TO: LISP 2 Project Members

FROM: Paul Abrahams

SUBJECT: CONVENTIONS FOR WRITING LISP 2 SYNTAX EQUATIONS

COPIES TO:

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

Rule 1: σ₁ σ₂ ... σₙ designate the concatenation of σ₁, σ₂, ... σₙ (n ≥ 1). Let a designate such a concatenation.

Rule 2: α₁ | α₂ | ... | αₙ designates the alternation among α₁, α₂, ... αₙ (n ≥ 1). Let β designate such an alternation.

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

Rule 4: [β] ≡ β | empty

Rule 5: [ designates the character [.

} designates the character ].

Rule 6: β₁

β₂

... 

βₙ

} ≡ β₁ | β₂ | ... | βₙ

β₁

β₂

... 

βₙ

[ ] = β₁ | β₂ | ... | βₙ | empty
Rule 7: \( \sigma^* \equiv 0 \text{ or more } \sigma' \text{s} \)
\( \{\beta\}^* \equiv 0 \text{ or more } \beta' \text{s} \)

Rule 8: \( \sigma_{k+1} \equiv 1 \text{ or more } \sigma' \text{s} \)
\( \{\beta\}^* \equiv 1 \text{ or more } \beta' \text{s} \)

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

Rule 10: \( \{\beta\}_1^* \equiv 1 \text{ or more } \beta' \text{s} \text{ separated by } \delta \)

Rule 11: \( \{ a_1 | | a_2 | | ... | | a_n \} \equiv \text{ exactly one occurrence of each } a_i \) 
in any order. (Each \( a_i \) is either a \( \sigma \) or a \( \{\beta\} \).

Rule 12: \( \{ a_1 | | a_2 | | ... | | a_n \}^* \equiv \text{ exactly one occurrence of each } a_i \), 
with the non-empty occurrences separated by \( \delta \).

Rule 13: The notation \( N_1 \) preceding an equation designates footnote number 1; 
\( N_2 \) designates footnote number 2, etc.)

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

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

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

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

Rule 18: The use of italics is the normal one of emphasis. If the first mention of a term is a prose description, it is also permissible 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.