ECMA/TR 2, Formal Definition of the Syntax of COBOL

SCOPE
This formal definition of the syntax of COBOL was prepared by the
ECMA Technical Committee on COBOL (TC6).

The work was initially undertaken at the request of the CODASYL
COBOL Publication Subcommittee. It resulted in the publication in
1967 of a Preliminary Edition based on COBOL Edition 65. This new
edition is based on the ISO Draft Recommendation 1989 on COBOL.

The document comprises four distinct parts and an appendix.
The first part briefly describes the notation used, the second part is the
formal definition of the COBOL syntax, the third part is an index
showing where each meta-variable is defined and where it is used, the
fourth part contains explanatory notes for those definitions marked with
an asterisk, and the appendix is a complete and rigorous description of
the metalanguage. The second part is divided into three sections:
syntactic definitions of general nature, Level 1 syntax defining the
COBOL text and Level 2 syntax defining the COBOL program. The
Level 1 syntax describes the basic structure of the COBOL Language. It
defines a set of strings, called COBOL texts, in terms of generalized
words (including COBOL words, literals, arithmetic and relational
operators, etc.) and word separators.
The Level 2 syntax describes the detailed structure of the COBOL
Language. It defines a set of strings, called COBOL programs, in terms
of specific sequences of generalized words and word separators.
Although a COBOL text and a COBOL program have each been defined
as a string of characters, an attempt has been made to show the
relationship between such a string and the Reference Format.

The metalanguage used is an extension of the metalanguage used in the
ALGOL 60 Report, known as the Backus normal form. It is introduced
in the first part: "Introduction to the notation used" and described in
detail in the appendix under the title "Formalism for syntactical
definition".
Most extensions have been introduced to reduce the number and
complexity of production rules constituting the formal definition of the
COBOL syntax. For example certain extensions greatly simplify the
description of the nested structure of records. Whenever these extensions
are used, the usual Backus notation, based on Chomsky context-free
grammars (type 2), could have been used. However, the convention
adopted to show relationship between declaration of data-names and the
subsequent use of those data-names is different in that this relationship
could not be expressed in Backus notation. This is a well known context-
dependent aspect of programming Languages. English text has been
used where needed to adequately supplement the metalanguage.

It has been difficult to decide whether some COBOL rules should be
included in the syntax and somewhat arbitrary decisions had to be made.
The level of detail expressed in the production rules is also somewhat
arbitrary. It is often founded on an attempt to facilitate the use of this
formal definition by the human reader, in conjunction with the existing
descriptions of COBOL. For the same reason, the names of
metavariables have been chosen to reflect their meaning, and the names
defined in the draft ISO Recommendation on COBOL have been used
wherever feasible.

The application of the production rules given in level 2 syntax will
generate all valid COBOL programs. However, invalid programs will
also be generated. For example the following is not reflected:
- uniqueness of names
- relationship between qualifiers and the corresponding data hierarchy.
- relationships between subscripts or indices and the corresponding table
declarations
- some relationships between clauses and/or statements
- possible indentation of data description entries.
With the exceptions mentioned above, this formal definition is believed to be in agreement with the ISO Recommendation on COBOL.

However, the modular structure of the ISO Recommendation is not reflected; the syntax shown applies to the combination of the upper levels of all modules.