Albert-Ludwigs-Universität, Inst. für Informatik
Prof. Dr. Fabian Kuhn
M. Ahmadi, P. Bamberger, H. Ghodselahi, Y. Maus, P. Schneider

Theoretical Computer Science - Bridging Course
Summer Term 2017
Exercise Sheet 4

Hand in (electronically or hard copy) by 12:15 pm, June 12th, 2017

Exercise 1: Context-Free Languages (2+2+1 points)

Give context-free grammars that generate the following languages. The alphabet set is $\Sigma = \{0, 1\}$.

a) $\{w \mid w$ contains at least three ones$\}$

b) $\{w \mid w$ the length of w is odd and its middle symbol is a $0\}$

c) The empty language.
Note: The empty language is not the language containing only the empty string!

Solution

a) $S \rightarrow R1R1R1R$
$R \rightarrow 0R \mid 1R \mid \varepsilon$

b) $S \rightarrow 0 \mid 0S0 \mid 0S1 \mid 1S0 \mid 1S1$

c) $S \rightarrow S$

Exercise 2: Chomsky Normal Form (2+5 points)

Consider the following context-free grammar (CFG):

$$S \rightarrow aSb \mid D$$
$$D \rightarrow ccDcc \mid \varepsilon$$

a) Which language does this grammar define?

b) Convert this CFG into an equivalent one in Chomsky Normal Form. Give the grammar you obtained after each step of the conversion algorithm.

Solution

a) The language consists of all strings of the form xyz with the following properties:

- $x \in \{a\}^*$
- $z \in \{b\}^*$
- $y \in \{c\}^*$
- $|x| = |z|$
- $|y| = 4 \cdot n$ for an $n \in \mathbb{N}$

b) Add a new start variable S_0 and the rule $S_0 \to S$.

$$\begin{aligned} S_0 &\to S \\ S &\to aSb \mid D \\ D &\to ccDcc \mid \varepsilon \end{aligned}$$

Remove all ε-rules: Delete the rule $D \to \varepsilon$ and add the rules $S \to \varepsilon$ and $D \to cccc$.

$$\begin{aligned} S_0 &\to S \\ S &\to aSb \mid D \mid \varepsilon \\ D &\to ccDcc \mid cccc \end{aligned}$$

Remove $S \to \varepsilon$ and add $S \to ab$ and $S_0 \to \varepsilon$ (the ε-rule for the start variable is allowed).

$$\begin{aligned} S_0 &\to S \mid \varepsilon \\ S &\to aSb \mid ab \mid D \\ D &\to ccDcc \mid cccc \end{aligned}$$

Next remove unit rules.
Remove $S_0 \to S$ and add $S_0 \to aSb \mid ab \mid D$.

$$\begin{aligned} S_0 &\to \varepsilon \mid aSb \mid ab \mid D \\ S &\to aSb \mid ab \mid D \\ D &\to ccDcc \mid cccc \end{aligned}$$

Remove $S_0 \to D$ and add $S_0 \to ccDcc \mid cccc$.

$$\begin{aligned} S_0 &\to \varepsilon \mid aSb \mid ab \mid ccDcc \mid cccc \\ S &\to aSb \mid ab \mid D \\ D &\to ccDcc \mid cccc \end{aligned}$$

Remove $S \to D$ and add $S \to ccDcc \mid cccc$.

$$\begin{aligned} S_0 &\to \varepsilon \mid aSb \mid ab \mid ccDcc \mid cccc \\ S &\to aSb \mid ab \mid ccDcc \mid cccc \\ D &\to ccDcc \mid cccc \end{aligned}$$

Convert the rules into the proper form.

Add $S_1 \to Sb$ and adjust the rules accordingly.

$$
\begin{aligned}
S_0 &\to \varepsilon \mid aS_1 \mid ab \mid ccDcc \mid cccc \\
S &\to aS_1 \mid ab \mid ccDcc \mid cccc \\
S_1 &\to Sb \\
D &\to ccDcc \mid cccc
\end{aligned}
$$

Add $U_1 \to a$ and Add $U_2 \to b$ and adjust.

$$
\begin{aligned}
S_0 &\to \varepsilon \mid U_1S_1 \mid U_1U_2 \mid ccDcc \mid cccc \\
S &\to U_1S_1 \mid U_1U_2 \mid ccDcc \mid cccc \\
S_1 &\to SU_2 \\
U_1 &\to a \\
U_2 &\to b \\
D &\to ccDcc \mid cccc
\end{aligned}
$$

Add $S_2 \to cS_3$, $S_3 \to DS_4$ and $S_4 \to cc$ and adjust.

$$
\begin{aligned}
S_0 &\to \varepsilon \mid U_1S_1 \mid U_1U_2 \mid cS_2 \mid cccc \\
S &\to U_1S_1 \mid U_1U_2 \mid cS_2 \mid cccc \\
S_1 &\to SU_2 \\
S_2 &\to cS_3 \\
S_3 &\to DS_4 \\
S_4 &\to cc \\
U_1 &\to a \\
U_2 &\to b \\
D &\to cS_2 \mid cccc
\end{aligned}
$$

Add $S_5 \to cS_6$ and $S_6 \to cc$ and adjust.

$$
\begin{aligned}
S_0 &\to \varepsilon \mid U_1S_1 \mid U_1U_2 \mid cS_2 \mid cS_5 \\
S &\to U_1S_1 \mid U_1U_2 \mid U_3S_2 \mid cS_5 \\
S_1 &\to SU_2 \\
S_2 &\to cS_3 \\
S_3 &\to DS_4 \\
S_4 &\to cc \\
S_5 &\to cS_6 \\
S_6 &\to cc \\
U_1 &\to a \\
U_2 &\to b \\
D &\to cS_2 \mid cS_5
\end{aligned}
$$

Finally, add $U_3 \to c$ and adjust.

$$
\begin{aligned}
S_0 &\to \varepsilon \mid U_1S_1 \mid U_1U_2 \mid U_3S_2 \mid U_3S_5 \\
S &\to U_1S_1 \mid U_1U_2 \mid cS_2 \mid U_3S_5 \\
S_1 &\to SU_2 \\
S_2 &\to U_3S_3 \\
S_3 &\to DS_4 \\
S_4 &\to U_3U_3 \\
S_5 &\to U_3S_6 \\
S_6 &\to U_3U_3 \\
U_1 &\to a \\
U_2 &\to b \\
U_3 &\to c \\
D &\to U_3S_2 \mid U_3S_5
\end{aligned}
$$

Exercise 3: Pushdown Automata (6 points)

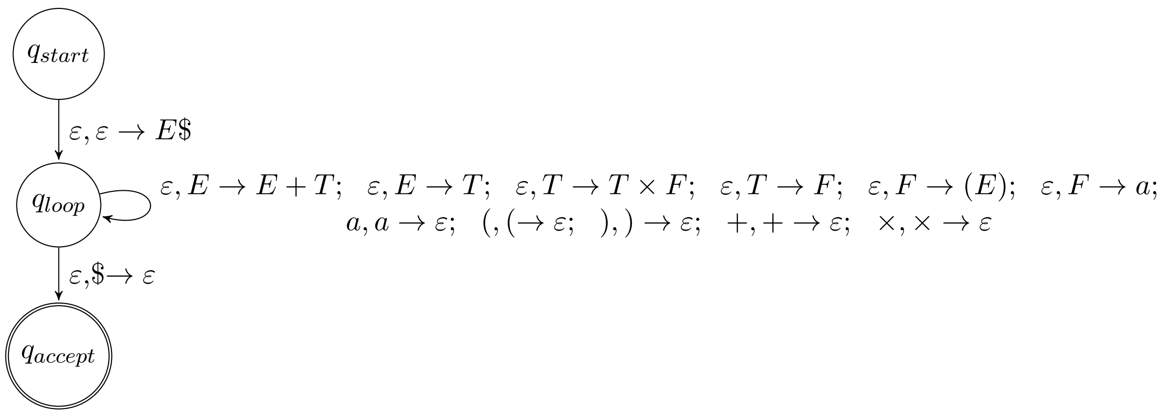
Convert the following CFG to an equivalent pushdown automaton. The alphabet is $\Sigma = \{a, +, \times, (,)\}$ and the set of variables $V = \{E, T, F\}$.

$$
\begin{aligned}
E &\to E + T \mid T \\
T &\to T \times F \mid F \\
F &\to (E) \mid a
\end{aligned}
$$

(You already saw this grammar in the lecture with $< expression >, < term >$ and $< factor >$ instead of E, T and F).

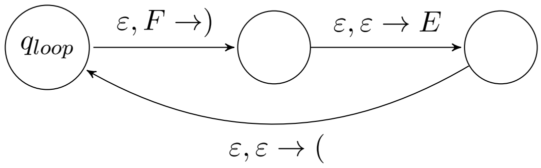
Solution

The Automaton is given by the following diagram.



Remark: The general transformation procedure is given in on Slide 59 of the third set of lecture slides which tackles context free languages. There it is proven (constructively) that any context free grammar can be transformed into a PDA.

Note that transitions like $\varepsilon, F \to (E)$ are shortcuts. In fact, one has to introduce additional states and define the transition as



Exercise 4: Context-Free Languages and Set Operations (3+3 points)

(a) Show that context-free languages are not closed under taking intersections (i.e., the intersection of two context-free languages is not necessarily context free).
Hint: You can use that the language $\{a^i b^i c^i | i \geq 0\}$ is not context-free.

(b) Show that context-free languages are not closed under taking complements.
Hint: You can use DeMorgan's law and the fact that the set of context-free languages is closed under performing union operations.

Solution

(a) Assume context-free languages are closed under intersection operation. We prove the claim by contradiction. Consider the following languages: $L_1 = \{a^i b^i c^j | i, j \geq 0\}$, $L_2 = \{a^i b^j c^j | i, j \geq 0\}$, and $L_3 = L_1 \cap L_2 = \{a^i b^i c^i | i \geq 0\}$. It is easy to prove that L_1 and L_2 are both context-free languages (as you can easily derive the corresponding context-free grammar). According to our assumption, we know L_3 will be context-free language as well. However, we know by the pumping lemma that L_3 is not context-free. Hence, we have a contradiction, which implies the assumption is wrong. Therefore, we have proved the claim.

(b) Assume context-free languages are closed under taking complement operation. We prove the claim by contradiction. Let L_1 and L_2 both be context-free languages. According to the assumption, we know $\overline{L_1}$ and $\overline{L_2}$ must both be context-free languages as well. Since context-free languages are closed under union operation. We know $\overline{L_1} \cup \overline{L_2}$ is context-free too. Now, according to the DeMorgan's law, we know $\overline{\overline{L_1} \cup \overline{L_2}} = L_1 \cap L_2$. As a result, we can further conclude $L_1 \cap L_2$ is context-free. Since L_1 and L_2 are arbitrary context-free languages, this implies context-free languages are closed under intersection operation. However, from previous exercise, we know this is not true. Therefore, we have a contradiction, and have hence proved the original claim.

Exercise 5: Pumping Lemma for Context-Free Languages (3+3 points)

Use the pumping lemma to show that the following languages over the alphabet $\Sigma = \{a, b\}$ are not context free:

(a) $\{ww \mid w \in \{a, b\}^*\}$

Hint: Show that the string $s = a^p b^p a^p b^p$ with p the pumping length cannot be pumped.

(b) $\{a^n b a^{2n} b a^{3n} \mid n \geq 0\}$

(Be careful to read the strings correctly: For example ab^4 is equal to abbbb and not to abababab.)

Solution

(a) Assume the language was context free. Let p be the pumping length. We show that the string $s = a^p b^p a^p b^p$ cannot be pumped, leading to a contradiction. Let $s = uvxyz$ with $|vxy| \leq p$ and $|vy| > 0$.

First, we show that the substring vxy straddles the midpoint of s. If not, then vxy is either fully contained in the first or fully contained in the second half of s. If it is contained in the first half, we obtain that $uv^2xy^2z = tb^pa^pb^p$. Because of $|vy| > 0$ it follows that $|t| > p$ and because of $|uvxy| \leq 2p$ it follows that $|t| < 3p$. So uv^2xy^2z has a b in the first position of its second half, making it impossible to have the form ww. Similarly, if vxy is contained in the second half of s, then the string uv^2xy^2z has an a in the last position of its first half, making it again impossible to have the form ww.

But if vxy straddles the midpoint of s, then because of $|vxy| \leq p$, pumping s down to uxz leads to the string $a^pb^ia^jb^p$. As $|vy| > 0$, either i or j (or both) are strictly less than p. So this string has not the form ww.

(b) Assume the language was context free with p the pumping length. Define $s := a^pba^{2p}ba^{3p}$ and let $s = uvxyz$ be a decomposition of s with $|vxy| \leq p$ and $|vy| > 0$. We show that uv^2xy^2z cannot be in the language, giving a contradiction. If v or y contained b, the string uv^2xy^2z would have more than two b's and is therefore not in the language. So assume that neither v nor y contains b. That means that v as well as y is fully contained in one of the three segments a^p, a^{2p} and a^{3p}. But then pumping s up to uv^2xy^2z would violate the $1:2:3$ length ratio of the segments, because the length of at least one segment is changed (as $|vy| > 0$) and at least one segment keeps its length.