1.

(a) 

(b) See Slide 82 of Week 2 Lecture Notes

(c) According to the algorithm in Slide 95, both are already minimal. In fact, it is easy to see that they are identical up to state renaming.

2.

(a) 

The start symbol: \textit{bexpr}

Nonterminals: \textit{bexpr, bterm, bfactor}

Terminals: \textit{or and not ( ) true false}

(b) The leftmost derivation:

\[
\begin{align*}
bexpr & \Rightarrow_{lm} bterm & P2 \\
& \Rightarrow_{lm} bfactor & P4 \\
& \Rightarrow_{lm} \text{not} bfactor & P5 \\
& \Rightarrow_{lm} \text{not} ( bexpr ) & P6 \\
& \Rightarrow_{lm} \text{not} ( bexpr or bterm ) & P1 \\
& \Rightarrow_{lm} \text{not} ( bterm or bterm ) & P2 \\
& \Rightarrow_{lm} \text{not} ( bterm and bfactor or bterm ) & P3 \\
& \Rightarrow_{lm} \text{not} ( bfactor and bfactor or bterm ) & P4 \\
& \Rightarrow_{lm} \text{not} ( false and bfactor or bterm ) & P8 \\
& \Rightarrow_{lm} \text{not} ( false and true or bterm ) & P7 \\
& \Rightarrow_{lm} \text{not} ( false and true or bfactor ) & P4 \\
& \Rightarrow_{lm} \text{not} ( false and true or true ) & P7 
\end{align*}
\]
(c) The rightmost derivation:

\[
\begin{align*}
\text{bexpr} & \Rightarrow_{\text{rm}} \text{bterm} & \text{P2} \\
& \Rightarrow_{\text{rm}} \text{bfactor} & \text{P4} \\
& \Rightarrow_{\text{rm}} \text{not} \text{bfactor} & \text{P5} \\
& \Rightarrow_{\text{rm}} \text{not} (\text{bexpr}) & \text{P6} \\
& \Rightarrow_{\text{rm}} \text{not} (\text{bexpr or bterm}) & \text{P1} \\
& \Rightarrow_{\text{rm}} \text{not} (\text{bexpr or bfactor}) & \text{P4} \\
& \Rightarrow_{\text{rm}} \text{not} (\text{bexpr or true}) & \text{P7} \\
& \Rightarrow_{\text{rm}} \text{not} (\text{bterm or true}) & \text{P2} \\
& \Rightarrow_{\text{rm}} \text{not} (\text{bterm and bfactor or true}) & \text{P3} \\
& \Rightarrow_{\text{rm}} \text{not} (\text{bterm and true or true}) & \text{P7} \\
& \Rightarrow_{\text{rm}} \text{not} (\text{bfactor and true or true}) & \text{P4} \\
& \Rightarrow_{\text{rm}} \text{not} (\text{false and true or true}) & \text{P8}
\end{align*}
\]

(d) The parse tree:

```
  bexpr
   |   
  bterm
   |   
  bfactor
   |   
  not  bfactor
   |   
  (   bexpr   )
   |   
  bexpr or bterm
   |   
  term and bfactor true
   |   
  bterm and bfactor true
   |   
  bfactor true
   |   
  false
```

(e)

Precedence:

or, and, not are in the order of increasing precedence.

Associativity:

and and or are left associative and not is right associative
3.
(a) The parse tree:

(b) In this revised grammar, the roles of or and and have been swapped. Thus, this grammar gives or higher precedence than and. As a consequence, the given expression would evaluate to true rather than false, an undesirable situation that should be avoided! This has been reflected in the parse tree in 3(a).

4.
(a) The parse tree:

(b) The parse tree:
(c) In this revised grammar, and and or have the same precedence and are both left associative. This has been reflected in the parse trees drawn in 4(a) and 4(b).

(d) At first glance, this grammar looks ambiguous because both and and or appear in the same production. However, it turns out that parse trees generated according to the grammar are unique.

5.

(a) 

(b) 

Neither operator has a higher precedence than another.

Precedence of or and and:

The parse tree 5(a) tells us that or has higher precedence than and but the parse tree 5(b) says the the opposite.

Associativity of or and and:

Consider the two parse trees for not (false or true or true), which can be obtained from 5(a) and 5(b) by replacing the two and’s with or’s. Clearly, or is left associative according the left tree
and right associative according to the right tree.

Similarly, consider the two parse trees for not (false and true and true), which can be obtained from 5(a) and 5(b) by replacing the two or’s with and’s. Clearly, and is left associative according the left tree and right associative according to the right tree.

This is why in the original grammar of Question 2, three different nonterminals are used, making sure that all the operators are given the appropriate precedence and associativity.