Let σ designate a metasyntactic symbol, e.g., <u>identifier</u>, or an explicit token, e.g., FOR. Then:

Rule 1: $\sigma_1 \, \sigma_2 \ldots \sigma_n$ designate the concatenation of σ_1, σ_2, $\ldots$ σ_n $(n \geqslant 1)$. Let α designate such a concatenation.

Rule 2: $\alpha_1 \mid \alpha_2 \mid \ldots \mid \alpha_n$ designates the alternation among α_1, α_2, $\ldots$ α_n $(n \geqslant 1)$. Let β designate such an alternation.

Rule 3: The brackets { and }, without subscripts or superscripts, are used for grouping.

Rule 4: $[\beta] \equiv \beta \mid$ empty

Rule 5: $\underline{[}$ designates the character [.
$\underline{]}$ designates the character].

Rule 6:
$$\left\{ \begin{array}{c} \beta_1 \\ \beta_2 \\ \cdot \\ \cdot \\ \cdot \\ \beta_n \end{array} \right\} \equiv \beta_1 \mid \beta_2 \mid \ldots \mid \beta_n$$

$$\left[\begin{array}{c} \beta_1 \\ \beta_2 \\ \cdot \\ \cdot \\ \cdot \\ \beta_n \end{array} \right] \equiv \beta_1 \mid \beta_2 \mid \ldots \mid \beta_n \mid \text{empty}$$

Rule 7: σ_* $\equiv$ 0 or more σ's

$\quad\quad\quad$ $\{\beta\}_*$ $\equiv$ 0 or more β's

Rule 8: σ_{*+1} $\equiv$ 1 or more σ's

$\quad\quad\quad$ $\{\beta\}_*$ $\equiv$ 1 or more β's

Rule 9: $\{\beta\}_*^{\delta}$ $\equiv$ 0 or more β's separated by δ
$\quad\quad\quad\quad\quad\quad\quad\quad\quad$ (where δ is an explicit token)

Rule 10: $\{\beta\}_{*+1}^{\delta}$ $\equiv$ 1 or more β's separated by δ

Rule 11: $\{\ \alpha_1\ ||\ \alpha_2\ ||\ \ldots\ ||\ \alpha_n\}$ $\equiv$ exactly one occurrence of each α_i
$\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad$ in any order. (Each α_i is either a σ or
$\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad$ a $\{\beta\}$.

Rule 12: $\{\alpha_1\ ||\ \alpha_2\ ||\ \ldots\ ||\ \alpha_n\}^{\delta}$ $\equiv$ exactly one occurrence of each α_i,
$\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad$ with the non-empty occurrences separated
$\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad$ by δ.

Rule 13:   The notation N1 preceding an equation designates footnote number 1;
$\quad\quad\quad\quad$ N2 designates footnote number 2, etc.)

Rule 14:   Terminal symbols in syntax equations are written with all capital
$\quad\quad\quad\quad$ Roman letters.

Rule 15:   Non-primitive syntactic entities, e.g., those things which are on
$\quad\quad\quad\quad$ the left-hand side of syntax equations, are written in all lower
$\quad\quad\quad\quad$ case Roman letters.

Rule 16:   The name of a non-primitive syntactic entity is written without
$\quad\quad\quad\quad$ blanks. If it is composed of more than one word, any character
$\quad\quad\quad\quad$ appropriate to the situation may be used as a hyphenator. If split
$\quad\quad\quad\quad$ over line boundaries, the hyphenator is not repeated on the second
$\quad\quad\quad\quad$ line.

Rule 17:   In prose description, the distinction between non-terminal syntactic
$\quad\quad\quad\quad$ entities and other uses of the same terms (e.g., "atom" as a LISP
$\quad\quad\quad\quad$ entity and "atom" as a chemical entity should be maintained by using
$\quad\quad\quad\quad$ good English. If the distinction is not clear, then spell it out
$\quad\quad\quad\quad$ with a few extra words.

Rule 18:   The use of italics is the normal one of emphasis. If the first
$\quad\quad\quad\quad$ mention of a term is a prose description, it is also permissible
$\quad\quad\quad\quad$ to use italics in this instance.

Examples:

SL: section-list ≡   {section-name}$^,_{*+1}$

IL: section-list ≡   section-name | (section-name$_{*+1}$)

IL: parameter-list ≡   (parameter$_*$ indef-parameter)

SL: attribute-list ≡   {type || storage-mode || reference-mode},

IL: attribute-list ≡   {type || storage-mode || reference-mode}

NOTE: These rules are intended for use in connection with the syntax
        equations that show how the complete language is built up from
        tokens. In these equations, tokens are the lowest-level entities
        that are considered. However, there are also equations for the
        syntax of tokens; these equations describe how tokens are built
        up from characters. At this level, special provisions need to
        be made to handle matters such as spacing, since the problems
        of delimitation are much more acute there. These special
        provisions will be the subject of a later memo, probably attached
        to the memo on the syntax of tokens.