IMPORTANT: This proof was automatically translated from original in Czech language to English!

(0.5 point) Design a context-free grammar for the following context-free language:

$$L_1 = \{w : w \in \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}^* \wedge w = w^{\mathrm{R}}\} \cap \{u : u \in \{\mathtt{a}, \mathtt{b}\}^* \wedge |u|_{\mathtt{a}} \bmod 3 = 0\}$$

Note: Using the notation $|x|_a$ to denote the number of symbols a in the string x.

SOLUTION:

We can notice that $L_2 := \{w : w \in \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}^* \wedge w = w^{\mathrm{R}}\}$ contains all palindromes, i.e.:

$$L_2 = \{\epsilon, a, b, c, aba, bab, cac, abbcbba, \dots\}.$$

$L_3 := \{u : u \in \{\mathtt{a}, \mathtt{b}\}^* \wedge |u|_{\mathtt{a}} \bmod 3 = 0\}$ contains all strings that have the number of symbols a such that it leaves no remainder when divided by 3, i.e., the string must contain 0, 3, 6, 9, etc., symbols a.

$$L_3 = \{\epsilon, b, bb, bbb, bbbb, \dots, aaa, abaa, baaabbbbaaba, \dots\}$$

The intersection L_2 and L_3 must therefore contain all strings w, for which it holds that w is a palindrome $\wedge$ is over the alphabet $\{a, b\}$ (L_3 excludes all strings that have at least one symbol c) $\wedge$ the number of symbols a leaves no remainder when divided by 3.

That is:

$$L_1 = L_2 \cap L_3 := \{w : w \in \{a, b\}^* \wedge w = w^R \wedge |w|_a \mod 3 = 0\}$$

For example:

$$L_1 = \{b, bb, bbb, \dots, aaa, ababa, baaab, aaabaaa, bbaababababb, \dots\}$$

$G = (N, \Sigma, P, S)$ is a context-free grammar that generates exactly such a language, i.e., $L(G) = L_1$, where:

$$\begin{aligned}
&N := \{S, B, C\} \\
&\Sigma := \{a, b\} \\
&P := \{ \\
&\quad S \to \epsilon \mid aBa \mid bSb \mid b, && |w|_a \bmod 3 = 0 \\
&\quad B \to a \mid bBb \mid aCa, && |w|_a \bmod 3 = 2 \\
&\quad C \to aa \mid bCb \mid aSa, && |w|_a \bmod 3 = 1 \\
&\}
\end{aligned}$$

Examples of derivations:

$$\begin{aligned}
&S \Rightarrow \epsilon \\
&S \Rightarrow b \\
&S \Rightarrow aBa \Rightarrow aaa \\
&S \Rightarrow aBa \Rightarrow abBba \Rightarrow ababa \\
&S \Rightarrow aBa \Rightarrow abBba \Rightarrow abbBbba \Rightarrow abbabba \\
&S \Rightarrow aBa \Rightarrow abBba \Rightarrow abbBbba \Rightarrow abbaCabba \Rightarrow abbaaaabba \\
&S \Rightarrow aBa \Rightarrow abBba \Rightarrow abbBbba \Rightarrow abbaCabba \Rightarrow abbabCbabba \Rightarrow abbabaababba \\
&S \Rightarrow aBa \Rightarrow abBba \Rightarrow abbBbba \Rightarrow abbaCabba \Rightarrow abbaaSaabba \Rightarrow abbaabSbaabba \Rightarrow abbaabbbaabba
\end{aligned}$$

(0.5 point) Consider the following two statements. For each statement, decide whether it is true or false. Then properly prove your answer.

Statement 1 Every finite language is regular.

SOLUTION:

We will prove the truth of this statement using induction:

Let us have a finite language L, which has n strings, $n \in \mathbb{N}_0$.

Base Case 1:

For $n = 0$:

We have an empty language, which can be generated by a regular grammar:

$$G = (\{S\}, \{\}, \{\}, S)$$

and thus it is a regular language.

For $n = 1$:

We have a language that contains only one string. We will prove that such a language is regular:

We will again prove this by induction:

We need, using a regular grammar, to generate one string of length $m \in \mathbb{N}_0$, which by definition is a finite sequence of symbols of the alphabet.

Base Case 2:

For $m = 0$, we have the empty string ϵ, which is generated by the regular grammar:

$$G = (\{S\}, \{\}, \{S \to \epsilon\}, S)$$

i.e., $L(G) = \{\epsilon\}$ is a regular language.

For $m = 1$, we have to generate the string $w = w_1$. We can do this using a regular grammar:

$$G = (\{W_1\}, \{w_1\}, \{W_1 \to w_1\}, W_1)$$

For $m = 2$, we have to generate the string $w = w_1w_2$. Again, we can do this using a regular grammar:

$$G = (\{W_1, W_2\}, \{w_1, w_2\}, \{W_1 \to w_1W_2, W_2 \to w_2\}, W_1)$$

For $m = 3$, we have to generate the string $w = w_1w_2w_3$. Again, we can do this using a regular grammar:

$$G = (\{W_1, W_2, W_3\}, \{w_1, w_2, w_3\}, \{W_1 \to w_1W_2, W_2 \to w_2W_3, W_3 \to w_3\}, W_1)$$

Induction Step 2:

For $m + 1$, we have to generate the string $w = w_1w_2w_3 \ldots w_mw_{m+1}$. We do this as follows:

By the induction hypothesis, let us have a regular grammar G, which generates one string of length m (and therefore also a regular language containing this string):

$$\begin{aligned}
&G = (\\
&\{W_1, W_2, W_3, W_4, \ldots, W_m\}, \\
&\{w_1, w_2, w_3, \ldots, w_m\}, \\
&\{ \\
&\qquad W_1 \to w_1W_2, \\
&\qquad W_2 \to w_2W_3, \\
&\qquad W_3 \to w_3W_4, \\
&\qquad \vdots \\
&\qquad W_{m-1} \to w_{m-1}W_m, \\
&\qquad W_m \to w_m \\
&\}, \\
&W_1 \\
&)
\end{aligned}$$

We generate the symbol at position $m+1$ simply by extending the last production rule and adding another rule, which generates the symbol w_{m+1} at the last position:

$$
\begin{aligned}
&G = (\\
&\{W_1, W_2, W_3, W_4, \dots, W_m\} \cup \{W_{m+1}\}, \\
&\{w_1, w_2, w_3, \dots, w_m\} \cup \{w_{m+1}\}, \\
&\{ \\
&\qquad W_1 \to w_1 W_2, \\
&\qquad W_2 \to w_2 W_3, \\
&\qquad W_3 \to w_3 W_4, \\
&\qquad \vdots \\
&\qquad W_{m-1} \to w_{m-1} W_m, \\
&\qquad W_m \to w_m W_{m+1}, \\
&\qquad W_{m+1} \to w_{m+1} \\
&\}, \\
&W_1 \\
&)
\end{aligned}
$$

The resulting grammar for $m+1$ symbols is again regular, and thus the generated language is also regular.

Thus, we have shown that any language which has exactly one string of any length is always regular.

Induction Step 1:

We will now prove that a language that has $n+1$ strings is also regular: We know that regular languages are closed under the operation of union. Let L_{n+1} be a language that has $n+1$ strings. Then it holds that:

$$L_{n+1} = L_n \cup L_1,$$

where L_n is a language that has n strings and L_1 is a language that has 1 string.

From the induction hypothesis, L_n is a regular language. From **Base Case 1** we know that L_1 is also regular. Then the union of L_n and L_1 will also be a regular language, i.e., the language L_{n+1} is regular.

Thus, we have proved that every finite language is regular.

Statement 2 Every regular language is finite.

SOLUTION:

We will prove by contradiction that the statement is false:

It suffices to find a regular grammar that will generate infinitely many strings:

$$G = (\{S\}, \{a\}, \{S \to a \mid aS\}, S).$$

Then:

$$L(G) = \{a, aa, aaa, aaaa, aaaaa, aaaaaa, \dots\}$$

contains infinitely many strings consisting only of the symbol a. Therefore, this regular language generated by a regular grammar is not finite, which contradicts the statement that every regular language is finite.