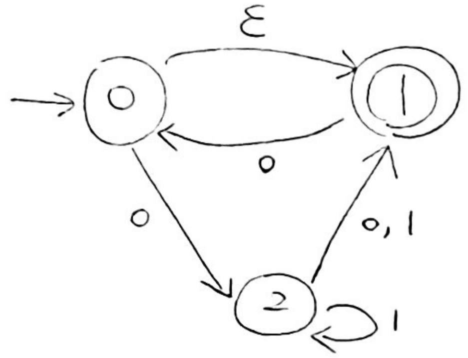
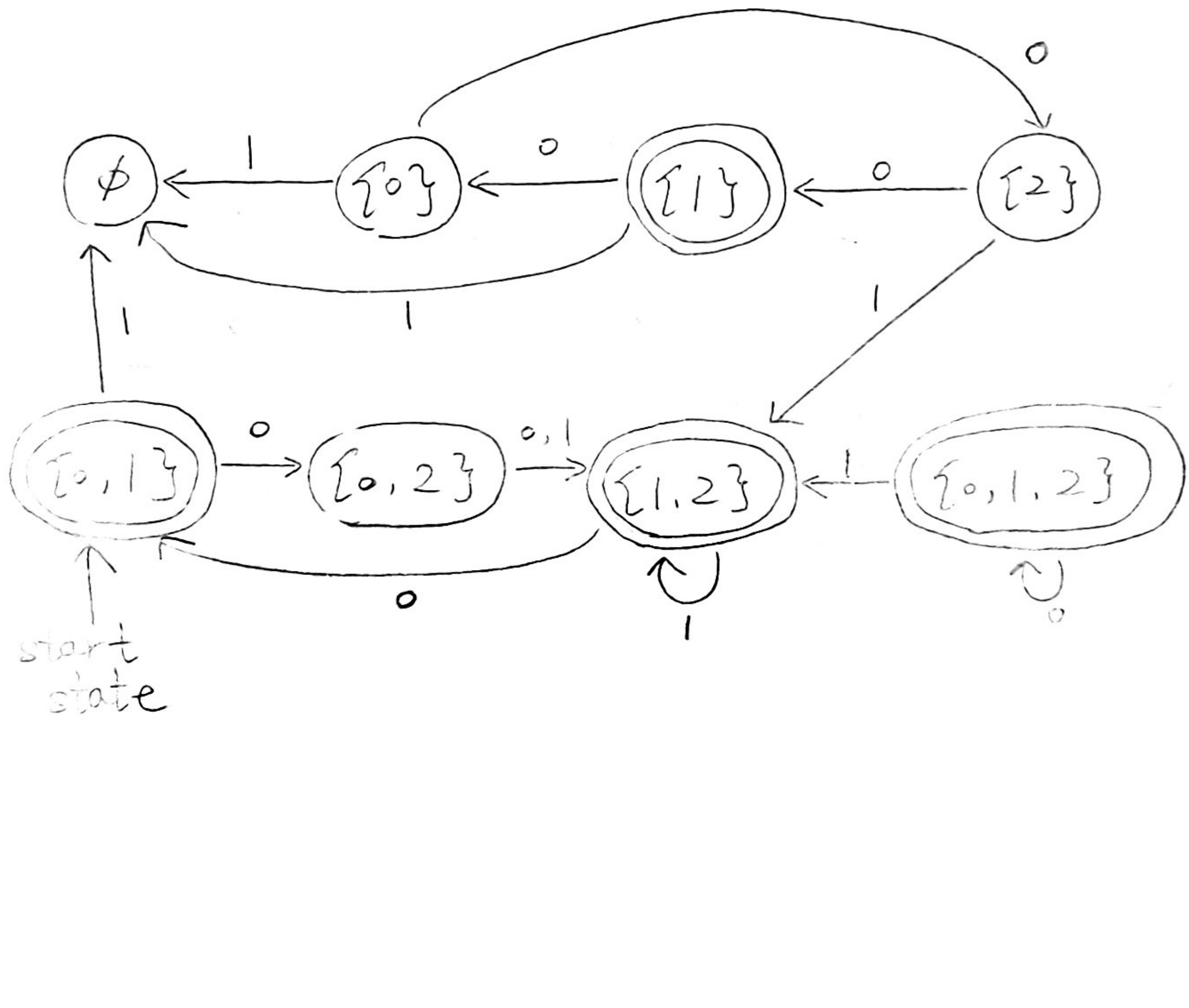
CS 181 HW 3

Problem 1 Create a DFA that is equivalent to the NFA



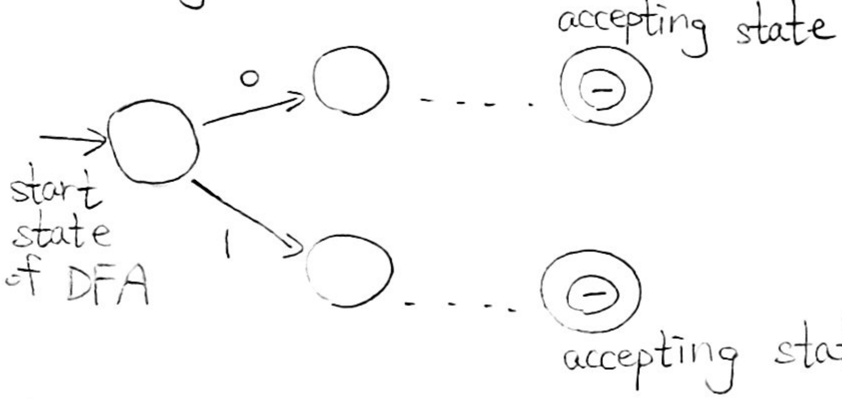


Problem 2a)

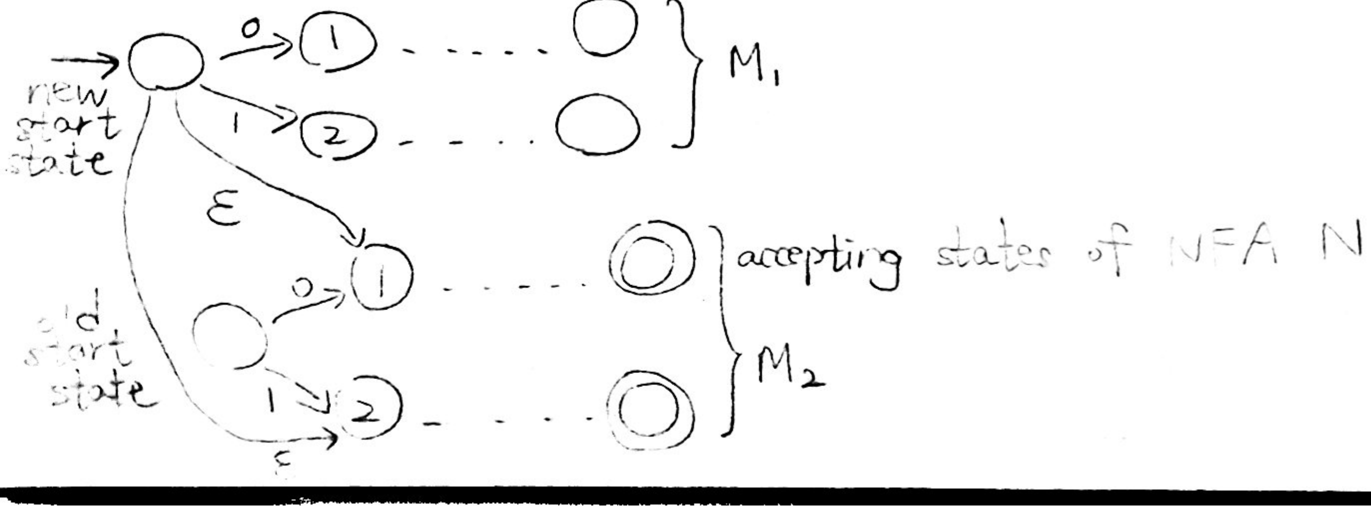
For a language L, define DropFirst(L) to be the language containing all strings that can be obtained by removing the first symbol starting from L. DropFirst(L) = $\{c : bc \in L, b \in \{0,1\}\}$. Show that the class of regular languages is closed under DropFirst. If L is accepted by a DFA, then so is DropFirst.

Proof :

This is the original DFA without dropping the first symbol starting from L.



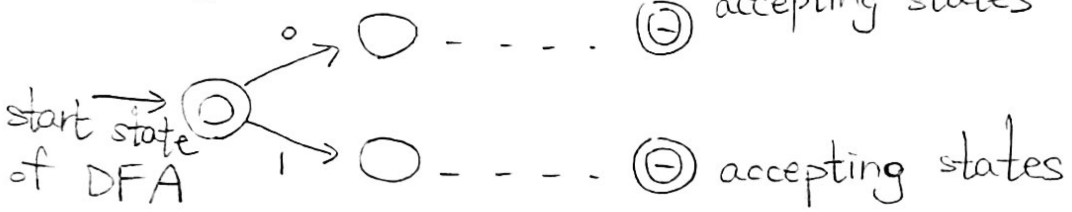
This is the modified NFA after adding appropriate ε transition from the start & adding a new state instead of reusing the old start state.



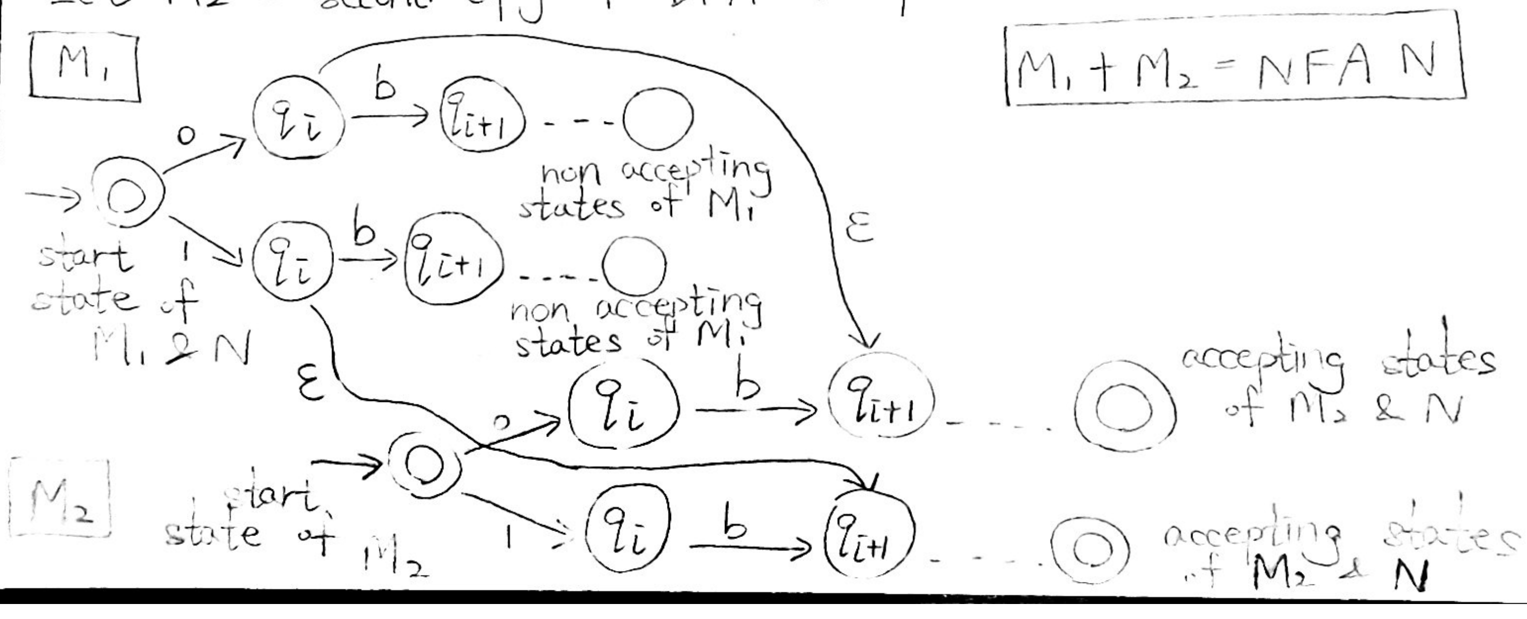
Given DFA M accept L. I drew out an NFA N that accepts DropFirst(L) which is the language containing all strings that can be obtained by removing the first symbol of a string in L. First, we start with two disjoint copies of M which are M_1 & M_2. The start state of the NFA N is the same as the start state of M_1. The accepting states of N is the same as the accepting states of M_2. For the first b transition (0 or 1) in M_1, we will take the ε transition fom M_1 to move to the state 1 or state 2 in M_2. The act of this simulate reading symbol b (0 or 1) to shift to M_2.
After that, we just continue the path as if we were still in M_1 before shifting.
The shift happens once from the very beginning, so we can drop the first symbol of a string L in order for the new string to be accepted.
According to class theorem, for every NFA N, $\exists$ a DFA D' such that $N(x) = D'(x)$ $\forall x$. Therefore, there is an equivalent DFA D' for our modified NFA N.
So, if L is accepted by a DFA, then so is DropFirst(L).

Problem 2b) Define DropOut(L) to be the language containing all strings that can be obtained by removing one symbol from a string in L. Thus, DropOut(L) $= \{ac : abc \in L,\ a, c \in \{0,1\}^*,\ b \in \{0,1\}\}$.
Show that the class of regular languages is closed under Dropout. That is if L is accepted by a DFA, then so is DropOut(L).

Proof:
This is the original DFA without removing one symbol from a string in L.



This is the modified NFA after making 2 copies & adding appropriate ε transitions.
Let M_1 = first copy of DFA accept L
Let M_2 = second copy of DFA accept L



Given DFA M accept L. I drew out an NFA N that accept DropOut(L) which is the language containing all strings that can be obtained by removing one symbol from a string in L. First, we start with two disjoint copies of M which are M_1 & M_2. The start state of the NFA N is same as the start state of M_1. The accepting states of N is the same as the accepting states of M_2. Both DFA M_1 & M_2 will keep their original transitions, the path to each of their own accepting states. For every b transition from a state q_i to a state q_{i+1} in M_1, I will add an ε transition from q_i in M_1 to the copy of q_{i+1} in M_2. If we remove symbol b, the path will first start in M_1 normally until we reach a point before taking in b, we take the new ε transition to get to the next state q_{i+1}. The act of this simulate reading symbol b & shift to M_2. After that, we just continue the path as if we were still in M_1 before shifting. The shift from M_1 to M_2 can happen only once, so we can remove the symbol b from a string in L in order for the new string to be accepted. According to the class theorem, for every NFA N, $\exists$ a DFA D' such that $N(x) = D'(x)$ $\forall x$. Therefore, there is an equivalent

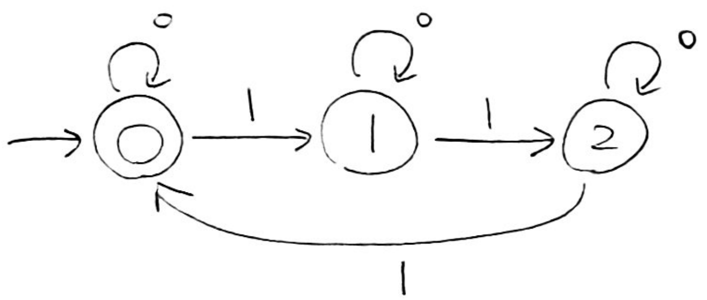
DFA D' for our modified NFA N.
So, if L is accepted by a DFA, then so is DropOut(L).

Problem 3 Find a regular expression for

$L = \{w : \text{the number of 1's in } w \text{ is divisible by } 3\}$

$$\boxed{L = \{0^*10^*10^*10^*\}^*}$$

The following diagram can accept all binary strings with number of 1's is divisible by 3



remainder 0, 1, 2

Problem 4a

Show the following is not regular :

$F : \{0,1\}^* \to \{0,1\}$ defined by $G(x) = 1$ if $\sum_{j=0}^{|x|} x_j \geq |x|/4$ and $G(x)$ otherwise. (at least a quarter of the elements of x are 1)

Proof by contradiction :

Suppose that $\mathcal{L}$ is regular, So by pumping lemma, $\exists P > 0$ st. $\forall s \in \mathcal{L}$ where $|s| \geq P$, we can write $s = xyz$ where

(a) $xy^i z \in \mathcal{L}$, $\forall i \geq 0$

(b) $|y| > 0$

(c) $|xy| \leq P$

Let's consider the string $0^P 1^P 0^P \in \mathcal{L}$

By above, we can make $0^P 1^P 0^P = xyz$

By property c), $|xy| \leq P \Rightarrow xy = 0^m$ for some $m \leq P$,

By property b), $|y| > 0 \Rightarrow y = 0^j$ for some $j > 0$

(the number of 1's that z contains is P)

With property a), If we consider $i = 200p$, $s = xy^{200P}z$, the total number of zeros in the string is at least

$= (p + p - 1) + 200p = 202p - 1$.

The total number of ones in the string is P.

Therefore $202p - 1 \neq P$, so $xy^{200P}z \notin \mathcal{L}$.

Therefore, there is a contradiction occurs under property a. So, $\mathcal{L}$ is not a regular expression. $\blacksquare$

Problem 4b

Show the following is not regular:

$F: \{0,1\}^* \rightarrow \{0,1\}$ defined by $F(x) = 1$ iff $x = 1^{3^i}$ for some $i > 0$.

Proof by contradiction: Suppose that $\mathcal{L}$ is regular, So, by pumping lemma, $\exists P > 0$ st. $\forall s \in \mathcal{L}$ where $|s| \geq P$, we can write $s = xyz$ where

(a) $xy^iz \in \mathcal{L}$, $\forall i \geq 0$

(b) $|y| > 0$

(c) $|xy| \leq P$

Let's consider the string 1^{3^P}. Then, we have $1^{3^P} = xyz$

By property b & property c $\Rightarrow$ $y = 1^m$ where $0 < m \leq P$

By property (a) $\Rightarrow$ $xy^2z = 1^{3^P + m} \in \mathcal{L}$

$\Rightarrow 3^P + m$ is a perfect cube

Now, we want to show $3^P + m$ cannot be a perfect cube

Show $3^P < 3^P + m < 3^{P+1}$

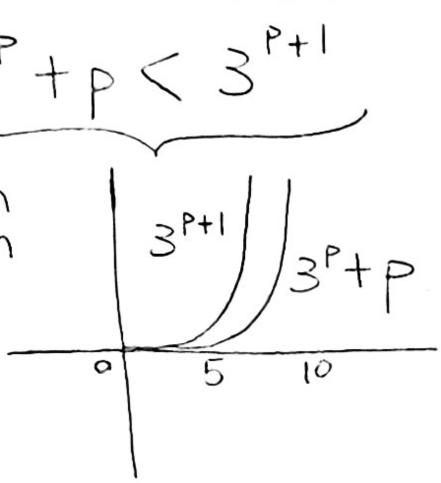
consecutive perfect cube

since $m > 0$: $3^P + m > 3^P$

since $m \leq p$: $3^P + m \leq 3^P + P$

$$3^p + m \leq \underbrace{3^p + p < 3^{p+1}}$$

sketch of graph



$\Rightarrow$ $3^p + m$ lies strictly between 2 adjacent perfect cube & so $3^p + m$ is not a perfect cube.

$$\Rightarrow xy^2z = 1^{3^p + m} \notin \mathcal{L}$$

Therefore, there is a contradiction $\Rightarrow$ $\mathcal{L}$ is not a regular expression $\blacksquare$

Problem 4C

Show the following is not regular:

$ADD = \{0^m 1 0^n 1 0^{m+n} : m, n \geq 1\}$

Proof by contradiction: Let $ADD = \mathcal{L}$.

Suppose that $\mathcal{L}$ is regular, So, by pumping lemma, $\exists P > 0$ st. $\forall s \in \mathcal{L}$ where $|s| \geq P$, we can write $s = xyz$ where

(a) $xy^iz \in \mathcal{L}$, $\forall i \geq 0$

(b) $|y| > 0$

(c) $|xy| \leq P$

Let's consider the string $0^P 1 0^P 1 0^{2P}$

By above, we can make $0^P 1 0^P 1 0^{2P} = xyz$

By property c), $|xy| \leq P \Rightarrow xy = 0^m$ for some $m \leq P$, on the other hand, the number of 1's that z contains is 2.

By property b), $|y| > 0 \Rightarrow 0^j$ for some $j > 0$

With property a), If we consider $i \geq 2$ for $s = xy^iz$, there will be more zeros on one side & it does not equal to the other side. For example, this will require adding $> m+n$ zeros at the end of string.

Therefore, $s = xy^iz$, with $i \geq 2$ $\notin \mathcal{L}$.

There is a contradiction occurs under property a.

So, ADD is not a regular expression.