Sample Assignment#2 Key

1a. EveryOther(L) = { $a_1 a_3 \ldots a_{2n-1}$ | $a_1 a_2 a_3 \ldots a_{2n-1} a_{2n}$ is in L }

- Approach 1: Let **L** be a Regular language over the finite alphabet **Σ**. For each **a∈Σ**, define **f(a) = {a,a'}**, **g(a) = a'** and **h(a) = a**, **h(a') = λ**,
  f is a substitution, **g** and **h** are homomorphisms.
  EveryOther(L) = h(f(L) ∩ **(Σ ▪ g(Σ))*)**
- Why this works:
  f(L) gets us every possible random priming of letters of strings in L.
  (Σ ▪ g(Σ))* gets every word composed of pairs of unprimed and primed letters from **Σ**. Intersecting this with **f(L)** gets strings of the form
  $\mathbf{a_1\, a_2'\, a_3\, a_4' \ldots a_{2n-1} a_{2n}'}$ where $\mathbf{a_1\, a_2\, a_3\, a_4 \ldots a_{2n-1} a_{2n}}$ is in **L**
  Applying the homomorphism **h** erases all primed letters resulting in every string $\mathbf{a_1\, a_3 \ldots a_{2n-1}}$ where $\mathbf{a_1\, a_2\, a_3\, a_4 \ldots a_{2n-1} a_{2n}}$ 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) = { $a_1\ a_3\ \ldots\ a_{2n-1}$ | $a_1\ a_2\ a_3\ \ldots\ a_{2n-1}a_{2n}$ is in L }

- Approach 2: Let **L** be a Regular language over the finite alphabet **Σ**. Assume **L** is recognized by the DFA $\mathbf{A_1 = (Q, \Sigma, \delta_1, q_1, F)}$. Define NFA $\mathbf{A_2 = (Q, \Sigma, \delta_2, q_1, F)}$, where $\mathbf{\delta_2(q,a) = union(b \in \Sigma)\ \{\ \delta_1(\delta_1(q,a),b)\ \}}$
- Why this works:
  Every transition that $\mathbf{A_2}$ takes is one that $\mathbf{A_1}$ would have taken when reading a pair that starts with the character read by $\mathbf{A_1}$ followed by any arbitrary character.

1b. Half(L) = { x | there exists a y, |x| = |y| and xy is in L }

- Let **L** be a Regular language over the finite alphabet **Σ**. Assume **L** is recognized by the DFA **A_1 = (Q, Σ, δ_1, q_1, F)**. Define the NFA
  A_2 = ((Q×Q ×Q)∪{q_0}, Σ, δ_2, q_0, F'), where
  $\delta_2(q_0,\lambda)$ = union(q∈Q) {<q_1, q, q>} and
  δ_2(< q, r, s > ,a) = union(b∈Σ) { < δ_1(q,a), δ_1(r,b), s > } , q,r,s ∈ Q
  F' = union(q∈Q) {<q, f, q>}, f∈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, **δ_2*(< q_1, q, q > ,x) = {δ_1*(q_0,x), δ_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. Let L = { $a^m b^n c^t$ | t = min(m,n) }

a.) Use the **Myhill-Nerode Theorem** to show **L <u>is not</u>** 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, [**$a^i b^i$**] **≠ [$a^j b^j$]** when **$j \neq i$** and so the index of **R_L** is infinite.

By Myhill-Nerode, **L** is not Regular.

2. Let L = { $a^m b^n c^t$ | t = min(m,n) }

b.) Use the **Pumping Lemma for Regular Languages** to show **L** **<u>is not</u>** a CFL

Me: **L** is a CFL

PL: Provides **N>0**

Me: **w = $a^N b^N c^N$**

PL: **w = xyz, |xy| ≤ N**, **|y| > 0**, and **∀i≥0 xy^iz ∈ L**

Me: Since **| xy | ≤ N**, it can consist of **a**'s only.
Let **i=0** which decreases the number of **a**'s, but not the **b**'s or **c**'s, and so the result is **$a^{N-|y|} b^N c^N$, with |y| > 0**. Thus, this contradicts that **∀i≥0 xy^iz ∈ L** and so **L** is not Regular by Pumping Lemma.