Advanced Topics in Theoretical Computer Science — **Problem Set 5**, **Due December 6, 2012**

Problem 5.1. (10 points) Consider the following CFG G.

$$
\begin{aligned}
&S \to A \mid B\\
&A \to 0\,B\,1 \mid 1 \mid \varepsilon\\
&B \to 1\,A\,0
\end{aligned}
$$

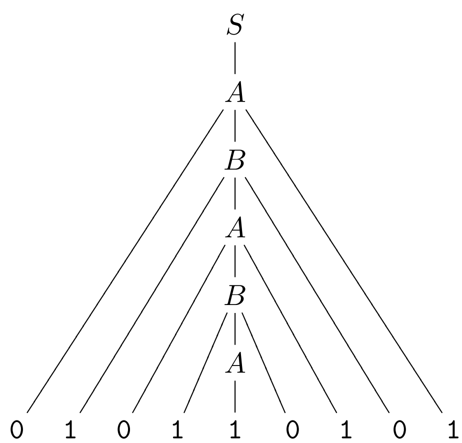
a. Give a derivation tree for 010110101.

b. Convert G into an equivalent CFG in Chomsky normal form.

c. Give a PDA that recognizes the same language as G.

Solution:

a.



b.

$$
\begin{aligned}
&S \to A \mid B\\
&A \to 0\,B\,1 \mid 1 \mid \varepsilon\\
&B \to 1\,A\,0
\end{aligned}
$$

Remove $A \to \varepsilon$ rule.

$$
\begin{aligned}
&S \to A \mid B \mid \varepsilon\\
&A \to 0\,B\,1 \mid 1\\
&B \to 1\,A\,0 \mid 1\,0
\end{aligned}
$$

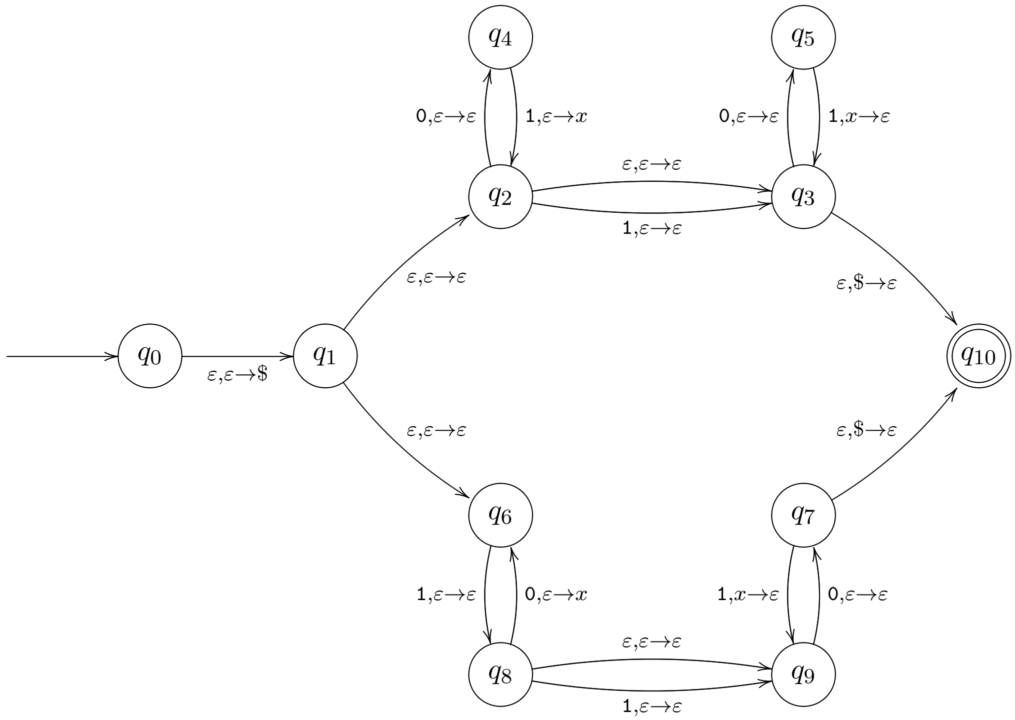
Remove $S \to A$ and $S \to B$ and add new nonterminals.

$$
\begin{aligned}
&S \to 0\,B\,1 \mid 1 \mid 1\,A\,0 \mid 1\,0 \mid \varepsilon\\
&A \to 0\,B\,1 \mid 1\\
&B \to 1\,A\,0 \mid 1\,0\\
&Z \to 0\\
&O \to 1\\
&C \to Z\,B\\
&D \to O\,A
\end{aligned}
$$

Last step:

$S \to C\,O \mid \mathtt{1} \mid D\,Z \mid O\,Z \mid \varepsilon$
$A \to C\,O \mid \mathtt{1}$
$B \to D\,Z \mid O\,Z$
$Z \to \mathtt{0}$
$O \to \mathtt{1}$
$C \to Z\,B$
$D \to O\,A$

c. First we do a nondeterministic choice whether the word comes from nonterminal A or B. If it comes from nonterminal A, the word is of the kind $(\mathtt{01})^n\mathtt{1}(\mathtt{01})^n$ or $(\mathtt{01})^n(\mathtt{01})^n$ with $n \geq 0$. If it comes from nonterminal B, the word is of the kind $\mathtt{1}(\mathtt{01})^n\mathtt{1}(\mathtt{01})^n\mathtt{0}$ or $\mathtt{1}(\mathtt{01})^n(\mathtt{01})^n\mathtt{0}$ with $n \geq 0$.



Problem 5.2. (20 points) Which of the three languages L_i over the alphabet $\{\mathtt{0},\mathtt{1},\mathtt{2}\}$ is context-free? For each L_i, either give a PDA that recognizes L_i or a CFG that generates L_i, or use the pumping lemma for the context-free languages to prove that L_i is not context-free.

a. $L_1 = \{\mathtt{0}^i\mathtt{1}^j\mathtt{2}^k \mid j = 2 \cdot i + k\}$.

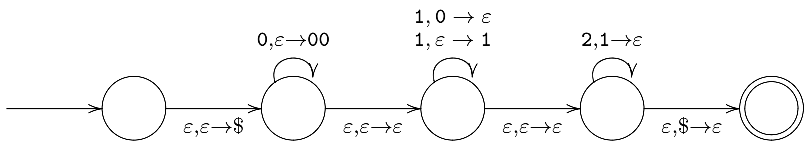
b. $L_2 = \{\mathtt{0}^i\mathtt{1}^j\mathtt{2}^k \mid i < j < k\}$.

c. L_3 is the complement of L_2.

Solution:

a. This language is context-free. A PDA for the language works as follows. Every time it sees a zero, it pushes two symbols 0 onto the stack. Then every time it sees a one, it either pops the symbol 0 from the stack, or pushes the symbol 1 onto the stack. Then every time it sees a two, it pops the symbol 1 from the stack. Finally, it accepts on empty stack.

This PDA is represented below:



A CFG for the language is:

$$\begin{aligned} S &\to\ A\,B \\ A &\to\ \varepsilon \mid 0\,A\,1\,1 \\ B &\to\ \varepsilon \mid 1\,B\,2 \end{aligned}$$

c. We will use the pumping lemma to prove by contradiction that the language $L_2 = \{0^i\ 1^j\ 2^k \mid i < j < k\}$ is not context-free.

For an arbitrary pumping length p we choose the word $w = 0^p 1^{p+1} 2^{p+2} \in L_2$. We consider any arbitrary decomposition of the word $w = xyzuv$ such that $|yu| > 0$ and $|yzu| \leq p$.

Case 1: If y or u consists of two different symbols (0 and 1 or 1 and 2), then pumping once (i.e., choosing $i = 2$) we will have that either the sequence 01 or the sequence 12 appears twice in the word and hence the word will not be in L_2. Also note that since $|yzu| \leq p$ it cannot be the case that yzu contains both 0's and 2's.

Case 2: If yzu contains 0's then yu does not contain any 2's because $|yzu| \leq p$. Hence pumping twice (i.e., choosing $i = 3$), we increase the number of 0 by at least two but keep the number of 2's constant and hence $xy^3zu^3v \notin L_2$.

Case 3: If yu only consists of 1's then we pump once (i.e., we choose $i = 2$) and we increase the number of 1's and keep the number of 2's constant and hence the resulting word is not in the language L_2. In the case of yu only consisting of 2's, we can pump down ($i = 0$) and the same argument can be used because the number of 2's decreases.

Case 4: If y consists of 1's only and u consists of 2's only, we choose $i = 0$: if $|y| > 0$, we will decrease the number of 1's but the number of 0's is constant and hence xzv is not in L_2; on the other hand, if $y = \varepsilon$, then we decrease the number of 2's (it must be the case that $|u| > 0$), but the number of 1's is constant and again we can conclude that xzv is not in L_2.

Therefore, for any decomposition $w = xyzuv$, we have found an i such that $xy^izu^iv \notin L_2$. This means that L_2 is not context-free because it does not have the pumping property.

c. This language is context-free.

We first note that a word w in L_3 has three properties:

1. All 0's precede the first 1 and all 1's precede the first 2.
2. $w \in \{0^i\ 1^j\ 2^* \mid i < j\}$
3. $w \in \{0^*\ 1^j\ 2^k \mid j < k\}$

Therefore a word is in L_3 if it does not have one or more of these properties (De Morgan law).

We build the grammar in three parts:

- S_1 represents the words for which the first property doesn't hold (i.e., a 1 precedes a 0 or a 2 precedes a 0 or a 1).

– S_2 represents the words of the form $\texttt{0}^*\texttt{1}^*\texttt{2}^*$ which contain at least as many 0's as frm[o]–'s.

– S_3 represents the words of the form $\texttt{0}^*\texttt{1}^*\texttt{2}^*$ which contain at least as many 1's as 2's.

We use E_{01} and E_{12} to represents words of the form $\texttt{0}^n\texttt{1}^n$, respectively $\texttt{1}^n\texttt{2}^n$. M represents the words of the form $(\texttt{0} \cup \texttt{1} \cup \texttt{2})^*$ and M_0, M_1, M_2 respectively $\texttt{0}^*$, $\texttt{1}^*$, $\texttt{2}^*$.

$$S \to S_1 \mid S_2 \mid S_3$$

$$S_1 \to M\,\texttt{1}\,\texttt{0}\,M \mid M\,\texttt{2}\,\texttt{1}\,M \mid M\,\texttt{2}\,\texttt{0}\,M$$
$$S_2 \to M_0\,E_{01}\,M_2$$
$$S_3 \to M_0\,M_1\,E_{12}$$

$$E_{01} \to \texttt{0}\,E_{01}\,\texttt{1} \mid \varepsilon$$
$$E_{12} \to \texttt{1}\,E_{12}\,\texttt{2} \mid \varepsilon$$

$$M \to \texttt{0}\,M \mid \texttt{1}\,M \mid \texttt{2}\,M \mid \varepsilon$$
$$M_0 \to \texttt{0}\,M_0 \mid \varepsilon$$
$$M_1 \to \texttt{1}\,M_1 \mid \varepsilon$$
$$M_2 \to \texttt{2}\,M_2 \mid \varepsilon$$

The corresponding PDA can also be separated into the same three parts (each represented on one row below). Note that the first part (which corresponds to S_1) does not use the stack.