Problem 1

Answer.

Problem 2

Answer. (a) No.

Assume $B$ is a regular language. By pumping lemma, there exists a constant $p$ such that $\forall s \in B, |s| > p$, $s$ can be broke into three sub-strings $s = xyz$ such that

- $|y| > 0$;
- $|xy| \leq p$;
- $\forall i \geq 0, xy^iz \in B$.

Consider $s = 1^p 001^p$. As $|xy| \leq p$, $y$ can be $1^n$ for any integer $n \in [1, p]$. By pumping lemma, $xy^2z$ is in $B$. However, $xy^2z = 1^{p+n} 001^p$ is not in $B$.

(b) Yes. Consider the regular expression $(0(0 \cup 1)^*0) \cup ((1(0 \cup 1)^+1))$. Assume it is language $L$. Here for expression $R$, $R^+ = R \circ R^*$.

We will show $C = L$.

For every $s = s_0 \circ s_1 \cdots s_n \in C$, the first character of $s$ is $s_0$. The last character of $s$ is also $s_n = s_0$ according to the definition of $C$. Here $s_0$ can only be 0 or 1 and $s_2 \cdots s_{n-1} \in (0 \cup 1)^+$. So $s \in L$.

For every $s = s_0 \circ s_1 \cdots s_n \in L$, we know $s_0 = s_n$. Let $w = s_0, t = s_2 \cdots s_{n-1},$ then $s = wtw^R$. So $s \in C$.

So $C = L$. Thus $C$ is a regular language.

Problem 3

Answer. For a regular language $A$, let $M = (Q, \Sigma, \delta, q_0, F)$ be the DFA that recognizes $A$.

We construct an NFA $M' = (Q', \Sigma', \delta', q'_0, F')$. Here $M'$ is a copy of $M$ with the following modification. For each state $q \in F'$, we delete all its outgoing edges.
Formally, \( M' \) is an NFA such that \( Q' = Q, \Sigma' = \Sigma, q'_0 = q_0, F' = F \). Also \( \forall q \in Q, \forall a \in \Sigma, \) if \( q \notin F \) then \( \delta'(q, a) = \{ \delta(q, a) \} \), if \( q \in F, \delta'(q, a) = \emptyset \).

We claim \( M' \) decides NOPREFIX(A).

For each string \( s \in \text{NOPREFIX}(A) \), as \( s \in A \), it corresponds to a path \( P \) from \( q_0 \) to a state \( q_e \) in \( F \). On path \( P \), there is no other accept state except the last state \( q_e \). Otherwise, assume there is another state \( q_1 \in F \) which is not the last state on \( P \), then we can find a proper prefix of \( s \) which is also in \( A \), thus \( s \) is not in \( \text{NOPREFIX}(A) \). As a result, no edge on \( P \) is deleted. So \( s \) can be accepted by \( M' \).

On the other hand, let \( s \in L(M') \). There is a path \( P \) that corresponds to \( s \) in \( M' \), starting from \( q'_0 \) and ending at a state in \( F' \). According to the construction of \( M' \), we know \( P \) is also a path in \( M \) from \( q_0 \) to a state in \( F \). So we know \( s \in A \). Also as \( M \) is a DFA, \( P \) is the only path corresponding to \( s \) in \( M \). If there is another string \( s' \in A \) which is a proper prefix of \( s \), then by reading \( s' \), \( M' \) ends at an accept state \( q_1 \in F \) on \( P \). However, in \( M' \), \( q_1 \) has no outgoing edge, thus \( P \) can not be a path ending at an accept state in \( M' \). So there is no string \( s' \in A \) which is a proper prefix of \( s \). So \( s \in \text{NOPREFIX}(A) \).

As a result, \( M' \) recognizes \( \text{NOPREFIX}(A) \). So regular languages are closed under the NOPREFIX operation. \( \square \)

**Problem 4**

**Answer.** Consider the following grammar \( G \).

- \( S \rightarrow 0S1|1A|A0 \)
- \( A \rightarrow 1A|0A|\epsilon \)

Assume the complement of the language \( \{0^n1^n|n \geq 0\} \) is \( L \). We can see \( L \) is the union of the following three distinct languages.

- \( A = \{w| \text{There is a 1 before a 0 in } w\} \)
- \( B = \{w \in 0^*1^*| \text{There are more 1’s than 0’s in } w\} \)
- \( C = \{w \in 0^*1^*| \text{There are more 0’s than 1’s in } w\} \)

For every string \( s \) generated by \( G \), we show that it is in \( L \). When applying the grammar, we know that \( S \rightarrow 1A|A0 \) has to be conducted at least once. If \( S \rightarrow 1A \) is conducted, we can see \( s \in A \cup B \). If \( S \rightarrow A0 \) is conducted, we can see \( s \in A \cup C \). As a result, \( s \in L \).

On the other hand, we claim that \( A, B \) and \( C \) can all be generated by our grammar.

For \( A \), it can be generated by

- \( S \rightarrow 0S1|1A \)
- \( A \rightarrow 1A|0A|\epsilon \).

Because \( S \rightarrow 1A \) can generate the 1 that is before some 0’s and \( A \rightarrow 1A|0A|\epsilon \) can generate any string after that 1.

For \( B \), it can be generated by

- \( S \rightarrow 0S1|1A \)
- \( A \rightarrow 1A|\epsilon \).
Because we can use $S \rightarrow 0S1$ to generate enough 0's and use $S \rightarrow 1A$ and $A \rightarrow 1A|\epsilon$ to generate the extra 1's.

According to similar reasons, for $C$, it can be generated by
- $S \rightarrow 0S1|A0$
- $A \rightarrow 0A|\epsilon$

So $A$, $B$ and $C$ can all be generated by $G$. Thus $G$ is the CFG for $L$. 

\[\square\]