Name: Aytaç SEKMEN

Student ID: 2575983

QUESTION 1:

1) At first I think that would be good to write the transitions:

δ(q0,a)=q1, δ(q0,b)=q2

δ(q1,a)=q1, δ(q1,b)=q2

δ(q2,a)=q2, δ(q2,b)=q3

δ(q3,a)=q4, δ(q3,b)=q5

δ(q4,a)=q0, δ(q4,b)=q2

δ(q5,a)=q2, δ(q5,b)=q3

Step-1:

And now I can seperate the states to 2 equivalence class: the accepting and non accepting ones: ($\equiv_0$)

{q0,q1,q4,q3} and {q2,q5}.

Step-2:

Now I can look for whether the states in same classes for $\equiv_0$ are in same equivalence class for $\equiv_1$. To do it I should look at whether the ending of states in same classes are same when they get same input word.

Check for q0 and q1:

Since δ(q0,a)=δ(q1,a)=q1 and δ(q0,b)= δ(q1,b)=q2, So q0 and q1 must be in the same set for the equivalence class of $\equiv_1$.

Check for q3 and q4:

δ(q3,a)=q4, δ(q4,a)=q0 and q0 $\equiv_0$ q4. δ(q3,b)=q5 δ(q4,b)=q2 q5 $\equiv_0$ q2. So nothing changes q3 and q4 must be in the same equivalence class for $\equiv_1$.

Check for q1 and q3:

δ(q1,a)=q1, δ(q3,a)=q4 and q1 $\equiv_0$ q4. δ(q1,b)=q2 δ(q3,b)=q5 and q0 $\equiv_0$ q4. So nothing changes q1 and q3 must be in the same equivalence class for $\equiv_1$.

Check for q2 and q5:

δ(q2,a)=δ(q5,a)=q2 δ(q2,b)=δ(q5,b)=q3 So nothing changes q2 and q5 must be in the same equivalence class for $\equiv_1$.

Conclusion:

So actually nothing changes between $\equiv_1$ and $\equiv_0$. So my final equivalence classes become {q0,q1,q4,q3} and {q2,q5}. And my transition function change like this:

δ ({q0,q1,q4,q3},a)= {q0,q1,q4,q3}

δ ({q0,q1,q4,q3},b)= {q2,q5}

δ ({q2,q5},a)= {q2,q5}

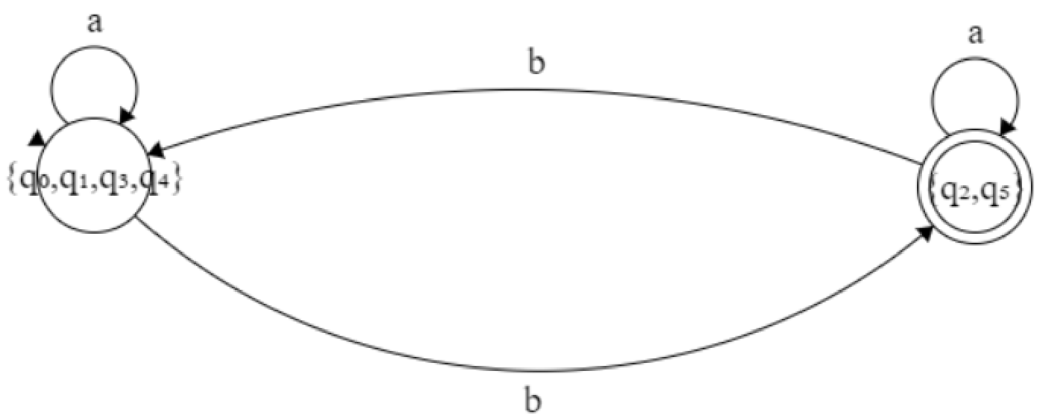
δ ({q2,q5},b)= {q0,q1,q4,q3}

My final state is: {q2,q5}

My initial state is: {q0,q1,q4,q3}

Acceptance state: {q2,q5}

So lets draw the DFA:



2)

[e]=Lba* U {}*

[b]=L

3)Let's say that n=m=k=4x and u=x for a certain string

1) $w=a^{4x}b^{4x}c^{4x}d^{x}$

Let's say that n=m=k=4x and u=y and x≠y for another string

2) $q=a^{4x}b^{4x}c^{4x}d^{y}$

Then if I concatenate these two strings with the string $z=d^{x}$

We will see that wz(Because 8x=4x+2(x+x)) in the language L' but qz is not (Because 4x+2(x+y) =6x+2y which is not equal to 8x since x≠y)

So I concluded that these two strings are at different equivalence classes.

And I actually proved that the language L' is not a regular language by using Myhill Nerode Theorem since I can produce infinitely many equivalence classes for infinite pair of (x,y) such that x≠y

QUESTION 2:

1) G = (V, Σ, R, S) where V = {S,B,R} ∪ Σ, Σ = {a,b} and R = {S → Bb | RS | SRB, B → Bb, B → e, R → RR | aRb | bRa | e}

- The rule R actually represents the conditions for equal amount of a and b.
- The rule B actually represents the condition for b*
- The S represents the conditions that can we can obtain.

2) G = (V, Σ, R, S) where V = {S,X,Y} ∪ Σ, Σ = {0, 1,2} and R = {S→XY, X→ 0X1|e, Y→ 1Y2| e}

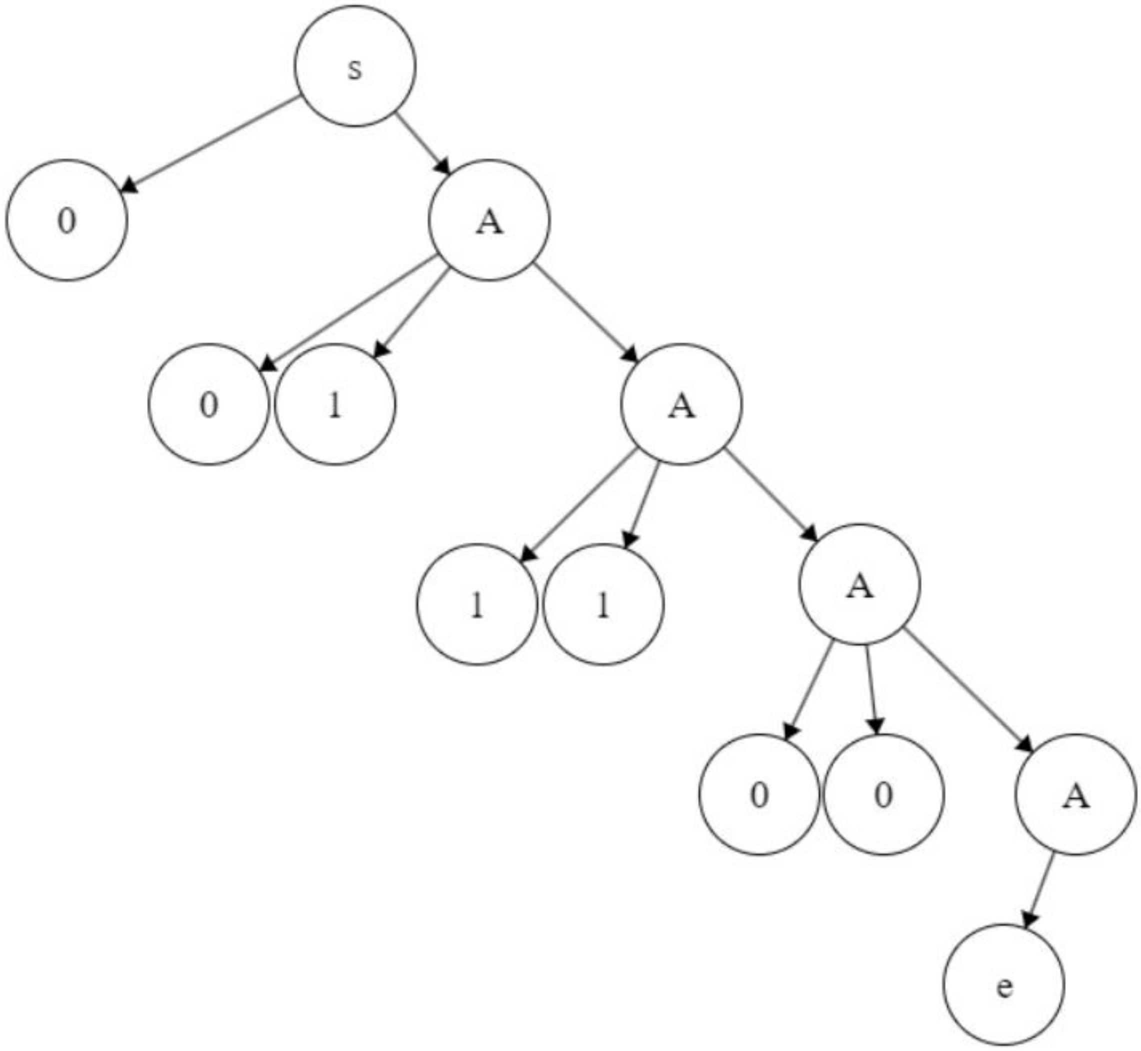
- X actually creates equal amount of 0 and 1.
- Y actually creates equal amount of 1 and 2.

3) G = (V, Σ, R, S) where V = {S,A} ∪ Σ, Σ = {0, 1} and R = { S→0A|1A, A→ e| 00A| 10A| 11A| 01A}

- Rule from A actually show the all posibilites after we get the first digit of the strings.
- And the rules from A actually guarentees at least 1 digit.

Derivation:

S=>0A=>001A=>00111A=>0011100A=>0011100



Corresponding Parse Tree

QUESTION 3:

1) L = {w | w=xzx, |x|=1 and z,x ∈ {0, 1}* } U {e}

- This x actually represents the same string with length 1 both at the beginning and end.

2) L = {w | w includes at least two 1s and w ∈ {0, 1}* }

- I'm not sure this verbal statement is a correct way of stating the language but I could also state it like this (I mean I am not sure, whether stating languages by using this kind of verbal statement is true or not):

$L = \{w \mid w = x1y1z \text{ and } x, y, z \in \{0, 1\}^* \}$