

An alternative way to state the productions of a grammar is by using **Backus normal form** (or **Backus–Naur form** or **BNF**). In BNF the nonterminal symbols typically begin with “<” and end with “>.” The production $S \rightarrow T$ is written $S ::= T$. Productions of the form

$$S ::= T_1, \quad S ::= T_2, \quad \dots, \quad S ::= T_n$$

may be combined as

$$S ::= T_1 \mid T_2 \mid \dots \mid T_n.$$

The bar “|” is read “or.”

EXAMPLE A GRAMMAR FOR INTEGERS

An integer is defined as a string consisting of an optional sign (+ or –) followed by a string of digits (0 through 9). The following grammar generates all integers.

$$\langle \text{digit} \rangle ::= 0 \mid 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9$$

$$\langle \text{integer} \rangle ::= \langle \text{signed integer} \rangle \mid \langle \text{unsigned integer} \rangle$$

$$\langle \text{signed integer} \rangle ::= + \langle \text{unsigned integer} \rangle \mid - \langle \text{unsigned integer} \rangle$$

$$\langle \text{unsigned integer} \rangle ::= \langle \text{digit} \rangle \mid \langle \text{digit} \rangle \langle \text{unsigned integer} \rangle$$

The starting symbol is $\langle \text{integer} \rangle$.

For example, the derivation of the integer –901 is

$$\begin{aligned} \langle \text{integer} \rangle &\Rightarrow \langle \text{signed integer} \rangle \\ &\Rightarrow - \langle \text{unsigned integer} \rangle \\ &\Rightarrow - \langle \text{digit} \rangle \langle \text{unsigned integer} \rangle \\ &\Rightarrow - \langle \text{digit} \rangle \langle \text{digit} \rangle \langle \text{unsigned integer} \rangle \\ &\Rightarrow - \langle \text{digit} \rangle \langle \text{digit} \rangle \langle \text{digit} \rangle \\ &\Rightarrow -9 \langle \text{digit} \rangle \langle \text{digit} \rangle \\ &\Rightarrow -90 \langle \text{digit} \rangle \\ &\Rightarrow -901. \end{aligned}$$

In the notation of Definition 10.3.3, this language consists of

1. The set $N = \{ \langle \text{digit} \rangle, \langle \text{integer} \rangle, \langle \text{signed integer} \rangle, \langle \text{unsigned integer} \rangle \}$ of nonterminal symbols
2. The set $T = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, +, -\}$ of terminal symbols
3. The productions

$$\langle \text{digit} \rangle \rightarrow 0, \dots, \langle \text{digit} \rangle \rightarrow 9$$

$$\langle \text{integer} \rangle \rightarrow \langle \text{signed integer} \rangle$$

$$\langle \text{integer} \rangle \rightarrow \langle \text{unsigned integer} \rangle$$

$$\langle \text{signed integer} \rangle \rightarrow + \langle \text{unsigned integer} \rangle$$

$$\langle \text{signed integer} \rangle \rightarrow - \langle \text{unsigned integer} \rangle$$

$$\langle \text{unsigned integer} \rangle \rightarrow \langle \text{digit} \rangle$$

$$\langle \text{unsigned integer} \rangle \rightarrow \langle \text{digit} \rangle \langle \text{unsigned integer} \rangle$$

4. The starting symbol $\langle \text{integer} \rangle$. ■

Computer languages, such as FORTRAN, Pascal, and C++, are typically specified in BNF. Example 10.3.7 shows how an integer constant in a computer language might be specified in BNF.