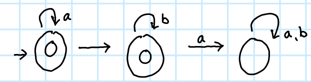
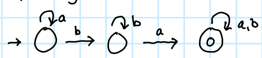
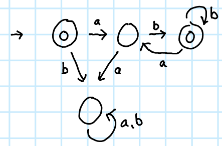
Assignment #1

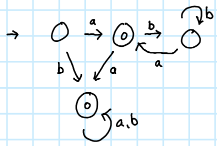
Wednesday, January 19, 2022 8:14 PM

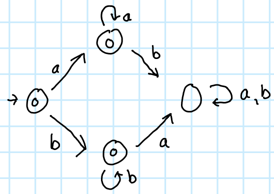
1.5.

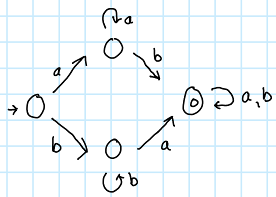
d. $L = \{w \mid w$ any string in $a^*b^*\}$ DFA of L: 

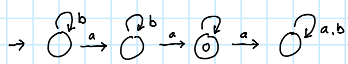
DFA of $\bar{L}$: 

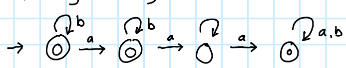
e. $L = \{w \mid w$ is any string in $(ab^+)^*\}$ DFA of L: 

DFA of $\bar{L}$ = 

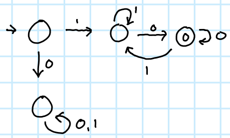
f. $L = \{w \mid w$ is any string in $a^* \cup b^*\}$ DFA of L: 

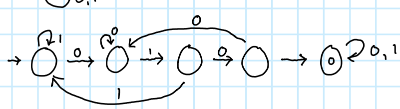
DFA of $\bar{L}$: 

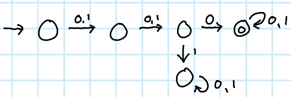
g. $L = \{w \mid w$ is any string containing exactly two a's$\}$ DFA of L: 

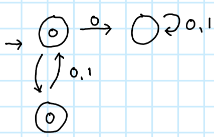
DFA of $\bar{L}$: 

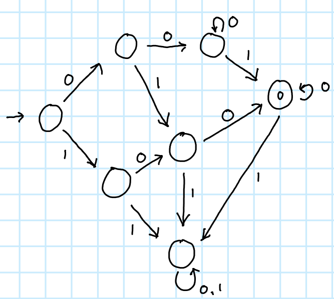
1.6

a. DFA of L: 

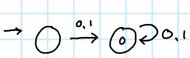
c. DFA of L: 

d. DFA of L: 

i. DFA of L: 

j. DFA of L: 

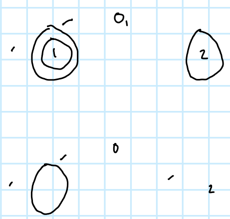
m. DFA of L: 

n. DFA of L: 

1.14

a. M' is the new DFA that has swapped accept and non-accept states of M. We say that there is some string a which M' accepts. Since M and M' have swapped accept states, this means that M does not accept string a. However, if M accepts a, then M' does not accept a. We can say then that if M' accepts a, then $a \epsilon B' (B\ complement)\ and\ a \notin B$ and vice verse if M accepts a. It can be said then that M' accepts the strings not accepted by M. Therefore, M' accepts languages that are the complement of B, i.e. B', which are also regular. As M recoignizes B, there exists M' which recognizes the complement of B. Hence, the class of regular languages is closed under complement.

b. M:



M':

We define some language C such that C = { w ∈ Σ∗ | w must end in 01}, where Σ = {0,1}. We see then that M accepts the string 001, as does M'. However, $01 \notin C' =$ { w ∈ Σ∗ | w does not end in 01}, so M' does not recognize the language C'.

The class of languages recognized by NFA's is closed under complement. We can demonstrate this using two facts:

1. Every NFA has an equivalent DFA (Theorem 1.39)
2. Swapping the accept and non-accept states of a DFA yields a new DFA that recognizes the complementary language, i.e. closed under complement, as seen in part A,

Every NFA thus has a DFA that is closed under complement. Therefore, we can say that the class of languages recognizes by NFA's is also closed under complement

1.20

a. Members
- aaaaaab
- aaaaabb

Not Members
- abababa
- bababab

b. Members
- abababab
- ababababab

Not Members
- abaaaaa
- ababbbb

c. Members
- aaaaaaa
- bbbbbbb

Not Members
- baaaaaab
- baabaab

e Members
- aaabbbaaa
- aaaabbbbaaaa

Not Members
- aaaaaaa
- bbbbbbb

f. Members (None of size ≥ 7)
- aba
- bab

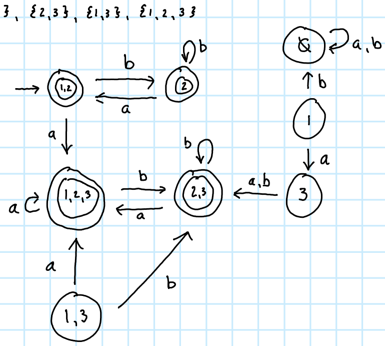
Not Members
- abbbbbb
- bbbbbaa

g. Members (None of size ≥ 7)
- b
- ab

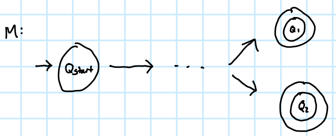
Not Members
- bbbbbbb
- aaaaaaa

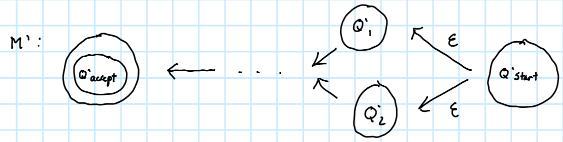
1.16 Possible States : {1}, {2}, {3}, {1,2}, {2,3}, {1,3}, {1,2,3}

State	$\mathcal{E}$	a	b
1	12	3	∅
2	2	12	2
3	3	23	23
12	12	123	2
23	23	123	23
13	123	123	23
123	123	123	123



1.31 We write that M is the DFA that recognizes A. We then write an NFA, which we'll call M' for A^R, reversing all the arrows in M and writing the start state of M (q_{start}) as the accept state of M' (q'_{accept}). We then add a new start state (q'_{start}) and from that position, add epsilon-transitions to each state of M' which are accept states in M. We can then see that for any $w\epsilon\Sigma^*$, we know that there is a path from the start state of M to the accept state of M if and only if there is a path from q'_{start} to q'_{accept} that accepts string w^R, i.e. $w\epsilon A \leftrightarrow w^R \epsilon A^R$.





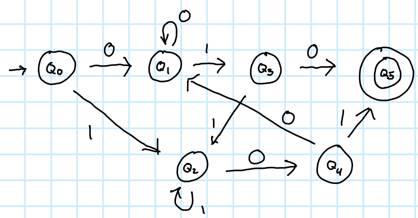
1.46

a. We define some infinite set of strings where $X = \{0^n 1^m \mid n, m \geq 0\}$. For all n, m , $0^n 1^m$ is distinguishable from all previous $0^i 1^m, 0 \leq i \leq n-1$ because there exists some z, $z = 0^n$, such that $0^n 1^m z \in B$ and $0^i 1^m z \notin B$, where $B = \{0^n 1^m 0^n \mid n, m \geq 0\}$. Since X is an infiite set, it means that B has a infinite index, thus proving it is not regular.

b. Similar to the previous problem, we can define some infinite set of strings where $X = \{0^n 1 \mid n \geq 0\}$. For all $n,$, $0^n 1$ is distinguishable from all previous $0^i 1$, *where* $0 \leq i \leq n-1$, because there exists some z, $z = 0^n$, such that $0^n 1z \notin B$ and $0^i 1z \in B$, where $B = \{w \epsilon \{0,1\}^* \mid w \text{ is not a palindrome}\}$.

1.98

a.



b. The above DFA is referred to as M_1. We will show by induction that:

a. $\delta^*_{M_1}(q_0, w) = q_0$ iff *w* is of length 0
b. $\delta^*_{M_1}(q_0, w) = q_1$ iff *w* does not contain 010 or 011, but ends with 0 and not 10
c. $\delta^*_{M_1}(q_0, w) = q_2$ iff *w* does not contain 010 or 011, but ends with 1 and not 01
d. $\delta^*_{M_1}(q_0, w) = q_3$ iff *w* does not contain 010 or 011, but ends with 01 and not 101
e. $\delta^*_{M_1}(q_0, w) = q_4$ iff *w* does not contain 010 or 011, but ends with 10 and not 010
f. $\delta^*_{M_1}(q_0, w) = q_5$ iff *w* does contain 010 or 101

Base case: If $|w| = 0$, then we know that the word does not contain either 010 or 011, so it is correctly rejected in Q_0. This confirms that condition (a) holds, and subsequently (b),(c),(d),(e),(f)

Induction Hypothesis: Assume that for any string *w* of length < i, hypotheses (a),(b),(c),(d),(e),(f) hold

Induction Step: Consider w of length *i,* where *i* > 0. Without loss of generality, *w* is of the form *va*, where $a \in \{0, 1\}$ and $v \in \{0,1\}^{i-1}$. We perform a case analysis:

- *Case* $q = q_0, a = 0$: $\delta^*_{M_1}(q_0, w = v0) = q_1$ iff $\delta^*_{M_1}(q_0, v) = \emptyset$. By induction (condition a) we have that $v = \emptyset$ and therefore $w = \emptyset \cup 0$ ends in a 0, so condition (b) is proved for when $q = q_0$
- Case $q = q_1, a = 0$: $\delta^*_{M_1}(q_0, w = v0) = q_1$ iff $\delta^*_{M_1}(q_0, v) = q_1$. By induction (condition b) we have that $v = \{0,1\}^{n-2}0$ and therefore $w = \{0,1\}^{n-2}00$ ends in a 0, so condition (b) is proved for when $q = q_1$
- *Case* $q = q_4, a = 0$: $\delta^*_{M_1}(q_0, w = v0) = q_1$ *iff* $\delta^*_{M_1}(q_0, v) = q_4$. By induction (condition e) we have that $v = \{0,1\}^{n-3}10$ and therefore $w = \{0,1\}^{n-3}100$ ends in 0, so condition (b) is proved for when $q = q_4$
- *Case* $q = q_0, a = 1$: $\delta^*_{M_1}(q_0, w = v1) = q_2$ *iff* $\delta^*_{M_1}(q_0, v) = q_0$. By induction (condition a) we have that $v = \emptyset$ and therefore $w = \emptyset \cup 1$ ends in a 1, so condition (c) is proved for when $q = q_0$
- *Case* $q = q_2, a = 1$: $\delta^*_{M_1}(q_0, w = v0) = q_2$ *iff* $\delta^*_{M_1}(q_0, v) = q_2$. By induction (condition c) we have that $v = \{0,1\}^{n-2}1$ and therefore $w = \{0,1\}^{n-2}11$ ends in a 1, so condition (c) is proved for when $q = q_2$
- *Case* $q = q_3, a = 1$: $\delta^*_{M_1}(q_0, v0) = q_2$ *iff* $\delta^*_{M_1}(q_0, v) = q_3$. By induction (condition d) we have that $v = \{0,1\}^{n-3}01$ and therefore $w = \{0,1\}^{n-3}011$ ends in a 1, so condition (b) is proved for when $q = q_3$
- *Case* $q = q_1, a = 1$: $\delta^*_{M_1}(q_0, v1) = q_3$ *iff* $\delta^*_{M_1}(q_0, v) = q_1$. By induction (condition b) we have that $v = \{0,1\}^{n-2}0$ and therefore $w = \{0,1\}^{n-2}01$ ends in a 01, so condition (d) is proved for when $q = q_1$
- *Case* $q = q_2, a = 0$: $\delta^*_{M_1}(q_0, v0) = q_4$ *iff* $\delta^*_{M_1}(q_0, v) = q_2$. By induction (condition c) we have that $v = \{0,1\}^{n-2}1$ and therefore $w = \{0,1\}^{n-2}10$ ends in a 10, so condition (e) is proved for when $q = q_2$
- *Case* $q = q_3, a = 0$: $\delta^*_{M_1}(q_0, v0) = q_5$ *iff* $\delta^*_{M_1}(q_0, v) = q_3$. By induction (condition d) we have that $v = \{0,1\}^{n-3}01$ and therefore $w = \{0,1\}^{n-3}010$ ends in a 010, so condition (f) is proved for when *q*

= q_3

- Case q = q_4, $a = 0$: $\delta^*_{M_1}(q_0, v1) = q_5$ iff $\delta^*_{M_1}(q_0, v) = q_4$. By induction (condition e) we have that $v = \{0,1\}^{n-3}10$ and therefore $w = \{0,1\}^{n-3}101$ ends in a 101, so condition (f) is proved for when q = q_4

1.19

a.

1. 54 bits, 17 even, 16 odd. 16-17 = -1. -1 (mod 3) ≠ 0, so this number is not divisible by 3.
2. 54 bits, 25 even, 24 odd. 24-25 = -1. -1 (mod 3) ≠ 0, so this number is not divisible by 3.
3. 54 bits, 4 even, 4 odd. 4-5 = -1. -1 (mod 3) ≠ 0, so this number is not divisible by 3.
4. 54 bits, 14 even, 14 odd. 14-14 = 0. 0 (mod 3) = 0, so this number is divisible by 3.

b. Each binary number can be represented in decimal form through the equation $2^n * a_n + 2^{n-1} * a_{n-1} + \ldots + 2^0 * a_0$, where the binary number is composed of the numbers $a_n a_{n-1} \ldots a_0$. We know by the symmetry of modulus operation that $a = b(mod\ n)\ if\ b = a(mod\ n)$ for all *a,b,* and *n*. We can then say that $2 = -1\ (mod\ 3)\ and\ -1 = 2\ (mod\ 3)$. So we can then apply the equivalence relation to say that:

- $a_1 + a_2 = b_1 + b_2\ (mod\ n)$
- $ka = kb\ (mod\ n)$ for any integer *k*
- $a^k = b^k\ (mod\ n)$ for any non-negative integer *k*

So, the binary to decimal number conversion can be rewritten as $(-1)^n * a_n + (-1)^{n-1} * a_{n-1} + \ldots + (-1)^0 * a_0$, when our binary string is $w = w_n w_{n-1} \ldots\ w_1 = a_n a_{n-1} \ldots\ a_1$, which will have the same divisibility that the decimal representation the binary string had for 3. This is equivalent to $\sum_{k=1}^{n}(-1)^k w_k$ (mod 3) , or, alternatively, adding the values of the even indices' bits and subtracting the odd indices' bits ($even + (-odd)$). This view of the alternating sum is direction-agnostic, as it means you could either read the binary string from right to left, or left to right, i.e. $w = w_1 \ldots w_{n-1}w_n$. This can be proven as follows: If the binary string has an odd number of characters/bits, then all of the odd bits remain in odd indices and all even bits remain in even indices as the "flip" pivots around some center odd indice. If the string has an even number of characters/bits, then the odd indice sum and even indice sum switch value, since all even indice bits become odd and all odd indice bits become even, or

$$even\_bits - odds\_bits = -(odd\_bits - even\_bits)$$

This negative sum ($-(odd\_bits - even\_bits)$) has the same modulo value as the positive sum ($even\_bits - odds\_bits$), meaning they share divisibility for 3.

For example, we will take the odd charactered string of 30 in binary: 11110

- From left to right, we have summed values of 2 odd and 2 even, meaning our total sum is $2 - 2 = 0$, which is divisible by 3.
- From right to left, we have summed values of 2 odd and 2 even, meaning our total sum is $2 - 2 = 0$, which is divisible by 3

We will now take an even numbered example of 3333 in binary: 110100000101

- From left to right, we have summed values of 1 odd and 4 even, meaning our total sum is $4 - 1 = 3$, which is divisible by 3.
- From right to left, we have summed values of 4 odd and 1 even, meaning our total sum is $1 - 4 = -3$, which is divisible by 3

Thus, we have proven that if $\sum_{k=1}^{n}(-1)^k w_k$ is a multiple of three, then both $w = w_n w_{n-1} \ldots\ w_1$ and $w = w_1 \ldots w_{n-1} w_n$ are multiples of three, therefore, $L_1 = L_2 = L_3$

c. The automaton seen in class is also capable of determining from a binary string if the number it represents is divisible by 3. We can demonstrate this with our string of 30 (11110) from the previous example. The flow through the automaton would be as follows: $\rightarrow q_0 \vec{1}\ q_1\ \vec{1}\ q_0 \vec{1}\ q_1 \vec{1}\ q_0 \vec{0}\ q_0$, bringing us back to the accept state of q_0, proving it is divisble by 3, as we know.