

Theory of Computation 2024F

Proposed Questions

Compilers: Marina Alchirch & Tashfeen

August 25, 2025

Contents

1 Student Proposed Problems for the 1st Assignment	3
1.1 Q6: Candidate Exam Question	3
1.2 Q6: Candidate Exam Question	3
1.3 Q6: Candidate Exam Question	4
1.4 Q6: Candidate Exam Question	4
1.5 Q6: Candidate Exam Question	5
1.6 Q6: Candidate Exam Question	5
1.7 Q6: Candidate Exam Question	6
1.8 Q6: Candidate Exam Question	7
1.9 Q6: Candidate Exam Question	8
1.10 Q6: Candidate Exam Question	8
1.11 Q6: Candidate Exam Question	9
1.12 Q6: Candidate Exam Question	10
1.13 Q6: Candidate Exam Question	11
1.14 Q6: Candidate Exam Question	11
1.15 Q6: Candidate Exam Question	11
1.16 Q6: Candidate Exam Question	12
1.17 Q6: Candidate Exam Question	12
1.18 Q6: Candidate Exam Question	13
1.19 Q6: Candidate Exam Question	13
1.20 Q6: Candidate Exam Question	14
1.21 Q6: Candidate Exam Question	14
1.22 Q6: Candidate Exam Question	15
1.23 Q6: Candidate Exam Question	15
1.24 Q6: Candidate Exam Question	16
2 Student Proposed Problems for the 2nd Assignment	17
2.1 Q6: Candidate Exam Question	17
2.2 Q6: Candidate Exam Question	18
2.3 Q6: Candidate Exam Question	18
2.4 Q6: Candidate Exam Question	19
2.5 Q6: Candidate Exam Question	19
2.6 Q6: Candidate Exam Question	22
2.7 Q6: Candidate Exam Question	22
2.8 Q6: Candidate Exam Question	24
2.9 Q6: Candidate Exam Question	25
2.10 Q6: Candidate Exam Question	26
2.11 Q6: Candidate Exam Question	26

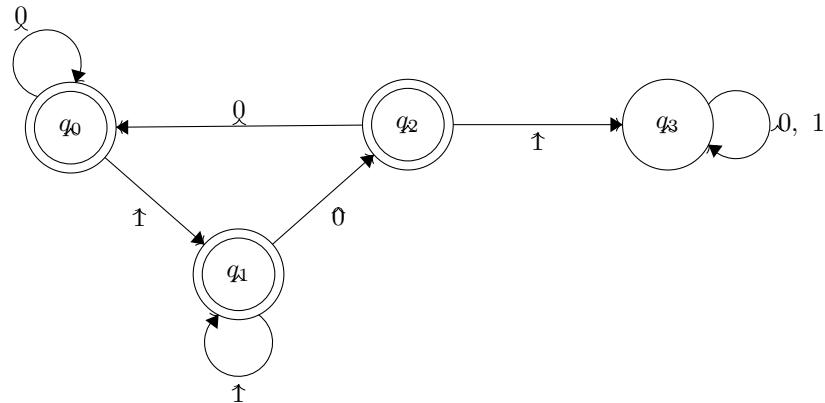
2.12	Q6: Candidate Exam Question	26
2.13	Q6: Candidate Exam Question	27
2.14	Q6: Candidate Exam Question	28
2.15	Q6: Candidate Exam Question	29
2.16	Q6: Candidate Exam Question	29
2.17	Q6: Candidate Exam Question	30
2.18	Q6: Candidate Exam Question	30
2.19	Q6: Candidate Exam Question	30
2.20	Q6: Candidate Exam Question	31
2.21	NFA Construction	31
2.22	2. DFA using Powerset Construction	32
2.23	DFA Diagram	33
2.24	Q6: Candidate Exam Question	33
2.25	Q6: Candidate Exam Question	34
2.26	Q6: Candidate Exam Question	34
2.27	Q6: Candidate Exam Question	34
2.28	Q6: Candidate Exam Question	34
2.29	Q6: Candidate Exam Question	35
2.30	Q6: Candidate Exam Question	35
2.31	Q6: Candidate Exam Question	36
3	Student Proposed Problems for the 4th Assignment	37
3.1	Q6: Candidate Exam Question	37
3.2	Q6: Candidate Exam Question	38
3.3	Q6: Candidate Exam Question	39
3.4	Q6: Candidate Exam Question	39
3.5	Q6: Candidate Exam Question	40
3.6	Q6: Candidate Exam Question	41
3.7	Q6: Candidate Exam Question	42
3.8	Q6: Candidate Exam Question	42
3.9	Q6: Candidate Exam Question	44
3.10	Q6: Candidate Exam Question	44
3.11	Q6: Candidate Exam Question	45
3.12	Q6: Candidate Exam Question	45
3.13	Q6: Candidate Exam Question	46
3.14	Q6: Candidate Exam Question	46
3.15	Q6: Candidate Exam Question	47
3.16	Q6: Candidate Exam Question	48
3.17	Q6: Candidate Exam Question	48
3.18	Q6: Candidate Exam Question	48
3.19	Q6: Candidate Exam Question	49
3.20	Q6: Candidate Exam Question	50
3.21	Q6: Candidate Exam Question	50
3.22	Q6: Candidate Exam Question	51
3.23	Q6: Candidate Exam Question	52
3.24	Q6: Candidate Exam Question	53
3.25	Q6: Candidate Exam Question	53
3.26	Q6: Candidate Exam Question	54
3.27	Q6: Candidate Exam Question	54
3.28	Q6: Candidate Exam Question	55
3.29	Q6: Candidate Exam Question	55

1 Student Proposed Problems for the 1st Assignment

1.1 Q6: Candidate Exam Question

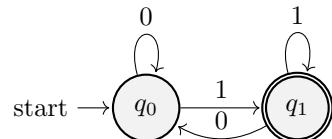
Draw a DFA recognizing a language $L_1 = \{w \mid w \text{ does not contain the substring } 1\ 0\ 1\}\$.

Answer:



1.2 Q6: Candidate Exam Question

Explain what language the DFA below recognizes



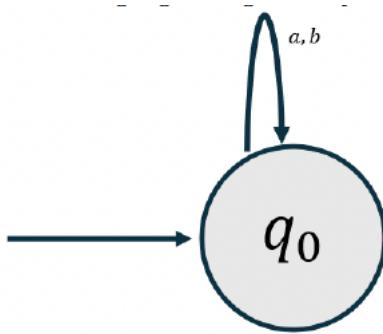
Answer:

$$L_1 = \{w \mid w \text{ ends with } 1\}$$

(1)

1.3 Q6: Candidate Exam Question

Describe the language recognized by the following DFA.



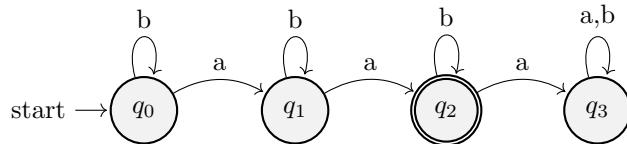
Answer: This DFA "recognizes" the empty language \emptyset . It has no accept states!

1.4 Q6: Candidate Exam Question

Draw a state diagram for a DFA recognizing the language :

$$L = \{w \mid w \text{ contains exactly two occurrences of the symbol } a\}, \Sigma = \{a, b\}$$

Answer:



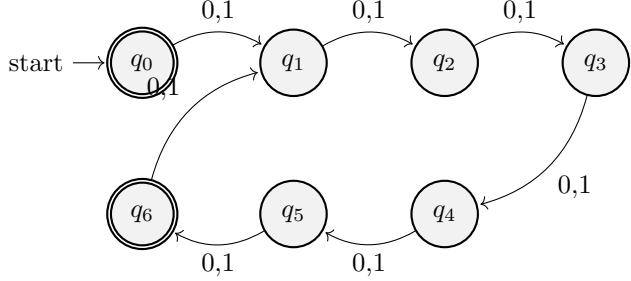
Explanation:

- q_0 is the initial state where no a has been encountered.
- q_1 represents the state where one a has been encountered.
- q_2 is the accepting state where exactly two a 's have been encountered.
- q_3 is a dead state for cases where more than two a 's are encountered.

1.5 Q6: Candidate Exam Question

Construct a DFA that recognizes the language $L_1 = \{w \mid \text{length of } w \text{ is divisible by 6}\}$, $\Sigma = \{0, 1\}$.

Answer:



1.6 Q6: Candidate Exam Question

Prove by induction that the sum of the first n odd numbers is equal to n^2 . That is, prove the statement:

$$P(n) : 1 + 3 + 5 + \dots + (2n - 1) = n^2$$

for all natural numbers $n \geq 1$.

Theorem 1. $\forall n \in \mathbb{N} : 1 + 2 + \dots + n + (n - 1) + \dots + 1 = n^2$

Answer:

Proof. **Base Case:** For $n = 1$, the sum of the first odd number is 1. According to the formula, we have:

$$1 = 1^2.$$

Since both sides are equal, the base case holds.

Inductive Step: Assume that the statement is true for some $k \geq 1$, i.e., assume:

$$1 + 3 + 5 + \dots + (2k - 1) = k^2.$$

We need to show that the statement is true for $k + 1$, i.e., we need to prove:

$$1 + 3 + 5 + \dots + (2k - 1) + (2(k + 1) - 1) = (k + 1)^2.$$

Starting from the inductive hypothesis, add the next odd number, $2(k + 1) - 1$, to both sides:

$$1 + 3 + 5 + \dots + (2k - 1) + (2(k + 1) - 1) = k^2 + (2(k + 1) - 1).$$

Simplify the right-hand side:

$$k^2 + (2(k + 1) - 1) = k^2 + 2k + 2 - 1 = k^2 + 2k + 1 = (k + 1)^2.$$

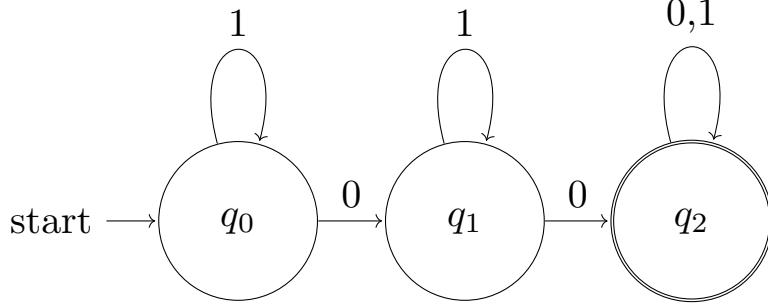
Thus, the statement holds for $k + 1$. □

Conclusion: By the principle of mathematical induction, the statement $1 + 3 + 5 + \dots + (2n - 1) = n^2$ is true for all $n \geq 1$.

1.7 Q6: Candidate Exam Question

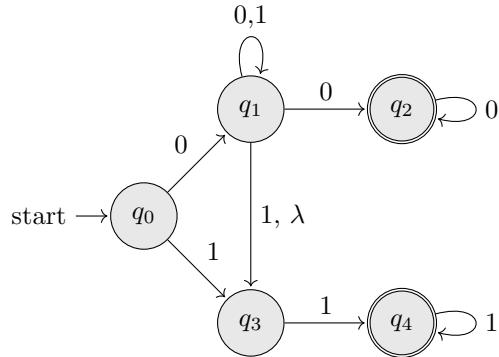
Draw a state diagram for a DFA recognizing the language $L = \{w \mid w \text{ contains at least two occurrences of } 0\}$, where $\Sigma = \{0, 1\}$.

Answer: The DFA can be represented using the following states and transitions:



- States:
 - q_0 (Initial state, not accepting)
 - q_1 (State with exactly one occurrence of '0')
 - q_2 (Accepting state with at least two occurrences of '0')
- Transitions:
 - From q_0 :
 - * On '0', go to q_1
 - * On '1', stay in q_0
 - From q_1 :
 - * On '0', go to q_2
 - * On '1', stay in q_1
 - From q_2 :
 - * On '0', stay in q_2
 - * On '1', stay in q_2
- Start State: q_0
- Accepting State: q_2

1.8 Q6: Candidate Exam Question



Give a formal description as a 5-tuple for the NFA M_5 above. Let $\Sigma = \{0, 1\}$.

Answer:

$$M_5 = (Q, \Sigma, \delta, q_0, F_5)$$

$$Q = \{q_0, q_1, q_2, q_3, q_4\}$$

$$\Sigma = \{0, 1\}$$

δ is given with:

δ	0	1	λ
q_0	$\{q_1\}$	$\{q_3\}$	\emptyset
q_1	$\{q_1, q_2\}$	$\{q_1, q_3\}$	$\{q_3\}$
q_2	$\{q_2\}$	\emptyset	\emptyset
q_3	\emptyset	$\{q_4\}$	\emptyset
q_4	\emptyset	$\{q_4\}$	\emptyset

q_0 is the start state

$$F_5 = \{q_2, q_4\}$$

1.9 Q6: Candidate Exam Question

List 3 key differences between Nondeterministic Finite Automata (NFA) and Deterministic Finite Automata (DFA).

Answer: Here are three key differences between Nondeterministic Finite Automata (NFA) and Deterministic Finite Automata (DFA):

- **State transitions:**

DFA: For each state and input symbol, there is exactly one next state.

NFA: For each state and input symbol, there can be zero, one, or multiple next states.

- **Empty string (ϵ) transitions:**

DFA: Does not allow ϵ -transitions (transitions without consuming an input symbol).

NFA: Allows ϵ -transitions, which can move to another state without reading an input.

- **Computation path:**

DFA: Has a single unique computation path for any given input string.

NFA: Can have multiple possible computation paths for a given input string.

1.10 Q6: Candidate Exam Question

Prove by induction that $n! > 2^n$ for $n \geq 4$

Answer:

Proof. Base Case: let $n = 4$. Then $4! > 2^4 \Rightarrow 24 > 16$

Prove: $(k+1)! > 2^{k+1}$

Induction: $k! > 2^k$ for $k \geq 4$

$k! > 2^k$

Multiply both sides by $(k+1)$

$(k+1)k! > (k+1)2^k$

Simplify both sides

$(k+1)k! > (k+1)2^k$

$(k+1)! > (k)2^k + 2^k$

Write a smaller expression with $k \geq 4$

$(k+1)! > (k)2^k + 2^k \geq (4)2^k + 2^k$

Write a smaller expression

$(k+1)! > (k)2^k + 2^k \geq (4)2^k + 2^k > (1)2^k + 2^k$

Simplify the equation by removing the expressions in between/middle.

$(k+1)! > 2^k + 2^k$

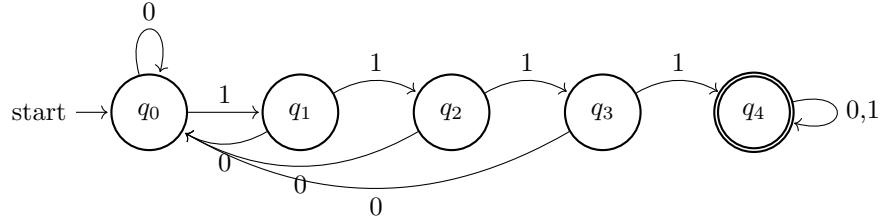
$(k+1)! > 2^k * 2$

$(k+1)! > 2^{k+1}$

□

1.11 Q6: Candidate Exam Question

Design a DFA state diagram and the respective 5-tuple for a machine M_1 that only accepts strings containing the substring $\{1111\}$. $\Sigma = \{0, 1\}$



Answer:

$$M_1 = (Q, \Sigma, \delta, q_0, F)$$

- The set of states Q is given by:

$$Q = \{q_0, q_1, q_2, q_3, q_4\}$$

- The alphabet Σ is:

$$\Sigma = \{0, 1\}$$

- The transition function δ is defined as:

$$\begin{aligned}
 \delta(q_0, 0) &= q_0 \\
 \delta(q_0, 1) &= q_1 \\
 \delta(q_1, 0) &= q_0 \\
 \delta(q_1, 1) &= q_2 \\
 \delta(q_2, 0) &= q_0 \\
 \delta(q_2, 1) &= q_3 \\
 \delta(q_3, 0) &= q_0 \\
 \delta(q_3, 1) &= q_4
 \end{aligned}$$

- The start state q_0 is:

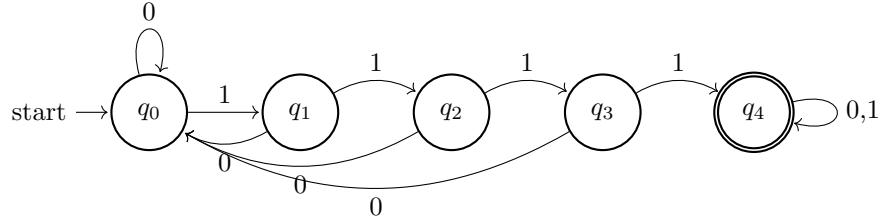
$$q_0$$

- The set of accepting (final) states F is:

$$F = \{q_4\}$$

1.12 Q6: Candidate Exam Question

Design a DFA state diagram and the respective 5-tuple for a machine M_1 that only accepts strings containing the substring $\{1111\}$. $\Sigma = \{0, 1\}$



Answer:

$$M_1 = (Q, \Sigma, \delta, q_0, F)$$

- The set of states Q is given by:

$$Q = \{q_0, q_1, q_2, q_3, q_4\}$$

- The alphabet Σ is:

$$\Sigma = \{0, 1\}$$

- The transition function δ is defined as:

$$\begin{aligned}
 \delta(q_0, 0) &= q_0 \\
 \delta(q_0, 1) &= q_1 \\
 \delta(q_1, 0) &= q_0 \\
 \delta(q_1, 1) &= q_2 \\
 \delta(q_2, 0) &= q_0 \\
 \delta(q_2, 1) &= q_3 \\
 \delta(q_3, 0) &= q_0 \\
 \delta(q_3, 1) &= q_4
 \end{aligned}$$

- The start state q_0 is:

$$q_0$$

- The set of accepting (final) states F is:

$$F = \{q_4\}$$

1.13 Q6: Candidate Exam Question

Draw a state diagram for a DFA that recognizes the language $L = \{w \mid w \text{ has an odd number of } 0\}, \Sigma\{0,1\}$

Answer:

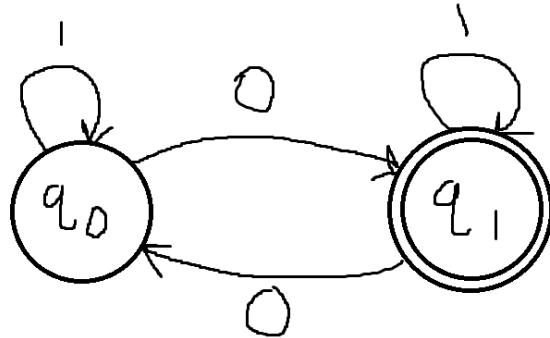


Figure 1: w with an odd number of 0s

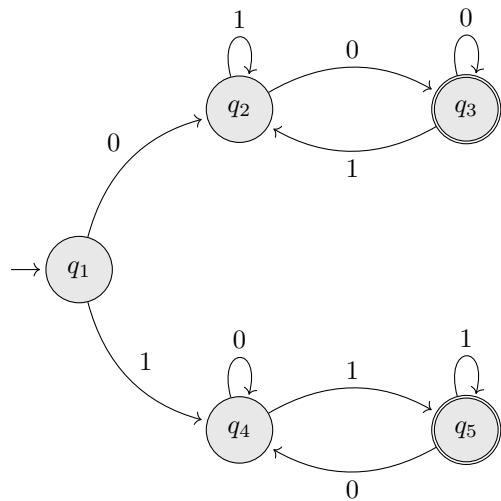
1.14 Q6: Candidate Exam Question

You are given the following parameters of a machine:

- $\Sigma = \{0,1\}$
- $L = \{ w : w \text{ begins and ends with the same symbol with total length of at least } 2\}$

Properly design the DFA that recognizes such language.

Answer:



1.15 Q6: Candidate Exam Question

Consider the following finite automaton (FA):

- States: q_0, q_1, q_2
- Alphabet: $\{a, b\}$
- Start State: q_0
- Final State: q_2
- Transitions:

$$\begin{aligned}\delta(q_0, a) &= q_1 \\ \delta(q_1, a) &= q_1 \\ \delta(q_1, b) &= q_2 \\ \delta(q_2, a) &= q_2 \\ \delta(q_2, b) &= q_2\end{aligned}$$

1. What is the language recognized by this automaton? 2. Is the language regular? Provide reasoning based on automata theory. **Answer:** 1. what is the language recognized by this automaton?

The automaton accepts strings over the alphabet $\{a, b\}$ that contain at least one 'a' followed by one 'b'. Here's the reasoning:

- In the start state q_0 , if the automaton reads 'a', it moves to q_1 .
- In q_1 , it can read multiple 'a's and stay in q_1 , but as soon as it reads a 'b', it moves to the final state q_2 .
- Once the automaton reaches q_2 , it stays there, regardless of further input.

Therefore, the language recognized is:

$$L = \{\text{strings that contain at least one 'a' followed by one 'b'}\}$$

or

$$L = \{a^+b\{a, b\}^*\}$$

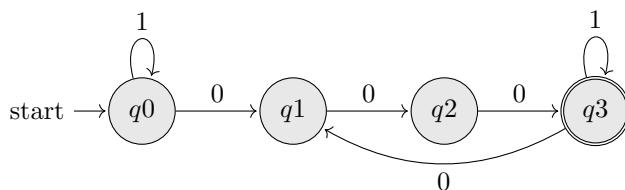
2. Is the Language Regular?

Yes, the language is regular. The automaton described is a finite automaton (FA), and it is a fundamental result from automata theory that any language that can be recognized by a finite automaton is regular.

1.16 Q6: Candidate Exam Question

Create the state diagram for DFAs that recognizes the following language:

$$L = w \mid w \text{ must contain 3 consecutive 0's}$$



1.17 Q6: Candidate Exam Question

Consider the language L for this question:

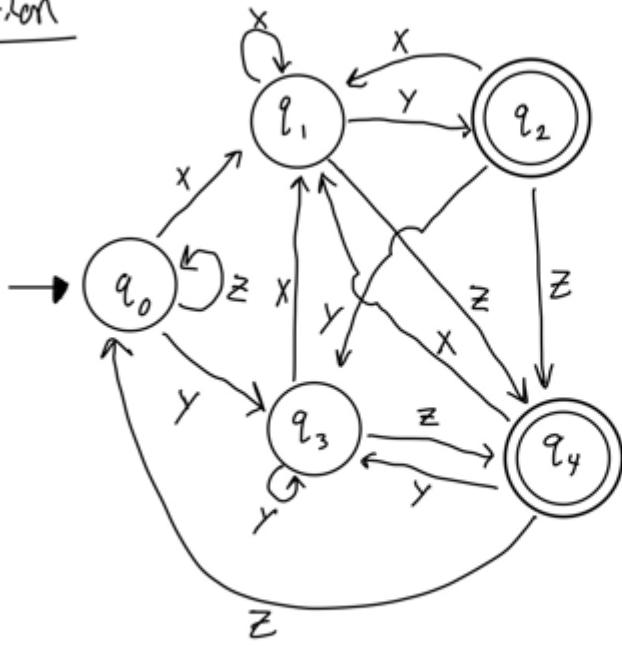
$$L = \{w \mid w \text{ ends with } xy, yz, \text{ or } xz\}, \Sigma = \{x, y, z\}.$$

1. Draw a DFA state diagram for L .

2. Provide the formal description of the DFA as a 5-tuple.

DFA for all strings over $\{x, y, z\}$, end with xy, yz , or xz

Solution



$$Q = \{q_0, q_1, q_2, q_3, q_4\}$$

$$\Sigma = \{x, y, z\}$$

δ	X	Y	Z
q_0	$\{q_1\}$	$\{q_3\}$	$\{q_0\}$
q_1	$\{q_1\}$	$\{q_2\}$	$\{q_4\}$
q_2	$\{q_1\}$	$\{q_3\}$	$\{q_4\}$
q_3	$\{q_1\}$	$\{q_3\}$	$\{q_4\}$
q_4	$\{q_1\}$	$\{q_3\}$	$\{q_0\}$

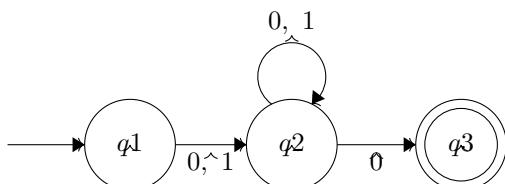
q_0 is the start state

$$F = \{q_2, q_4\}$$

Solution

1.18 Q6: Candidate Exam Question

Is this DFA or NFA? Why?



NFA, q2 has multiple possible transitions.

1.19 Q6: Candidate Exam Question

Let $X = \{1, 2, 3, 4\}$ and $Y = \{3, 4, 5, 6\}$.

- (i) Is $X \cap Y$ equal to $X \cup Y$? Justify your answer.
- (ii) What is $(X \cup Y) \setminus (X \cap Y)$?
- (iii) What is the Cartesian product $X \times Y$?
- (iv) Find the powerset of $X \cap Y$.

Answer

- (i) $X \cap Y = \{3, 4\}$ and $X \cup Y = \{1, 2, 3, 4, 5, 6\}$. Since $X \cap Y \neq X \cup Y$, they are not equal. The intersection $X \cap Y$ contains only the elements common to both sets, while the union $X \cup Y$ includes all elements from both sets.

(ii) $(X \cup Y) \setminus (X \cap Y) = \{1, 2, 5, 6\}$. This operation removes the elements that are in both X and Y from their union, leaving only the elements that are unique to each set.

(iii) The Cartesian product $X \times Y$ is:

$$X \times Y = \{(1, 3), (1, 4), (1, 5), (1, 6), (2, 3), (2, 4), (2, 5), (2, 6), (3, 3), (3, 4), (3, 5), (3, 6), (4, 3), (4, 4), (4, 5), (4, 6)\}$$

(iv) The powerset of $X \cap Y = \{3, 4\}$ is:

$$\mathcal{P}(X \cap Y) = \{\emptyset, \{3\}, \{4\}, \{3, 4\}\}$$

1.20 Q6: Candidate Exam Question

Prove by induction that $1 + r + r^2 + \dots + r^n = \frac{r^{n+1} - 1}{(r-1)}$ (formula for geometric series) for every positive integer n

Solution:

First, verify that $1 + r + r^2 + \dots + r^n = \frac{r^{n+1} - 1}{(r-1)}$ is true for $n = 1$

$$\frac{r^{1+1} - 1}{r - 1} = \frac{r^2 - 1}{r - 1} = \frac{(r + 1)(r - 1)}{r - 1} = r + 1 = 1 + r$$

Next the inductive step:

Assume that

$$1 + r + r^2 + \dots + r^k = \frac{r^{k+1} - 1}{(r - 1)}$$

where k is a positive integer. We show that,

$$1 + r + r^2 + \dots + r^k + r^{k+1} = \frac{r^{(k+1)+1} - 1}{(r - 1)}$$

Based on our previous assumption, we observe that

$$\begin{aligned} 1 + r + r^2 + \dots + r^k + r^{k+1} &= \frac{r^{k+1} - 1}{(r - 1)} + r^{k+1} \\ &= \frac{r^{k+1} - 1 + r^{k+1}(r - 1)}{r - 1} \\ &= \frac{r^{k+1} - 1 + r^{k+2} - r^{k+1}}{r - 1} \\ &= \frac{r^{k+2} - 1}{r - 1} \\ &= \frac{r^{(k+1)+1} - 1}{r - 1} \end{aligned}$$

By the Principle of Mathematical Induction, it follows that $1 + r + r^2 + \dots + r^n = \frac{r^{n+1} - 1}{(r-1)}$ for every positive integer n QED

1.21 Q6: Candidate Exam Question

Let A be the set $\{x, y, z\}$

Is the empty set \emptyset an element of A ? What about the power set $P(A)$ of A ?

Answer

The empty set is not an element of A , but it is an element of the power set of A . $\rightarrow P$

1.22 Q6: Candidate Exam Question

Problem:

Let set $A = \{2, 4, 6, 8\}$ and $A \setminus B = \{4, 6\}$.

- (a) If set B has two elements, what is set B ?
- (b) What is $A \cup B$?
- (c) What is $A \cap B$?
- (d) What is $A \times B$?
- (e) What is the powerset of A ?

Solution:

- (a) Since $A \setminus B = \{4, 6\}$, the elements of B must be the elements of A that are not in $A \setminus B$.

$$B = \{2, 8\}$$

- (b) $A \cup B$ is the set of elements in either A or B , that do not repeat.

$$A \cup B = \{2, 4, 6, 8\}$$

- (c) $A \cap B$ is the set of elements that are in both A and B .

Since $B = \{2, 8\}$ and $A = \{2, 4, 6, 8\}$, we have

$$A \cap B = \{2, 8\}.$$

- (d) $A \times B$ is the set of ordered pairs where the first element is from A and the second is from B .

So,

$$A \times B = \{(2, 2), (2, 8), (4, 2), (4, 8), (6, 2), (6, 8), (8, 2), (8, 8)\}.$$

- (e) The power set of A , $\mathcal{P}(A)$, is the set of all subsets of A .

So,

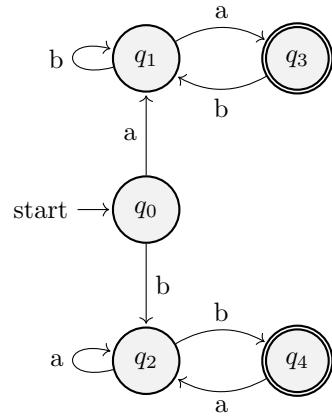
$$\begin{aligned} \mathcal{P}(A) = & \{\emptyset, \{2\}, \{4\}, \{6\}, \{8\}, \{2, 4\}, \{2, 6\}, \{2, 8\}, \{4, 6\}, \{4, 8\}, \{6, 8\}, \{2, 4, 6\}, \\ & \{2, 4, 8\}, \{2, 6, 8\}, \{4, 6, 8\}, \{2, 4, 6, 8\}\}. \end{aligned}$$

1.23 Q6: Candidate Exam Question

Question: Draw the state diagram for a DFA which recognizes the following language:

$$L = \{w \mid w \text{ begins and ends with the same symbol}\}, \Sigma = \{a, b\}.$$

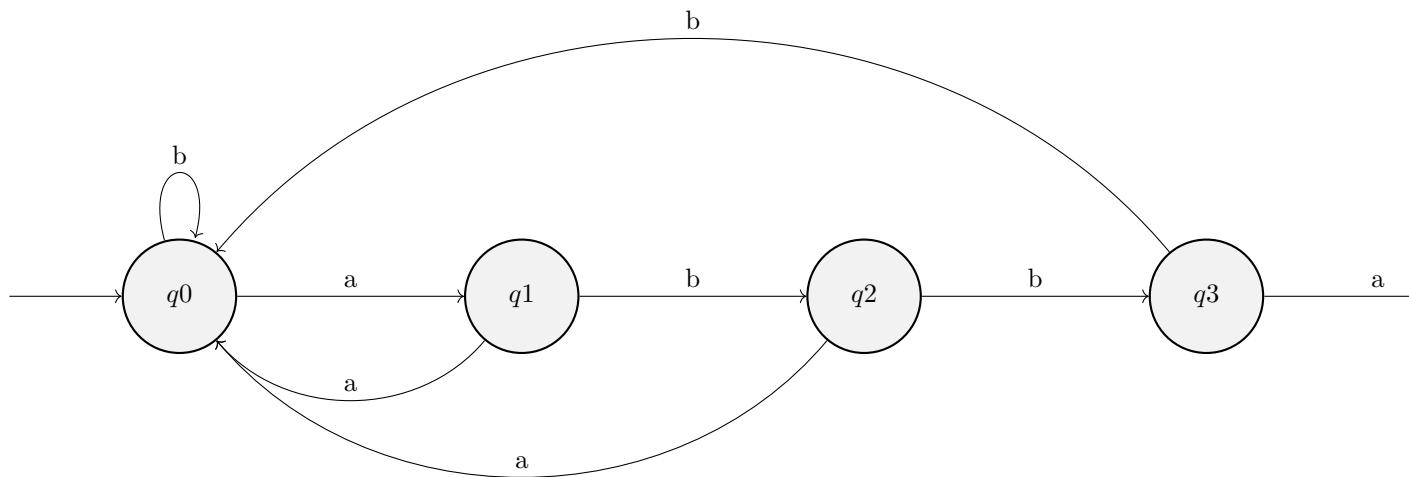
Possible Solution:



1.24 Q6: Candidate Exam Question

Problem: For the language construct a DFA $L_1 = \{w \mid \text{The sub string abba must occur at least once}\}$.

Solution: See DFA below. q_4 is accepting zoom in to see state better.



2 Student Proposed Problems for the 2nd Assignment

2.1 Q6: Candidate Exam Question

Use the Pumping Lemma method to prove that the following language is not regular:

$$L = \{a^s b^q a^r \mid r = s + q\}$$

Answer:

Proof. Let's assume that L is regular; based on the Pumping Lemma, there's some constant p where the size of an accepted input S is at least p (in other words, $S \in L$, $|S| \geq p$), and that S can be represented as xyz such that:

- $|xy| \leq p$,
 - $|y| > 0$, and
 - $xy^i z \in L$.
- As an example, let's consider the accepted string $S' = a^p b a^{p+1}$, where $|S'| = 2p + 2 \geq p$. Since $|xy|$ cannot be greater than p , then:
- $x = a^\alpha$, $\alpha \geq 0$,
 - $y = a^\beta$, $\beta \geq 1$, and
 - $z = a^{p-\alpha-\beta} b a^{p+1}$, where $\alpha + \beta \leq p$.

$$\begin{aligned} xy^i z &= a^\alpha a^{i\beta} a^{p-\alpha-\beta} b a^{p+1} \\ &= a^{p+i\beta-\beta} b a^{p+1} \end{aligned}$$

- However, we notice that, when setting $i = 2$:

$$\begin{aligned} xy^2 z &= a^{p+2\beta-\beta} b a^{p+1} \\ &= a^{p+\beta} b a^{p+1} \end{aligned}$$

which is not included in L because:

$$p + 1 \neq p + \beta + 1$$

since β cannot be less than 1.

- Thus, with proof by contradiction, L is not considered regular.

□

2.2 Q6: Candidate Exam Question

Let $\Sigma = \{0, 1\}$. Give regular expressions for the following languages:

1. $L_1 = \{w \mid w \text{ contains an even number of } 1\text{s}\}$.
2. $L_2 = \{w \mid w \text{ starts and ends with the substring '01'}\}$.

Answer:

1. The regular expression containing an even number of 1s can be shown as:

$$L_1 = (0^*(10^*10^*)^*)$$

Explanation: Where you pair 1 with another 1, allowing any number of 0s between.

2. The regular expression that starts and ends with the substring '01' can be shown as:

$$L_2 = 01(0 \mid 1)^*01$$

Explanation: Where we can observe the string beginning with 01, followed by a combination of 0s and/or 1s and ending with 01.

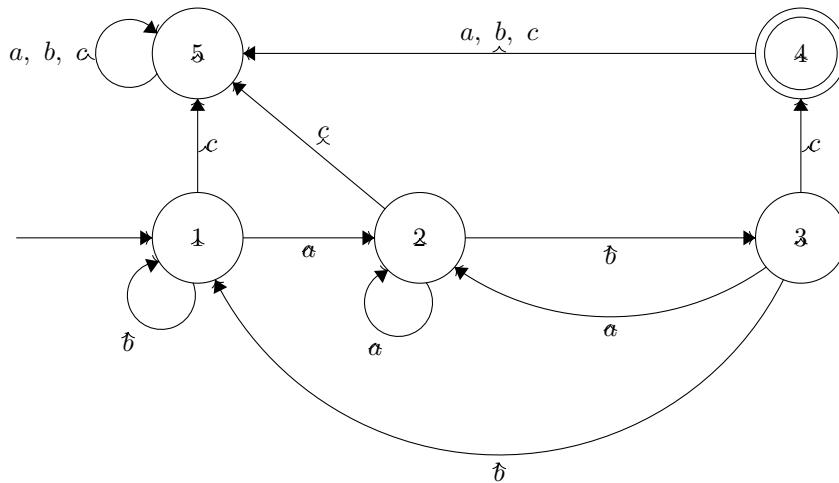
2.3 Q6: Candidate Exam Question

Let $\Sigma = \{a, b, c\}$. Consider the following language:

$$L = \{w \mid w \text{ contains no more than one } c \text{ and ends with } abc\}$$

- a) Rewrite L as a regular expression.
- b) Draw a state diagram representing L.

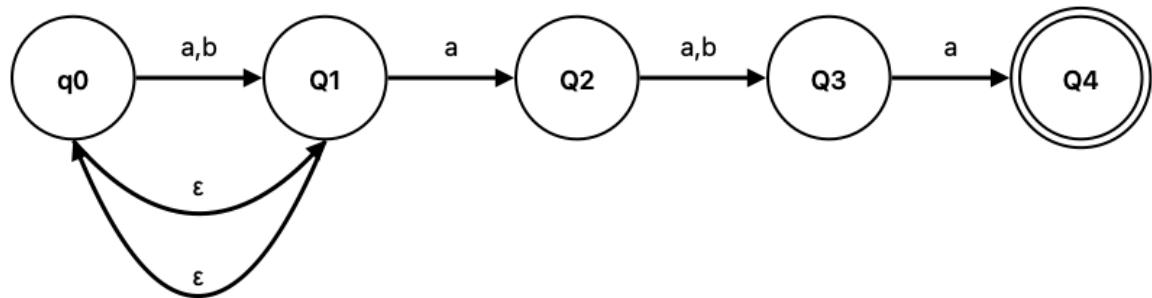
Answer: a) $(a + b)^*abc$
b)



2.4 Q6: Candidate Exam Question

Design an NFA for the regular expression:

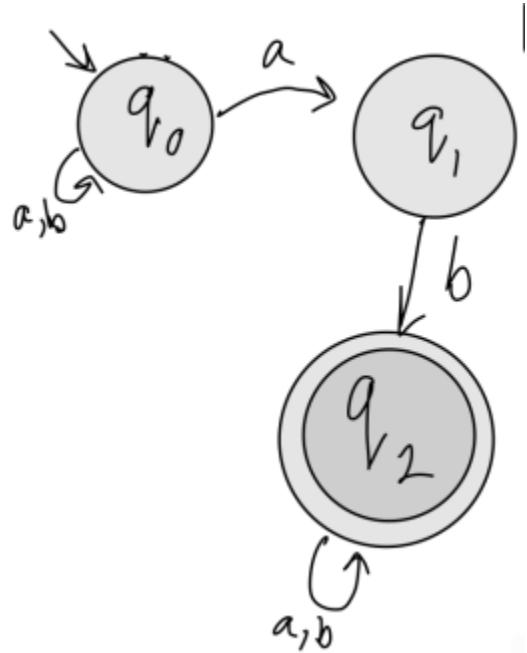
$$R = (a|b)^*a(a|b)a$$



Answer:

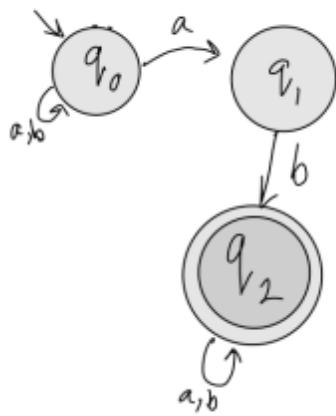
2.5 Q6: Candidate Exam Question

Consider the Non-deterministic Finite Automaton (NFA) for this question:



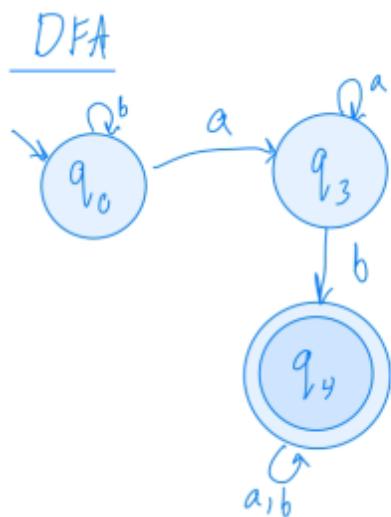
1. Convert the NFA to DFA.
2. Using your DFA convert it to a regular expression.

Convert NFA to DFA to regular expression



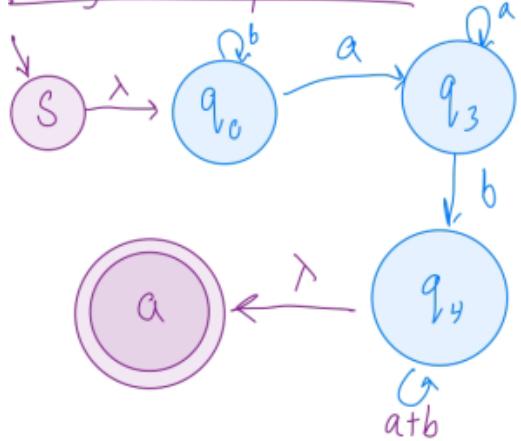
DFA

	a	b
q_0	q_3	q_0
q_1	\emptyset	q_2
q_2	q_2	q_2
$q_3 \leftarrow \{q_0, q_1\}$	q_3	q_4
$q_4 \leftarrow \{q_0, q_2\}$	q_4	q_4

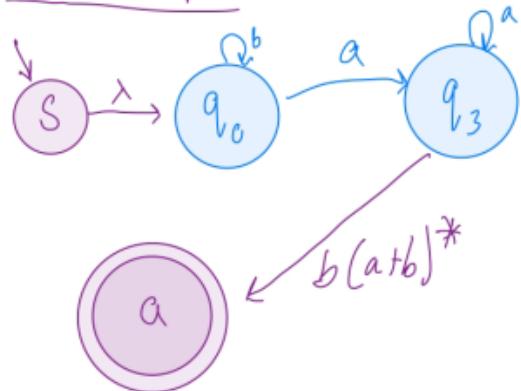


q_1 is never reached, which also means
 q_2 is never reached, so we don't create them

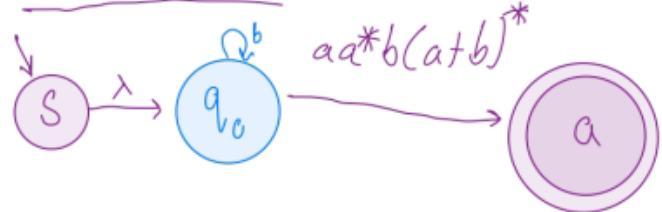
Regular Expression



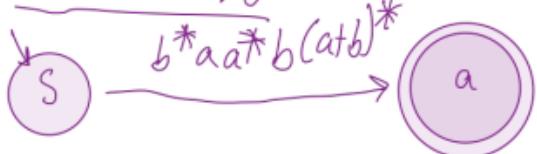
remove q_4



remove q_3



remove q_0



Answer:

2.6 Q6: Candidate Exam Question

Let $\Sigma = \{0, 1\}$. Give regular expressions for the following languages.

(i) $L_1 = \{w \mid w \text{ contains at most three } 1\text{s}\}$.

(ii) $L_2 = \{w \mid w \text{ contains exactly two } 0\text{s}\}$.

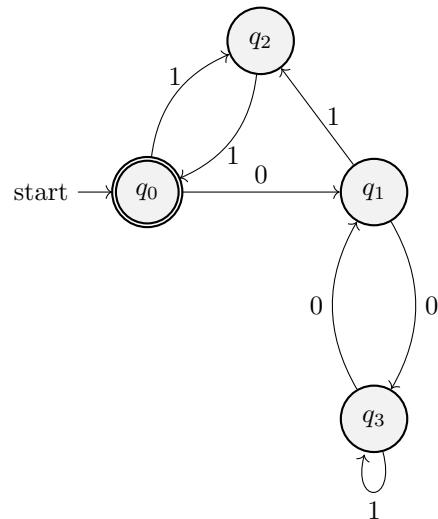
Answer:

(i) $0^* + (0^*)1(0^*) + (0^*)1(0^*)1(0^*) + (0^*)1(0^*)1(0^*)1(0^*)$

(ii) $(1^*)0(1^*)0(1^*)$

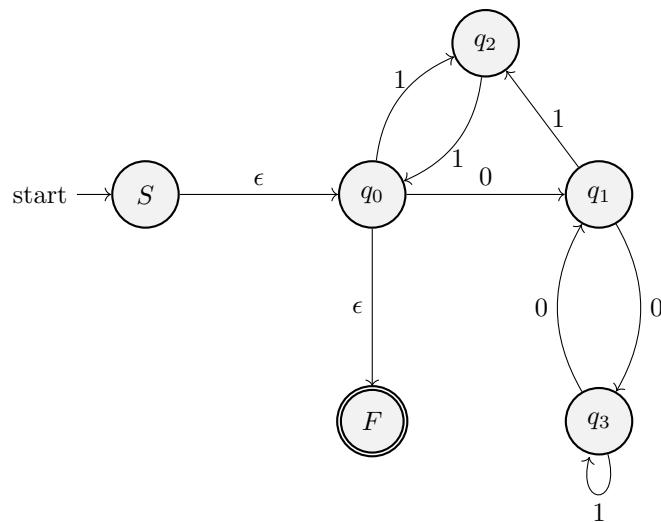
2.7 Q6: Candidate Exam Question

Convert the given NFA to a regular expression using the node deletion method.

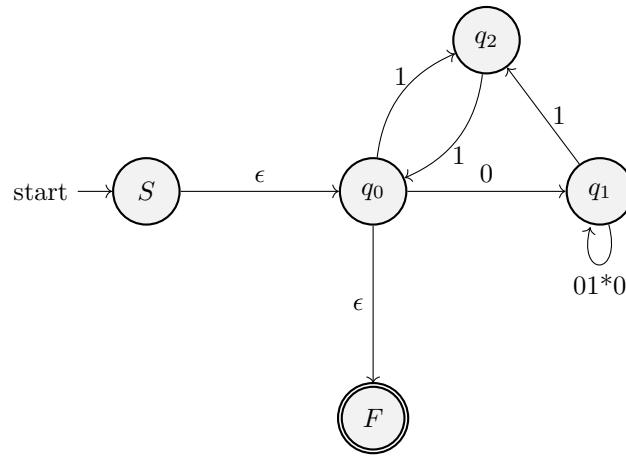


Answer:

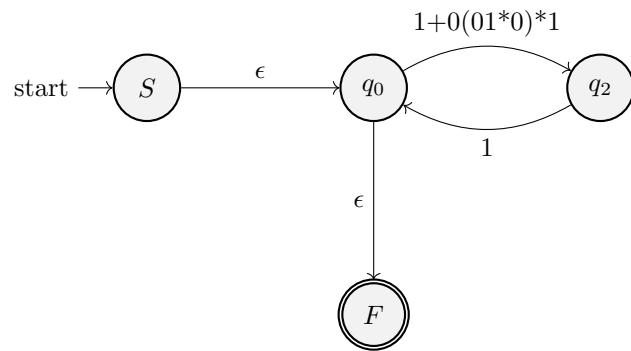
1. Normalize



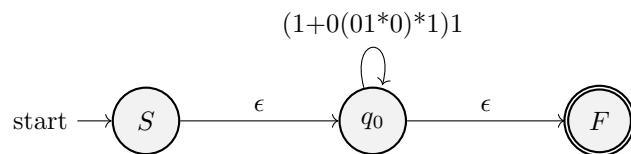
2. Rip q_3 and redraw q_1 .



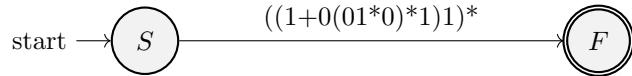
3. Rip q_1 and redraw q_0 to q_2 .



4. Rip q_2 and redraw q_0 .



5. Rip q_0 and redraw S to F.



REGEX: $((1+0(01*0)*1)1)^*$

2.8 Q6: Candidate Exam Question

Let $\Sigma = \{a, b\}$. Give regular expressions for the following languages:

1. $L_1 = \{w \mid w \text{ contains at least two a's and at least one b}\}$.
2. $L_2 = \{w \mid w \text{ ends with the substring aba}\}$.

Answer:

1. **Language** $L_1 = \{w \mid w \text{ contains at least two a's and at least one b}\} :$

$$L_1 : (a + b)^* a (a + b)^* a (a + b)^* b (a + b)^*$$

Explanation:

- The first $(a + b)^*$ - combination of 'a's and 'b's before the first 'a'.
- The first a - ensures the first required 'a'. Same with the second a .
- The second $(a + b)^*$ - combination of 'a's and 'b's between the first and second 'a'.
- The next $(a + b)^* b (a + b)^*$ - there is at least one 'b' somewhere in the string, with combination of 'a's and 'b's before or after the 'b'.

2. **Language** $L_2 = \{w \mid w \text{ ends with the substring aba}\} :$

$$L_2 : (a + b)^* aba$$

Explanation:

- The first $(a + b)^*$ - combination of 'a's and 'b's before the substring "aba".
- The substring "aba" ensures the string ends with the required substring.

2.9 Q6: Candidate Exam Question

Let $\Sigma = \{0, 1\}$. Provide regular expressions for the following languages:

- (i) $L_1 = \{w \mid w \text{ contains the substring } 001\}$.
- (ii) $L_2 = \{w \mid w \text{ does not contain the substring } 11\}$.
- (iii) $L_3 = \{w \mid w \text{ contains at least three } 1\text{s and at least two } 0\text{s}\}$.

Answer:

- (i) The regular expression for L_1 is:

$$L_1 = (0 + 1)^*001(0 + 1)^*$$

This regular expression ensures that the string contains the substring "001" somewhere, with any sequence of 0s and 1s before and after it.

- (ii) The regular expression for L_2 is:

$$L_2 = (0 + 10)^*$$

This ensures that no two consecutive 1s appear in the string. The pattern allows any combination of 0s and single 1s.

- (iii) The regular expression for L_3 is:

$$L_3 = (0 + 1)^*1(0 + 1)^*1(0 + 1)^*1(0 + 1)^*0(0 + 1)^*0(0 + 1)^*$$

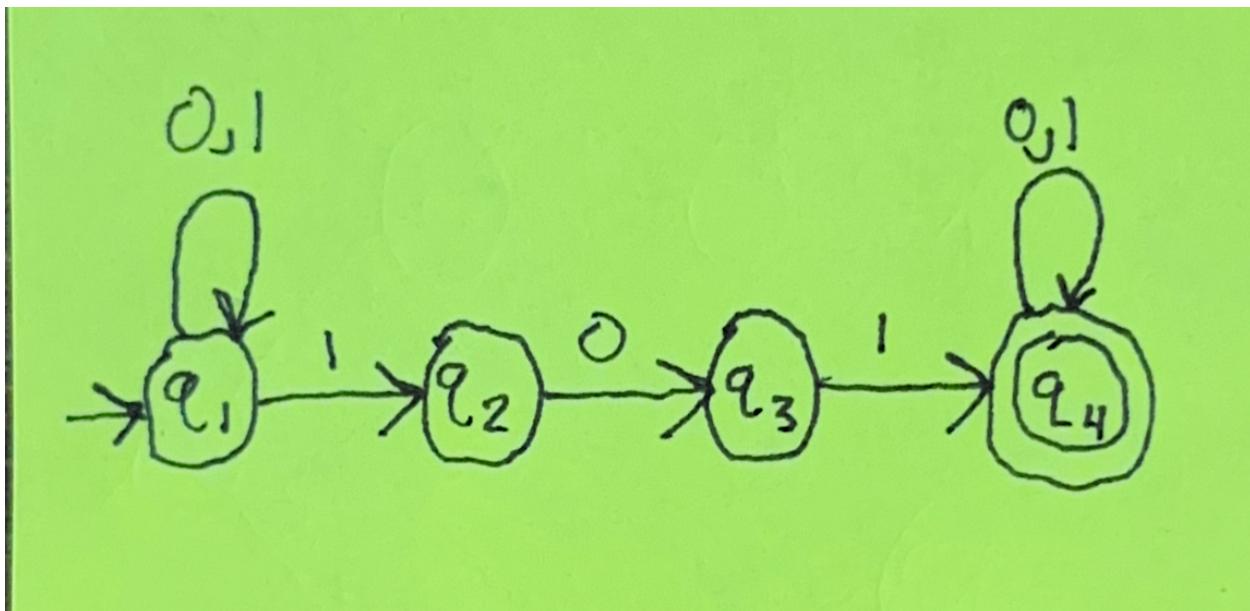
This regular expression ensures that the string contains at least three occurrences of 1 and at least two occurrences of 0, with any combination of 0s and 1s before, between, and after them.

2.10 Q6: Candidate Exam Question

Draw the NFA of the language:

$$L = \{w \mid w \text{ contains the substring } 101\}$$

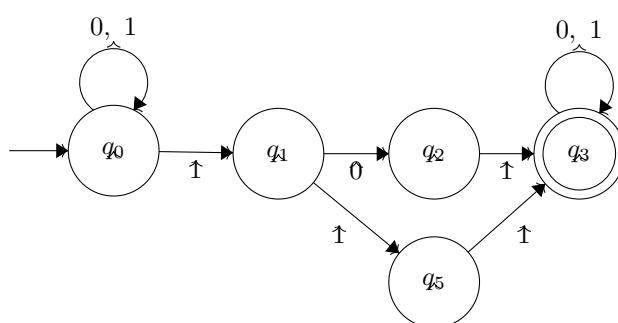
Answer:



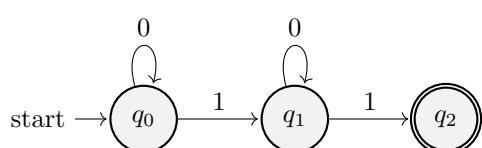
2.11 Q6: Candidate Exam Question

Give an NFA for the language $L = \{w \mid w \text{ contains } 101 \text{ or } w \text{ contains } 111\}$.

Answer

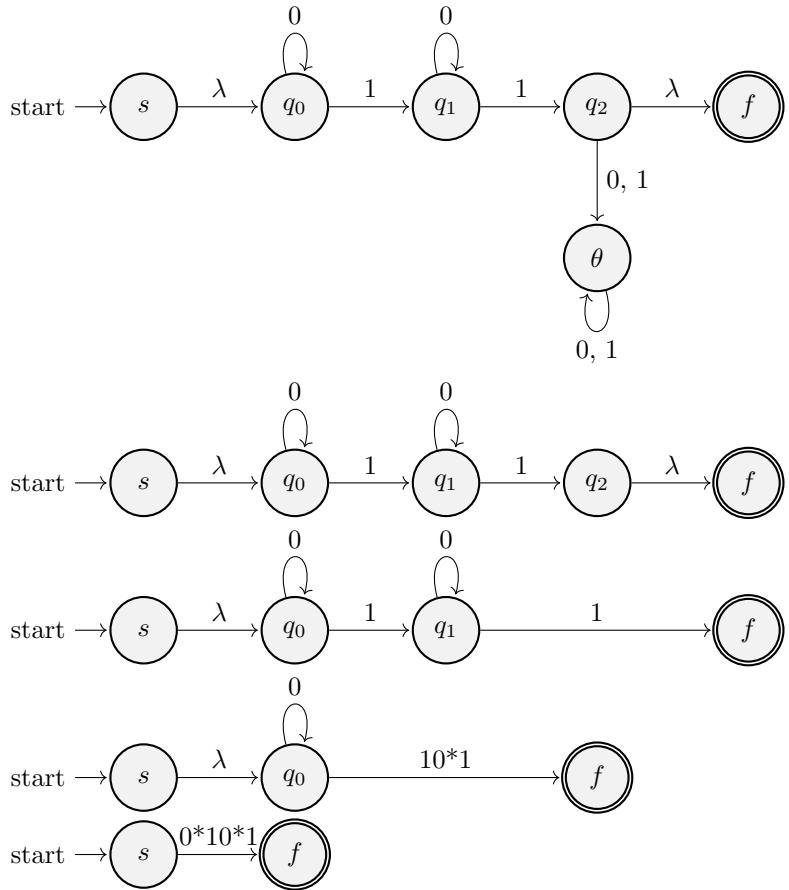


2.12 Q6: Candidate Exam Question



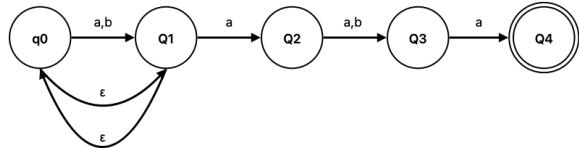
Convert this NFA into a GNFA

Answer:



2.13 Q6: Candidate Exam Question

- i) Convert the following NFA to a DFA using powerset construction and write it out as a 5-tuple.



- ii) The equivalent DFA needs only one state, not 16. Explain why.

Answer

- i) The NFA is $M = (Q, \Sigma, \delta, q_0, F)$ where $Q = \{q_0, q_1, q_2, q_3\}$, $\Sigma = \emptyset$, and $F = \{q_2\}$. The equivalent DFA is $M' = (Q', \Sigma, \delta', r_0, F')$ where:

- $Q' = P(Q) = \{\emptyset, \{q_0\}, \{q_1\}, \{q_2\}, \{q_3\}, \{q_0, q_1\}, \{q_0, q_2\}, \{q_0, q_3\}, \{q_1, q_2\}, \{q_1, q_3\}, \{q_2, q_3\}, \{q_0, q_1, q_2\}, \{q_0, q_1, q_3\}, \{q_0, q_2, q_3\}, \{q_1, q_2, q_3\}\}$
- $\delta'(R, a) = \{q \in Q | q \in \Lambda(\delta(r, a)) \text{ for some } r\} = \begin{cases} \emptyset & R = \emptyset \\ \{q_0, q_1, q_2, q_3\} & R \neq \emptyset \end{cases}$
- $r_0 = \Lambda(q_0) = \{q_0, q_1, q_2, q_3\}$
- $F' = \{R \in Q' | R \cap F \neq \emptyset\} = \{\{q_2\}, \{q_0, q_2\}, \{q_1, q_2\}, \{q_2, q_3\}, \{q_0, q_1, q_2\}, \{q_0, q_2, q_3\}, \{q_1, q_2, q_3\}, \{q_0, q_1, q_2, q_3\}\}$

- ii) None of the DFA's states other than q_0, q_1, q_2, q_3 are reachable. It starts at q_0, q_1, q_2, q_3 , and the "transition" function keeps it at q_0, q_1, q_2, q_3 (in fact the transition function can't even be actually invoked because $\Sigma = \emptyset$).

2.14 Q6: Candidate Exam Question

Let $\Sigma = \{0, 1\}$. Consider the language $L = \{w \mid w \text{ contains at least two consecutive } 1\text{s}\}$.

1. (i) [4 points] Construct a regular expression for the language L .
2. (ii) [6 points] Design an NFA that recognizes the language L .
3. (iii) [8 points] Use the powerset construction to convert your NFA from part (ii) into an equivalent DFA.

Answer

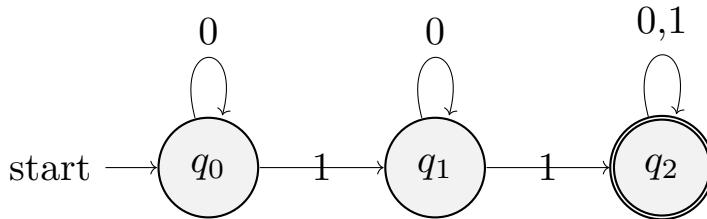
To create a regular expression for L , which consists of strings that contain at least two consecutive 1s, we can describe the solution in the following way:

- Any string of 0s or 1s can precede the first occurrence of the two consecutive 1s.
- After these two consecutive 1s appear, any number of 0s or 1s can follow.

Thus, the regular expression is:

$$(0 \mid 1)^* 11 (0 \mid 1)^*$$

NFA that recognizes the language $L = \{w \mid w \text{ contains at least two consecutive } 1\text{s}\}$.



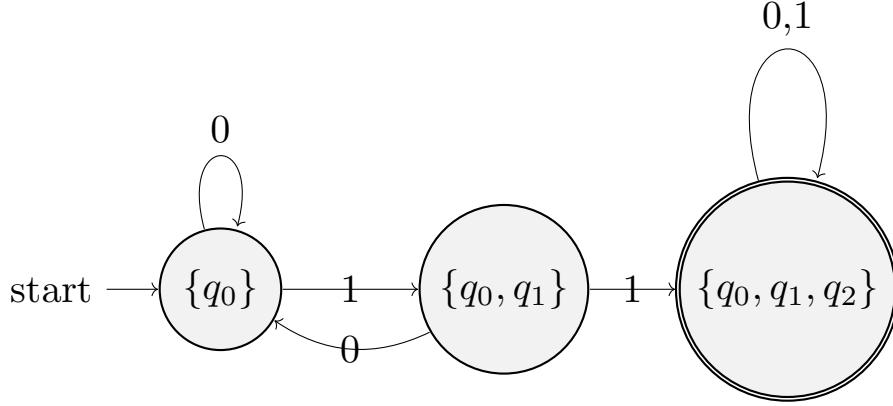
We convert the NFA from part (ii) to a DFA using the *powerset construction* method. The set of states for the NFA is $\{q_0, q_1, q_2\}$. The powerset of these states is:

$$P(Q) = \{\emptyset, \{q_0\}, \{q_1\}, \{q_2\}, \{q_0, q_1\}, \{q_1, q_2\}, \{q_0, q_1, q_2\}\}$$

The powerset construction method yields the following DFA:

- States:
 - $\{q_0\}$: Initial state, no 1s seen yet
 - $\{q_0, q_1\}$: One 1 has been seen
 - $\{q_0, q_1, q_2\}$: At least two consecutive 1s have been seen (accepting state)
- Transitions:
 - From $\{q_0\}$:
 - * On 0: Stay in $\{q_0\}$
 - * On 1: Move to $\{q_0, q_1\}$
 - From $\{q_0, q_1\}$:
 - * On 0: Move back to $\{q_0\}$
 - * On 1: Move to $\{q_0, q_1, q_2\}$ (accepting state)
 - From $\{q_0, q_1, q_2\}$:
 - * On 0: Stay in $\{q_0, q_1, q_2\}$
 - * On 1: Stay in $\{q_0, q_1, q_2\}$

DFA Diagram



2.15 Q6: Candidate Exam Question

a) Consider the string $x = a^pba^pba^pb$ in L_1 (where $w = a^p b$). Assume that L_1 is regular. By the Pumping Lemma, we can pump on the prefix a^p of x . Thus, a^p can be written as $a^p = a^i a^k a^j$, where $i + k + j = p$ and $k > 0$. This allows us to express the string as:

$$x_r = a^i (a^k)^r a^j ba^pba^pb = a^i a^{rk} a^j ba^pba^pb$$

which is in the language L_1 for all $r > 0$.

However, if we let $r = 2p + 2$ (with $k > 0$), the string $x_{2p+2} = a^i a^{2pk+2k} a^j ba^pba^pb$ cannot be of the form www for any $w \in \{a, b\}^*$, and hence cannot be in the language L_1 .

This contradiction proves that the language L_1 is not regular.

2.16 Q6: Candidate Exam Question

Let $\Sigma = \{0, 1\}$.

Find a regular expression for the language:

$$L_3 = \{w \mid w \text{ has an even number of } 1\text{s}\}.$$

The regular expression for this language is:

$$L_3 = (0^* (10^* 1))^* 0^*$$

Explanation:

- 0^* allows any number of 0s (including none) between or around the 1s.
- $10^* 1$ ensures that each block of 1s contains exactly two 1s.
- $(0^* (10^* 1))^*$ repeats this pattern of alternating between blocks with two 1s and any number of 0s.
- The final 0^* allows the string to end with any number of 0s without affecting the count of 1s.

This regular expression captures all strings containing an even number of 1s, while allowing any amount of 0s in between.

2.17 Q6: Candidate Exam Question

What is the difference between a Deterministic Finite Acceptor (DFA) and a Nondeterministic Finite Acceptor (NFA)?

Solution:

- A **Deterministic Finite Acceptor (DFA)** has exactly one transition for each symbol in the input alphabet from any given state. This means that for each input, the DFA can only be in one state at any point in time.

- A **Nondeterministic Finite Acceptor (NFA)**, on the other hand, allows multiple transitions for a given input symbol and state. An NFA can be in several possible states at once, and it can even have transitions that do not consume any input symbols (ε -transitions).

2.18 Q6: Candidate Exam Question

Show that the following is not a regular language:

$$F = \{a^n b^n c^n \mid n \geq 1\}$$

Answer:

We will use the Pumping Lemma for regular languages to prove that F is not regular.

Proof: Towards contradiction, by assuming that F is a regular language with pumping length p .

The Pumping Lemma says that if F is a regular language, then there is a pumping length p , where, if any string $s \in F$ and is also $|s| \geq p$, then s may be divided into three pieces, $s = xyz$ where:

1. $|xy| \leq p$
2. $|y| > 0$
3. $xy^i z \in A$, for every $i \in \{0, 1, 2, 3\dots\}$

The string we will use is $s = a^p b^p c^p$. This has a length $|s| = 3p \geq p$. From property 1, $|xy| \leq p$, so x and y can only consist of a 's. So since y is only made up of a 's, $y = a^l$, which means $l > 0$.

We pump the string now by setting $i = 2$ (from property 3) which gives us:

$$xy^2z = xyyz = a^{p+l}b^p c^p$$

This new string is not in F anymore because the number of a 's is not equal to the number of b 's and c 's which is required in order to be a valid string in F .

Therefore, we have a contradiction.

2.19 Q6: Candidate Exam Question

Question:

Consider the NFA given below with states $Q = \{q_1, q_2\}$, and transitions as:

- From q_1 to q_2 on input 0,
- From q_1 to q_1 or q_2 on input 1,
- From q_2 to q_2 on input 0.

Convert this NFA to a DFA using the powerset construction, and provide the resulting DFA.

Solution:

To apply the powerset construction:

1. The NFA has two states, $Q = \{q_1, q_2\}$. The powerset of Q is $P(Q) = \{\emptyset, \{q_1\}, \{q_2\}, \{q_1, q_2\}\}$.
2. Start by labeling the DFA states as corresponding subsets of Q . The new DFA states are $\emptyset, \{q_1\}, \{q_2\}, \{q_1, q_2\}$.

3. For each input 0 and 1, calculate the transitions:

- **From state $\{q1\}$:**

- On input 0, the NFA transitions to $q2$, so $\{q1\} \xrightarrow{0} \{q2\}$.
- On input 1, the NFA transitions to both $q1$ and $q2$, so $\{q1\} \xrightarrow{1} \{q1, q2\}$.

- **From state $\{q2\}$:**

- On input 0, the NFA stays in $q2$, so $\{q2\} \xrightarrow{0} \{q2\}$.
- On input 1, there is no transition defined, so it goes to \emptyset .

- **From state $\{q1, q2\}$:**

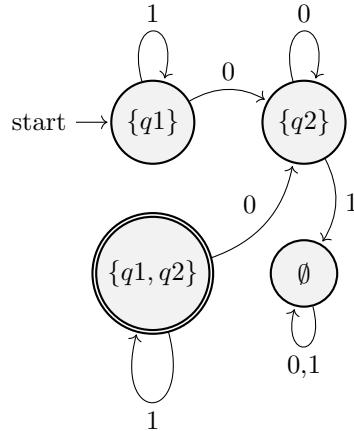
- On input 0, the NFA transitions to $q2$, so $\{q1, q2\} \xrightarrow{0} \{q2\}$.
- On input 1, the NFA transitions to $q1$ and $q2$, so $\{q1, q2\} \xrightarrow{1} \{q1, q2\}$.

- **From state \emptyset :**

- On both inputs, it remains in \emptyset .

4. The DFA has four states: \emptyset , $\{q1\}$, $\{q2\}$, $\{q1, q2\}$. The initial state is $\{q1\}$, and the accepting states depend on the accepting state in the NFA.

DFA Diagram:



2.20 Q6: Candidate Exam Question

Let the language L be defined as:

$$L = \{w \mid w \text{ is a binary string that ends with } 10 \text{ or } 01\}.$$

The alphabet of the language is $\Sigma = \{0, 1\}$.

1. Draw an NFA for the above language.
2. Convert the NFA into DFA using Powerset construction.

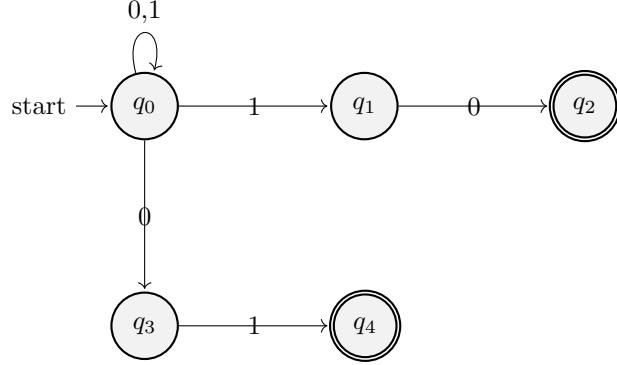
2.21 NFA Construction

The NFA for the language L is defined as follows:

- States: $Q = \{q_0, q_1, q_2, q_3, q_4\}$
- Alphabet: $\Sigma = \{0, 1\}$

- Start state: q_0
- Accept states: $F = \{q_2, q_4\}$
- Transition function δ :

NFA Diagram



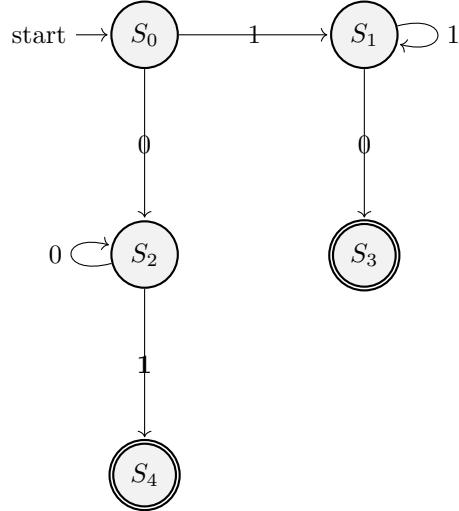
2.22 2. DFA using Powerset Construction

Using the powerset construction, we derive the following DFA states:

- **DFA States**:
 - $S_0 = \{q_0\}$ (Start state)
 - $S_1 = \{q_0, q_1\}$
 - $S_2 = \{q_0, q_3\}$
 - $S_3 = \{q_0, q_2, q_3\}$ (Accepting state)
 - $S_4 = \{q_0, q_1, q_4\}$ (Accepting state)
- **Transition Table**:

DFA State	0	1
$S_0 = \{q_0\}$	$S_5 = \{q_0, q_3\}$	$S_6 = \{q_0, q_1\}$
$S_1 = \{q_0, q_1\}$	$S_3 = \{q_0, q_2, q_3\}$	$S_1 = \{q_0, q_1\}$
$S_2 = \{q_0, q_3\}$	$S_2 = \{q_0, q_3\}$	$S_4 = \{q_0, q_1, q_4\}$
$S_3 = \{q_0, q_2, q_3\}$	(no transition)	(no transition)
$S_4 = \{q_0, q_1, q_4\}$	(no transition)	(no transition)

2.23 DFA Diagram



The final accepting states in the DFA will correspond to the sets containing q_2 or q_4 .

2.24 Q6: Candidate Exam Question

Show that the following Language is not regular: $L = \{w \in \{0,1\}^*: w \text{ has same number of 0's and 1's}\}$

i.e 0011, 0101, 1001

Solution: Proof by Contradiction of Closure Property.

1. Assume L is regular.
2. If L is regular, L should be closed under intersection of another regular language (regex).
3. $L' = L \cap 0^*1^*$ (regex) = $\{0^n1^n : n \geq 0\}$
 - (a) Assume L' is regular.
 - (b) Assume $\exists p$ (pumping constant) for L' .
 - (c) Choose $w = 0^p1^p$.
 - (d) Decompose w into XYZ s.t.
 - i. $|XY| \leq p$
 - ii. $|Y| \geq 1$
 - A. $X = 0^\alpha, \alpha \geq 0, \alpha + \beta \leq p$
 - B. $Y = 0^\beta, \beta \geq 1$
 - C. $Z = 0^{p-\alpha-\beta}1^p$
 - (e) Choose i s.t. $XY^iZ \notin L'$
 - i. $XY^iZ = 0^\alpha 0^{\beta^i} 0^{p-\alpha-\beta} 1^p$
 - ii. $XY^iZ = 0^{p+i\beta-\beta} 1^p$
 - iii. $p + i\beta - \beta \neq p$, (remove p)
 - iv. $i\beta - \beta \neq 0$, (plus β)
 - v. $i\beta \neq \beta$, (divide β)
 - vi. $i \neq 1$
 - vii. When i is any value other than 1, L' is not regular due to proof by contradiction via the Pumping Lemma Theorem.
4. Since L' which was generated from $L \cap 0^*1^*$ is proven to not be regular, L is shown to not be regular.

2.25 Q6: Candidate Exam Question

Let $\Sigma = \{a, b, c\}$. Give regular expressions for the following language.

- $L_1 = \{w \mid w \text{ contains at least three a's and ends with two b's, or contains the substring cabab}\}$.

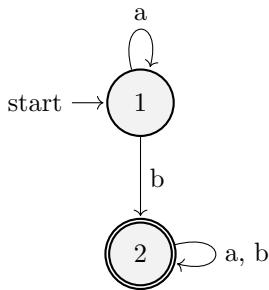
$$\Sigma^* a \Sigma^* a \Sigma^* a \Sigma^* bb + \Sigma^* cabab \Sigma^*$$

- $L_2 = \{w \mid w \text{ contains any arrangement of a or b, or only contains c's and has length 12}\}$.

$$(a+b)^* + c^{12}$$

2.26 Q6: Candidate Exam Question

How many steps are required in order to convert the following DFA into a regular expression? Explain Each step.



- Create a start and accept state with λ transitions.
- Remove state 2, connecting state 1 with the accept state with the expression $b(a+b)^*$.
- Remove state 1, connecting the start state with the accept state with the expression becoming $(a)^*b(a+b)^*$.

2.27 Q6: Candidate Exam Question

Let $\Sigma = \{a, b, c, d\}$. Give the regular expression for the following language.

$$L = \{w \mid w \text{ starts with } a, \text{ contains substring } bb, \text{ ends with substring } ccc\}$$

Solution: $a(a+b+c+d)^*bb(a+b+c+d)^*ccc$

The question is simple and aims to challenge the precision of the answers. Intuitively you would think $abbccc$ would be the answer however you would need to take in account of the characters between a and bb as well as bb and ccc .

2.28 Q6: Candidate Exam Question

Let $\Sigma = \{0, 1\}$. State the Pumping Lemma theorem and that knowledge to prove the language below is not regular.

$$L = \{0^m 1 0^n \mid m, n \geq 0, m = n \text{ or } m = 2n\}$$

Proposed Answer

By their of the pumping lemma, we have the pumping length of p and s as a string.

$$s = xyz$$

Three cases need to satisfy the condition to prove L is not regular:

$$|y| > 0$$

$$|xy| \leq p$$

for every $k \geq 0, xy^k z \in L$

We first assume L is regular for later proving contradiction. There exist p so that $|w| \geq p$. we can choose string $0^{p-m}10^21$. First, we consider $y = 0^m$ where $0 < m \leq p$ and $|xy| \leq p$. This will keep $|xy| \leq p$. This means that y must have multiple input strings of 0. Finally, we choose string $xy^i z = 0^{p-m}10^21$. The string in L has a length of $3p+1$, which is greater than p . Such string cannot be in L because the first series of 0 followed by 1 will have another set of different is no longer have the same length as the second series of 0 after 1. Double the amount of the series of 0 after 1 is also not in L . This a contradiction because there exist a string in L that cannot be pumped for any pumping length p . As a result, the language is not regular.

2.29 Q6: Candidate Exam Question

Let $\Sigma = \{0, 1\}$. Give regular expressions for the following languages.

- (i) $L_a = \{w \mid w \text{ contains at least two } 1s\}$.

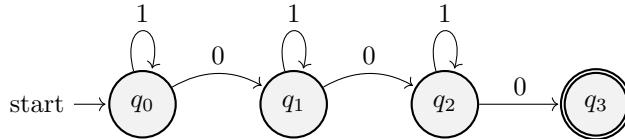
Answer: $(0 + 1)^*1(0 + 1)^*1(0 + 1)^*$

- (ii) $L_b = \{w \mid w \text{ starts and ends with the same symbol}\}$.

Answer: $(0 + 1) + (0(0 + 1)^*0) + (1(0 + 1)^*1)$

2.30 Q6: Candidate Exam Question

Convert the following DFA to its equivalent regular expression.



Where $\Sigma = \{0, 1\}$ represented by the language

$$L = \{w \mid w \text{ starts with } 0 \text{ or } 1, \text{ contains at least two } 0's, \text{ and ends with } 0\}.$$

- (i) **Create Regular Expression on q_0 .**

From q_0 to q_1 , there is a 0. On q_0 there exists a self-loop for a 1 for any amount of 1's. Thus, to get from q_0 to q_1 , we get the following regular expression.

$$1^*0$$

- (ii) **Create Regular Expression on q_1 .**

From q_1 to q_2 there is a 0. On q_1 there exists a self-loop for a 1 for any amount of 1's. Thus, to get from q_0 to q_2 , we get the following regular expression.

$$1^*0(1^*)0$$

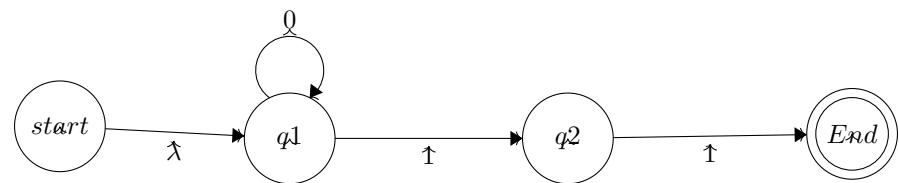
- (iii) **Create Regular Expression on q_2 .**

From q_2 to q_3 , there is a 0. On q_2 there exists a self-loop for a 1 for any amount of 1's. Thus, to get from q_0 to q_3 , we get the following regular expression.

$$(1^*)0(1^*)0(1^*)0$$

2.31 Q6: Candidate Exam Question

Q: Create an NFA state diagram that: $L_1 = \{w \mid w \text{ ends with two } 1\}$.



3 Student Proposed Problems for the 4th Assignment

3.1 Q6: Candidate Exam Question

Prove that $L = \{a^{n^2} : n \geq 0\}$ is not a context-free language.

Answer:

Proof. Towards contradiction, assume L is a CFL with pumping length p. Let us look at the string $s = a^{p^2}$. As $|s| = p^2 \geq p$, we can apply the pumping lemma for CFLs.

This means that $s = uvxyz$ s.t. 1. for each $i \geq 0$, $uv^i xy^i z \in L$ 2. $|vy| > 0$ 3. $|vxy| \leq p$

Let us pump up with $i = 2$. Thus, we are looking at $s' = uv^2 xy^2 z$. Therefore,

$$|s'| = |uvxyz| + |vy| \quad (2)$$

$$= p^2 + |vy| > p^2 \quad (3)$$

Let $|vxy| = k \leq p$. Thus, $|vy| = k - |x| \leq k \leq p$ as x can be the empty string. Therefore,

$$|s'| \leq p^2 + p = p(p+1) \quad (4)$$

$$p^2 < |s'| \leq p(p+1) \quad (5)$$

The next perfect square after p^2 is $(p+1)^2$. However, $p(p+1) < (p+1)^2$. Thus,

$$p^2 < |s'| < (p+1)^2 \quad (6)$$

Since $|s'|$ is within the interval of two consecutive perfect squares, non-inclusive, its length cannot be a perfect square. Therefore, $s' \notin L$, meaning L cannot be context-free.

□

3.2 Q6: Candidate Exam Question

Show that the following language is not context-free.

1. $L = \{w \in \{0, 1, 2\}^* \mid w \text{ has an equal number of 0's, 1's, and 2's}\}$

Answer:

Proof. To prove L is not context-free, we need to use the pumping lemma for context-free languages.

We first apply the pumping lemma by assuming that L is context-free. Then, there must exist a pumping length p such that any string $s \in L$ with $|s| \geq p$ can be decomposed as $s = uvxyz$ with the following conditions:

- For each $i \geq 0$, $uv^i xy^i z \in L$,
- $|vy| > 0$,
- $|vxy| \leq p$.

A string is constructed that satisfies the condition of having equal numbers of 0's, 1's, and 2's, and has a length at least p : In this case, let $s = 0^p 1^p 2^p$, where each symbol has length p and is contained in L .

Using the pumping lemma, s is split into $s = uvxyz$ where the conditions $|vxy| \leq p$ and $|vy| > 0$ hold. The substring vxy can contain at most two types of symbols because its length is at most p .

In this proof, we consider two cases for v and y :

1. The first case covers v and y containing only one type of symbol, where the symbols either contain only 0's, 1's, or 2's. Then, pumping v and y would increase the count of those symbols without affecting the counts of the other symbols, leading to an unequal number of 0's, 1's, and 2's, which is not contained in L .
2. The second case covers v and y containing two different symbols. For example, let v contain 0's and y contain 1's. Pumping v and y would lead to an unequal number of 0's and 1's as well as an unequal number of 2's, which is also not contained in L .

We have shown in both cases that there is a contradiction with pumping s , since it would lead to an unequal number of 0's, 1's, and 2's, which is not contained in L . Thus, we have proven that L is not context-free.

□

3.3 Q6: Candidate Exam Question

Show that $L = \{a^n b^m \mid m = n^2\}$ is not context-free.

Answer:

Proof. Using the pumping lemma, assume the contrary, and let $s = uvxyz = a^p b^{p^2}$. The lemma says $uv^i xy^i z \in L$ for any $i \geq 0$. Clearly, if either v or y straddles the boundary between a 's and b 's, pumping s will generate strings not in L . If v and y are composed entirely of a 's, then $uv^2 xy^2 z = a^{p+j} b^{p^2}$, $0 < j \leq p$, which is not in L since the number of b 's is no longer the square of the number of a 's. The same argument holds if $vy = b^k$. The only remaining case is when v is one or more a 's and y is one or more b 's. If this is the case, then we have $uv^2 xy^2 z = a^{p+j} b^{p^2+k}$, where $j, k > 0$. But $p^2 + k < (p+1)^2$ since $(p+1)^2 = p^2 + 2p + 1$ and $k < p$ (because $k+j \leq p$). Since $p^2 + k$ cannot be any perfect square, it certainly cannot be $(p+j)^2$ for any $j > 0$. Since every case results in a contradiction, L is not context-free. \square

3.4 Q6: Candidate Exam Question

Design a PDA recognizing the language L_5 . Let L_5 be the language defined as:

$$L_5 : A = \{w \in \{0,1\}^* \mid w \text{ contains at least three } 1\text{'s}\}$$

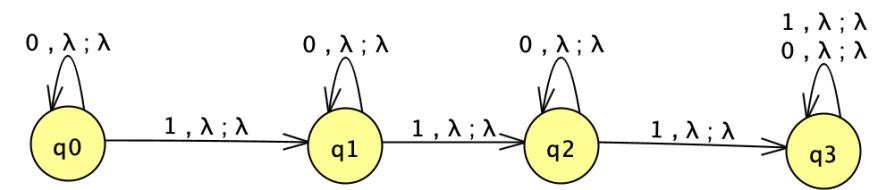


Figure 2: PDA representing L_5

Answer:

3.5 Q6: Candidate Exam Question

Consider the Context-Free Grammar (CFG) for this question:

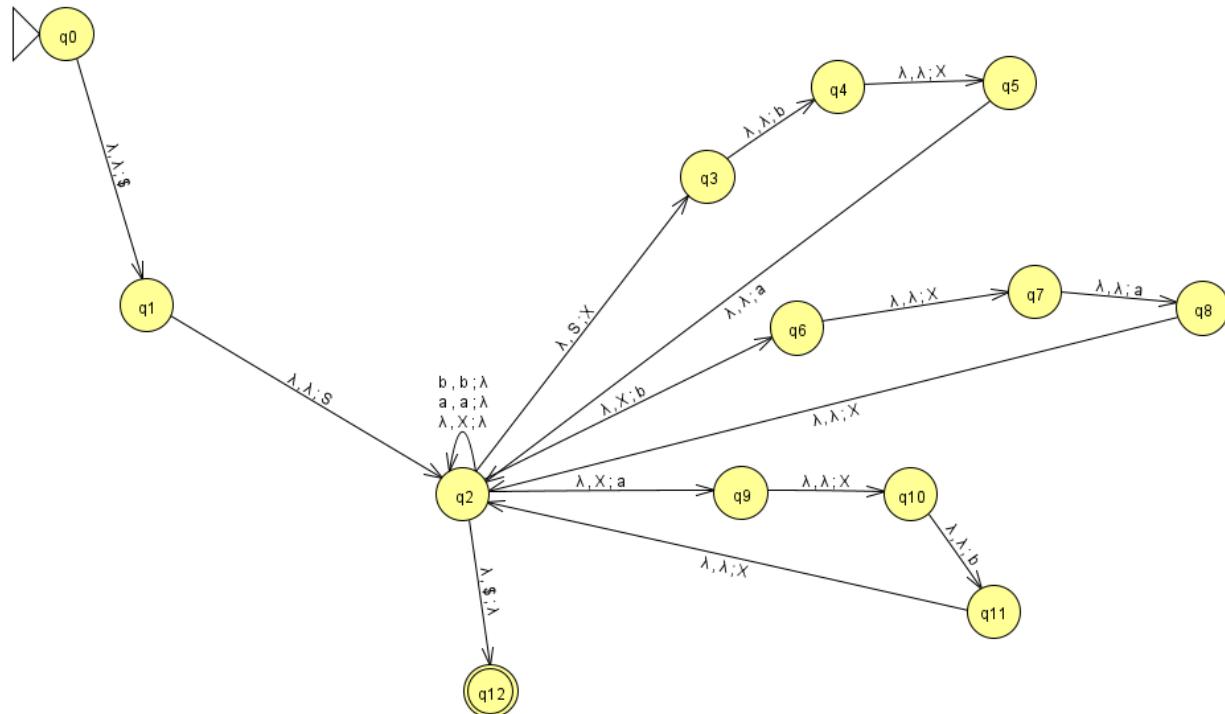
$$\begin{aligned} S &\rightarrow aXbX \\ X &\rightarrow XaXb \mid XbXa \mid \lambda \end{aligned}$$

- What Language does the CFG accept?
- Convert the CFG to PDA

Answer:

- The CFG accepts the language where number of a's = number of b's

CFG to PDA below:



Input	Result
ab	Accept
aa	Cancelled
aab	Cancelled
ababab	Accept
aaabbb	Accept
b	Reject
ba	Reject
aabb	Accept
	Reject

3.6 Q6: Candidate Exam Question

Convert the following context-free grammar into Chomsky Normal Form (CNF).

The grammar is defined as follows:

$$\begin{aligned} S &\rightarrow AB \mid CA \\ A &\rightarrow aA \mid B \mid a \\ B &\rightarrow bB \mid C \mid b \\ C &\rightarrow AB \mid \epsilon \end{aligned}$$

Answer: Step 1

Add New Start Variable:

$$\begin{aligned} \langle S_0 \rangle &\rightarrow S \\ S &\rightarrow AB \mid CA \\ A &\rightarrow aA \mid B \mid a \\ B &\rightarrow bB \mid C \mid b \\ C &\rightarrow AB \mid \epsilon \end{aligned}$$

Step 2

Remove All ϵ Rules:

$$\begin{aligned} \langle S_0 \rangle &\rightarrow S \mid \epsilon \\ S &\rightarrow AB \mid CA \mid A \mid B \mid C \\ A &\rightarrow aA \mid B \mid a \\ B &\rightarrow bB \mid C \mid b \\ C &\rightarrow AB \mid A \mid B \end{aligned}$$

Step 3

Remove All Unit Rules:

$$\begin{aligned} \langle S_0 \rangle &\rightarrow \epsilon \mid AB \mid CA \mid aA \mid a \mid bB \mid b \\ S &\rightarrow AB \mid CA \mid aA \mid a \mid bB \mid b \\ A &\rightarrow aA \mid a \mid bB \mid b \mid AB \\ B &\rightarrow bB \mid b \mid AB \\ C &\rightarrow AB \end{aligned}$$

Step 4

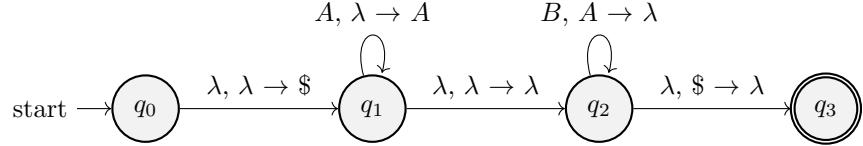
Chomsky Normal Form

$$\begin{aligned} \langle S_0 \rangle &\rightarrow \epsilon \mid AB \mid CA \mid a \mid b \mid \langle Z_1 \rangle A \mid \langle Z_2 \rangle B \\ S &\rightarrow AB \mid CA \mid a \mid b \mid \langle Z_1 \rangle A \mid \langle Z_2 \rangle B \\ A &\rightarrow a \mid b \mid AB \mid \langle Z_1 \rangle A \mid \langle Z_2 \rangle B \\ B &\rightarrow b \mid AB \mid \langle Z_2 \rangle B \\ C &\rightarrow AB \\ \langle Z_1 \rangle &\rightarrow a \\ \langle Z_2 \rangle &\rightarrow b \end{aligned}$$

3.7 Q6: Candidate Exam Question

Construct a PDA from the given language $\{A^n B^n \mid n \geq 0\}$, $\Sigma = \{A, B\}$

Answer:



3.8 Q6: Candidate Exam Question

Consider the language $L = \{0^m 1^n \mid m \neq n\}$ over the alphabet $\Sigma = \{0, 1\}$.

1. Show that L is not a context-free language using the Pumping Lemma for context-free languages.
2. If L were modified to include strings where $m \leq n$ (i.e., $L' = \{0^m 1^n \mid m \leq n\}$), would L' be recognized by a Pushdown Automaton? Provide a brief justification for your answer.

Answer:

Part 1: Showing $L = \{0^m 1^n \mid m \neq n\}$ is Not Context-Free

To show that L is not context-free, we'll use the **Pumping Lemma for context-free languages**, which provides a way to demonstrate that certain languages are not context-free by finding contradictions when "pumping" portions of strings within the language.

1. **Assume L is Context-Free:** We start by assuming, for contradiction, that L is a context-free language. According to the Pumping Lemma, there exists a pumping length p such that any string s in L with $|s| \geq p$ can be split into five parts, $s = uvwxy$, with specific properties:
 - $|vwx| \leq p$ (the "pumped" portion is limited in length),
 - $|vx| > 0$ (the pumped section is non-empty),
 - For any $i \geq 0$, $uv^iwx^i y \in L$.
2. **Choosing a String s in L :** Let's select a string $s = 0^p 1^{p+1}$ in L . This string has $m = p$ occurrences of 0 and $n = p + 1$ occurrences of 1, so $m \neq n$, meaning $s \in L$. The length $|s| = p + (p + 1) = 2p + 1$, which satisfies $|s| \geq p$.
3. **Applying the Pumping Lemma:** According to the Pumping Lemma, we can split $s = uvwxy$ with $|vwx| \leq p$ and $|vx| > 0$. Since $|vwx| \leq p$, the substring vwx consists only of 0s (it's in the first p characters of s , which are all 0s). This means that v and x only contain 0s, so pumping v and x (repeating them) will only add more 0s to the string.
4. **Pumping v and x :** If we increase the number of 0s by choosing $i = 2$, the resulting string becomes uv^2wx^2y , which has more than p 0s. Specifically, it now has more than $p + 1$ 0s, making $m \geq n$, which contradicts the original property $m \neq n$ for strings in L . Thus, $uv^2wx^2y \notin L$, as the pumped string does not satisfy the language condition $m \neq n$.
5. **Conclusion:** Since pumping v and x leads to a string that is not in L , this contradicts the Pumping Lemma. Therefore, L is not a context-free language.

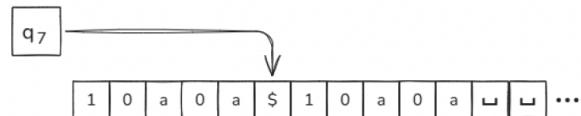
Part 2: Determining if $L' = \{0^m 1^n \mid m \leq n\}$ is Recognizable by a PDA

In this part, we consider a modified version of the language L , where we allow strings with **equal or more 1s than 0s**. We'll see if this modified language L' can be recognized by a **Pushdown Automaton (PDA)**.

1. **Understanding the Language L' :** The language $L' = \{0^m 1^n \mid m \leq n\}$ includes all strings where the number of 0s is less than or equal to the number of 1s. This means strings like $0^2 1^3$ (where $m = 2$ and $n = 3$) and $0^3 1^3$ (where $m = n = 3$) are included, but strings like $0^4 1^3$ are not, as they have more 0s than 1s.
2. **Constructing a PDA for L' :** A PDA can recognize a language like L' by using its stack to count the number of 0s and 1s:
 - The PDA starts by pushing a symbol onto the stack for each 0 it reads.
 - When it begins reading 1s, it pops a symbol from the stack for each 1.
 - If there are more 1s than 0s (i.e., the stack becomes empty while still reading 1s), the PDA can continue to accept these additional 1s without needing to push any more symbols.
 - If the PDA encounters an unmatched 0 (meaning more 0s than 1s by the end), it will reject.
3. **Justification:** The PDA can recognize L' because it only needs to keep track of the difference between the number of 0s and 1s. Since a PDA can push and pop symbols to handle counting and comparison, it is well-suited for this task. The PDA can accept any string where the number of 0s is less than or equal to the number of 1s.
4. **Conclusion:** Yes, L' can be recognized by a PDA, as the stack allows the PDA to compare the counts of 0s and 1s. The PDA will accept strings with $m \leq n$ by correctly balancing pushes and pops on the stack, making this language context-free.

3.9 Q6: Candidate Exam Question

Given the Turing machine diagram below, write the Turing machine configuration that describes it.



Answer: Configuration: $10a0aq_7\$10a0a$

3.10 Q6: Candidate Exam Question

Using the construction from class, construct a PDA that accepts the language L_1 defined by the grammar:

$$S \longrightarrow aSb \mid bSa \mid \lambda$$

Answer: The solution to this problem is shown in the figure below.

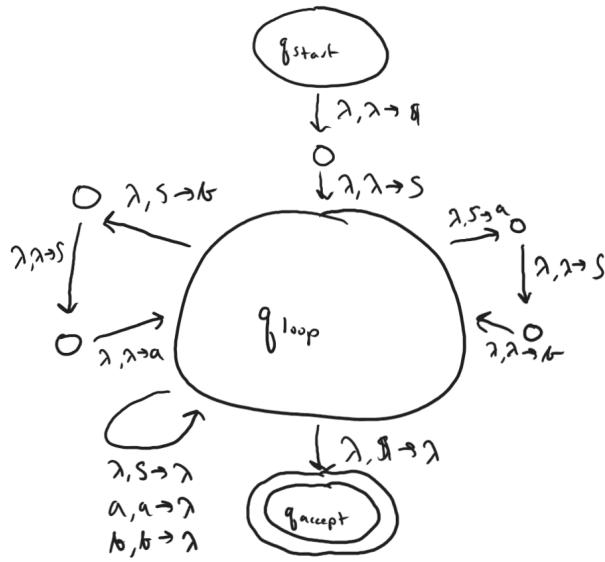


Figure 3: Solution to PDA Construction Problem

3.11 Q6: Candidate Exam Question

Answer with True or False and explain your answer:

The language $L = \{0^n 1^n \mid n \geq 1\}$ can be recognized by a deterministic finite automaton (DFA).

Answer: False. The language $L = \{0^n 1^n \mid n \geq 1\}$ needs memory to match the number of 0s with the number of 1s. This means that a deterministic finite automaton cannot handle recognize it because of its limited state-based memory. In order to recognize this language, we would need a pushdown automaton because it has access to a stack.

3.12 Q6: Candidate Exam Question

Let L be the language defined as:

$$L = \{a^n b^n c^n \mid n \geq 0\}$$

where a , b , and c are symbols in the alphabet $\Sigma = \{a, b, c\}$.

Using the Pumping Lemma for context-free languages, prove that L is **not** a context-free language.

Answer: To prove that L is not context-free, use the following steps based on the Pumping Lemma for context-free languages:

1. Assume that L is context-free.
2. By the Pumping Lemma, there exists a pumping length p such that any string $s \in L$ with $|s| \geq p$ can be written as $s = uvwxy$ with:
 - $|vwx| \leq p$,
 - $|vx| \geq 1$, and
 - For all $k \geq 0$, $uv^k wx^k y \in L$.
3. Choose a string $s \in L$ that satisfies $|s| \geq p$. Choose $s = a^p b^p c^p$.
4. Show that for any decomposition of s as $uvwxy$ that meets the conditions above, pumping v and x will produce a string that is not in L . This will lead to a contradiction, proving that L is not context-free.

Proof. 1. Assume that L is context-free. Then, by the Pumping Lemma for context-free languages, there exists a pumping length p such that any string $s \in L$ with $|s| \geq p$ can be written as $s = uvwxy$ with $|vwx| \leq p$, $|vx| \geq 1$, and for all $k \geq 0$, $uv^k wx^k y \in L$.

2. $s = a^p b^p c^p$, which is a string in L with $|s| \geq p$.

Since $|vwx| \leq p$, the substring vwx can contain symbols from at most two of the three groups of symbols a , b , or c , because vwx is confined to a length of p . This implies that either v and x contain only a 's, only b 's, or only c 's, or they contain a combination of two of these symbols but not all three.

Consider the cases: Case 1. If v and x contain only a 's and/or b 's, pumping s (i.e., increasing k in $uv^k wx^k y$) will change the number of a 's and/or b 's without affecting the number of c 's. then the resulting string will not have equal numbers of a 's, b 's, and c 's and will therefore not belong to L .

Case 2. If v and x contain only b 's and/or c 's, pumping s will alter the number of b 's and/or c 's without affecting the number of a 's, resulting in a string that is also not in L .

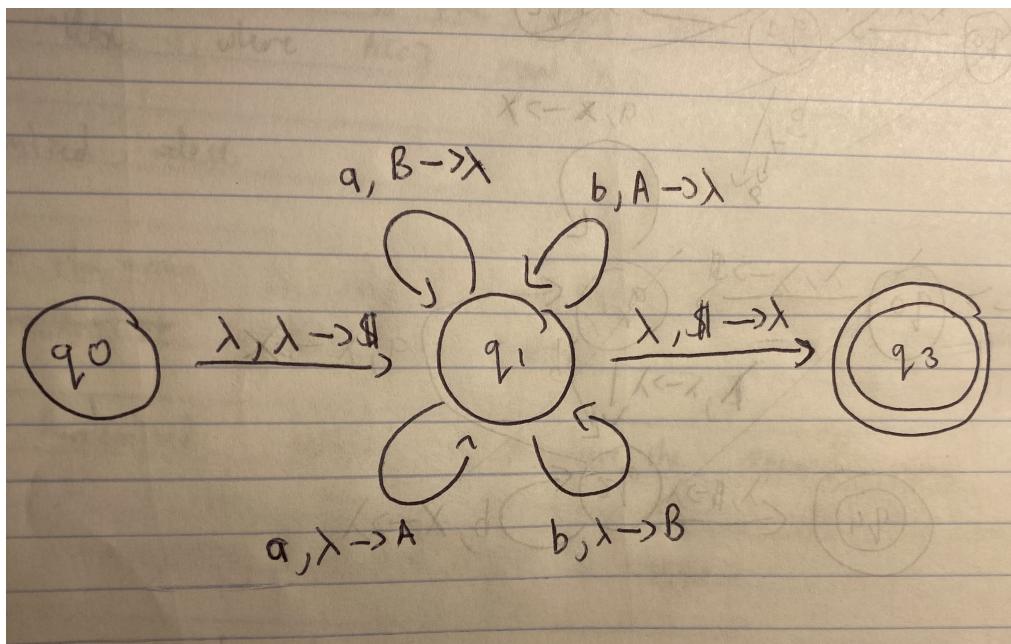
In all cases, pumping v and x leads to a contradiction, it produces strings that do not belong L . Therefore, L does not satisfy the Pumping Lemma for context-free languages, so L is **not** a context-free language. □

3.13 Q6: Candidate Exam Question

Create a PDA that recognizes the language L:

$$L = \{w \in \{a, b\}^*: \#_a = \#_b\}$$

Answer:



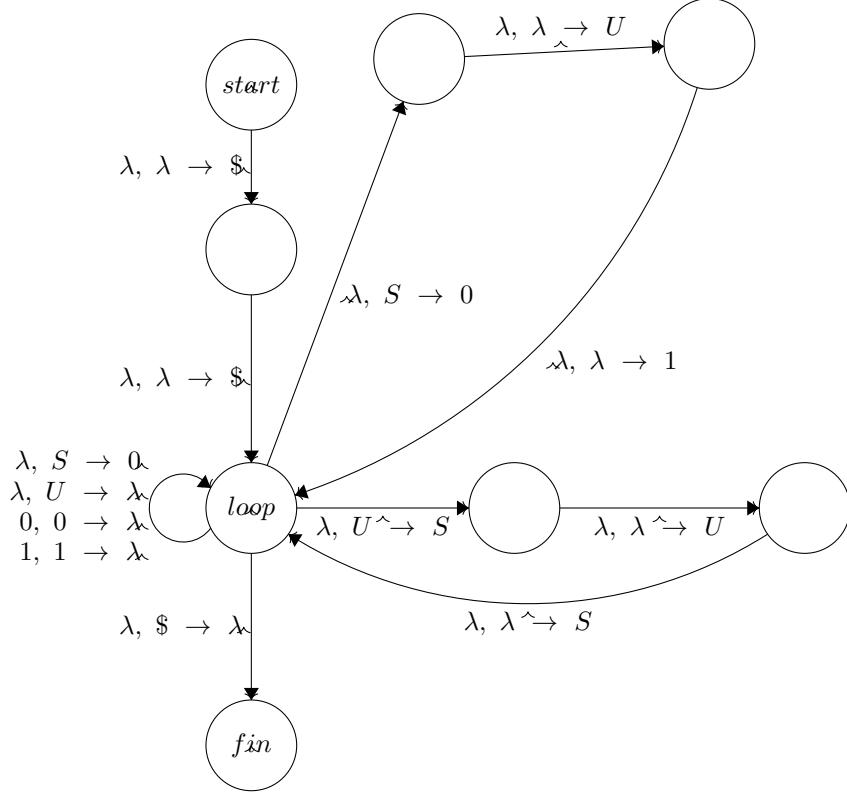
3.14 Q6: Candidate Exam Question

Construct a pushdown automata that accepts the language L_1 defined by the grammar:

$$S \rightarrow 0|1U0$$

$$U \rightarrow SUS|\lambda$$

Answer



3.15 Q6: Candidate Exam Question

Proof that $L_2 = \{1^n \mid n \text{ is prime}\}$ is Not Context-Free

To prove that L_2 is not context-free, we will use the pumping lemma for context-free languages. The pumping lemma states that for any context-free language L , there exists a constant p (the pumping length) such that any string s in L of length at least p can be decomposed into five parts $s = uvwxy$ satisfying the following conditions:

1. $|vwx| \leq p$
2. $|vx| \geq 1$
3. For all $i \geq 0$, the string $uv^iwx^i y \in L$.

Step 1: Assume L_2 is context-free

Assume, for the sake of contradiction, that L_2 is context-free. By the pumping lemma, there exists a pumping length p .

Step 2: Choose a string s

Let us choose the string $s = 1^q$, where q is a prime number greater than p . Clearly, $s \in L_2$ because q is prime.

Step 3: Decompose s

According to the pumping lemma, we can write $s = uvwxy$ such that:

- $|vwx| \leq p$
- $|vx| \geq 1$

Since $|vwx| \leq p$ and $p < q$, the segments v and x can only consist of 1s.

Let:

$$v = 1^m \quad (m \geq 0), \quad x = 1^n \quad (n \geq 0)$$

thus:

$$|v| + |x| \geq 1.$$

Step 4: Pump the string

Now, consider uv^2wx^2y :

$$uv^2wx^2y = 1^{q+m+n}$$

This new string must also be in L_2 . The length of this string is $q + m + n$.

Step 5: Analyze the result

1. If $m + n = 0$, then $|vx| = 0$, which contradicts our earlier assumption that $|vx| \geq 1$.
2. If $m + n > 0$, then $q + m + n$ is not prime. This is because q is prime and adding a positive integer (either m or n) results in a composite number. Thus, $q + m + n$ cannot be prime since we would have created a non-prime length.

3.16 Q6: Candidate Exam Question

Explain why Turing machines can recognize more languages than Pushdown Automata, and how this relates to the concept of decidability.

Solution:

- A **Pushdown Automaton** can only recognize context-free languages, as it relies on a single stack for memory. This limits its ability to handle languages with increased complexity. By contrast, a **Turing Machine** can recognize more languages due to its usage of an unlimited read/write tape as memory. It can move in both directions, allowing it to recognize languages that require more than a single stack for memory (i.e. languages that are not context-free).

- Turing machines can also be used to address **decidability** - deciding whether a problem can be solved algorithmically. A problem is decidable if there exists a Turing machine that halts on every input and correctly determines whether it belongs to a specific language.

- It is important to note the **halting problem**, which questions whether a given Turing machine will halt on a particular input. This problem is undecidable, meaning there does not exist a Turing machine that can decide for all possible inputs.

3.17 Q6: Candidate Exam Question

Let L_5 be the language $\{a^n b^n c^n \mid n \geq 0\}$ over $\Sigma = \{a, b, c\}$

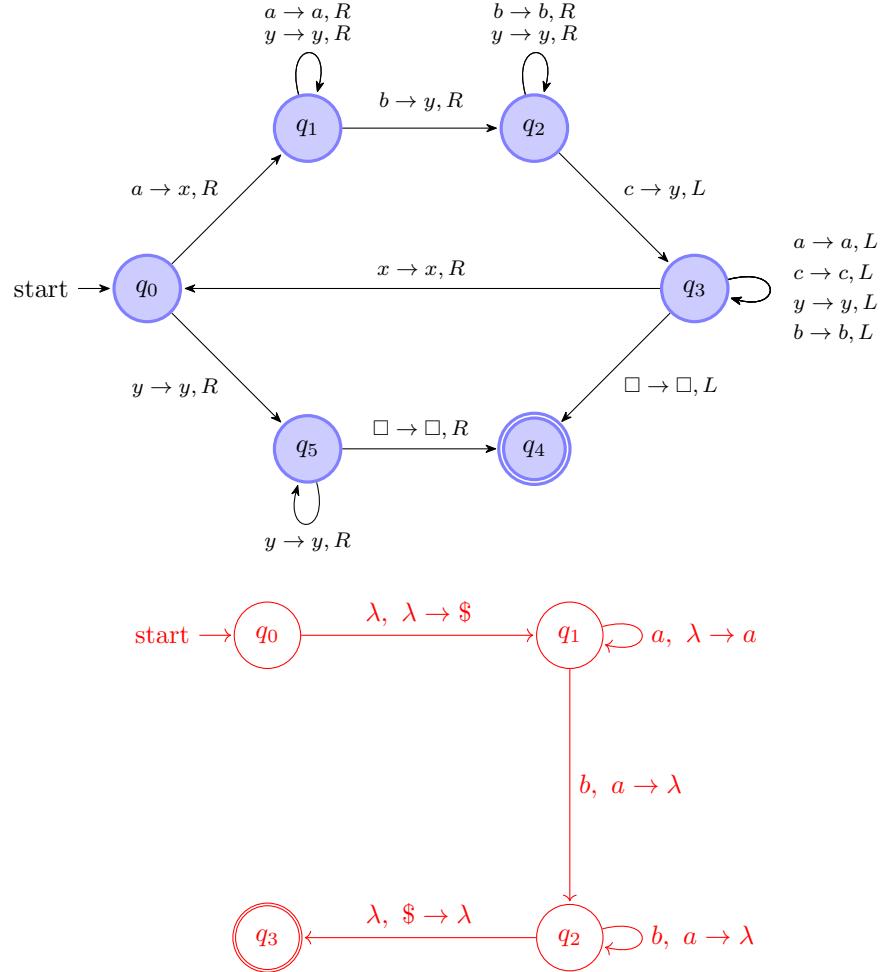
It will be assumed that all the transitions not depicted go to the reject state, which may or may not be drawn.

1. Draw the state diagram for M_1 , a deterministic Turing Machine that recognizes L1.

3.18 Q6: Candidate Exam Question

1. **Question:** Find the CFG that generates the language accepted by the pushdown automata $M = \{Q, \Sigma, \Gamma, \delta, q_{start}, F\}$, where:

- $Q = \{q_0, q_1, q_2, q_3\}$ is the set of valid states,
- $\Sigma = \{a, b, \lambda\}$ is the input alphabet,
- $\Gamma = \{a, \lambda, \$\}$ is the stack alphabet,
- $q_{start} = q_0$ is the start state,
- $F = \{q_3\}$ is the set of accept states, and



- $\delta : Q \times \Sigma \times \Gamma \rightarrow \mathcal{P}(Q \times \Gamma)$ is defined by the following state diagram:

2. **Answer:** $S \longrightarrow aSb \mid \lambda$

3.19 Q6: Candidate Exam Question

Using the Pumping Lemma, prove that the language $L = \{w \in \{a,b\}^* \mid w \text{ has twice as many a's as b's}\}$ is not context free.

Answer Proof: Towards contradiction by assuming that L is a context-free language.

The Pumping Lemma says that if L is context-free, then there is a pumping length p , where, if any string $s \in L$ and is also $|s| \geq p$, then s may be divided into $s = uvwxy$ where:

1. $|vwx| \leq p$
2. $|vx| > 1$
3. $uv^iwx^i y \in L$, for every $i \geq 0$.

The string we will use is $s = a^{2p}b^p \in L$. This has twice as many a's as it does b's. This has a length $|s| = 3p$ which does meet the requirement of the Pumping Lemma. Then divide $s = uvwxy$ so that $|vwx| \leq p$ and $|vx| \geq 1$. At this point, since $|vwx| \leq p$, this substring can only contain a's, which implies that v and x only contain a's. We will now pump up v and x by setting $i = 2$.

This will add more a's but keep the same number of b's. This gives us the string: uv^2wx^2y , which has more than twice the number of a's when compared to b's. Therefore, this pumped up version is $\notin L$. Therefore, we have a contradiction which shows that L cannot be context-free.

3.20 Q6: Candidate Exam Question

Show that the language $L = \{w \in \{0,1\}^* \mid w \text{ has an equal number of 0s and 1s}\}$ is not context-free using the pumping lemma.

Solution:

Proof by contradiction: assume L is context-free with pumping length p .

Consider the string $s = 0^p 1^p$.

The length of s is $|s| = 2p > p$ and $s \in L$, so the pumping lemma applies.

According to the pumping lemma, we can split s into five parts, $s = uvwxy$, such that:

- vwx has length at most p ,
- $vx \neq \lambda$,
- For all $i \geq 0$, $uv^iwx^i y \in L$.

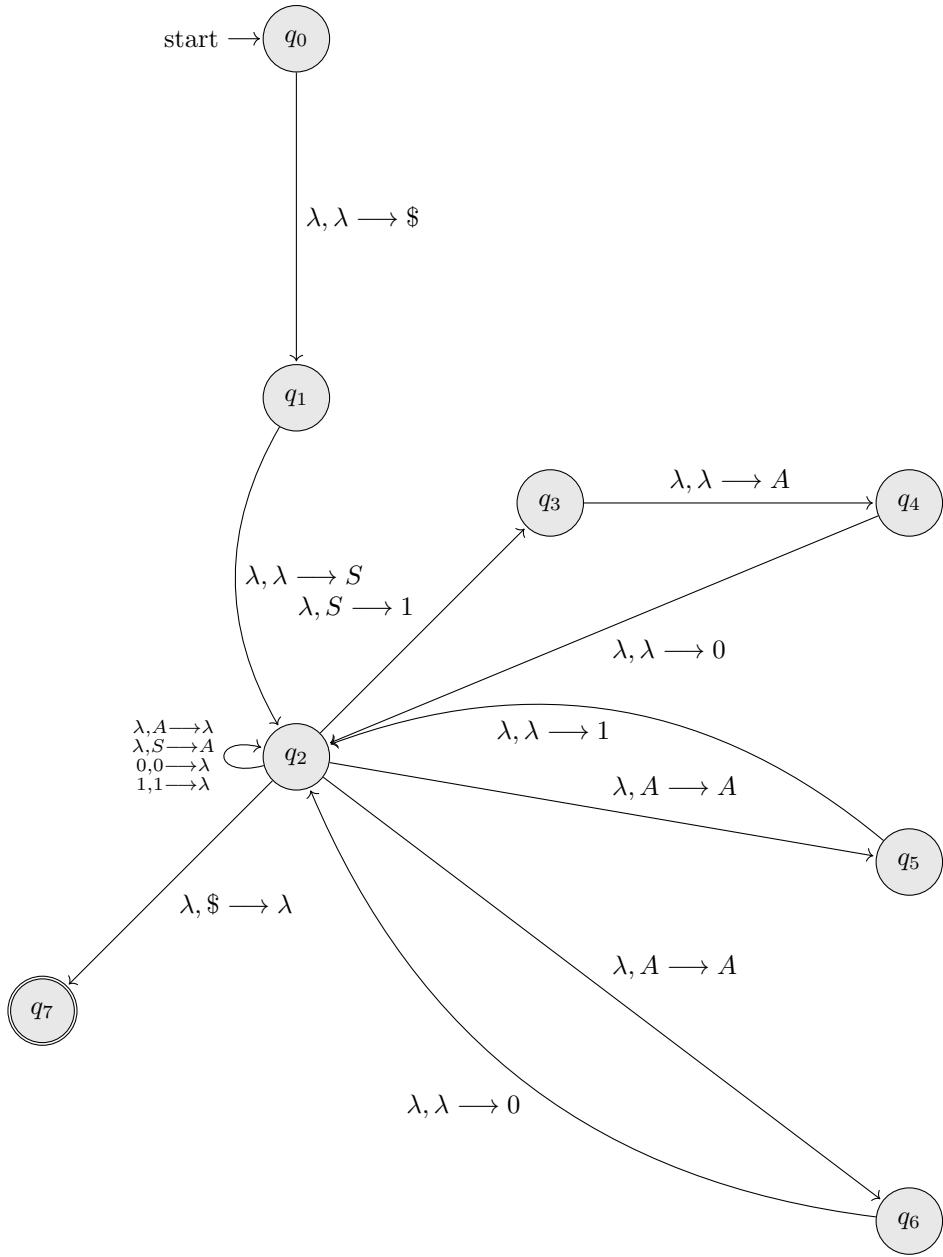
Since vwx has length at most p , it must contain only 0s or only 1s (or a mix that doesn't change the equal count).

Pumping v and x will result in strings with unequal numbers of 0s and 1s when $i \neq 1$, meaning $uv^iwx^i y \notin L$ for $i \neq 1$. Therefore, the pumping lemma is violated, proving that L is not context-free.

3.21 Q6: Candidate Exam Question

Construct a PDA that accepts the language L_1 defined by the grammar

$$\begin{array}{lcl} S & \longrightarrow & 0A1 \mid A \\ A & \longrightarrow & 1A \mid 0S \mid \lambda \end{array}$$



3.22 Q6: Candidate Exam Question

Consider the language L over the alphabet $\Sigma = \{0, 1\}$ defined as:

$$L = \{w \mid w \text{ contains at least two } 0's \text{ and exactly one } 1\}$$

- (a) Give a context-free grammar that generates L .
 - (b) Draw a pushdown automaton (PDA) that accepts L .
- (a)** Here is a context-free grammar for L :

$$S \rightarrow 0A0 \mid 00A$$

$$A \rightarrow 0A \mid 1 \mid 01 \mid 10$$

This grammar works because:

- The first rule ensures we have at least two 0's by requiring one at the start and one at the end (0A0), or two consecutive 0's (00A).
- The second rule allows us to:
 - Add more 0's if needed (0A),
 - Place the single 1 either by itself (1) or next to a 0 (01 or 10).

Examples of strings in L : "001", "010", "0010", "0100", "00100".

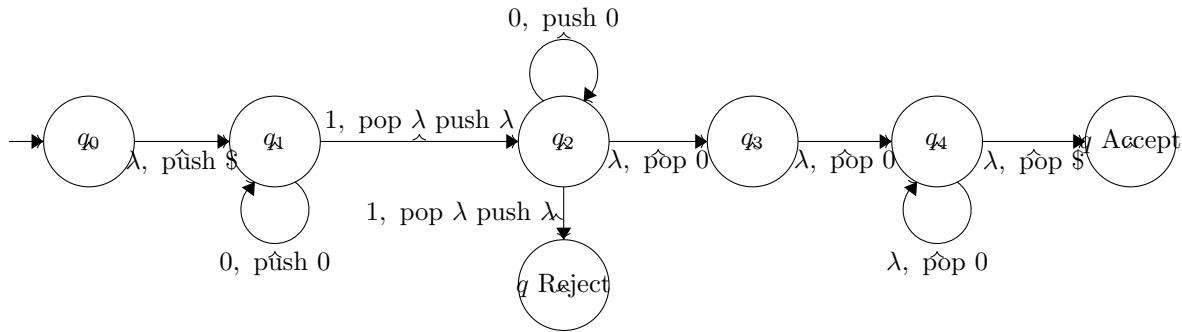
(b) A PDA that accepts L can be built as follows:

1. First, verify we see at least two 0's by counting them on the stack.
2. When we see a 1, mark it (we can use a state to remember we've seen it).
3. Reject if we see another 1.
4. Accept when we reach the end if we've seen exactly one 1 and at least two 0's.

Simple Example: For the input "0010":

- Read first 0: push it on stack.
- Read second 0: push it on stack.
- Read 1: enter "seen-one" state.
- Read 0: stay in "seen-one" state.
- Reach end: accept because we have ≥ 2 zeros (on stack) and saw exactly one 1.

Here is a state diagram to illustrate. This could also be a requirement for the question.



3.23 Q6: Candidate Exam Question

Let $\Sigma = \{0, 1\}$. Find a regular grammar that generates the language L shown below:

$$L = \{w \mid w \in \Sigma^* \text{ such that } w \text{ has an even number of 0s}\} .$$

$$\begin{array}{lcl} S & \longrightarrow & X0X0X \\ X & \longrightarrow & 0X0 \\ X & \longrightarrow & X1X \\ X & \longrightarrow & \epsilon \end{array}$$

3.24 Q6: Candidate Exam Question

Prove that the language:

$$A_{DFA} = \{\langle G \rangle \mid G \text{ is a DFA that accepted at least 3 strings}\}$$

Show that if G is decidable TM.

Answer

To help = visualize the problem. we assume $\Sigma = \{a, b\}$ and input G . We create TM M that decide A_{DFA} . For example, the w input string can be $w = \{a, b, ab\}$. w is an input that contains 3 strings. Now we turning machine M as follows:

$M =$ " On input w where G is a DFA

1. Simulate G input string w
1. if all G input w simulate reach to accepted states, accept
2. if there are less than 3 strings in w inputs reach accepted states, reject"

TM M is a decider based on the proposed turning machine because w input will halt at certain w steps. If there are greater than equal to 3 then M will accept. M will reject if there are at least three inputs or did not reach a non-accepting state. As a result, TM M is decider for A_{DFA} .

3.25 Q6: Candidate Exam Question

Draw a state diagram for the deterministic Turing Machine, M , recognizing the language $L = w$ and w ends with the substring 100, where $\Sigma = \{0, 1\}$.

Solution

M can be represented by the following state diagram:

- States:
 - q_0 (Start state, loops '0' until '1' occurs)
 - q_1 (State with first occurrence of '1')
 - q_2 (State with next occurrence of '0' after '1', returns to start state if '1' occurs again)
 - q_3 (State with next occurrence of '0', returns to start state if '0' occurs again and q_1 if '1' occurs again)
 - $q_{accepting}$ (Accepting state, string ends with '100')
- Transitions:
 - From q_0 :
 - * On '0 → R', stay in q_0
 - * On '1 → R', go to q_1
 - From q_1 :
 - * On '0 → R', go to q_2
 - * On '1 → R', stay in q_1
 - From q_2 :
 - * On '0 → R', go to q_3
 - * On '1 → R', go to q_0
 - From q_3 :
 - * On '0 → R', go to q_0
 - * On '1 → R', go to q_1

- * On ' $_ \rightarrow L$ ', go to $q_{accepting}$
- Start State: q_0
- Accepting State: $q_{accepting}$
- Q: $q_0, q_1, q_2, q_3, q_{accepting}$
- Σ : 0, 1
- Γ : 0, 1, $_$

3.26 Q6: Candidate Exam Question

Q: Find a context free grammar for the following language, then create a PDA from the grammar.

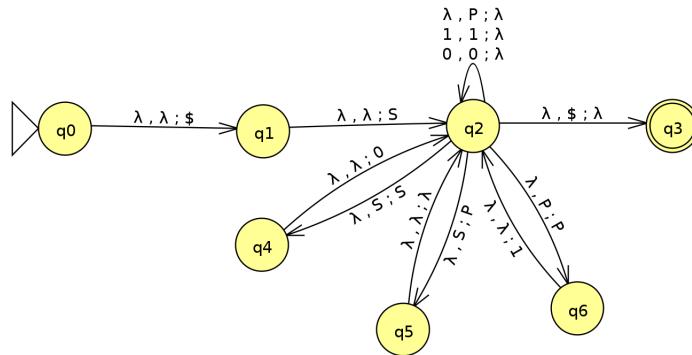
$$L = \{0^n 1^m : n, m \geq 0\}$$

Solution:

The context-free grammar is:

$$\begin{aligned} S &\rightarrow 0S \mid P \\ P &\rightarrow 1P \mid \lambda \end{aligned}$$

A PDA implementation is:



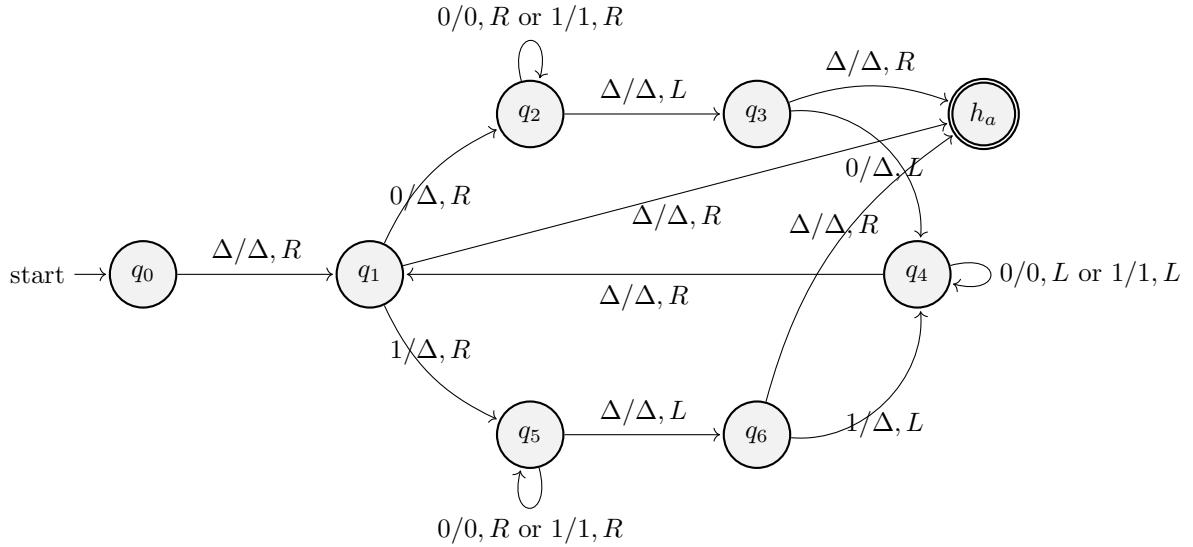
3.27 Q6: Candidate Exam Question

Let L be the language $\{w \in \{0, 1\}^* \mid w \text{ is a palindrome}\}$ over $\Sigma = \{0, 1\}$.

- Draw the state diagram for M , a deterministic Turing Machine that recognizes L . It will be assumed that all the transitions not depicted go to the reject state, which may or may not be drawn.
- Is your machine from the previous step a decider?

Solution

(i)



- (ii) As we can see from the previous part, our state diagram of our Turing Machine over the language L correctly decides on the language L . For all palindromes, our machine will accept. For all strings outside of the language, and therefore not a palindrome, our machine will reject.

3.28 Q6: Candidate Exam Question

Q: For the regular expression $(0 + 1)^*10(1)^*$, show that it is context-free

A: You need to show that there is grammar for it to be context free
 $G = (V, \Sigma, R, S)$

$V = S, A, B$

$\Sigma = 0, 1$

$R =$

$$\begin{aligned} S &\rightarrow A10B \\ A &\rightarrow 0A \mid 1A \mid \lambda \\ B &\rightarrow 1B \mid \lambda \end{aligned}$$

$S = S$

3.29 Q6: Candidate Exam Question

Refer, from the first homework question, to the language L_1 with the following grammar:

$$\begin{aligned} S &\rightarrow 0S1 \mid A \\ A &\rightarrow 1A0 \mid S \mid \lambda \end{aligned}$$

Assuming that the starting variable is S , provide the set of inputs of which the language consists.

Answer:

Let's analyze the possible inputs created by the given grammar:

- **Case 1: S production rules**

With the starting variable being S , the only rules applicable would be $S \rightarrow 0S1$ and $S \rightarrow A$. For $0S1$, we can repeatedly utilize this, giving us a forming string of:

$$0^n S 1^n$$

where $n \geq 1$. As we finish applying $S \rightarrow OS1$, we must ultimately replace S with A and, with $A \rightarrow \lambda$, get:

$$0^n 1^n, n \geq 1$$

- **Case 2: A production rules**

If we immediately apply $S \rightarrow A$, then we can repeatedly utilize A 's first production rule, giving us a forming string of:

$$1^m A 0^m$$

where $m \geq 1$. As we finish applying $1A0$, we can decide to apply $A \rightarrow \lambda$ to get:

$$1^m 0^m, m \geq 1$$

- **Case 3: Alternating between S and A rules**

At first glance, it may seem that L_1 would recognize $\{0^n 1^n \mid n \geq 1\} \cup \{1^m 0^m \mid m \geq 1\}$, but observe the production rule $A \rightarrow S$. This indicates that the alternating of 0s and 1s is also allowed. Look at the following valid products:

$$S \rightarrow A \rightarrow 1A0 \rightarrow 1S0 \rightarrow 10S10 \rightarrow 10A10 \rightarrow 101A010 \rightarrow 101010 \quad (7)$$

$$S \rightarrow 0S1 \rightarrow 0A1 \rightarrow 01A01 \rightarrow 011A001 \rightarrow 011001 \quad (8)$$

$$\begin{aligned} S &\rightarrow 0S1 \rightarrow 00S11 \rightarrow 00A11 \rightarrow 001A011 \rightarrow 0011A0011 \\ &\rightarrow 0011S0011 \rightarrow 00110S10011 \rightarrow 001100S110011 \\ &\rightarrow 001100A110011 \rightarrow 001100110011 \end{aligned} \quad (9)$$

The sets we initially thought do not satisfy these strings, even though they're valid. On the other hand, a common deduction between all of these examples revolves around their ordering. Though different, we observe that, from right to left, the string is equivalent to its complement. Even with the first two cases, if we combine the relevant steps, we will find the following string:

$$0^n 1^m 0^m 1^n \mid n, m \geq 1$$

Though reordered, we see that its complement is visible, and seeing the string reversed we find it equal:

- **Complement (replace all 0s with 1s and all 1s with 0s):**

$$1^n 0^m 1^m 0^n$$

- **Reversed (read from right to left):**

$$1^n 0^m 1^m 0^n$$

By the behavior of the production rules (with S 's rule being wrapped the opposite way of A 's rule), and the implication that there is an equal amount of 0s and 1s, we find such a unique set of inputs. With this, it is safe to say that:

$$L_1 = \{w \in \{0, 1\}^* \mid \text{its complement, } w^R = w \text{ reversed}\}$$