First Name: Zhihong
Last Name: Wang
Student ID: 1002095207

First Name: Jialiang
Last Name: Yi
Student ID: 1002286929

We declare that this assignment is solely our own work, and is in accordance with the University of Toronto Code of Behaviour on Academic Matters.

This submission has been prepared using LaTeX.

1. Proof of Correctness for iterative algorithms.

   (a) Design an iterative closest pair algorithm for finding the closest pair of points in 2D.

   Assume dis(points[0], points[1]) can get the distance between points[0] and points[1].

   Precondition: Input is a list of n points in the form (x_i, y_i), where $x_i, y_i \in \mathbb{R}$
   Postcondition: Return a closest pair of points, which is equal to the pair(a,b) in $\bigcup_{i=0}^{n-2} \bigcup_{j=i+1}^{n-1} \{(points[i], points[j]), closestpair\}$ which makes the distance minimum

```
def closestPair(points):
    n = len(points)
    distance = dis(points[0], points[1])
    closestpair = (points[0], points[1])
    i = 0
    while i < n:
        closestpair = helper(points, i, distance, closestpair)
        i += 1
    return closestpair
```

   Precondition: Input are a list of n points in the form (x_i, y_i), where $x_i, y_i \in \mathbb{R}$; the index i of the list of points, where $i \in \mathbb{N}$ and $0 \leq i \leq n-2$; the distance of points[0] and points[1], where distance $\in \mathbb{R}$; the points pair of points[0] and points[1]
   Postcondition: Return the closest pair of points in { (points[i] , the closest point to points[i]) , the input points pair }, which is equal to the pair(a,b) in $\bigcup_{j=i+1}^{n-1} \{(points[i], points[j]), closestpair\}$ which makes the distance minimum

```
def helper(points, i, distance, closestpair):
    j = i + 1
    n = len(points)
    while i < j < n:
        temp = dis(points[i], points[j])
        if temp < distance and i != j:
            distance = temp
            closestpair = (points[i], points[j])
        j += 1
    return closestpair
```

   (b) Find complexity class

Let n be the size of the list of points.
Compute the worse-case time complexity $T(n)$.
By helper's line 3, the worst-case time complexity is $n - i$ because it is a while loop depends on the beginning index i and the size of the list of points, which is $n - i$.
By closestPair's line 5, the worst-case time complexity is n because it is a while loop depends on the size of the list of points, which is n.
Therefore, $T(n) \in \theta(n^2)$.

(c) Prove correctness:

i. Define Loop Invariant
ii. Prove Partial Correctness
iii. Termination (use either theorem 2.5 in the notes or POW)

Before Formal Proof:

Lemma: helper is correct

Prove Lemma:

1. Define Loop Invariant

$LI(k)$: If this loop excuted at least k times,
then (1). $j_k = i + 1 + k$
(2). $distance_k = \min \bigcup_{j=i+1}^{i+k} \{dis(points[i], points[j]), distance\}$
(3). $closestpair_k =$ the pair(a,b) in $\bigcup_{j=i+1}^{i+k} \{(points[i], points[j]), closestpair\}$ which makes the distance minimum
(4). $i + 1 \leq j_k \leq n$

Prove $\forall k \in \mathbb{N}, LI(k)$
prove by simple induction:

Base Case:
Before enter the loop, $k = 0$
then $j_k = j_0 = i + 1$
then $i + 1 \leq j_k \leq n$
then $LI(0)$

I.S: Let $k \in \mathbb{N}$
I.H: Assume $LI(k)$
WTP: $LI(k + 1)$
Assume loop executes at least $k + 1$ times

Case 1: dis(points[i], points[i+1+k]) $< distance_k$
$j_{k+1} = j_k + 1$ # By Line 8
$= i + 1 + k + 1$ # By I.H.

By I.H, $distance_k = \min \bigcup_{j=i+1}^{i+k} \{dis(points[i], points[j]), distance\}$
By Line 6, $distance_{k+1}$ = dis(points[i], points[i+1+k])
$= \min \bigcup_{j=i+1}^{i+k+1} \{dis(points[i], points[j]), distance\}$
By I.H, $closestpair_k =$ the pair(a,b) in $\bigcup_{j=i+1}^{i+k} \{(points[i], points[j]), closestpair\}$
which makes the distance minimum
By Line 7, $closestpair_{k+1}$ = (points[i], points[i+1+k])
= the pair(a,b) in $\bigcup_{j=i+1}^{i+k+1} \{(points[i], points[j]), closestpair\}$
which makes the distance minimum

Case 2: dis(points[i], points[i+1+k]) $\geq distance_k$
$j_{k+1} = j_k + 1$ # By Line 8
$= i + 1 + k + 1$ # By I.H.
By I.H, $distance_k = \min \bigcup_{j=i+1}^{i+k} \{dis(points[i], points[j]), distance\}$
then, $distance_{k+1} = distance_k$
By I.H, $closestpair_k =$ the pair(a,b) in $\bigcup_{j=i+1}^{i+k} \{(points[i], points[j]), closestpair\}$
which makes the distance minimum
then, $closestpair_{k+1} = closestpair_k$

then $i + 1 \leq j_{k+1} \leq n$
then $LI(k + 1)$

Therefore $\forall k \in \mathbb{N}, LI(k)$

2. Prove Partial Correctness (pre and $term \rightarrow post$)
Assume Precondition and Termination
Assume this loop terminates after t times
By $LI(t), t = n - i - 1$
$j_t = i + 1 + t, i + 1 \leq j_t \leq n$
$distance_t = \min \bigcup_{j=i+1}^{i+t} \{dis(points[i], points[j]), distance\}$
$= \min \bigcup_{j=i+1}^{n-1} \{dis(points[i], points[j]), distance\}$
$closestpair_t =$ the pair(a,b) in $\bigcup_{j=i+1}^{i+t} \{(points[i], points[j]), closestpair\}$
which makes the distance minimum
= the pair(a,b) in $\bigcup_{j=i+1}^{n-1} \{(points[i], points[j]), closestpair\}$
which makes the distance minimum
then Postcondition

3. Prove Termination ($pre \rightarrow term$)
Assume Precondition
Let $k \in \mathbb{N}$ be the times of loops
while $j < n, j+ = 1$
Let $m_k = n - j_k$
then $m_k \geq 0$ # By LI, $i + 1 \leq j_k \leq n$
then $m_{k+1} = n - j_{k+1}$
$= n - j_k - 1$

$= m_k - 1$
$< m_k$
then m_k is decreasing
then Termination

Therefore, the helper is correct.

Formal Proof:

1. Define Loop Invariant
$LI(k)$: If this loop excuted at least k times,
then (1). $i = k$
(2). $closestpair_k =$
the pair(a,b) in $\bigcup_{i=0}^{k-1} \bigcup_{j=i+1}^{n-1} \{(points[i], points[j]), closestpair\}$
which makes the distance minimum
(3). $0 \leq i_k \leq n$

Prove $\forall k \in \mathbb{N}, LI(k)$
prove by simple induction:

Base Case:
Before enter the loop, $k = 0$
then $i_k = i_0 = 0$
then $0 \leq i_k \leq n$
then $LI(0)$

I.S: Let $k \in \mathbb{N}$
I.H: Assume $LI(k)$
WTP: $LI(k+1)$
Assume loop executes at least $k + 1$ times
then $i_{k+1} = i_k + 1$ # By Line 7
$= k + 1$ # By I.H.
By I.H, $closestpair_k =$ the pair(a,b) in $\bigcup_{i=0}^{k-1} \bigcup_{j=i+1}^{n-1} \{(points[i], points[j]), closestpair\}$
which makes the distance minimum
By line 6, $closestpair_{k+1} = helper(points, i, distance, closestpair)$
By Lemma, helper is correct and it returns a pair(a,b)
in $\bigcup_{j=i+1}^{n-1} \{(points[i], points[j]), closestpair\}$
which makes the distance minimum
$=$ the pair(a,b) in $\bigcup_{i=0}^{k} \bigcup_{j=i+1}^{n-1} \{(points[i], points[j]), closestpair\}$
which makes the distance minimum
then $0 \leq i_{k+1} \leq n$
then $LI(k+1)$
Therefore, $\forall k \in \mathbb{N}, LI(k)$

2. Prove Partial Correctness (pre and $term \rightarrow post$)

Assume Precondition and Termination
Assume this loop terminates after t times
By $LI(t), t = n - 1$
$i_t = t, 0 \leq i_t \leq n$

$$\begin{aligned}
closestpair_t &= \text{the pair(a,b) in } \bigcup_{i=0}^{t-1} \bigcup_{j=i+1}^{n-1} \{(points[i], points[j]), closestpair\} \\
&\quad \text{which makes the distance minimum} \\
&= \text{the pair(a,b) in } \bigcup_{i=0}^{n-2} \bigcup_{j=i+1}^{n-1} \{(points[i], points[j]), closestpair\} \\
&\quad \text{which makes the distance minimum}
\end{aligned}$$

then Postcondition

3. Prove Termination ($pre \to term$)
Assume Precondition
Let $k \in \mathbb{N}$ be the times of loops
while $i < n, i+ = 1$
Let $m_k = n - i_k$
then $m_k \geq 0$ # By LI, $0 \leq i_k \leq n$

$$\begin{aligned}
\text{then } m_{k+1} &= n - i_{k+1} \\
&= n - i_k - 1 \\
&= m_k - 1 \\
&< m_k
\end{aligned}$$

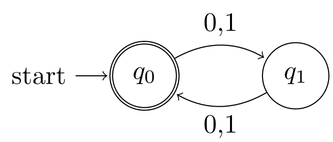
then m_k is decreasing
then Termination

Therefore, the function closestPair is correct.

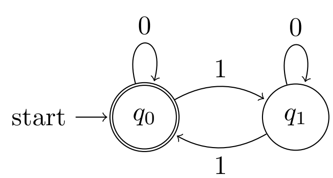
2. DFSAs and their operations

   (a) Define and draw DFSAs on binary alphabet $\Sigma = \{0, 1\}$ for 2 languages: $L_1(M_1) = \{$all strings with even number of characters in a string$\}$, $L_2(M_2) = \{$all strings that have even number of 1s$\}$

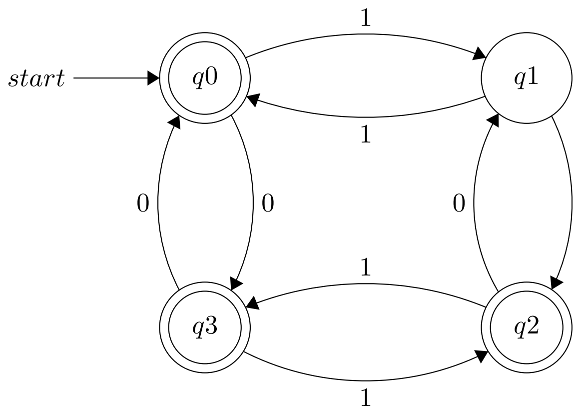
   $L_1(M_1) = \{$all strings with even number of characters in a string$\}$
   $W_1 = (Q_1, \Sigma_1, \delta_1, q_0, F_1)$
   $Q_1 = \{q_0, q_1\}$
   $\Sigma_1 = \{0, 1\}$
   $\delta_1(q_0, 0) = q_1, \delta_1(q_0, 1) = q_1$
   $\delta_1(q_1, 0) = q_0, \delta_1(q_1, 1) = q_0$
   $F_1 = \{q_0\}$



   $L_2(M_2) = \{$all strings that have even number of 1s$\}$
   $W_2 = (Q_2, \Sigma_2, \delta_2, q_0, F_2)$
   $Q_2 = \{q_0, q_1\}$
   $\Sigma_2 = \{0, 1\}$
   $\delta_2(q_0, 0) = q_0, \delta_2(q_0, 1) = q_1$
   $\delta_2(q_1, 0) = q_1, \delta_2(q_1, 1) = q_0$
   $F_2 = \{q_0\}$

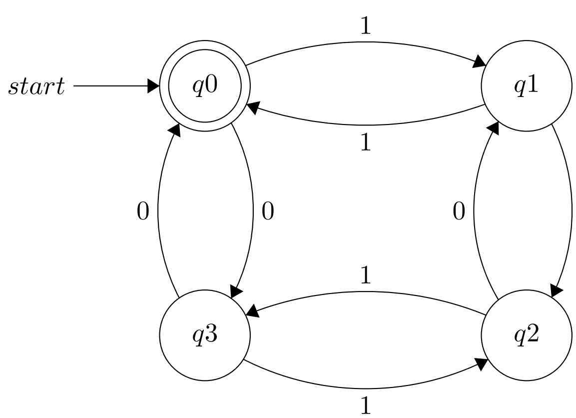


   (b) Identify DFSA M_3 for the union of languages $L_1 \cup L_2$ - you can define it formally (don't need to draw).



$L_3(M_3) = L_1 \cup L_2 = \{$all strings with even number of characters or even number of 1s$\}$
$W = (Q, \Sigma, \delta, q_0, F)$
$Q = \{q_0, q_1, q_2, q_3\}$
$\Sigma = \{0, 1\}$
$\delta(q_0, 0) = q_3, \delta(q_0, 1) = q_1$
$\delta(q_1, 0) = q_2, \delta(q_1, 1) = q_0$
$\delta(q_2, 0) = q_1, \delta(q_2, 1) = q_3$
$\delta(q_3, 0) = q_0, \delta(q_3, 1) = q_2$
$F = \{q_0, q_2, q_3\}$

(c) Identify DFSA M_4 for the intersection of languages $L_1 \cap L_2$ - you can define it formally (don't need to draw).



$L_4(M_4) = L_1 \cap L_2 = \{$all strings with even number of characters and even number of 1s$\}$
$W = (Q, \Sigma, \delta, q_0, F)$
$Q = \{q_0, q_1, q_2, q_3\}$
$\Sigma = \{0, 1\}$
$\delta(q_0, 0) = q_3, \delta(q_0, 1) = q_1$

$\delta(q_1, 0) = q_2, \delta(q_1, 1) = q_0$
$\delta(q_2, 0) = q_1, \delta(q_2, 1) = q_3$
$\delta(q_3, 0) = q_0, \delta(q_3, 1) = q_2$
$F = \{q_0\}$

(d) Find and prove a state invariant for M_3.

State Invariant: $\forall w \in \Sigma^*$

$$\delta^*(q_0, w) = \begin{cases} q_0 & \text{iff } |\ w\ | \text{ is even and } w \text{ has even number of 1s} \\ q_1 & \text{iff } |\ w\ | \text{ is odd and } w \text{ has odd number of 1s} \\ q_2 & \text{iff } |\ w\ | \text{ is even and } w \text{ has odd number of 1s} \\ q_3 & \text{iff } |\ w\ | \text{ is odd and } w \text{ has even number of 1s} \end{cases}$$

Let us first prove the forward direction of the statement (i.e., if condition, then q_0). After that, we will show that the backward direction (i.e., if q_0, then condition) can easily be derived based on the fact that the state invariants are exhaustive.

Proof: $\forall w \in \Sigma^*$

$$P(w) : \delta^*(q_0, w) = \begin{cases} q_0 & \text{if } |\ w\ | \text{ is even and } w \text{ has even number of 1s} \\ q_1 & \text{if } |\ w\ | \text{ is odd and } w \text{ has odd number of 1s} \\ q_2 & \text{if } |\ w\ | \text{ is even and } w \text{ has odd number of 1s} \\ q_3 & \text{if } |\ w\ | \text{ is odd and } w \text{ has even number of 1s} \end{cases}$$

prove by structural induction:

Base Case:
when $w = \varepsilon, |\ w\ |$ is even and w has even number of 1s
then $\delta^*(q_0, w) = q_0$
then $P(w)$

I.S: Let $w = xa,$ for $x \in \Sigma^*, a \in \Sigma$
I.H: Assume $P(x)$
WTP: $P(xa)$ holds # $P(w)$ holds

Case 1: $a = 0$

1.1: $|\ x\ |$ is even and x has even number of 1s
$\delta^*(q_0, x) = q_0$ # By I.H.
$\delta^*(q_0, xa) = \delta(\delta^*(q_0, x), a)$ # By Extended Transition Function
$= \delta(q_0, 0)$
$= q_3$

then $P(x0)$

1.2: $| x |$ is odd and x has odd number of 1s

$\delta^*(q_0, x) = q_1$ # By I.H.

$\delta^*(q_0, xa) = \delta(\delta^*(q_0, x), a)$ # By Extended Transition Function

$= \delta(q_1, 0)$

$= q_2$

then $P(x0)$

1.3: $| x |$ is even and x has odd number of 1s

$\delta^*(q_0, x) = q_2$ # By I.H.

$\delta^*(q_0, xa) = \delta(\delta^*(q_0, x), a)$ # By Extended Transition Function

$= \delta(q_2, 0)$

$= q_1$

then $P(x0)$

1.4: $| x |$ is odd and x has even number of 1s

$\delta^*(q_0, x) = q_3$ # By I.H.

$\delta^*(q_0, xa) = \delta(\delta^*(q_0, x), a)$ # By Extended Transition Function

$= \delta(q_3, 0)$

$= q_0$

then $P(x0)$

Therefore $P(x0)$

Case 2: $a = 1$

2.1: $| x |$ is even and x has even number of 1s

$\delta^*(q_0, x) = q_0$ # By I.H.

$\delta^*(q_0, xa) = \delta(\delta^*(q_0, x), a)$ # By Extended Transition Function

$= \delta(q_0, 1)$

$= q_1$

then $P(x1)$

2.2: $| x |$ is odd and x has odd number of 1s

$\delta^*(q_0, x) = q_1$ # By I.H.

$\delta^*(q_0, xa) = \delta(\delta^*(q_0, x), a)$ # By Extended Transition Function

$= \delta(q_1, 1)$

$= q_0$

then $P(x1)$

2.3: $| x |$ is even and x has odd number of 1s

$\delta^*(q_0, x) = q_2$ # By I.H.

$\delta^*(q_0, xa) = \delta(\delta^*(q_0, x), a)$ # By Extended Transition Function

$= \delta(q_2, 1)$

$= q_3$

then $P(x1)$

2.4: $\mid x \mid$ is odd and x has even number of 1s

$\delta^*(q_0, x) = q_3$ # By I.H.

$\delta^*(q_0, xa) = \delta(\delta^*(q_0, x), a)$ # By Extended Transition Function

$= \delta(q_3, 1)$

$= q_2$

then $P(x1)$

Therefore $P(x1)$

Therefore $P(xa)$

Therefore $\forall w \in \Sigma^*, P(w)$ holds

$$P(w) : \delta^*(q_0, w) = \begin{cases} q_0 & \text{if } \mid w \mid \text{ is even and } w \text{ has even number of 1s} \\ q_1 & \text{if } \mid w \mid \text{ is odd and } w \text{ has odd number of 1s} \\ q_2 & \text{if } \mid w \mid \text{ is even and } w \text{ has odd number of 1s} \\ q_3 & \text{if } \mid w \mid \text{ is odd and } w \text{ has even number of 1s} \end{cases}$$

Because state invariants are exhaustive, all conditions after q_0, q_1, q_2, q_3 covered all situations in this question,

1. Prove $\delta^*(q_0, w) = q_0 \rightarrow \mid w \mid$ is even and w has even number of 1s
Prove by contrapositive:
Prove $\neg(\mid w \mid$ is even and w has even number of 1s$) \rightarrow \delta^*(q_0, w) \neq q_0$
Assume $\neg(\mid w \mid$ is even and w has even number of 1s$)$
then $\mid w \mid$ is odd or w has odd number of 1s
then $\delta^*(q_0, w) = q_1$ or q_2 or $q_3 \neq q_0$ # By definition of $\delta^*(q_0, w)$
Therefore $\neg(\mid w \mid$ is even and w has even number of 1s$) \rightarrow \delta^*(q_0, w) \neq q_0$
Therefore $\delta^*(q_0, w) = q_0 \rightarrow \mid w \mid$ is even and w has even number of 1s

2. Prove $\delta^*(q_0, w) = q_1 \rightarrow \mid w \mid$ is odd and w has odd number of 1s
Prove by contrapositive:
Prove $\neg(\mid w \mid$ is odd and w has odd number of 1s$) \rightarrow \delta^*(q_0, w) \neq q_1$
Assume $\neg(\mid w \mid$ is odd and w has odd number of 1s$)$
then $\mid w \mid$ is even or w has even number of 1s
then $\delta^*(q_0, w) = q_0$ or q_2 or $q_3 \neq q_1$ # By definition of $\delta^*(q_0, w)$
Therefore $\neg(\mid w \mid$ is odd and w has odd number of 1s$) \rightarrow \delta^*(q_0, w) \neq q_1$
Therefore $\delta^*(q_0, w) = q_1 \rightarrow \mid w \mid$ is odd and w has odd number of 1s

3. Prove $\delta^*(q_0, w) = q_2 \rightarrow \mid w \mid$ is even and w has odd number of 1s
Prove by contrapositive:
Prove $\neg(\mid w \mid$ is even and w has odd number of 1s$) \rightarrow \delta^*(q_0, w) \neq q_2$
Assume $\neg(\mid w \mid$ is even and w has odd number of 1s$)$
then $\mid w \mid$ is odd or w has even number of 1s
then $\delta^*(q_0, w) = q_0$ or q_1 or $q_3 \neq q_2$ # By definition of $\delta^*(q_0, w)$
Therefore $\neg(\mid w \mid$ is even and w has odd number of 1s$) \rightarrow \delta^*(q_0, w) \neq q_2$
Therefore $\delta^*(q_0, w) = q_2 \rightarrow \mid w \mid$ is even and w has odd number of 1s

4. Prove $\delta^*(q_0, w) = q_3 \rightarrow \mid w \mid$ is odd and w has even number of 1s
Prove by contrapositive:
Prove $\neg(\mid w \mid$ is odd and w has even number of 1s$) \rightarrow \delta^*(q_0, w) \neq q_3$
Assume $\neg(\mid w \mid$ is odd and w has even number of 1s$)$
then $\mid w \mid$ is even or w has odd number of 1s
then $\delta^*(q_0, w) = q_0$ or q_1 or $q_2 \neq q_3$ # By definition of $\delta^*(q_0, w)$
Therefore $\neg(\mid w \mid$ is odd and w has even number of 1s$) \rightarrow \delta^*(q_0, w) \neq q_3$
Therefore $\delta^*(q_0, w) = q_3 \rightarrow \mid w \mid$ is odd and w has even number of 1s

Therefore: $\forall w \in \Sigma^*$

$$\delta^*(q_0, w) = \begin{cases} q_0 & \text{iff } \mid w \mid \text{ is even and } w \text{ has even number of 1s} \\ q_1 & \text{iff } \mid w \mid \text{ is odd and } w \text{ has odd number of 1s} \\ q_2 & \text{iff } \mid w \mid \text{ is even and } w \text{ has odd number of 1s} \\ q_3 & \text{iff } \mid w \mid \text{ is odd and } w \text{ has even number of 1s} \end{cases}$$

3. Equivalence of languages and regular expressions
   Language L over alphabet $\Sigma = \{a, b\}$ consists of all strings that start with a and have odd lengths or start with b and have even lengths: $\{s|s$ starts with a and has odd length, or starts with b and has even length$\}$.

   (a) What is a regular expression R corresponding to language L?

   $R = a(aa + ab + ba + bb)^* + b(aa + ab + ba + bb)^*(a + b)$

   (b) Prove that your regular expression R is indeed equivalent to L

   Before Formal Proof:

   Lemma: $\forall x \in \Sigma^*, \mid x \mid$ is even $\rightarrow x \in L(aa + ab + ba + bb)^*$

   Prove Lemma:
   $P(n) = \forall x \in \Sigma^*, \mid x \mid = n$ is even $\rightarrow x \in L(aa + ab + ba + bb)^*$
   prove by complete induction:

   Base Case:
   when $n = 0 = \mid x \mid, x \in L(aa + ab + ba + bb)^*$
   then $P(0)$

   I.S: Let $x \in \Sigma^*, \mid x \mid = n$ is even
   I.H: Assume $\forall 0 \leq i \leq n, i$ is even, $P(i)$

   Case 1: when $n = 0, P(0)$ $\sharp$ By Base Case

   Case 2: when $n \geq 2$,
   then $n - 2 \in i$
   Let $x = yaa$ or yab or yba or ybb
   then $\mid y \mid = n - 2$
   then $\mid y \mid$ is even
   then $y \in L(aa + ab + ba + bb)^*$ $\sharp$ By I.H.
   then $\exists k \in \mathbb{N}, y \in L(aa + ab + ba + bb)^k$
   Because $aa, ab, ba, bb \in L(aa + ab + ba + bb)$
   then $y(aa + ab + ba + bb) \in L(aa + ab + ba + bb)^k L(aa + ab + ba + bb)$
   $= L(aa + ab + ba + bb)^{k+1}$
   $\subseteq L(aa + ab + ba + bb)^*$
   then $x \in L(aa + ab + ba + bb)^*$
   Therefore $\forall x \in \Sigma^*, \mid x \mid$ is even $\rightarrow x \in L(aa + ab + ba + bb)^*$

   Formal Proof: prove that your regular expression R is indeed equivalent to L

Prove $L = L(R) \leftrightarrow$ prove $L \subseteq L(R)$ and $L(R) \subseteq L$

1. Prove $L \subseteq L(R) \leftrightarrow$ Prove $\forall w \in L, w \in L(R)$
Let $w \in L$
Let $\forall x \in \Sigma^*, \mid x \mid=$ even, $w = ax$ or $bx(a+b)$
then $x \in L(aa+ab+ba+bb)^*$ $\sharp$ By Lemma

Case 1: $\forall x \in \Sigma^*, w = ax$
WTP: $w \in L(R)$
Because $a \in L(a), x \in L(x) = L(aa+ab+ba+bb)^*$ $\sharp$ By Lemma
then $ax \in L(a) \circ L(aa+ab+ba+bb)^*$
then $ax \in L(a(aa+ab+ba+bb)^*)$
then $w \in L(a(aa+ab+ba+bb)^*)$
then $w \in L(R)$

Case 2: $\forall x \in \Sigma^*, c \in \Sigma, w = bxc$
WTP: $w \in L(R)$
Because $b \in L(b), c \in L(a+b), x \in L(x) = L(aa+ab+ba+bb)^*$ $\sharp$ By Lemma
then $bxc \in L(b) \circ L(aa+ab+ba+bb)^* \circ L(a+b)$
then $bxc \in L(b(aa+ab+ba+bb)^*(a+b))$
then $w \in L(b(aa+ab+ba+bb)^*(a+b))$
then $w \in L(R)$

Therefore $L \subseteq L(R)$

2. Prove $L(R) \subseteq L \leftrightarrow$ Prove $\forall w \in L(R), w \in L$
Let $w \in L(R) = L(a(aa+ab+ba+bb)^*) \cup L(b(aa+ab+ba+bb)^*(a+b))$

Case 1: $w \in L(a(aa+ab+ba+bb)^*)$
$= L(a) \circ L(aa+ab+ba+bb)^*$
WTP: $w \in L$
Let $w = ax, a \in L(a), x \in L(aa+ab+ba+bb)^*$
then $\exists k \in \mathbb{N}, x \in L(aa+ab+ba+bb)^k$
then $x \in L(aa+ab+ba+bb)^k$
$= (L(aa) \cup L(ab) \cup L(ba) \cup L(bb))^k$
$= \{aa, ab, ba, bb\}^k$
then $\mid x \mid= 2k$
then $\mid w \mid=\mid ax \mid= 2k+1, \mid w \mid$ is odd
then w starts with a and has odd length
then $w \in L$

Case 2: $w \in L(b(aa+ab+ba+bb)^*(a+b))$
$= L(b) \circ L(aa+ab+ba+bb)^* \circ L(a+b)$
$= L(b) \circ L(aa+ab+ba+bb)^* \circ (L(a) \cup L(b))$

WTP: $w \in L$
Let $w = bxc, b \in L(b), x \in L(aa + ab + ba + bb)^*, c \in L(a + b)$
then $\exists k \in \mathbb{N}, x \in L(aa + ab + ba + bb)^k$
then $x \in L(aa + ab + ba + bb)^k$
$\quad = (L(aa) \cup L(ab) \cup L(ba) \cup L(bb))^k$
$\quad = \{aa, ab, ba, bb\}^k$
then $\mid x \mid = 2k$
then $\mid w \mid = \mid bxc \mid = 2k + 2, \mid w \mid$ is even
then w starts with b and has even length
then $w \in L$

Therefore $L(R) \subseteq L$

Therefore $L \subseteq L(R)$ and $L(R) \subseteq L$
Therefore $L = L(R)$