1.28

(a)

\[
\begin{align*}
\text{a} & \quad \rightarrow \quad \text{a} \\
\text{b} & \quad \rightarrow \quad \text{b} \\
\text{abb} & \quad \rightarrow \quad \epsilon \rightarrow \text{a} \rightarrow \epsilon \rightarrow \text{b} \rightarrow \epsilon \rightarrow \text{b} \\
(abb)^* & \quad \rightarrow \quad \epsilon \rightarrow \text{a} \rightarrow \epsilon \rightarrow \text{b} \rightarrow \epsilon \rightarrow \text{b} \\
a(abb)^* \cup b & \quad \rightarrow \quad \epsilon \rightarrow \text{a} \rightarrow \epsilon \rightarrow \text{a} \rightarrow \epsilon \rightarrow \text{b} \rightarrow \epsilon \rightarrow \text{b}
\end{align*}
\]
b) \[ a^+ = a a^+ \]

\[ a^+ \cup (ab)^+ \]

Corresponding DFA is:
2. Show that whenever $L$ is a regular language with alphabet $\Sigma$

   a) $P = \{ u : uv \in L; u, v \in \Sigma^* \}$ is regular.
   b) $P = \{ u : vu \in L; u, v \in \Sigma^* \}$ is regular.
   c) $R = \{ u^{rev} : u \in L \}$ is regular. ($u^{rev}$ is the reversal of $u$, so that if $u = u_1u_2...u_m$, then $u^{rev} = u_mu_{m-1}...u_2u_1$.)

Call the language of (a) Prefix(L), the language of (b) Suffix(L), and the language of (c) Reverse(L).

(a) Prefix(L) = Reverse(Suffix(Reverse(L)), so if we establish the (b) and (c) parts, the (a) part follows.

(b) Because L is regular, there is a DFA that recognizes L. We make an NFA that accepts Suffix(L). First we modify the DFA to remove states that can never be reached. To do this, start by labeling the start state as reachable. Now whenever a state q is reachable and there is a transition from state q to state r, label state r as reachable. Repeat until all states are marked reachable, or until all states that are not reachable have no incoming transition from a reachable state. Delete all unreachable states. Next we modify the DFA to form an NFA as follows. Add a new start state s. Then for every (reachable) state r, add an $\epsilon$-transition from s to r. Leave all accept states as in the DFA. This NFA accepts precisely the suffixes of L. To see this, note that for uv in L, there must be a state r in which the DFA arrives after reading u. Now r is a reachable state, and so in the NFA we can take the $\epsilon$-transition from s to r without reading any input, and then follow the computation of the DFA from there on to accept v. Because Suffix(L) is accepted by an NFA, it is a regular language.
(c) Because $L$ is regular, there is a regular expression $E$ that generates $L$. We will form a regular expression for $\text{Reverse}(L)$. We do this recursively, building up the regular expression $R(E)$ that generates $\text{Reverse}(L)$. According to Definition 1.52 we need to treat six situations:

a. If $E = \emptyset$, set $R(E) = \emptyset$

b. If $E = \epsilon$, set $R(E) = \epsilon$

c. If $E = a$ for $a$ in $\Sigma$, set $R(E) = a$

d. If $E = (E_1)^*$, set $R(E) = (R(E_1))^*$

e. If $E = E_1 \cup E_2$, set $R(E) = R(E_1) \cup R(E_2)$

f. If $E = E_1 \circ E_2$, set $R(E) = R(E_2) \circ R(E_1)$

Now a string appears in $\text{Reverse}(L)$ if and only if it is generated by $R(E)$, and since $R(E)$ is a regular expression, $\text{Reverse}(L)$ is a regular language.

3. Proof:

We need to show both directions.

1) If $L$ is a regular language, then $L$ is generated by a regular grammar.

Suppose $L$ is a regular language, then $L$ can be accepted by a DFA: $(Q, \Sigma, \delta, S, F)$. We form a regular grammar in which the variables are the states $Q$, and the set of terminals is $\Sigma$. Place the following rules in the grammar:

i. Whenever $\delta(R, a) = T$, add $R \rightarrow aT$.

ii. Whenever $R$ is in $F$, add $R \rightarrow \epsilon$.

The start variable is $S$.

Any derivation in the grammar corresponds to a computation of the DFA, and if the string derived has only terminals, the DFA must end in an accept state. So the language accepted by the DFA and the language generated by the grammar are the same.

2) If $L$ is generated by a regular grammar, then $L$ is a regular language.

We can construct a NFA, which is able to accept $L$. Let the DFA be $(Q, \Sigma, \delta, S, F)$. All variables in the regular grammar form the states set $Q$, all the terminals form $\Sigma$, the starting state is the start variable in the grammar, the accept states are those variables, which are on the left side of the following rules: $R \rightarrow \epsilon$ and $R \rightarrow a$. And also, the transition function $\delta$ is defined as: whenever $R \rightarrow aT$, then $\delta(R, a) = T$.

Hence, $L$ can be accepted by this NFA, which implies that $L$ is a regular language.
4. Solution
   a) If we think \( l \) as '(' and \( r \) as ')', the language \( B \) is the set of all strings with balanced parentheses.

   \[
   S \Rightarrow SS \Rightarrow S \Rightarrow \varepsilon
   \]

   \[
   S \Rightarrow \varepsilon
   \]

   These are two leftmost derivations of the same string in the language, which means the grammar is ambiguous. Note that we must start with the start variable, and end with no variables (just terminals).

   b) Assume that \( B \) regular. Let \( p \) be the pumping length given by the pumping lemma. Let \( s \) be the string \( l^p r^p \). With guarantees that \( s \) can be split into three pieces, \( s = xyz \), where for any \( i \geq 0 \) the string \( xyz^i \) is in \( B \).

   According to condition 3 in the pumping lemma, we must have \( |xy| \leq p \). If this is the case, then \( y \) must consist only of \( l \)'s, so \( xyyz \notin B \). Therefore \( s \) cannot be pumped. This is a contradiction.

5. Solution
   a) \( L \) is a symmetric language. Consider a context-free grammar for \( L : (V, \Sigma, R, S) \), where

   i. \( V = \{S\} \)
   ii. \( \Sigma = \{0,1\} \)
   iii. \( R : S \rightarrow 0S|1S|0|1|\varepsilon \)
   iv. \( S = S \in V \)
c) Assume that $L$ regular. Let $p$ be the pumping length given by the pumping lemma. Let $s$ be the string $1^{p-1}101^p$. With guarantees that $s$ can be split into three pieces, $s = xyz$, where for any $i \geq 0$ the string $xy^i z$ is in $L$.

According to condition 3 in the pumping lemma, we must have $|xy| \leq p$. If this is the case, then $y$ must consist only of 1’s, so $xy^i z \notin L$. Therefore $s$ cannot be pumped. This is a contradiction.