1 CFGs

Give CFG grammars (using BNF notation) for the following languages or say it cannot be done

1. All strings over the alphabet \{0,1,2,3\} sorted in decreasing order.

   Answer:

   \[
   \begin{align*}
   <S> & ::= <\text{threes}> <\text{twos}> <\text{ones}> <\text{zeros}> \mid \text{epsilon} \\
   <\text{threes}> & ::= 3 <\text{threes}> \mid \text{epsilon} \\
   <\text{twos}> & ::= 2 <\text{twos}> \mid \text{epsilon} \\
   <\text{ones}> & ::= 1 <\text{ones}> \mid \text{epsilon} \\
   <\text{zeros}> & ::= 0 <\text{zeros}> \mid \text{epsilon}
   \end{align*}
   \]

2. All strings over the alphabet \{a,b\}, in which every b is both immediately preceded by and followed by at least one a.

   Answer:

   \[
   \begin{align*}
   <S> & ::= a <\text{BA} > <S> \mid \text{epsilon} \\
   <\text{BA} > & ::= b \ a <\text{BA} > \mid \text{epsilon}
   \end{align*}
   \]

3. \(a^n b^m a^n b^m n, m \geq 0\)

   cannot be expressed by CFG
2 english descriptions

Give English descriptions of the languages described by each of the following grammars.

a) \(<S> ::= <S> a <S> b <S> | <S> b <S> a <S> | \text{epsilon}\)

Answer:
    All strings with an equal number of a’s and b’s.

Example:
\[
\begin{align*}
<S> &\Rightarrow <S> a <S> b <S> \\
    &\Rightarrow <S> b <S> a <S> a <S> b <S> \\
    &\Rightarrow <S> b <S> a a <S> b <S> \\
    &\Rightarrow <S> b <S> a a <S> b <S> a <S> b <S> \\
    &\Rightarrow \text{baabab}
\end{align*}
\]

b) \(<S> ::= a <S> a | b <S> b | c <S> c | \text{epsilon}\)

Answer: All even length strings over \{a,b,c\}, where the second half of the string is the reverse of the first half.

Example:
\[
\begin{align*}
<S> &\Rightarrow a <S> a \\
    &\Rightarrow a b <S> b a \\
    &\Rightarrow a b c <S> c b a \\
    &\Rightarrow a b c b <S> b c b a \\
    &\Rightarrow \text{abcba}
\end{align*}
\]

c) Adjust the grammar in (b) to make it generate all palindromes.

What do we need? We need to also generate odd length strings.

Answer: \(<S> ::= a <S> a | b <S> b | c <S> c | a | b | c | \text{epsilon}\)
3 ambiguity

a) \(<S> ::= <S> \text{ and } <S> \mid <S> \text{ or } <S> \mid a \mid b \mid c\)

Is this grammar ambiguous?
Yes. Two distinct parse trees for string ‘a and b or c’.

Develop a grammar that generates the same strings, but is not ambiguous.

\(<S> ::= <OR>\)
\(<OR> ::= <AND> \mid <OR> \text{ or } <AND>\)
\(<AND> ::= <VAR> \mid <AND> \text{ and } <VAR>\)
\(<VAR> ::= a \mid b \mid c\)

b) \(<S> ::= a <S> a \mid b <S> b \mid \text{epsilon}\)

Not ambiguous.