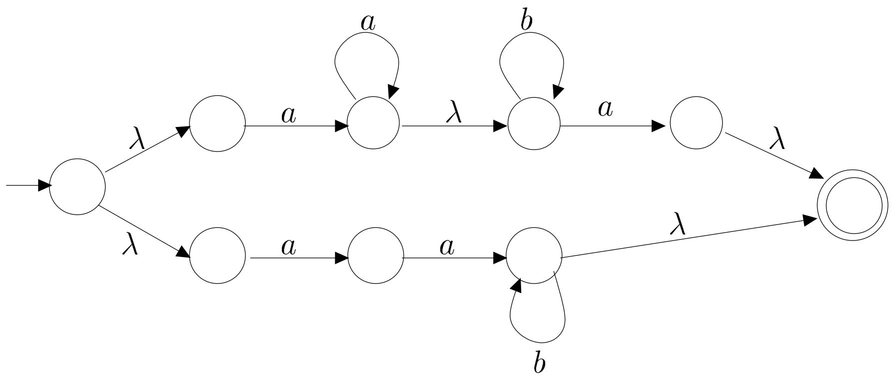
CSCI 2400 – Models of Computation, Section 3

# Solutions to the Practice Midterm Exam

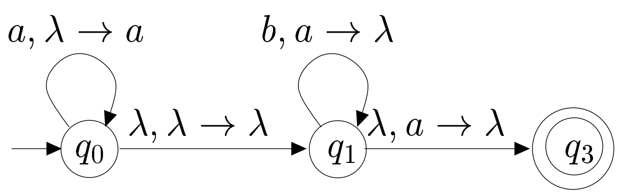
**1.**




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

**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 $a^m b^5 c^m$ which is in the language $L$. Since $xyz = a^m b^5 c^m$ 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 $a^{m+p} b^5 c^m$. By the pumping lemma, we have that $a^{m+p} b^5 c^m$ is in the language $L$. However, $a^{m+p} b^5 c^m$ 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.**

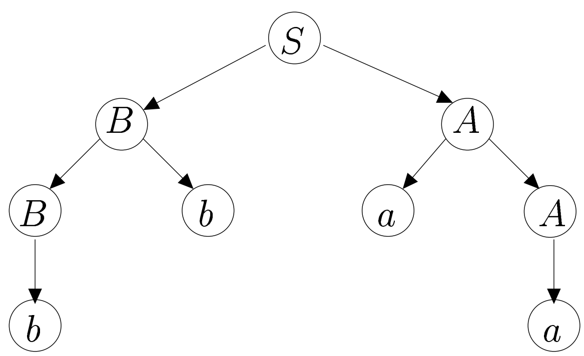
(a)

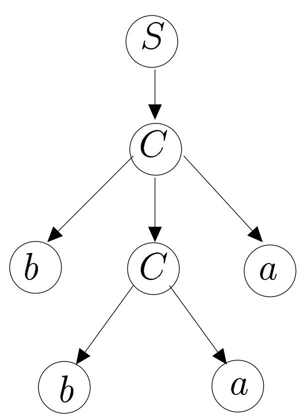
$$S \to aSa|bSb|A$$
$$A \to aAb|\lambda$$

(b)

$$S \Rightarrow aSa \Rightarrow abSba \Rightarrow abAba \Rightarrow abaAbba \Rightarrow abaaAbbba \Rightarrow abaabbba$$

**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:



**7.**

$$S \to AV_1$$
$$V_1 \to T_bV_2$$
$$V_2 \to BT_a$$
$$A \to AV_3$$
$$V_3 \to BT_a$$
$$A \to a$$
$$B \to BV_4$$
$$V_4 \to T_aA$$
$$B \to b$$
$$T_a \to a$$
$$T_b \to b$$