Example: A simple programming language grammar.

\[
G: \ <\text{Exp}>^* \ ::= \ <\text{Exp}> + \ <\text{Exp}>
\| \ <\text{Exp}> * \ <\text{Exp}>
\| \ (\ <\text{Exp}\ >) \\
\| \ a \\
\| \ b \\
\| \ c
\]

Terminal symbols!!!

\[
S = a \\
S = a + b \\
S = a + b * c \\
S = (a + b) * c \\
S = ((a + b)) \\
S = c(a + b) \\
S = c(b) \\
S = (c)+(b) \\
S = b++
\]

\[
S \in L(G)\?
\]

2, 4, 5, 6
The empty symbol: `<empty>`

You can think of `<empty>` being defined by the implicit rule:

```
<empty> ::= ""
```

That is the `<empty>` symbol derives nothing.
Consider the grammar:

\[
G: \quad \langle A \rangle^* ::= a \langle B \rangle \mid a \\
    \langle B \rangle ::= b \langle B \rangle \mid b
\]

\[
G': \quad \langle A \rangle^* ::= a \langle B \rangle \\
    \langle B \rangle ::= b \langle B \rangle \mid \langle \text{empty} \rangle
\]
Consider the following grammar fragment:

\[
\begin{align*}
<\text{if-stmt}> & ::= \text{if} \ <\text{exp}> \ \text{then} \ <\text{stmt}> \ <\text{else-part}> \\
<\text{else-part}> & ::= \text{else} \ <\text{stmt}> \mid <\text{empty}> \\
<\text{exp}> & ::= \ldots \\
<\text{stmt}> & ::= \ldots
\end{align*}
\]
2.1 a) Let $L(G)$ be the language of all string consisting of zero or more a’s.

2.1 i) Let $L(G)$ be the set of strings consisting of one or more a’s with a comma between each a and the next.

2.1 d) Let $L(G)$ be the set of all strings consisting of one or more digits 0 – 9.