Solutions to Assignment 3 (60 marks)

5.1.2[15]
(a) Leftmost: $S \Rightarrow A1B \Rightarrow 0A1B \Rightarrow 00A1B \Rightarrow 001B \Rightarrow 0010B \Rightarrow 00101B \Rightarrow 00101$
Rightmost: $S \Rightarrow A1B \Rightarrow A10B \Rightarrow A101B \Rightarrow A101 \Rightarrow 0A101 \Rightarrow 00A101 \Rightarrow 00101$
(b), (c) same idea as (a).

5.1.3[30] To do this question we follow the inductive definition of regular expressions (see section 3.1.2). The Basis Case is easy: $\varepsilon, \emptyset$, and symbols a. The corresponding CFGs are the following:

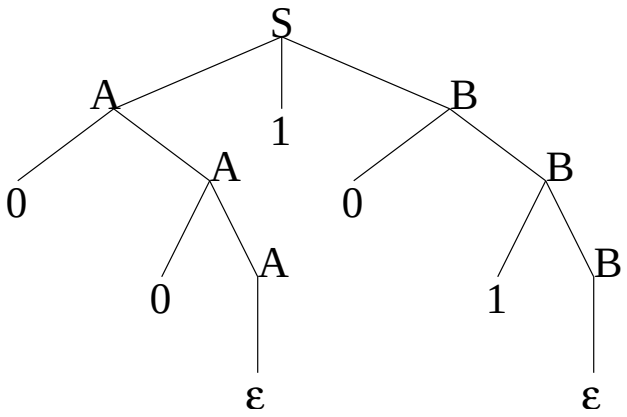
$$G_\varepsilon = (\{S\}, \{\varepsilon\}, \{S \to \varepsilon\}, S) \qquad \text{not marked}$$
$$G_\emptyset = (\{P\}, \emptyset, \emptyset, P)$$
$$G_a = (\{P\}, \{a\}, \{P \to a\}, P)$$

Induction: If E, F are regular expressions, then so are $E + F$, EF, E^* and (E). Consider E^*. By induction, we already have a CFG for E, call it $G_E = (V_E, T_E, P_E, S_E)$. We now show how to construct a CFG G_{E^*} such that $L(E^*) = L(G_{E^*})$. Let:

$$G_{E^*} = (V_E \cup \{N\}, T_E, P_E \cup \{N \to NS_E|\varepsilon\}, N)$$

where N is a *new* variable. Note that by adding the production rule $N \to NS_E|\varepsilon$, we allow any number of repetitions of strings derived from S_E, which has the effect of allowing us to derive from N any string in $L(E^*)$.

Here are the remaining three cases (we assume that we have CFGs G_E and G_F; we also assume that $V_E \cap V_F$, because if they are not, we can always rename the variables of V_F to *new* variable names, *without* altering the meaning of the grammar!):

$$G_{E+F} = (V_E \cup V_F \cup \{N\}, T_E \cup T_F, P_E \cup P_F \cup \{N \to S_E|S_F\}, N)$$
$$G_{EF} = (V_E \cup V_F \cup \{N\}, T_E \cup T_F, P_E \cup P_F \cup \{N \to S_E S_F\}, N)$$
$$G_{(E)} = (V_E \cup \{N\}, T_E \cup \{(,)\}, P_E \cup \{N \to (S_E)\}, N)$$

5.2.1[15] Here we just present a solution for (a); same idea for (b) and (c).