Assignment#2 Key
1a. EveryOther(L) = \{ a_1 a_3 \ldots a_{2n-1} \mid a_1 a_2 a_3 \ldots a_{2n-1} a_{2n} \text{ is in } L \} \\

• Approach 1: Let L be a Regular language over the finite alphabet \( \Sigma \). For each \( a \in \Sigma \), define \( f(a) = \{ a, a' \} \), \( g(a) = a' \) and \( h(a) = a \), \( h(a') = \lambda \), \( f \) is a substitution, \( g \) and \( h \) are homomorphisms. EveryOther(L) = h(f(L) \cap (\Sigma \cdot g(\Sigma))^* ) \\

• Why this works: \( f(L) \) gets us every possible random priming of letters of strings in L. \((\Sigma \cdot g(\Sigma))^* \) gets every word composed of pairs of unprimed and primed letters from \( \Sigma \). Intersecting this with \( f(L) \) gets strings of the form \( a_1 a_2' a_3 a_4' \ldots a_{2n-1}a_{2n}' \) where \( a_1 a_2 a_3 a_4 \ldots a_{2n-1}a_{2n} \text{ is in } L \) 

Applying the homomorphism \( h \) erases all primed letters resulting in every string \( a_1 a_3 \ldots a_{2n-1} \) where \( a_1 a_2 a_3 a_4 \ldots a_{2n-1}a_{2n} \text{ is in } L \), precisely the language EveryOther(L) that we sought. This works as Regular Languages are closed under intersection, concatenation, *, substitution and homomorphism.
1a. EveryOther(L) = \{ a_1 a_3 \ldots a_{2n-1} \mid a_1 a_2 a_3 \ldots a_{2n-1}a_{2n} \text{ is in } L \} \\

• Approach 2: Let L be a Regular language over the finite alphabet \( \Sigma \). Assume L is recognized by the DFA \( A_1 = (Q, \Sigma, \delta_1, q_1, F) \). Define NFA \( A_2 = (Q, \Sigma, \delta_2, q_1, F) \), where \( \delta_2(q,a) = \text{union}(b \in \Sigma) \{ \delta_1(\delta_1(q,a),b) \} \)

• Why this works: Every transition that \( A_2 \) takes is one that \( A_1 \) would have taken when reading a pair that starts with the character read by \( A_1 \) followed by any arbitrary character.
1b. \( \text{Half}(L) = \{ x \mid \text{there exists a } y, |x| = |y| \) and \( xy \) is in \( L \) \}

- Let \( L \) be a Regular language over the finite alphabet \( \Sigma \). Assume \( L \) is recognized by the DFA \( A_1 = (Q, \Sigma, \delta_1, q_1, F) \). Define the NFA \( A_2 = ((Q \times Q \times Q) \cup \{q_0\}, \Sigma, \delta_2, q_0, F') \), where \( \delta_2(q_0, \lambda) = \text{union}(q \in Q) \{< q_1, q, q >\} \) and \( \delta_2(< q, r, s >, a) = \text{union}(b \in \Sigma) \{ < \delta_1(q, a), \delta_1(r, b), s > \} \), \( q, r, s \in Q \)

\( F' = \text{union}(q \in Q) \{< q, f, q >\}, f \in F \)

- Why this works:
  - The first part of a state \( < q, r, s > \) tracks \( A_1 \).
  - The second part of a state \( < q, r, s > \) tracks \( A_1 \) for precisely all possible strings that are the same length as what \( A_1 \) is reading in parallel. This component starts with a guess as to what state \( A_1 \) might end up in.
  - The third part of a state \( < q, r, s > \) remembers the initial guess.

Thus, \( \delta_2(< q_1, q, q >, x) = \{ \delta_1*(q_0, x), \delta_1*(q, y), q >\} \) for arbitrary \( y, |x| = |y| \)

We accept if the initial guess was right and the second component is final, meaning \( xy \) is in \( L \).
2. \( L = \{ a^m b^n c^t \mid t = \min(m,n) \} \)

a.) Use the **Myhill-Nerode Theorem** to show \( L \) is not Regular. Define the equivalence classes \([a^i b^i], \ i \geq 0\)
Clearly \( a^i b^i c^i \) is in \( L \), but \( a^j b^j c^i \) is not in \( L \) when \( j \neq i \)
Thus, \([a^i b^i] \neq [a^j b^j]\) when \( j \neq i \) and so the index of \( R_L \) is infinite.
By Myhill-Nerode, \( L \) is not Regular.
2. \( L = \{ a^m b^n c^t \mid t = \min(m,n) \} \)

b.) Use the **Pumping Lemma for CFLs** to show \( L \) is not a CFL

**Me:** \( L \) is a CFL

**PL:** Provides \( N > 0 \)

**Me:** \( z = a^N b^N c^N \)

**PL:** \( z = uvwxy, |vwx| \leq N, |vx| > 0, \) and \( \forall i \geq 0 \ uv^iwx^iy \in L \)

**Me:** Since \( |vwx| \leq N \), it can consist of \( a \)'s and/or \( b \)'s or \( b \)'s and/or \( c \)'s but never all three. Assume it contains no \( c \)'s then \( i = 0 \) decreases the number of \( a \)'s and/or the number of \( b \)'s, but not the \( c \)'s and so there are more \( c \)'s than the minimum of \( a \)'s and \( b \)'s.

Assume it contains \( c \)'s then \( i = 2 \) increases the number of \( c \)'s and maybe number of \( b \)'s, but not the \( a \)'s and so there are more than \( N \) \( c \)'s but just \( N \) \( a \)'s.
2. \( L = \{ a^m b^n c^t \mid t = \min(m,n) \} \)

c.) Present a CSG for \( L \) to show it is context sensitive

\[ G = ( \{ A, B, C, <a>, <b> \}, \{ a, b, c \}, R, A ) \]

\[
A \rightarrow aBbc \mid abc \mid a \mid b \mid \lambda \\
B \rightarrow aBbC \mid a<a>bC \mid ab<b>C \quad \text{// allow more a’s or more b’s} \\
Cb \rightarrow bC \quad \text{// Shuttle C over to a c} \\
Cc \rightarrow cc \quad \text{// Change C to a c} \\
<a> \rightarrow a<a> \mid \lambda \\
<b> \rightarrow b<b> \mid \lambda
\]