

# Introduction to the Theory of Computation 2021 — Midterm 2

## Solutions

**Problem 1 (15 pts).** In our lecture, we proved the language

$$\{1^{n^2} \mid n \geq 0\}.$$

is not regular by choosing  $i = 2$  in the pumping lemma, and showed that there exists  $x, y, z$  such that  $xy^2z$  is not in the language for all possible length  $p$ .

- (a) (5 pts) Is it possible to use  $i = 3$  for the proof?
- (b) (10 pts) We mentioned some issues in using  $i = 0$ . If the language is changed to

$$\{1^{n^2} \mid n \geq 1\},$$

can  $i = 0$  be used?

*Solution.*

- (a) Yes, we can use  $i = 3$  in the proof. Given  $p > 0$ , and take

$$s = 1^{p^2} \in \{1^{n^2} \mid n \geq 0\}.$$

Suppose

$$s = xyz,$$

and it satisfies

$$|xy| \leq p$$

and

$$|y| > 0.$$

Therefore, we imply that

$$p^2 < |xy^3z| \leq p^2 + 2p = (p+1)^2 - 1 < (p+1)^2,$$

which means that  $xy^3z$  is not in

$$\{1^{n^2} \mid n \geq 0\}.$$

Hence, we have done the proof.

(b) Yes, if the language is

$$\{1^{n^2} \mid n \geq 1\},$$

we can use  $i = 0$  in the proof. Let us focus on the inequality

$$(p-1)^2 \leq p^2 - p$$

in the slides. If

$$p \geq 2,$$

it can be derived that

$$(p-1)^2 = p^2 - p + (-p+1) < p^2 - p \leq |xy^0z| < p^2,$$

which implies that

$$xy^0z \notin \{1^{n^2} \mid n \geq 1\}.$$

Thus, we only need to consider the case

$$p = 1.$$

When

$$p = 1,$$

$y$  must be “1” by the condition

$$|y| > 0$$

of the pumping lemma. Hence, we find that

$$x = z = \epsilon,$$

and

$$xy^0z = \epsilon \notin \{1^{n^2} \mid n \geq 1\}.$$

Overall, we have done the proof.

Note: for this problem, we do not state explicitly, but we meant the same proof in the lecture. That is,  $s$  is the same. Thus, your answer is considered incorrect if you check other  $s$ . We think our description has implied that. Moreover, you should ask if things are not clear.

**Problem 2 (10 pts).** In proving the opposite statement of the pumping lemma, i.e., Eq.(1) on page 5 of chap1\_pumpinglemma3, we used the truth table to show that the opposite of

$$A \& B$$

is

$$A \rightarrow \neg B.$$

Consider

$$A \& B \& C.$$

Give the detailed truth table to show that the opposite is

$$A \& B \rightarrow \neg C.$$

*Solution.*

$A$	$B$	$C$	$A \& B \& C$	$\neg(A \& B \& C)$	$A \& B$	$\neg C$	$A \& B \rightarrow \neg C$
0	0	0	0	1	0	1	1
0	0	1	0	1	0	0	1
0	1	0	0	1	0	1	1
0	1	1	0	1	0	0	1
1	0	0	0	1	0	1	1
1	0	1	0	1	0	0	1
1	1	0	0	1	1	1	1
1	1	1	1	0	1	0	0

From the truth table we can see that  $\neg(A \& B \& C)$  is indeed equivalent to  $A \& B \rightarrow \neg C$ .

**Problem 3 (25 pts).** Are the following two languages regular? In using the pumping lemma, your  $s$  and  $i$  must be clearly given. You cannot just roughly say the existence of them.

(a) (15 pts)

$$L_1 = \{\mathbf{w} \mid w_k \in \{0, 1\}, |\mathbf{w}| \neq 0, \frac{\sum_k w_k}{|\mathbf{w}|} \leq 0.9\},$$

where  $w_k$  is the  $k$ th component of  $\mathbf{w}$ . You must give clear explanation instead of just giving the answer.

(b) (10 pts)

$$L_2 = \{0^h 1^k \mid h > 5k \text{ and } h, k > 0\}$$

*Solution.*

(a)  $L_1$  is not a regular language. Given  $p > 0$ , we take

$$s = 1^{9p} 0^p \in L_1, \text{ where } \frac{\sum_k s_k}{|s|} = 0.9.$$

Thereby, we have

$$y = 1 \cdots 1 = 1^a, \text{ where } 0 < a < p,$$

since

$$|xy| \leq p \text{ and } |y| > 0.$$

Therefore, when

$$i = 2,$$

we have

$$xy^2z = 1^{9p+a} 0^p.$$

Since

$$\frac{9p+a}{10p+a} > \frac{9}{10},$$

we know that

$$xy^2z \notin L_1.$$

(b)  $L_2$  is not a regular language. Given  $p > 0$ , we take

$$s = 0^{5p+1}1^p \in L_2.$$

Hence, we have

$$y = 0 \cdots 0$$

because

$$|xy| \leq p.$$

Let us see the case

$$xy^0z = 0^{5p+1-n}1^p,$$

where

$$n = |y| > 1.$$

Therefore,

$$xy^0z \notin L_2$$

since

$$5p + 1 - n \leq 5p.$$

**Problem 4 (20 pts).** Consider the following language:

$$\{w \in \{0, 1\}^* \mid \text{the sum of } w \bmod 6 = 0\}, \quad (1)$$

where the sum of  $w$  means the sum of the symbols in  $w$ . Note that we define

$$\text{the sum of } w = 0 \text{ if } w = \epsilon.$$

- (a) (10 pts) Using a proof similar to what we did in the previous exam, we can show that this language can not be recognized by a DFA with less than 6 states. However, this may not hold for PDA. Please give a PDA that recognizes this language where the number of states is **less than or equal to 2**. Besides the state diagram, you should also give its **formal definition**.
- (b) (10 pts) Give a CFG for this language, where the number of rules and variables are less than or equal to **5** and **3**, respectively. For easy grading, please use  $S$  as the start variable and use  $A$  and  $B$  as the name of other variables.

*Solution.*

- (a) The idea is to “store the state” on the stack. The formal definition of a PDA that recognizes the language is

$$M = (Q, \Sigma, \Gamma, \delta, q_0, F)$$

$$Q = \{q_0, q_1\}$$

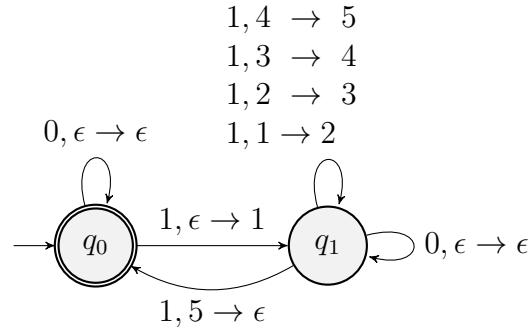
$$\Sigma = \{0, 1\}$$

$$\Gamma = \{1, 2, 3, 4, 5\}$$

$$F = \{q_0\}$$

$\delta$ :	0		1					
input								
stack	1, 2, 3, 4, 5	$\epsilon$	1	2	3	4	5	$\epsilon$
$q_0$	$\emptyset$	$\{(q_0, \epsilon)\}$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\{(q_1, 1)\}$
$q_1$	$\emptyset$	$\{(q_1, \epsilon)\}$	$\{(q_1, 2)\}$	$\{(q_1, 3)\}$	$\{(q_1, 4)\}$	$\{(q_1, 5)\}$	$\{(q_0, \epsilon)\}$	$\emptyset$

and  $\delta(q, \epsilon, x) = \emptyset \forall q \in Q, x \in \Gamma$ . The state diagram of it is as follows:



(b) The following CFG generates the language:

$$\begin{aligned}
 S &\rightarrow AS \mid B \\
 A &\rightarrow B1B1B1B1B1B1 \\
 B &\rightarrow 0B \mid \epsilon
 \end{aligned}$$

There are also CFG that use less rules, for example:

$$S \rightarrow SS \mid S1S1S1S1S1S1 \mid 0 \mid \epsilon$$

or

$$S \rightarrow S1S1S1S1S1S1S \mid 0S \mid \epsilon$$

**Problem 5 (10 pts).** Consider the following CFG.

$$\begin{aligned}
 S &\rightarrow CS \mid \epsilon \\
 C &\rightarrow SCS \mid CS \mid 1
 \end{aligned}$$

Transform it to CNF by following the procedures in Theorem 2.9.

- (i): Add a new start variable  $S_0$  and the rule  $S_0 \rightarrow S$ .
- (ii): Take care of all  $\epsilon$ -rules.
- (iii): Handle all unit rules.
- (iv): Convert all remaining rules into the proper form.

You need to show details of the steps.

*Solution.*

Add  $S_0$

$$\begin{aligned}
 S_0 &\rightarrow S \\
 S &\rightarrow CS \mid \epsilon \\
 C &\rightarrow SCS \mid CS \mid 1
 \end{aligned}$$

Remove  $S \rightarrow \epsilon$

$$\begin{aligned} S_0 &\rightarrow S \mid \epsilon \\ S &\rightarrow CS \mid C \\ C &\rightarrow SCS \mid CS \mid 1 \mid C \mid SC \end{aligned}$$

Remove  $C \rightarrow C$

$$\begin{aligned} S_0 &\rightarrow S \mid \epsilon \\ S &\rightarrow CS \mid C \\ C &\rightarrow SCS \mid CS \mid 1 \mid SC \end{aligned}$$

Remove  $S \rightarrow C$

$$\begin{aligned} S_0 &\rightarrow S \mid \epsilon \\ S &\rightarrow CS \mid SCS \mid 1 \mid SC \\ C &\rightarrow SCS \mid CS \mid 1 \mid SC \end{aligned}$$

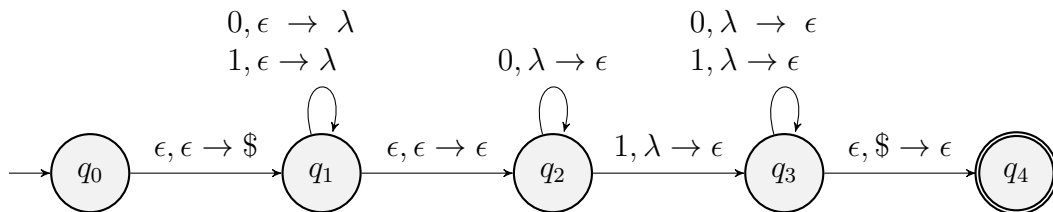
Remove  $S_0 \rightarrow S$

$$\begin{aligned} S_0 &\rightarrow CS \mid SCS \mid 1 \mid SC \mid \epsilon \\ S &\rightarrow CS \mid SCS \mid 1 \mid SC \\ C &\rightarrow SCS \mid CS \mid 1 \mid SC \end{aligned}$$

Remove  $SCS$

$$\begin{aligned} S_0 &\rightarrow CS \mid SB \mid 1 \mid SC \mid \epsilon \\ S &\rightarrow CS \mid SB \mid 1 \mid SC \\ C &\rightarrow SB \mid CS \mid 1 \mid SC \\ B &\rightarrow CS \end{aligned}$$

**Problem 6 (20 pts).** Consider the following PDA:

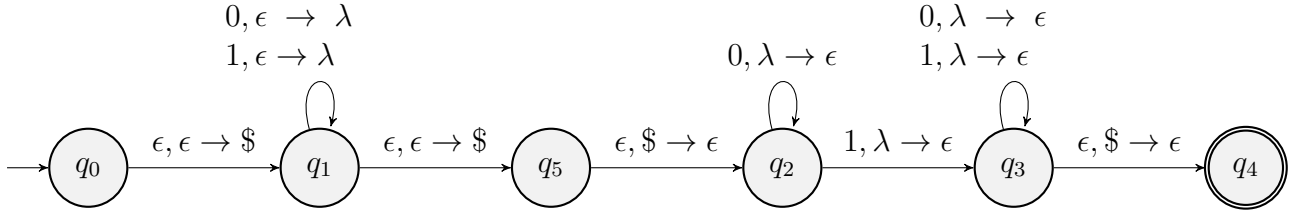


where  $\Sigma = \{0, 1\}$  and  $\Gamma = \{\lambda, \$\}$ .

- (10 pts) What is the language recognized by this PDA? (i.e., what kinds of strings are accepted?) You should give explanations instead of just the answer.
- (10 pts) Please follow the procedure in slide chap2\_PDA4 and chap2\_PDA5 (Lemma 2.27 in text) to generate a CFG for the corresponding language. First, modify the PDA to satisfy the preconditions (page 2 of chap2\_PDA4), and give the equivalent CFG. You need to specify all the rules, but you need not list all of them one by one. You are required to use the same  $\Sigma$  and  $\Gamma$  in your modified PDA.

*Solution.*

- (a) By pushing a \$ at the beginning and popping it at the end, we can make sure that all  $\lambda$ s are consumed at the end of the string. The PDA pushes a  $\lambda$  for the first half of the string and pops them off in the second half, thus making sure that it has even length. Moreover, it must have a 1 in the second half to go from  $q_2$  to  $q_3$ . Therefore, this PDA recognizes the strings that have even length and has at least a 1 in the second half. More precisely, the language is  $\{ \{0,1\}^i 0^j 1 \{0,1\}^k \mid i = j + 1 + k \}$ .
- (b) Since \$ is pushed at the start and popped before entering final state, the stack must be empty upon acceptance. Also, it has only one accept state. Thus, we only have to make sure each transition either push or pop by adding an extra state:



Now, we apply the procedure. The variables in the CFG are  $A_{ij}$  for  $i, j \in \{0, \dots, 5\}$  and the start variable is  $A_{04}$ . Now we construct the rules by considering each stack symbol:

- (a) For \$, we add the rules

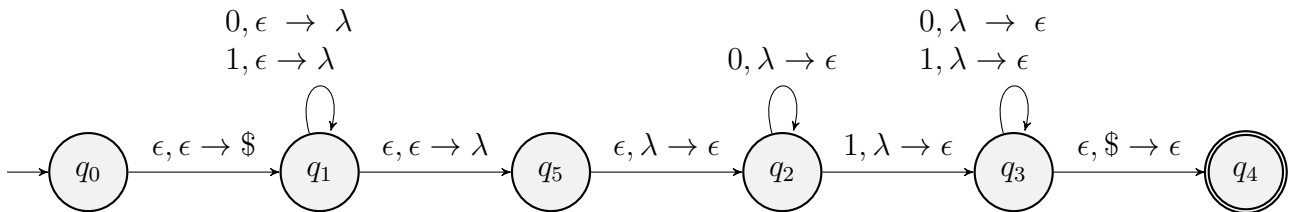
$$\begin{aligned} A_{04} &\rightarrow \epsilon A_{13} \epsilon \\ A_{12} &\rightarrow \epsilon A_{55} \epsilon \\ A_{02} &\rightarrow \epsilon A_{15} \epsilon \\ A_{14} &\rightarrow \epsilon A_{53} \epsilon \end{aligned}$$

- (b) For  $\lambda$ , we have the rules

$$\begin{aligned} A_{12} &\rightarrow 0A_{12}0 \mid 1A_{12}0 \\ A_{13} &\rightarrow 0A_{13}0 \mid 0A_{13}1 \mid 1A_{13}0 \mid 1A_{13}1 \mid 0A_{12}1 \mid 1A_{12}1 \end{aligned}$$

The remaining rules are  $A_{pp} \rightarrow \epsilon$  for  $p \in \{0, \dots, 5\}$  and  $A_{pq} \rightarrow A_{pr}A_{rq}$  for  $p, q, r \in \{0, \dots, 5\}$ .

**Alternative solution:**



If we use the stack symbol  $\lambda$  for the transitions of  $q_5$  instead, we will get the following result by applying the procedure:

- (a) for \$, we add the rules:

$$A_{04} \rightarrow \epsilon A_{13} \epsilon$$

(b) for  $\lambda$ , we add the rules:

$$\begin{aligned} A_{12} &\rightarrow 0A_{15}\epsilon \mid 1A_{15}\epsilon \mid 0A_{12}0 \mid 1A_{12}0 \mid \epsilon A_{52}0 \mid \epsilon A_{55}\epsilon \\ A_{13} &\rightarrow 0A_{13}0 \mid 1A_{13}0 \mid 0A_{13}1 \mid 1A_{13}1 \mid 0A_{12}1 \mid 1A_{12}1 \\ &\mid \epsilon A_{52}1 \mid \epsilon A_{53}0 \mid \epsilon A_{53}1 \end{aligned}$$

The remaining rules are the same. These two CFG should be equivalent since a symbol that is pushed and popped immediately does not affect the language recognized by the PDA.

**Common mistake:** Note that we required that  $\Sigma$  and  $\Gamma$  to stay the same.