Q1:

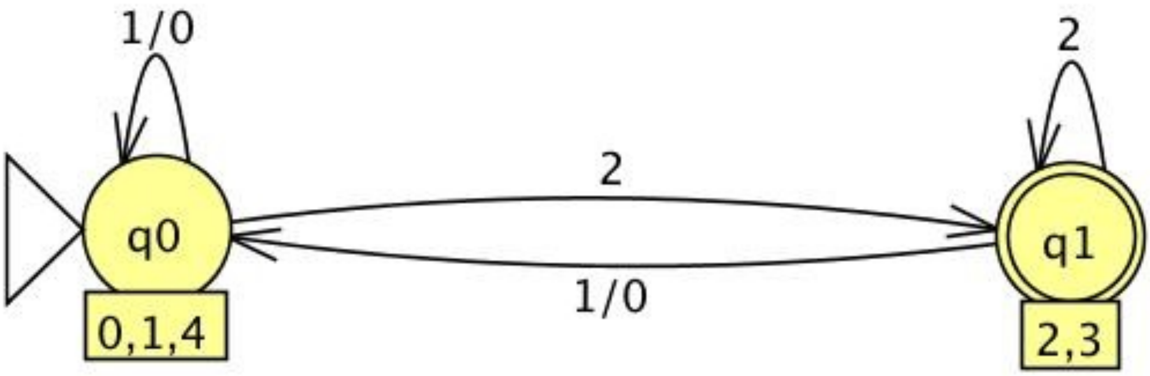
a). To find the equivalent states, we apply the following algorithm (Mange, J., 2008):

1. Create a table called DISTINCT, initially blank;
2. For each pairs of states (p, q), if one of them is a final state and the other is not, then DISTINCT(p, q)=ε;
3. Loop all pairs (p, q) and do the following until no change is made:
   If DISTINCT(p, q) is empty and DISTINCT(δ(p), δ(q)) is not for some character a belongs to the language, then DISTINCT(p, q)=a
4. After step 3, if for some pairs of states (p, q), DISTINCT(p, q) is empty, then they are equivalent, combine them, and you will get the minimized DFA.

For this DFA, the DISTINCT table is:

q0	-	-	-	-	-
q1		-	-	-	-
q2	ε	ε	-	-	-
q3	ε	0		-	-
q4			0	ε	-
	q0	q1	q2	q3	q4

Therefore, q0 & q1 & q4 are equivalent, q2 & q3 are equivalent, hence, the minimized DFA is:



Therefore, the minimized DFA recognize the language L={δ∈{0, 1, 2}* | δ ends with 2}.

b). If we change q3 to rejecting state and q4 to accepting state, then the changed DFA is already minimized.

c). Two, because when we convert a DFA to accept its complement, the only thing we need to do is to flip the states: change the rejecting states to accepting states, and change the accepting states to rejecting states. So there is no additional states being added to the DFA.

Q2:

Yes, proof:

Since the language is defined as the set of strings containing only {a}, of which the length is divisible by some power of 2. We first claim that the language L={a^n l n is even}. The proof is simple, by definition, L={(aa)*}∪{(aaaa)*}∪…, notice that {$(a^4)^*$} is a subset of {$(a^2)^*$}, and so on. Hence L={a^n l n is even}. So now we can write a regular expression for the language: L=(aa)*. Which also proves that L is regular. Q.E.D.

Q3:

proof:

We prove by contradiction. Assume that L is regular, then by pumping lemma, there exists an integer p such that all strings, of which the lengths are greater than p, can be divided into three substrings x,y & z with the properties: lxyl<p, y is nonempty, & xy^kz is also in L for all k=1, 2, 3, … Let p=2^m, then consider the partition xyz, such that lxl=a, lyl=b>0, and a+b=2^m, then lxyzl=p, now consider lxyyzl=a+b+b=a+2b=p+b, which is not necessarily equal to a power of 2, which also means that xyyz is not in L, contradiction.

Therefore, L is not regular. Q.E.D.

Q4:

Proof:

We prove by contradiction. Suppose L is regular, then by pumping lemma, there exists an integer m such that all strings, of which the lengths are greater than p, can be divided into three substrings x,y & z with the properties: lxyl<m, y is nonempty, & xy^kz is also in L for all k=1, 2, 3, … Let m be such an integer, consider the string w=a^mb^{m+1}, let xyz be one of w's partition, and x=a^p, y=a^q, where q>1, and a+b=m, then consider xyyz=$a^{p+q+q}b^{m+1}$=$a^{m+q}b^{m+1}$, since q>1, so m+q>m+1, which indicates that xyyz is not in L, contradiction.

Therefore, L is not regular. Q.E.D.

Q5:

Proof:

1. If L is regular, then by closure properties, L* is also regular;
2. If L is not regular, then

References

Mange, J. (2008). *DFA Minimization*. Retrieved at: https://www.cs.wmich.edu/elise/courses/cs6800/DFA_Minimization.ppt