Automata and Grammars

SS 2018

Assignment 9: Solutions to Selected Problems

Problem 9.1. [Greibach Normal Form]

Convert the context-free grammar $G_1 = (\{S, E, F\}, \{a, (,), +, *\}, P_1, S)$ into an equivalent grammar that is in Greibach Normal Form, where

$$P_1 = \{S \to (E), E \to F + F, E \to F * F, F \to a, F \to S\}.$$

Use the construction detailed in the proof of Theorem 3.10, and notice that this construction already applies to context-free grammars that are in weak CNF.

Solution. We must first eliminate the chain rule ($F \to S$), which is done by deleting this rule and by adding the rule $F \to (E)$, which yields the set of productions

$$\hat{P}_1 = \{S \to (E), E \to F + F, E \to F * F, F \to a, F \to (E)\}.$$

Then we introduce new nonterminals A, B, C to replace the terminal symbols), +, and $*$ on the right-hand side of productions, which yields the grammar

$$G_1' = (\{S, E, F, A, B, C\}, \{a, (,), +, *\}, P_1', S),$$

where P_1' is defined as follows:

$$\begin{aligned} P_1' \;=\; & \{S \to (EA, E \to FBF, E \to FCF, F \to a, F \to (EA, \\ & A \to), B \to +, C \to *\}. \end{aligned}$$

Next we choose the ordering $S < E < F$. Then $X < Y$ holds already for each production $X \to Y\alpha$ of P_1', where X, Y are nonterminals. Thus, it remains to apply step (3) of the conversion algorithm:

- Replace the production $E \to FBF$ by the productions $E \to aBF$ and $E \to (EABF$.
- Replace the production $E \to FCF$ by the productions $E \to aCF$ and $E \to (EACF$.

This yields the grammar

$$G_2'' = (\{S, E, F, A, B, C\}, \{a, (,), +, *\}, P_1'', S),$$

where P_1'' is defined as follows:

$$\begin{aligned} P_1'' \;=\; & \{S \to (EA, E \to aBF, E \to (EABF, E \to aCF, E \to (EACF, F \to a, F \to (EA, \\ & A \to), B \to +, C \to *\}. \end{aligned}$$

□

Problem 9.2. [The Pumping Lemma for Context-Free Languages]

Prove that the following languages are not context-free by applying the Pumping Lemma for context-free languages (Theorem 3.14):

(a) $L_1 = \{ a^i b^i c^{2i} \mid i \geq 1 \}$,

(b) $L_2 = \{ a^i b^j c^k \mid 0 \leq i \leq j \leq k \}$,

(c) $L_3 = \{ ww \mid w \in \{a, b\}^* \}$.

Solution. (a) Assume that L_1 is context-free. Then by Theorem 3.14, there exists a constant r such that each word $z \in L_1$ of length $|z| \geq r$ has a factorization of the form $z = uvwxy$ such that $|vx| > 0$, $|vwx| \leq r$, and $uv^iwx^iy \in L_1$ for all $i \geq 0$.
We choose the word $z = a^r b^r c^{2r} \in L_1$. Then $|z| = 4r \geq r$, and hence, z has a factorization $z = uvwxy$ that satisfies the three conditions above. As $|vwx| \leq r$, we see that vwx is a factor of $a^r b^r$ or of $b^r c^{2r}$. Thus, the word $uv^0wx^0y = uwy$ contains less a's and/or less b's than z, but the same number of c's, or it contains the same number of a's, but less b's and/or c's. This implies that $uv^0wx^0y \notin L_1$, that is, L_1 does not satisfy the Pumping Lemma for Context-free languages. This proves that L_1 is not context-free.

(b) Assume that L_2 is context-free. Then by Theorem 3.14, there exists a constant r such that each word $z \in L_2$ of length $|z| \geq r$ has a factorization of the form $z = uvwxy$ such that $|vx| > 0$, $|vwx| \leq r$, and $uv^iwx^iy \in L_1$ for all $i \geq 0$.
We choose the word $z = a^r b^r c^r \in L_2$. Then $|z| = 3r \geq r$, and hence, z has a factorization $z = uvwxy$ that satisfies the three conditions above. As $|vwx| \leq r$, we see that vwx is a factor of $a^r b^r$ or of $b^r c^r$.

Case 1: vwx is a factor of $a^r b^r$. Then the word uv^2wx^2y contains more than r occurrences of the letter a and/or b, but it still just contains r occurrences of the letter c. It follows that $uv^2wx^2y \notin L_2$.

Case 2: vwx is a factor of $b^r c^r$. Then the word uv^0wx^0y contains less than r occurrences of the letter b and/or c, but it still contains r occurrences of the letter a. This implies that $uv^0wx^0y \notin L_2$.

As each of the two cases leads to a contradiction, we see that L_2 does not satisfy the Pumping Lemma for Context-free languages. This proves that L_2 is not context-free.

(c) Assume that L_3 is context-free. Then by Theorem 3.14, there exists a constant r such that each word $z \in L_3$ of length $|z| \geq r$ has a factorization of the form $z = uvwxy$ such that $|vx| > 0$, $|vwx| \leq r$, and $uv^iwx^iy \in L_1$ for all $i \geq 0$.
We choose the word $z = a^r b^r a^r b^r \in L_3$. Then $|z| = 4r \geq r$, and hence, z has a factorization $z = uvwxy$ that satisfies the three conditions above. As $|vwx| \leq r$, we see that vwx is a factor of the prefix $a^r b^r$, of the infix $b^r a^r$, or of the suffix $a^r b^r$. In each of these cases it is easily seen that the word uv^0wx^0y is not a square anymore, that is, $uv^0wx^0y \notin L_3$. Thus, L_3 does not satisfy the Pumping Lemma for Context-free languages. This proves that L_3 is not context-free. □

Problem 9.3. [Context-Free Languages]

Determine which of the following languages are context-free:

(a) $L_1 = \{ a^m b^n \mid 0 \leq m \leq n \leq 2m \}$,

(b) $L_2 = \{ w \in \{a, b, c\}^* \mid |w|_a = |w|_b = |w|_c \}$,

(c) $L_3 = \{ a^{n_1} b a^{n_2} b a^{n_3} \mid n_1 \geq n_2 \geq n_3 \geq 0 \}$.

Hint: You are expected to provide proofs for your answers!

Solution. (a) The language L_1 is context-free. Just take the context-free grammar

$$G_1 = (\{S\}, \{a, b\}, \{S \to aSb, S \to aSbb, S \to \varepsilon\}, S).$$

Then $S \to^m a^m S b^{m+k} \to a^m b^{m+k}$ for all $0 \leq m$ and $0 \leq k \leq m$. Hence, $L(G_1) = L_1$.

(b) The language L_2 is not context-free. Assume to the contrary that L_2 is context-free, and let r be the corresponding constant from the Pumping Lemma. Let $z = a^r b^r c^r$. Then $z \in L_2$ and $|z| = 3r > r$. Hence, z admits a factorization of the form $z = uvwxy$ such that $|vx| > 0$, $|vwx| \leq r$, and $uv^i wx^i y \in L_2$ for all $i \geq 0$. As $|vwx| \leq r$, we see that vwx is a factor of $a^r b^r$ or of $b^r c^r$. Hence, the word $uv^0 wx^0 y$ has less a- and/or b-symbols than c-symbols, or it has less b- and/or c-symbols than a-symbols, which implies that $uv^0 wx^0 y \notin L_2$. This contradiction implies that L_2 is not context-free.

(c) The language L_3 is not context-free. Assume to the contrary that L_3 is context-free, and let r be the corresponding constant from the Pumping Lemma. Let $z = a^r b a^r b a^r$. Then $z \in L_3$ and $|z| = 3r + 2 > r$. Hence, z admits a factorization of the form $z = uvwxy$ such that $|vx| > 0$, $|vwx| \leq r$, and $uv^i wx^i y \in L_3$ for all $i \geq 0$. From the definition of L_3 we see immediately that $|v|_b = |x|_b = 0$, that is, $v = a^s$ and $x = a^t$ for some $s, t \geq 0$ satisfying $0 < s + t \leq r$.

Case 1: If v is contained in the first factor a^r, then x is also contained in this factor or it is contained in the second factor of this form. In the former case it follows that $uv^0 wx^0 y = a^{r-s-t} b a^r b a^r \notin L_3$, as $r - s - t < r$, and in the latter case it follows that $uv^0 wx^0 y = a^{r-s} b a^{r-t} b a^r \notin L_3$, as $r - s < r$ or $r - t < r$.

Case 2: If v is contained in the second factor a^r, then $uv^2 wx^2 y = a^r b a^{r+s+t} b a^r$ or $uv^2 wx^2 y = a^r b a^{r+s} b a^{r+t}$. Then $r < r + s$ or $r < r + t$, which implies that $uv^2 wx^2 y \notin L_3$.

Case 3: If v is contained in the third factor a^r, then so is x, and $uv^2 wx^2 y = a^r b a^r b a^{r+s+t} \notin L_3$.

Thus, we see that L_3 does not satisfy the Pumping Lemma for context-free languages, which implies that L_3 is not context-free. □

Problem 9.4. [Pushdown Automata]

Give an example of an accepting computation for the following PDA

$$M = (\{q_0, q_1, q_2, q_3\}, \{a, b\}, \{A, \#\}, \delta, q_0, \#, \{q_3\}),$$

where δ is given by

δ	q_0	q_1	q_2	q_3
$(a, \#)$	$(q_1, \#A)$	$-$	$-$	$-$
(a, A)	$-$	(q_1, AA)	$-$	$-$
$(b, \#)$	$-$	$-$	$-$	$-$
(b, A)	$-$	(q_2, A)	(q_3, ε)	(q_2, A)

,

and determine the languages that are accepted by it by considering both acceptance conditions:

(1) acceptance by final state and

(2) acceptance by empty pushdown.

Solution. On input $a^m b^n$ $(m, n \geq 1)$, M_1 can execute the following computation:

$$\begin{array}{llllll} (q_0, \#, a^m b^n) & \vdash_{M_1} & (q_1 \#A, a^{m-1} b^n) & \vdash_{M_1}^{m-1} & (q_1, \#A^m, b^n) \\ & \vdash_{M_1} & (q_2, \#A^m, b^{n-1}) & \vdash_{M_1} & (q_3, \#A^{m-1}, b^{n-2}) \\ & \vdash_{M_1} & (q_2, \#A^{m-1} b^{n-3}) & \vdash_{M_1} & (q_3, \#A^{m-2}, b^{n-4}). \end{array}$$

Now it follows that $L(M_1) = \{\, a^m b^{2n} \mid m \geq n \geq 1 \,\}$, while $N(M_1) = \emptyset$, as the bottom marker # cannot be popped from the pushdown. □

Problem 9.5. [Pushdown Automata]

Prove Theorem 3.18, that is, from a given PDA M_2, construct a PDA M_1 such that $L(M_1) = N(M_2)$.

Solution. Let $M_2 = (Q, \Sigma, \Gamma, \delta, q_0, \#, \emptyset)$ be a PDA, and let $L = N(M)$ be the language it accepts with empty pushdown. We construct a PDA $M_1 = (Q \cup \{p_0, p_f\}, \Sigma, \Gamma \cup \{\&\}, \delta_1, p_0, \#, \{p_f\})$ by defining the transition relation δ_1 as follows:

$$\begin{array}{lcll} \delta_1(p_0, \varepsilon, \#) & = & \{(q_0, \&\#)\}, & \\ \delta_1(q, a, B) & = & \delta(q, a) & \text{for all } q \in Q, a \in \Sigma \cup \{\varepsilon\}, \text{ and } B \in \Gamma, \\ \delta_1(q, \varepsilon, \&) & = & \{(p_f, \varepsilon)\}. & \end{array}$$

Then M_1 proceeds as follows. Given a word $w \in \Sigma^*$ as input, it starts from the initial configuration $(p_0, \#, w)$ and enters the configuration $(q_0, \&\#, w)$. Now it simulates a computation of M_2 that starts from the configuration $(q_0, \#, w)$. If the latter reaches an accepting configuration of the form $(q, \varepsilon, \varepsilon)$, then M_1 reaches the configuration $(q, \&, \varepsilon)$, from which it can reach the accepting configuration $(p_f, \varepsilon, \varepsilon)$. Conversely, if M_1 accepts on input w, then the last transition in a corresponding accepting computation is the step from $(q, \&, \varepsilon)$ to $(p_f, \varepsilon, \varepsilon)$, as this is the only way in which M_1 can enter its final state. Thus, from $(q_0, \&\#, w)$ to $(q, \&, \varepsilon)$ it must simulate a computation of M_2 that transfers $(q_0, \#, w)$ into $(q, \varepsilon, \varepsilon)$, which implies that $w \in N(M_2)$. This shows that $L(M_1) = N(M_2)$, as required. □