Program Analysis: Assignment-2

Tushar Goyal - UG20210087

▼ Question 1

Translate the following regular expression into a context-free grammar: (ab*)+(bc)?. See section 2.2 of the textbook for the meaning of ? in regular expressions. (Convention: For easy mnemonics, use non-terminal names like BSTAR, BCQUESTION etc.).

$$
\begin{aligned}
S &\rightarrow (\text{ABSTARPLUS})(\text{BCQUESTION}) \\
\text{ABSTARPLUS} &\rightarrow (\text{ABSTAR}) \mid (\text{ABSTARPLUS})(\text{ABSTARPLUS}) \\
\text{BCQUESTION} &\rightarrow \epsilon \mid \text{bc} \\
\text{ABSTAR} &\rightarrow a(\text{BSTAR}) \\
\text{BSTAR} &\rightarrow \epsilon \mid b(\text{BSTAR})
\end{aligned}
$$

▼ Question 2

Write an unambiguous grammar that accepts strings that match the regular expression a*b* and have more a's than b's.

$$
\begin{aligned}
S &\rightarrow aAB \\
A &\rightarrow aA \mid \epsilon \\
B &\rightarrow \epsilon \mid aBb
\end{aligned}
$$

▼ Question 3

Write an unambiguous grammer over the alphabet {a, b, c, +, . , !} that accepts all boolean expressions over three input binary signals {a, b, c}, operated upon by three boolean operators AND (.), OR (+) and NOT (!). Break the ambiguity using the following precedence order from

highest to lowest: !, ., +. In addition, enforce association from the left. Assume that no more than one NOT operator can be applied to a single expression, e.g., !!a is never present in the input string. Clarification: the input strings do not contain parentheses.

$$
\begin{aligned}
S &\rightarrow S + T \mid T \\
T &\rightarrow T.F \mid F \\
F &\rightarrow !id \mid id \\
id &\rightarrow a \mid b \mid c
\end{aligned}
$$