1. \( L : \{ ww^R \mid w \in (a+b)^* \} \) is not a regular set.

Suppose \( L \) is regular. Then there exists a constant \( n \) such that for every string \( w \) in \( L \) such that \( |w| \geq n \), we can break \( w \) into three strings \( w = xyz \) such that:

\[
y \neq \epsilon \\
|xy| \leq n
\]

For all \( k \geq 0 \), the string \( xy^kz \) is also in \( L \).

Let \( w \) be strings of the form \( a^n b b a^n \). Because \( |xy| \leq n \), the substring \( xy \) must be in the form \( a^m \), we can pump \( y \) up and let \( k = 2 \), then the \( w \) becomes \( a^{n + |y|} b b a^n \).

This is not in the language \( L \), therefore we proved \( L \) not regular by the pumping lemma.

2. We can express this language as \( C_1 \cap C_2 \) where,

\( C_1 : \{ a^m b c^m d^m \mid n \geq 1, m \geq 1 \} \)

Grammar:

\[
S \rightarrow aMbcNd \\
M \rightarrow aMb | \epsilon \\
N \rightarrow cNd | \epsilon
\]

\( C_2 : \{ a^* b^n c^n d^* \mid n \geq 1 \} \)

Grammar:

\[
S \rightarrow aSd | bBc \\
B \rightarrow bBc | \epsilon
\]

Because \( C_1 \) forces \( a \) and \( b \), \( c \) and \( d \) to be the same length, and \( C_2 \) forces \( b \) and \( c \) to be the same length, the intersection of the two language has \( a, b, c, d \) all be the same length.
3. We can express this language as $C_1 \cap C_2$ where, 

$C_1$: $(0^n10^{n+1}1)^*(0^*1 + \varepsilon)$
Grammar: 
$$S_1 \rightarrow 0B01S \mid E \mid \varepsilon$$
$$E \rightarrow 0E \mid 1$$
$$B \rightarrow 0B0 \mid 1$$

$C_2$: $01(0^n10^{n+1}1)^*(0^*1 + \varepsilon)$
Grammar: 
$$S_2 \rightarrow 01S_1$$

$C_1$ forces number of zeros in blocks 1-2, 3-4, 5-6 ... to be incremental by 1. $C_2$ forces number of zeros in block 2-3, 4-5, 6-7 ... to be incremental by 1, and forces the string to start with 01. So the intersection of two languages starts with 01 and increments by 1 zero with each subsequent blocks.

4. Grammar:
$$S \rightarrow (S) \mid SS \mid \varepsilon$$

5. To convert many-state PDA (here we all it $P_m$) or one-state PDA ($P_s$), we can encode the states information in $P_m$ in the stack symbols of $P_s$. All stacks symbols in $P_s$ can be written in the form $[pXq]$, where $p$ and $q$ are the states information that came from $P_m$. If $[pXq]$ is on the top of the stack, it indicates that we are currently in state $p$, and if this symbol is popped, we guess that we would be in state $q$. For example,

if $P_m$ contains
$$\delta(p,a,X) = \{(q,\varepsilon)\}$$

Then in $P_s$, it translates to
$$\delta(s,a,[pXq]) = \{(s,\varepsilon)\}, s \text{ is the only state in } P_s$$

Is the state we guessed does not match the state of the symbol on top of the stack the symbol can never be deleted.

6. $L : \{a^ib^jc^k \mid i < k \text{ and } j > k\}$

According to the pumping lemma, if $L$ is a CFL: There exists a constant $n$ such
that if \( z \) is any string in \( L \) such that \( |z| \) is at least \( n \), then we can write \( z = uvwxy \), and:

\[ |vwx| \leq n \]

\[ vx \neq \epsilon \]

for all \( i \geq 0, xv^ixy \) is in \( L \).

We let \( z \) be all strings of the form \( a^n b^{n+2} c^{n+1} \), this is in the language \( L \). And to meet the three conditions above, the substring \( vwx \) can be of the form:

\( a^+, a^+b^+, b^+, b^+c^+ \) or \( c^+ \).

We look at each case individually.

i. \( a^+, a^+b^+ \):

In these two cases, \( v \) has to contain some number of \( a \), so if we pump up, the length of \( a \) will be the same or exceed the length of \( c \).

ii. \( b^+ \):

In this case, \( v \) and \( x \) must both be bs, so if we pump down, \( b \) will be the same length as \( c \) or less.

iii. \( b^+c^+, c^+ \):

In this case, \( x \) has to contain some number of \( c \), so if we pump down, the length of \( c \) will be the same or less than the length of \( a \).

We have shown that \( z \) cannot be pumped, therefore \( L \) is not CFL.