1.

2. $(1 + 0)^* 11(1 + 0)^* + 11(1 + 0)^*$

3. Let’s assume for contradiction that $L$ is a regular language. We apply the pumping lemma to $L$. Let $m$ be the parameter of the pumping lemma. We choose to pump the string $a^m b^\lambda c^m$ which is in the language $L$. Since $xyz = a^m b^\lambda c^m$ and $|xy| \leq m$ we have that the string $y$ is a substring of the first $a^m$. Therefore, the string $y$ has the form $y = a^p$, for some integer $p$, $1 \leq p \leq m$ (since $|y| \geq 1$). Now, we pump up $y$ once and we obtain the string $a^{m+p} b^\lambda c^m$. By the pumping lemma, we have that $a^{m+p} b^\lambda c^m$ is in the language $L$. However, $a^{m+p} b^\lambda c^m$ is not in the language $L$ since $m + p \neq m$. Therefore, we have a contradiction, and thus the language $L$ is not be regular.

4.

The initial stack symbol is $. State q_0 reads the $a$’s and pushes them into the stack. State $q_1$ reads the $b$’s and pops an $a$ from the stack for each input
b. Finally, state $q_3$ is the accept state which the automaton enters only if there is an $a$ in the stack, which means that the numbers of $a$’s was more than the number of $b$’s.

5.

(a) 

\[ S \rightarrow aSa|bSb|A \]
\[ A \rightarrow aAb|\lambda \]

(b) 

\[ S \Rightarrow aSa \Rightarrow abSba \Rightarrow abAba \Rightarrow abaAbba \Rightarrow abaaAbba \Rightarrow abaaabbba \]

6. Yes, the grammar is ambiguous. The reason is that there is string generated by the grammar that has two different derivation trees. This string is $bbaa$. The two derivation trees are:

![Diagram of derivation tree 1]

![Diagram of derivation tree 2]
7.

\[ S \rightarrow AV_1 \]
\[ V_1 \rightarrow T_b V_2 \]
\[ V_2 \rightarrow BT_a \]
\[ A \rightarrow AV_3 \]
\[ V_3 \rightarrow BT_a \]
\[ A \rightarrow a \]
\[ B \rightarrow BV_4 \]
\[ V_4 \rightarrow T_a A \]
\[ B \rightarrow b \]
\[ T_a \rightarrow a \]
\[ T_b \rightarrow b \]