CS F351 Theory of Computation
Tutorial-7

Problem 1 Show using mathematical induction that the strings produced by the following context free grammar with productions

$$S \to 0 \mid S0 \mid 0S \mid 1SS \mid SS1 \mid S1S$$

has more 0's than 1's.

Solution:

Induction on the number of derivation steps which derives x.

Let $n_i(x)$ denote the number of i's in string x.

Base case: $S \to 0$ is the only production which produces string in a single derivation.

Induction hypothesis: Assume that if S derives x in one or more steps then $n_0(x) > n_1(x)$.

Induction step: Let x' be the string which is derivable from S in one or more steps and it uses at exactly one more derivation step than the number of derivation steps is used to derive x.

To derive x' from S if we may use any of the productions given above. We will prove that in all cases $n_0(x') > n_1(x')$.

$S \to 0S \mid S0$ has been used first then by induction hypothesis we can say that right side S derive string in which $n_0(x) > n_1(x)$ after appending or prepending 0 in x this inequality will hold.

$S \to S1S \mid 1SS \mid SS1$ has been used first. For the sake of clarity, relabel the symbols on the right hand side as $S \to S_1 1 S_2 \mid 1S_1S_2 \mid S_1S_2 1$. Also, assume that S_1 derives x_1 and S_2 derives x_2. Then by induction hypothesis we can say that $n_0(x_1) > n_1(x_1)$ and $n_0(x_2) > n_1(x_2)$. Hence $n_0(x_1 1 x_2) > n_1(x_1 1 x_2)$, $n_0(x_1x_2 1) > n_1(x_1x_2 1)$ and $n_0(1x_1x_2) > n_1(1x_1x_2)$.

Problem 2 consider a grammar $G = (V, T, P, S)$ where $V = \{S, A, B\}$, $T = \{a, b\}$ and

$P = \{S \to aB \mid bA, A \to a \mid aS \mid BAA, B \to b \mid bS \mid ABB\}$.

(a) Show that $ababba \in L(G)$.

Solution: $S \Rightarrow aB \Rightarrow abS \Rightarrow abaB \Rightarrow ababS \Rightarrow ababbA \Rightarrow ababba$

(b) Give a property defining $L(G)$.

Solution: $L(G) = \{w \in \{a, b\}^* : w \text{ has the same number of } a's \text{ and } b's\}$. To prove the correctness, we do the case analysis:

(i) The derivation starts with $S \to aB$. The next steps are the applications of the rules: $R_1 : B \to b$, $R_2 : B \to bS$, or $R_3 : B \to ABB$.
By R_1, we get $ab \in L$.
By R_2, we get $S \overset{*}{\Rightarrow} abS$.
By R_3, we get $S \overset{*}{\Rightarrow} aABB$. Here, A can become a, or aS or BAA. In all these cases, we get words with the same number of a's and b's.

(ii) The derivation starts with $S \to bA$. The case is similar to the above.

Problem 3 Construct CFG for the following languages.

(a) $\{wcw^R : w \in \{a,b\}\}$.

Solution: $S \to aSa|bSb|c$

(b) $\{ww^R : w \in \{a,b\}\}$.

Solution: $S \to aSa|bSb|\epsilon$

(c) $\{w \in \{a,b\}^* : w$ has twice as many $b's$ as $a's\}$

Solution: $S \to \epsilon$,

$S \to Sabb \mid aSbb \mid abSb \mid abbS$,

$S \to Sbab \mid bSab \mid baSb \mid babS$, and

$S \to Sbba \mid bSba \mid bbSa \mid bbaS$.

Problem 4 Show that the grammar $G = (V, T, P, S)$ where $V = \{S\}$, $T = \{a, b\}$, and $P = \{S \to aSa \mid bSb \mid a \mid b \mid \epsilon\}$ generates the language $L(G) = \{w \in \{a,b\}^* : w = w^R\}$.

Solution:

Observation: For any three strings $x, y, z \in \Sigma^*$, we have $(xyz)^R = z^R y^R x^R$.

Observe that the rules $S \to aSa \mid bSb \mid \epsilon$ generate the language $L_1 = \{ww^R : w \in \Sigma^*\}$. Further, by adding $S \to a \mid b$, we get the language $L(G) = L_1 \cup \{waw^R : w \in \Sigma^*\} \cup \{wbw^R : w \in \Sigma^*\}$.

We note that any string $w \in L(G)$ must have the form $w = xx^R$ or $w = xax^R$, or $w = xbx^R$ for some $x \in \{a,b\}^*$.

(a) *Suppose that* $w = xx^R$. Thus, $w^R = (xx^R)^R = (x^R)^R x^R = xx^R = w$.

(b) *Suppose that* $w = xax^R$. Thus, $w^R = (xax^R)^R = (x^R)^R a^R x^R = xax^R = w$. (note that $a^R = a$)

(c) *Suppose that* $w = xbx^R$. Thus, $w^R = (xbx^R)^R = (x^R)^R b^R x^R = xbx^R = w$. (note that $b^R = b$)
