

programs look like; advanced users and language-system implementers, who need an exact, detailed definition to work from; and automatic tools, which derive other programs from your grammar automatically.

Perhaps even more important is the positive influence grammars have on language design. You can see this influence when you compare pre-BNF languages (like early Fortran) with post-BNF languages. A language with a simple, readable, short grammar has a simple, memorable phrase structure. That makes the language easier to learn and use.

Exercises

Exercise 1 Give a BNF grammar for each of the languages below. For example, a correct answer for “the set of all strings consisting of zero or more concatenated copies of the string ab” would be this grammar:

$$\langle S \rangle ::= ab \langle S \rangle \mid \langle \text{empty} \rangle$$

There are often many correct answers.

- a. The set of all strings consisting of zero or more as.
- b. The set of all strings consisting of an uppercase letter followed by zero or more additional characters, each of which is either an uppercase letter or one of the digits 0 through 9.
- c. The set of all strings consisting of one or more as.
- d. The set of all strings consisting of one or more digits. (Each digit is one of the characters 0 through 9.)
- e. The set of all strings consisting of zero or more as with a semicolon after each one.
- f. The set of all strings consisting of the keyword begin, followed by zero or more statements with a semicolon after each one, followed by the keyword end. Use the non-terminal $\langle \text{statement} \rangle$ for statements, and do not give productions for it.
- g. The set of all strings consisting of one or more as with a semicolon after each one.
- h. The set of all strings consisting of the keyword begin, followed by one or more statements with a semicolon after each one, followed by the keyword end. Use the non-terminal $\langle \text{statement} \rangle$ for statements, and do not give productions for it.
- i. The set of all strings consisting of one or more as, with a comma between each a and the next. (There should be no comma before the first or after the last.)

Further Reading

On the subject of formal language, the author recommends (with scarcely a blush) this excellent introduction:

Webber, Adam Brooks. *Formal Language: A Practical Introduction*. Wilsonville, OR: Franklin, Beedle & Associates, 2008.

Suppose the lexical structure of the language allows any number of spaces to occur anywhere in the expression. Give a BNF grammar that defines this explicitly, at the character level, using one grammar to capture both the phrase structure and the lexical structure. For example, your grammar should generate both $(a+b)$ and $(a + b)$. Use a single-quoted space, $' '$, to indicate the space character in your grammar.

$$\langle \text{exp} \rangle ::= \langle \text{exp} \rangle + \langle \text{exp} \rangle \mid \langle \text{exp} \rangle * \langle \text{exp} \rangle \mid (\langle \text{exp} \rangle) \mid a \mid b \mid c$$

Exercise 4 Consider the earlier simple grammar for expressions:

Exercise 3 Give a syntax diagram for each of the languages of Exercise 1. Use branching and loops in your syntax diagrams to make them as clear as possible.

Exercise 2 Give an EBNF grammar for each of the languages of Exercise 1. Use the EBNF extensions wherever possible to simplify the grammars. In particular, you should eliminate explicit recursion from the grammars wherever possible. Don't forget to put single quotes around any EBNF metasymbol when using it as a token.

- i. The set of all strings consisting of an open bracket $[$ followed by a list of one or more digits separated by commas, followed by a closing bracket (the symbol $]$).
- j. The set of all strings consisting of an open bracket $[$ followed by a list of one or more digits separated by commas, followed by a closing bracket (the symbol $]$).
- k. The set of all strings consisting of zero or more as, with a comma between each a and the next. (There should be no comma before the first or after the last.)
- l. The set of all strings consisting of an open bracket $[$ followed by a list of zero or more digits separated by commas, followed by a closing bracket (the symbol $]$).