MTH-682 Automata
Assignment (1): Regular Languages

Mostafa Hassanein

22 October 2025

1.4

Each of the following languages is the intersection of two simpler languages. In each part, construct DFAs for the simpler languages, then combine them using the construction discussed in footnote 3 (page 46) to give the state diagram of a DFA for the language given. In all parts, $\Sigma = \{a, b\}$.

c. $L = \{w|w$ has an even number of a's and one or two b's$\}$

Solution:

Languages:

$L = L_1 \cap L_2$
$L_1 = \{w|\ w$ has an even number of a's$\}$
$L_2 = \{w|\ w$ has one or two b's$\}$

Automata:

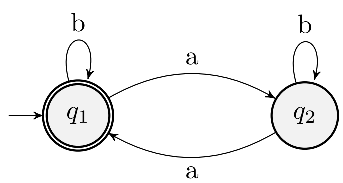

Figure 1: DFA_{L_1}

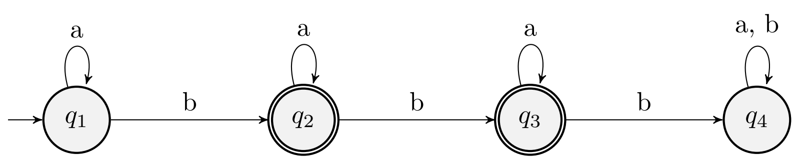

Figure 2: DFA_{L_2}

State	a	b
q_{11}	q_{21}	q_{12}
q_{21}	q_{11}	q_{22}
q_{12}	q_{22}	q_{13}
q_{22}	q_{12}	q_{23}
q_{23}	q_{13}	q_{24}
q_{13}	q_{23}	q_{14}
q_{24}	q_{14}	q_{24}
q_{14}	q_{24}	q_{14}

Table 1: Transition Table

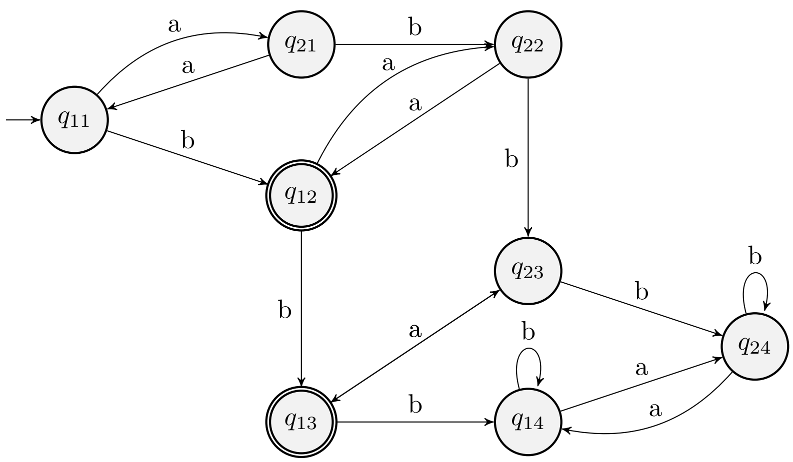

Figure 3: DFA_L

f. $L = \{w|w$ has an odd number of a's and ends with a $b\}$

Solution:

Languages:

$L = L_1 \cap L_2$
$L_1 = \{w|\ w$ has an odd number of a's$\}$
$L_2 = \{w|\ w$ ends with a $b\}$

Automata:

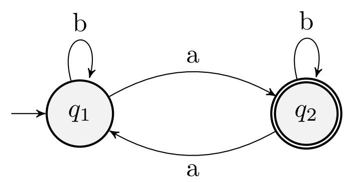

Figure 4: DFA_{L_1}

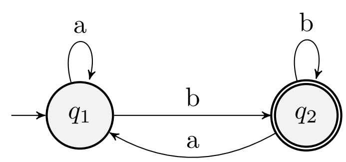

Figure 5: DFA_{L_2}

State	**a**	**b**
q_{11}	q_{21}	q_{12}
q_{12}	q_{21}	q_{12}
q_{21}	q_{11}	q_{22}
q_{22}	q_{11}	q_{22}

Table 2: Transition Table

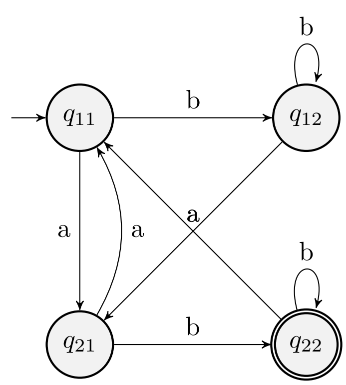

Figure 6: DFA_L

1.5

Each of the following languages is the complement of a simpler language. In each part, construct a DFA for the simpler language, then use it to give the state diagram of a DFA for the language given. In all parts, $\Sigma = \{a, b\}$.

c. $L = \{w|w$ contains neither the substrings ab nor $ba\}$

<u>Solution:</u>

$L^c = \{w|w \text{ contains the substring } ab \text{ or } ba\}$

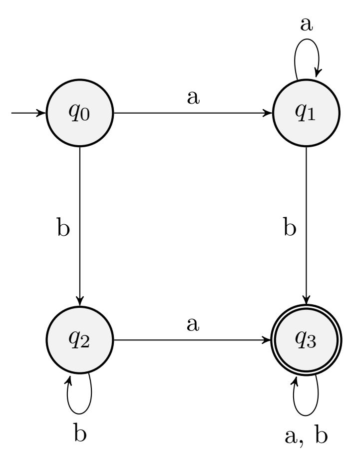

Figure 7: DFA_{L^c}

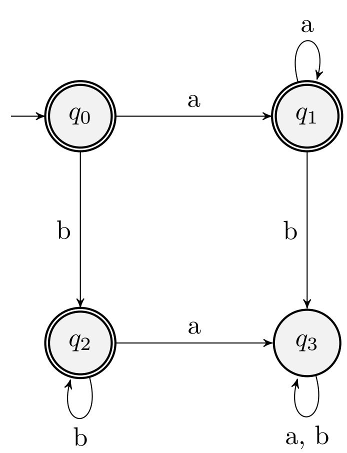

Figure 8: DFA_L

f. $L = \{w|w$ is any string not in $a^* \cup b^*\}$

<u>Solution:</u>

$L^c = a^* \cup b^* = \{w|w \text{ contains only } a\text{'s or only } b\text{'s (including } \varepsilon)\}$

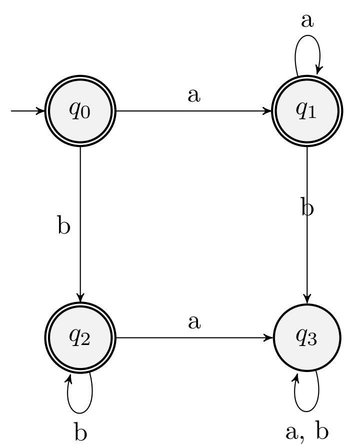

Figure 9: DFA_{L^c}

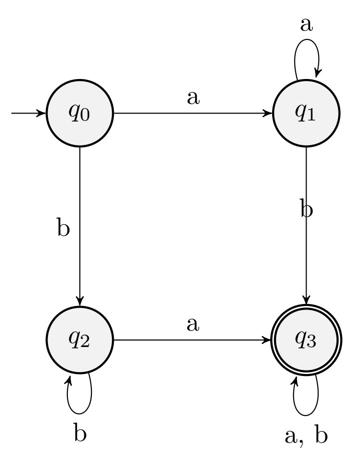

Figure 10: DFA_L

1.6

Give state diagrams of DFAs recognizing the following languages. In all parts, the alphabet is $0, 1$.

a. $L = \{w|w$ begins with a 1 and ends with a $0\}$

<u>Solution:</u>

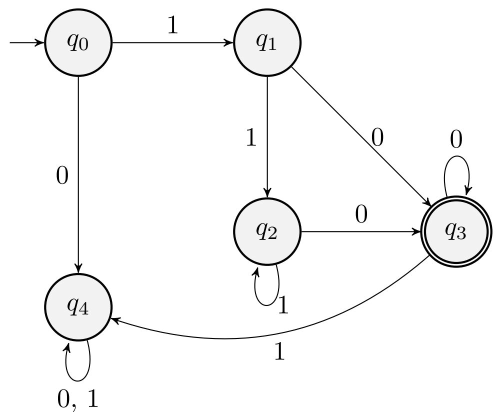

Figure 11: DFA_L

b. $L = \{w|w$ **contains at least three** 1**'s**$\}$

<u>Solution:</u>

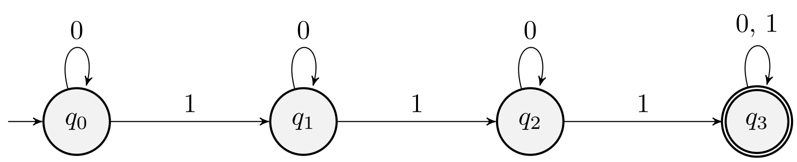

Figure 12: DFA_L

c. $L = \{w|w$ **contains the substring** 0101 **(i.e., w = x0101y for some x and y)**$\}$

<u>Solution:</u>

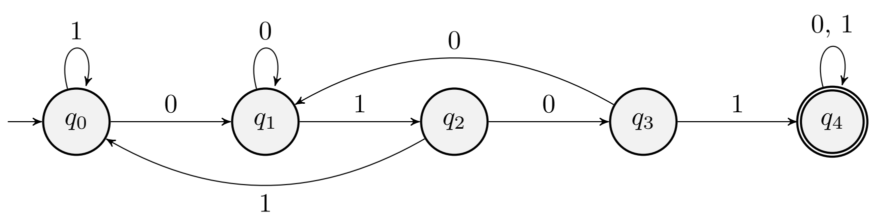

Figure 13: DFA_L

1.7

Give state diagrams of NFAs with the specified number of states recognizing each of the following languages. In all parts, the alphabet is $\{0, 1\}$.

b. $L = \{w|w$ **contains the substring** 0101 **(i.e., w = x0101y for some x and y)**$\}$**.**
$States = 5$**.**

Solution:

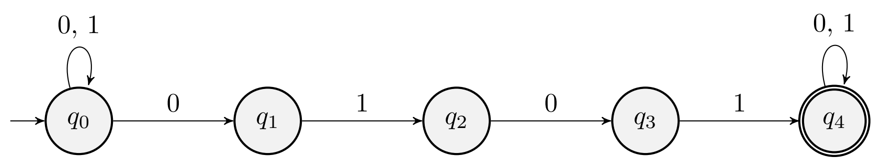

Figure 14: NFA_L

e. L **is the language of the regular expression** $0^*1^*0^+$**.**
$States = 3$**.**

Solution:

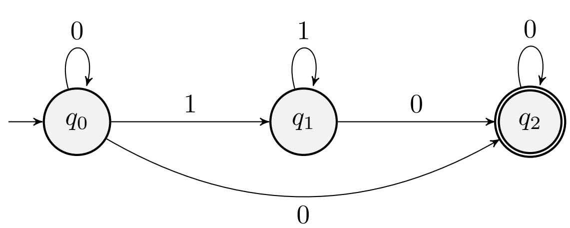

Figure 15: NFA_L

1.8

Use the construction in the proof of Theorem 1.45 to give the state diagrams of NFAs recognizing the union of the languages described in:

a. Exercises 1.6.a and 1.6.b

Solution:

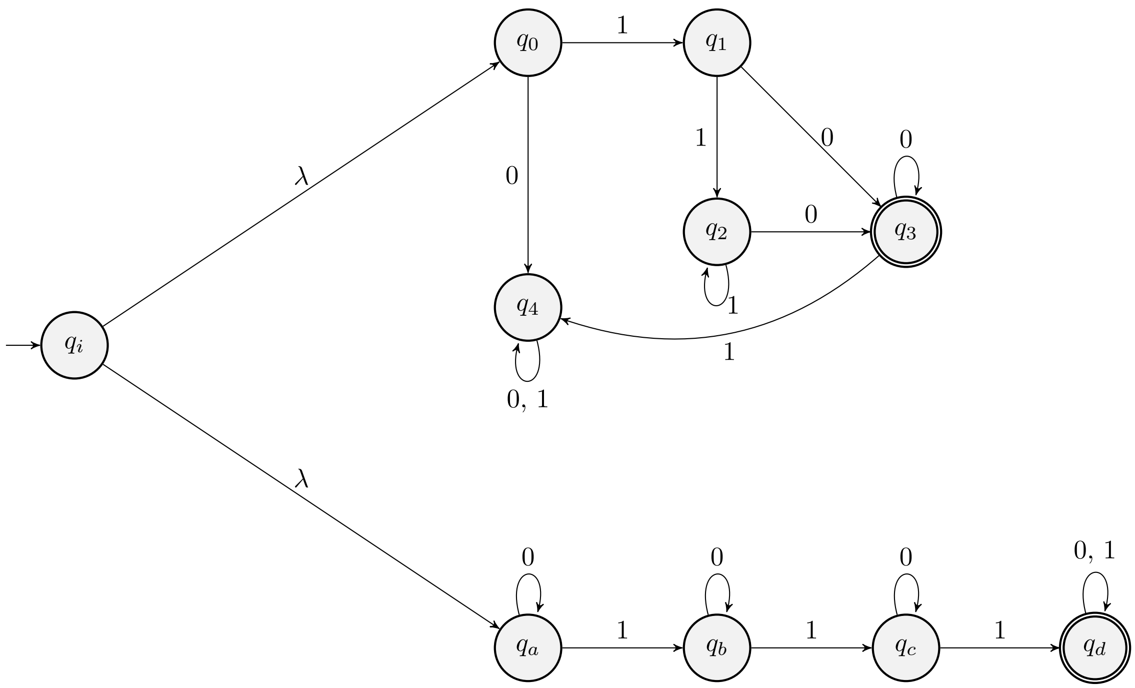

Figure 16: DFA_L

1.10

Use the construction in the proof of Theorem 1.49 to give the state diagrams of NFAs recognizing the star of the languages described in:

a. Exercise 1.6b

Solution:

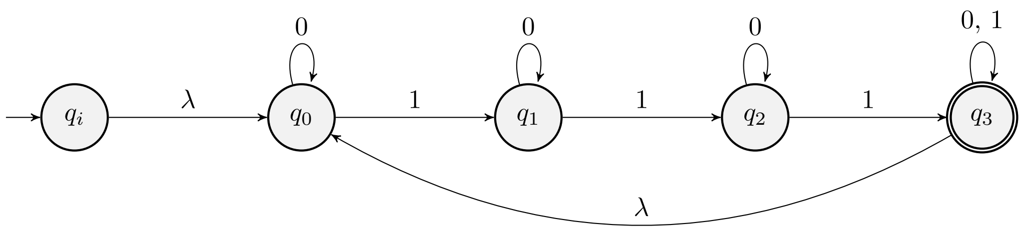

Figure 17: DFA_L

1.12

$D = \{w | w$ contains an even number of a's and an odd number of b's and does not contain the substring $ab\}$. Give a DFA with five states that recognizes D and a regular expression that generates D. (Suggestion: Describe D more simply.)

<u>Solution:</u>

Since the substring ab is not allowed, then all b symbols (at least one b symbol must exist) must come at the front before all a symbols (if any).

<u>Regular Expression:</u>

$$L = (bb)^*b(aa)^*$$

<u>Automaton (DFA):</u>

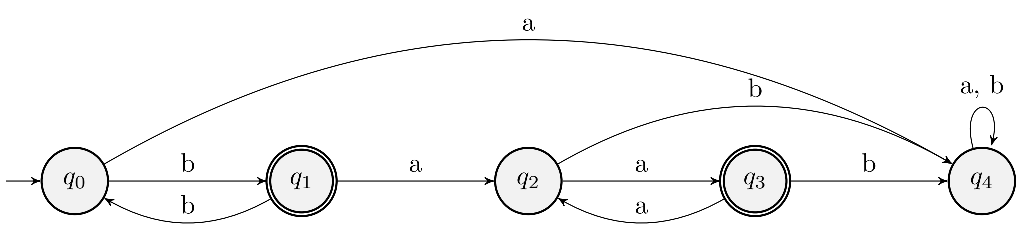

Figure 18: DFA_L

1.13

Let F be the language of all strings over $\{0, 1\}$ that do not contain a pair of 1s that are separated by an odd number of symbols. Give the state diagram of a DFA with five states that recognizes F. (You may find it helpful first to find a 4-state NFA for the complement of F.)

<u>Solution:</u>

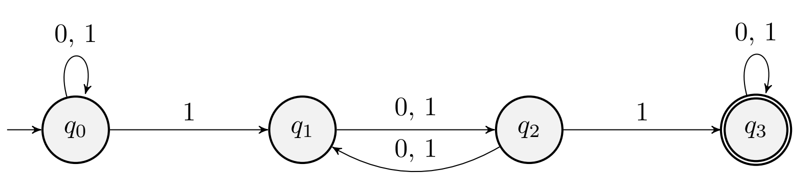

Figure 19: NFA_{F^c}

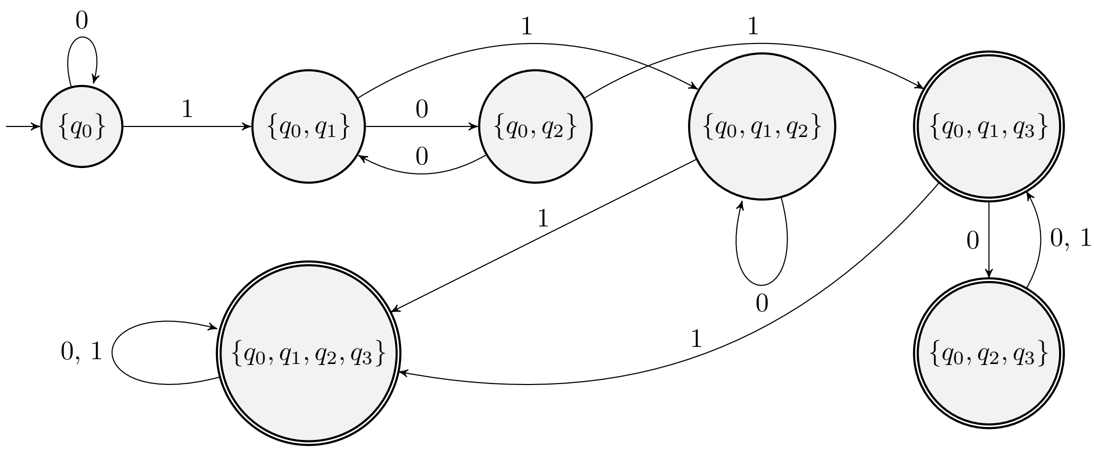

Figure 20: DFA_{F^c}

All three accepting states can be merged into one accepting state (since transitions from any accepting state lead to another accepting state).

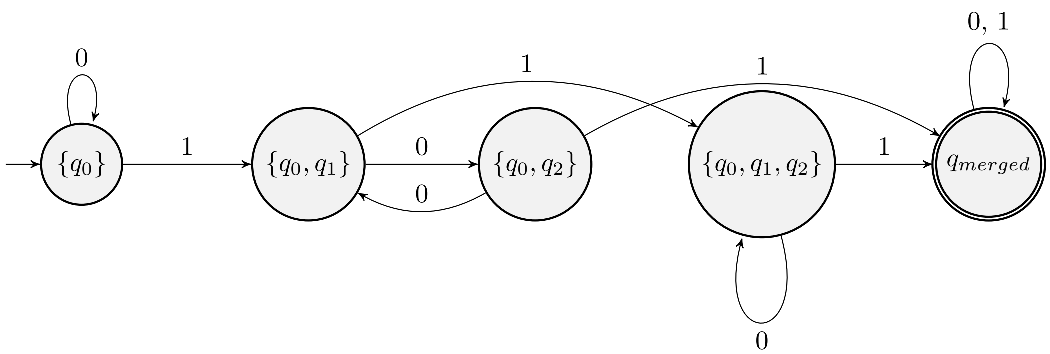

Figure 21: DFA_{F^c} (Merged Accepting States)

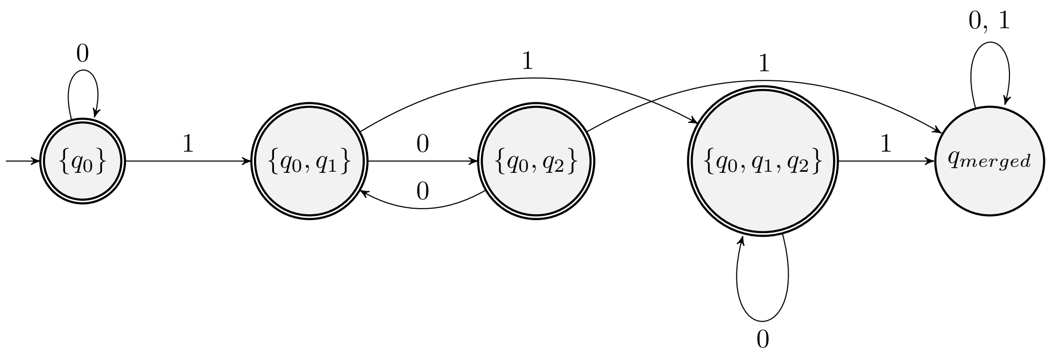

Figure 22: DFA_F

1.16

Use the construction given in Theorem 1.39 to convert the following two nondeterministic finite automata to equivalent deterministic finite automata.

b.

Solution:

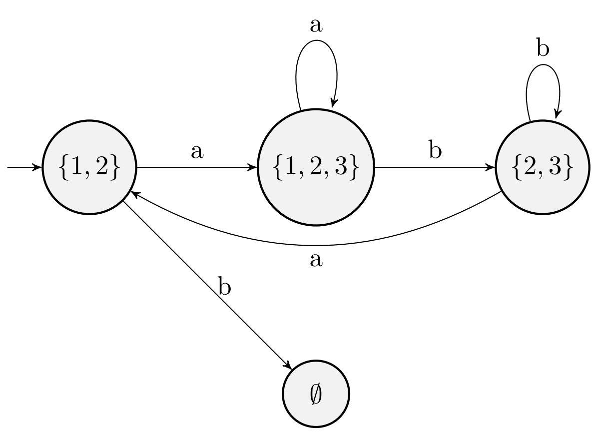

Figure 23: DFA

1.21

Use the procedure described in Lemma 1.60 to convert the following finite automata to regular expressions.

b.

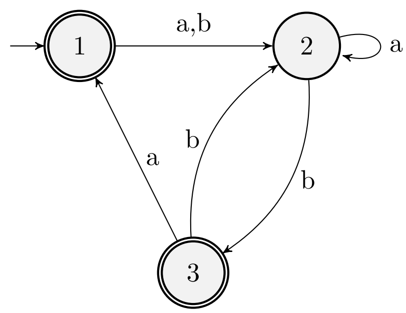

Figure 24: DFA

<u>Solution:</u>

To find the equivalent regular expression, we transform the DFA into the equivalent GDFA.

We start by introducing a new initial state and a new accepting state:

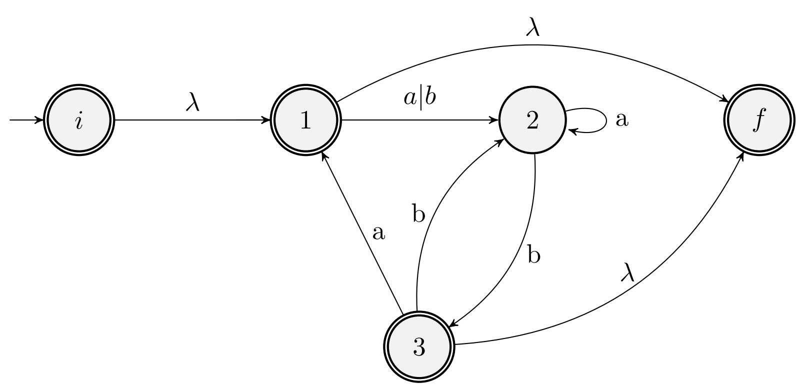

Figure 25: $GDFA_1$

Next, we eliminate state 2:

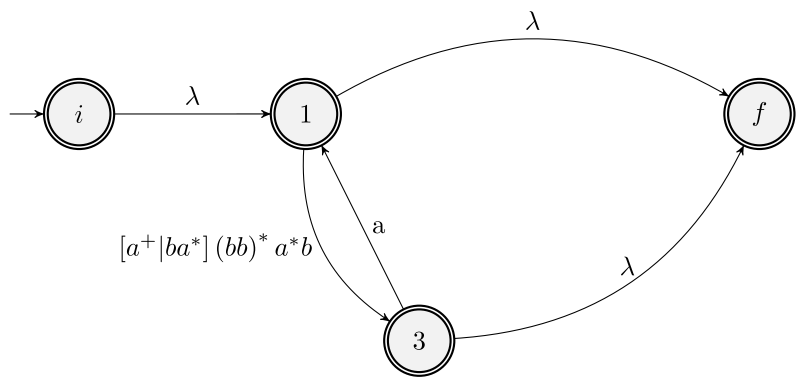

Figure 26: $GDFA_2$

Next, we eliminate state 3:

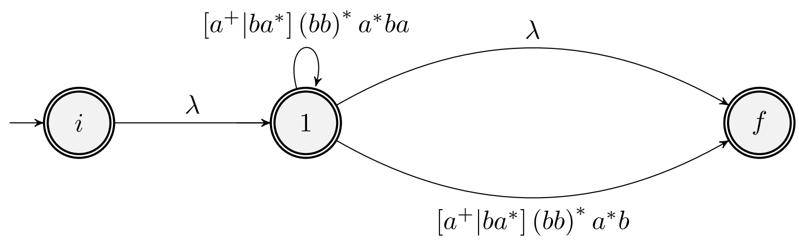

Figure 27: $GDFA_3$

Finally, we eliminate state 1:

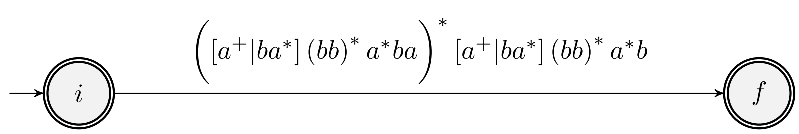

Figure 28: $GDFA_4$

Therefore:

$$R = \left(\left[a^+|ba^*\right](bb)^*\,a^*ba\right)^*\left[a^+|ba^*\right](bb)^*\,a^*b$$

1.22

In certain programming languages, comments appear between delimiters such as /# and #/. Let C be the language of all valid delimited comment strings. A member of C must begin with /# and end with #/ but have no intervening #/. For simplicity, assume that the alphabet for C is $\Sigma = \{a, b, /, \#\}$.

a.

Give a DFA that recognizes C.

<u>Solution:</u>

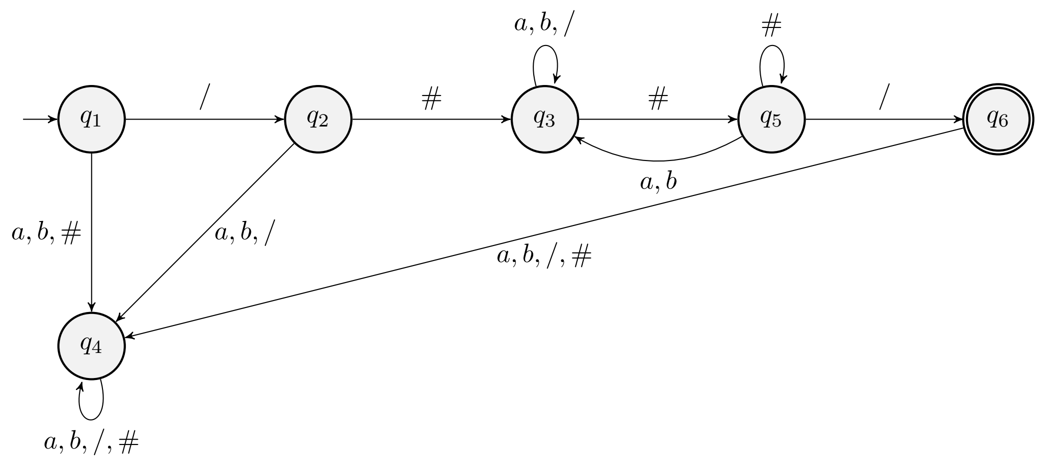

Figure 29: DFA_C

b.

Give a regular expression that generates C.

<u>Solution:</u>

To find the equivalent regular expression, we transform the DFA into the equivalent GDFA.

First, we eliminate q_2:

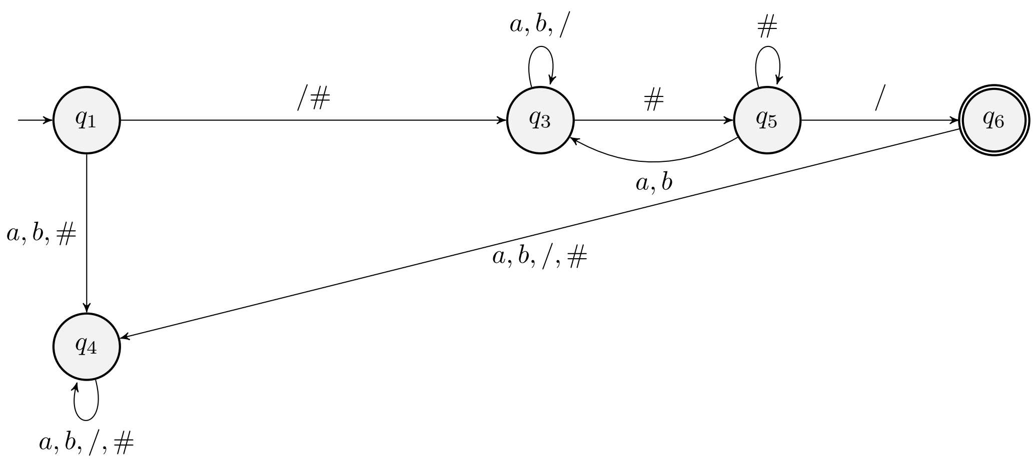

Figure 30: DFA_C

Next, we eliminate q_5:

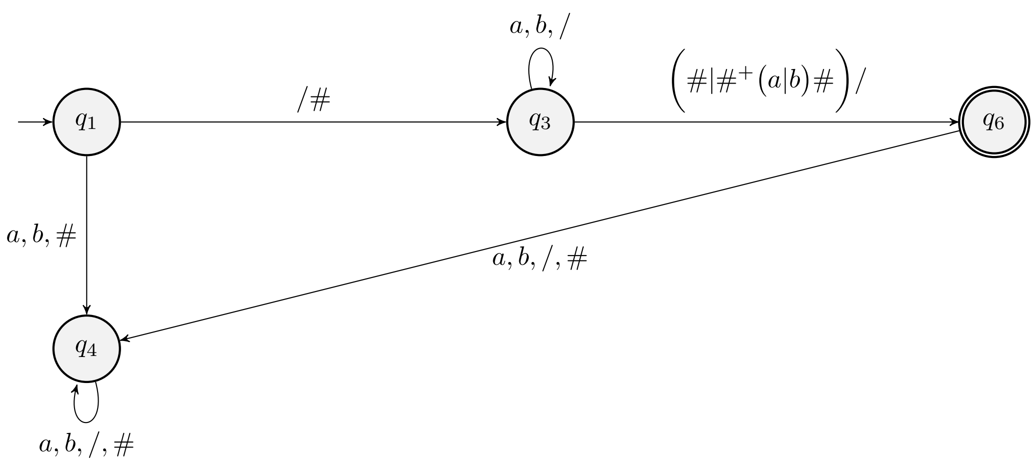

Figure 31: DFA_C

Next, we eliminate q_3:

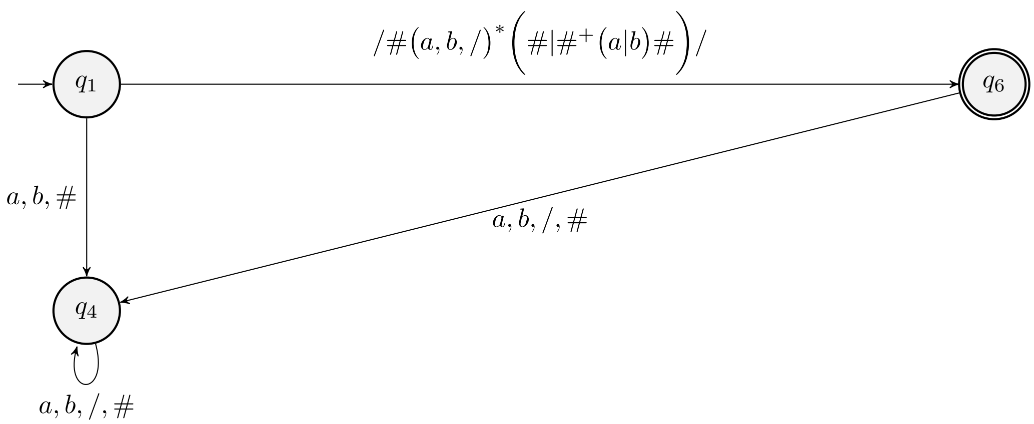

Figure 32: DFA_C

Finally, we eliminate q_4:

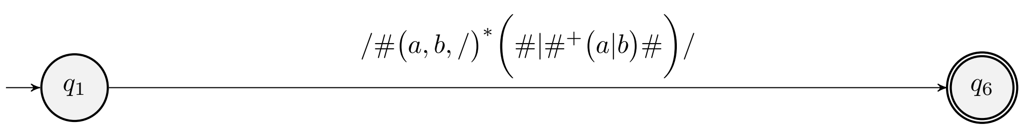

Figure 33: DFA_C

$$R = /\#(a,b,/)^*\left(\#|\#^+(a|b)\#\right)/$$

1.27

Read the informal definition of the finite state transducer given in Exercise 1.24. Give the state diagram of an FST with the following behavior. Its input and output alphabets are $\{0,1\}$. Its output string is identical to the input string on the even positions but inverted on the odd positions. For example, on input 0000111 it should output 1010010.

Solution:

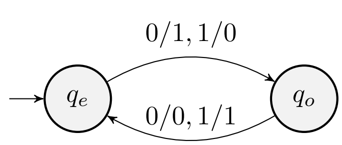

Figure 34: DFA

1.28

Convert the following regular expressions to NFAs using the procedure given in Theorem 1.54. In all parts, $\Sigma = \{a, b\}$.

a.

$$L = a\,(abb)^* \cup b$$

<u>Solution:</u>

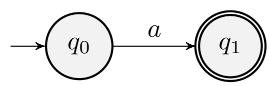

Figure 35: $NFA_{L=a}$

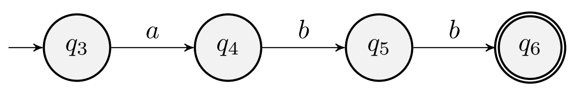

Figure 36: $NFA_{L=abb}$

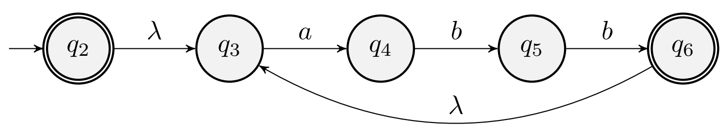

Figure 37: $NFA_{L=(abb)^*}$

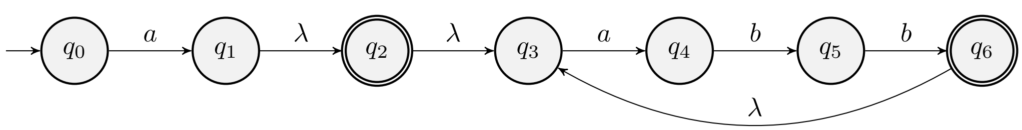

Figure 38: $NFA_{L=a(abb)^*}$

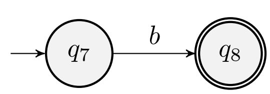

Figure 39: $NFA_{L=b}$

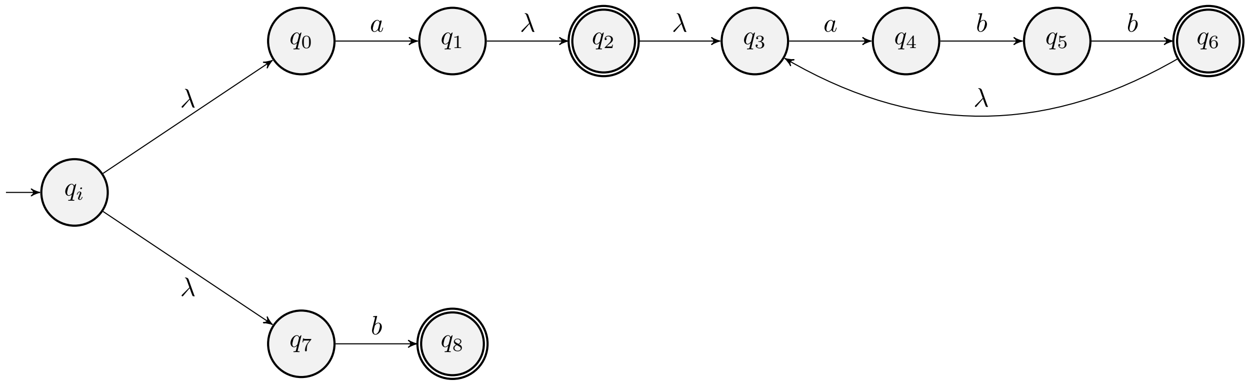

Figure 40: $NFA_{L=a(abb)^{*}\cup b}$

1.46

Prove that the following languages are not regular. You may use the pumping lemma and the closure of the class of regular languages under union, intersection, and complement.

a.

$$L = \{0^n 1^m 0^n \mid m, n \geq 0\}$$

<u>Solution:</u>

Proof.

Suppose L is regular. Then, there exists a pumping length p for L.

Let $s = 0^p 1^p 0^p$.

Since $s \in L$ and L is regular, then, by the pumping lemma:
i. $s = xyz$
ii. $|y| > 0$
iii. $|xy| \leq p$
iv. $\forall i \geq 0$, the strings $xy^i z \in L$.

Conditions ii, iii imply that y consits of all zeros.

But this implies that the string $t = xy^2z$ contains more zeros on the left than on the right, and thus $t \notin L$.

This shows that L cannot be pumped, and, thus, not regular.

□

1.53

Let $\Sigma = \{0, 1, +, =\}$ and

$ADD = \{x = y + z | x, y, z$ are binary integers, and x is the sum of y and z$\}$.

<u>Solution:</u>

Proof.

Suppose ADD is regular. Then, there exists a pumping length p for ADD.

Let s be $1^p = 1^p + 0^p$.

Since $s \in ADD$ and ADD is regular, then, by the pumping lemma:
i. $s = abc$
ii. $|b| > 0$
iii. $|ab| \leq p$
iv. $\forall i \geq 0$, the strings $ab^ic \in ADD$.

Conditions ii, iii imply that b consits of all 1's.

But this implies that the string $t = xy^2z$ contains more 1's on the left than on the right (therefore, is not the valid result of an add operation), and thus $t \notin L$.

This shows that L cannot be pumped, and, thus, not regular.

$\square$