CS 481 Homework #5

Problem 1

Prove non-regularity of the following languages:

1. \( L = \{0^i | i \geq 0\} \)

   Assume \( L \) is regular. Let \( n \) be the integer of the pumping lemma such that for each string \( w \) in \( L \) with \( |w| \geq n \), we can write \( w = xyz \) where

   - \( y \neq \varepsilon \)
   - \( |xy| \leq n \)
   - for all \( k \geq 0 \), \( xy^kz \) is in \( L \)

   Pick \( w = xyz = 0^n \), where \( n \) is the integer of the pumping lemma. Let \( k = 2 \), giving \( w' = xy^2z \). Since \( |xy| \leq n \), we have \( |y| \leq n \). Finally, note that

   \[ n^2 < |xy^2z| \leq n^2 + n < (n + 1)^2 \]

   Therefore, \( |xy^2z| \) is not a perfect square and \( w' \) is not in \( L \) which is a contradiction. \( L \) is not regular.

2. \( L = \{v | v = \{0,1\}^*, v = v^R\} \), where \( v^R \) is the reverse of the string \( v \).

   Assume \( L \) is regular. Pick \( w = xyz = 0^n1^n \), where \( n \) is the integer of the pumping lemma. Let \( k = 0 \), giving \( w' = xz \). Since \( y \neq \varepsilon \), \( y \) must contain at least one ‘0’ leaving the number of ‘0’s on the left to be less than that of the right, i.e. \( w' = 0^m1^n \), with \( m < n \).

   Hence, \( w' \) is not in \( L \) and we have a proof by contradiction that \( L \) is not regular.

3. Assume the set \( \text{PAREN} \) to be regular and let \( w = xyz = (\text{n})^n \), where \( n \) is the integer of the pumping lemma. Pick \( k = 0 \), giving \( w' = xz \). As \( y \neq \varepsilon \), \( y \) must contain at least one ‘(‘. Consequently, the number of left parenthesis in \( w' \) will be at least one less than the number of right parenthesis.

   Thus, \( w' \) is not in \( L \) and we conclude by proof of contradiction that the set \( \text{PAREN} \) is not regular.

4. \( L = \{0^i\text{v}0^j | i,j > 0, \text{v} = \{0,1\}^*\} \)

   Assume \( L \) to be regular. Let \( w = xyz = 0^i0^i1^i0^0 \) (\( v = 1^i0 \)), where \( n \) is the integer of the pumping lemma. Pick \( k = 0 \), giving \( w' = xz \). Since \( |xy| \leq n \), and \( y \neq \varepsilon \), \( y \) must contain at least one ‘1’ (it may include the 0 if \( |x| = 0 \)). We see that \( w' \) has fewer 1’s on the left than on the right.

   Therefore \( w' \) is not in \( L \), and \( L \) is shown not to be regular by proof of contradiction.
Problem 2. Prove that if $A$ is a regular language over $\Sigma$ and $a, b \in \Sigma$, then the following language is also regular:

$$L = \{ c^n \mid \exists w \in A, \#a(w) + \#b(w) = n \}.$$ 

Proof: let $h$ be a homomorphism on $\Sigma$ which takes $a \mapsto c$ and $b \mapsto c$, and maps all other elements in $\Sigma$ onto $\epsilon$. Then for any $w \in \Sigma$, $h(w) = c^n$, where $\#a(w) + \#b(w) = n$. We know that homomorphisms take regular sets to regular sets, so we conclude that $h(A)$ is also regular. For every $n$ such that $c^n \in L$, there must be some $w \in A$ such that $\#a(w) + \#b(w) = n$, and since $h(w) = c^n$, we conclude that $h(A) = L$. Hence, if $A$ is regular, then $L$ is also regular.
4.2.11:
Show that the regular languages are closed under the following operation:
\[ \text{cycle}(L) = \{ w \mid \text{we can write } w \text{ as } w = xy, \text{ such that } xy \text{ is in } L \}. \]
For example, if \( L = \{01, 011\} \), then \( \text{cycle}(L) = \{01, 10, 011, 110, 101\} \). Hint: Start with a DFA for \( L \) and construct an \( \varepsilon \)-NFA for \( \text{cycle}(L) \).

Set up the problem:
For a regular language \( L \), assume we have the following deterministic finite automata:

We can also write \( w \) as:

Let \( q_0 \) be the initial point/start state, \( q_f \) be the final point/accepting state, and \( q \) be any point in between of string \( w \).

To show that \( \text{cycle}(L) \) is regular, we design a nondeterministic finite automata as follows:

This automata has states that consists of two parts: the first element is the point \( q \) in the string at which the FA starts and the second state keeps track of where the FA is in the string. The FA will go through the \( y \) portion of the string and then start guessing for the \( x \) part of the string because of its nondeterminism. This FA moves exactly like the former automata but its only difference is that it accepts when the first element matches up with the second element. This is to say that the FA will have completed a cycle by coming back to its start state before it accepts. Hence, regular languages are closed under \( \text{cycle} \).

The transition function for the second automata is:
\[ \delta_2 ([p, q], a) = [p, \delta_1(q, a)] \]
where subscripts 1 and 2 denote the first and second FA, respectively. \( p \) is the start state, \( q \) is the current state, \( a \in \Sigma^* \).

Note: If students do draw the \( \varepsilon \)-NFA, the epsilon moves go from the start state to any point in the string because any point in the string can be the start of the cycle.
Problem 4

Let $p = 2$ be the pumping length. Let $s \in L$, $|s| \geq p$, $s = a^i b^j c^k$. We will show that $s$ can be broken up into $s = xyz$ satisfying the conditions of the pumping lemma. There are 4 cases.

Case 1: $i = 0$. We set $x = \varepsilon$, $y = \text{first symbol of } s$, $z = \text{the rest of } s$. Note that $y$ can be $b$ or $c$. Either way, when we pump $y$, we only change the number of that letter in our string. All the $b$'s (if $s$ had any) will still be before the $c$'s, and the number of $a$'s will still be 0. So the resulting string is in $L$.

Case 2: $i = 1$. We set $x = \varepsilon$, $y = a$, $z = b^j c^k$. We know that $j = k$ in this case. If we pump $y$ (up or down), the number of $a$'s in the string goes up or down, but since $j = k$, the resulting string must be in $L$.

Case 3: $i = 2$. We set $x = \varepsilon$, $y = aa$, $z = b^j c^k$. When we pump $y$, the number of $a$'s can go down to 0 or it can go up, but it will never be 1, so the resulting string must be in $L$.

Case 4: $i > 2$. We set $x = \varepsilon$, $y = a$, $z = a^{i-1} b^j c^k$. When we pump $y$, the number of $a$'s can go down to 2 or it can go up, but it will never be 1, so the string must still be part of $L$.

To show that $L$ is non-regular: Suppose $L$ is regular. Then $L' = L \cap ab^* c^* = \{ab^j c^k : i \geq 0\}$ is also regular. Let $h(a) = \varepsilon$, $h(b) = 0$, $h(c) = 1$ be a homomorphism. This implies that $L'' = h(L') = \{0^i 1^i : i \geq 0\}$ is regular. Since we know $L''$ is non-regular (something that we can prove using the pumping lemma), we have a contradiction. So $L$ must have been non-regular.

This does not contradict the pumping lemma since it only states that regular languages all satisfy its conditions. However, the converse is not necessarily true: A language satisfying the pumping lemma does not have to be regular. (This is an important fact to keep in mind when writing proofs involving the pumping lemma.)