Theory of Computation (COMP-3540)
Sample Midterm 2 With Solution

Qn.1 Show that the following context-free grammar is ambiguous.

$$S \rightarrow ABA|aBaa$$
$$A \rightarrow a|aA$$
$$B \rightarrow b$$

by finding a string that has two leftmost or rightmost derivations.

[8 marks]

Ans: The string $abaa$ has the following two leftmost derivations.

1. $S \Rightarrow aBaa \Rightarrow abaa$
2. $S \Rightarrow ABA \Rightarrow aBA \Rightarrow abA \Rightarrow abaA \Rightarrow abaa$

Thus the grammar G is ambiguous.

Qn.2 Find a context-free grammar that generates the language accepted by the DPDA $P = (\{q_0, q_1\}, \{0, 1, 2\}, \{0, 1, 2, Z_0\}, \delta, q_0, Z_0\}$ whose transition function is given below:

$$\begin{gathered}
\delta(q_0, 0, Z_0) = \{(q_0, 0Z_0)\} \\
\delta(q_0, 0, 0) = \{(q_0, 00)\} \\
\delta(q_0, 1, Z_0) = \{(q_1, Z_0)\} \\
\delta(q_0, 1, 0) = \{(q_1, 0)\} \\
\delta(q_1, 2, 0) = \{(q_1, \epsilon)\} \\
\delta(q_1, \epsilon, Z_0) = \{(q_1, \epsilon)\}
\end{gathered}$$

It is enough to provide templates for the productions corresponding to the first two transitions.

[8 marks]

Ans:

We define all the components of the equivalent grammar $G = (V, T, P, S)$.

$V = S \cup \{[pXq] \mid p, q \in \{q_0, q_1\}, X \in \{0, 1, 2, Z_0\}\}$
$T = \{0, 1, 2\}$

The set of productions P include all the productions:

$S \rightarrow [q_0 Z_0 p]$, for each $p \in \{q_0, q_1\}$.

In addition, we have the following productions, in template form, for each one of the $\delta()$-moves of the given PDA.

$[q_0 Z_0 ?] \leftarrow 0[q_0 0 !][! Z_0 ?]$

$[q_0 0 ?] \leftarrow 0[q_0 0 !][! 0 ?]$

$[q_0 Z_0 ?] \leftarrow 1[q_1 Z_0 ?]$

$[q_0 0 ?] \leftarrow 1[q_1 0 ?]$

$[q_1 0 q_1] \leftarrow 2$

$[q_1 Z_0 q_1] \leftarrow \epsilon,$

where $?, ! \in \{q_0, q_1\}$

Qn.3 Design a context-free grammar that generates the language $L = \{a^n b^m c^{n+m} | n \geq 0, m \geq 0\}$. You must provide explanations for the productions of your grammar.

Hint: We can rewrite the string $a^n b^m c^{n+m}$ as $a^n b^m c^m c^n$, which shows that the outer (a, c)-pairs and the inner (b, c)-pairs have to be generated by different variables.

[8 marks]

Ans:

Following the hint, we have the productions:

$S \to aSc \mid B$

$B \to bBc|\epsilon$,

where the start variable S is used to generate the outer (a, c) pairs and the variable B is used to generate the inner (b, c) pairs.

For illustration, let us derive the string $a^2b^2c^4$ in the grammar. We have:

$S \Rightarrow aSc \Rightarrow aaScc \Rightarrow aaBcc \Rightarrow aabBccc \Rightarrow aabbBcccc \Rightarrow aabbcccc$

Qn.4 Use the pumping lemma (for context-free languages) to show that the language $L = \{a^i b^{2i} c^{3i} | i \geq 0\}$ over $\Sigma = \{a, b, c\}$ is not context-free. (**Hint**: You may choose the string $a^n b^{2n} c^{3n}$ in L to which to apply the PL). Clearly mention the i that you choose to show that $z' = uv^i wx^i y$ is not in L in each of the different cases that arises.

[8 marks]

Ans: Assume that L is a cfl, and let n be the PL constant for the given cfl L.

Set $z = a^n b^{2n} c^{3n}$. Clearly, $z \in L$ and as $|z| = 6n > n$, the string is long enough, satisfying all constraints on the choice of a string z.

Let $uvwxy$ be an adversarial decomposition of z, where $|vwx| \leq n$ and $|vx| > 0$ (or what is the same thing $vx \neq \epsilon$, or $|v| + |x| \neq 0$).

Several cases arise, dependong on the position of the string vwx within the chosen string z.

Case 1: vwx consists of a's (b's or c's) alone. It cannot consist of a's and c's since it will have to include all the b's in it, violating the constraint $|vwx| \leq n$.

According to the PL, for all $i \geq 0$, the strings $uv^i wx^i y$ are in L. Let us choose $i = 0$ and show that for this choice the resulting string $z' = uwy$ is not in L.

Since vwx is a suvstring of a^n, the string $z' = a^{n-|v|-|x|} b^{2n} c^{3n}$, we get fewer a's, disturbing the $1:2:3$ ratio of the a's, b's and c's. Hence $w' \notin L$, violating the conclusion of the PL.

Thus L is not a cfl in this case. A similar argument applies if the substring vwx spans only the b's or only the c's.

Case 2: vwx consists of a's and b's (b's and c's). That is, vwx is a substring of $a^n b^{2n}$.

Choosing $i = 0$, we get the substring$z' = uwy$, which has fewer a's or b's than the string z. It is possible that the ratio of the number of a's to the b's is still $1:2$ but their ratios with respect to the c, namely 1:3 and 2:3, are not both preserved.

Hence in this case too $z' \notin L$, violating the conclusion of the PL.

A similar argument applies to the case when the substring vwx spans only the b's and c's, allowing us to conclude that L is not a cfl.

The above cases are exhaustive and shows that L is not a cfl.

Qn.5 Use the CYK algorithm to determine whether the string $w = ababa$ is in the language generated by the following context-free grammar in CNF.

$$S \to AB$$
$$A \to BB|a$$
$$B \to AB|b$$

[8 marks]

Ans:

$S_{11} = \{A\}$, $S_{22} = \{B\}$, $S_{33} = \{A\}$, $S_{44} = \{B\}$, $S_{55} = \{A\}$

$S_{12} = \{S, B\}$, $S_{23} = \{\}$, $S_{34} = \{S, B\}$, $S_{45} = \{\}$

$S_{13} = \{\}$, $S_{24} = \{A\}$, $S_{35} = \{\}$

$S_{14} = \{A\}$, $S_{25} = \{\}$

$S_{15} = \{\}$

Since $S_{15} = \{\}$, and thus does not contain the start variable S, the string *ababa* is not in $L(G)$.

We work out the details for computing the set S_{14}, which is initially empty. This set should be populated by variables that can generate the substring of *ababa*, stretching from index position 1 to index position 4. That is, the substring *abab*.

Since this substring can be split into a prefix string and a suffix string in three ways, namely, *a* and *bab*, *ab* and *ab*, *aba* and *b*, we have to find all concatenations of two variables in the grammnar such that the first generates the prefix string and the second the suffix for each of the three possible ways we can break up *abab* into a prefix string and a suffix string.

Consider, for example, the prefix-suffix decomposition *a* and *bab*. Variables that can generate the prefix string *a* are to be found in the set S_{11}, while variables that can generate the suffix string *bab* are to be found in the set S_{24}.

All such combinations are to be found by concatenating the sets S_{11} and S_{24}, which have already been computed and saved.

Thus, $S_{11}S_{24} = \{A\}\{A\} = \{AA\}$. However, the grammar has no production whose right-hand side is AA and no variable gets added to the set S_{14}.

We next compute the concatenation of the sets S_{12} and S_{34} corresponding to the prefix-suffix decomposition ab and ab.

We have $S_{12}S_{34} = \{S, B\}\{S, B\} = \{SS, SB, BS, BB\}$. While SS, SB, BS are not the right-hand side of any productions of the grammar G, there is a production whose right-hand side is BB, namely $A \to BB$. Thus we add the right-hand side of the production A, to S_{14} so that $S_{14} = \{A\}$ now.

Continuing on, we consider the final prefix-suffix decomposition aba and b and concatenate the pre-computed sets S_{13} and S_{44}. We have $S_{13}S_{44} = \{\}\{B\} = \{\}$. Since S_{13} is empty, this concatenation is also empty and therefore adds no new variable to the set S_{14}.

Conclusion: $S_{14} = \{A\}$.