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 statifies

$$|xy| \leq p$$

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.

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	$A\&B\&C$	$\neg(A\&B\&C)$	$A\&B$	$\neg C$	$A\&B \to \neg C$
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 $A\&B \to \neg C$.

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)

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

where w_k is the kth component of $\boldsymbol{w}$. 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

$$s = 1^p 0^p \in L_1, \text{ where } \frac{\sum_k s_k}{|s|} = 0.5 \leq 0.9.$$

Thereby, we have

$$y = 1 \cdots 1,$$

since

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

Therefore, we can imply that

$$\frac{\sum_k (xy^i z)_k}{|xy^i z|} \to 1$$

as $i \to \infty$. That is, there exists an integer $8p + 2$ such that

$$\begin{aligned}
&\sum_k (xy^i z)_k - 0.9|xy^i z| \\
=&p + (i-1)|y| - 0.9 \cdot (2p + (i-1)|y|) \\
=&p + (i-1)|y| - 1.8p - 0.9(i-1)|y|) \\
=&-0.8p + 0.1(i-1) \cdot |y| \\
\geq&-0.8p + 0.1(i-1) \cdot 1 \\
>&0
\end{aligned}$$

for all $i \geq 8p+2$, which implies that

$$\frac{\sum_k (xy^iz)_k}{|xy^iz|} > 0.9.$$

Hence,

$$xy^iz \notin L_1, \ \forall i \geq 8p+2.$$

(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:

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

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.

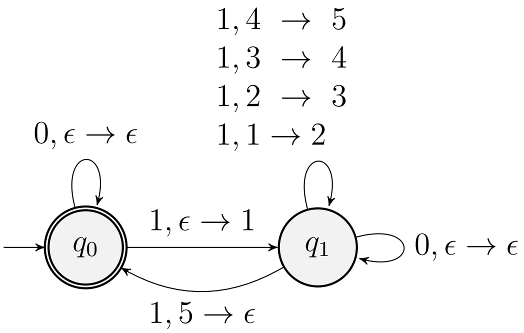
(a) The idea is to "store the state" on the stack. The formal definition of a PDA that recognizes the language is

$$\begin{aligned} M &= (Q, \Sigma, \Gamma, \delta, q_0, F) \\ Q &= \{q_0, q_1\} \\ \Sigma &= \{0, 1\} \\ \Gamma &= \{1, 2, 3, 4, 5\} \\ F &= \{q_0\} \end{aligned}$$

δ:

input	0		1					
stack	$1, 2, 3, 4, 5$	ϵ	1	2	3	4	5	ϵ
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:


$1, 4 \rightarrow 5$
$1, 3 \rightarrow 4$
$1, 2 \rightarrow 3$
$1, 1 \rightarrow 2$
$0, \epsilon \rightarrow \epsilon$
$1, \epsilon \rightarrow 1$
q_0
q_1
$0, \epsilon \rightarrow \epsilon$
$1, 5 \rightarrow \epsilon$

(b) The following CFG generates the language:

$$\begin{aligned} S &\rightarrow AS \mid B \\ A &\rightarrow B1B1B1B1B1 \\ B &\rightarrow 0B \mid \epsilon \end{aligned}$$

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

$$S \rightarrow SS \mid S1S1S1S1S1 \mid 0 \mid \epsilon$$

or

$$S \rightarrow S1S1S1S1S1S \mid 0S \mid \epsilon$$

.

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 $S_0 \to S$.

(ii): Take care of all ϵ-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

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

Remove $S \to \epsilon$

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

Remove $C \to C$

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

Remove $S \to C$

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

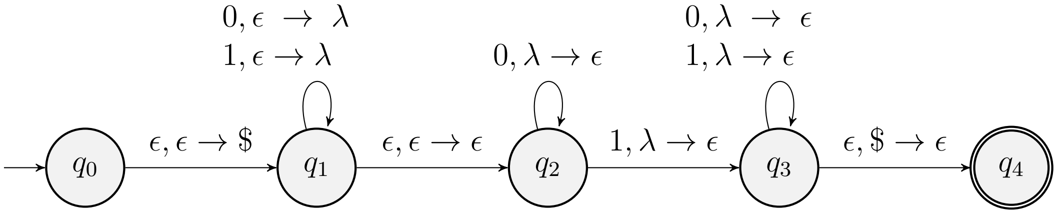
Remove $S_0 \to S$

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

Remove SCS

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

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


$0, \epsilon \to \lambda$
$1, \epsilon \to \lambda$
$0, \lambda \to \epsilon$
$0, \lambda \to \epsilon$
$1, \lambda \to \epsilon$
$\epsilon, \epsilon \to \$$
$\epsilon, \epsilon \to \epsilon$
$1, \lambda \to \epsilon$
$\epsilon, \$ \to \epsilon$
q_0
q_1
q_2
q_3
q_4

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

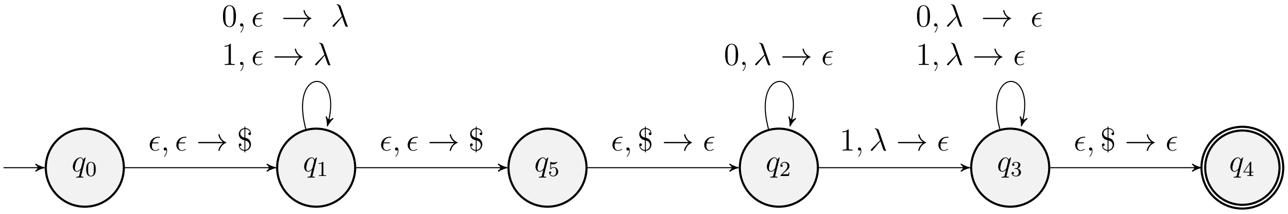
(a) (10 pts) What is the language recognized by this PDA? (i.e., what kinds of strings are accepted?) You should give explanations instead of just the answer.

(b) (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 Σ and Γ in your modified PDA.

Solution.

(a) By pushing a \$ at the beginning and popping it at the end, we can make sure that all λs are consumed at the end of the string. The PDA pushes a λ 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 $\left\{\{0,1\}^i 0^j 1 \{0,1\}^k \mid i = j + 1 + k\right\}$.

(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:


$0, \epsilon \to \lambda$
$1, \epsilon \to \lambda$
$0, \lambda \to \epsilon$
$0, \lambda \to \epsilon$
$1, \lambda \to \epsilon$
$\epsilon, \epsilon \to \$$
$\epsilon, \epsilon \to \$$
$\epsilon, \$ \to \epsilon$
$1, \lambda \to \epsilon$
$\epsilon, \$ \to \epsilon$
q_0
q_1
q_5
q_2
q_3
q_4

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

(a) For \$, we add the rules

$$A_{04} \to \epsilon A_{13} \epsilon$$
$$A_{12} \to \epsilon A_{55} \epsilon$$

(b) For λ, we have the rules

$$A_{12} \to 0A_{12}0 \mid 1A_{12}0$$
$$A_{13} \to 0A_{13}0 \mid 0A_{13}1 \mid 1A_{13}0 \mid 1A_{13}1 \mid 0A_{12}1 \mid 1A_{12}1$$

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