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

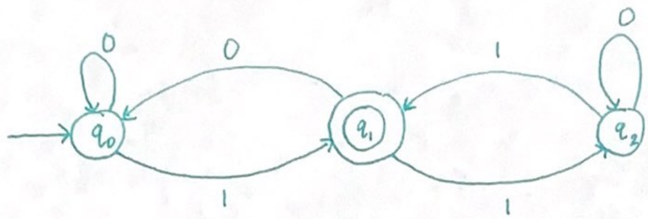


Figure 2.1

String 0001

Loops 3 times at q_0 then finishes at q_1 (the final state)

$\therefore$ String 0001 is accepted by the dfa

String 01001

	current state		next state
01001 :	q_0	$\rightarrow$	q_0
01001 :	q_0	$\rightarrow$	q_1
01001 :	q_1	$\rightarrow$	q_0
01001 :	q_0	$\rightarrow$	q_0
01001 :	q_0	$\rightarrow$	q_1

$\therefore$ String 01001 is accepted by the dfa

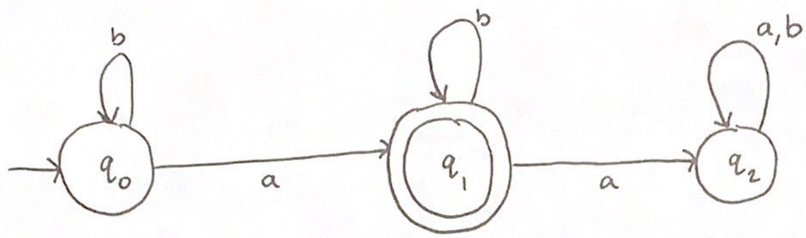
String 0000110

String	current state		next state
0000110	q_0	$\rightarrow$	q_0
0000110	q_0	$\rightarrow$	q_0
0000110	q_0	$\rightarrow$	q_0
0000110	q_0	$\rightarrow$	q_0
0000110	q_0	$\rightarrow$	q_1
0000110	q_1	$\rightarrow$	q_2
0000110	q_2	$\rightarrow$	q_2

q_2 is not the final state $\therefore$ String 0000110 is not accepted by the dfa.

2. For $\Sigma = \{a, b\}$, construct dfa's that accept the sets consisting of

2a. All strings with exactly one a



Test cases:

String a

String	Current State		Next State
a	q_0	$\rightarrow$	q_1

$q_1 \in F$ $\therefore$ the dfa accepts the String a

String ba

String	Current State		Next State
b a	q_0	$\rightarrow$	q_0
b a	q_0	$\rightarrow$	q_1

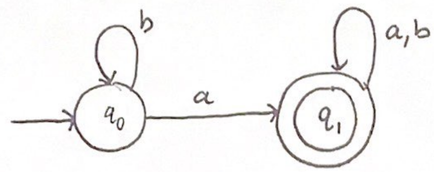
$q_1 \in F$ $\therefore$ dfa accepts the String ba

String baa

String	Current State		Next State
b a a	q_0	$\rightarrow$	q_0
b a a	q_0	$\rightarrow$	q_1
b a a	q_1	$\rightarrow$	q_2

$q_2 \notin F$ $\therefore$ the dfa does not accept the String baa

2b. All Strings with at least one a



<u>Test cases:</u>

<u>String a</u>

String	Current State		Next State
a	q_0	$\rightarrow$	q_1

$q_1 \in F$ $\therefore$ String a is accepted by the dfa

<u>String ab</u>

String	Current State		Next State
ab	q_0	$\rightarrow$	q_1
ab	q_1	$\rightarrow$	q_1

$q_1 \in F$ $\therefore$ String ab is accepted by the dfa

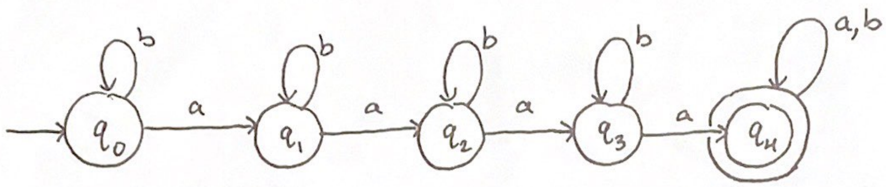
<u>String bb</u>

String	Current State		Next State
bb	q_0	$\rightarrow$	q_0
bb	q_0	$\rightarrow$	q_0

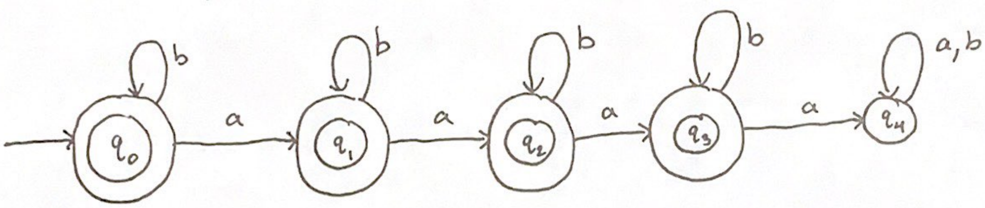
$q_0 \notin F$ $\therefore$ String bb is not accepted by the dfa

2c. All strings with no more than three a's

To simplify the problem, lets design a dfa that accepts all strings with more than three a's.



Now flip each state and keep everything else the same



Test Cases:

String a

String	Current State		Next state
a	q_0	$\rightarrow$	q_1

$q_1 \in F$ $\therefore$ String a is accepted by the dfa

String aba

String	Current State		Next State
aba	q_0	$\rightarrow$	q_1
aba	q_1	$\rightarrow$	q_1
aba	q_1	$\rightarrow$	q_2

$q_2 \in F$ $\therefore$ String aba is accepted by the dfa

String ababa

String	Current State		Next State
ababa	q_0	$\rightarrow$	q_1
ababa	q_1	$\rightarrow$	q_1
ababa	q_1	$\rightarrow$	q_2
ababa	q_2	$\rightarrow$	q_2
ababa	q_2	$\rightarrow$	q_3

$q_3 \in F$ $\therefore$ String ababa is accepted by the dfa

String abaaa

String	Current State		Next state
abaaa	q_0	$\rightarrow$	q_1
abaaa	q_1	$\rightarrow$	q_1
abaaa	q_1	$\rightarrow$	q_2
abaaa	q_2	$\rightarrow$	q_3
abaaa	q_3	$\rightarrow$	q_4

$q_4 \notin F$ $\therefore$ String abaaa is not accepted by the dfa

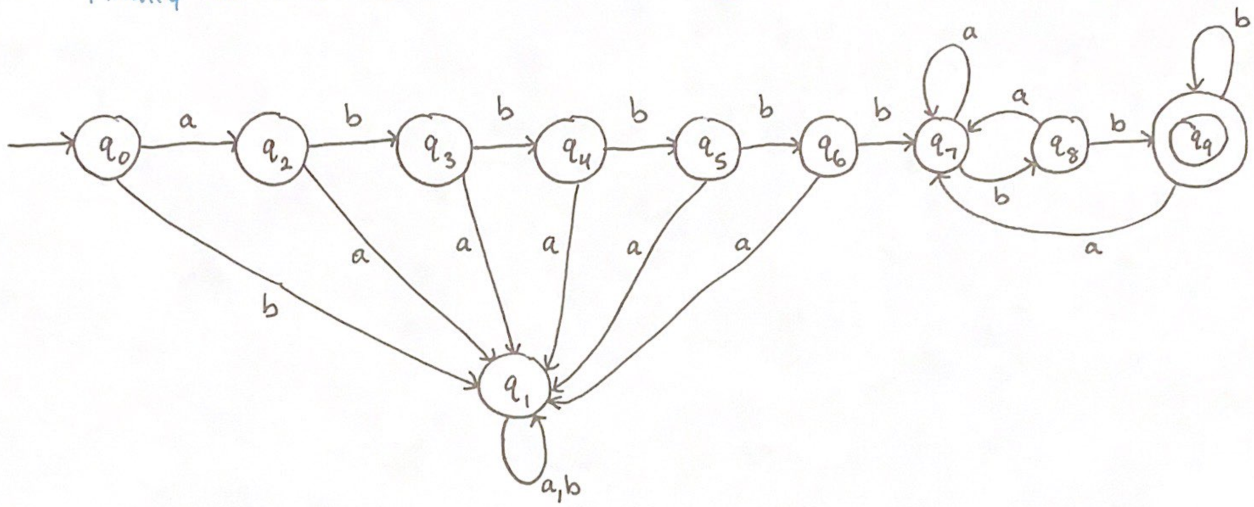
5. Give dfa's for the languges

5a. $L = \{ab^5wb^2 : w \in \{a,b\}^*\}$

We must have a prefix of the form abbbbb

If we receive any symbol out of order, the whole string must be rejected.

Finally we must look for the suffix bb

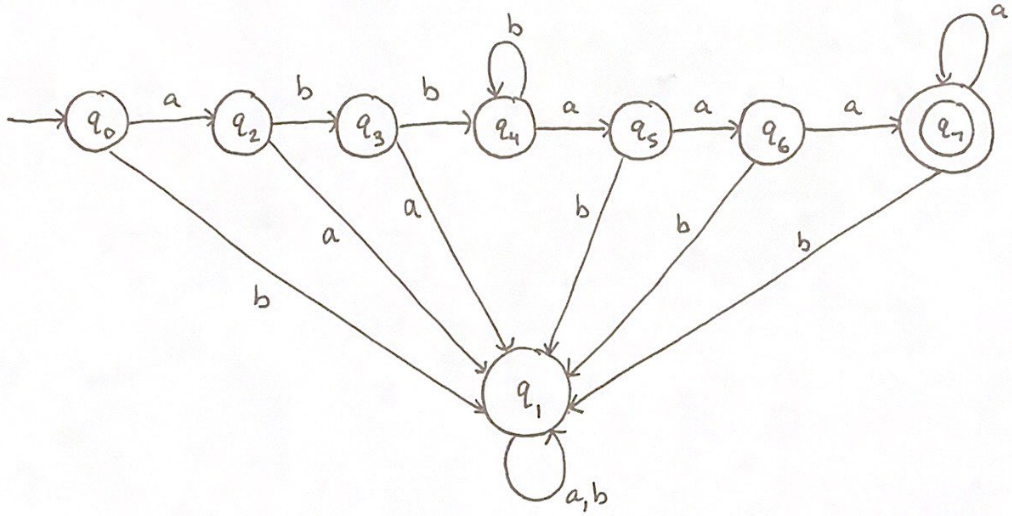


5b. $L = \{ab^n a^m : n \geq 2, m \geq 3\}$

1) We must have the symbol a as the string prefix otherwise the whole string is rejected.

2) After the prefix, at minimum there must be 2 b's consecutively, otherwise the whole string is rejected.

3) Last, there must be at least 3 consecutively after the sequence of b's, otherwise the whole string is rejected.



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

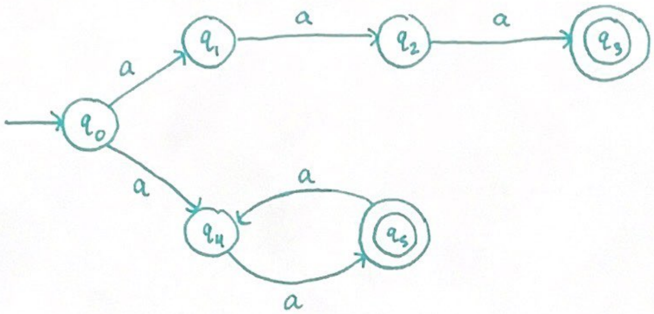


Figure 2.8

From the diagram, $L = \{aaa\} \cup \{a^{2n} : n \geq 1\}$

Remember: We build a DFA out of the subsets of the states in an NFA

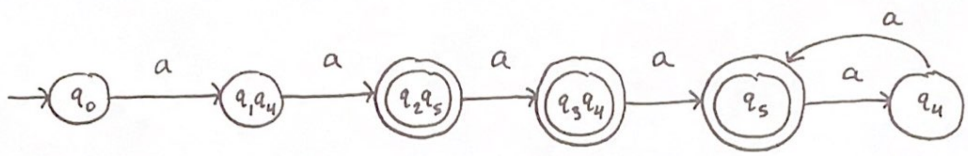
NFA

	a
$\rightarrow q_0$	$\{q_1, q_4\}$
q_1	$\{q_2\}$
q_2	$\{q_3\}$
q_3	$\emptyset$
q_4	$\{q_5\}$
q_5	$\{q_4\}$

Final State

DFA

	a	
$\rightarrow q_0$	$q_1 q_4$	$q_1 \cup q_4$
$q_1 q_4$	$q_2 q_5$	$(q_1 \rightarrow q_2) \cup (q_4 \rightarrow q_5) : q_2 q_5$
$q_2 q_5$	$q_3 q_4$	$(q_2 \rightarrow q_3) \cup (q_5 \rightarrow q_4) : q_3 q_4$
$q_3 q_4$	q_5	$(q_3 \rightarrow \emptyset) \cup (q \rightarrow q_5) : q_5$
q_5	q_4	$(q_5 \rightarrow q_4) : q_4$
q_4	q_5	$(q_4 \rightarrow q_5) : q_5$



To Solve NFA → DFA:

1) Identify the language being accepted by the NFA
2) Build a transition table for NFA
3) Build a transition table for DFA
4) Determine the final state(s) for DFA based on NFA final state(s)
5) Draw DFA

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

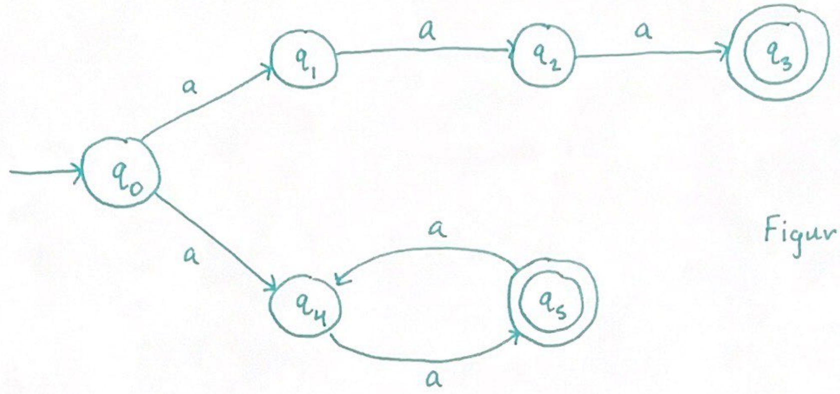
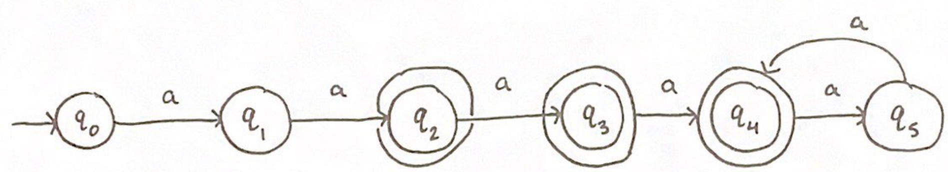
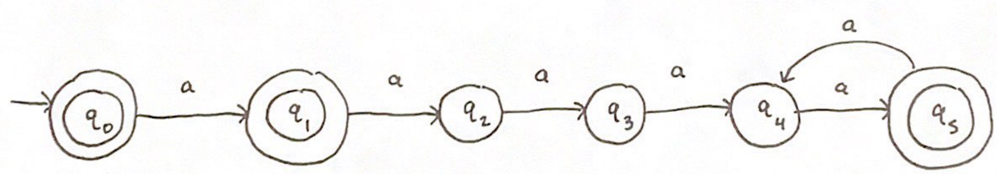


Figure 2.8

From exercise 2:



The complement would flip all states from internal → final or final → internal, but keep all transitions the same



6. For the nfa in Figure 2.9, find $\delta^*(q_0, 1010)$ and $\delta^*(q_1, 00)$.

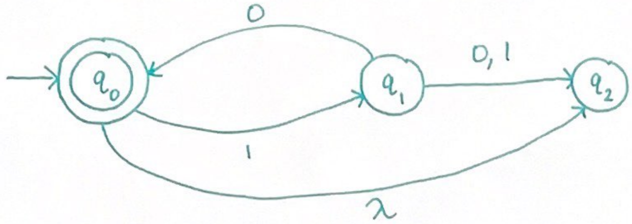


Figure 2.9

Current State	Symbol	Next State
q_0	1	q_1
q_0	λ	q_2
q_1	0	$\{q_0, q_2\}$
q_1	1	q_2

a) $\delta^*(q_0, 1010) = ?$

$\delta^*(q_0, 1) = \delta(q_0, 1) = q_1$

$\delta^*(q_0, 10) = \delta(\delta^*(q_0, 1), 0)$
$= \delta(q_1, 0)$
$= \{q_0, q_2\}$

$\delta^*(q_0, 101) = \delta(\delta^*(q_0, 10), 1)$
$= \delta(\{q_0, q_2\}, 1)$
$= \delta(q_0, 1) \cup \delta(q_2, 1)$
$= \{q_1\} \cup \emptyset$
$= q_1$

$\delta^*(q_0, 1010) = \delta(\delta^*(101), 0)$
$= \delta(q_1, 0)$
$= \{q_0, q_2\}$

$\therefore \delta^*(q_0, 1010) = \{q_0, q_2\}$

b) $\delta^*(q_1, 00) = ?$

$$\delta^*(q_1, 0) = \delta(q_1, 0)$$
$$= \{q_0, q_2\}$$

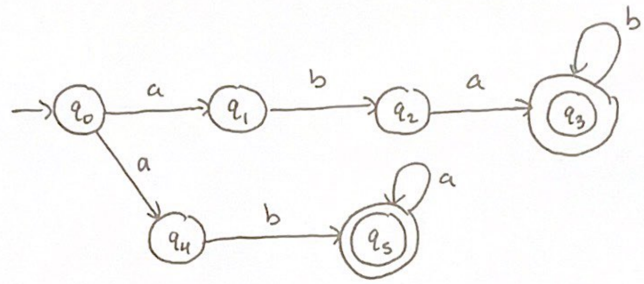
$$\delta^*(q_1, 00) = \delta(\delta^*(q_1, 0), 0)$$
$$= \delta(\{q_0, q_2\}, 0)$$
$$= \delta(q_0, 0) \cup \delta(q_2, 0)$$
$$= \emptyset \cup \emptyset$$
$$= \emptyset$$

$\therefore \delta^*(q_1, 00) = \emptyset$ (Dead State)

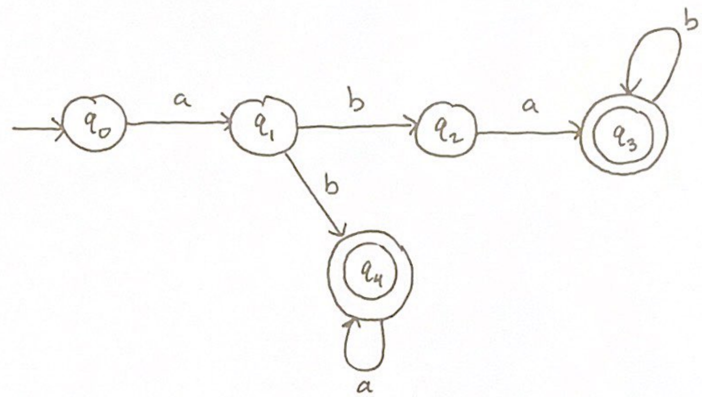
7. Design an nfa with no more than five states for the set

$\{abab^n : n > 0\} \cup \{aba^n : n \geq 0\}$

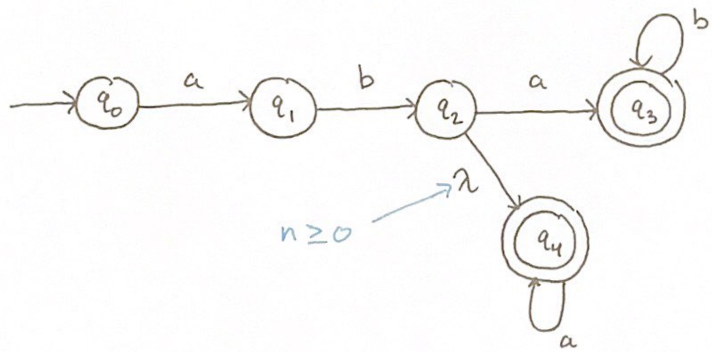
Step 1:



Step 2:

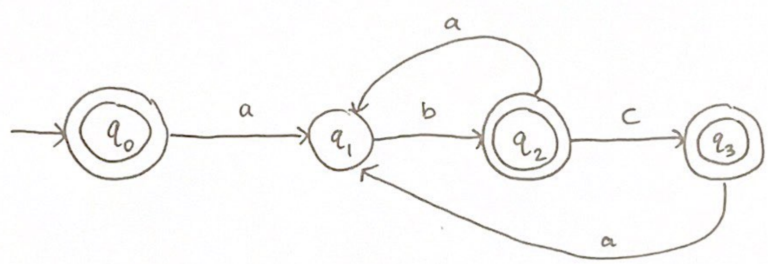


Step 3:

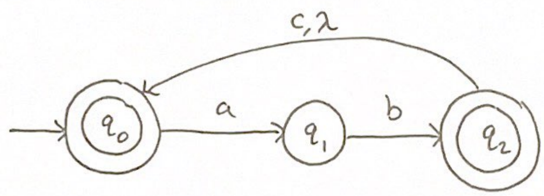


9. Do you think Exercise 8 can be solved with fewer than three states?

8. Construct an nfa with three states that accepts the language $\{ab, abc\}^*$.



Initial Design



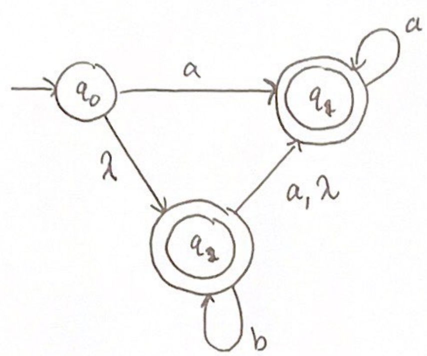
3 - state design

Now using the diagram made for Exercise 8, we can determine if fewer than three states is possible.

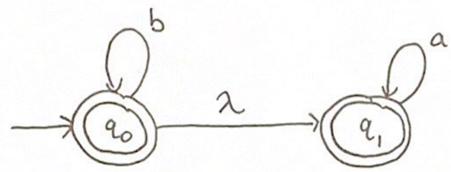
We find that the string abc has three different symbols which cannot be constructed in a 2 - state design.

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

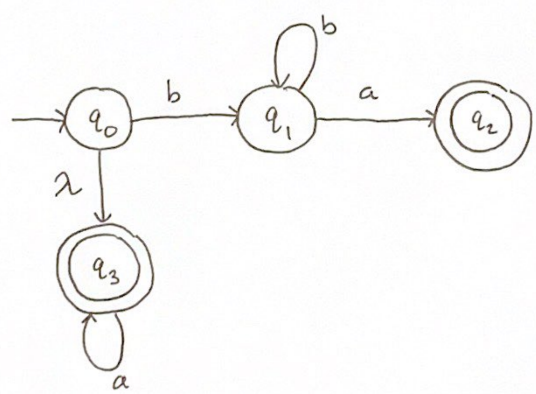
$$L = \{a^n : n \geq 1\} \cup \{b^m a^k : m \geq 0, k \geq 0\}$$



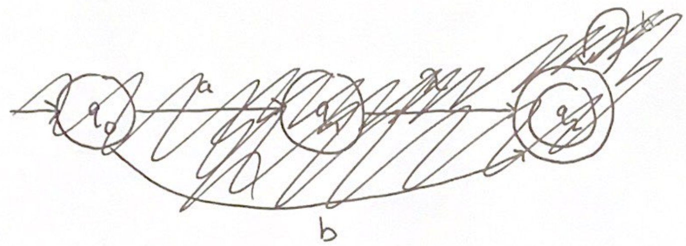
10b. Do you think the languages in part (a) can be accepted by an nfa with fewer than three states?



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

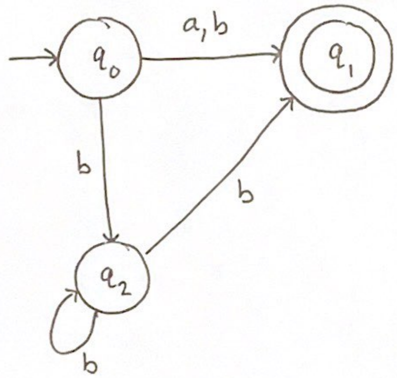


8. Find an nfa without λ-transitions and with a single final state that accepts the set $\{a\} \cup \{b^n : n \geq 1\}$



~~Initial Design~~

~~λ-transition means transition to next state on empty string input~~



Final Design

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

length 0 = Not possible

length 1 = Not possible

length 2 = ab, bb

length 3 = aba, bba

$\therefore$ {ab, bb, aba, bba}

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

$aaa(a^*)(bb)^*$

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

There are 2 cases:

i) Both n and m are even

$(aa)^*(bb)^*$

ii) Both n and m are odd

$a(aa)^*b(bb)^*$

$\therefore$ The regular expression is the combination of both cases

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

Note: Alternative Form

$a(aa)^*b(bb)^* \Rightarrow (aa)^*ab(bb)^*$

$(aa)^*ab(bb)^* + (aa)^*(bb)^*$

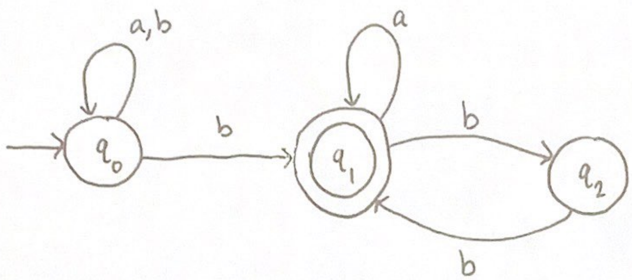
$= (aa)^*ab(bb)^* + (aa)^*\lambda(bb)^*$

$= (aa)^*(ab+\lambda)(bb)^*$

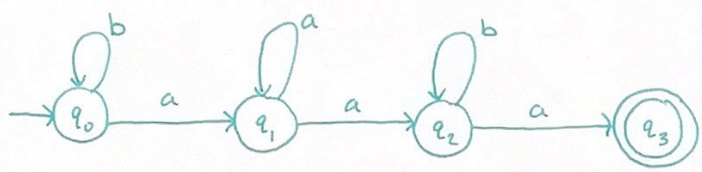
3. Give an nfa that accepts the language $L((a+b)^*b(a+bb)^*)$.

$(a+b)^*$ includes λ

$(a+bb)^*$ includes λ



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



q_0: contains a self-loop on the input string b having length zero or more. Moves to q_1 on the input symbol a

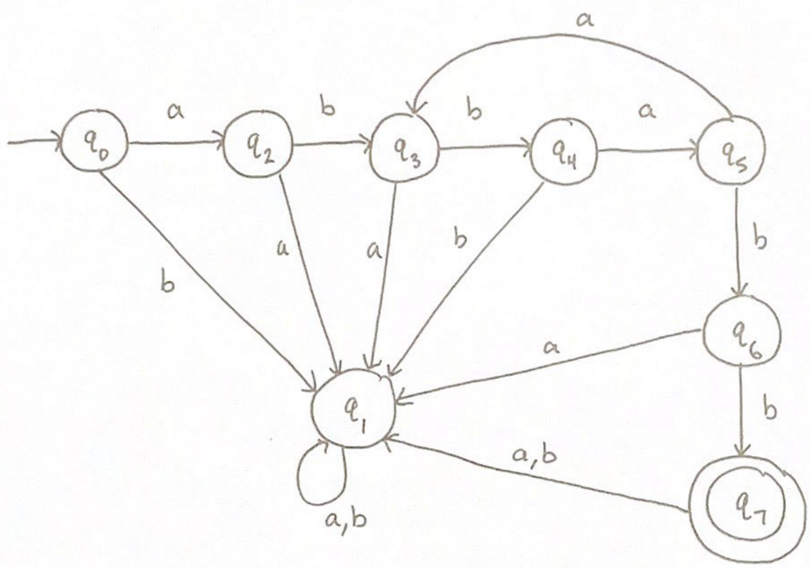
q_1: contains a self-loop and outgoing edge on the input symbol a

q_2: contains a self-loop on the input string b having length zero or more. Moves to q_3, a final state, on the input string a

∴ the regular expression for this automata is $b^*aa^*ab^*a$

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

$$S \to abA,$$
$$A \to baB,$$
$$B \to aA \mid bb$$



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

A grammar $G = (V, T, S, P)$

where $V = \{S, A, B\}$

$T = \{a, b\}$

$P = \{ S \to aA,$

$A \to aA \mid B$

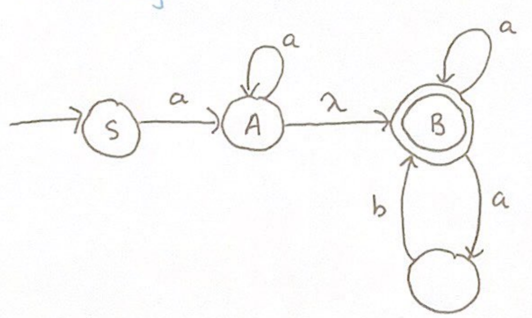
$B \to abB \mid aB \mid \lambda \}$

~~Test String: aaaababa~~

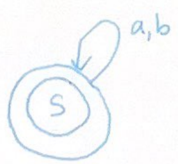
~~S ⇒ aA ⇒ aaA ⇒ aaaA ⇒ aaaB ⇒ aaaabB ⇒ aaaabaB~~

~~⇒ aaaababA ⇒ aaaababaA ⇒ aaaababa~~

NFA Diagram:



Note:



dfa or nfa diagram

$(a+b)^*$

Regular Expression

$S \to aS \mid bS \mid \lambda$

Regular Grammar