Define: A palindrome is a string that is the same as itself when read front-to-back or back-to-front.
Define: L1 be the set of all palindromes over \{0, 1\}.
Define: L2 be the set of strings over \{0, 1\} defined recursively:
1. \(\varepsilon, 0,\) and 1 are in L2.
2. If s is in L2, then 0s0 is in L2.
3. If s is in L2, then 1s1 is in L2.

To prove: L1 and L2 are equal.

To prove this equality, I will show that L1 is a subset of L2, and L2 is a subset of L1.

L1 is a subset of L2

To prove this, we will show by induction on the lengths of elements of L1 that all elements in L1 are also in L2.

Define P(n) as the subset of L1 such that the length of each element is \(n\).

Base Cases: We must show that P(0) and P(1) are subsets of L2. The only element of length zero is the empty string \(\varepsilon\). By the first recursive rule of L2, \(\varepsilon\) is in L2. Thus P(0) is a subset of L2. The only elements of length one are the strings “1” and “0”. By the first recursive rule of L2, 1 and 0 are in L2. Thus P(1) is a subset of L2.

Inductive Step:
We must show that P(n) in L2 \(\rightarrow\) P(n+2) in L2 for all \(n \geq 0\).

Assume that P(n) is a subset of L2 (inductive hypothesis).

To prove: P(n+2) is a subset of L2

All palindromes of length \(n\) can be used to create all of the palindromes of length \((n+2)\) by adding the same character to both the front and end of the length \(n\) palindromes, over all possible characters. In this way, the first and last characters match, and internally the string is still a palindrome. This is accomplished by taking all elements in P(n), and placing an element of \{0,1\} at the beginning and end of the string over all possible elements. Thus, P(n+2) = \([0\ P(n)\ 0] \cup [1\ P(n)\ 1]\). Since P(n) is a subset of L2, by the second and third recursive rules for L2, \([0\ P(n)\ 0]\) and \([1\ P(n)\ 1]\) are in L2. The union of two subsets is still a subset. Therefore, P(n+2) is a subset of L2.

We’ve shown that for all \(n \geq 0\), P(n) is a subset of L2. P(n) for \(n \geq 0\) is equal to L1 because it is defined as the set of strings of all lengths up through infinity that belong to L1. Thus, L1 is a subset of L2.
L2 is a subset of L1.

To prove this, we will show by induction on recursive rules for elements of L2 that all elements in L2 are also in L1.

Define P(n) as the subset of L2 such that the elements are built after running n steps from the recursive rules.

Base Cases: We must show that P(1) is a subset of L1. The only elements that are obtained in one step are ε, 0, and 1. All of these are in L1, that is, they are all palindromes by the definition of a palindrome. Thus P(1) is a subset of L1.

Inductive Step:
We must show that P(n) in L1 -> P(n+1) in L1 for all n >= 1.
Assume that P(n) is a subset of L1 (inductive hypothesis).

To prove: P(n+1) is a subset of L1
At each step of the recursive rules after the first step, the next batch of elements for L2 are obtained by sandwiching the current strings between 0, and between 1, and taking the union of the two. This is like saying that P(n+1) = [0 P(n) 0] U [1 P(n) 1]. P(n) is in L1, that is, it is a palindrome. After these two operations on P(n), the first and last characters match, and internally the string is still a palindrome because of our assumption. Thus each yields a subset of L1. The union of two subsets is still a subset. Therefore, P(n+1) is a subset of L1.

We've shown that for all n >= 1, P(n) is a subset of L1. P(n) for n >= 1 is equal to L2 because it is defined as the set of elements of L2 after the recursive rules have been run any whole number of times up to infinity. Thus, L2 is a subset of L1.

Conclusion
We've shown that L1 is a subset of L2 and L2 is a subset of L1, therefore L1=L2, QED.