Assignment 1

CMPUT 474

Pranav Wadhwa
1629510

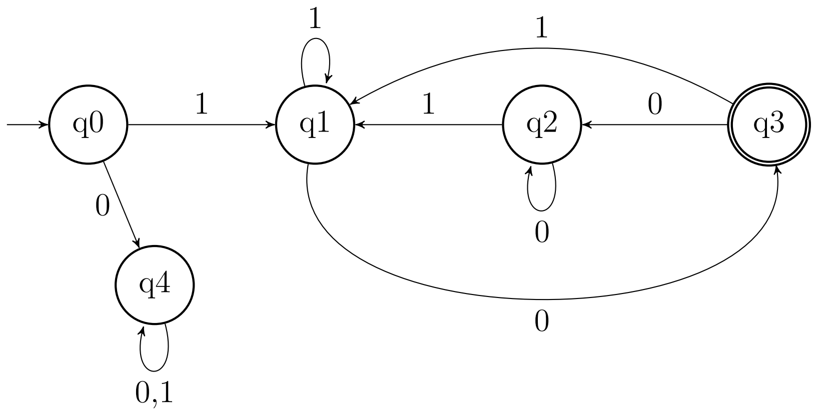
January 30, 2022

1. How many finite automata are there? How many regular expressions are there? How many regular languages are there? (Give a separate answer for each)

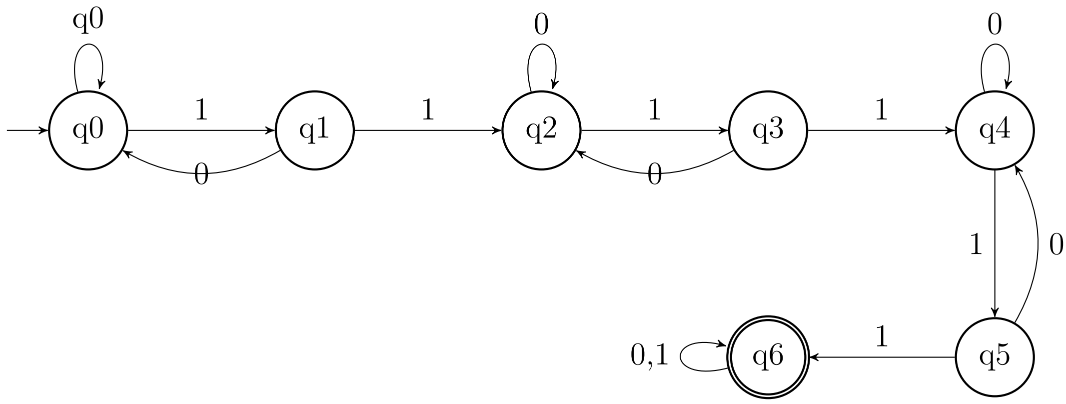
> **Solution:**
>
> Every finite automata can be described as a string. So the set of automatas are subsets of strings. There are uncountable infinite number of strings. So there are *countable infinite* finite automata.
>
> Since there are countably infinite finite automata and each finite automata is decribed by a regular expresion then there are also *countable infinite* regular expressions.
>
> Since each regular expression describles a regular language and each regular language has a regular expression then there are *conutable infinite* regular languages.

2. Give state diagrams for DFAs that recognize the following languages. Assume that alphabet is $\Sigma = \{0, 1\}$ in each case.
   (a) Strings that begin with 1 and end with 10
   (b) Strings that contain at least three occurrences of 11
   (c) Strings without leading 0s
   (d) Strings of length at most 4
   (e) Strings containing 10 and 11
   (f) Strings with length divisible by 4

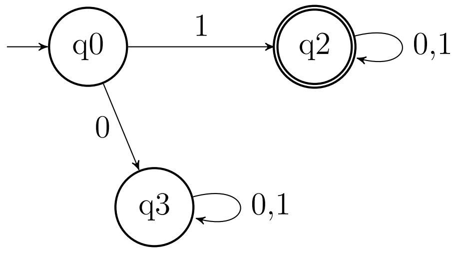
> **Solution:**
>
> (a) Strings that begin with 1 and end with 10



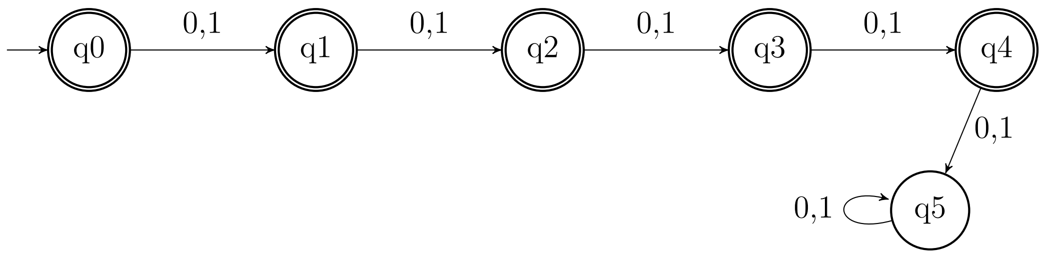
(b) Strings that contain at least three occurrences of 11



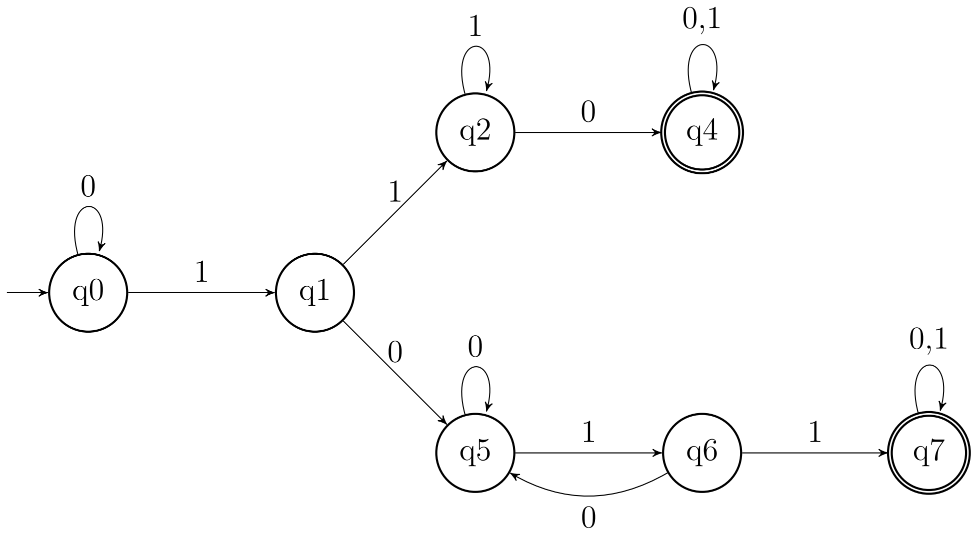
(c) Strings without leading 0s



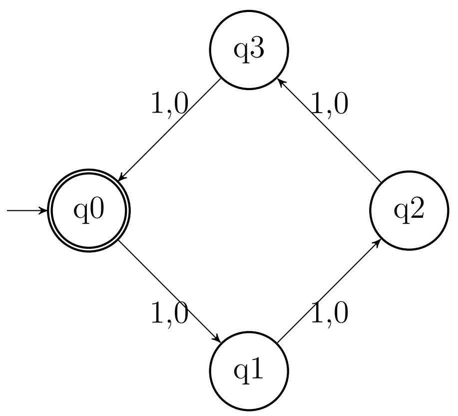
(d) Strings of length at most 4



(e) Strings containing 10 and 11

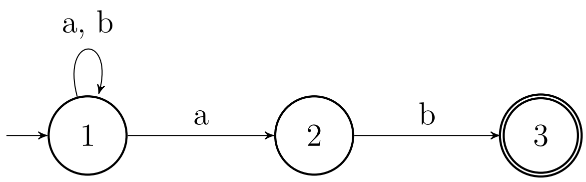


(f) Strings with length divisible by 4

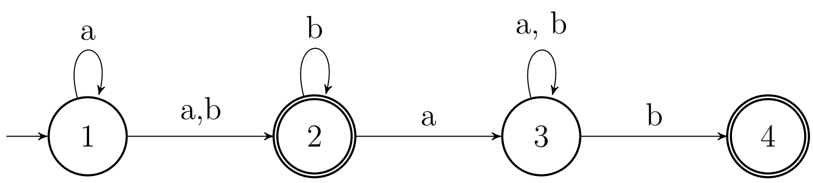


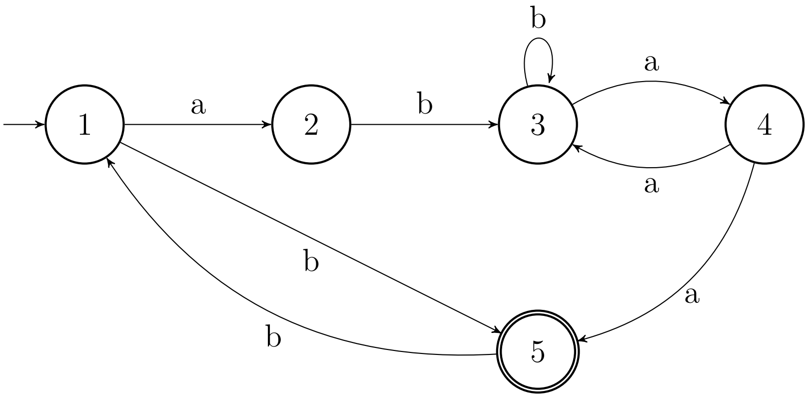
3. For each of the following NFAs, draw a DFA that accepts the same language.

(a)

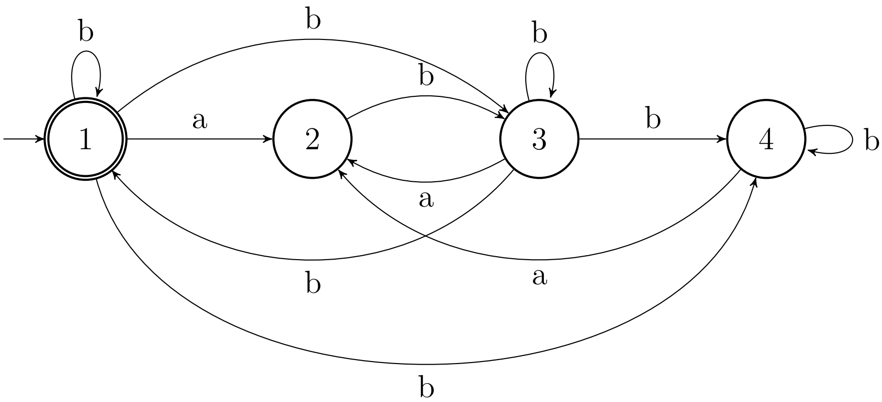


(b)





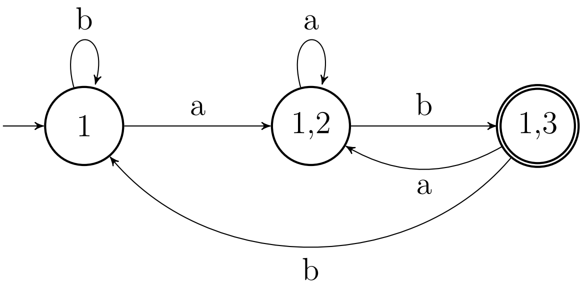
(c)



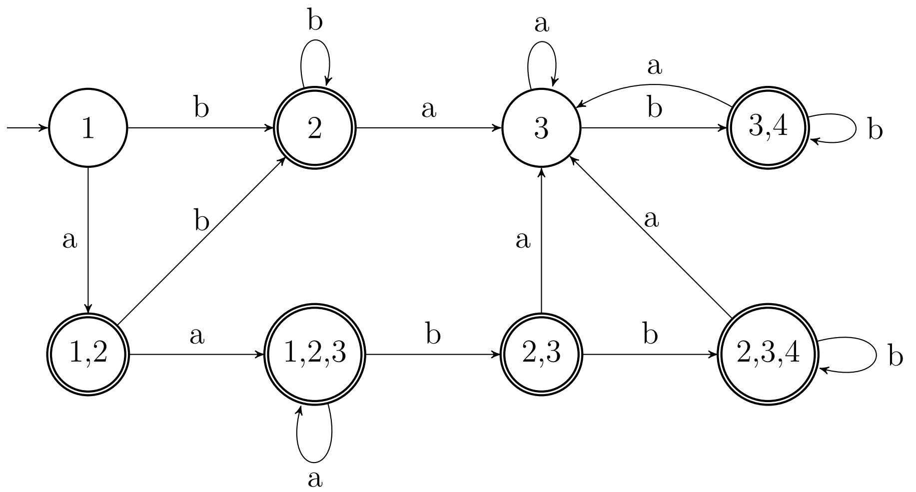
(d)

Solution:

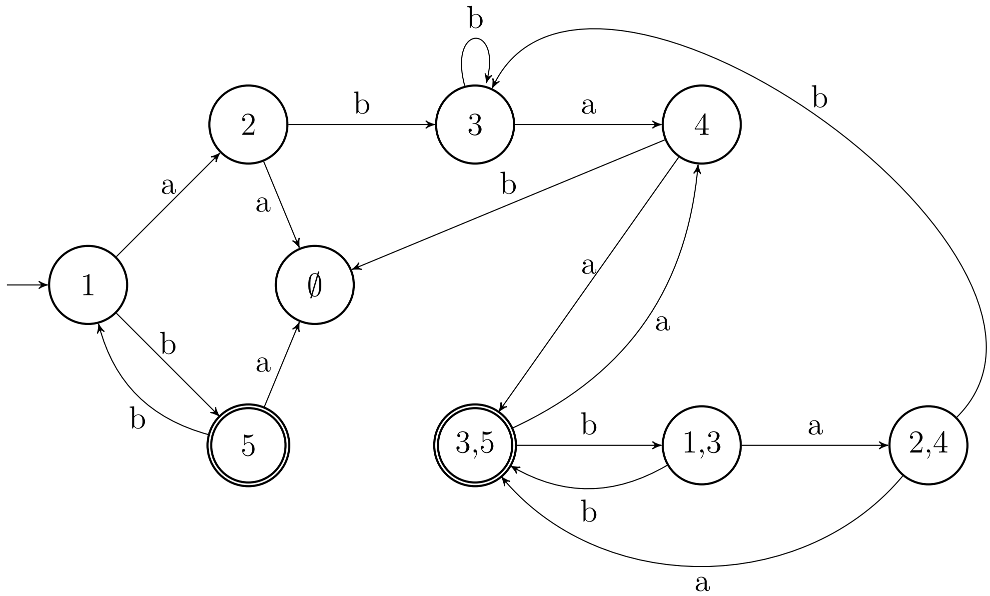
(a) DFA for part a)



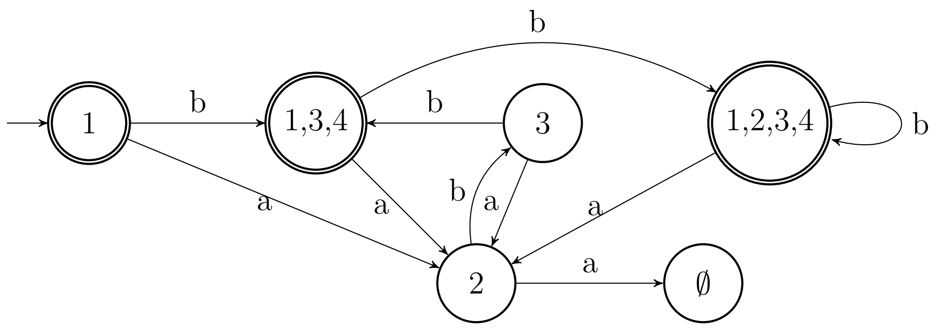
(b) DFA for part b)



(c) DFA for part c)



(d) DFA for part d)



4. Let the alphabet be $\Sigma = \{0, 1\}$. Write a regular expression for the language consisting of

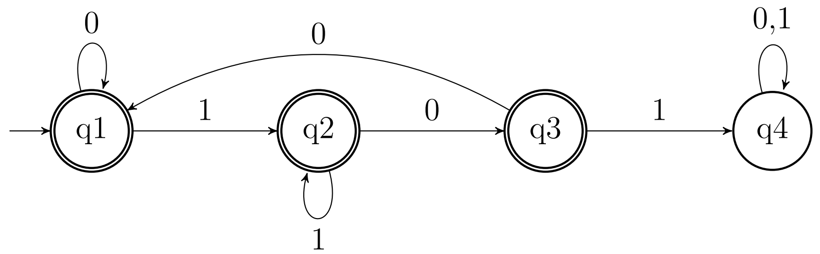
strings that do not contain 101. Prove your claim.

Solution:

Idea: To write an regular expression that does not contain the string 101, every 1 need to be followed by 2 0s.
Sol:
The regular expression that does not contain the string 101

$$0^*(1^*00^*)^*1^*0^*$$

Proof: We can create a DFA that recognizes this expression.



5. Let the alphabet be $\Sigma = \{0, 1, +, =\}$ and consider the language

$$ADD = \{x = y + z | x, y, z \text{ are binary integers and } x \text{ is the sum of } y \text{ and } z\}$$

Show that ADD is not regular.

Solution:

Idea: We see that the language can only contain one of =, + each. We will use this condition to form some form of a binary equation that makes the equation incorrect.

Proof: Let ADD be the language

$$ADD = \{x = y + z | x, y, z \text{ are binary integers and } x \text{ is the sum of } y \text{ and } z\}$$

We use the pumping lemma to prove ADD is not regular. The proof is by contradiction.

Assume to the contrary that ADD is regular. Let p be the length given by the pumping lemma. We choose $s = 1^{p+1} = 10^p + 1^p$ to be a string and s is a member of ADD. As s is a member of ADD, $|s| \geq p$ so the pumping lemma guarantees that s can be split into $s = xyz$, where for any $i \geq 0$ the string xy^iz is in ADD.

Now we observe in s that the first p are all before the = sign then y must consists of all 1 s. We know through the pumping lemma that x and z can be ϵ. So we take $x = \epsilon$ and take $i = 0$. Then $s \notin ADD$, as that expression in not an equation at all as there is nothing before the = sign.

Another way to prove it would be to look at the string $s = xz$. Here the left hand side and the right hand sign of the equation would not be equal, making the equation not true.

Thus, there is a contradiction and ADD is not regular.

6. Let

$$\Sigma_3 = \{ \begin{bmatrix}0\\0\\0\end{bmatrix}, \begin{bmatrix}0\\0\\1\end{bmatrix}, \begin{bmatrix}0\\1\\0\end{bmatrix}, \ldots, \begin{bmatrix}1\\1\\1\end{bmatrix} \}$$

That is, Σ_3 contains all 3 columns of 0*s* and 1*s*. A string of symbols in Σ_3 goves three rows of 0*s* and 1*s*. Consider each row to be a bianry number *ordered from the least signifcant bits on the left to the most signifcant bits on the right* (that is, reversed from the normal convention). Let

$$B = \{w \in \Sigma_3^* | \text{ the bottom row of } w \text{ is the sum of the top two rows}\}$$

For example,

$$\begin{bmatrix}1\\1\\0\end{bmatrix} \begin{bmatrix}1\\0\\0\end{bmatrix} \begin{bmatrix}0\\0\\1\end{bmatrix} \in B, \text{but} \begin{bmatrix}1\\0\\1\end{bmatrix} \begin{bmatrix}0\\0\\1\end{bmatrix} \notin B.$$

Show that B is regular.

Solution:

To show that B is regular we can draw a NFA that recognizes B.

Idea: Since the alphabet only recognizes columns 0s and 1s. We will have to keep count of the carry and take it over the next string. We can easily count it using states as the carry over bit can only be 0 or 1. So there will be 2 state q_0 and q_1 in the NFA which will only take a subset of strings from Σ_3 and rest would not be accepted.

Proof: Let M be a NFA that recognizes B such that $M = (Q, \Sigma_r, \delta, q_0, F)$.

1. $Q = \{q_0, q_1\}$
2. $\Sigma_3 = \{ \begin{bmatrix}0\\0\\1\end{bmatrix} \begin{bmatrix}0\\0\\1\end{bmatrix}, \ldots, \begin{bmatrix}1\\1\\1\end{bmatrix} \}$
3. Start State: q_0
4. $F = q_0$
5. δ is given by:

$$\delta(q_0, a) = q_0 \text{ if } a = \begin{bmatrix}0\\0\\0\end{bmatrix}, \begin{bmatrix}0\\1\\1\end{bmatrix}, \begin{bmatrix}1\\0\\1\end{bmatrix}$$

$$\delta(q_0, a) = q_1 \text{ if } a = \begin{bmatrix}1\\1\\0\end{bmatrix}$$

$$\delta(q_q, a) = q_1 \text{ if } a = \begin{bmatrix}1\\0\\0\end{bmatrix}, \begin{bmatrix}0\\1\\0\end{bmatrix}, \begin{bmatrix}1\\1\\1\end{bmatrix}$$

$$\delta(q_1, a) = q_0 \text{ if } a = \begin{bmatrix}0\\0\\1\end{bmatrix}$$

Rest of the string not mentioned above would not be accepted by M.

For example: We can see that the string $\begin{bmatrix}1\\0\\1\end{bmatrix}\begin{bmatrix}0\\0\\1\end{bmatrix}$ would not be accepted by M as after accepting $\begin{bmatrix}1\\0\\1\end{bmatrix}$ the machine is is at state q_0. There is no transition state for $\begin{bmatrix}0\\0\\1\end{bmatrix}$ from q_0 so the string is not accepted.

7. Suppost A is a regular language. Show that the language

$$B = \{x|\text{there is some } y \text{ such that } |x| = |y| \text{ and } xy \in A\}$$

is also regular.

Solution:

Idea: Since xy is a concatenation of 2 strings. We can use the proof of clousure under concatenation.

Proof: Assume that A is a regular language and $B = \{x|\text{there is some } y \text{ such that } |x| = |y| \text{ and } xy \in A\}$. Let xy be a string such that $xy \in A$, $|x| = |y|$ and $x \in B$. We want to show that B is regular.

Lets assume that B is not a regular language. Under the proof of closure any expression $R = R_1 \circ R_2$ where $R_1$and R_2 are regular expression would result in a regular expression R. Since B is not regular then x is not a regular expression. But $x \circ y \in A$ which is a regular language. This is a contradiction under closure of concatenation. Hence B is also regular.

8. (a) Let $A = \{1^k y | y \in \{0,1\}^* \text{ and } y \text{ contains at least } k\ 1s, \text{for } k \geq 1\}$. Show that A is regular.

   (b) Let $B = \{1^k y | y \in \{0,1\}^* \text{ and } y \text{ contains at most } k\ 1s, \text{for } k \geq 1\}$/ Show that B is not regular.

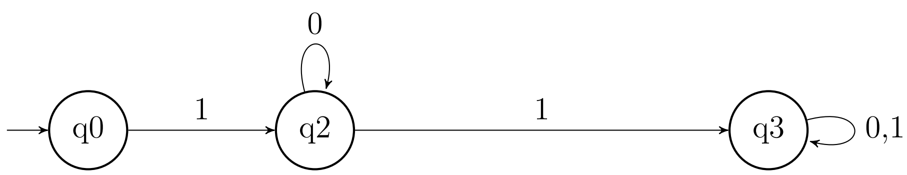
Solution:

(a) *Idea*: Since the question doesn't restrict the value of k we can set $k = 1$. This means that every string should start with 1 and then concatenate $1y$ so that the restriction is fulfilled.

A regular expression describing A:

$$10^*1\{0,1\}^*$$

A DFA describing the regular expression.



(b) Let $B = \{1^k b | b \in \{0,1\}^*$ and b contains at most k $1s$, for $k \geq 1\}$/ Show that B is not regular.

We will use the pumping lemma to show that B is not regular.

Assume the contrary that B is regular. Let p be the pumping lenth given by the pumping lemma. Let s be a string $s = 1^p 0 1^p$ and $s \in B$. The lemma guarantees that s can be split into 3 pices $s = xyz$, where for any $i \geq 0$ $xy^i z \in B$ and $|xy| \leq p$. Clearly, y contains the substring of 1^p to satify the condition. Let the $n = |y|$. The string $xz = 1^{(p-n)} 0 1^p = 1^{p-n} b$ where b contains p $1s$. Since $p - n < p$ the string xz is not in B which is a contradiction. So B is not regular.

9. Every regular expression specifies a regular language over string from a finite alphabet Σ. However, the set of regular expressions, itself is a language over strings from an expanded alphabet $\Sigma \cup \{\epsilon, \emptyset, (,), \cup, \circ, *\}$. Is the set of regular expression a regular language. Explain your answer.

10. Is there a *universal* finite automation? That is, is there a single finite automaton, say M, such that given a string $s_N w$, where s_N is a string that describes a finite automaton N and w is an input string, M accepts $s_N w$ if and only if N accepts w? Explain your answer.

Solution:

Assume that there is universal finite automation M described by this 5-tuple $(Q, \Sigma, \delta, q_0, F)$ that takes a $s_N w$ where s_N describes a finite automaton N and w is an input string that N accepts. Here N is described as $(Q_N, \Sigma_N, \delta_N, q_{N_0}, F_n)$

We know that there are countable infinite regular languages which has a countable infinite automaton. Lets assume that the finite automation N has a total number of state $|Q_N| = n$ Let's say that in the universal automaton M there are m set of state. So $|Q| = m$. For the universal automation to give the correct result it needs to have atleast $m \geq n$. Now assumne another finite automation P which has number of state $|Q_P| = p > m + 1$ and assume a string $s_P w_P$ where s_P describes P and w_p is an input string.

But M needs $|Q| \geq |Q_p|$ to accept the given string. This casues a contradiction. Therfore, M is is not a universal finite automation.