Solutions of Assignment 5

> *Theory of Computation, Fall 2022*

Corrections

- **Q2.** The key to Q2 and corresponding contents in notes are corrected.

Keys

Q1

$$\begin{array}{c} P = (K, \Sigma, \Gamma, \Delta, s, F) \\ \begin{array}{rcl} K & = & \{q_1\} \\ \Sigma & = & \{0, 1\} \\ \Gamma & = & \{\$\} \\ s & = & q_1 \\ F & = & \{q_1\} \end{array} \end{array} \tag{1}$$

The list of $((q, a, \beta), (p, \gamma))$ shown below contains all elements in Δ.

(q, a, β)	(p, γ)
$(q_1, 0, \$)$	(q_1, e)
$(q_1, 1, e)$	$(q_1, \$)$
$(q_1, e, \$)$	(q_1, e)

Q2

$$\begin{array}{c} P = (K, \Sigma, \Gamma, \Delta, s, F) \\ \begin{array}{rcl} K & = & \{q_1\} \\ \Sigma & = & \{a, b\} \\ \Gamma & = & \{A, B\} \\ s & = & q_1 \\ F & = & \{q_1\} \end{array} \end{array} \tag{2}$$

The list of $((q, a, \beta), (p, \gamma))$ shown below contains all elements in Δ.

(q, a, β)	(p, γ)
(q_1, a, e)	(q_1, A)
(q_1, a, B)	(q_1, e)
(q_1, b, e)	(q_1, BB)
(q_1, b, A)	(q_1, B)
(q_1, b, AA)	(q_1, e)

Q3

$$L = \emptyset \tag{3}$$

Q4

Given $G = (V, \Sigma, R, S)$, construct a PDA $P = (K, \Sigma, \Gamma, \Delta, s, F)$ like follows:

$$\begin{aligned} K &= \{s, f\} \\ \Sigma &= \{0, 1\} \\ \Gamma = V &= \{0, 1, S, A\} \\ s &= s \\ F &= \{f\} \end{aligned} \tag{4}$$

The list of $((q, a, \beta), (p, \gamma))$ shown below contains all elements in Δ.

(q, a, β)	(p, γ)
(s, e, e)	(f, S)
(f, e, S)	$(f, 0S1)$
(f, e, S)	(f, A)
(f, e, A)	(f, S)
$(f, 0, 0)$	(f, e)
$(f, 1, 1)$	(f, e)

Q5

Proof:

It is known that there is a NFA $M = (K, \Sigma, \Delta, s, F)$ that accepts regular language A.

Construct a PDA $P = (K, \Sigma, \Gamma, \Delta', s, F)$ like follows:

$$\begin{aligned} \Gamma &= \emptyset \\ \Delta' &= \{((q, a, e), (p, e)) \mid ((q, a), p) \in \Delta\} \end{aligned} \tag{5}$$

We can easily prove that P accepts A.

Therefore, A is context-free.

Q6

It is known that there is a NFA $M_A = (K_A, \Sigma_A, \Delta_A, s_A, F_A)$ that accepts regular language A and a NFA $M_B = (K_B, \Sigma_B, \Delta_B, s_B, F_B)$ that accepts regular language B.

Construct a PDA $P = (K, \Sigma, \Gamma, \Delta, s, F)$ like follows:

$$\begin{array}{rcl} K &=& K_A \cup K_B \\ \Sigma &=& \Sigma_A \cup \Sigma_B \\ \Gamma &=& \{\$\} \\ s &=& s_A \\ F &=& F_B \\ \Delta &=& \{((q_A, a, e), (p_A, \$)) \mid ((q_A, a), p_A) \in \Delta_A\} \quad \cup \\ & & \{((q_B, b, \$), (p_B, e)) \mid ((q_B, b), p_B) \in \Delta_B\} \quad \cup \\ & & \{((f_A, e, e), (s_B, e)) \mid f_A \in F_A\} \end{array} \tag{6}$$

We can easily prove that P accepts $A \diamond B = \{xy \mid x \in A, y \in B, |x| = |y|\}$.

Therefore, $A \diamond B$ is context-free.

Grading

100 pts. in total

- Q1. 20 pts.
- Q2. 20 pts.
- Q3. 10 pts.
- Q4. 15 pts.
- Q5. 15 pts.
- Q6. 20 pts.

Notes

Q1

Idea: Cause an "underflow" of the stack of the constructed PDA at any time meeting more 0's than 1's when parsing the string. This ensures no prefix has more 0's than 1's.

Q2

Idea: w has twice as many a's as b's, so when we read a b, we behave as if we read two symbols. That is, we push or pop twice as many symbols when meeting a b.

To avoid stack "underflow", the situations that either a or b leads the string should both be considered. Moreover, $((q_1, b, A), (q_1, B)) \in \Delta$ is indispensable in order to accept strings like aba.

To sum up, when we read an b, (i) we push two B's onto the stack, or (ii) pop two A's, or (iii) pop an A and push a B.

Q3

The derivation never halts since there is no rule yielding only terminal states.

Note that $e \notin L$. The correct answer is $L = \emptyset$ instead of $L = \{e\}$.

Q4

Note that two states in K, one as the initial state and the other as the only final state, are indispensable, or the PDA would accept the empty string e.

Q5

In fact, it is also a good idea to prove using the CFG. You can search the idea of regular grammar. It is known that FAs are equivalent to regular grammars, whose rules is the subset of that of CFGs, in computability.

Q6

The PDA uses the feature of non-determinism, so we do not have to deal with the annoying dead state.
