In the first homework assignment (problem 4), you were given five languages \( L_1, \ldots, L_5 \), two of which \( (L_1 \text{ and } L_5) \) were regular, and the remaining three were not. You can find the proof that \( L_2 \) is not regular in the textbook (at the end of chapter 1). In this problem set you will show that the remaining languages are not regular.

**Problem 1**

Prove, using the pumping lemma, that

\[
L_4 = \{ w \in \Sigma^* : w \text{ contains more 0's than 1's} \}
\]

is not a regular language.

**Solution Problem 1**

The proof is by contradiction. Assume \( L_4 \) is regular. Then by the Pumping Lemma, there is an integer \( p > 0 \), such that for every \( w \in L_4, |w| \geq p \), there exists a partition of \( w = xyz \), such that \( |y| > 0, |xy| \leq p \), and for all \( i \ xy^iz \in L_4 \).

Let \( w = 0^{p+1}1^p \) and let \( x, y, z \) strings satisfying the conditions of the pumping lemma. Since \( |xy| \leq 0 \) then \( y = 0^k \) for some \( k \geq 1 \). By the Pumping Lemma \( w_1 = xy^iz = x(0^k)^iz \) belongs to \( L_4 \) for any value of \( i \). But setting \( i = 0 \) makes the number of 0’s in \( w_1 \) always equal or less than the number of 1s and \( w_1 \notin L_4 \). Thus when \( i = 0 \), \( w_1 \) does not belong to \( L_4 \), contradicting the pumping lemma.

**Problem 2**

There are two possible ways to encode signals: level transitions or pulses. When using pulses, you normally send a 0, and when you want to send a signal you send a 1. E.g., if you want to send signals at times 3, 6, 7, you would send string 001001100.... The same signals can be encoded using level transitions by the string 001110111111111.... This time you alternate the transmission of 0’s with the transmission of 1’s, with changes from one symbol to the other in correspondence to the times when you want to send the signals.

Both ways to encode signals are useful, and actually used in different applications, and sometime one needs to translate from one to the other. In this problem you will prove that a set of signals can be recognized by a finite state automaton independently from the encoding chosen for the signals.
Part (a) Let $T$ be the translation function from impulses to levels. For example $T(000100100100) = 0000111000111$, $T(011111) = 010101$, $T(111) = 101$. The translation function is applied to languages in the obvious way: $T(L) = \{T(w) : w \in L\}$.

Prove that if $L$ is regular, then also $T(L)$ is regular.

Part (b) Define the inverse translation function $I(w) = T^{-1}(w)$, from level transitions to pulses. E.g., $I(0001110101) = 0001001111$ $I(111111) = 100000$, $I(\epsilon) = \epsilon$.

Prove that if $L$ is regular, then $I(L)$ is also regular.

Solution Problem 2

Part (a) The proof is by construction.
Given $L$ regular, then we can assume there exists a DFA which recognizes it. Let that be $M = (Q, \Sigma, \delta, q_o, F)$. Our goal is to build a DFA which should recognize the language $T(L)$, namely $M_{TL}$. $M_{TL} = (Q_{TL}, \Sigma, \delta_{TL}, q_{TL0}, F_{TL})$, where:

1. $Q_{TL} = Q \times \{0,1\}$.
2. $\Sigma$ remains the same.
3. the start state: $q_{TL0} = (q_o, 0)$
4. $\delta_{TL}((q,0),0) = (\delta(q,0),0)$
   $\delta_{TL}((q,0),1) = (\delta(q,1),1)$
   $\delta_{TL}((q,1),0) = (\delta(q,1),0)$
   $\delta_{TL}((q,1),1) = (\delta(q,0),1)$
5. $F_{TL} = \{(q_L,q_T)\}$, where $q_L \in F$.

The construction of $M_{TL}$ is complete and that concludes the proof of the regularity of $T(L)$.

Part (b) Proof is by construction.
Given $L$ regular, then we can assume there exists a DFA which recognizes it. Let that be $M = (Q, \Sigma, \delta, q_o, F)$. Our goal is to build a DFA which should recognize the language $I(L)$, namely $M_I = (Q_I, \Sigma, \delta_I, q_I, F)$

1. $Q_{IL} = Q \times \{0,1\}$.
2. $\Sigma$ remains the same.
3. the start state: $q_{IL0} = (q_o, 0)$
4. \[ \delta_{IL}((q,0),0) = (\delta(q,0),0) \]
\[ \delta_{IL}((q,0),1) = (\delta(q,1),1) \]
\[ \delta_{IL}((q,1),0) = (\delta(q,0),0) \]
\[ \delta_{IL}((q,1),1) = (\delta(q,1),1) \]

5. \( F_{IL} = \{(q_L,q_T)\} \), where \( q_L \in F \).

**Problem 3**

Prove that the language

\[ L_3 = \{ w \in \Sigma^* : \text{the length of } w \text{ is odd and the middle character of } w \text{ is a } 1 \} \]

is not regular, using the non regularity of any of the following languages studied in class

- \( L_w = \{ w : w \text{ contains the same number of } 0\text{'s as } 1\text{'s} \} \)
- \( L_\succ = \{ w : w \text{ contains more } 0 \text{ than } 1\text{'s} \} \)
- \( L_{nn} = \{ 0^n1^n : n \geq 0 \} \)

and the closure of regular languages under the regular operations (union, concatenation, star), and the translation operation \( T \) and \( I \) defined in problem 2. (You may also use the fact that languages \{0\} and \{1\} are regular, and therefore any language denoted by a regular expression is regular.)

**Solution Problem 3**

Assume for the purpose of achieving a contradiction that \( L_3 \) is regular. Then we use the operations preserving regularity (complement, union, intersection, Kleene closure, concatenation, \( T \), and \( L \)) to obtain another (presumably) regular language. In this particular case the goal is to express \( L_{nn} \) in terms of \( L_3 \) and other regular languages. So, if \( L_3 \) were regular so would be \( L_{nn} \). But we have proved, in class using the Pumping Lemma, that \( L_{nn} \) is not regular. Thus we have a contradiction, which shows that our initial assumption that \( L_3 \) was regular, is false.

We first take the intersection of \( L_3 \) with the regular language denoted by the regular expression \( 0^*10^* \). \( L = L_3 \cap 0^*10^* \) is the language \( \{ 0^n10^n : n \geq 0 \} \). Then, we apply the function \( T \) from the previous problem to get \( T(L) = 0^n1^{n+1} \). Finally, \( L_{nn} = \epsilon \cup 0 \cdot T(L) \)

**Problem 4**

For each of the following languages, say whether it is regular or not, and prove your answer.
Solution Problem 4

- \( L = \{ a^m b^n a^{m+n} \} \), where \( m, n \geq 0 \). \( L \) is not regular.
  
  The proof is by contradiction. Assume \( L \) is regular. Then by the Pumping Lemma, there exists an integer \( p > 0 \), such that for every \( w \in L \), \( |w| \geq p \), there exists a partition of \( w = xyz \), such that \( |y| > 0 \), \( |xy| \leq p \), and for all \( i xy^i z \in L \).
  
  Let \( w = a^p b^p a^{2p} \). Then any legal partition of \( w \) into 3 strings would have \( y = a^k \), for some \( k \geq 1 \) and \( k \leq p \). Thus for any partition the string \( w_1 = xy^0 z = a^{p-k} b^p a^{2p} \). This is a contradiction because \( w_1 \) has \( p - k + p \) number of \( a \)'s and \( b \)'s followed by \( 2p \) \( a \)'s, and \( p - k + p = 2p \). Thus \( L \) is not regular.

- \( L_r = \{ xwx^R \} \), where \( x, w \) are arbitrary non-empty strings over the alphabet \( \Sigma = \{ a, b \} \). \( L \) is a regular language.
  
  Since \( x \) and \( w \) are arbitrary strings, we can set \( x = a \) then \( L \) becomes the language of all strings that begin and end on \( a \). In similar fashion we can set \( x = b \). Thus \( L \) is precisely the language of all strings which begin and end on the same symbol of the alphabet. The regular expression describing the language is \( a \Sigma \Sigma^* a + b \Sigma \Sigma^* b \)

- \( L = \{ mm^n n \} \), where \( m, n \) are arbitrary non-empty strings over the alphabet \( \Sigma = \{ 0, 1 \} \). \( L \) is not a regular language. (Notice: if \( m, n \) could be empty strings then \( L \) would be the regular language \( \Sigma^* \).

  The proof is by contradiction. Assume \( L \) is regular. Then, using the closure properties of regular languages we get that also \( L' = L \cap ((0)(1)^*) = \{(10)^n(1)^{n+k} : n, k \geq 0 \} \) is regular. Then, we apply the transformation \( I \) to get the language \( L'' = I(L') \cdot 1 = \{1^{2n} 01^{2k} : k \geq n \geq 0 \} \). Finally, we use the pumping lemma to prove that this last language is not regular. Assume \( L'' \) is regular and let \( p \) be the pumping length. Consider the string \( w = 1^{2p} 01^{2p} \).

  This string is in \( L'' \) and has length bigger than \( p \). Therefore if can be written as \( w = xyz \) where \( x, y, z \) satisfy the conditions in the pumping lemma. Since \( |xy| \leq p \), it must be \( x = 1^a, y = 1^b \) and \( z = 1^{2p-a-b} 01^{2p} \) for some \( a, b \). By the pumping lemma \( xz = y^0 z \) is also in \( L'' \), but this is a contradiction because \( xz = 1^{2p-a} 01^{2p} \) and the number of 1's before the 0 is strictly smaller than the number of 1's after the 0.