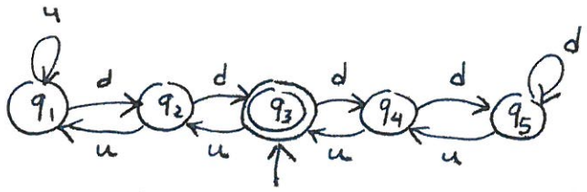
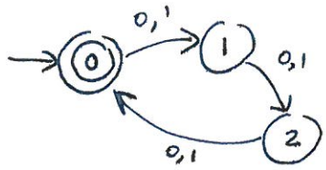
HW 1 Solution

Problem 1

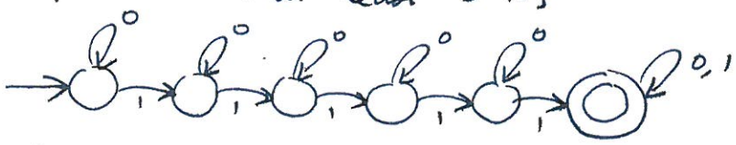


Problem 2

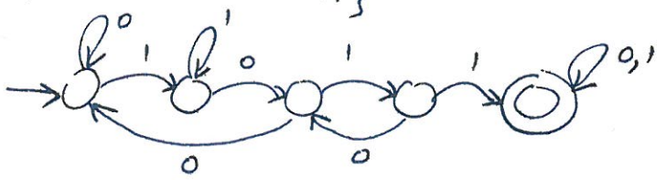
- $\{w \mid |w| \equiv 0 \bmod 3\}$



- $\{w \mid w$ contains at least 5 1s$\}$

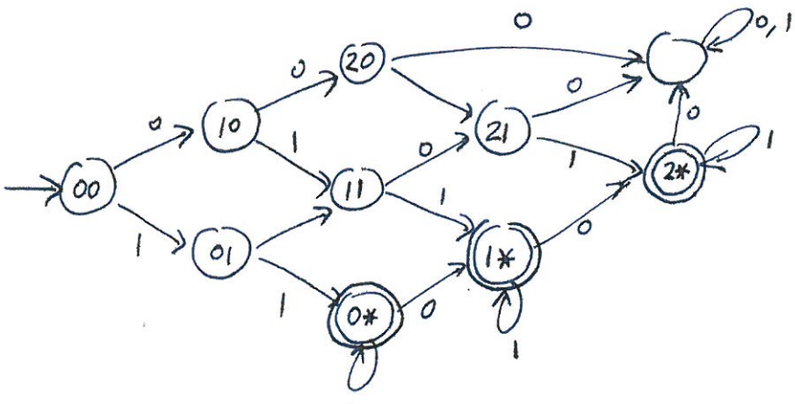


- $\{w \mid w$ contains 1011$\}$



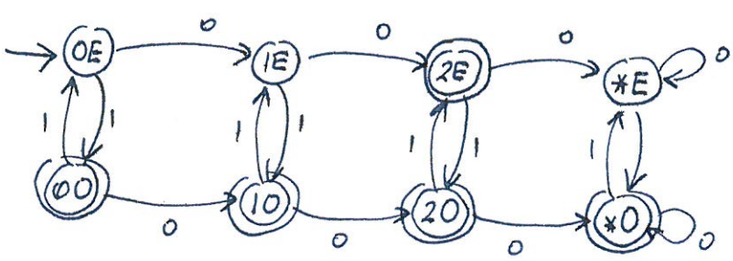
- $\{w \mid w$ contains at least two 1s and at most two 0s$\}$

let's keep track of the # of 0s and 1s.
Each state has 2 counts, * means ≥ 2

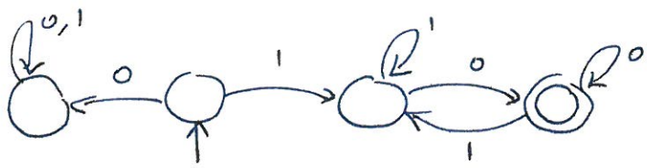


- $\{w \mid w$ contains exactly two 0s or an odd # of 1s$\}$

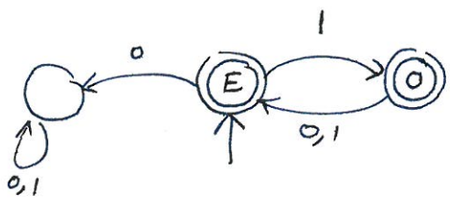
let's keep track of 0s and 1s. Here * means > 2, E and O mean even and odd respectively.



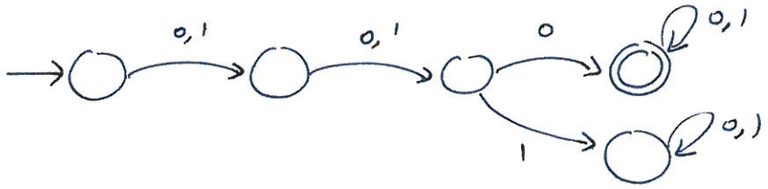
- $\{w \mid w$ starts with 1 and ends with $0\}$



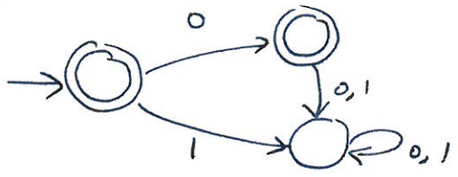
- $\{w \mid$ every odd position in w is $1\}$



- $\{w \mid |w| \geq 3$ and its 3rd symbol is $0\}$

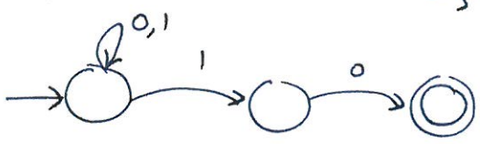


- $\{\varepsilon, 0\}$



Problem 3. Construct NFAs

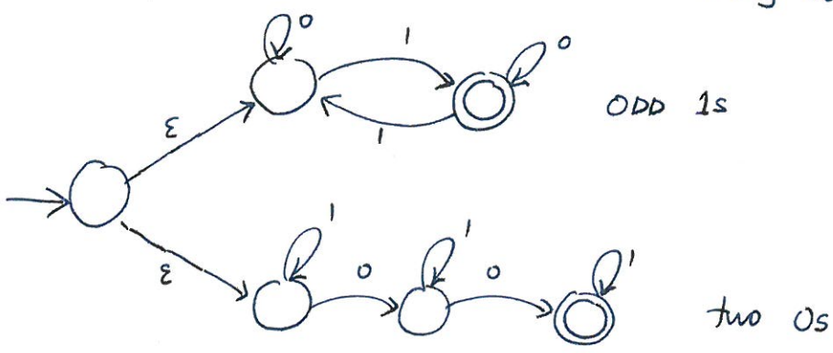
- $\{w \mid w$ ends with $10\}$ 3 states



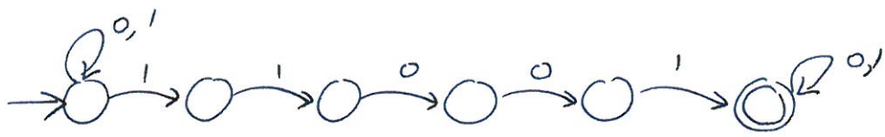
- $\{w \mid w$ contains $1011\}$ 5 states

See Problem 2 part (3)

- $\{w \mid w$ contains an odd # 1s or exactly two 0s$\}$ 6 states

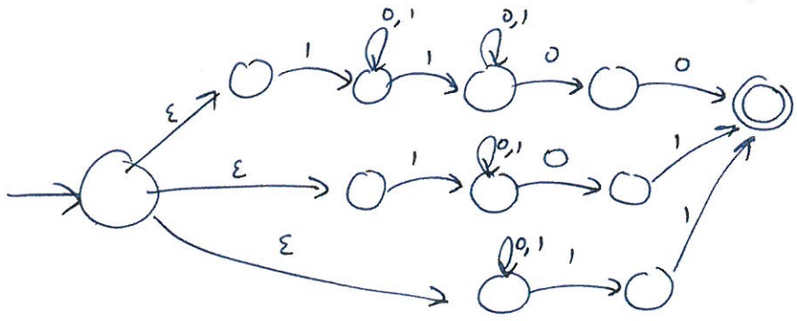


- $\{w \mid w \text{ contains } 11001\}$

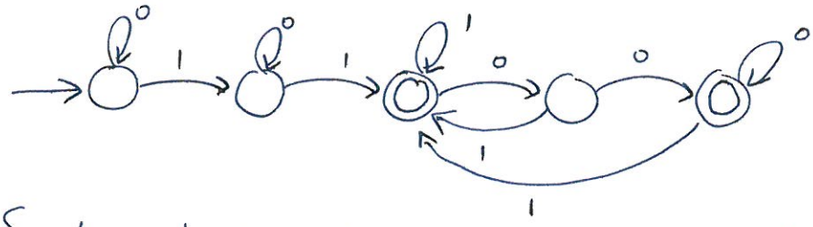


- $\{w \mid w \text{ contains at least two 1s and does not end in } 10\}$

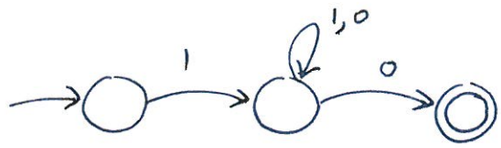
we can end in either 00, 01, or 11



There is a smaller DFA:



- $\{w \mid w \text{ begins with 1 and ends with 0}\}$



Problem 4.

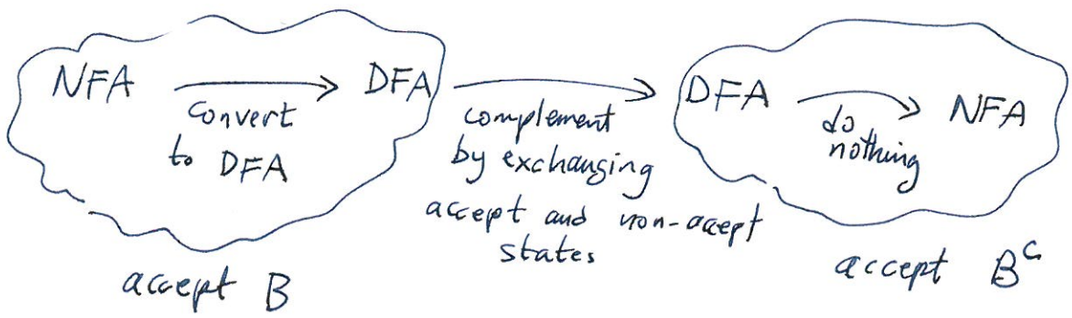
(a) Given a DFA M such that $B = L(M)$, if we exchange the accept and non-accept states in M, we obtain a DFA M'. Now, it's easy to show that

$$w \in B \iff M' \text{ does not accept } w.$$

- If $w \in B \Rightarrow M$ accept w. This means following transitions in M by reading the symbols of w puts M in an accept state. Therefore, this puts M' in a non-accept state.
- If M' does not accept w, then w puts M' in a non-accept state. This means w puts M in an accept state, so $w \in B$.

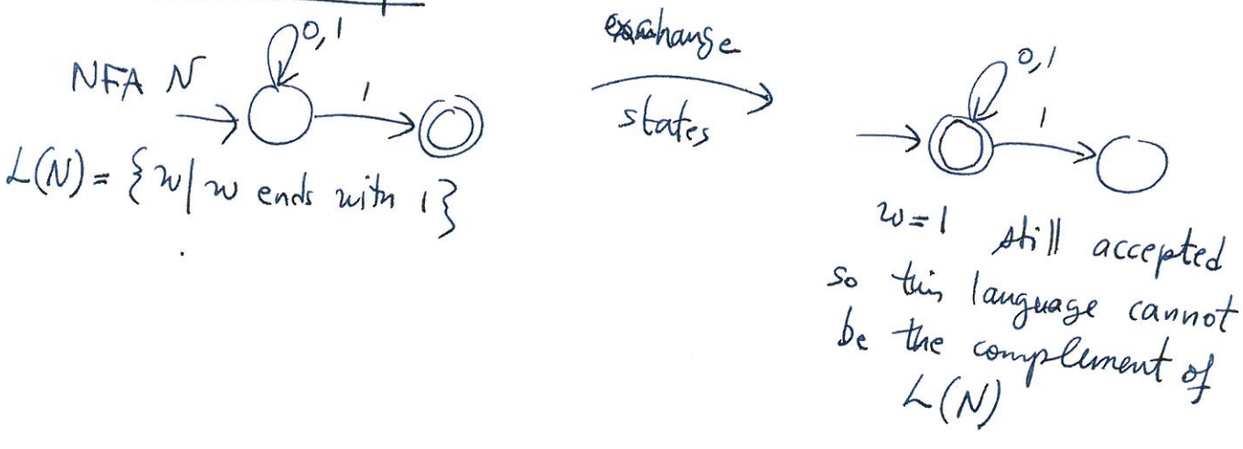
we conclude that the language of M' is the complement of B and, therefore, regular languages are closed under complement.

(b) If an NFA accepts input w, this means there exists a branch of computation that ends in an accept state. Some other branches of the computation could end in a non-accept state. By exchanging the accept and non-accept states, we could still have a branch of computation that accepts w, i.e. ends in an accept state. Therefore, exchanging accept and non-accept states does not produce an NFA that accepts the complement of the language. That does not imply that no such NFA exists. In fact, it does by equivalence of NFA and DFA

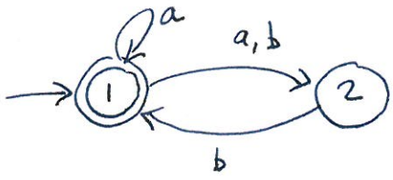


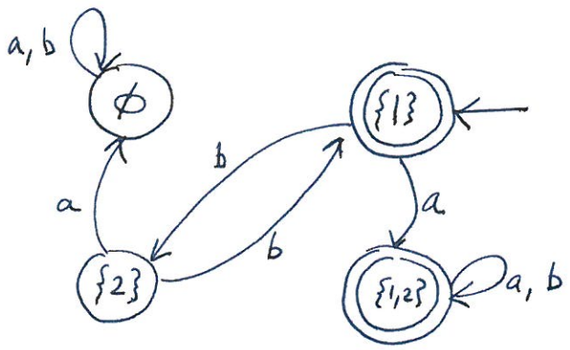
The class of languages accepted by NFA (regular languages) is still closed under complement, of course.

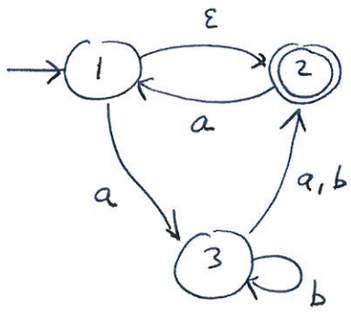
Counter example:

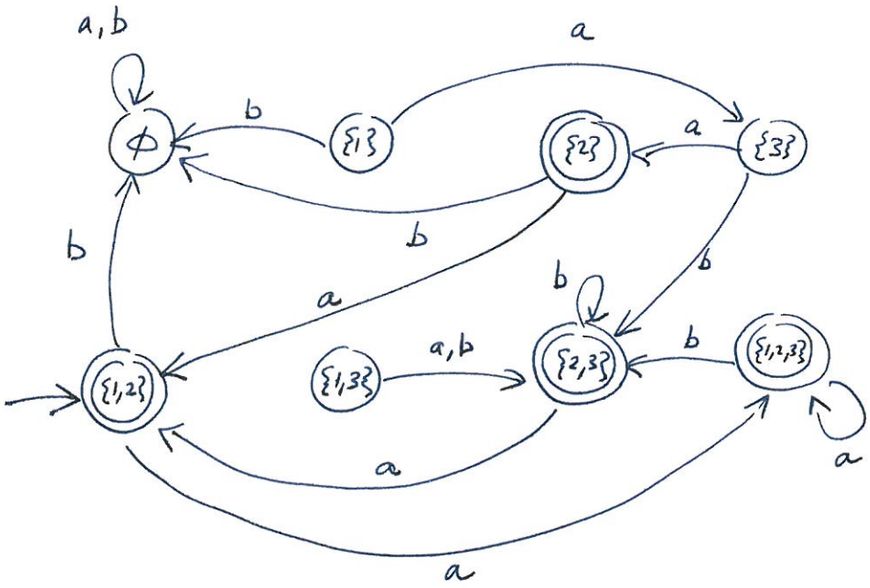


Problem 5 : Convert NFA to DFA









Problem 6.

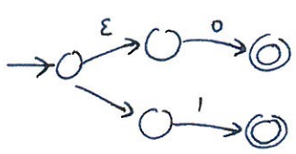
Convert regular expressions to NFAs.

$(0 \cup 1)^* 000 (0 \cup 1)^*$

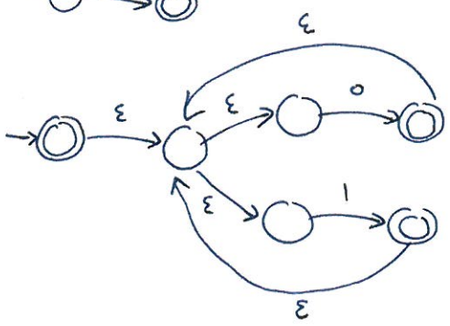
0 :

1 :

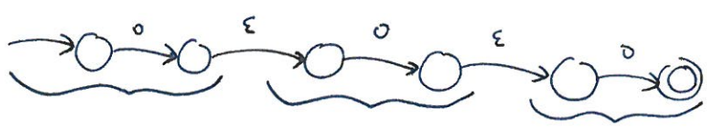
$0 \cup 1$:



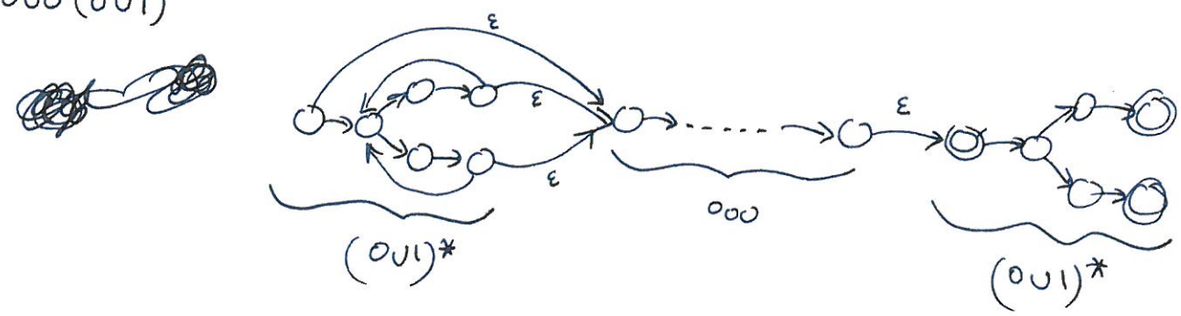
$(0 \cup 1)^*$:



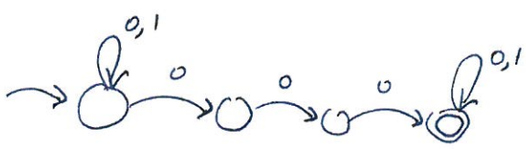
000 :



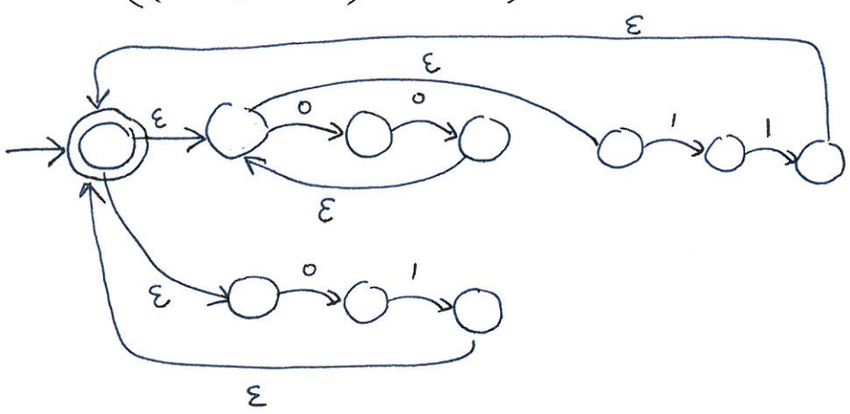
$(0 \cup 1)^* 000 (0 \cup 1)^*$



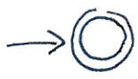
This was the systematic approach discussed in class. Often we can construct a much smaller NFA.



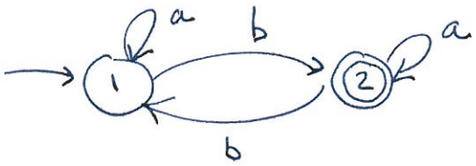
$(((00)^* 11) \cup 01)^*$

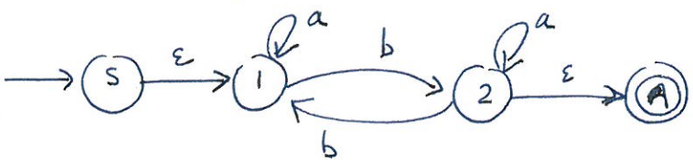


$\emptyset^* = \{\varepsilon\}$

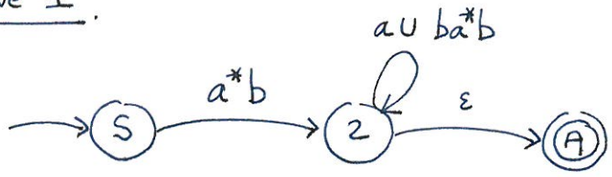


Problem 7 Convert automata to regular expressions.

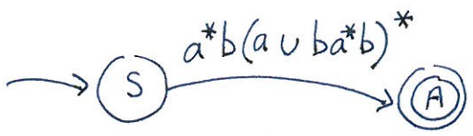


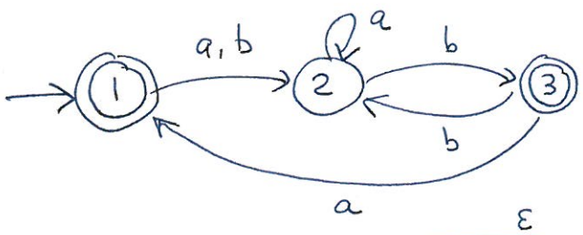


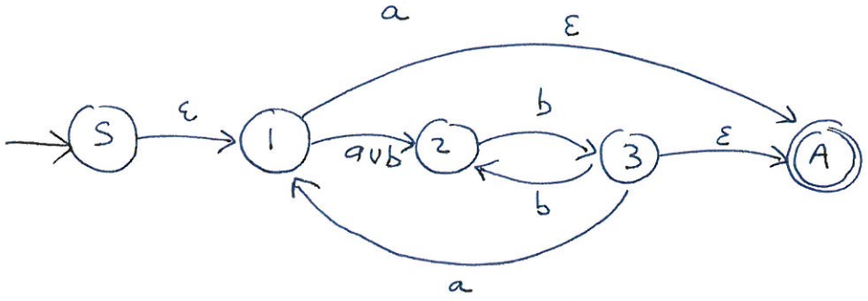
remove 1.



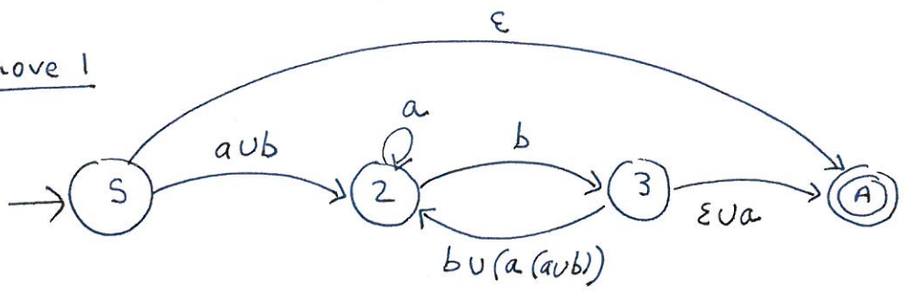
remove 2.



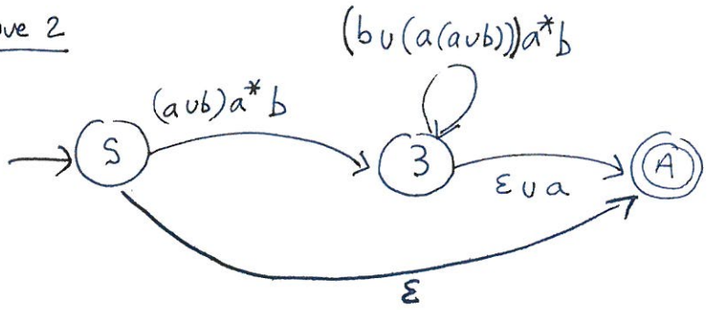




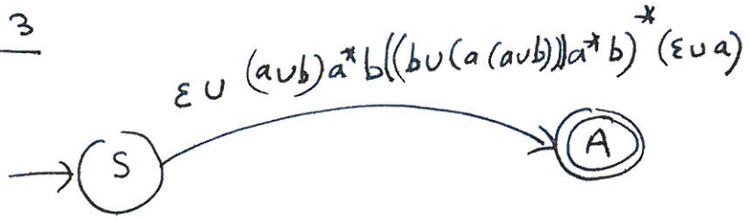
remove 1



remove 2



remove 3



Problem 8 Pumping Lemma

- $\{0^n 1^n 2^n \mid n \geq 0\}$

Let l be the critical length and consider the string $w = 0^l 1^l 2^n$.

By the pumping lemma w must be written as $w = xyz$ such that

$|y| > 0$

$|xy| \leq l$

Therefore y contains only 0s, and thus $xy^i z$ cannot be in the language for $i = 0, 2, 3, \ldots$.

- $\{ww \mid w \in \{a,b\}^*\}$

Let l be the critical length and consider the string

$$\underbrace{a^l b^l}_{w} \underbrace{a^l b^l}_{w} \quad \text{where } w = a^l b^l$$

By the pumping lemma, $a^l b^l a^l b^l = xyz$ where $|y| > 0$

$|xy| \leq l$

so y is entirely contained in the first l as. As a result $xy^i z$ cannot be in the language for $i \neq 1$.

- $\{ 1^{2^n} \mid n \geq 0 \}$

Let l be the critical length and let n be such that $2^n > l$.

$$w = \underbrace{111\ldots\ldots\ldots1}_{2^n}$$

w must be written as xyz, where

$|y| > 0$

$|xy| \leq l$

Therefore y contains a number of 1s strictly less then 2^n. Now $xyyz$ cannot be in the language because ~~$|xyyz| < 2^n$~~

$$|xyyz| = |xyz| + |y| = 2^n + |y| < 2^n + 2^n = 2^{n+1}.$$