Assignment#2 Key

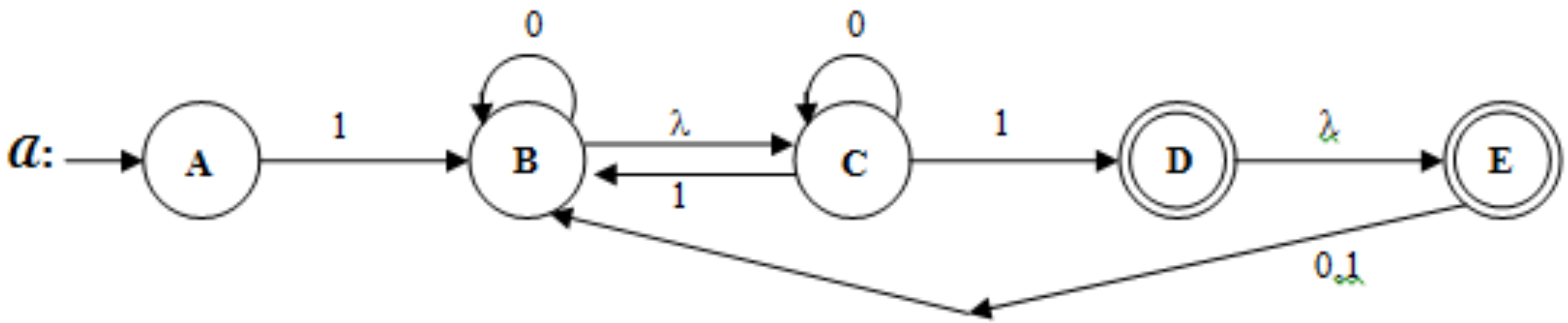
1a. NoLeadingOrLaggingAs(L) = { x| a*xa* is in L and x has no leading or lagging a's }

- 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.
  NoLeadingOrLaggingAs(L) = h(f(L) ∩ (g(a*) (Σ-{a}) Σ* (Σ-{a}) g(a*)))
- Why this works:
  f(L) gets us every possible random priming of letters of strings in **L**.
  (g(a*) (Σ-{a}) Σ* (Σ-{a}) g(a*)) gets every word composed of a sequence of zero of more **a**'s followed by a non-**a** character from **Σ** then a sequence of characters from **Σ** that ends in a non-**a** character from **Σ** followed by a sequence of zero of more **a**'s. Intersecting this with **f(L)** gets strings of the desired form that occur in **L.** Applying the homomorphism **h** erases all primed letters resulting in every string the language **NoLeadingOrLaggingAs(L)** that we sought. This works as Regular Languages are closed under intersection, concatenation, *, substitution and homomorphism.

1b. MidThird(L) = { y | there exists a x and z, |x| = |y| = |z| and xyz 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
  $\mathbf{A_2 = ((Q\times Q\times Q\times Q\times Q)\cup\{q_0\}, \Sigma, \delta_2, q_0, F')}$, where
  $\mathbf{\delta_2(q_0,\lambda) = union(q,r\in Q)\ \{<q_1, q, q, r, r>\}}$ **and**
  $\mathbf{\delta_2(<s, t, u, v, w>, b) = union(a,c\in\Sigma)\ \{ <\delta_1(s,a), \delta_1(t,b), u, \delta_1(v,c), w> \}, s,t,u,v,w \in Q}$
  $\mathbf{F' = union(q\in Q)\ \{<q, q, r, f, r>\}, f\in F}$
- Why this works:
  The first part of a state **< 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 **< 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 **< 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 D + E



A = λ
B = A1 + C1 + E(0+1) + B0 = 1 + B0*1 + D(0+1) + B0 = 1 + C1(0+1) + B(0*1 + 0)
= 1 + B0*1(0+1) + B(0*1 + 0) = 1(0*1(0+1) + 0*1 + 0)*
C = B + C0 = B0* = 1(0*1(0+1) + 0*1 + 0)* 0*
D = C1 = 1(0*1(0+1) + 0*1 + 0)* 0*1 = 1(0*1(0+1+λ) + 0)* 0*1
E = D

3. L = $\{ a^m b^{2^n} \mid m,n > 0 \}$

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

Define the equivalence classes $\mathbf{[ab^{2^i}], i \geq 0}$

Clearly $\mathbf{ab^{2^i}b^{2^i}}$ is in **L**, but $\mathbf{ab^{2^j}b^{2^i}}$ is not in **L** when $\mathbf{j > i}$

Thus, $\mathbf{[ab^{2^i}] \neq [ab^{2^j}]}$ when $\mathbf{j \neq i}$ and so the index of $\mathbf{R_L}$ is infinite.

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

3. L = { $a^m\ b^{2\wedge n}$ | m,n > 0 }

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

Me: **L** is a CFL

PL: Provides **N>0**

Me: **z = a $b^{2\wedge N}$**

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

Me: If **vwx** includes the one **a** then set **i=0** and we get a string with no **a**’s which is not in **L**. Thus, we can assume **vwx** is over **b**’s only. Since **0<|vwx|≤N**, setting **i=2** adds at least one **b** and at most **N b**’s. However, the smallest member of **L** longer than **a $b^{2\wedge N}$** has **2^N** more **a**’s and so **uv^2wx^2y** cannot be in **L**. Thus, the Pumping Lemma shows **L** to not be a CFL.

3. L = $\{ a^m b^{2^n} \mid m,n > 0 \}$

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

G = ({ S, A, B, C, D, E, X, <a>, <b> }, { a, b }, R, S)

S	→ <a>bb	// Base case of a^+b^2
S	→ Bb<b>	// One b and a pseudo b <b> that is a sentinel; gets all cases about b^2
Bb	→ XbC	// C will shuttle left doubling b's; first b doubled later
Cb	→ bbC	// Double the number of b's
C<b>	→ Db<b>	// Double but keep that sentinel at end
C<b>	→ Ebb	// Done doubling so get rid of sentinel
bD	→ Db	// Shuttle D back to start
XD	→ Bb	// Change XD back to Bb to continue increasing b's
bE	→ Eb	// Shuttle E back to start
XE	→ <a>b	// Switch to generating a's
<a>	→ a<a> | a	// Start with a^+