

CSC236 A3

Kaiqu Liang, HanTang Li

December 3, 2018

Problem 1

Proof:

Let $i \in \mathbb{N}$, assume $x \in \mathbb{N}$,

First I want to show if the loop executed at least i times then $y_i = x_i^3$

Let $P(i)$ be predicate:

$P(i)$: if loop execute at least i times, then $y_i = x_i^3$ and $y_i, x_i \in \mathbb{N}$

I will use simple induction to prove $\forall i, P(i)$ holds.

Basis:

$i = 0$, then the loop not executed

we have $x_0 \in \mathbb{N}$ by assumption, and from the second line of the function, $y_0 = x_0^3$ thus $y_0 \in \mathbb{N}$. so $P(0)$ holds.

Inductive step:

Let i be arbitrary nature number,

assume $P(i)$ holds, i.e $y_i = x_i^3$ and $x_i, y_i \in \mathbb{N}$.

Want to show $P(i + 1)$ holds if the loop executed at least $i + 1$ times.

by fourth line $x_{i+1} = x_i - 1$,

since by IH $y_i = x_i^3$, and by loop condition, $y_i \neq 0$

thus $\sqrt[3]{y_i} = x_i \neq 0$.

Also by IH we know $x_i \in \mathbb{N}$,

thus $x_i \geq 1$,

therefore $x_{i+1} = x_i - 1 \in \mathbb{N}$.

From the fifth line:

$$\begin{aligned}
 y_{i+1} &= y_i - 3(x_{i+1})^2 - 3(x_{i+1}) - 1 \\
 &= y_i - 3(x_i - 1)^2 - 3(x_i - 1) - 1 \quad \# \text{ since by line 4, } x_{i+1} = x_i - 1 \\
 &= x_i^3 - 3(x_i - 1)^2 - 3(x_i - 1) - 1 \quad \# \text{ since by IH, } y_i = x_i^3 \\
 &= x_i^3 - 3(x_i^2 - 2x_i + 1) - 3x_i + 3 - 1 \\
 &= x_i^3 - 3x_i^2 + 6x_i - 3 - 3x_i + 3 - 1 \\
 &= x_i^3 - 3x_i^2 + 3x_i - 1 \\
 &= (x_i - 1)^3 \\
 &= x_{i+1}^3
 \end{aligned}$$

Since $x_{i+1} \in \mathbb{N}$, so $y_{i+1} = x_{i+1}^3 \in \mathbb{N}$

Hence, $P(i+1)$ holds.

Want to show the loop can terminates.

Proof:

Want to show there is a decreasing sequence linked to the loop.

Assume if there is $i+1$ iteration, and $i \in \mathbb{N}$,

Want to show $y_{i+1} < y_i$.

Since in each loop, the fourth line shows that

$$x_{i+1} = x_i - 1$$

thus

$$x_{i+1} < x_i$$

$$x_{i+1}^3 < x_i^3$$

$$y_{i+1} < y_i \quad \# \text{ by loop invariant, } y_{i+1} = x_{i+1}^3 \text{ and } y_i = x_i^3$$

Also by loop invariant we proved above, if there is a loop then variable y inside the loop is nature number.

Thus we have exhibited a decreasing sequence of natural numbers linked to loop iterations. The last element of this sequence has the index of the last loop iteration, so the loop terminates.

Problem 2

a) Here is my specification for $M_a = \{Q, \Sigma, \delta, s, F\}$ that accept L_a

$$\begin{aligned} Q &= \{A, D_a\}, \\ \Sigma &= \{a, b\} \\ \delta &= \begin{array}{|c|c|c|} \hline \delta & A & D_a \\ \hline a & A & D_a \\ \hline b & D_a & D_a \\ \hline \end{array} \\ s &= q_0 = A \\ F &= \{A\} \end{aligned}$$

Proof: First, define Σ^* as the smallest set such that:

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

(b) $s \in \Sigma^* \implies sa \in \Sigma^* \wedge sb \in \Sigma^*$

prove that M_a accepts L_a : Define $P(s)$ as:

$$P(s) : \delta^*(A, s) = \begin{cases} A & \text{if } s \text{ contains no } b \\ D_a & \text{if } s \text{ contains } b \end{cases}$$

I prove $\forall s \in \Sigma^*, P(s)$ by structural induction.

Base case: $s = \epsilon$, ϵ certainly has no b.

we have $\delta^*(A, \epsilon) = A$,

so the implication in the first line of the invariant is true in this case.

Also, since ϵ does not contain b,

the implication in the second line of the invariant is vacuously true,

So $P(\epsilon)$ holds.

Inductive step:

Let $s \in \Sigma^*$ and assume $P(s)$. I will show that $P(sa)$ and $P(sb)$ follow.

There are two cases to consider:

case sa: Then

$$\begin{aligned} \delta^*(A, sa) &= \delta(\delta^*(A, s), a) \\ &= \begin{cases} \delta(A, a) & \text{if } s \text{ contains no } b, & \# \text{by } P(s) \\ \delta(D_a, a) & \text{if } s \text{ contains } b, & \# \text{by } P(s) \end{cases} \\ &= \begin{cases} A & \text{if } sa \text{ contains no } b, & \# \text{one more } a \\ D_a & \text{if } sa \text{ contains } b, & \# \text{one more } a \end{cases} \end{aligned}$$

case sb: Then

$$\begin{aligned}
\delta^*(A, sb) &= \delta(\delta^*(A, s), b) \\
&= \begin{cases} \delta(A, b) & \text{if } s \text{ contains no } b, & \# \text{by } P(s) \\ \delta(D_0, b) & \text{if } s \text{ contains } b, & \# \text{by } P(s) \end{cases} \\
&= \begin{cases} D_a & \text{if } sb \text{ contains no } b, & \# \text{one more } b \\ D_a & \text{if } sb \text{ contains } b, & \# \text{one more } b \end{cases}
\end{aligned}$$

Since sb certainly contains b ,
the statement if sb contains no b , then

$$\delta^*(A, sb) = A$$

is vacuously true.

Therefore

$$\delta^*(A, sb) = \begin{cases} A & \text{if } sb \text{ contains no } b \\ D_a & \text{if } sb \text{ contains } b \end{cases}$$

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

The first line of the invariant ensures that all strings that contains no b are accepted. The contrapositive of the second line ensure that any string that does not drive the machine to state D_a does not contains b , in other words all strings that drive the machine to state A does not contains b . So, M_a accepts L_a .

b) Here is my specification for $M_b = \{Q, \Sigma, \delta, s = q_0, F\}$ that accept L_b

$$\begin{aligned}
Q &= \{B, D_b\}, \\
\Sigma &= \{a, b\} \\
\delta &= \begin{array}{|c|c|c|} \hline \delta & B & D_b \\ \hline a & D_b & D_b \\ \hline b & B & D_b \\ \hline \end{array} \\
s &= q_0 = B \\
F &= \{B\}
\end{aligned}$$

Proof: First, define Σ^* as the smallest set such that:

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

(b) $s \in \Sigma^* \implies sa \in \Sigma^* \wedge sb \in \Sigma^*$

prove that M_b accepts L_b : Define $P(s)$ as:

$$P(s) : \delta^*(B, s) = \begin{cases} B & \text{if } s \text{ contains no } a \\ D_b & \text{if } s \text{ contains } a \end{cases}$$

I prove $\forall s \in \Sigma^*, P(s)$ by structural induction.

Base case: $s = \epsilon$, ϵ certainly has no a .

we have $\delta^*(B, \varepsilon) = B$,

so the implication in the first line of the invariant is true in this case.

Also, since ε does not contains a,

the implication in the second line of the invariant is vacuously true,

So $P(\varepsilon)$ holds.

Inductive step:

Let $s \in \Sigma^*$ and assume $P(s)$. I will show that $P(sa)$ and $P(sb)$ follows.

There are two cases to consider:

case sa: Then

$$\begin{aligned} \delta^*(B, sa) &= \delta(\delta^*(B, s), a) \\ &= \begin{cases} \delta(B, a) & \text{if } s \text{ contains no } a, & \# \text{by } P(s) \\ \delta(D_b, a) & \text{if } s \text{ contains } a, & \# \text{by } P(s) \end{cases} \\ &= \begin{cases} D_b & \text{if } sa \text{ contains } a, & \# \text{one more } a \\ D_b & \text{if } sa \text{ contains } a, & \# \text{one more } a \end{cases} \end{aligned}$$

Since sa certainly contains a,

the statement "if sa contains no a then $\delta^*(B, sa) = B$ "

is vacuously true.

Therefore

$$\delta^*(B, sa) = \begin{cases} B & \text{if } sa \text{ contains no } a \\ D_b & \text{if } sa \text{ contains } a \end{cases}$$

case sb: Then

$$\begin{aligned} \delta^*(B, sb) &= \delta(\delta^*(B, s), b) \\ &= \begin{cases} \delta(B, b) & \text{if } s \text{ contains no } a, & \# \text{by } P(s) \\ \delta(D_b, b) & \text{if } s \text{ contains } a, & \# \text{by } P(s) \end{cases} \\ &= \begin{cases} B & \text{if } sb \text{ contains no } a, & \# \text{one more } b \\ D_b & \text{if } sb \text{ contains } a, & \# \text{one more } b \end{cases} \end{aligned}$$

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

The first line of the invariant ensures that all strings that contains no a are accepted. The contrapositive of the second line ensure that any string that does not drive the machine to state D_b does not contains a, in other words all strings that drive the machine to state B does not contains a. So, M_b accepts L_b .

c) Here is my specification for $M_2 = \{Q, \Sigma, \delta, s = q_0, F\}$ that accept L_2

$$\begin{aligned} Q &= \{E, O\}, \\ \Sigma &= \{a, b\} \\ \delta &= \begin{array}{|c|c|c|} \hline \delta & E & O \\ \hline a & O & E \\ \hline b & O & E \\ \hline \end{array} \\ s &= q_0 = E \\ F &= \{E\} \end{aligned}$$

Proof: First, define Σ^* as the smallest set such that:

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

(b) $s \in \Sigma^* \implies sa \in \Sigma^* \wedge sb \in \Sigma^*$

prove that M_2 accepts L_2 : Define $P(s)$ as:

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

I prove $\forall s \in \Sigma^*, P(s)$ by structural induction.

Base case: $s = \epsilon$,

$|\epsilon| = 0$ which is even and $\delta^*(E, \epsilon) = E$,

so the implication in the first line of the invariant is true in this case.

Also, since ϵ is not odd,

the implication in the second line of the invariant is vacuously true,

So $P(\epsilon)$ holds.

Inductive step:

Let $s \in \Sigma^*$ and assume $P(s)$. I will show that $P(sa)$ and $P(sb)$ follows.

There are two cases to consider:

case sa: Then

$$\begin{aligned} \delta^*(E, sa) &= \delta(\delta^*(E, s), a) \\ &= \begin{cases} \delta(E, a) & \text{if } |s| \text{ is even,} & \# \text{by } P(s) \\ \delta(O, a) & \text{if } |s| \text{ is odd,} & \# \text{by } P(s) \end{cases} \\ &= \begin{cases} O & \text{if } |sa| \text{ is odd,} & \# \text{one more a} \\ E & \text{if } |sa| \text{ is even,} & \# \text{one more a} \end{cases} \end{aligned}$$

case sb: Then

$$\begin{aligned} \delta^*(E, sb) &= \delta(\delta^*(E, s), b) \\ &= \begin{cases} \delta(E, b) & \text{if } |s| \text{ is even,} & \# \text{by } P(s) \\ \delta(O, b) & \text{if } |s| \text{ is odd,} & \# \text{by } P(s) \end{cases} \\ &= \begin{cases} O & \text{if } |sb| \text{ is odd,} & \# \text{one more b} \\ E & \text{if } |sb| \text{ is even,} & \# \text{one more b} \end{cases} \end{aligned}$$

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

The first line of the invariant ensures that all strings with an even length are accepted. The contrapositive of the second line of the invariant ensures that any string that does not drive the machine to state O does not has an odd length, in other words all strings that drive the machine to state E have even length. So, M_2 accepts L_2 .

d) Here is my specification for $M_{a|b} = \{Q, \Sigma, \delta, s = q_0, F\}$ that accept $L_a \cup L_b$

$$\{Q = \{(A, B), (A, D_b), (D_a, B), (D_a, D_b)\},$$

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

δ	(A, B)	(A, D_b)	(D_a, B)	(D_a, D_b)
$\delta =$	a	(A, D_b)	(A, D_b)	(D_a, D_b)
	b	(D_a, B)	(D_a, D_b)	(D_a, D_b)

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

$$F = \{(A, B), (A, D_b), (D_a, B)\}$$

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

Denote the states for M_a as Q_1 ,

the states for M_b as Q_2 ,

their respective transition function 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 $(q_1, q_2, c) \in Q_1 \times Q_2 \times \Sigma$,

then $\delta_{a|b}((q_1, q_2), c) = (\delta_a(q_1, c), \delta_b(q_2, c))$.

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

$$P(s) = \delta^*((A, B), s) = \begin{cases} (A, B) & \text{if } s \text{ contains no } a \text{ and no } b \\ (A, D_b) & \text{if } s \text{ contains no } b \text{ but contains } a \\ (D_a, B) & \text{if } s \text{ contains } b \text{ and contains no } a \\ (D_a, D_b) & \text{if } s \text{ contains both } a \text{ and } b \end{cases}$$

The implication on the first line ensures that string that contains no a and no b which is an empty string drives the machine to (A, B) , hence accepted by $M_{a|b}$.

The implication on the second line ensures that all strings that contains no b but contains a drive the machine to (A, D_b) , hence accepted by $M_{a|b}$.

The implication on the third line ensures that all string that contains b and contains no a drive the machine to (D_a, B) , hence accepted by $M_{a|b}$.

The contrapositive of the implications on the fourth line ensure that any string that does not drive the machines to state in fourth line do not contains both a and b , in other words, any string that drives the machine to states in first, second or third line must contain no a and no b , or contain no b but contain a , or contain b and contain no a .

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

e) Here is my specification for $M_{a|b \text{ even}} = \{Q, \Sigma, \delta, s = q_0, F\}$ that accept $(L_a \cup L_b) \cap L_2$

$$\begin{aligned} Q &= \{(A, B, E), (A, D_b, O), (A, D_b, E), (D_a, B, E), \\ &\quad (D_a, B, O), (D_a, D_b, O), (D_a, D_b, E)\}, \\ \Sigma &= \{a, b\} \end{aligned}$$

$$\delta = \begin{array}{c|cccc} \delta & (A, B, E) & (A, D_b, O) & (A, D_b, E) & (D_a, B, E) \\ \hline a & (A, D_b, O) & (A, D_b, E) & (A, D_b, O) & (D_a, D_b, O) \\ \hline b & (D_a, B, O) & (D_a, D_b, E) & (D_a, D_b, O) & (D_a, B, O) \end{array}$$

$$\begin{array}{c|ccc} \delta & (D_a, B, O) & (D_a, D_b, O) & (D_a, D_b, E) \\ \hline a & (D_a, D_b, E) & (D_a, D_b, E) & (D_a, D_b, O) \\ \hline b & (D_a, B, E) & (D_a, D_b, E) & (D_a, D_b, O) \end{array}$$

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

$$F = \{(A, B, E), (A, D_b, E), (D_a, B, E)\}$$

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

the stage for M_a as Q_a , the stage for M_b as Q_b , the stage for M_2 as Q_2 .

Their respective transition function as δ_a , δ_b and δ_2 ,

and the transition function for $M_{a|b \text{ even}}$ as $\delta_{a|b \text{ even}}$

Inspection of $\delta_{a|b \text{ even}}$ shows that if $(q_1, q_2, q_3, c) \in Q_a \times Q_b \times Q_2 \times \Sigma$

then $\delta_{a|b \text{ even}}((q_1, q_2, q_3), c) = (\delta_a(q_1, c), \delta_b(q_2, c), \delta_2(q_3, c))$

except when $(q_1, q_2, q_3) = (A, B, O)$

$(q_1, q_2, q_3) = (A, B, O)$ is not a valid state since it is impossible to get a string contains no a and b with an odd length.

Thus the following invariant follows by simply taking conjunctions

of the component machines, except the special case that I discussed above

for any $s \in \Sigma^*$

$$P(s) = \delta^*((A, B, E), s) = \begin{cases} (A, B, E) & \text{if } s \text{ contains no a and no b, with an even length} \\ (A, D_b, O) & \text{if } s \text{ contains no b but contains a, with odd length} \\ (A, D_b, E) & \text{if } s \text{ contains no b but contains a, with even length} \\ (D_a, B, E) & \text{if } s \text{ contains no a but contains b, with even length} \\ (D_a, B, O) & \text{if } s \text{ contains no a but contains b, with odd length} \\ (D_a, D_b, O) & \text{if } s \text{ contains a and b, with odd length} \\ (D_a, D_b, E) & \text{if } s \text{ contains a and b, with even length} \end{cases}$$

The implication on the first line ensure that string that contains no a and no b with an even length which is an empty string ends up in state (A, B, E) , accepted by $M_{a|b \text{ even}}$

The implication on the third line ensures that all strings that contain no b but contain a with an even length end up in state (A, D_b, E) , accepted by $M_{a|b \text{ even}}$

The implication on the fourth line ensures that all strings that contain no a but contain b with an even length end up in state (D_a, B, E) , accepted by $M_{a|b \text{ even}}$

The contrapositive of the implications on the other 4 lines ensure that any string that does not drive the machine to one of those 4 states must contains no a and no b with even length,

or contains no b but contains a with even length , or contains b and contains no a with even length.

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

Problem 3

Here is my specification for $M_0 = \{Q, \Sigma, \delta, s = q_0, F\}$ that accept $L_a \cup \{\varepsilon\}$

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

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

$$\delta =$$

δ	q_0	q_1	q_2
0	q_0	q_1	q_2
3	q_0	q_1	q_2
6	q_0	q_1	q_2
9	q_0	q_1	q_2
1	q_1	q_2	q_0
4	q_1	q_2	q_0
7	q_1	q_2	q_0
2	q_2	q_0	q_1
5	q_2	q_0	q_1
8	q_2	q_0	q_1

$$s = q_0$$

$$F = \{q_0\}$$

For $M_1 = \{Q, \Sigma, \delta, s = q_0, F\}$ that accept $L_1\}$

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

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

$$\delta =$$

δ	q_0	q_1	q_2
0	q_0	q_1	q_2
3	q_0	q_1	q_2
6	q_0	q_1	q_2
9	q_0	q_1	q_2
1	q_1	q_2	q_0
4	q_1	q_2	q_0
7	q_1	q_2	q_0
2	q_2	q_0	q_1
5	q_2	q_0	q_1
8	q_2	q_0	q_1

$$s = q_0$$

$$F = \{q_1\}$$

For $M_2 = \{Q, \Sigma, \delta, s = q_0, F\}$ that accept $L_1\}$

$$\{Q = \{q_0, q_1, q_2\}, \\ \Sigma = \{a, b\}$$

$$\delta =$$

δ	q_0	q_1	q_2
0	q_0	q_1	q_2
3	q_0	q_1	q_2
6	q_0	q_1	q_2
9	q_0	q_1	q_2
1	q_1	q_2	q_0
4	q_1	q_2	q_0
7	q_1	q_2	q_0
2	q_2	q_0	q_1
5	q_2	q_0	q_1
8	q_2	q_0	q_1

$$s = q_0$$

$$F = \{q_2\}$$

Explain: To show $L_0 = \text{Rev}(L_0)$,

we can show both L_0 and $\text{Rev}(L_0)$ are accepted by M_0

First, I want to explain that let $e_1, e_2 \in \Sigma^*$, q_0 be starting state,

if e_1, e_2 can takes the machine to same final state, then e_1, e_2

contains the same number of symbols that belongs to set $\{1, 4, 7\} \subset \Sigma$

and $\{2, 5, 8\} \subset \Sigma$, i.e $\delta^*(q_0, e_1) = \delta^*(q_0, e_2)$

I define the following transition as positive transition:

Let $a \in \{1, 4, 7\} \subset \Sigma$ be arbitrary. By the transition function,

$$\delta(q_0, a) = q_1$$

$$\delta(q_1, a) = q_2$$

$$\delta(q_2, a) = q_0$$

I define the following transition as negative transition:

Let $b \in \{2, 5, 8\} \subset \Sigma$ be arbitrary. By the transition function,

$$\delta(q_0, b) = q_2$$

$$\delta(q_1, b) = q_0$$

$$\delta(q_2, b) = q_1$$

Show these two transition commute:

Let $e \in \Sigma^*$ be arbitrary then, concatenate e with ab, ba separately, want to show

$$\delta^*(q_0, eab) = \delta^*(q_0, eba)$$

case1: if $\delta^*(q_0, e) = q_0$

then

$$\text{LHS} = \delta^*(q_0, eab) = \delta(\delta^*(q_0, ea), b) = \delta(\delta(\delta^*(q_0, e), a), b) = \delta(\delta(q_0, a), b) = \delta(q_1, b) = q_0$$

$$\text{RHS} = \delta^*(q_0, eba) = \delta(\delta^*(q_0, eb), a) = \delta(\delta(\delta^*(q_0, e), b), a) = \delta(\delta(q_0, b), a) = \delta(q_2, a) = q_0$$

LHS = RHS, so case1 holds

case2: if $\delta^*(q_0, e) = q_1$

then

$$\text{LHS} = \delta^*(q_0, eab) = \delta(\delta^*(q_0, ea), b) = \delta(\delta(\delta^*(q_0, e), a), b) = \delta(\delta(q_1, a), b) = \delta(q_2, b) = q_1$$

$$\text{RHS} = \delta^*(q_0, eba) = \delta(\delta^*(q_0, eb), a) = \delta(\delta(\delta^*(q_0, e), b), a) = \delta(\delta(q_1, b), a) = \delta(q_0, a) = q_1$$

LHS = RHS, so case2 holds

case3: if $\delta^*(q_0, e) = q_2$

then

$$\text{LHS} = \delta^*(q_0, eab) = \delta(\delta^*(q_0, ea), b) = \delta(\delta(\delta^*(q_0, e), a), b) = \delta(\delta(q_2, a), b) = \delta(q_0, b) = q_2$$

$$\text{RHS} = \delta^*(q_0, eba) = \delta(\delta^*(q_0, eb), a) = \delta(\delta(\delta^*(q_0, e), b), a) = \delta(\delta(q_2, b), a) = \delta(q_1, a) = q_2$$

LHS = RHS, so case3 holds

Thus, the positive and negative transition is commute for any string $\in \Sigma^*$

Also, from the transition function,

we can see the property of $\{0, 3, 6, 9\}$

let $c \in \{0, 3, 6, 9\} \subset \Sigma$

$$\delta^*(q_0, ec) = \delta(\delta^*(q_0, e)c) = \delta^*(q_0, e)$$

Hence, $\delta^*(q_0, e) = \delta^*(q_0, e \text{ with symbols in } \{0, 3, 6, 9\} \text{ kicked off})$

Final step: explain $L_0 = \mathcal{Rev}(L_0)$.

Let $x^R \in \mathcal{Rev}(L_0)$ be arbitrary

then there exist $x \in L_0$ which reverse of x^R is x

we know from definition of M_0 , $\delta^*(q_0, x) = q_0$.

Since x^R and x contains the same amount of elements belongs to set

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

since all transitions are commute

we can rearrange the symbols in x^R to get x , by previous explanation

$$\delta^*(q_0, x^R) = \delta^*(q_0, x), \text{ thus } L_0 = \mathcal{Rev}(L_0).$$

WLOG, $L_1 = \mathcal{Rev}(L_1)$ and $L_2 = \mathcal{Rev}(L_2)$ by using similar statements above

Problem 4

Define $P(r) : \exists r' \in \mathcal{RE}, \text{Rev}(L(r)) = L(r')$

I'll show $\forall r \in \mathcal{RE}, P(r)$ by using structural induction.

Base case 1: $r = \emptyset$

Take $r' = \emptyset$

Then $\text{Rev}(L(r)) = \text{Rev}(\emptyset) = \emptyset$

$L(r') = L(\emptyset) = \emptyset$

We have $\text{Rev}(L(r)) = L(r')$

$P(\emptyset)$ holds

Base case 2: $r = \epsilon$

Take $r' = \epsilon$

Then $\text{Rev}(L(r)) = \text{Rev}(\{\epsilon\}) = \epsilon$ (Reverse of empty string is also a empty string)

$L(r') = L(\epsilon) = \{\epsilon\}$

We have $\text{Rev}(L(r)) = L(r')$

$P(\epsilon)$ holds

Base case 3: $r = 0$

Take $r' = 0$

Then $\text{Rev}(L(r)) = \text{Rev}(\{0\}) = \{0\}$ (Reverse of string 0 is also string 0)

$L(r') = L(0) = \{0\}$

We have $\text{Rev}(L(r)) = L(r')$

$P(0)$ holds

Base case 4: $r = 1$

Take $r' = 1$

Then $\text{Rev}(L(r)) = \text{Rev}(\{1\}) = \{1\}$ (Reverse of string 1 is also string 1)

$L(r') = L(1) = \{1\}$

We have $\text{Rev}(L(r)) = L(r')$

$P(1)$ holds

Inductive step:

Let $e_1, e_2 \in \mathcal{RE}$. Assume $P(e_1)$ and $P(e_2)$.

So we have $\exists r_1 \in \mathcal{RE}$ such that $\text{Rev}(L(e_1)) = L(r_1)$

and $\exists r_2 \in \mathcal{RE}$ such that $\text{Rev}(L(e_2)) = L(r_2)$

① First I want to show $P(e_1 + e_2)$

Take $r' = r_1 + r_2$

I first want to show $L(r') \subseteq \text{Rev}(L(e_1 + e_2))$

Let $x \in L(r')$. Then $x \in L(r_1) \cup L(r_2)$

Then $x \in L(r_1)$ or $x \in L(r_2)$.

Case 1: $x \in L(r_1)$

Then $x \in \text{Rev}(L(e_1))$ (By IH)

Therefore $\exists y \in L(e_1)$ such that $y^R = x$

Then I want to show $\exists z \in L(e_1 + e_2)$ such that $z^R = x$

Take $z = y$

Since $z = y \in L(e_1)$, we have $z \in L(e_1) \cup L(e_2), z \in L(e_1 + e_2)$

Then $z^R = y^R = x$

Hence $x \in \text{Rev}(L(e_1 + e_2))$

Case 2: $x \in L(r_2)$

I can also conclude that $x \in \text{Rev}(L(e_1 + e_2))$ WLOG.

Therefore $x \in \text{Rev}(L(e_1 + e_2))$

We have $L(r') \subseteq \text{Rev}(L(e_1 + e_2))$

Secondly, I want to show $\text{Rev}(L(e_1 + e_2)) \subseteq L(r')$

Let $x \in \text{Rev}(L(e_1 + e_2))$

Then $\exists y \in L(e_1 + e_2)$ such that $y^R = x$

Case 1: $y \in L(e_1)$

Then $y^R \in \text{Rev}(L(e_1))$

$x \in \text{Rev}(L(e_1))$

$x \in L(r_1)$ (By IH)

$x \in L(r_1) \cup L(r_2) = L(r_1 + r_2) = L(r')$

Case 2: $y \in L(e_2)$

I can conclude that $x \in L(r')$ WLOG

By case 1 and case 2, $\text{Rev}(L(e_1 + e_2)) \subseteq L(r')$

Therefore I can conclude that $\text{Rev}(L(e_1 + e_2)) = L(r')$

$P(e_1 + e_2)$ holds.

② Second, I want to show $P(e_1 e_2)$

Take $r' = r_2 r_1$

I want to show $\text{Rev}(L(e_1 e_2)) = L(r_2 r_1)$

First I will show $\text{Rev}(L(e_1 e_2)) \subseteq L(r_2 r_1)$

Let $x \in \text{Rev}(L(e_1 e_2))$

Then $\exists y \in L(e_1 e_2)$ such that $y^R = x$

Let $y = y_1 y_2$, where $y_1 \in L(e_1), y_2 \in L(e_2)$

Then $x = y^R = y_2^R y_1^R$

It's clear that $y_2^R \in \text{Rev}(L(e_2)), y_1^R \in \text{Rev}(L(e_1))$

Hence $x \in \text{Rev}(L(e_2)) \circ \text{Rev}(L(e_1)) = L(r_2) \circ L(r_1) = L(r_2 r_1)$ (By IH)

I've shown that $\text{Rev}(L(e_1 e_2)) \subseteq L(r_2 r_1)$

Secondly, I want to show $L(r_2 r_1) \subseteq \text{Rev}(L(e_1 e_2))$

Let $x \in L(r_2 r_1)$

Then $x = x_1 x_2$ where $x_1 \in L(r_2), x_2 \in L(r_1)$

Since $L(r_2) = \text{Rev}(L(e_2)), L(r_1) = \text{Rev}(L(e_1))$ by IH,

we have $x_1 \in \text{Rev}(L(e_2)), x_2 \in \text{Rev}(L(e_1))$

By definition of Rev, we get $\exists y_1 \in L(e_2)$ such that $x_1 = y_1^R$

and $\exists y_2 \in L(e_1)$ such that $x_2 = y_2^R$

Take $y = y_2 y_1$

Then $y^R = (y_2 y_1)^R = y_1^R y_2^R = x_1 x_2 = x$

Hence $x \in \text{Rev}(L(e_1 e_2))$

Therefore $L(r_2 r_1) \subseteq \text{Rev}(L(e_1 e_2))$

By $\text{Rev}(L(e_1 e_2)) \subseteq L(r_2 r_1)$ and $L(r_2 r_1) \subseteq \text{Rev}(L(e_1 e_2))$, I can conclude that $L(r_2 r_1) = \text{Rev}(L(e_1 e_2))$

Hence, $P(e_1 e_2)$ holds.

③ Thirdly, I want to show $P(e_1^*)$

Take $r' = r_1^*$

I want to show $\text{Rev}(L(e_1^*)) = L(r_1^*)$

First I want to prove $\text{Rev}(L(e_1^*)) \subseteq L(r_1^*)$

Let $x \in \text{Rev}(L(e_1^*))$.

Then we have $\exists y \in L(e_1^*)$ such that $x = y^R$

WTS: $x \in L(r_1^*)$

We know that $\text{Rev}(L(e_1)) = L(r_1)$

Then apply Kleene star on both side: $[\text{Rev}(L(e_1))]^\circ = [L(r_1)]^\circ$

By property of L , we have $[L(r_1)]^\circ = L(r_1^*)$

Therefore we only need to show $x \in [\text{Rev}(L(e_1))]^\circ$

Since $y \in L(e_1^*) = [L(e_1)]^\circ$

$\exists m_1, m_2, \dots, m_k \in L(e_1)$ such that $y = m_1 m_2 \dots m_k$

Then $x = y^R = m_k^R m_{k-1}^R \dots m_2^R m_1^R$

Since $m_1, m_2, \dots, m_k \in L(e_1)$

we have $m_1^R, m_2^R, \dots, m_k^R \in \text{Rev}(L(e_1))$

Therefore $x \in [\text{Rev}(L(e_1))]^\circ$

Then we have $x \in L(r_1^*)$

Secondly, I want to show $L(r_1^*) \subseteq \text{Rev}(L(e_1^*))$

Let $x \in L(r_1^*)$

Since we know $\text{Rev}(L(e_1)) = L(r_1)$,

we get $[\text{Rev}(L(e_1))]^\circ = [L(r_1)]^\circ = L(r_1^*)$

Therefore $x \in [\text{Rev}(L(e_1))]^\circ$

Then we get $\exists x_1, x_2, \dots, x_k \in \text{Rev}(L(e_1))$ such that $x = x_1 x_2 \dots x_k$

I want to show $x \in \text{Rev}(L(e_1^*))$

That is to show $\exists y \in L(e_1^*)$ such that $y^R = x$

Since for $1 \leq i \leq k, x_i \in \text{Rev}(L(e_1))$

That is $\exists m_i \in L(e_1)$ such that $x_i = m_i^R$

Therefore $x = m_1^R m_2^R \dots m_k^R$ where $m_1, m_2, \dots, m_k \in L(e_1)$

Take $y = m_k m_{k-1} \dots m_2 m_1$

Then $y^R = (m_k m_{k-1} \dots m_1)^R = m_1^R m_2^R \dots m_k^R = x$

Hence $x \in \text{Rev}(L(e_1^*))$

I can conclude that $L(r_1^*) \subseteq \text{Rev}(L(e_1^*))$

By $\text{Rev}(L(e_1^*)) \subseteq L(r_1^*)$ and $L(r_1^*) \subseteq \text{Rev}(L(e_1^*))$,

we have $\text{Rev}(L(e_1^*)) = L(r_1^*)$

Therefore $P(e_1^*)$ holds.

By ①②③, we know $P(e_1 + e_2), P(e_1 e_2), P(e_1^*)$ holds.

By Base case 1, 2, 3, 4 and inductive step, I can conclude that $\forall r \in \mathcal{RE}, P(r)$

(b)

Define $P(r) : \exists r' \in \mathcal{RE}, \text{Prefix}(L(r)) = L(r')$

I will show $\forall r \in \mathcal{RE}, P(r)$ by structural induction.

Base case 1: $r = \emptyset$

Take $r' = \emptyset$

Then $\text{Prefix}(L(r)) = \text{Prefix}(\emptyset) = \emptyset$

$L(r') = L(\emptyset) = \emptyset$

We have $\text{Prefix}(L(r)) = L(r')$

$P(\emptyset)$ holds

Base case 2: $r = \epsilon$

Take $r' = \epsilon$

Then $\text{Prefix}(L(r)) = \text{Prefix}(\{\epsilon\}) = \epsilon$

$L(r') = L(\epsilon) = \epsilon$

We have $\text{Prefix}(L(r)) = L(r')$

$P(\epsilon)$ holds

Base case 3: $r = 0$

Take $r' = 0 + \epsilon$

Then $\text{Prefix}(L(r)) = \text{Prefix}(\{0\}) = \{0, \epsilon\}$

$L(r') = L(0 + \epsilon) = \{0, \epsilon\}$

We have $\text{Prefix}(L(r)) = L(r')$

$P(0)$ holds

Base case 4: $r = 1$

Take $r' = 1 + \epsilon$

Then $\text{Prefix}(L(r)) = \text{Prefix}(\{1\}) = \{1, \epsilon\}$

$L(r') = L(1 + \epsilon) = \{1, \epsilon\}$

We have $\text{Prefix}(L(r)) = L(r')$

$P(1)$ holds

Inductive step:

Let $e_1, e_2 \in \mathcal{RE}$. Assume $P(e_1)$ and $P(e_2)$.

So we have $\exists r_1 \in \mathcal{RE}$ such that $\text{Prefix}(L(e_1)) = L(r_1)$

and $\exists r_2 \in \mathcal{RE}$ such that $\text{Prefix}(L(e_2)) = L(r_2)$

① First I want to show $P(e_1 + e_2)$

Take $r' = r_1 + r_2$

I want to show $\text{Prefix}(L(e_1 + e_2)) = L(r')$

I first want to show $L(r') \subseteq \text{Prefix}(L(e_1 + e_2))$

Let $x \in L(r')$. Then $x \in L(r_1 + r_2) = L(r_1) \cup L(r_2)$

Case 1: $x \in L(r_1)$

Then $x \in \text{Prefix}(L(e_1))$.

by definition of Prefix, we get $\exists y_1 \in \Sigma^*$ such that $xy_1 \in L(e_1)$

WTS: $\exists y \in \Sigma^*$ such that $xy \in L(e_1 + e_2)$.

Take $y = y_1$

Then $xy = xy_1 \in L(e_1)$

Since $L(e_1) \subseteq (L(e_1) \cup L(e_2))$, $L(e_1) \cup L(e_2) = L(e_1 + e_2)$.

we have $L(e_1) \subseteq L(e_1 + e_2)$.

It follows that $xy \in L(e_1 + e_2)$

Therefore $x \in \text{Prefix}(L(e_1 + e_2))$

Case 2: $x \in L(r_2)$

I can also conclude that $x \in \text{Prefix}(L(e_1 + e_2))$ WLOG.

By case 1 and case 2, I can conclude that $x \in \text{Prefix}(L(e_1 + e_2))$.

Hence $L(r') \subseteq \text{Prefix}(L(e_1 + e_2))$.

Second I want to show $\text{Prefix}(L(e_1 + e_2)) \subseteq L(r')$

Let $x \in \text{Prefix}(L(e_1 + e_2))$

Then $\exists y \in \Sigma^*$ such that $xy \in L(e_1 + e_2) = L(e_1) \cup L(e_2)$

Case 1: $xy \in L(e_1)$

Then $x \in \text{Prefix}(L(e_1)) = L(r_1)$ by IH,

$L(r_1) \subseteq L(r_1) \cup L(r_2) = L(r_1 + r_2)$

Then $x \in L(r_1 + r_2)$

Case 2: $xy \in L(e_2)$

Then $x \in \text{Prefix}(L(e_2)) = L(r_2)$ by IH.

$L(r_2) \subseteq L(r_1) \cup L(r_2) = L(r_1 + r_2)$

Then $x \in L(r_1 + r_2)$

By case 1 and case 2, I can conclude that $x \in L(r_1 + r_2)$

Therefore $\text{Prefix}(L(e_1 + e_2)) \subseteq L(r')$

By $L(r') \subseteq \text{Prefix}(L(e_1 + e_2))$ and $\text{Prefix}(L(e_1 + e_2)) \subseteq L(r')$

I can conclude that $L(r') = \text{Prefix}(L(e_1 + e_2))$

Hence $P(e_1 + e_2)$ holds.

② Second I want to show $P(e_1e_2)$

Take $r' = r_1 + e_1r_2$

First I want to show $L(r') \subseteq \text{Prefix}(L(e_1e_2))$

Let $x \in L(r')$

Then $x \in L(r_1 + e_1r_2) = L(r_1) \cup L(e_1r_2)$.

WTS: $x \in \text{Prefix}(L(e_1e_2))$

That is to show $\exists y \in \Sigma^*, xy \in L(e_1e_2)$

Case 1: $x \in L(r_1)$

Then we have $x \in \text{Prefix}(L(e_1))$ by IH, Therefore by definition of prefix, we get

$\exists y_1 \in \Sigma^*, xy_1 \in L(e_1)$

Let $k \in L(e_2)$

Then take $y = y_1k$

Since $xy_1 \in L(e_1)$ and $k \in L(e_2)$,

we have $xy_1k \in L(e_1e_2)$

Therefore $xy \in L(e_1e_2)$

It follows that $x \in \text{Prefix}(L(e_1e_2))$.

Case 2: $x \in L(e_1r_2)$

Let $x = x_1x_2$, where $x_1 \in L(e_1)$ and $x_2 \in L(r_2)$.

Therefore $x_2 \in \text{Prefix}(L(e_2))$ by IH.

By definition of Prefix, we get $\exists y_2 \in \Sigma^*, x_2y_2 \in L(e_2)$.

Take $y = y_2$

Since $x_1 \in L(e_1)$, $x_2y_2 \in L(e_2)$.

we have $x_1x_2y_2 \in L(e_1e_2)$

Therefore $xy \in L(e_1e_2)$.

It follows that $x \in \text{Prefix}(L(e_1e_2))$

By case 1 and case 2, I can conclude that $L(r') \subseteq \text{Prefix}(L(e_1e_2))$

Then I want to show $\text{Prefix}(L(e_1e_2)) \subseteq L(r')$

Let $x \in \text{Prefix}(L(e_1e_2))$.

WTS: $x \in L(r_1 + e_1r_2)$

Case 1: $x \in \text{Prefix}(L(e_1))$

Then $x \in L(r_1)$ by IH.

Therefore $x \in L(r_1) \cup L(e_1r_2) = L(r_1 + e_1r_2)$.

Case 2: $x \notin \text{Prefix}(L(e_1))$

Since $x \in \text{Prefix}(L(e_1e_2))$ and $x \notin \text{Prefix}(L(e_1))$,

we get $\exists y \in \Sigma^*, xy \in L(e_1e_2)$ and $\forall z \in \Sigma^*, xz \notin L(e_1)$.

By definition, we get x is not in the prefix of $L(e_1)$ but in the prefix of $L(e_1e_2)$

Therefore x contains a string which is in $L(e_1)$ and the rest must be in $\text{Prefix}(L(e_2))$.

We can denote x by $x = x_1x_2$ where $x_1 \in L(e_1)$ and $x_2 \in \text{Prefix}(L(e_2))$.

Therefore $x \in L(e_1) \circ \text{Prefix}(L(e_2))$.

Then we have $x \in L(e_1) \circ L(r_2) = L(e_1r_2)$ since $\text{Prefix}(L(e_2)) = L(r_2)$ by IH.

Since $L(e_1r_2) \subseteq L(r_1) \cup L(e_1r_2)$,

we have $x \in L(r_1) \cup L(e_1r_2) = L(r_1 + e_1r_2)$

By case 1 and case 2, I can conclude that $x \in L(r_1 + e_1r_2)$.

Therefore $\text{Prefix}(L(e_1e_2)) \subseteq L(r')$.

By $L(r') \subseteq \text{Prefix}(L(e_1e_2))$ and $\text{Prefix}(L(e_1e_2)) \subseteq L(r')$,

I can conclude that $\text{Prefix}(L(e_1e_2)) = L(r')$.

Hence, $P(e_1e_2)$ holds.

③Thirdly, I want to show $P(e_1^*)$.

Take $r' = e_1^*r_1$.

First, I want to show $L(r') \subseteq \text{Prefix}(L(e_1^*))$.

Let $x \in L(r')$.

Then $x \in L(e_1^* r_1)$.

Let $x = x_1 x_2$ where $x_1 \in L(e_1^*)$, $x_2 \in L(r_1)$.

Therefore $x_2 \in \text{Prefix}(L(e_1))$ by IH.

By definition of prefix, we have $\exists y_2 \in \Sigma^*$, $x_2 y_2 \in L(e_1)$.

Take $y = y_2$.

Since $x_1 \in L(e_1^*)$, $x_2 y_2 \in L(e_1)$,

$x_1 x_2 y_2 \in L(e_1^*) \circ L(e_1) = L(e_1^*)$.

Therefore $xy \in L(e_1^*)$.

I can conclude that $L(r') \subseteq \text{Prefix}(L(e_1^*))$.

Second I want to show $\text{Prefix}(L(e_1^*)) \subseteq L(r')$.

Let $x \in \text{Prefix}(L(e_1^*))$.

Then we have $\exists y \in \Sigma^*$, $xy \in L(e_1^*)$.

We can denote x as $x = x_1 x_2$ where $x_1 \in L(e_1^*)$, $x_2 \in \Sigma^*$.

Then I will show $x_2 \in \text{Prefix}(L(e_1^*))$.

Since $xy \in L(e_1^*)$, $x_1 x_2 y \in L(e_1^*) = L(e_1^*) \circ L(e_1^*)$.

We know $x_1 \in L(e_1^*)$, therefore $x_2 y \in L(e_1^*)$.

It follows that $x_2 \in \text{Prefix}(L(e_1^*))$.

Then we have $x = x_1 x_2$, where $x_1 \in L(e_1^*)$ and $x_2 \in \text{Prefix}(L(e_1^*))$.

That is $x \in L(e_1^*) \circ \text{Prefix}(L(e_1^*))$.

Therefore, $x \in L(e_1^*) \circ L(r_1)$ (By IH, $\text{Prefix}(L(e_1)) = L(r_1)$).

It follows that $x \in L(e_1^* r_1) = L(r')$.

We have $\text{Prefix}(L(e_1^*)) \subseteq L(r')$.

By $L(r') \subseteq \text{Prefix}(L(e_1^*))$ and $\text{Prefix}(L(e_1^*)) \subseteq L(r')$, I can conclude that $\text{Prefix}(L(e_1^*)) = L(r')$.

Hence, $P(e_1^*)$ holds.

By ①②③, we know $P(e_1 + e_2)$, $P(e_1 e_2)$, $P(e_1^*)$ holds.

By all the base case and inductive step, I can conclude that $\forall r \in \mathcal{RE}$, $P(r)$.

(c)

Define $P(r)$:if r does not contain Kleene star, then $|L(r)|$ is finite.

WTS: $\forall r \in \mathcal{RE}$, $P(r)$ by using structural induction.

Base case:

When $r \in \{\emptyset, \epsilon, 0, 1\}$,

we know it does not contain Kleene star. Moreover,

$|L(\emptyset)| = |\emptyset| = 0$

$|L(\epsilon)| = |\{\epsilon\}| = 1$

$|L(0)| = |\{0\}| = 1$

$|L(1)| = |\{1\}| = 1$

Therefore $|L(r)|$ is finite.

Hence $P(\emptyset)$, $P(\epsilon)$, $P(0)$, $P(1)$ hold.

Inductive step:

Let $r_1, r_2 \in \mathcal{RE}$. Assume $P(r_1)$ and $P(r_2)$.

① Show $P(r_1 + r_2)$

Assume $r_1 + r_2$ does not contain $*$,

then r_1 does not contain $*$ and r_2 does not contain $*$.

By $P(r_1), P(r_2), |L(r_1)|$ and $|L(r_2)|$ are finite.

Therefore $|L(r_1 + r_2)| = |L(r_1) \cup L(r_2)| \leq |L(r_1)| + |L(r_2)|$.

Since $|L(r_1)| + |L(r_2)|$ is finite by $|L(r_1)|$ and $|L(r_2)|$ are finite,
we know $|L(r_1 + r_2)|$ is finite.

Hence $P(r_1 + r_2)$ holds.

② Show $P(r_1 r_2)$

Assume $r_1 r_2$ does not contain $*$,

then r_1 and r_2 both don't contain $*$.

By $P(r_1), P(r_2), |L(r_1)|$ and $|L(r_2)|$ are finite.

Then $|L(r_1 r_2)| = |L(r_1) \circ L(r_2)| \leq |L(r_1)| \cdot |L(r_2)|$.

Since $|L(r_1)| \cdot |L(r_2)|$ is finite by $|L(r_1)|$ and $|L(r_2)|$ are finite,
we know $|L(r_1 r_2)|$ is finite.

Hence $P(r_1 r_2)$ holds.

③ Show $P(r_1^*)$

Since r_1^* contains $*$, $P(r_1^*)$ is vacuously true.

By ①②③, we know $P(r_1 + r_2), P(r_1 r_2), P(r_1^*)$ hold.

By base case and inductive step, I can conclude that $\forall r \in \mathcal{RE}, P(r)$.

Problem 5

First I want to show it is impossible that two distinct length-2 prefixes of strings in L_{R4} drive the machine to the same state.

I will prove it by using contradiction.

Suppose there exists two distinct length-2 prefixes of strings in L_{R4} drive the machine to the same state.

Then let the first string $s_1 = x_1x_2$ where $x_1, x_2 \in \{a, b, c\}$

Let the second string $s_2 = y_1y_2$ where $y_1, y_2 \in \{a, b, c\}$

We know $s_1 \neq s_2$, therefore $x_1 \neq y_1$ or $x_2 \neq y_2$

Since s_1 and s_2 drive the machine to the same state,

for any string x , s_1x, s_2x should end at same state.

Choose $x = x_2x_1$.

Then $s_1x = x_1x_2x_2x_1$.

Since $s_1x = (s_1x)^R$, $|s_1x| = 4$, it is accepted.

On the other hand, $s_2x = y_1y_2x_2x_1$.

Since $x_1 \neq y_1$ or $x_2 \neq y_2$,

we know $s_2x \neq (s_2x)^R$

Therefore s_2x is rejected.

We get the contradiction since s_1x and s_2x do not end at same state.

Therefore the assumption is false which implies that it is impossible that two distinct length-2 prefixes of strings in L_{R4} drive the machine to the same state.

Then let's calculate the number of distinct length-2 strings.

Since $\sigma = \{a, b, c\}$, all the combination is $3 \times 3 = 9$.

Therefore all these 9 strings do not end at same state which implies that there are at least 9 states.

Generalize my result:

No DFA can accept L_R .

First I want to show $\forall k \in \mathbb{N}, k \geq 4 \Rightarrow$ DFA that accepts $L_{Rk} = \{x \in \Sigma^* \mid |x| = k \wedge x = x^R\}$ has at least $3^{\lfloor \frac{k}{2} \rfloor}$ states.

Let $k \in \mathbb{N}$. Assume $k \geq 4$. let's consider two cases:

Case 1: k is even

First I want to show it is impossible that two distinct length- $\frac{k}{2}$ prefixes of strings in L_{Rk} drive the machine to the same state.

I will prove it by using contradiction.

Suppose there exists two distinct length- $\frac{k}{2}$ prefixes of strings in L_{Rk} drive the machine to the same state.

Then let the first string $s_1 = m_1m_2 \dots m_{\frac{k}{2}}$ where $m_1, m_2, \dots, m_{\frac{k}{2}} \in \{a, b, c\}$

Let the second string $s_2 = n_1n_2 \dots n_{\frac{k}{2}}$ where $n_1, n_2, \dots, n_{\frac{k}{2}} \in \{a, b, c\}$

We know $s_1 \neq s_2$, therefore $\exists i \in \mathbb{N}$ such that $m_i \neq n_i$ and $1 \leq i \leq \frac{k}{2}$.

Since s_1 and s_2 drive the machine to the same state,
for any string x , s_1x, s_2x should end at same state.

Choose $x = m_{\frac{k}{2}}m_{\frac{k}{2}-1} \cdots m_1$.

Then $s_1x = m_1m_2 \cdots m_{\frac{k}{2}}m_{\frac{k}{2}-1} \cdots m_1$.

Since $s_1x = (s_1x)^R$, it is accepted.

On the other hand, $s_2x = n_1n_2 \cdots n_{\frac{k}{2}}m_{\frac{k}{2}}m_{\frac{k}{2}-1} \cdots m_1$.

Since $\exists i \in \mathbb{N}$ such that $m_i \neq n_i$ and $1 \leq i \leq \frac{k}{2}$,

we know $s_2x \neq (s_2x)^R$

Therefore s_2x is rejected.

We get the contradiction since s_1x and s_2x do not end at same state.

Therefore the assumption is false which implies that it is impossible that two distinct length- $\frac{k}{2}$ prefixes of strings in L_{Rk} drive the machine to the same state.

Then let's calculate the number of distinct length- $\frac{k}{2}$ strings.

Since $\Sigma = \{a, b, c\}$, all the combination is $3^{\frac{k}{2}} = 3^{\lfloor \frac{k}{2} \rfloor}$ since k is even.

Therefore all these $3^{\lfloor \frac{k}{2} \rfloor}$ strings do not end at same state which implies that there are at least $3^{\lfloor \frac{k}{2} \rfloor}$ states.

Case 2: k is odd

First I want to show it is impossible that two distinct length- $(\frac{k-1}{2})$ prefixes of strings in L_{Rk} drive the machine to the same state.

I will prove it by using contradiction.

Suppose there exists two distinct length- $(\frac{k-1}{2})$ prefixes of strings in L_{Rk} drive the machine to the same state.

Then let the first string $s_1 = m_1m_2 \cdots m_{\frac{k-1}{2}}$ where $m_1, m_2, \dots, m_{\frac{k-1}{2}} \in \{a, b, c\}$

Let the second string $s_2 = n_1n_2 \cdots n_{\frac{k-1}{2}}$ where $n_1, n_2, \dots, n_{\frac{k-1}{2}} \in \{a, b, c\}$

We know $s_1 \neq s_2$, therefore $\exists i \in \mathbb{N}$ such that $m_i \neq n_i$ and $1 \leq i \leq \frac{k-1}{2}$.

Since s_1 and s_2 drive the machine to the same state,

for any string x , s_1x, s_2x should end at same state.

Choose $x = m_{\frac{k+1}{2}}m_{\frac{k-1}{2}}m_{\frac{k-3}{2}} \cdots m_1$ where $m_{\frac{k+1}{2}} \in \{a, b, c\}$.

Then $s_1x = m_1m_2 \cdots m_{\frac{k-1}{2}}m_{\frac{k+1}{2}}m_{\frac{k-1}{2}} \cdots m_2m_1$.

Since $s_1x = (s_1x)^R$, it is accepted.

On the other hand, $s_2x = n_1n_2 \cdots n_{\frac{k-1}{2}}m_{\frac{k+1}{2}}m_{\frac{k-1}{2}} \cdots m_2m_1$.

Since $\exists i \in \mathbb{N}$ such that $m_i \neq n_i$ and $1 \leq i \leq \frac{k-1}{2}$,

we know $s_2x \neq (s_2x)^R$

Therefore s_2x is rejected.

We get the contradiction since s_1x and s_2x do not end at same state.

Therefore the assumption is false which implies that it is impossible that two distinct length- $(\frac{k-1}{2})$ prefixes of strings in L_{Rk} drive the machine to the same state.

Then let's calculate the number of distinct length- $(\frac{k-1}{2})$ strings.

Since $\Sigma = \{a, b, c\}$, all the combination is $3^{\frac{k-1}{2}} = 3^{\lfloor \frac{k}{2} \rfloor}$ since k is odd.

Therefore all these $3^{\lfloor \frac{k}{2} \rfloor}$ strings do not end at same state which implies that there are at least

$3^{\lfloor \frac{k}{2} \rfloor}$ states.

By case 1 and case 2, I can conclude that $\forall k \in \mathbb{N}, k \geq 4 \Rightarrow$ DFA that accepts $L_{Rk} = \{x \in \Sigma^* \mid |x| = k \wedge x = x^R\}$ has at least $3^{\lfloor \frac{k}{2} \rfloor}$ states.

Since k which denotes the length of strings in L_{Rk} can be infinitely large,

when $k \rightarrow \infty, 3^{\lfloor \frac{k}{2} \rfloor} \rightarrow \infty$

then if there exists DFA that accepts L_R , it should contains infinite states.

We know that DFA can never contain infinite state, therefore we can conclude that no DFA can accept L_R .