ETH zürich
Departement Informatik

Theoretische Informatik

Prof. Dr. Juraj Hromkovič
Dr. Hans-Joachim Böckenhauer
https://courses.ite.inf.ethz.ch/theoInf20

Exemplary Solutions – Sheet 12

Zürich, December 11, 2020

Solution to Exercise 32

(a) The language L_1 is generated by the following regular grammar $G_1 = (\{S, X_1, X_2, X_3\}, \{0,1\}, P_1, S)$ with

$$\begin{aligned} P_1 = \{&S \to 0S, S \to 1S, S \to 1X_1,\\ &X_1 \to 001X_2, X_1 \to 011X_2, X_1 \to 101X_2, X_1 \to 111X_2,\\ &X_2 \to 001X_3, X_2 \to 011X_3, X_2 \to 101X_3, X_2 \to 111X_3,\\ &X_3 \to 0X_3, X_3 \to 1X_3, X_3 \to \lambda\}\,. \end{aligned}$$

First, an arbitrary subword w over $\{0,1\}$ is generated from the start symbol. Then, using the rule $S \to 1X_1$, the first 1 of the desired pattern is generated. Afterwards, from the nonterminal X_1, the subword x of length 2 and the subsequent 1 are generated. The rules for X_2 generate the subword y and the subsequent 1. Finally, the (potentially empty) suffix z is generated using X_3.

(b) The language L_2 can be expressed as $L_2 = L_{21} \cup L_{22}$ with

$$L_{21} = \{x \in \{a,b\}^* \mid (|x|_a + 2|x|_b) \bmod 3 = 2\}$$

and

$$L_{22} = \{x \in \{a,b\}^* \mid x \text{ starts and ends by } bb\}\,.$$

We first construct two regular grammars G_{21} and G_{22} for the languages L_{21} and L_{22}. The grammar $G_{21} = (\{X_0, X_1, X_2\}, \{a,b\}, P_{21}, X_0)$ with

$$\begin{aligned} P_{21} = \{&X_0 \to aX_1, X_0 \to bX_2, X_1 \to aX_2, X_1 \to bX_0,\\ &X_2 \to aX_0, X_2 \to bX_1, X_2 \to \lambda\} \end{aligned}$$

generates the language L_{21}. The underlying idea is that the nonterminal X_i is produced if and only if the prefix x of the final word satisfies the condition $(|x|_a + 2|x|_b) \bmod 3 = i$. The rule $X_2 \to \lambda$ makes sure that the derivation can only end if the generated word is in the language L_{21}.

The grammar $G_{22} = (\{Y, Z\}, \{a, b\}, P_{22}, Y)$ with

$$P_{22} = \{Y \to bb, Y \to bbb, Y \to bbZ, Z \to aZ, Z \to bZ, Z \to bb\}$$

generates the language L_{22}. The first two rules generate the two short words in which the bb's at the beginning and the end of the word overlap. All longer words from L_{22} can be generated using the remaining rules: The rule $Y \to bbZ$ generates the prefix bb, an arbitrary middle part can be generated using the rules $Z \to aZ$ and $Z \to bZ$, and, finally, the rule $Z \to bb$ generates the suffix bb.

A grammar for L_2 can now be obtained by introducing a new start symbol S and adding the two new rules $S \to X_0$ and $S \to Y$. This way, the first derivation step decides if a word from L_{21} or L_{22} will be generated. This construction assumes that the sets of nonterminals in G_{21} and G_{22} are disjoint which is clearly satisfied here. Overall, this yields the grammar $G_2 = (\{S, X_0, X_1, X_2, Y, Z\}, \{a, b\}, P_2, S)$ with

$$\begin{aligned}
P_2 &= \{S \to X_0, S \to Y\} \cup P_{21} \cup P_{22} \\
&= \{S \to X_0, S \to Y, \\
&\qquad X_0 \to aX_1, X_0 \to bX_2, X_1 \to aX_2, X_1 \to bX_0, X_2 \to aX_0, X_2 \to bX_1, X_2 \to \lambda, \\
&\qquad Y \to bb, Y \to bbb, Y \to bbZ, Z \to aZ, Z \to bZ, Z \to bb\}\,.
\end{aligned}$$

Solution to Exercise 33

We consider the normalized regular grammar $G = (\Sigma_N, \Sigma_T, P, S)$ for L. Since all rules in P (potentially except the rule $S \to \lambda$) have the form $A \to aB$ or $A \to a$ for $A, B \in \Sigma_N$ and $a \in \Sigma_T$, we can construct a context-free grammar G' for $L' = \{vwv^{\mathsf{R}} \mid v, w \in L\}$ as follows:

$$\begin{aligned}
G' &= (\Sigma_N \cup \Sigma'_N, \Sigma_T, P \cup P', S') \quad \text{with} \\
\Sigma'_N &= \{A' \mid A \in \Sigma_N\}, \quad \text{where } \Sigma_N \cap \Sigma'_N = \emptyset, \\
P' &= \{A' \to aB'a \mid A \to aB \in P\} \cup \{A' \to aSa \mid A \to a \in P\} \cup \{S' \to S \mid S \to \lambda \in P\}.
\end{aligned}$$

The idea behind this construction is as follows: In the normalized grammar G, every derivation step (except when applying the rule $S \to \lambda$) produces exactly one symbol of the generated word $u = a_1 \dots a_m$. A terminating rule (i.e., one of the form $A \to a$) is only applied in the last step. The production proceeds from left to right, i.e., the symbol a_1 is produced first and the symbol a_m last. The grammar G' first generates two copies v and v^{R} simultaneously from outside inwards. In the last step (when simulating the terminating rule from G), the start symbol of G is inserted once again. From this, the word w can be generated by the rules from P.

Solution to Exercise 34

The grammar $G_3 = (\{S, A, X\}, \{0, 1, 2\}, P_3, S)$ with

$$P_3 = \{S \to AS2, S \to X, AX \to 0X1, A0 \to 0A, X \to \lambda\}$$

generates the language L_3. It is based on the following idea: Using the rule $S \to AS2$, an equal number of A's and 2's is generated. The A's serve as placeholders for 0's and 1's here. Using the rule $AX \to 0X1$, a symbol A is transformed into a 0 and 1. The rule $A0 \to 0A$ moves the produced 0 across all A's to the left. Once the produced 0 has been moved left at least once, the rule $AX \to 0X1$ can be applied again. If the rule $X \to \lambda$ is applied although A's are still present, then no terminal word can be produced, which means that we do not get a valid derivation. The empty word can also be derived using $S \Rightarrow X \Rightarrow \lambda$. A derivation of the word 000111222 can be as follows:

$$\begin{aligned} S &\Rightarrow AS2 \Rightarrow AAS22 \Rightarrow AAAS222 \Rightarrow AAAX222 \\ &\Rightarrow AA0X1222 \Rightarrow A0AX1222 \Rightarrow 0AAX1222 \Rightarrow 0A0X11222 \\ &\Rightarrow 00AX11222 \Rightarrow 000X111222 \Rightarrow 000111222\,. \end{aligned}$$