

CONFIRMED  
FEBRUARY 1992

# Method of defining syntactic metalanguage

UDC 681.3.06:800.92

NO COPYING WITHOUT BSI PERMISSION EXCEPT AS PERMITTED BY COPYRIGHT LAW

**BSI**  
Documents

AVA

Document Sharing Hub

# Cooperating organizations

The Office and Information Standards Committee, under whose direction this British Standard was prepared, consists of representatives from the following:

|                                                                                                                                                                 |                                                              |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------|
| British Computer Society Ltd.*                                                                                                                                  | HM Customs and Excise                                        |
| British Paper and Board Industry Federation (PIF)                                                                                                               | Her Majesty's Stationery Office                              |
| Business Equipment Trade Association*                                                                                                                           | Institute of Administrative Management                       |
| Central Computer and Telecommunications Agency                                                                                                                  | Institute of Cost and Management Accountants                 |
| Committee of London Clearing Bankers on behalf of the Committee of Scottish Clearing Bankers, Cooperative Bank, Central Trustee Savings Bank and Yorkshire Bank | Institute of Data Processing Management                      |
| Computer Services Association                                                                                                                                   | Institute of Purchasing and Supply                           |
| Department of Industry (Computers Systems and Electronics)*                                                                                                     | Inter-university Committee on Computing                      |
| Department of Industry (National Physical Laboratory)*                                                                                                          | Local Authorities Management Services and Computer Committee |
| Drawing Office Material Manufacturers' and Dealers' Association                                                                                                 | London Transport Executive                                   |
| Electricity Supply Industry in England and Wales*                                                                                                               | Ministry of Defence*                                         |
|                                                                                                                                                                 | National Computer Users' Forum                               |
|                                                                                                                                                                 | National Computing Centre Ltd.*                              |
|                                                                                                                                                                 | National Research Development Corporation                    |
|                                                                                                                                                                 | Office Machines and Equipment Federation                     |
|                                                                                                                                                                 | Post Office*                                                 |

The organizations marked with an asterisk in the above list, together with the following, were directly represented on the Technical Committee entrusted with the preparation of this British Standard:

|                                                                        |                                                           |
|------------------------------------------------------------------------|-----------------------------------------------------------|
| Association for Literary and Linguistic Computing                      | Control and Automation Manufacturers' Association (BEAMA) |
| Association of Computer Units in Colleges of Higher Education (ACUCHE) | Engineering Equipment Users' Association                  |
| British Gas Corporation                                                | Hatfield Polytechnic                                      |
| Central Computer Agency (Civil Service Department)                     | UK APL Users Group                                        |
| Computing Services Association                                         | University of Edinburgh                                   |
|                                                                        | University of London                                      |

This British Standard, having been prepared under the direction of the Office and Information Standards Committee, was published under the authority of the Executive Board and comes into effect on 31 December 1981

© BSI 12-1999

The following BSI references relate to the work on this standard:  
Committee reference OIS/5  
Draft for comment 79/64467 DC

ISBN 0 580 12530 0

## Amendments issued since publication

| Amd. No. | Date of issue | Comments |
|----------|---------------|----------|
|          |               |          |
|          |               |          |
|          |               |          |
|          |               |          |
|          |               |          |

# Contents

|                                                                      | Page               |
|----------------------------------------------------------------------|--------------------|
| Cooperating organizations                                            | Inside front cover |
| Foreword                                                             | ii                 |
| 0 Introduction                                                       | 1                  |
| 1 Scope                                                              | 3                  |
| 2 References                                                         | 3                  |
| 3 Definitions                                                        | 3                  |
| 4 Form of each syntactic element of the metalanguage                 | 4                  |
| 5 Symbols represented by each syntactic element                      | 5                  |
| 6 Layout and comments                                                | 7                  |
| 7 Representation of each terminal-character in the metalanguage      | 8                  |
| 8 Examples                                                           | 10                 |
| Appendix A Two-level grammars                                        | 17                 |
| Table 1 — Representation of terminal-characters                      | 9                  |
| Table 2 — Alternative representation of terminal-characters          | 9                  |
| Table 3 — Character pairs that represent a single terminal-character | 10                 |
| Table 4 — Invalid sequences of characters                            | 10                 |
| Publications referred to                                             | Inside back cover  |

## Foreword

This British Standard has been prepared under the direction of the Office and Information Standards Committee.

EDITORIAL NOTE. It is normal convention in British Standards to use italic type for algebraic quantities. Since the status of such quantities contained in this standard may or may not directly represent true variable quantities, this convention has not been adopted in this standard.

A British Standard does not purport to include all the necessary provisions of a contract. Users of British Standards are responsible for their correct application.

**Compliance with a British Standard does not of itself confer immunity from legal obligations.**

### Summary of pages

This document comprises a front cover, an inside front cover, pages i and ii, pages 1 to 18, an inside back cover and a back cover.

This standard has been updated (see copyright date) and may have had amendments incorporated. This will be indicated in the amendment table on the inside front cover.

## 0 Introduction

**0.1 General.** A syntactic metalanguage is an important tool of computer science. The concepts are well known, but many slightly different notations are in use. As a result syntactic metalanguages are still not widely used and understood, and the advantages of rigorous notations are unappreciated by many people. A metalanguage brings some order to the formal definition of a syntax and is useful not just for the definition of programming languages, but for many other formal definitions.

Since the definition of the programming language Algol 60<sup>1)</sup>, the custom has been to define the syntax of a programming language formally. Algol 60 was defined with a notation now known as BNF or Backus Naur Form. This notation has proved a suitable basis for subsequent languages, but has frequently been extended or slightly altered. The many different notations are confusing and have prevented the advantages of formal unambiguous definitions from being widely appreciated. The syntactic metalanguage described in this standard is based on Backus Naur Form and includes the most widely adopted extensions.

**0.2 Syntactic metalanguages.** A syntactic metalanguage is a notation for defining the syntax of a language by use of a number of rules. Each rule names part of the language (called a non-terminal symbol of the language) and then defines its possible forms. A terminal symbol of the language is an atom that cannot be split into smaller components of the language.

A syntactic metalanguage is useful whenever a clear formal description and definition is required, e.g. the format for references in papers submitted to a journal, or the instructions for performing a complicated task.

A formal syntax definition has three distinct uses:

- a) it names the various syntactic parts (i.e. non-terminal symbols) of the language;
- b) it shows which sequences of symbols are valid sentences of the language;
- c) it shows the syntactic structure of any sentence of the language.

**0.3 The need for a standard syntactic metalanguage.** Without a standard syntactic metalanguage every programming language starts by specifying the metalanguage used to define its syntax. This causes various problems:

- a) *Many different notations.* It is rare for two different programming languages to use the same metalanguage. Thus human readers are handicapped by having to learn a new metalanguage before they can study a new language.
- b) *Concepts not widely understood.* The lack of a standard notation hinders the use of rigorous unambiguous definitions.
- c) *Imperfect notations.* Because a metalanguage needs to be defined for every programming language, almost inevitably the metalanguage contains defects. For example, errors occurred in the drafting of BS 5904 and BS 5905 because the metalanguages could not be typed easily.
- d) *Special purpose notations.* A metalanguage defined for a particular programming language is often simplified by taking advantage of special features in the language to be defined. However, the metalanguage is then unsuitable for other programming languages.
- e) *Few general syntax processors.* The multiplicity of syntactic metalanguages has limited the availability of computer programs to analyse and process syntaxes, e.g. to list a syntax neatly, to make an index of the symbols used in the syntax, to produce a syntax-checker for programs written in the language.

In practice, experienced readers have little difficulty in picking up and learning a new notation, but even so the differences obscure mutual understanding and hinder communication. A standard metalanguage enables more people to crystallize vague ideas into an unambiguous definition. It is also useful because other people needing to provide formal definitions no longer need to reinvent similar concepts.

**0.4 The objectives to be satisfied.** It is desirable that a syntactic metalanguage should be:

- a) concise, so that languages can be defined briefly and thus be more easily understood;
- b) precise, so that the rules are unambiguous;
- c) formal, so that the rules can be parsed, or otherwise processed; by a computer when required;

<sup>1)</sup> NAUR, P.(Ed). Revised Report on the Algorithmic Language ALGOL 60, *Computer Journal*, January, 1963, Vol 5, No. 4 pp 349–367.

d) natural, so that the notation and format are relatively simple to learn and understand, even for those who are not themselves language designers; (The meaning of a symbol should not be surprising. It should also be possible to define the syntax of a language in a way that helps to indicate the meaning of the constructions.)

e) general, so that the notation is suitable for many purposes including the description of many different languages;

f) simple in its character set and with a notation that avoids, as far as is practicable, using characters that are not generally available on standard keyboards (both typewriters and computer terminals) so that the rules can be typed and can be processed by computer programs;

g) self-describing, so that the notation is able to describe itself;

h) linear, so that the syntax can be expressed as a single stream of characters read from left to right. (This simplifies printing a syntax. Computer processing of a syntax is also simpler.)

**0.5 Some common syntactic metalanguages.** Unfortunately none of the existing syntactic metalanguages was suitable for adoption as the standard, for example:

a) COBOL (ISO 1989) lists alternatives vertically and uses brackets spreading over many lines. This is inconvenient for computer processing and cannot be prepared on typewriters.

b) Backus Naur Form (used in ALGOL 60) has problems if the metasymbols ( $< > | ::=$ ) occur in the language being defined. Some common forms of construction (e.g. comments) cannot be expressed naturally, other constructions (e.g. repetition) are long-winded.

c) FORTRAN 77 (ISO 1539) has “railroad tracks”. These are easy to understand but difficult to prepare and to process on a computer or typewriter.

Most other languages use a variant of one of these metalanguages. Most of them cannot be candidates for standardization because they use characters not in the language being defined as metasymbols of the metalanguage. This simplifies the metalanguage, but prevents it from being used generally.

**0.6 The standard metalanguage.** The metalanguage defined in this standard is based on a suggestion by Wirth<sup>2)</sup> that is based on Backus Naur Form and that contains the most common extensions, i.e.:

a) Terminal symbols of the language are quoted so that any character, including one used in the metalanguage, can be defined as a terminal symbol of the language being defined.

b) [ and ] indicate optional symbols.

c) { and } indicate repetition.

d) Each rule has an explicit final character so that there is never any ambiguity about where a rule ends.

e) Brackets group items together. It is an obvious convenience to use ( and ) in their ordinary mathematical sense.

The main differences in the metalanguage defined in this standard are further features that experience has shown are often required when providing a formal definition.

1) *Defining an explicit number of items.* Fortran contains a rule that a label field contains exactly five characters; an identifier in PL/I or COBOL has up to 32 characters: rules such as these can be expressed only with difficulty in Backus Naur Form. In practice, such definitions are often left incomplete and the rules qualified informally in English.

2) *Defining something by specifying the few exceptional cases.* An Algol **end**-comment ends at the first **end**, **else** or semicolon. A rule like this cannot be expressed concisely or clearly in Backus Naur Form and is also usually specified informally in English.

3) *Including comments.* Programming languages and other structures with a complicated syntax need many rules to define them. The syntax will be clearer if explanations and cross-references can be provided; accordingly the standard metalanguage contains a comment facility so that ordinary text can be added to a syntax for the benefit of a human reader without affecting the formal meaning of the syntax.

---

<sup>2)</sup> WIRTH, N. What can we do about the unnecessary diversity of notation for syntactic definitions? *Comm ACM*, November 1977, Vol 20, No 11, pp 822–823.

4) *Meta-identifier*. A meta-identifier (the name of a non-terminal symbol in the language) need not be a single word or enclosed in brackets because there is an explicit concatenate symbol. This also ensures that the layout of a syntax (except in a terminal symbol) does not affect the language being defined.

5) *Extending the metalanguage*. A user may wish to extend the syntactic metalanguage. A special-sequence is provided for this purpose; the format and meaning are not defined in the standard except to ensure that the start and end of an extension can always be seen easily. Various possible extensions are outlined in the following paragraphs.

**0.7 Limitations and extensions.** The main limitation of this standard metalanguage is that the language being defined needs to be linear, i.e. the symbols in a sentence of the language can be placed in an ordered sequence. For example, knitting patterns and recipes in cooking are linear languages, but electric circuit diagrams are not.

A further limitation is that the notation is inadequate for defining more complex forms of grammars. Such facilities were not provided because it was thought the main need was to define a notation sufficient for the simpler and commoner requirements.

Instead the metalanguage has been designed so that various extensions can be made in a natural way. There are two simple ways of extending the standard metalanguage. First, the special-sequence concept provides a basic framework for any extension, the format between the special-sequence-characters being almost completely arbitrary. This method would be suitable for an action grammar, i.e. one specifying actions that are to take place as a sentence is parsed. Secondly, a meta-identifier can never be followed immediately by a left parenthesis in the standard metalanguage; thus another method of extending the metalanguage is to define the syntax and meaning of a meta-identifier followed by a sequence of parameters enclosed in parentheses. This would be reasonable in an attribute grammar where the rules ensure consistency between different parts of a sentence in the language being defined.

More complicated extensions are also possible. Appendix A suggests how the metalanguage might be extended to define a two-level grammar.

## 1 Scope

This British Standard defines a notation for specifying the syntax of a linear sequence of symbols. The standard defines both the logical structure of the notation and its graphical representation.

The metalanguage has applications in the definition of programming and other languages, as well as in other formal definitions, for example the commands to an operating system, or the precise format of data and results.

Examples of use of the metalanguage are given in clause 8.

**NOTE** Like many other notations, this standard metalanguage can still be misused; thus it does not prevent someone from trying to define an unparsable or ambiguous language.

## 2 References

The titles of the publications referred to in this standard are listed on the inside back cover.

## 3 Definitions

For the purposes of this British Standard, the following definitions apply in addition to those in BS 3527.

### 3.1

#### **sequence**

an ordered list of zero or more items

**NOTE** This definition differs from the one in BS 3527.

### 3.2

#### **subsequence**

a sequence within a sequence

### 3.3

#### **non-terminal symbol**

a syntactic part of the language being defined

**3.4****meta-identifier**

the name of a non-terminal symbol

**3.5****start symbol**

a non-terminal symbol that is defined by a syntax rule but does not occur in any other syntax rule

**3.6****sentence**

a sequence of symbols that represents the start symbol

**3.7****terminal symbol**

a sequence of one or more characters forming an irreducible element of a language

**NOTE** In this standard a terminal symbol of the metalanguage is called a terminal-character, and a terminal symbol of a language being defined by a syntax is represented by a terminal-string.

**4 Form of each syntactic element of the metalanguage**

**NOTE** The following conventions are used.

- a) Each meta-identifier of the metalanguage is written as one or more words joined together by hyphens;
- b) A meta-identifier ending with “-symbol” is the name of a terminal symbol of the metalanguage.

The normal character representing each operator of the metalanguage and its implied precedence is: (highest precedence at the top)

|   |                             |
|---|-----------------------------|
| * | repetition-symbol           |
| — | except-symbol               |
| , | concatenate-symbol          |
|   | definition-separator-symbol |
| = | defining-symbol             |
| ; | terminator-symbol           |

The normal precedence is overridden by the following bracket pairs:

|    |                         |                         |    |
|----|-------------------------|-------------------------|----|
| ‘  | first-quote-symbol      | first-quote symbol      | ’  |
| “  | second-quote-symbol     | second-quote-symbol     | ”  |
| (* | start-comment-symbol    | end-comment-symbol      | *) |
| (  | start-group-symbol      | end-group-symbol        | )  |
| [  | start-option-symbol     | end-option-symbol       | ]  |
| {  | start-repeat-symbol     | end-repeat-symbol       | }  |
| ?  | special-sequence-symbol | special-sequence-symbol | ?  |

**4.1 General.** The logical structure of the metalanguage is defined in **4.2** to **4.21**.

**4.2 Syntax.** The syntax of a language consists of one or more syntax-rules.

**4.3 Syntax-rule.** A syntax-rule consists of a meta-identifier (the name of the non-terminal symbol being defined) followed by a defining-symbol followed by a definitions-list followed by a terminator-symbol.

**4.4 Definitions-list.** A definitions-list consists of one or more single-definitions separated from each other by a definition-separation-symbol.

**4.5 Single-definition.** A single-definition consists of an ordered list of one or more syntactic-terms separated from each other by a concatenate-symbol.

**4.6 Syntactic-term.** A syntactic-term consists of either:

- a) a syntactic-factor, or
- b) a syntactic-factor followed by an except-symbol followed by a syntactic-exception.

**NOTE** If a syntactic-exception is permitted to be an arbitrary syntactic-factor, the metalanguage can define a wider class of languages than the context-free grammars, including attempts which lead to Russell-like paradoxes, e.g.

xx = “A” – xx;

Is “A” an example of xx? Such licence is undesirable and the form of a syntactic-exception is therefore restricted to cases that can be proved to be safe.