Universität Zürich
Institut für Informatik

Autumn 2020

Sven Seuken
Dennis Komm

Student Name:
Student Number:

Foundations of Computing II
Assignment 3 – Solutions

Context-Free Grammars and Languages

Distributed: 26.10.2020 – Due Date: 01.11.2020

Upload your solutions to the OLAT system.

3.1 Context-Free Grammars and Languages

As defined in the lecture, for a word w, w^{R} denotes the "reversal" of w; for instance, $(aabbba)^{\mathsf{R}}$ is $abbbaa$. Furthermore, as already defined in Exercise 2.1, $|w|_a$ denotes the number of occurrences of the letter a in w. Construct context-free grammars for the following languages.

a) $L_1 = \{w \in \{0,1\}^* \mid w \text{ ends with } 01 \text{ and } |w|_0 \text{ is even}\}$

The language L_1 is the language of the CFG $G_1 = (\{S, X_0, X_1\}, \{0,1\}, P_1, S)$ with

$$\begin{aligned} P_1 = \{&S \to X_0 01, \\ &X_0 \to 0X_1 \mid 1X_0, \\ &X_1 \to 0X_0 \mid 1X_1 \mid \varepsilon\}. \end{aligned}$$

b) $L_2 = \{w \in \{a,b\}^* \mid (2|w|_a + |w|_b) \bmod 5 = 0\}$

The language L_2 is the language of the CFG $G_2 = (\{S, X_0, X_1, X_2, X_3, X_4\}, \{a,b\}, P_2, S)$ with

$$\begin{aligned} P_2 = \{&S \to X_0, \\ &X_0 \to bX_1 \mid aX_2 \mid \varepsilon, \\ &X_1 \to bX_2 \mid aX_3, \\ &X_2 \to bX_3 \mid aX_4, \\ &X_3 \to bX_4 \mid aX_0, \\ &X_4 \to bX_0 \mid aX_1\}. \end{aligned}$$

The idea is similar to the one of a), namely that, after every intermediate derivation step, there is one variable X_i at the last position, and we have that

the number of bs plus 2 times the number of as modulo 5 is i

for the terminal word left of X_i. Again, since this expression must be 0, there is a production $X_0 \to \varepsilon$.

c) $L_3 = \{w \in \{a,b\}^* \mid w \text{ ends with } aa \text{ or } w = w^{\mathsf{R}}\}$

The language L_3 is the language of the CFG $G_3 = (\{S, X, Y\}, \{a, b\}, P_3, S)$ with

$$\begin{aligned} P_3 = \{S &\to Xaa \mid Y, \\ X &\to aX \mid bX \mid \varepsilon, \\ Y &\to aYa \mid bYb \mid a \mid b \mid \varepsilon\}. \end{aligned}$$

d) As defined in the lecture, a grammar is called a "regular grammar" if it has only productions of the form $X \to aY$, $X \to a$, $X \to \varepsilon$, where X and Y are non-terminals and a is a terminal. As the name suggests, exactly the regular languages allow for regular grammars. For which of the above languages L_1, L_2, and L_3 can you give regular grammars? How surprising is your result?

First, we note that the language L_3 is not regular. Therefore, there is no regular grammar for L_3 by definition. Second, the grammar G_2 for L_2 is already a regular grammar. This is not surprising since we have seen in the lecture that the construction is analogous to the construction of a DFA for L_2. The grammar G_1 for L_1 is not regular, but only due to the first production $S \to X_0 01$. However, it is not difficult to transform G_1 into an equivalent regular grammar. To this end, we delete the first production and make X_1 the new starting symbol. Then, we replace the last production $X_1 \to \varepsilon$ by the three productions $X_1 \to 0Y_0$, $Y_0 \to 1Y_1$, and $Y_1 \to \varepsilon$.

3.2 Parsing Strings

The CFG $G_4 = (\{S, A, B\}, \{0, 1\}, P, S)$ with

$$\begin{aligned} P_4 = \{S &\to A1B, \\ A &\to 0A \mid \varepsilon, \\ B &\to 0B \mid 1B \mid \varepsilon\} \end{aligned}$$

generates the language L_4 which corresponds to the regular expression $0^*1(0+1)^*$.

a) Give both the leftmost and rightmost derivations of 1101.

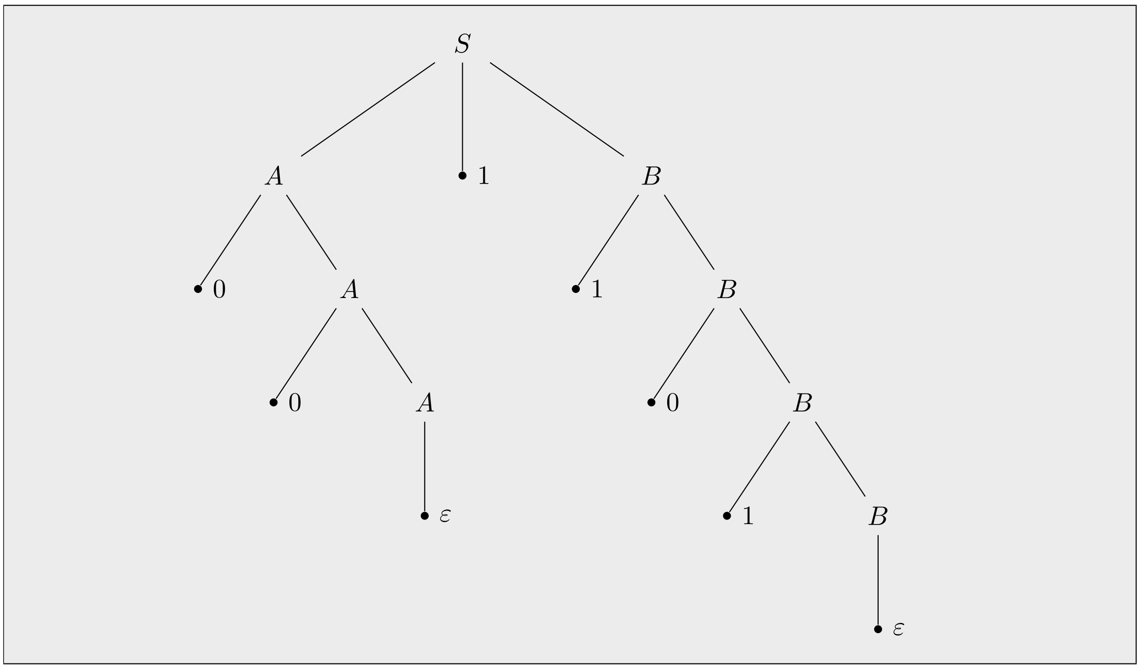
The leftmost derivation of 1101 is

$$S \Rightarrow A1B \Rightarrow 1B \Rightarrow 11B \Rightarrow 110B \Rightarrow 1101B \Rightarrow 1101,$$

and the rightmost derivation is

$$S \Rightarrow A1B \Rightarrow A11B \Rightarrow A110B \Rightarrow A1101B \Rightarrow A1101 \Rightarrow 1101.$$

b) Write down the parse tree of 001101.



3.3 Normal Forms

a) Use the method presented in the lecture to eliminate all ε-productions of the CFG $G_5 = (\{S, A, B, C, D\}, \{a, b, c\}, P_5, S)$ with

$$P_5 = \{S \to ABCD, A \to \varepsilon, A \to BB, B \to AA, A \to a, B \to b, C \to bc, D \to \varepsilon\}.$$

I. First, we find the *nullable variables* of G, that is, variables that may derive ε. Using the iterative approach from the lecture, we first set $\mathrm{Null}_1 = \{A, D\}$ due to the productions $A \to \varepsilon$ and $D \to \varepsilon$. Next, we set $\mathrm{Null}_2 = \mathrm{Null}_1 \cup \{B\} = \{A, B, D\}$ due to the production $B \to AA$. Then the method terminates and the nullable variables are A, B, and D.

II. Second, again by applying the method from the lecture, we obtain the new productions $S \to BCD$, $S \to ACD$, $S \to ABC$, $S \to CD$, $S \to BC$, $S \to AC$, and $S \to C$ due to the production $S \to ABCD$. Due to $A \to BB$, we add $A \to B$, and due to $B \to AA$ we add $B \to A$. Therefore, we get the new set of productions

$$\begin{aligned} P_5' = P_5 \setminus &\{A \to \varepsilon, D \to \varepsilon\} \\ \cup\, &\{S \to BCD, S \to ACD, S \to ABC, S \to CD, S \to BC\} \\ \cup\, &\{S \to AC, S \to C, A \to B, B \to A\}. \end{aligned}$$

Note that this may have caused that some variables become non-generating. However, this is not a problem, because, we will find and remove all such variables in a last step when simplifying a CFG.

b) Use the method presented in the lecture to eliminate all unit productions in the CFG $G_6 = (\{S, A, B, C, D\}, \{a, b, c, d\}, P_6, S)$ with

$$P_6 = \{S \to ABC, A \to B, B \to C, B \to b, B \to bB, C \to D, D \to d\}.$$

I. First, we find all *unit pairs*; again, this is done iteratively. The unit pairs from G are (S,S), (A,A), (A,B), (A,C), (A,D), (B,B), (B,C), (B,D), (C,C), (C,D), (D,D).

II. Second, for every unit pair (X,Y), we add the production $X \to \alpha$ if there is a non-unit production $Y \to \alpha$ in P_6. This results in the table on the right.

Pairs	Productions
(S,S)	$S \to ABC$
(A,A)	
(A,B)	$A \to b, A \to bB$
(A,C)	
(A,D)	$A \to d$
(B,B)	$B \to b, B \to bB$
(B,C)	
(B,D)	$B \to d$
(C,C)	
(C,D)	$C \to d$
(D,D)	$D \to d$

Therefore, we obtain a simplified CFG $G_6' = (\{S,A,B,C,D\}, \{a,b,c,d\}, P_6', S)$ with

$$\begin{aligned} P_6' = \{&S \to ABC, A \to b, A \to bB, \\ &A \to d, B \to b, B \to bB, \\ &B \to d, C \to d, D \to d\}. \end{aligned}$$

c) Use the method presented in the lecture to eliminate all useless symbols in the CFG $G_7 = (\{S,A,B,C,D,E\}, \{a,b,c,d\}, P_7, S)$ with

$$\begin{aligned} P_7 = \{&S \to A, S \to AaB, S \to BbA, B \to bB, A \to aa, A \to Ab, \\ &C \to cD, D \to c, D \to Ad, D \to EE, D \to dd\}. \end{aligned}$$

I. First, we compute the *generating symbols* of G_7 using the iterative approach introduced in the lecture. We start by setting $\text{Gen} = \{a,b,c,d\}$. Then we add A due to $A \to aa$ and D due to $D \to c$ (and $D \to dd$); this yields $\text{Gen} = \{a,b,c,d,A,D\}$. Next, we add S due to $S \to A$ and C due to $C \to cD$. This gives $\text{Gen} = \{a,b,c,d,S,A,C,D\}$. Then the method terminates.
The new CFG is $G_7' = (\{S,A,C,D\}, \{a,b,c,d\}, P_7', S)$ with

$$P_7' = \{S \to A, A \to aa, A \to Ab, C \to cD, D \to c, D \to Ad, D \to dd\}.$$

II. Second, we compute the *reachable symbols* of G_7', again using the iterative approach from the lecture. We initially set $\text{Reach}_V = \{S\}$ and $\text{Reach}_T = \emptyset$. Then, we add A to Reach_V due to $S \to A$; this yields $\text{Reach}_V = \{S,A\}$ and $\text{Reach}_T = \emptyset$. After that, we add a and b to Reach_T due to $A \to aa$ and $A \to Ab$; this gives $\text{Reach}_V = \{S,A\}$ and $\text{Reach}_T = \{a,b\}$. Then the method terminates.
The new CFG is $G_7'' = (\{S,A\}, \{a,b\}, P_7'', S)$ with

$$P_7'' = \{S \to A, A \to aa, A \to Ab\}.$$

d) Convert the CFG $G_8 = (\{S,A,B,C\}, \{a,b\}, P_8, S)$ with

$$\begin{aligned} P_8 = \{&S \to aS, S \to Sb, S \to Aa, S \to bbB, A \to aBb, A \to ab, \\ &B \to bCa, B \to ba, C \to b\} \end{aligned}$$

into Chomsky normal form.

I. First, we introduce two new variables X_a and X_b. Then we apply the following changes to the productions of G_8 to make sure that no terminals appear in the bodies anymore.

$$\begin{array}{lll} S \to aS & \text{becomes} & S \to X_a S, \\ S \to Sb & \text{becomes} & S \to S X_b, \\ S \to Aa & \text{becomes} & S \to A X_a, \\ S \to bbB & \text{becomes} & S \to X_b X_b B, \\ A \to aBb & \text{becomes} & A \to X_a B X_b, \\ A \to ab & \text{becomes} & A \to X_a X_b, \\ B \to bCa & \text{becomes} & B \to X_b C X_a, \\ B \to ba & \text{becomes} & B \to X_b X_a. \end{array}$$

Moreover, two new productions $X_a \to a$ and $X_b \to b$ are added to P_8. Note that the production $C \to b$ remains unchanged since its body only contains one terminal.

II. Second, we must make sure that no body of a production that is not a single terminal has a length different from 2. To this end, we introduce new variables Y_1, Y_2, and Y_3 and apply the following changes.

$$\begin{array}{lll} S \to X_b X_b B & \text{becomes} & S \to X_b Y_1, Y_1 \to X_b B, \\ A \to X_a B X_b & \text{becomes} & A \to X_a Y_2, Y_2 \to B X_b, \\ B \to X_b C X_a & \text{becomes} & B \to X_b Y_3, Y_3 \to C X_a. \end{array}$$

The new CFG is therefore $G_8' = (\{S, A, B, C, X_a, X_b, Y_1, Y_2, Y_3\}, \{a, b\}, P_8', S)$ with

$$\begin{aligned} P_8' = \{ & S \to X_a S, S \to S X_b, S \to A X_a, A \to X_a X_b, B \to X_b X_a, \\ & S \to X_b Y_1, Y_1 \to X_b B, A \to X_a Y_2, Y_2 \to B X_b, B \to X_b Y_3, \\ & Y_3 \to C X_a, X_a \to a, X_b \to b, C \to b\}, \end{aligned}$$

and we immediately see that it only contains productions of the form $X \to YZ$ with $X, Y, Z \in V$ or $X \to x$ with $X \in V$ and $x \in T$, which means that it is in Chomsky normal form.

3.4 CYK Algorithm

Consider the CFG $G_9 = (\{S, A, B, C\}, \{a, b\}, P_9, S)$ in Chomsky normal form with

$$\begin{aligned} P_9 = \{ & S \to AB \mid BC, \\ & A \to BA \mid a, \\ & B \to CC \mid b, \\ & C \to AB \mid a\}. \end{aligned}$$

Use the CYK algorithm to determine whether each of the following strings is in $L(G_9)$.

a) $ababa$

Using the CYK algorithm, we obtain the following table.

$$
\begin{array}{|ccccc}
\{S,A,C\} & & & & \\
\{B\} & \{B\} & & & \\
\{B\} & \{S,C\} & \{B\} & & \\
\{S,C\} & \{S,A\} & \{S,C\} & \{S,A\} & \\
\{A,C\} & \{B\} & \{A,C\} & \{B\} & \{A,C\} \\
\hline
a & b & a & b & a
\end{array}
$$

Since S appears in the upper-left corner, *ababa* is in the language of G_9.

b) *baaab*

Using the CYK algorithm, we obtain the following table.

$$
\begin{array}{|ccccc}
\{S,C\} & & & & \\
\{S,A,C\} & \{S,C\} & & & \\
\emptyset & \{S,A,C\} & \{B\} & & \\
\{S,A\} & \{B\} & \{B\} & \{S,C\} & \\
\{B\} & \{A,C\} & \{A,C\} & \{A,C\} & \{B\} \\
\hline
b & a & a & a & b
\end{array}
$$

Since S appears in the upper-left corner, *baaab* is in the language of G_9.

c) *aabab*

Using the CYK algorithm, we obtain the following table.

$$
\begin{array}{|ccccc}
\{S,C\} & & & & \\
\{S,A,C\} & \{B\} & & & \\
\{B\} & \{B\} & \{S,C\} & & \\
\{B\} & \{S,C\} & \{S,A\} & \{S,C\} & \\
\{A,C\} & \{A,C\} & \{B\} & \{A,C\} & \{B\} \\
\hline
a & a & b & a & b
\end{array}
$$

Since S appears in the upper-left corner, *aabab* is in the language of G_9.