The grammar below is a simplification of C++ (global) function headers including pointer notation, e.g.:

\[
\text{int } \ast \text{ my\_function( int } \ast \text{ A, int n )}
\]

In the following grammar, words beginning with capital letters are variables. Words all in lower case are terminals (tokens). Special characters including as “(”, “)”, “,”, and “*” are terminals.

\[
\begin{align*}
\text{Fun} & \rightarrow \text{Type identifier ( Plist )} \\
\text{Type} & \rightarrow \text{int Slist | double Slist} \\
\text{Slist} & \rightarrow \ast \text{ Slist | epsilon} \\
\text{Plist} & \rightarrow \text{Type identifier MoreList} \\
\text{MoreList} & \rightarrow , \text{Type identifier MoreList | epsilon}
\end{align*}
\]

The FIRST and FOLLOW sets for this grammar are as follows:

**First sets:**

\[
\begin{align*}
\text{FIRST(Fun)} & = \{ \text{int double } \} \\
\text{FIRST(Type)} & = \{ \text{int double } \} \\
\text{FIRST(Slist)} & = \{ \ast \text{ epsilon } \} \\
\text{FIRST(Plist)} & = \{ \text{int double } \} \\
\text{FIRST(MoreList)} & = \{ \text{epsilon , } \}
\end{align*}
\]

**Follow sets:**

\[
\begin{align*}
\text{FOLLOW(Fun)} & = \{ \text{eof } \} \\
\text{FOLLOW(Type)} & = \{ \text{identifier } \} \\
\text{FOLLOW(Slist)} & = \{ \text{identifier } \} \\
\text{FOLLOW(Plist)} & = \{ \} \\
\text{FOLLOW(MoreList)} & = \{ \} \\
\end{align*}
\]

**Your task:**

1. Give a predictive parsing table for the given grammar.
DFA, NFA, and Regular Expressions

1. Given the following NFA, construct an equivalent DFA.

![Graph](image)

2. Given the following regular expression, convert to an equivalent NFA.

\[(a | b)^* (b | c)\]