Problem 1 (15 pts). Consider $\Sigma = \{0, 1\}$ and the language

$$A = \{0^m1^{m+n}0^n \mid m, n \geq 1\}.$$

(a) (10 pts) Construct a PDA $P_1$ with $\leq 5$ states to recognize $A$. Please check (b) and you are required to ensure $P_1$ satisfies the three conditions and no modification is needed.

(b) (5 pts) Convert your PDA $P_1$ to a CFG by using the procedure in Lemma 2.27 of the textbook. For simplicity, you only need to write each $A_{pq} \to aA_{rs}b$ rule. The rules $A_{pq} \to A_{pr}A_{rq}$ and $A_{pp} \to \epsilon$ are not needed. In order to prepare for $A_{pq} \to aA_{rs}b$ rules, please give table(s) for each stack alphabet $u$ pushed/popped, similar to what we had in slides.

Solution.

(a) See the following diagram.

We see that the three conditions are satisfied:

(a) Single accept state
(b) By using $\$, the stack is emptied before accepting
(c) Every transition either pushes or pops, but not both

(b) See the following tables.

- $u = 1$:  
  \[
  \begin{array}{cccccc}
  p & r & s & q & a & b \\
  3 & 3 & 3 & 4 & 1 & 0 \\
  3 & 3 & 4 & 4 & 1 & 0 \\
  \end{array}
  \]

- $u = \$:  
  \[
  \begin{array}{cccccc}
  p & r & s & q & a & b \\
  1 & 2 & 4 & 5 & \epsilon & \epsilon \\
  \end{array}
  \]

- $u = 0$:  
  \[
  \begin{array}{cccccc}
  p & r & s & q & a & b \\
  2 & 2 & 2 & 3 & 0 & 1 \\
  2 & 2 & 3 & 3 & 0 & 1 \\
  \end{array}
  \]
Problem 2 (25 pts). Consider $\Sigma = \{0, 1\}$ and the language

$$B = \{0^m1^n \mid m \geq n \geq 0\}.$$ 

(a) (10 pts) Construct a PDA $P_2$ with $\leq 2$ states to recognize $B$. The stack alphabet is restricted to be $\Gamma = \{0\}$.

Draw the state diagram and give the formal definition of your PDA $P_2$. Hint: See (c), so you may want to have a diagram as deterministic as possible. Transitions like $\varepsilon, \varepsilon \rightarrow \varepsilon$ may make (c) difficult.

(b) (10 pts) Prove that a PDA with only one state and with $\Gamma = \{0\}$ cannot recognize $B$. Your proof must be clearly written.

(c) (5 pts) Construct a DPDA with $\leq 3$ states (including the $q_r$ state that was introduced in examples in our slides) for $B$ and show the table of $\delta$.

Solution.

(a) See the following diagram.

$$P_2 = (Q, \Sigma, \Gamma, \delta, q_0, F), \text{ where } Q = \{q_0, q_1\}, \Gamma = \{0\}, F = Q, \text{ and } \delta \text{ is shown below.}$$

(b) Suppose $P_2$ has only one state $q$. To accept $\varepsilon$, $q$ must be an accept state. Then every rule loop from $q$ to itself.
Proof 1: Because $01 \in B$, it must be accepted by $P_2$. The machine has an empty stack in the beginning. Therefore, regardless of whether the stack after processing $01$ is empty or not, the machine $P_2$ can take the same path before to process another $01$. Therefore, $0101$ is accepted, a contradiction. So $P_2$ must have at least two states.

Proof 2: One reasoning:

- Consider the string $1$, which should be rejected. Consider the four kinds of transitions when a $1$ is read.

\[
\begin{align*}
&1, \epsilon \rightarrow \epsilon \\
&1, \epsilon \rightarrow 0 \\
&1, 0 \rightarrow \epsilon \\
&1, 0 \rightarrow 0
\end{align*}
\]

There cannot be $1, \epsilon \rightarrow \epsilon$ or $1, \epsilon \rightarrow 0$; otherwise $1$ would be accepted.

- Consider the string $01$, which should be accepted. To read $1$, with the properties from the previous paragraph, a rule must be $1, 0 \rightarrow \epsilon$ or $1, 0 \rightarrow 0$. Thus before reading $1$, a $0$ must be pushed to the stack. That is, we need a rule either $0, \epsilon \rightarrow 0$ or $\epsilon, \epsilon \rightarrow 0$. However, the rule $\epsilon, \epsilon \rightarrow 0$ along with either $1, 0 \rightarrow \epsilon$ or $1, 0 \rightarrow 0$ would accept $1$, so the rule $0, \epsilon \rightarrow 0$ is needed.

- We conclude that the diagram must have at least

\[
\begin{align*}
&0, \epsilon \rightarrow 0, \quad 1, 0 \rightarrow \epsilon \\
&\text{or} \\
&0, \epsilon \rightarrow 0, \quad 1, 0 \rightarrow 0
\end{align*}
\]

The former accepts $0101$ and the latter accepts $011$, but neither should be accepted. Thus it is impossible to have only one state with $\Gamma = \{0\}$.

Alternative reasoning:

- To reject $1$, we have $\delta(q, 1, \epsilon) = \emptyset$. This implies $\delta(q, 1, 0) \neq \emptyset$; otherwise $1$ can never be processed, but strings such as $01$ are in the language $B$. We then know that when we read a $1$ there need to be at least a $0$ on the stack.

- Consider the initial empty stack, when there is nothing to pop. We cannot have both $\delta(q, 0, \epsilon) = \delta(q, \epsilon, \epsilon) = \emptyset$, because along with $\delta(q, 1, \epsilon) = \emptyset$ from the previous paragraph, no input can be processed. Therefore, either $\delta(q, 0, \epsilon) \neq \emptyset$ or $\delta(q, \epsilon, \epsilon) \neq \emptyset$.

- If $(q, 0) \in \delta(q, 0, \epsilon)$, then $010$ would be accepted. If $(q, 0) \in \delta(q, \epsilon, \epsilon)$, then $1$ would be accepted. But neither of $010$ or $1$ should be accepted. Thus we have $(q, 0) \notin \delta(q, 0, \epsilon)$ and $(q, 0) \notin \delta(q, \epsilon, \epsilon)$. This means before reading any $1$, no $0$ can be pushed to the stack, and thus $1$ can never be processed. Thus it is impossible to have only one state with $\Gamma = \{0\}$.

(c) See the following diagram and table.

\[
\begin{align*}
&0, \epsilon \rightarrow 0 \\
&1, 0 \rightarrow \epsilon \\
&\epsilon, \epsilon \rightarrow \epsilon
\end{align*}
\]

\[
\begin{align*}
&0, \epsilon \rightarrow 0 \\
&1, 0 \rightarrow \epsilon \\
&\epsilon, \epsilon \rightarrow \epsilon
\end{align*}
\]

\[
\begin{align*}
&0, \epsilon \rightarrow 0 \\
&1, 0 \rightarrow \epsilon \\
&\epsilon, \epsilon \rightarrow \epsilon
\end{align*}
\]
Table 2: $\delta$. Blank entries signify $\emptyset$.

For the first row, from $\delta(q_0, 0, \epsilon) \neq \emptyset$ and $\delta(q_0, 1, 0) \neq \emptyset$, all other entries are $\emptyset$. For the second row, from $\delta(q_1, 1, 0) \neq \emptyset$, we have

Table 3:

Thus we can have either $\delta(q_1, 0, 0) \neq \emptyset$ or $\delta(q_1, 0, \epsilon) \neq \emptyset$, but not both. For the third row, by $\delta(q_r, \epsilon, \epsilon) \neq \emptyset$, all other entries are $\emptyset$.

**Problem 3 (20 pts).** Consider $\Sigma = \{0, 1\}$ and the language

$$C = \{0^m1^n \mid 0 \leq m \leq n\}.$$ 

Some propose the following PDA $P_3$.

(a) (5 pts) Check whether $P_3$ works for 000111 by drawing the simulation tree. Recall that in our PDA (part 2) slides, a tree is drawn like the following with input $aabcc$. 

```
  q_0 1, 0 $\rightarrow$ q_1
           / \                     / \       / \       / \       / \       / \\
 q_1 \{a,a,\} q_2 \{a,\} q_3 \{a,\} q_4 \{\} q_5 \{a,\} q_6 \{a,\} q_7 \emptyset
  a /          \       /          \       /          \       /          \ /          \ /          \\
 q_2 \{a,\} q_3 \{a,a,\} q_4 \{\} q_5 \{a,\} q_6 \{a,\} q_7 \emptyset
  b /          \       /          \       /          \       /          \ /          \ /          \\
 q_3 \{a,\} q_5 \{a,a,\} q_6 \{a,\} q_7 \emptyset
  c /          \       /          \       /          \       /          \\
 q_5 \{a,a,\} q_6 \{a,\} q_7 \emptyset
  c /          \\
 q_6 \{\} q_7 \emptyset
```
(b) (5 pts) Show that the PDA $P_3$ is actually wrong by finding one string $s$ on which $P_3$ fails, i.e., $s \notin C$ but $P_3$ accepts $s$, or $s \in C$ but $P_3$ rejects $s$. Then draw a simulation tree like the one in (a).

(c) (10 pts) Construct a PDA $P_3'$ with $\leq 4$ states to recognize $C$. $P_3'$ is restricted to have only one accept state. Draw the state diagram, and simulate $P_3'$ on your string considered in (b).

Solution.

(a) See the following tree.

```
q_0 \emptyset

0

q_0 \{0\}

0

q_0 \{0,0\}

0

q_0 \{0,0,0\}

1

q_1 \{0,0\}

1

q_1 \{0\}

1

q_1 \emptyset

q_1 \{0\}

q_1 \{0\}

q_1 \{0\}

q_1 \{0,0\}
```

Note: We intend to check if you understand the non-deterministic setting. Thus if you drew a path instead of a tree, then you do not get any point.

(b) 001 $\notin C$ but $P_3$ accepts it. See the following tree.
A simpler example is that $0 \notin C$ but $P_3$ accepts it. See the following tree.

(c) Either

See the following trees that show the rejection of 001.

Either
or

See the following tree that shows the rejection of 0.

$q_0 \emptyset \rightarrow q_1 \{\$\}$

0

\[ q_1 \{0, \$\} \]

0

\[ q_1 \{0, 0, \$\} \rightarrow q_2 \{0, 0, \$\} \]

1

\[ q_2 \{0, \$\} \]

\[ q_2 \{0, 0, \$\} \]
Common mistakes:

- $\epsilon$ is rejected
- 1 is rejected

**Problem 4 (10 pts).** Consider the following CFG.

$$S \rightarrow A | 0A$$
$$A \rightarrow SAS | \epsilon$$

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 $S_0$

$$S_0 \rightarrow S$$
$$S \rightarrow A | 0A$$
$$A \rightarrow SAS | \epsilon$$

Remove $A \rightarrow \epsilon$

$$S_0 \rightarrow S$$
$$S \rightarrow A | 0A | \epsilon | 0$$
$$A \rightarrow SAS | SS$$

Remove $S \rightarrow \epsilon$

$$S_0 \rightarrow S | \epsilon$$
$$S \rightarrow A | 0A | 0$$
$$A \rightarrow SAS | SS | AS | SA | S$$

Remove $S \rightarrow A$

$$S_0 \rightarrow S | \epsilon$$
$$S \rightarrow 0A | 0 | SAS | SS | AS | SA | S$$
$$A \rightarrow SAS | SS | AS | SA | S$$

Remove $A \rightarrow S$

$$S_0 \rightarrow S | \epsilon$$
$$S \rightarrow 0A | 0 | SAS | SS | AS | SA$$
$$A \rightarrow SAS | SS | AS | SA | 0A | 0$$
Remove $S_0 \to S$

\[
S_0 \to 0A \mid 0 \mid SAS \mid SS \mid AS \mid SA \mid \epsilon \\
S \to 0A \mid 0 \mid SAS \mid SS \mid AS \mid SA \\
A \to 0A \mid 0 \mid SAS \mid SS \mid AS \mid SA \\
\]

Remove $SAS$

\[
S_0 \to 0A \mid 0 \mid SB \mid SS \mid AS \mid SA \mid \epsilon \\
S \to 0A \mid 0 \mid SB \mid SS \mid AS \mid SA \\
A \to 0A \mid 0 \mid SB \mid SS \mid AS \mid SA \\
B \to AS \\
\]

Add $U \to 0$

\[
S_0 \to UA \mid 0 \mid SB \mid SS \mid AS \mid SA \mid \epsilon \\
S \to UA \mid 0 \mid SB \mid SS \mid AS \mid SA \\
A \to UA \mid 0 \mid SB \mid SS \mid AS \mid SA \\
B \to AS \\
U \to 0 \\
\]

Common mistakes:

- In removing $A \to \epsilon$, many forgot to add the rule $A \to SS$.
- In removing $S \to \epsilon$, many forgot to add the rule $A \to S$.

**Problem 5 (35 pts).** For $w = w_1 \cdots w_n$, we denote its reverse by $w^R = w_n \cdots w_1$. Consider

\[ \Sigma = \{0, 1, \#\} \]

and the language

\[ D = \{w\#w^R \mid w \in \{0, 1\}^*\} . \]

For all following subproblems, the number of states includes $q_{\text{accept}}$ and $q_{\text{reject}}$; however, you do not need to show $q_{\text{reject}}$ and the transitions going to $q_{\text{reject}}$. You do not get any point if more states than the specified limit are used.

(a) (13 pts) Draw a TM diagram with $\leq 9$ states to decide $D$ by the outside-in strategy

Step 1: Replace the leftmost 0 or 1 in $w$ with $\sqcup$.

Step 2: Move rightwards until the first $\sqcup$ (skipping all non-$\sqcup$ symbols) in or after $w^R$. Move left and check whether the element matches the previously seen leftmost one. If so, replace it with $\sqcup$. Otherwise, reject.

Step 3: Move back to find the first non-$\sqcup$ in $w$. Go to Step 1.

Step 4: In Step 3, if no more 0/1 in the left, check whether the head is pointing to a $\#$ and there is also nothing in the right.
and simulate two strings 01#10 and 010#0 by this TM. You are required to have

\[ \Gamma = \{0, 1, \#, \square\}. \]

For the simulation, please show the sequence of configurations for each step, and write the result (accept or reject). **You must finish the simulation in order to get points.**

(b) (12 pts) Draw a TM diagram with \( \leq 8 \) states to decide \( D \) and follow the inside-out strategy:

Step 1: To know whether we are at the beginning of the tape, replace the first 0, 1 with \( $0, $1 \) respectively.

Step 2: Find the first \#.

Step 3: Find the first 0 or 1 after \# that has not been marked as \( \times \) (crossing with \( \times \)), and check whether it matches the symbol before \# that has not been marked as \( \times \) (also crossing with \( \times \)). If the two do not match, reject. Otherwise, repeat this step.

Step 4: In the end, if we find the matching \( $0/1 \) in the left (instead of matching 0/1), check whether there is nothing in the right except for \( \times \) and \#.

and simulate the same strings used in (a). You are required to have

\[ \Gamma = \{0, 1, \#, $0, $1, \times, \square\}. \]

Note that \( $0, $1, \times \) are special symbols that never appear in the input string. Please use this property to save the number of states. **You must finish the simulation in order to get points.**

(c) (10 pts) Draw a two-tape TM diagram with \( \leq 4 \) states to decide \( D \), and simulate the same strings used in (a). You are required to have

\[ \Gamma = \{0, 1, \#, \square\}. \]

**You must finish the simulation in order to get points.**

*Solution.*

(a) See the following diagram. For all the following diagrams, we assume that the head moves right in each of the transitions to \( q_{\text{reject}} \).
Simulation for 01#10 and 010#0:

- 01#10:

  \[
  q_1 01#10 \Rightarrow \sqcup q_2 1\#10 \Rightarrow \sqcup 1q_2 1\#10 \Rightarrow \sqcup 1\#q_2 10 \Rightarrow \sqcup 1\#1q_2 0 \\
  \Rightarrow \sqcup 1q_3\#1 \Rightarrow \sqcup q_3 1q_4 \Rightarrow \sqcup 1q_6 1\# \Rightarrow \sqcup q_6 1\#1 \\
  \Rightarrow \sqcup 1\#q_5 \Rightarrow \sqcup 1\#q_6 \Rightarrow \sqcup 1q_6 \# \Rightarrow \sqcup 1q_1 \Rightarrow \sqcup \#q_7 \Rightarrow \\
  \Rightarrow \sqcup q_{accept} \Rightarrow \text{Accept}
  \]

- 010#0:

  \[
  q_1 010\#0 \Rightarrow \sqcup q_2 10\#0 \Rightarrow \sqcup 1q_2 0\#0 \Rightarrow \sqcup 10q_2 \#0 \Rightarrow \sqcup 10\#q_2 0 \\
  \Rightarrow \sqcup 10q_3\#1 \Rightarrow \sqcup 10q_3 \# \Rightarrow \sqcup 1q_6 0\# \Rightarrow \sqcup q_6 10\# \\
  \Rightarrow \sqcup q_4 10\# \Rightarrow \sqcup 10q_4 \# \Rightarrow \sqcup 10q_4\# \Rightarrow \sqcup 100\# \Rightarrow \sqcup 10\#q_3 \Rightarrow \\
  \Rightarrow \sqcup 0q_5 \# \Rightarrow \sqcup \#q_{reject} \Rightarrow \text{Reject}
  \]

(b) See the following diagram.
Note that we do not worry about multiple #’s because of the # → R loop at q3. In the end, we use q1 → q6 to ensure that only a string with one # is accepted.

Simulation for 01#10 and 010#0:

- **01#10:**
  
  \[
  q_101\#10 \quad \Rightarrow \quad q_0q_21\#10 \quad \Rightarrow \quad q_01q_2\#10 \quad \Rightarrow \quad q_01\#310
  \]

- **010#0:**
  
  \[
  q_1010\#0 \quad \Rightarrow \quad q_0q_210\#0 \quad \Rightarrow \quad q_01q_20\#0 \quad \Rightarrow \quad q_010q_2\#0
  \]

Alternative solution:
Simulation for 01#10 and 010#0:

- 01#10:
  
  \[
  q_101#10 \Rightarrow \$$q_21#10 \Rightarrow \$$q_3#10 \Rightarrow \$$q_4#20 \Rightarrow \$$q_51#20 \Rightarrow \$$q_60\#20 \Rightarrow \$$q_{accept} \Rightarrow \text{Accept}
  \]

- 010#0:
  
  \[
  q_1010#0 \Rightarrow \$$q_210#0 \Rightarrow \$$q_3#0 \Rightarrow \$$q_4#0 \Rightarrow \$$q_5#0 \Rightarrow \$$q_60#0 \Rightarrow \$$q_{reject} \Rightarrow \text{Reject}
  \]

Another alternative solution (simulation omitted):
Note that we cannot have a link from $q_4/q_5$ to $q_1$, because $0\#01\#1$ will be accepted.

**Common mistakes:**

- Many wrongly accept $0\#0\#$.
- Many wrongly reject $\#$.

(c) See the following diagram.

```
Simulation for 01\#10 and 010\#0:

- 01\#10:
  \[
  \begin{align*}
  &\Rightarrow q_1 \begin{array}{c} 0 \# \# \end{array} \Rightarrow 0 q_1 \begin{array}{c} \# \# \end{array} \Rightarrow 0 q_1 \begin{array}{c} \# \# \end{array} \\
  &\Rightarrow 0 q_1 \begin{array}{c} \# \# \end{array} \\
  &\Rightarrow q_\text{accept} \begin{array}{c} \# \# \end{array} \Rightarrow \text{Accept}
  \end{align*}
  \]

- 010\#0:
  \[
  \begin{align*}
  &\Rightarrow q_1 \begin{array}{c} 0 0 \# \# \end{array} \Rightarrow 0 q_1 \begin{array}{c} 0 0 \# \# \end{array} \Rightarrow 0 q_1 \begin{array}{c} 0 0 \# \# \end{array} \\
  &\Rightarrow 0 q_1 \begin{array}{c} 0 0 \# \# \end{array} \\
  &\Rightarrow q_\text{reject} \begin{array}{c} 0 0 \# \# \end{array} \Rightarrow \text{Reject}
  \end{align*}
  \]