Fun with grammars

Here are a set of abstract languages and grammars which can generate them.

**Set of all strings over A,B ending in A**

\[
\begin{align*}
  s & : zA \\
  z & : Az \mid Bz \mid \text{empty}
\end{align*}
\]

Note that the non-terminal \( z \) generates all strings composed of A's and B's. The non-terminal \( s \) simply appends all strings generated by \( z \) with an \( A \) to give the desired result.

**Set of all strings over A,B beginning and ending in A**

\[
\begin{align*}
  s & : AzA \mid A \\
  z & : Az \mid Bz \mid \text{empty*}
\end{align*}
\]

Here, \( z \) plays the same role as before. In this case, \( s \) simply prepends an \( A \) as well to give the desired result.

**Set of all strings over A,B with three consecutive A’s**

\[
\begin{align*}
  s & : zAAAz \\
  z & : Az \mid Bz \mid \text{empty*}
\end{align*}
\]

Now, \( s \) places a \( z \) at both ends to ensure that there are three consecutive A’s somewhere in the sentence.

**Set of all strings over A,B such that there is a pair of A’s separated by \( 4i \), \( i \geq 0 \), characters**

\[
\begin{align*}
  s & : zAlAz \\
  z & : Az \mid Bz \mid \text{empty*} \\
  l & : fl \mid \text{empty*} \\
  f & : Ag \mid Bg \\
  g & : Ah \mid Bh \\
  h & : Ai \mid Bi \\
  i & : A \mid B
\end{align*}
\]

Similar to before, but note that \( l \) generates zero or more \( f \) strings. Note that an \( i \) string is composed of a single character. Therefore, an \( h \) string is composed of two characters, a \( g \) string is composed of three characters and and \( f \) string is composed of four characters. Since there are zero or more \( f \) strings between the two \( A \)'s in an \( s \) string, we get the desired language.

**Set of all strings over A,B such that at no two A’s and no two B’s are adjacent**

\[
\begin{align*}
  s & : a \mid b \\
  b & : Ba \mid \text{empty*} \\
  a & : Ab \mid \text{empty*}
\end{align*}
\]

The non-terminals \( a \) and \( b \) flip flop back and forth to ensure no two like characters are adjacent.

**Palindromes over A,B**

\[
\begin{align*}
  s & : A \mid B \mid AsA \mid BsB \mid \text{empty*}
\end{align*}
\]

If we place an \( A \) in the front, we must place one in the back. Likewise for \( B \)'s.

**Balanced parentheses**

\[
\begin{align*}
  s & : (s) \mid ss \mid \text{empty*}
\end{align*}
\]

or

\[
\begin{align*}
  s & : (s)s \mid \text{empty*}
\end{align*}
\]

It is a temptation to define \( s \) as \((s)\). This disallows such balanced strings as \((()())\).
Set of a strings over $A,B$ so that the number of $A$s equals the number of $B$'s

$s : Ab \mid Ba \mid \text{empty}$
$a : As \mid Baa$
$b : Bs \mid Abb$

Note that $b$ stands for strings with 1 more $B$ than $A$ and $a$ stands for strings with 1 more $A$ than $B$. These non-terminals arise naturally out of the observation that the strings must start with an $A$ or $B$ and the consequences of that starting character on the remainder of the string.