CS 420, Spring 2018
Homework 6 Solutions

1. Using the method from class, transform the following right regular grammar into an NFA

\[ S \rightarrow 0S|1S|0T \\
T \rightarrow 0T|1U \\
U \rightarrow 1 \]

**Solution:** We first replace the rule \( U \rightarrow 1 \) to obtain:

\[ S \rightarrow 0S|1S|0T \\
T \rightarrow 0T|1U \\
U \rightarrow 1Z \\
Z \rightarrow \varepsilon \]

We then obtain the following NFA:

\[ \]

2. Read Definition 2.8, Theorem 2.9 and Example 2.10 in the textbook (third edition) concerning Chomsky Normal Form and then put the following grammar into Chomsky Normal Form.

\[ S \rightarrow T|TaS \\
T \rightarrow aTa|TT |\varepsilon \]

**Step 1:** Introduce new start symbol
\[ S_0 \to S \\
S \to T|TaS \\
T \to aTb|bTa|TT|\varepsilon \]

**Step 2:** Eliminate \( \varepsilon \)-rules.
Eliminate \( T \to \varepsilon \):

\[
\begin{align*}
S_0 & \to S \\
S & \to T|TaS|aS|\varepsilon \\
T & \to aTb|bTa|TT|ab|ba|T
\end{align*}
\]

Eliminate \( S \to \varepsilon \):

\[
\begin{align*}
S_0 & \to S|\varepsilon \\
S & \to T|TaS|aS|Ta|a \\
T & \to aTb|bTa|TT|ab|ba|T
\end{align*}
\]

**Step 3:** Eliminate unit rules
Eliminate \( S_0 \to S \):

\[
\begin{align*}
S_0 & \to \varepsilon|T|TaS|aS|Ta|a \\
S & \to T|TaS|aS|Ta|a \\
T & \to aTb|bTa|TT|ab|ba|T
\end{align*}
\]

Eliminate \( S_0 \to T \):

\[
\begin{align*}
S_0 & \to \varepsilon|T|TaS|aS|Ta|a|aTb|bTa|TT|ab|ba \\
S & \to T|TaS|aS|Ta|a \\
T & \to aTb|bTa|TT|ab|ba|T
\end{align*}
\]

[Note that we did not add back in \( S_0 \to T \) even though we have the rule \( T \to T \), because we do not add back in a rule after (or when) we eliminate it.]

Eliminate \( S \to T \):

\[
\begin{align*}
S_0 & \to \varepsilon|T|TaS|aS|Ta|a|aTb|bTa|TT|ab|ba \\
S & \to TaS|aS|Ta|a|aTb|bTa|TT|ab|ba \\
T & \to aTb|bTa|TT|ab|ba|T
\end{align*}
\]

Eliminate \( T \to T \):

\[
\begin{align*}
S_0 & \to \varepsilon|T|TaS|aS|Ta|a|aTb|bTa|TT|ab|ba \\
S & \to TaS|aS|Ta|a|aTb|bTa|TT|ab|ba \\
T & \to aTb|bTa|TT|ab|ba
\end{align*}
\]
Step 4: Eliminate long rules

Step 5: Eliminate terminals in wrong place

3. Problem 2.26 Show that if $G$ is a CFG in Chomsky normal form, then for any string $w \in L(G)$ of length $n \geq 1$, exactly $2n - 1$ steps are required for any derivation of $w$.

Proof: First note that in rules of the form $A \rightarrow BC$, neither $A$ nor $B$ can be $S$. This means that in a derivation of a string $w$ of length greater than 0, the rule $S \rightarrow \varepsilon$, if it exists in $G$, cannot be used, so the derivation can use only rules of the forms $A \rightarrow BC$ and $A \rightarrow a$. Each application of a rule of the first form lengthens the string derived by one symbol. Since the $A \rightarrow a$ rules leave the length of the string the same, this means that $A \rightarrow BC$ rules are used exactly $n - 1$ times in deriving a string $w$ of length $n$. Since each use of a rule $A \rightarrow a$ introduces one terminal and this terminal can never be removed, rules of the form $A \rightarrow a$ are used exactly $n$ times in the derivation of string $w$ consisting of $n$ terminals. In total, the derivation of $w$ has length $n - 1 + n = 2n - 1$.

We can also give a proof by induction. First note that if $A$ is a variable of $G$ different from $S$, then it is impossible to derive $\varepsilon$ from $A$. We show the following more general statement: if $A$ is a variable and $w$ is a string of terminals of length $n \geq 1$, then any derivation of $w$ from $A$ in $G$ takes $2n - 1$ steps. The proof is by induction on $n$. If $n = 1$, then $w$ is a single terminal. If the derivation of $w$ from $A$ began with the rule $A \rightarrow BC$, then, since neither $B$ nor $C$ is $S$, $w$ would have length at least 2. Thus, the derivation must consist of a single application of the rule $A \rightarrow w$ and so has length 1. Since $2 \times 1 - 1 = 1$, the desired relationship holds in this case. Now, suppose that $n > 1$ and the result holds for strings of length $j$ with $1 \leq j \leq n - 1$. If $w$ is a string of terminals of length $n$ that can be
derived from $A$, then the derivation must begin with a rule $A \rightarrow BC$. Let $x = yz$ where $y$ is the part of $x$ derived from $B$ and $z$ is the part derived from $C$. Since neither $B$ nor $C$ is $S$, neither $y$ nor $z$ is $\varepsilon$. Thus, we can apply the inductive hypothesis to both $y$ and $z$. This means that if $|y| = k$ and $|z| = n - k$, then the derivation of $y$ from $B$ takes $2k - 1$ steps and the derivation of $z$ from $C$ takes $2(n - k) - 1$ steps. The total length of the derivation of $w$ from $A$ is then $1 + (2k - 1) + (2(n - k) - 1) = 2n - 1$, which completes the induction.

4. Give PDAs that recognize the following languages:

   (a) $\{0^{2n}1^n | n \geq 0\}$:

   (b) $\{0^n1^m | n \geq 2m\}$:

   (c) $\{w\#u | w, u \in \{0, 1\}^* \text{ and } |w| < |u|\}$:
5. Let $G$ be the grammar

$$
S \rightarrow T[VaT]VaS \\
T \rightarrow \varepsilon[aUbT]bVaT \\
U \rightarrow \varepsilon[aUbU] \\
V \rightarrow \varepsilon[bVaV]
$$

(a) Using the method from class, give a PDA $M$ with $L(M) = L(G)$. 

(d) $\{x_1\#x_2\#x_3|x_1, x_2, x_3 \in \{a, b\}^* \text{ and } x_2 = x_3^R\}$. 

5.
(b) Show an accepting computation for $M$ on the string $babaaba$ by giving a chart with the state, tape contents, and stack contents after each step.

<table>
<thead>
<tr>
<th>State</th>
<th>Tape</th>
<th>Stack</th>
</tr>
</thead>
<tbody>
<tr>
<td>$q_{start}$</td>
<td>babaaba</td>
<td>$\varepsilon$</td>
</tr>
<tr>
<td>$q_{loop}$</td>
<td>babaaba</td>
<td>$S$</td>
</tr>
<tr>
<td>$q_{loop}$</td>
<td>babaaba</td>
<td>$VaT$</td>
</tr>
<tr>
<td>$q_{loop}$</td>
<td>babaaba</td>
<td>$bVaT$</td>
</tr>
<tr>
<td>$q_{loop}$</td>
<td>abaaba</td>
<td>$VaVaT$</td>
</tr>
<tr>
<td>$q_{loop}$</td>
<td>abaaba</td>
<td>$aVaT$</td>
</tr>
<tr>
<td>$q_{loop}$</td>
<td>baaba</td>
<td>$Va$</td>
</tr>
<tr>
<td>$q_{loop}$</td>
<td>baaba</td>
<td>$bVaVaT$</td>
</tr>
<tr>
<td>$q_{loop}$</td>
<td>aaba</td>
<td>$VaVaT$</td>
</tr>
<tr>
<td>$q_{loop}$</td>
<td>aaba</td>
<td>$aVaT$</td>
</tr>
<tr>
<td>$q_{loop}$</td>
<td>aba</td>
<td>$VaT$</td>
</tr>
<tr>
<td>$q_{loop}$</td>
<td>aba</td>
<td>$aT$</td>
</tr>
<tr>
<td>$q_{loop}$</td>
<td>ba</td>
<td>$T$</td>
</tr>
<tr>
<td>$q_{loop}$</td>
<td>ba</td>
<td>$bVaT$</td>
</tr>
<tr>
<td>$q_{loop}$</td>
<td>a</td>
<td>$VaT$</td>
</tr>
<tr>
<td>$q_{loop}$</td>
<td>a</td>
<td>$aT$</td>
</tr>
<tr>
<td>$q_{loop}$</td>
<td>$\varepsilon$</td>
<td>$T$</td>
</tr>
<tr>
<td>$q_{loop}$</td>
<td>$\varepsilon$</td>
<td>$$</td>
</tr>
<tr>
<td>$q_{accept}$</td>
<td>$\varepsilon$</td>
<td>$\varepsilon$</td>
</tr>
</tbody>
</table>