Assignment#2 Key

1a. ProperSuffix(L) = { y | w is in L, x is not lambda and w = xy }

- 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.
  ProperSuffix(L) = h(f(L) ∩ (g(Σ^+) Σ*))
- Why this works:
  f(L) gets us every possible random priming of letters of strings in **L**.
  g(Σ^+) Σ* gets every word that starts with at least one letter primed and ends in a sequence (possibly null) of unprimed letters. Intersecting this with **f(L)** gets strings in **L** with non-null prefixes primed and the rest(the proper suffix) unprimed**.**
  Applying the homomorphism **h** erases all primed letters getting proper suffixes. This works as Regular Languages are closed under intersection, concatenation, *, $^+$, substitution, and homomorphism.
- Can also create an NFA from DFA for **L**, but that's too much work.

1b. SomeHalf(L) = { y | there exists a string x , |x| = |y| and either xy is in L or yx is in L }

Proof. Want to show that if L is regular, SomeHalf(L) is also regular.

Let A1 = (Q, Σ, δ, q0, F).

The NFA has a 5-tuple of state sets where the first element is the state on the first half, the second is the union of states on the guessed second half, the third is the union of states on the guessed first half, the fourth is the state on the second half, and the fifth is the guessed state at the midpoint. The start state is <q0, h, q0, h, h> and the accepting states are <h,f,h,f',h> where either f or f' is a final state.

Let A2 = $(Q', \Sigma, \delta', q'_0, F')$ where:$Q' = (Q \times 2^Q \times 2^Q \times 2^Q \times 2^Q) \cup \{q'_0\}$

$\delta'(q'_0, \lambda) = \text{union}(h \in Q)\{< q_0, h, q_0, h , h >\}$

$\delta'(< q, r, s, t, h>, a) = \text{union}(b \in \Sigma)\{< \delta(q, a), \delta(r, b), \delta(s, b), \delta(t, a), h >\}$, $q, r, s, t \in Q$

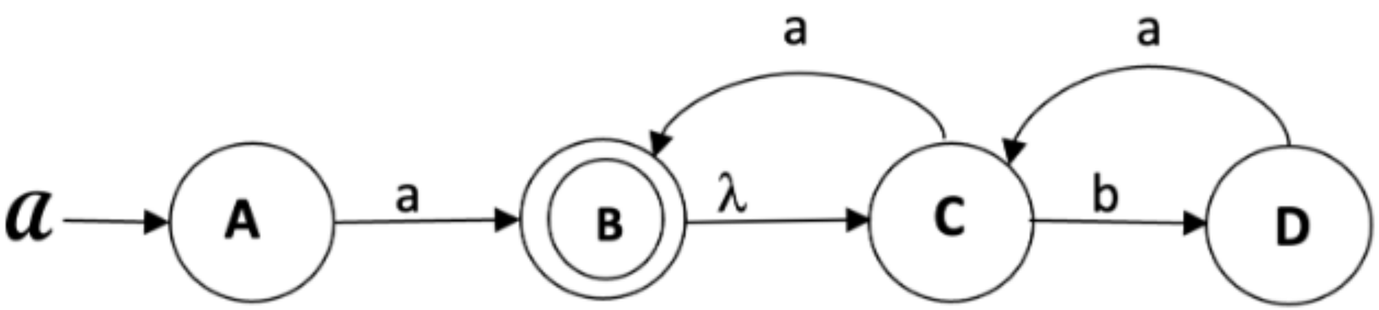
$F' = \{< h, f, s, t, h>\} \cup \{< q, r, h, f, h>\}$ $\qquad h,q,r,s,t \in Q, f \in F,$

$\text{union}(h \in Q)\ \delta'^*(< q_0, h, q_0, h , h >, x)$ contains $<\delta^*(q_0,x), \delta^*(h,y), \delta^*(q_0,y), \delta^*(h,x), h>$

where $|x|= |y|$, $xy \in L$ or $yx \in L$

Thus, regular expressions are closed under SomeHalf

2. Use Regular Equations to Solve for B



A = λ

B = Aa + Ca **= a + Ca = a + B(ba)* a = a((ba)* a)***

C = B + Da **= B + (Cb) a = B(ba)***

D = Cb

L = B = a((ba)* a)*

3. $L = \{ a^m b^n c^t \mid t = m \% n \}$

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

Assume **L** is Regular

Let **N>0** be value provided by PL

Choose $\mathbf{a^N b^N c^0}$ as a string in **L**

PL splits $\mathbf{a^N b^N c^0}$ into xyz such that $|xy| \le N$ and $|y| > 0$. $\forall i \ge 0$, $xy^iz \in L$

y is strictly over **a's**. Set **i=0** and we get $\mathbf{a^{N-|y|} b^N}$ but then the there are no **c**'s that should exist following so the resulting string is not in **L**.

So **L** is not Regular based on the PL.

3. $L = \{ a^m b^n c^t \mid t = m \% n \}$

b.) Use the **Myhill-Nerode Theorem** to show **L <u>is not</u>** Regular.

Define the equivalence classes $\mathbf{[a^i b^{i+1}], i \geq 0}$

Clearly $\mathbf{a^i b^{i+1} c^i}$ is in **L**, but $\mathbf{a^j b^{j+1} c^i}$ is not in **L** when **$j \neq i$, i, j>0**

Thus, $\mathbf{[a^i b^{i+1}] \neq [a^j b^{j+1}]}$ when **$j \neq i$, i, j>0** and so the index of $\mathbf{R_L}$ is infinite.

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