Solutions for CSE303 Homework 5

1. Construct nondeterministic pushdown automata (npda) that accept the following regular languages. Note: Observe that all the languages are regular languages, so the solutions are essentially NFA’s (or npda’s with inactive stack). For all the languages, \( f \) is the final state.

(a) \( L_1 = L(\text{aaa}^*b) \)

**Solution:** The npda \( M = (K, \Sigma, \Gamma, \Delta, s, F) \), where \( K = \{ q_0, q_1, q_2, f \} \), \( \Sigma = \{ a, b \} \), \( F = \{ f \} \), and \( \Delta \) as the set of following rules accepts \( L_1 \).

\[
((q_0, a, c), (q_1, c)) \\
((q_1, a, c), (q_2, c)) \\
((q_2, a, c), (q_2, c)) \\
((q_2, b, c), (f, c))
\]

(b) \( L_2 = L(abb^*aba^*) \)

**Solution:** The npda \( M = (K, \Sigma, \Gamma, \Delta, s, F) \), where \( K = \{ r_0, r_1, r_2, r_3, f \} \), \( \Sigma = \{ a, b \} \), \( F = \{ f \} \), and \( \Delta \) as the set of following rules accepts \( L_2 \).

\[
((r_0, a, c), (r_1, e)) \\
((r_1, b, c), (r_2, e)) \\
((r_2, b, c), (r_2, e)) \\
((r_2, a, c), (r_3, e)) \\
((r_3, b, c), (f, e)) \\
((f, a, c), (f, e))
\]

(c) the union of \( L_1 \) and \( L_2 \)

**Solution:** The npda \( M = (K, \Sigma, \Gamma, \Delta, s, F) \), where \( K = \{ q_0, q_1, q_2, r_0, r_1, r_2, r_3, f \} \), \( \Sigma = \{ a, b \} \), \( F = \{ f \} \), and \( \Delta \) as the set of following rules accepts \( L_1 \cup L_2 \).

\[
((s, e, e), (q_0, e)) \\
((s, e, e), (r_0, e))
\]

where \( q_0 \) and \( r_0 \) are as defined above.

2. Construct npda’s that accept the following languages on \( \Sigma = \{ a, b, c \} \).

(a) \( L = \{ a^n b^{2n} : n \geq 0 \} \)

**Solution:** The npda \( M = (K, \Sigma, \Gamma, \Delta, s, F) \), where \( K = \{ s, f \} \), \( \Sigma = \{ a, b \} \), \( F = \{ f \} \), and \( \Delta \) as the set of following rules accepts \( L \).

\[
((s, c, e), (f, e)) \\
((s, a, c), (s, aa)) \\
((s, b, a), (f, e)) \\
((f, b, a), (f, e))
\]

(b) \( L = \{ a^n b^m c^{2n+m} : n \geq 0, m \geq 0 \} \)

**Solution:** The npda \( M = (K, \Sigma, \Gamma, \Delta, s, F) \), where \( K = \{ s_0, s_1, f \} \), \( \Sigma = \{ a, b \} \), \( F = \{ s_1 \} \), and \( \Delta \) as the set of following rules accepts \( L \).

\[
((s_0, c, e), (f, e)) \\
((s_0, a, c), (s_0, a)) \\
((s_0, c, a), (f, e)) \\
((s_0, b, e), (s_1, b)) \\
((s_1, b, e), (s_1, b)) \\
((s_1, c, b), (f, e)) \\
((f, c, b), (f, e)) \\
((f, c, a), (f, e))
\]
3. Find an npda on $\Sigma = \{a, b, c\}$ that accepts the language $L = \{w_1cw_2 : w_1, w_2 \in \{a, b\}^*, w_1 \neq w_2^R\}$

**Solution:** Idea: Observe that if $w_1 = w_2^R$, then the npda should not accept the input. But $w_1 = w_2^R$ is equivalent to $w_1^R = w_2$. Hence, we first, push $w_1$ onto stack, then while matching $w_2$, if a mismatch is found (then $w_1^R \neq w_2$), consume all the input and empty the stack then go to final state; otherwise (then $w_1^R = w_2$ which means $w_1 = w_2^R$), the input is not accepted. Thus the npda $M = (K, \Sigma, \Gamma, \Delta, s, F)$, where $K = \{s_0, s_1, d, f\}$, $\Sigma = \{a, b\}$, $F = \{f\}$, and $\Delta$ as the set of following rules accepts $L$.

\[
\begin{align*}
(s_0, a, e, (s_0, a)) \\
(s_0, b, e, (s_0, b)) \\
(s_1, a, a, (s_1, c)) \\
(s_1, b, b, (s_1, c)) \\
(d, e, a, (d, e)) \\
(d, e, b, (d, e)) \\
(d, a, c, (d, e)) \\
(d, b, c, (d, e)) \\
(d, e, c, (f, e))
\end{align*}
\]

4. Construct an npda corresponding to the grammar

\[
S \rightarrow aABB \mid aAA,
A \rightarrow aBB \mid a,
B \rightarrow bBB \mid A
\]

**Solution:** The npda $M = (K, \Sigma, \Gamma, \Delta, s, F)$, where $K = \{s_0, s_1\}$, $\Sigma = \{a, b\}$, $F = \{s_1\}$, and $\Delta$ as the set of following rules implements the grammar above.

\[
\begin{align*}
(s_0, c, e, (s_1, S)) \\
(s_1, a, S, (s_1, ABB)) \\
(s_1, a, S, (s_1, AA)) \\
(s_1, a, A, (s_1, BB)) \\
(s_1, a, A, (s_1, c)) \\
(s_1, b, B, (s_1, BB)) \\
(s_1, a, B, (s_1, BB)) \\
(s_1, a, B, (s_1, S))
\end{align*}
\]

Another solution:

\[
\begin{align*}
(s_0, c, e, (s_1, S)) \\
(s_1, e, S, (s_1, aABB)) \\
(s_1, c, S, (s_1, aAA)) \\
(s_1, c, A, (s_1, aBB)) \\
(s_1, e, B, (s_1, BB)) \\
(s_1, e, B, (s_1, A)) \\
(s_1, a, a, (s_1, c)) \\
(s_1, b, b, (s_1, c))
\end{align*}
\]

5. Show that the language $L = \{ww : w \in \{a, b\}^*\}$ is not context-free.

**Solution:** Consider the string $a^mb^ma^mb^m$. Now, let $v$ and $y$ contain only the first $a$’s and let $v = a^p$ and $y = a^q$. Then consider $uxz = a^kb^ma^mb^m$, $k < m$, which is not in $L$. For other choices of $v$ and $y$, we can make similar arguments (since each symbol occurs exactly $m$ times). Thus, $L$ is not context-free.
6. Show that the language \( L = \{a^n : n \geq 0\} \) is not context-free.

**Solution:** In this case, this is same as showing that \( L \) is not regular (since the language consists entirely of alphabet over single symbol). Let \( w = uxyz \) such that \( w = \epsilon \). Then we need to show that \( w = xz'z, \) for \( i = 0, 1, 2, \ldots \). Let \( w = a^n \). Let \( |y| = k \leq m \). Then \( xz \) has length \( m! - k \). This string is in \( L \) only if there exists a \( j \) such that \( m! - k = j! \). But this is impossible, since for \( m > 2 \) and \( k \leq m \), we have \( m! - k > (m - 1)! \). Therefore \( L \) is not context-free (or regular, either).

7. Construct Turing machines that will accept the following languages on \( \{a, b\} \).

(a) \( L = \{w : |w| \text{ is even}\} \)

**Solution:** Here we keep checking off two input symbols and if ultimately we encounter end of input in the start state, we accept the input.

The Turing machine \( M = (K, \Sigma, \delta, s, H) \), where \( K = \{q_0, q_1, h\} \), \( \Sigma = \{a, b, \texttt{⊔}, \texttt{⊥}\} \), \( s = q_0 \), \( H = \{h\} \) and \( \delta \) as follows accepts \( L \).

<table>
<thead>
<tr>
<th>( q )</th>
<th>( \sigma )</th>
<th>( \delta(q, \sigma) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( q_0 )</td>
<td>( a )</td>
<td>( (q_1, \rightarrow) )</td>
</tr>
<tr>
<td>( q_0 )</td>
<td>( b )</td>
<td>( (q_1, \rightarrow) )</td>
</tr>
<tr>
<td>( q_1 )</td>
<td>( a )</td>
<td>( (q_0, \rightarrow) )</td>
</tr>
<tr>
<td>( q_1 )</td>
<td>( b )</td>
<td>( (q_0, \rightarrow) )</td>
</tr>
<tr>
<td>( q_0 )</td>
<td>( \texttt{⊔} )</td>
<td>( (h, \texttt{⊥}) )</td>
</tr>
</tbody>
</table>

(b) \( L = \{w : n_a(w) = n_b(w)\} \)

where \( n_a(w) \) is number of \( a \)'s in \( w \) and \( n_b(w) \) is number of \( b \)'s in \( w \).

**Solution:** First find leftmost \( a \) or \( b \). If \( a \) (or \( b \)) is found, then replace it with \( x \) and find a matching \( b \) (or \( a \)), replace it with \( x \) and then go back to the left end of the tape (so as to find next leftmost \( a \) or \( b \)). Repeat.

The Turing machine \( M = (K, \Sigma, \delta, s, H) \), where \( K = \{q_0, q_1, q_2, q_3, h\} \), \( \Sigma = \{a, b, \texttt{⊔}, \texttt{⊥}\} \), \( s = q_0 \), \( H = \{h\} \) and \( \delta \) as given below accepts \( L \).

<table>
<thead>
<tr>
<th>( q )</th>
<th>( \sigma )</th>
<th>( \delta(q, \sigma) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( q_0 )</td>
<td>( x )</td>
<td>( (q_0, \rightarrow) )</td>
</tr>
<tr>
<td>( q_0 )</td>
<td>( a )</td>
<td>( (q_1, x) )</td>
</tr>
<tr>
<td>( q_0 )</td>
<td>( b )</td>
<td>( (q_2, x) )</td>
</tr>
<tr>
<td>( q_1 )</td>
<td>( x )</td>
<td>( (q_1, \rightarrow) )</td>
</tr>
<tr>
<td>( q_1 )</td>
<td>( a )</td>
<td>( (q_1, \rightarrow) )</td>
</tr>
<tr>
<td>( q_1 )</td>
<td>( b )</td>
<td>( (q_3, x) )</td>
</tr>
<tr>
<td>( q_2 )</td>
<td>( x )</td>
<td>( (q_2, \rightarrow) )</td>
</tr>
<tr>
<td>( q_2 )</td>
<td>( a )</td>
<td>( (q_3, x) )</td>
</tr>
<tr>
<td>( q_2 )</td>
<td>( b )</td>
<td>( (q_3, \rightarrow) )</td>
</tr>
<tr>
<td>( q_3 )</td>
<td>( x )</td>
<td>( (q_3, \leftarrow) )</td>
</tr>
<tr>
<td>( q_3 )</td>
<td>( a )</td>
<td>( (q_3, \leftarrow) )</td>
</tr>
<tr>
<td>( q_3 )</td>
<td>( b )</td>
<td>( (q_3, \leftarrow) )</td>
</tr>
<tr>
<td>( q_3 )</td>
<td>( \texttt{⊔} )</td>
<td>( (q_0, \rightarrow) )</td>
</tr>
<tr>
<td>( q_0 )</td>
<td>( \texttt{⊥} )</td>
<td>( (h, \texttt{⊥}) )</td>
</tr>
</tbody>
</table>