Assignment#2 Key

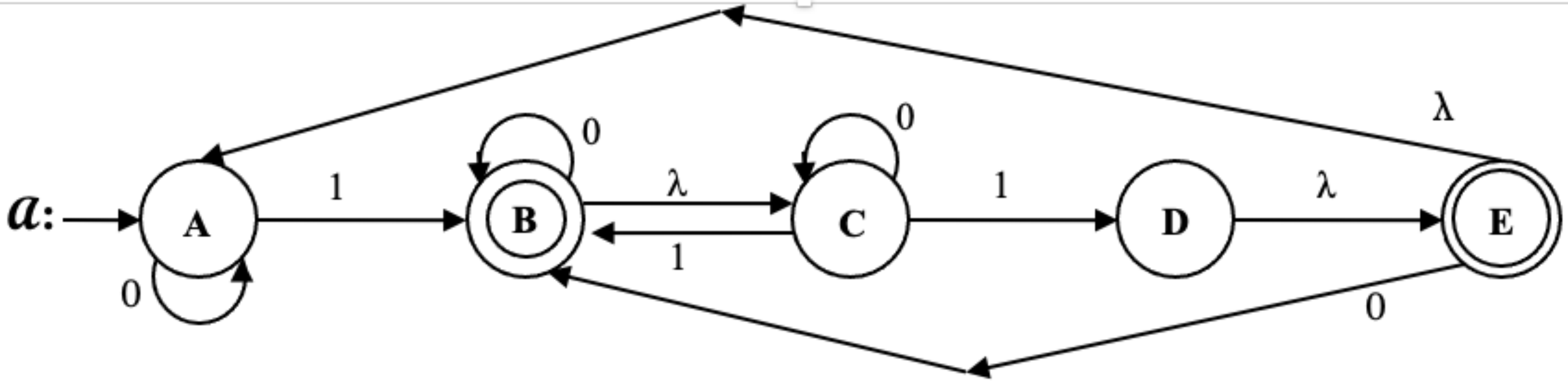
1a. OnePairOfZerosRemoved(L) = { xy | w is in L and w = x00y }

- Let **L** be a Regular language over the finite alphabet **Σ** that contains a **0**. 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.
  OnePairOfZerosRemoved(L) = h(f(L) ∩ Σ* 0'0' Σ*)
- Why this works:
  f(L) gets us every possible random priming of letters of strings in **L**.
  Σ* 0'0' Σ* gets every word that cnatins a pair of zeros somewhere, with that pair primed in this expression. Intersecting this with **f(L)** gets strings of the desired form that occur in **L.**
  Applying the homomorphism **h** erases all primed letters, which in this cases is just a pair of **0**'s occurring somewhere in the string. 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. LastHalfReversed(L) = { y | there exists a string x , |x| = |y| and xy^R is in L }

- 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 the NFA
  $\mathbf{A_2 = ((Q\times Q\times Q)\cup\{q_0\}, \Sigma, \delta_2, q_0, F')}$, where
  $\mathbf{\delta_2(q_0,\lambda) = union(q,r\in Q)\ \{<q_1, q, q, r, r>\}}$ **and**
  $\mathbf{\delta_2(< s, t, u, v, w >, b) = union(a,c\in\Sigma)\ \{ < \delta_1(s,a), \delta_1(t,b), u, \delta_1(v,c), w> \}}$ **, s,t,u,v,w ∈ Q**
  $\mathbf{F' = union(q\in Q)\ \{<q, q, r, f, r>\}, f\in F}$
- Why this works:
  The first part of a state $\mathbf{< s, t, u, v, w >}$ tracks $\mathbf{A_1}$ for all possible strings that are the same length as what $\mathbf{A_2}$ is reading in parallel**.** We guess it will end up in state **q** and so **u=q** to remember that guess.
  The second part of state $\mathbf{< s, t, u, v, w >}$ tracks $\mathbf{A_1}$ as if it has read a string that ended in state **q** (**u=q**). This part actually reads the mid part of a string divided into thirds.
  The third part of a state $\mathbf{< s, t, u, v, w >}$ tracks $\mathbf{A_1}$ for all possible strings that are the same length as what $\mathbf{A_2}$ is reading in parallel**.** We guess that reading the mid part will end up in state **r** (**w=r**).
- Thus, we start with a guess (**q**) as to what state $\mathbf{A_1}$ might end up in reading a string of length **x.**
  The guess is checked by requiring us to start up in state **q** in the mid part which reads **y**, were **|x|=|y|**.
  We guess that we will end up in state **r** after reading **y.**
  The guess is checked by requiring us to start up in state **r** in the third part which simulates reading a string **z**, where **|x|=|y|=|z|**.
- The final states check that our guesses were correct, and the third part could end in a final state of $\mathbf{A_1}$.

2. Use Regular Equations to Solve for B + E



$A = \lambda + E + A0 = 0^* + E0^*$ $= 0^* + D0^*$

$B = A1 + C1 + E0 + B0$ $= 0^*1 + B0^*1\ 0^*1 + B0^*1 + B0^*1\ 0 + B0$

$= 0^*1 + B(0 + 0^*1(0^*1+0+\lambda))$

$= 0^*1(0 + 0^*1)^*$

$C = B + C0$ $= B0^*$

$D = C1 = B0^*1$ $= 0^*1(0 + 0^*1)^*0^*1$

$= 0^*1(0 + 0^*1)^*$

$E = D$

$B+E = B+D$ $= 0^*1(0 + 0^*1)^*$

3. L = { $a^n b^{3^n}$ | n > 0 }

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

Define the equivalence classes **[a^i], i > 0**

Clearly **$a^i b^{3^i}$** is in **L**, but **$a^j b^{3^i}$** is not in **L** when **j ≠ i, i, j>0**

Thus, **[a^i] ≠ [a^j]** when **j ≠ i, i, j>0** and so the index of **R_L** is infinite.

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

3. L = { a^n $b^{3^\wedge n}$ | n > 0 }

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 $b^{3^\wedge N}$**

PL: **z = uvwxy, |vwx| ≤ N**, **|vx| > 0**, and **∀i≥0 uv^iwx^iy ∈ L**

Me: If **vwx** includes the one **a** then set **i=2** and we get a string with at least **N+1 a**'s. If it contains any **b**'s, then there will be at most **N-1 b**'s added, but in the simplest case where we add just one **a**, we would need to add **$3^{(N+1)}$-3^N = 3(3^N)-3^N = 2(3^N) b**'s. But **2(3^N)>(N-1)** for all **N,** and so **uv^2wx^2y** is not in **L**. Thus, we can assume **vwx** is over **b**'s only. But then setting **i = 0** reduces the number of **b**'s without reducing **a**'s and so **uwy** is not in **L**. That covers all cases, leading to contradictions in each, so **L** is not a CFL.

3. L = $\{ a^n b^{3^n} \mid n > 0 \}$

c.) Present a CSG for **L** to show it **<u>is</u>** context sensitive

G = ({ S, A, B, C, D }, { a, b }, R, S)

S	→ abbb |	// Base case of ab^3 and kick start for other cases
S	→ aAbbB	// One a and two b's and a character that will become a B
A	→ aAC	// C will shuttle right tripling b's; we will still have a B left
A	→ aD	// D will shuttle right tripling b's; we will not have a B left
Cb	→ bbbC	// Triple the number of b's
CB	→ bbB	// Triple last b (B), but one b is still a B
Db	→ bbbD	// Triple the number of b's
DB	→ bbb	// Triple last b (B), leaving no non-terminal