Example: A simple programming language grammar.

$$G: \langle \text{Exp}\rangle^* ::= \langle \text{Exp}\rangle + \langle \text{Exp}\rangle$$
$$| \langle \text{Exp}\rangle \ast \langle \text{Exp}\rangle$$
$$| (\langle \text{Exp}\rangle)$$
$$| a$$
$$| b$$
$$| c$$

Terminal symbols!!!

$$S = a$$
$$S = a + b$$
$$S = a + b \ast c$$
$$S = (a + b) \ast c$$
$$S = ((a + b))$$
$$S = c(a + b)$$
$$S = (c) + (b)$$
$$S = b++$$
Grammars in Action

- The empty symbol: `<empty>`
- The only non-terminal that does not have a rule defining it.
- That is the `<empty>` symbol derives nothing.
Consider the grammar:

\[
G: \quad <A> ::= a <B> \ | \ a \\
    <B> ::= b <B> \ | \ b
\]

\[
G': \quad <A> ::= a <B> \\
         <B> ::= b <B> \ | \ \langle \text{empty} \rangle
\]
Consider the following grammar fragment:

\[
<\text{if-stmt}> ::= \text{if} \ <\text{exp}> \ \text{then} \ <\text{stmt}> \ <\text{else-part}>
\]
\[
<\text{else-part}> ::= \text{else} \ <\text{stmt}> \mid \ <\text{empty}>
\]
\[
<\text{exp}> ::= \ldots
\]
\[
<\text{stmt}> ::= \ldots
\]
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.

HW#1 – see website