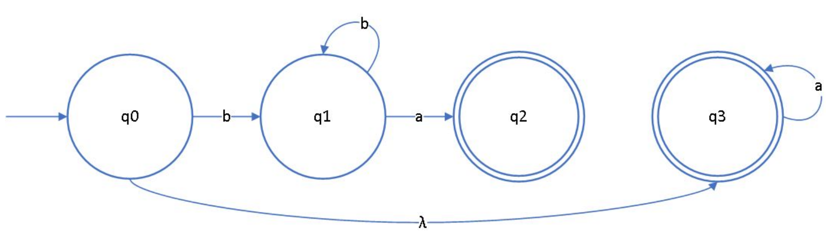
$L= \{a^n : n \geq 1\} \cup \{b^m a^k : m \geq 0, k \geq 0\}$



(b) Do you think the language in part (a) can be accepted by an nfa with fewer than three states?

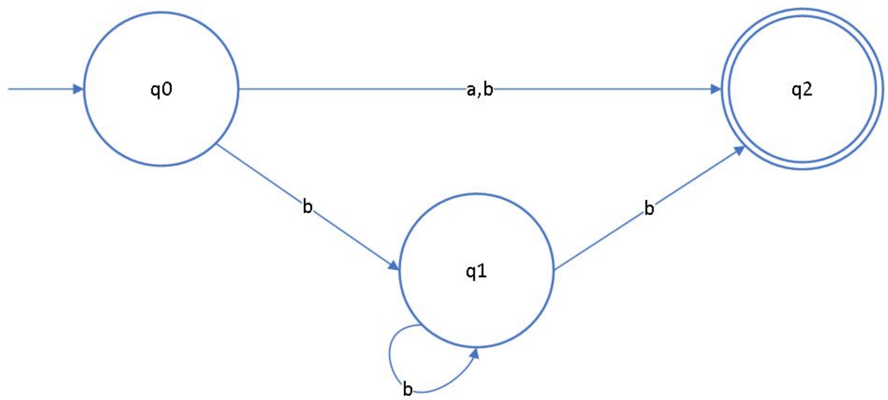
Yes, I think it can be accepted by two states. By eliminating q2 and making a self loop on q0 where the loop is “b”, then you will be able to accept the language.

11. Find an nfa with four states for $L= \{a^n : n \geq 0\} \cup \{b^n a : n \geq 1\}$.



2.3

8. Find an nfa without λ-transitions and with a single final state that accepts the set $\{a\} \cup \{b^n : n \geq 1\}$.



Ch.3 Linz 3.1: 1, 4, 5; 3.2: 3, 10(a); 3.3: 1, 2

3.1

1. Find all strings in L((a + b) b (a + ab)*) of length less than four.

    {ab, bb, aba, bba}

4. Find a regular expression for the set $\{a^n b^m : n \geq 3, m \text{ is even}\}$.

    {aaabb, aaaabb, aaaaabb, aaabbbb, aaaabbbb, aaaaabbbb, …}

    - Has to start with 3 a's so (aaa)
    - There can be more a's that follow after the 3, so (aaa)a*
    - There has to be an even amount of b's, so force two b's : (bb)*
        - $(bb)^0 = \epsilon$, $(bb)^1 = \{bb\}$, $(bb)^2 = \{bbbb\}$

    Therefore the answer is...

    (aaa)a*(bb)*

5. Find a regular expression for the set $\{a^n b^m : (n + m) \text{ is even}\}$.

    - n and m have to both be either even or odd in order for sum to be even
    - {ϵ, aa, bb, ab, abbb, aaab, aabb, …}
    - If there is an "a" it will always precede the "b"
    - If there is no leading "a", there must be even amount of "b's"
    - It's either even or odd so for even amounts of "a" and "b" : (aa)*(bb)*
    - We need to handle odd number of a's : a(aa)*
    - And odd number of b's: b(bb)*

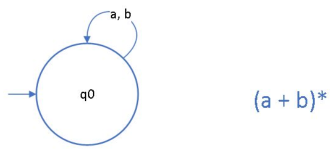
    Therefore we have:

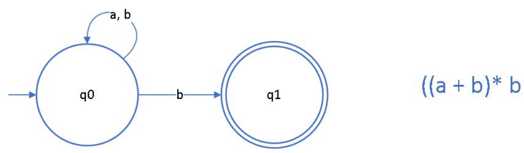
    (aa)*(bb)* + a(aa)*b(bb)*

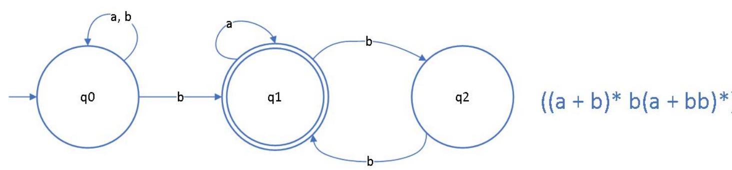
3.2

3. Give an nfa that accepts the language L((a + b)* b(a + bb)*).

L={ b, ba, ab, bbb, aba, abbb, bbbb}

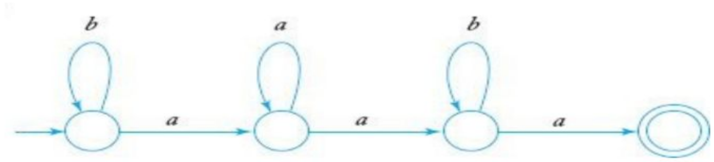






10. Find regular expressions for the languages accepted by the following automata.

(a)



- The first state can accept b ≥ 0 and moves to next state on an "a". (b*a)
- The next state accepts a ≥ 0 and also moves forward on an "a". (a*a)
- The last state is the same as first state.

Therefore the answer is :

(b*a)(a*a)(b*a)

3.3

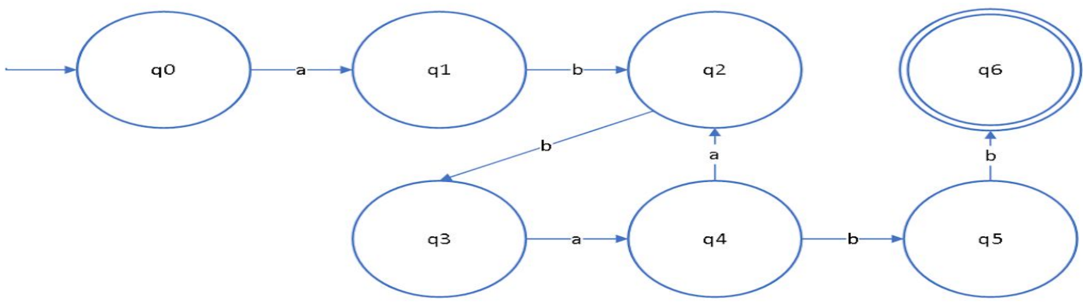
1. Construct a dfa that accepts the language generated by the grammar

S → abA,

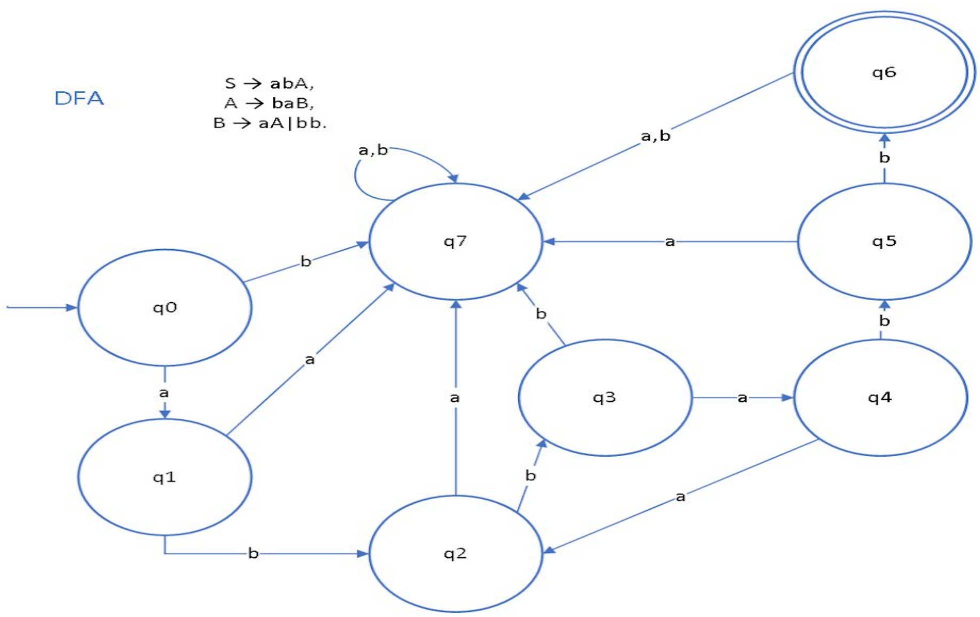
A → baB,

B → aA|bb.

NFA



DFA



2. Find a regular grammar that generates the language L (aa* (ab+ a)*).

S → aA

A → aA|B

B →abB|aB|λ