Assignment 3

Kewei Qiu

 # 1003798116

Tianyi Long

 # 1002902889

Question 1

To prove term(x) terminates we first need to find a loop invariant

Define $\mathrm{P}(k)$: after the k th of the loop (if occurs), We have $y_k = x_k^3,\ x_k = x - k$.

WTS $k \in \mathbb{N},\ \mathrm{P}(k)$

Proof: (Simple Induction)

Base Case: $k = 0$

This means that the loop has not been executed and only the line out of loop is executed

Then by that initialization line, $y_0 = x_0^3,\ x_0 = x = x - k$

$P(0)$ holds

Inductive Step: Let $k \in \mathbb{N},\ k > 0$. Assume $P(k)$ holds

We want to show that $P(k+1)$ follows

$y_{k+1} = y_k - 3x_{k+1}^2 - 3x_{k+1} - 1$ (By code in the loop)

$= x_k^3 - 3x_{k+1}^2 - 3x_{k+1} - 1$ (By (k) $y_k = x_k^3$)

$= (x_{k+1} + 1)^3 - 3x_{k+1}^2 - 3x_{k+1} - 1$ (Since $x_{k+1} = x_k - 1$ by code)

$= x_{k+1}^3 + 3x_{k+1}^2 + 3x_{k+1} + 1 - 3x_{k+1}^2 - 3x_{k+1} - 1$

$= x_{k+1}^3$

$x_{k+1} = x_k - 1$ (By code in the loop)

$= x - k - 1$ (By (k) $x_k = x - k$)

$= x - (k+1)$

Hence $P(k+1)$ follows.

By simple induction, we proved that $\forall k \in \mathbb{N}$, after the k th of the loop (if occurs), we have $y_k = x_k^3,\ x_k = x - k$, which is the loop invariant. ■

Then we prove termination using loop invariant

Proof: By loop invariant we have $x_k = x - k$

Since $k \in \mathbb{N}$ and k is strictly increasing

We know that x_k is strictly decreasing

Also since $\mathrm{x} \in \mathbb{N}$, we have exhibited a decreasing sequence of natural numbers linked to loop iterations.

By principle of well ordering, the set of $\mathrm{x_k}$ has a smallest element, which has the index of the last loop iteration

Hence the loop terminates ■

Question 2

(a)

Let $M_a = \{Q, \Sigma, \delta, q_0, F\}$

And $A = \{a^k \middle| k \in \mathbb{N}\}$,

$D_A = \{x \in \Sigma | \; x\ has\ at\ least\ one\ b\}$

$\{Q = \{A, D_A\}$,

$\Sigma = \{\mathrm{a}, \mathrm{b}\}$

$\delta =$

δ	A	D_A
a	A	D_A
b	D_A	D_A

$q_0 = A$,

$F = \{A\}\}$

Define the smallest set Σ^* such that:

(a) $\varepsilon \in \Sigma^*$

(b) $s \in \Sigma^*, sa \in \Sigma^*\ and\ sb \in \Sigma^*$

Prove $M_a\ accepts\ L_a$

Define $\mathrm{P}(s)$: $\delta^*(S_1, s) = \begin{cases} A & , if\ s = a^k, k \in \mathbb{N} \\ D_A, & if\ s\ has\ at\ least\ one\ b \end{cases}$, WTS $\forall s \in \Sigma^*, P(s)$

Proof: (Structural Induction)

Base Case: $s = \varepsilon$

The string has 0 a, we have $\delta^*(A, \varepsilon) = A$, so the implication in the first line of invariant is true.

The string also has 0 b, so the implication in the second line of invariant is vacuously true.

$P(\varepsilon)$ holds.

Inductive Step: Let $s \in \Sigma^*$, assume $P(s)$ holds.

WTS $P(sa), P(sb)$ follow

There are two cases to consider

Case sa:

$$\delta^*(A, sa) = \delta(\delta^*(A, s), a) = \begin{cases} \delta(A, a) & , if\ s = a^k, k \in \mathbb{N} \\ \delta(D_A, a), & if\ s\ has\ at\ least\ one\ b \end{cases} \text{ (By IH } P(s))$$

$$= \begin{cases} A & , if\ s = a^k, k \in \mathbb{N} \\ D_A, & if\ s\ has\ at\ least\ one\ b \end{cases} \text{ (one more a)}$$

Case 2: $P(sb)$

$$\delta^*(A, sb) = \delta(\delta^*(A, s), b) = \begin{cases} \delta(A, b) & , if\ s = a^* \\ \delta(D_A, b), & if\ s\ has\ at\ least\ one\ b \end{cases} \text{ (By IH } P(s))$$

$$= \begin{cases} D_A & , if\ s = a^* \\ D_A, & if\ s\ has\ at\ least\ one\ b \end{cases} \text{ (add one b)}$$

So $P(sa), P(sb)$ follow

The first line of the invariant ensures that all strings with only as are accepted. The contrapositive of the second line of the invariant ensures that any string that does not drive the machine to state A has at least one b, in other words all strings that drive the machine to state D_A have at least one b.

So $M_a\ accepts\ L_a$. ∎

(b)

Let $M_b = \{Q, \Sigma, \delta, q_0, F\}$

And $B = \{b^j \mid j \in \mathbb{N}\}$,

$D_B = \{x \in \Sigma \mid x \text{ has at least one } a\}$

$\{Q = \{B, D_B\}$,

$\Sigma = \{\mathrm{a}, \mathrm{b}\}$

$\delta =$

δ	B	D_B
a	D_B	D_B
b	B	D_B

$q_0 = B$

$F = \{B\}\}$

Define the smallest set Σ^* such that:

(a) $\varepsilon \in \Sigma^*$

(b) $s \in \Sigma^*, sa \in \Sigma^* \text{ and } sb \in \Sigma^*$

Prove $M_a \text{ accepts } L_a$

Define $\mathrm{P}(s)$: $\delta^*(S_1, s) = \begin{cases} B & , \text{if } s = b^j, j \in \mathbb{N} \\ D_B, & \text{if } s \text{ has at least one } a \end{cases}$, WTS $\forall s \in \Sigma^*, P(s)$

Proof: (Structural Induction)

Base Case: $s = \varepsilon$

The string has 0 b, we have $\delta^*(B, \varepsilon) = B$, so the implication in the first line of invariant is true.

The string also has 0 a, so the implication in the second line of invariant is vacuously true.

$P(\varepsilon)$ holds.

Inductive Step: Let $s \in \Sigma^*$, assume $P(s)$ holds.

WTS $P(sa), P(sb)$ follow

There are two cases to consider

Case sb:

$$\delta^*(B,sb)=\delta(\delta^*(B,s),b)=\begin{cases}\delta(B,b) & , if\ s=b^j, j\in\mathbb{N}\\ \delta(D_B,b), & if\ s\ has\ at\ least\ one\ a\end{cases} \text{ (By IH } P(s))$$

$$=\begin{cases}B & , if\ s=b^j, j\in\mathbb{N}\\ D_B, & if\ s\ has\ at\ least\ one\ a\end{cases} \text{ (one more b)}$$

Case sa:

$$\delta^*(B,sa)=\delta(\delta^*(B,s),a)=\begin{cases}\delta(B,a) & , if\ s=b^j, j\in\mathbb{N}\\ \delta(D_B,a), & if\ s\ has\ at\ least\ one\ a\end{cases} \text{ (By IH } P(s))$$

$$=\begin{cases}D_B & , if\ s=b^j, j\in\mathbb{N}\\ D_B, & if\ s\ has\ at\ least\ one\ a\end{cases} \text{ (add one a)}$$

So $P(sa), P(sb)$ follow

The first line of the invariant ensures that all strings with only bs are accepted. The contrapositive of the second line of the invariant ensures that any string that does not drive the machine to state B has at least one a, in other words all strings that drive the machine to state D_B have at least one a.

So $M_b\ accepts\ L_b$. ∎

(c)

Let $M_2=\{Q,\Sigma,\delta,q_0,F\}$

And $E=\{x\in\Sigma|\ |x|\ is\ even\}$, $O=\{x\in\Sigma|\ |x|\ is\ odd\}$

$\{Q=\{E,O\}$,

$\Sigma=\{\mathrm{a},\mathrm{b}\}$

$\delta=$

δ	E	O
a	O	E
b	O	E

$q_0=E$,

$F=\{E\}\}$

Define the smallest set Σ^* such that:

(a) $\varepsilon\in\Sigma^*$

(b) $s\in\Sigma^*, sa\in\Sigma^*\ and\ sb\in\Sigma^*$

Prove $M_2\ accepts\ L_2$

Define $\mathrm{P}(s)$: $\delta^*(E,s) = \begin{cases} O\ \ ,if\ |s|\ is\ odd \\ E, if\ |s|\ is\ even \end{cases}$, WTS $\forall s \in \Sigma^*, P(s)$

Proof: (Structural Induction)

Base Case: $s = \varepsilon$

The string has length 0, and 0 is an even number

We have $\delta^*(E,\varepsilon) = E$, so the implication in the first line of invariant is true.

The string length 0, and 0 is an even number

So the implication in the second line of invariant is vacuously true.

$P(\varepsilon)$ holds.

Inductive Step: Let $s \in \Sigma^*$, assume $P(s)$ holds.

WTS $P(sa), P(sb)$ follow

There are two cases to consider

Case sa:

$$\delta^*(E,sa) = \delta(\delta^*(E,s),a) = \begin{cases} \delta(E,a)\ ,\ if\ |s|\ is\ even \\ \delta(O,a)\ ,\ if\ |s|\ is\ odd \end{cases} \text{ (By IH } P(s)\text{)}$$

$$= \begin{cases} O,\ if\ |s|\ is\ even \\ E\ ,\ if\ |s|\ is\ odd \end{cases} \text{ (one more element)}$$

Case sb:

$$\delta^*(E,sb) = \delta(\delta^*(E,s),b) = \begin{cases} \delta(E,b)\ ,\ if\ |s|\ is\ even \\ \delta(O,b),\ \ if\ |s|\ is\ odd \end{cases} \text{ (By IH } P(s)\text{)}$$

$$= \begin{cases} O\ ,\ if\ |s|\ is\ even \\ E\ ,\ if\ |s|\ is\ odd \end{cases} \text{ (one more element)}$$

So $P(sa), P(sb)$ follow

The first line of the invariant ensures that all strings with even length accepted. The contrapositive of the second line of the invariant ensures that any string that does not drive the machine to state A does not have even length, in other words all strings that drive the machine to state D_A have odd length.

So $M_2\ accepts\ L_2$. ∎

(d)

Let $M_{a|b} = \{Q, \Sigma, \delta, q_0, F\}$

$\{Q = \{(A,B), (A,D_B), (D_A,B), (D_A,D_B)\},$

$\Sigma = \{\mathrm{a}, \mathrm{b}\}$

$\delta =$

δ	(A,B)	(A,D_B)	(D_A,B)	(D_A,D_B)
a	(A,D_B)	(D_A,D_B)	(A,D_B)	(D_A,D_B)
b	(D_A,B)	(D_A,B)	(D_A,D_B)	(D_A,D_B)

$q_0 = (A,B),$

$F = \{(A,B), (A,D_B), (D_A,B)\}\}$

To show $M_{a|b}$ accepts $L_a \cup L_b$:

Denote the states for M_a as Q_a, the states for M_b as Q_b, their respective transition functions as δ_a and δ_b, and the transition function for $M_{a|b}$ as $\delta_{a|b}$.

Inspection of $\delta_{a|b}$ shows that if $(\mathrm{q_a}, \mathrm{q_b}, \mathrm{c}) \in Q_a \times Q_b \times \Sigma^*$,

then $\delta_{a|b}\big((\mathrm{q_a}, \mathrm{q_b}), \mathrm{c}\big) = \delta_a(\mathrm{q_a}, c), \delta_b(\mathrm{q_b}, c)$.

Thus the following invariant follows by simply taking conjunctions of the invariants of the component machines, for any $s \in \Sigma^*$

$$P(s)\colon \delta^*((A,B),s) = \begin{cases} (A,B) & \text{, if s has 0 } a \text{ and 0 } b \\ (A,D_B) & \text{, if } s \text{ has only } b \\ (D_A,B) & \text{, if } s \text{ has only } a \\ (D_A,D_B) & \text{, if } s \text{ has both } a \text{ and } b \end{cases}$$

The implication on the first line ensures that all strings with 0 a and 0 b end up in state (A,B).

The implication on the second line ensures that all strings with only b end up in state (A,D_B).

The implication on the third line ensures that all strings with only a end up in state (D_A,B).

The contrapositive of the implications on the forth line ensure that any string that does not drive the machines to one of those 3 states must have both a and b.

Hence $M_{a|b}$ accepts $L_a \cup L_b$

(e)

Let $M_{a|b\ even} = \{Q, \Sigma, \delta, q_0, F\}$

$\{Q = \{(E, q_a), (E, q_b), (E, q_n), (E, q_m), (O, q_a), (O, q_b), (O, q_n), (O, q_m)\},$

$\Sigma = \{\mathrm{a}, \mathrm{b}\}$

$\delta =$

δ	$((A,B),E)$	$((A,D_B),E)$	$((D_A,B),E)$	$((D_A,D_B),E)$	$((A,B),O)$	$((A,D_B),O)$	$((D_A,B),O)$	$((D_A,D_B),O)$
a	$((A,D_B),O)$	$((A,D_B),O)$	$((D_A,D_B),O)$	$((D_A,D_B),O)$	$((A,D_B),E)$	$((A,D_B),E)$	$((D_A,D_B),E)$	$((D_A,D_B),E)$
b	$((D_A,B),O)$	$((D_A,D_B),O)$	$((D_A,B),O)$	$((D_A,D_B),O)$	$((D_A,B),E)$	$((D_A,D_B),E)$	$((D_A,B),E)$	$((D_A,D_B),E)$

$q_0 = ((A,B),E),$

$F = \{\big((A,B),E\big), \big((A,D_B),E\big), \big((D_A,B),E\big)\}\}$

To show $M_{a|b\ even}$ accepts $(L_a \cup L_b) \cap L_2$:

Denote the states for $M_{a|b}$ as Q_1, the states for M_2 as Q_2, their respective transition functions as δ_1 and δ_2, and the transition function for $M_{a|b\ even}$ as δ_3.

Inspection of δ_3 shows that if $(\mathrm{q_1}, \mathrm{q_2}, \mathrm{c}) \in Q_1 \times Q_2 \times \Sigma^*$,

Then $\delta_{1|2}\big((\mathrm{q_1}, \mathrm{q_2}), \mathrm{c}\big) = \delta_1(\mathrm{q_1}, c), \delta_2(\mathrm{q_2}, c)$.

Thus the following invariant follows by simply taking conjunctions of the invariants of the component machines, for any $s \in \Sigma^*$

$$P(s)\colon \delta^*((A,B),s) = \begin{cases} \big((A,B),E\big) & \text{, if s has 0 a and 0 b, and |s| is even} \\ \big((A,D_B),E\big) & \text{, if s has only b, and |s| is even} \\ \big((D_A,B),E\big) & \text{, if s has only a, and |s| is even} \\ \big((D_A,D_B),E\big) & \text{, if s has both a and b, and |s| is even} \\ \big((A,B),O\big) & \text{, if s has 0 a and 0 b, and |s| is odd} \\ \big((A,D_B),O\big) & \text{, if s has only b, and |s| is odd} \\ \big((D_A,B),O\big) & \text{, if s has only a, and |s| is odd} \\ \big((D_A,D_B),O\big) & \text{, if s has both a and b, and |s| is odd} \end{cases}$$

The implication on the first line ensures that all strings with 0 a and 0 b, and even length end up in state $\big((A,B),E\big)$.

The implication on the first line ensures that all strings with only b and even length end up in state $\big((A,D_B),E\big)$.

The implication on the first line ensures that all strings with only a and even length end up in state $\big((D_A,B),E\big)$.

The contrapositive of the implications on the other 5 lines ensure that any string that does not drive the machines to one of those 3 states must have both a and b or odd length

Hence $M_{a|b\ even}$ accepts $(L_a \cup L_b) \cap L_2$

Question 3

(a) construct machines

(1) Let $M_0 = \{Q, \Sigma, \delta, S_0, F\}$

And:

$S_0 = \{x \in \Sigma |\ x\ represents\ a\ number\ equivalent\ to\ 0\ mod\ 3\ in\ base\ 10\ \}$

$S_1 = \{x \in \Sigma |\ x\ represents\ a\ number\ equivalent\ to\ 1\ mod\ 3\ in\ base\ 10\ \}$

$S_2 = \{x \in \Sigma |\ x\ represents\ a\ number\ equivalent\ to\ 2\ mod\ 3\ in\ base\ 10\ \}$

$\{Q = \{S_0, S_1, S_2\},$

$\Sigma = \{\varepsilon, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$

$\delta =$

δ	S_0	S_1	S_2
ε	S_0	S_1	S_2
0	S_0	S_1	S_2
1	S_1	S_2	S_0
2	S_2	S_1	S_0
3	S_0	S_1	S_2
4	S_1	S_2	S_0
5	S_2	S_1	S_0
6	S_0	S_1	S_2
7	S_1	S_2	S_0
8	S_2	S_1	S_0
9	S_0	S_1	S_2

$q_0 = S_0,$

$F = \{S_0\}\}$

(2) Let $M_1 = \{Q, \Sigma, \delta, S_0, F\}$

And:

$S_0 = \{x \in \Sigma |\ x\ represents\ a\ number\ equivalent\ to\ 0\ mod\ 3\ in\ base\ 10\ \}$

$S_1 = \{x \in \Sigma |\ x\ represents\ a\ number\ equivalent\ to\ 1\ mod\ 3\ in\ base\ 10\ \}$

$S_2 = \{x \in \Sigma |\ x\ represents\ a\ number\ equivalent\ to\ 2\ mod\ 3\ in\ base\ 10\ \}$

$\{Q = \{S_0, S_1, S\},$

$\Sigma = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$

$\delta =$

δ	S_0	S_1	S_2
0	S_0	S_1	S_2
1	S_1	S_2	S_0
2	S_2	S_1	S_0
3	S_0	S_1	S_2
4	S_1	S_2	S_0
5	S_2	S_1	S_0
6	S_0	S_1	S_2
7	S_1	S_2	S_0
8	S_2	S_1	S_0
9	S_0	S_1	S_2

$q_0 = S_0,$

$F = \{S_1\}\}$

(3) Let $M_2 = \{Q, \Sigma, \delta, S_0, F\}$

And:

$S_0 = \{x \in \Sigma |\ x\ represents\ a\ number\ equivalent\ to\ 0\ mod\ 3\ in\ base\ 10\ \}$

$S_1 = \{x \in \Sigma |\ x\ represents\ a\ number\ equivalent\ to\ 1\ mod\ 3\ in\ base\ 10\ \}$

$S_2 = \{x \in \Sigma |\ x\ represents\ a\ number\ equivalent\ to\ 2\ mod\ 3\ in\ base\ 10\ \}$

$\{Q = \{S_0, S_1, S_2\},$

$\Sigma = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$

$\delta =$

δ	S_0	S_1	S_2
0	S_0	S_1	S_2
1	S_1	S_2	S_0
2	S_2	S_1	S_0
3	S_0	S_1	S_2
4	S_1	S_2	S_0
5	S_2	S_1	S_0

6	S_0	S_1	S_2
7	S_1	S_2	S_0
8	S_2	S_1	S_0
9	S_0	S_1	S_2

$q_0 = S_0,$

$F = \{S_2\}\}$

(b) explain equality

(1) Let $R_0 = Rev(M_0) = \{Q, \Sigma, \delta, q_0, F\}$

And:

$S_0 = \{x \in \Sigma |\ x\ represents\ a\ number\ equivalent\ to\ 0\ mod\ 3\ in\ base\ 10\ \}$

$S_1 = \{x \in \Sigma |\ x\ represents\ a\ number\ equivalent\ to\ 1\ mod\ 3\ in\ base\ 10\ \}$

$S_2 = \{x \in \Sigma |\ x\ represents\ a\ number\ equivalent\ to\ 2\ mod\ 3\ in\ base\ 10\ \}$

$\{Q = \{S_0, S_1, S_2\},$

$\Sigma = \{\varepsilon, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$

$\delta =$

δ	S_0	S_1	S_2
ε	S_0	S_1	S_2
0	S_0	S_1	S_2
1	S_2	S_1	S_0
2	S_1	S_2	S_0
3	S_0	S_1	S_2
4	S_2	S_1	S_0
5	S_1	S_2	S_0
6	S_0	S_1	S_2
7	S_2	S_1	S_0
8	S_1	S_2	S_0
9	S_0	S_1	S_2

$q_0 = S_0,$

$F = \{S_0\}\}$

When creating a machine accepting $R_0 = Rev(M_0)$, we find that it is the same as M_0 itself:

① Both of two machines have the same 3 states

② Both of two machines the same start and accept state

③ Giving $\varepsilon, 0, 3, 6, 9$ to any state will let the machine hold its current state

④ Giving $1, 4, 7$ to any state will let the machine jump to the next state and this creates a loop of states with no duplicate and no omit

⑤ Giving $2, 5, 8$ to any state will let the machine jump to the previous state and this creates a loop of states with no duplicate and no omit

By this we may find out that R_0 $accepting$ L_0 and M_0 $accepting$ $Rev(L_0)$ are aexctly the same

Hence we can conclude that $L_0 = Rev(L_0)$

(2) Let $R_1 = Rev(M_1) = \{Q, \Sigma, \delta, q_0, F\}$

And:

$$S_0 = \{x \in \Sigma |\ x\ represents\ a\ number\ equivalent\ to\ 0\ mod\ 3\ in\ base\ 10\ \}$$

$$S_1 = \{x \in \Sigma |\ x\ represents\ a\ number\ equivalent\ to\ 1\ mod\ 3\ in\ base\ 10\ \}$$

$$S_2 = \{x \in \Sigma |\ x\ represents\ a\ number\ equivalent\ to\ 2\ mod\ 3\ in\ base\ 10\ \}$$

$\{Q = \{S_0, S_1, S_2\},$

$\Sigma = \{\varepsilon, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$

$\delta =$

δ	S_0	S_1	S_2
ε	S_0	S_1	S_2
0	S_0	S_1	S_2
1	S_2	S_1	S_0
2	S_1	S_2	S_0
3	S_0	S_1	S_2
4	S_2	S_1	S_0
5	S_1	S_2	S_0
6	S_0	S_1	S_2
7	S_2	S_1	S_0
8	S_1	S_2	S_0
9	S_0	S_1	S_2

$q_0 = S_1,$

$F = \{S_0\}\}$

When creating a machine accepting $R_0 = Rev(M_0)$, we find that it is the same as M_0 itself:

① Both of two machines have the same 3 states

② Both of two machines have one start and one accepting state and they are different

③ Giving $\varepsilon, 0, 3, 6, 9$ to any state will let the machine hold its current state

④ Giving $1, 4, 7$ to any state will let the machine jump to the next state and this creates a loop of states with no duplicate and no omit

⑤ Giving $2, 5, 8$ to any state will let the machine jump to the previous state and this creates a loop of states with no duplicate and no omit

By this we may find out that $R_1\ accepting\ L_1\ and\ M_1\ accepting\ Rev(L_1)$ are aexctly the same

Hence we can conclude that $L_1 = Rev(L_1)$

(3) Let $R_2 = Rev(M_2) = \{Q, \Sigma, \delta, q_0, F\}$

And:

$$S_0 = \{x \in \Sigma |\ x\ represents\ a\ number\ equivalent\ to\ 0\ mod\ 3\ in\ base\ 10\ \}$$

$$S_1 = \{x \in \Sigma |\ x\ represents\ a\ number\ equivalent\ to\ 1\ mod\ 3\ in\ base\ 10\ \}$$

$$S_2 = \{x \in \Sigma |\ x\ represents\ a\ number\ equivalent\ to\ 2\ mod\ 3\ in\ base\ 10\ \}$$

$\{Q = \{S_0, S_1, S_2\},$

$\Sigma = \{\varepsilon, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$

$\delta =$

δ	S_0	S_1	S_2
ε	S_0	S_1	S_2
0	S_0	S_1	S_2
1	S_2	S_1	S_0
2	S_1	S_2	S_0
3	S_0	S_1	S_2
4	S_2	S_1	S_0
5	S_1	S_2	S_0
6	S_0	S_1	S_2
7	S_2	S_1	S_0
8	S_1	S_2	S_0
9	S_0	S_1	S_2

$q_0 = S_2,$

$F = \{S_0\}\}$

When creating a machine accepting $R_0 = Rev(M_0)$, we find that it is the same as M_0 itself:

① Both of two machines have the same 3 states

② Both of two machines have one start and one accepting state and they are different

③ Giving $\varepsilon, 0, 3, 6, 9$ to any state will let the machine hold its current state

④ Giving $1, 4, 7$ to any state will let the machine jump to the next state and this creates a loop of states with no duplicate and no omit

⑤ Giving $2, 5, 8$ to any state will let the machine jump to the previous state and this creates a loop of states with no duplicate and no omit

By this we may find out that $R_2\ accepting\ L_2\ and\ M_2\ accepting\ Rev(L_2)$ are aexctly the same

Hence we can conclude that $L_2 = Rev(L_2)$

Question 4

(a)

Let $r \in RE,\ r' = r^R$, then $r' \in RE.$

Define $P(r)\colon \exists r' RE\ s.t. Rev\big(L(r)\big) = L(r')$

Want to show: $\forall r \in RE, P(r)$

Proof: (Structural Induction)

Base Case: Case 1: $r = \emptyset,\ r' = r^R = \emptyset.$ Then $Rev\big(L(r)\big) = Rev(\{\,\}) = \{\,\} = L(r')$

Case 2: $r = \varepsilon,\ r' = r^R = \varepsilon.$ Then $Rev\big(L(r)\big) = Rev(\{\varepsilon\}) = \{\varepsilon\} = L(r')$

Case 3: $r = a,\ a \in \Sigma^*,\ r' = r^R = a.$

Then $Rev\big(L(r)\big) = Rev(\{a\}) = \{a\} = L(r')$

So $P(\emptyset), P(\varepsilon), P(a)$ hold.

Inductive Step: Let $\forall r \in RE$, assume $P(r)$

i.e. $\forall k < n, k \in \mathbb{N}, |r| = k\ and\ Rev\big(L(r)\big) = L\big(r'\big).$

Want to show: $n \in \mathbb{N}, |r| = n\ and\ Rev\big(L(r)\big) = L\big(r'\big)$

There are 3 cases to consider

Case 1: $r = r_1^*$ for some r_1 be regular expression over $\Sigma = \{0, 1\}$.

And $r' = r^R = (r_1^R)^*$

Then $Rev\big(L(r)\big) = Rev\big(L(r_1^*)\big)$

$= Rev\left(Rev\big(L((r_1^R)^*)\big)\right)$ (by IH)

$$= Rev\left(Rev\left(L(r')\right)\right)$$

$$= L(r')$$

Case 2: $r = r_1 r_2$ for some $r_1$and r_2 be regular expression over $\Sigma = \{0, 1\}$.

And $r' = r^R = r_1^R r_2^R$

Then $Rev(L(r)) = Rev(L(r_1 r_2))$

$$= Rev\left(Rev(L(r_1^R))Rev(L(r_2^R))\right) \quad \text{(Since IH)}$$

$$= Rev\left(Rev(L(r_2^R))Rev(L(r_1^R))\right)$$

$$= L(r_2^R)L(r_1^R)$$

$$= L(r_2^R r_1^R)$$

$$= L(r^R)$$

$$= L(r')$$

Case 3: $r = r_1 + r_2$ for some $r_1$and r_2 be regular expression over $\Sigma = \{0, 1\}$.

And $r' = r^R = r_1^R + r_2^R$

Then $Rev(L(r)) = Rev(L(r_1 + r_2))$

$$= Rev(L(r_1) \cup L(r_2))$$

$$= Rev\left(Rev(L(r_1^R)) \cup Rev(L(r_2^R))\right) \quad \text{(Since IH)}$$

$$= Rev\left(Rev(L(r_1^R + r_2^R))\right)$$

$$= L(r^R) = L(r')$$

So all three cases hold.

By structural induction, combine base cases and inductive step, we proved

$\forall r \in RE, \exists r' RE\ s.t. Rev(L(r)) = L(r')$ ■

(b)

Let $r \in RE, r = xy,\ x, y \in RE.$

$r' = x$

Define $P(r)$: $\exists r' RE\ s.t. Prefix(L(r)) = L(r')$

Want to show: $\forall r \in RE, P(r)$

Proof: (Structural Induction)

Base Case: Case 1: $r = \emptyset,\ r' = xy = \emptyset.$

Then $Prefix(L(r)) = Prefix(\{\,\}) = \{\,\} = L(r')$

Case 2: $r = \varepsilon,\ r' = xy = \varepsilon.$

Then $Prefix\big(L(r)\big) = Prefix(\{\varepsilon\}) = \{\varepsilon\} = L(r^{'})$

Case 3: $r = a,\ a \in \Sigma^*,\ r^{'} = xy = a\varepsilon.$

Then $Prefix\big(L(r)\big) = Prefix(\{a\}) = \{a\} = L(r^{'})$

So $P(\emptyset), P(\varepsilon), P(a)$ hold.

Inductive Step: Assume $P(r)\ holds$

i.e. $\forall k < n, k \in \mathbb{N}, |r| = k\ and\ Prefix\big(L(r)\big) = L\big(r^{'}\big).$

Want to show: $n \in \mathbb{N}, |r| = n\ and\ Prefix\big(L(r)\big) = L\big(r^{'}\big)$

There are 3 cases to consider

Case 1: $r = r_1^*$ for some $r_1$be regular expression over $\Sigma = \{0, 1\}$.

And $r^{'} = r = r_1^*$

Then $Prefix\big(L(r)\big) = Prefix\big(L(r_1^*)\big)$

$= L(r_1^*)$ (by IH)

$= L(r^{'})$

Case 2: $r = r_1 r_2$ for some $r_1$and r_2 be regular expression over $\Sigma = \{0, 1\}$.

And $r^{'} = r = r_1 r_2$

Then $Prefix\big(L(r)\big) = Prefix\big(L(r_1 r_2)\big)$

$= Prefix(L(r_1))Prefix(L(r_2))$ (Since IH)

$= L(r_1)L(r_2)$

$= L(r_1 r_2)$

$= L(r^{'})$

Case 3: $r = r_1 + r_2$ for some $r_1$and r_2 be regular expression over $\Sigma = \{0, 1\}$.

And $r^{'} = r = r_1 + r_2$

Then $Prefix\big(L(r)\big) = Prefix\big(L(r_1 + r_2)\big)$

$= Prefix\big(L(r_1) \cup L(r_2)\big)$

$= Prefix(L(r_1)) \cup Prefix(L(r_2))$

$= L(r_1 + r_2)$ (Since IH)

$= L(r^{'})$

So all three cases hold.

By structural induction, combine base cases and inductive step, we proved

$\forall r \in RE, \exists r^{'} RE\ s.t.\ Prefix\big(L(r)\big) = L(r^{'})$ ■

(c)

Define $P(r): r \in RE, r\ does\ not\ contain\ Kleene\ Star, then\ |RE(r)|\ is\ finite.$

Want to show: $\forall r \in RE, P(r)$

Proof: (Structural Induction)

Base Case: Case 1: $r = \emptyset$, r does not have Kleene Star

Then $RE(r) = \{\}$ is finite

Case 2: $r = \varepsilon$, r does not have Kleene Star

Then $RE(r) = \{\varepsilon\}$ is finite

Case 3: $r = a,\ a \in \Sigma^*$, r does not have Kleene Star

Then $PE(r) = \{a\}$ is finite

So $P(\emptyset), P(\varepsilon), P(a)$ hold.

Inductive Step: Let $r_1, r_2 \in RE$, assume $P(r_1), P(r_2)$ hold

There are 3 cases to consider

Case 1: $r = r_1^*,\ r$ has a Kleene Star then $P(r)$ is vacuously true.

Case 2: $r = r_1 r_2,\ r$ does not have Kleene Star, r_1, r_2 neither.

$L(r) = L(r_1 r_2) = L(r_1)L(r_2),\ L(r_1), L(r_2)$ is finite (By IH) so is $L(r)$.

Case 3: $r = r_1 + r_2,\ r$ does not have Kleene Star, r_1, r_2 neither.

$L(r) = L(r_1 + r_2) = L(r_1) \cup L(r_2),\ L(r_1), L(r_2)$ is finite (By IH).

Then $L(r_1) \cup L(r_2)$ is finite so is $L(r)$.

So all three cases hold.

By structural induction, combine base cases and inductive step, we proved that

$\forall r \in RE, r\ does\ not\ contain\ Kleene\ Star, then\ |RE(r)|\ is\ finite.$ ■

Question 5

(a)

Proof: (Contradiction)

Suppose, for the propose of contradiction, that we can find a correct DfA that has 8 states

By Pigeonhole Principle, if we choose 9 strings over Σ, then at least two of those strings must end at the same state

We pick nine prefixes of length 2 of x, as nine strings, i.e. aa, ab, ac, ba, bb, bc, ca, cb, cc

For one of the pairs of strings x, y in previous line (we don't know which pair),

the supposed 8-state DFA is forced into the same state for both strings (because of

Pigeonhole), and xz and yz must be both accepted or both rejected, for any string z s.t. $|xz| = |yz| = 4$

We will show, for each possible pair, that this is NOT true

Pair 1: aa and ab

Choose z = aa

Then xz = aaaa is accepted, yz = abaa is rejected

Pair 2: aa and ac

Choose z = aa

Then xz = aaaa is accepted, yz = acaa is rejected

Pair 3: aa and ba

Choose z = aa

Then xz = aaaa is accepted, yz = baaa is rejected

Pair 4: aa and bb

Choose z = aa

Then xz = aaaa is accepted, yz = bbaa is rejected

Pair 5: aa and bc

Choose z = aa

Then xz = aaaa is accepted, yz = bcaa is rejected

Pair 6: aa and ca

Choose z = aa

Then xz = aaaa is accepted, yz = caaa is rejected

Pair 7: aa and cb

Choose z = aa

Then xz = aaaa is accepted, yz = cbaa is rejected

Pair 8: aa and cc

Choose z = aa

Then xz = aaaa is accepted, yz = ccaa is rejected

Pair 9: ab and ac

Choose z = ba

Then xz = abba is accepted, yz = acba is rejected

Pair 10: ab and ba

Choose z = ba

Then xz = abba is accepted, yz = baba is rejected

Pair 11: ab and bb

Choose z = ba

Then xz = abba is accepted, yz = bbba is rejected

Pair 12: ab and bc

Choose z = ba

Then xz = abba is accepted, yz = bcba is rejected

Pair 13: ab and ca

Choose z = ba

Then xz = abba is accepted, yz = caba is rejected

Pair 14: ab and cb

Choose z = ba

Then xz = abba is accepted, yz = cbba is rejected

Pair 15: ab and cc

Choose z = ba

Then xz = abba is accepted, yz = ccba is rejected

Pair 16: ac and ba

Choose z = ca

Then xz = acca is accepted, yz = baca is rejected

Pair 17: ac and bb

Choose z = ca

Then xz = acca is accepted, yz = bbca is rejected

Pair 18: ac and bc

Choose z = ca

Then xz = acca is accepted, yz = bcca is rejected

Pair 19: ac and ca

Choose z = ca

Then xz = acca is accepted, yz = caca is rejected

Pair 20: ac and cb

Choose z = ca

Then xz = acca is accepted, yz = cbca is rejected

Pair 21: ac and cc

Choose z = ca

Then xz = acca is accepted, yz = ccca is rejected

Pair 22: ba and bb

Choose z = ab

Then xz = baab is accepted, yz = bbab is rejected

Pair 23: ba and bc

Choose z = ab

Then xz = baab is accepted, yz = bcab is rejected

Pair 24: ba and ca

Choose z = ab

Then xz = baab is accepted, yz = caab is rejected

Pair 25: ba and cb

Choose z = ab

Then xz = baab is accepted, yz = cbab is rejected

Pair 26: ba and cc

Choose z = ab

Then xz = baab is accepted, yz = ccab is rejected

Pair 27: bb and bc

Choose z = bb

Then xz = bbbb is accepted, yz = bcbb is rejected

Pair 28: bb and ca

Choose z = bb

Then xz = bbbb is accepted, yz = cabb is rejected

Pair 29: bb and cb

Choose z = bb

Then xz = bbbb is accepted, yz = cbbb is rejected

Pair 30: bb and cc

Choose z = bb

Then xz = bbbb is accepted, yz = ccbb is rejected

Pair 31: bc and ca

Choose z = cb

Then xz = bccb is accepted, yz = cacb is rejected

Pair 32: bc and cb

Choose z = cb

Then xz = bccb is accepted, yz = cbcb is rejected

Pair 33: bc and cc

Choose z = cb

Then xz = bccb is accepted, yz = cccb is rejected

Pair 34: ca and cb

Choose z = ac

Then xz = caac is accepted, yz = cbac is rejected

Pair 35: ca and cc

Choose z = ac

Then xz = caac is accepted, yz = ccac is rejected

Pair 36: cb and cc

Choose z = bc

Then xz = cbbc is accepted, yz = ccbc is rejected

By now we have showed that none two of those nine strings we chosen must end at the same state, which is a contradiction to our assumption

Hence we can conclude that any DFA accepts L_{R4} has at least nine states ■

(b)

By (a) we may find out that the critical of finding list number of states is finding the largest number of prefixes with the same length and at the same time every pair of them may end at different states

To make the number of prefixes as large as possible, we need to let the length of the prefix be as large as possible

So for a string s s.t. $|s|$ = n, $\left\lfloor\frac{n}{2}\right\rfloor$ is the largest possible length of the prefix which can ensure that every pair of them may end at different states

Since $|\Sigma|$ = 3, for a string s s.t. $|s|$ = n, there are $3^{\left\lfloor\frac{n}{2}\right\rfloor}$ different prefixes

Hence by conclusion of (a), any DFA accepts L_R has at least $3^{\left\lfloor\frac{n}{2}\right\rfloor}$ states

Since the length of a string s, n, which belongs to $\mathbb{N}$, is infinite

$3^{\left\lfloor\frac{n}{2}\right\rfloor}$ may also be infinite because it depends only on n

This gives us a result that the DFA may end up with infinite number of states when s is infinite long, which is impossible because the number of states of a DFA is finite

Hence we cannot say anything about a DFA that accepts L_R simply by generalizing the result of part (a)