1. Consider a boolean function \( f \) with \( n \) variables:

\[
\text{BOOL} \overset{\text{def}}{=} \{1, 0\} \\
f : \text{BOOL}^n \rightarrow \text{BOOL}
\]

Given such a function \( f \), the following construction allows us to calculate a deterministic finite state automaton, which accepts strings \( w \) over the alphabet 1 and 0 if and only if \( f(w) = 1 \):

\[
M \overset{\text{def}}{=} \langle \Sigma, K, F, S, t \rangle
\]

where

- Alphabet: \( \Sigma \overset{\text{def}}{=} \text{BOOL} \)
- States: \( K \overset{\text{def}}{=} \{ \sigma \_ \varepsilon \} \cup \{ \sigma \_ x \mid x \in \text{BOOL}^*, \exists y : \text{BOOL}^* \cdot f(xy) = 1 \} \)
- Final states: \( F \overset{\text{def}}{=} \{ \sigma \_ x \mid f(x) = 1 \} \)
- Start state: \( S \overset{\text{def}}{=} \sigma \_ \varepsilon \)
- Transition function: \( t(\sigma \_ x, a) \overset{\text{def}}{=} \sigma \_ xa \) if \( \sigma \_ xa \in K \)

(a) Using the above description construct an automaton for the function OR:
Or : $\text{Bool}^2 \rightarrow \text{Bool}$

\[
\begin{align*}
\text{Or}(00) & \overset{\text{def}}{=} 0 \\
\text{Or}(01) & \overset{\text{def}}{=} 1 \\
\text{Or}(10) & \overset{\text{def}}{=} 1 \\
\text{Or}(11) & \overset{\text{def}}{=} 1
\end{align*}
\]

(b) Minimise the automaton you obtained in question (a).

2. (a) Formally state the following property of language $L$:

Every occurrence of symbol $a$ in a string in $L$ is always preceded by an occurrence of symbol $b$.

(b) Use standard constructions to obtain a finite state automaton which recognises the language described by the regular expression $(b^+a^+ + b^+)$.

(c) Using standard constructions transform the following automaton into an equivalent deterministic one:

(c) Using standard constructions transform the following automaton into an equivalent deterministic one:

3. (a) Given a string $s$ over alphabet $\{o, z, x\}$, we define its inversion $s^\downarrow$ to be $s$ with all occurrences of $o$ replaced by $o$ and occurrences of $z$ replaced by $o$. Thus, for example $(zzzoxo)^\downarrow = ooozxz$.

Formally define $s^\downarrow$.

(b) Using string induction prove that $(s^\downarrow)^\downarrow = s$.

(c) We define the inversion of a language $L$ as follows:

$L^\downarrow \overset{\text{def}}{=} \{ s^\downarrow \mid s \in L \}$
It can be proved that the inversion of the language described by a regular expression \( e \) is the language described by \( e \) with all instances of \( o \) in \( e \) replaced by \( z \) and instances of \( z \) replaced by \( o \).

Using this result (which you are not required to prove), and using standard constructions give a regular expression which describes the inversion of the language accepted by the following automaton:

4. (a) Consider the regular grammar \( G = \langle \Sigma, N, P, S \rangle \), where \( \Sigma = \{o,x\} \), \( N = \{O,X\} \), \( S = O \) and \( P \) is the set of the following productions:
   
   \[
   \begin{align*}
   O & \rightarrow oX \mid \varepsilon \\
   X & \rightarrow xX \mid xO
   \end{align*}
   \]

   Prove that for any string \( s \) in \( L(G) \), the string \( sox \) is also in \( L(G) \).

   **Hint:** You may assume that every sentential form of a regular grammar is of the form \( sA \) or \( s \), where \( s \in \Sigma^* \) and \( A \in N \).

   (b) Using standard constructions give an automaton which accepts the language \( (L(G)) \cup (L(G))^2 \).

5. (a) Prove that all strings in the language generated by the grammar with start symbol \( S \) and the following production rules have a number of \( a \) which is exactly divisible by 3:

   \[
   \begin{align*}
   S & \rightarrow AAA \mid ASASA \\
   A & \rightarrow ab \mid aAASa
   \end{align*}
   \]

   (b) Consider the following description of a non-deterministic push-down automaton:

   \[
   M = \langle K, \Sigma, \Sigma, P, k_1, A_1, F \rangle
   \]

   where \( \Sigma = \{a,b\} \), \( \Gamma = \{a\} \), \( K = \{X, Y, Z\} \), \( k_1 = X \), \( F = \{Y, Z\} \), \( A_1 = \perp \) and transition function \( P \) is defined below:

   \[
   t((X, \perp), a) = \{(X, a\perp), (X, aa\perp)\}
   \]
\[ t((X, a), a) = \{(X, aa), (X, aaa)\} \]
\[ t((X, a), b) = \{(Y, a)\} \]
\[ t((X, a), c) = \{(Z, a)\} \]
\[ t((Y, a), b) = \{(Y, \varepsilon)\} \]
\[ t((Z, a), c) = \{(Z, \varepsilon), (Y, \varepsilon)\} \]

\(P\) returns the empty set for all other inputs.

i. Show your understanding of the above formal description by drawing the NPDA just described.

ii. Using standard constructions convert the above NPDA into an equivalent one but which has only one final state and which empties the stack if and only if the final state is reached.