Problem 1

Answer. Consider the language \( L = \{1^{k-2}\} \).

First we give a DFA \( M = (Q, \Sigma, \delta, q_0, F) \) that recognizes \( L \), where \( |Q| = k \).

Let \( Q = \{q_0, q_1, \ldots, q_{k-1}\} \) and \( \Sigma = \{0, 1\} \).

The transition function \( \delta \) is defined as follows. Let \( \delta(q_i, 1) = q_{i+1} \) and \( \delta(q_i, 0) = q_{k-1} \), for \( i = 0, 1, \ldots, k-2 \). Let \( \delta(q_{k-1}, 0) = q_{k-1}, \delta(q_{k-1}, 1) = q_{k-1} \).

Let \( q_0 \) be the start state.

There is only one path from the start state to the accept state. The path is \( (q_0, q_1, \ldots, q_{k-2}) \). So \( M \) can accept and only accept \( 1^{k-2} \). This shows that \( M \) recognizes \( L \).

On the other hand, assume we have another DFA \( M' = (Q', \Sigma, \delta', q_0', F') \) which recognizes \( L \) but only has \( k-1 \) states. Let the sequence of states (path \( P \)), following \( M' \) reading \( 1^{k-2} \), be \( p_0 = q_0', p_1, \ldots, p_{k-2} \) where \( p_{k-2} \in F' \). Consider \( q' = \delta(q_0', 0) \). We must have \( q' \neq \{p_0, p_1, \ldots, p_{k-2}\} \). Otherwise \( M' \) will accept another string which starts with a 0. Thus \( p_0, p_1, \ldots, p_{k-2} \in Q - \{q'\} \). As \( |Q - \{q'\}| = k-2 \), at least two of \( p_0, p_1, \ldots, p_{k-2} \) are equal. This shows that there is a cycle in \( P \). So \( M' \) accepts more than one string, contradicting that \( M' \) recognizes \( L \).

Problem 2

Answer. (a) Assume this language is \( L \) and it is regular. According to the pumping lemma, there exists a constant \( p \) such that \( \forall s \in L, |s| \geq p, s \) can be broke into three sub-strings \( s = xyz \) such that

- \( |y| > 0 \)
- \( |xy| \leq p \)
- \( \forall i \geq 0, xy^iz \in L \)

Let \( s = 0^p10^p \). So \( y = 0^l, 1 \leq l \leq p \). Consider \( xy^2z \). It should be in \( L \) according to the pumping lemma. However \( xy^2z = 0^{p+1}10^p, n_1 \neq p \). This is a contradiction. So \( L \) is not regular.

(b) Assume this language is \( L \) and it is regular. Let the pumping length of \( L \) be \( p \).

Consider the string \( s = 0^p10^p+p^l \). As \( p \neq p + pl \), \( s \) is not a palindrome. So \( s \) is in \( L \). By the pumping lemma, \( s \) can be broke into \( xyz \) where \( |xy| \leq p, |y| > 0 \). We know \( y = 0^l \) where \( i \) is an integer in \( [1, p] \). Also by the pumping lemma, \( xy^pl+1z \) is in \( L \). However \( xy^pl+1z = 0^p+p^l10^{p+p^l} \) which is a palindrome. This is a contradiction.

As a result, \( L \) is not regular.

Problem 3

Answer. (a) Yes. We show that \( B = L \) where \( L = 10^*1(0|1)^* \). For every \( s \in B \), let \( s = 1^k \). If \( k = 1 \), we know \( y \) has at least one 1. So \( y \in 0^*(10|1)^* \). Thus \( s \in L \). If \( k \geq 2 \), \( 1^k \) is in \( 10^*1(0|1)^* \) and \( y \) is in \( (0|1)^* \). So \( s = 1^k y \in L \). On the other hand, for any \( s \in L \), as \( s \) starts with a 1, \( s \) can be viewed as \( 1y \) where \( y \in 0^*(10|1)^* \). Thus \( y \) contains at least one 1. According to the definition of \( B \), \( s \in B \). So \( L = B \).
(b) No. We use the pumping lemma. Let $p$ be the pumping length of $C$. Let $s = 1^p01^p \in C$. By the pumping lemma, $s = xyz$ where $|xy| \leq p$. So $y = 1^l$ for some $l \in [1, p]$. By the pumping lemma, $xz \in C$. However, $xz$ does not have the form $1^k a'$ with $a'$ has at most $k'$ 1's. So $C$ is not regular.

Problem 4

Answer. (a) The grammar is as follows.

- $S \rightarrow 0A0|1A1$
- $A \rightarrow A0|A1|\epsilon$

(b) The grammar is as follows.

- $S \rightarrow 0S0|1S1|0|1|\epsilon$

Problem 5

Answer. The grammar is as follows.

- $S \rightarrow 0S0|1S1|A$
- $A \rightarrow 0B1|1B0$
- $B \rightarrow 0B0|0B1|B0|B1|\epsilon$

First we prove that every string $w = w_0w_1\cdots w_{n-1}$ generated by the grammar is in $D$.

Notice that for every rule in the grammar, the number of terminals in the right hand side is even. As a result, $w$ has even length. Assume $w = xy$. In the procedure of generating $w$, the rule $A \rightarrow 0B1|1B0$ has to be applied for at least once. This means that there exists an index $i < n/2$, such that $w_i \neq w_{n-1-i}$. Thus $x \neq y^R$. So $w \in D$.

Next we prove that every string $w$ in $D$ can be generated by the grammar.

According to the definition of $D$, there exists an index $i < n/2$, such that $w_i \neq w_{n-1-i}$ and $\forall j \leq i-1, w_j = w_{n-1-j}$. We give a process generating $w$ following the rules in our grammar. We generate the first $i-1$ characters and the last $i-1$ characters of $w$ following $S \rightarrow 0S0|1S1$. This can be done because by following $S \rightarrow 0S0|1S1$, for every string $u$, we can get $uSu^R$. Then we apply the rule $S \rightarrow A$ and $A \rightarrow 0B1|1B0$ to generate $w_i$ and $w_{n-1-i}$. After that we generate the remaining bits of $w$ by following $B \rightarrow 0B0|0B1|B0|B1|\epsilon$. This can be done because any even length string can be generated by following $B \rightarrow 0B0|0B1|B0|B1|\epsilon$. As a result, $w$ can be generated by our grammar.

This shows that $D$ is a context-free language.