

JIS

JAPANESE INDUSTRIAL STANDARD

**Industrial robots — Programming
language SLIM**

JIS B 8439—1992

Translated and Published

by

Japanese Standards Association

Printed in Japan

In the event of any doubt arising,
the original Standard in Japanese is to be final authority.

JAPANESE INDUSTRIAL STANDARD

J I S

Industrial robots - Programming language SLIM

B 8439-1992

1. Scope

This Japanese Industrial Standard specifies the descriptive forms and functions, hereafter referred to as the "language specifications", of the programming language SLIM (Standard Language for Industrial Manipulators), hereafter referred to as SLIM, for enhancing the transplantability of motion programs between the various kinds of industrial robots, hereafter referred to as the "robots", to be used mainly for assembling works.

Remarks 1. The method of entry of the syntax specified in this Standard are specified in the Annex.

2. The applicable standards to this Standard are given in the following:

JIS X 0001 Glossary of terms used in information processing (Fundamental terms)

JIS X 0015 Glossary of terms used in information processing (Programming languages)

JIS X 0201 Code for information interchange

2. Definitions

For the purposes of this Standard, the following definitions apply, and others are in accordance with JIS X 0001 and JIS X 0015.

- (1) processing system definitions As regards the processing method and semantics of a program, the definitions specified in each processing system. In this case, the processing method and semantics of program shall have been codified concretely on each processing system.
- (2) extension To modify the language specification of SLIM in a range of not excluding a program having the language specification of SLIM and concurrency with standard.

Remarks: As regards the concurrency with the standard, it shall be specified in 4.

- (3) exceptional conditions The conditions when the processing system has recognized that a program can not be operated in accordance with the language specifications of SLIM or it is not in operation.

- (4) rounding The method to obtain the numerical value of low accuracy from the numerical value of high accuracy. In the case where to obtain an integer value from an actual value, the maximum integer value not exceeding a value added with 0.5 to the original value shall be considered to be the result.

3. Specified matters

The specified matters of this Standard shall be in accordance with the following:

- (1) The syntax of the program in accordance with the language specifications of SLIM.
- (2) The data type capable of being in/out put for a robot system which is controlled by the program written in SLIM, and the accuracies and range of numerical expressions.
- (3) The motions of the robot system being controlled by the program written in SLIM.
- (4) Matters related to the interpretation of the semantics of the program written in SLIM.
- (5) Errors and exceptional conditions to be detected, and the processing method of these.

4. Concurrency with standard to the SLIM

4.1 Concurrency with standard In this Standard, the following two concurrencies with standard shall be specified.

- (1) The concurrency with standard of the program written in SLIM.
- (2) The concurrency with standard of processing system which processes the program written in SLIM.

4.2 Concurrency with standard of program The program, when it satisfies the following conditions, has the concurrency with standard of the program.

- (1) Respective statements in the program shall be in accordance with the right syntax specified in this Standard.
- (2) Respective statements in the program shall have right semantics specified clearly in this Standard.
- (3) The string of statements in the program shall, as a whole, constitute the right program having right semantics specified clearly in this Standard.

4.3 Concurrency with standard of processing system The processing system, when it satisfies the following conditions, has the concurrency with standard of the processing system.

- (1) It shall be capable of processing accepting the program having the concurrency with standard of a program.

- (2) For the program which has no concurrency with standard of a program, it shall be capable of reporting that it can not accept this.
- (3) It shall be capable of interpreting and processing errors and exceptional conditions in accordance with those as specified in this Standard (See 4.4 and 4.5).
- (4) It shall be capable of interpreting the semantics of respective statements in the program having the concurrency with standard of program in accordance with those as specified in this Standard.
- (5) It shall be capable of interpreting the semantics of collectively statements in the program having the concurrency with standard of program in accordance with those as specified in this Standard.
- (6) As regards numerical values, it shall at least be capable of processing in the accuracy and range specified in this Standard.
- (7) The motions in respect to the functions as "not be specified" in this Standard and the functions taken as the "processing system definitions" shall have been defined clearly.

4.4 Processing of errors The processing system having the concurrency with standard of processing system may not necessarily report that there is the error on syntax or error on semantics in the program.

4.5 Processing of exceptional conditions The processing of exceptional conditions shall be in accordance with the following:

- (1) The processing system having the concurrency with standard of a processing system shall report all of the exceptional conditions specified in this Standard. However, in the case where a function to process exceptional conditions is provided, the exceptional conditions may not be reported.
- (2) The processing system having the concurrency with standard of a processing system, when the exceptional conditions have been detected, the execution of the program shall be halted. However, where exceptional processing procedure is specified, it shall be carried out in accordance with that procedure. However, even in the case where the prescription of this exceptional processing exists, in the case where the procedure of exceptional processing can not be employed due to the restrictions such as the hardware of a processing system and operational environment, the execution of the program shall be concluded.
- (3) In this Standard, the sequence of the detection of exceptional conditions and the sequence of processing shall not be specified.

4.6 Concurrency with standard of extension The processing system having the concurrency with standard of a processing system may accept the program written by the extended language specifications. In the case where this processing system accepts such a program, it is not necessary to report the part which is not in concurrence with the language specifications of SLIM. However, even in this case, when a statement and other elements are not in concurrence with the syntax rule specified in the language specifications of SLIM, the processing system shall either report errors or the syntax rule and semantics of the parts which are not in concurrence shall have been given in advance.

5. Programming elements

5.1 Characters The characters to be used in the SLIM shall be in accordance with Table 1..

Table 1. Sets of characters for SLIM

Letters	A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
Digits	0 1 2 3 4 5 6 7 8 9
Symbols	! " ' # \$ % & () * + - . , / : ; = < > ? @ ` [¥] ^ { } —
Space	Space characters
Underscore	_
Processing system definition characters	Characters of other than given above usable in processing system

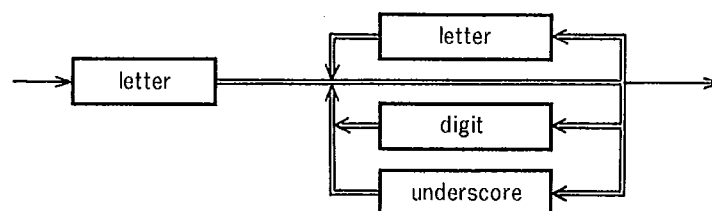
Remarks 1. The use of English small letters shall depend on the processing system.

2. For the coding scheme for information interchange, it should be preferable to use the coding scheme of 7 units or 8 units of JIS X 0201.

5.2 Identifiers

5.2.1 Syntax The syntax of identifier shall be in accordance with Fig. 1.

Fig. 1. Identifier



5.2.2 Semantics As regards the restriction of the number of characters of the identifier, it shall be as the processing system definition. However, the restriction of this number of characters shall be not less than 8 characters.

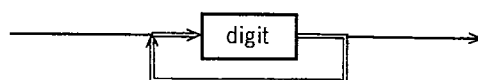
In addition, at least to the 8 characters in front of the identifier shall be identifiable.

6. Constants

6.1 Syntax The syntax of the constants shall be in accordance with (1) to (10) of the following:

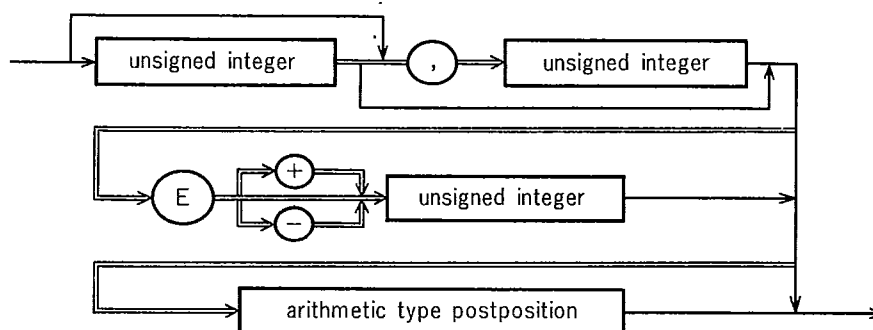
- (1) Unsigned integer The syntax of the unsigned integer shall be in accordance with Fig. 2.

Fig. 2. Unsigned integer



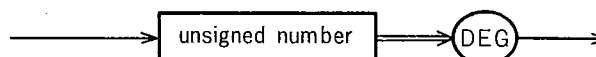
- (2) Unsigned number The syntax of the unsigned number shall be in accordance with Fig. 3.

Fig. 3. Unsigned number



- (3) Degree number The syntax of the degree number⁽¹⁾ shall be in accordance with Fig. 4.

Fig. 4. Degree number



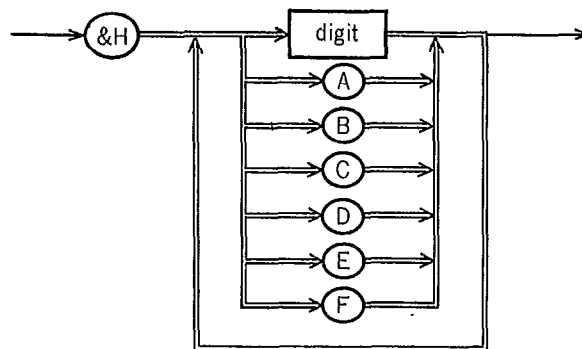
Note ⁽¹⁾ This shall be used when it is intended to express an angle not by the "radian" but by the "degree" (See Example).

Example: An example to indicate the sine of 90 degree
: SIN (90DEG)

Informative reference: 90DEG is that has inscribed $\frac{\pi}{2}$ radian by a degree expression, but it does not mean a constant 90.

- (4) Hexadecimal number The syntax of the hexadecimal number shall be in accordance with Fig. 5.

Fig. 5. Hexadecimal number

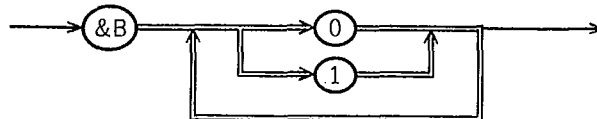


Example 1. The example where 16 of decimal number has been indicated by a hexadecimal number: &H10

Example 2. The example where 255 of decimal number has been indicated by a hexadecimal number: &HFF

- (5) Binary number The syntax of the binary number shall be in accordance with Fig. 6.

Fig. 6. Binary number

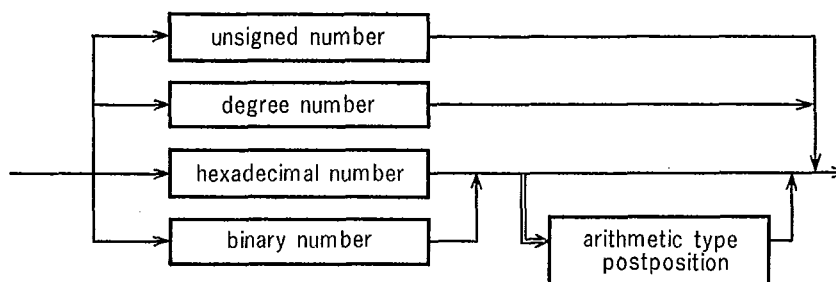


Example 1. The example where 4 of decimal number has been indicated by a binary number: &B0100

Example 2. The example where 15 of decimal number has been indicated by a binary number: &B1111

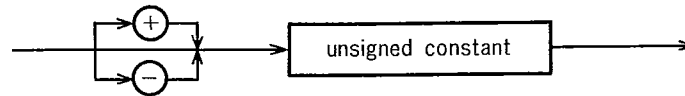
- (6) Unsigned constant The syntax of the unsigned constant shall be in accordance with Fig. 7.

Fig. 7. Unsigned constant



- (7) Constant The syntax of the constant shall be in accordance with Fig. 8.

Fig. 8. Constant



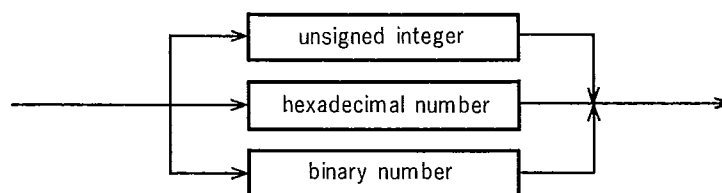
Example 1. -10.5E-10

Example 2. +90DEG

Example 3. -&HOF

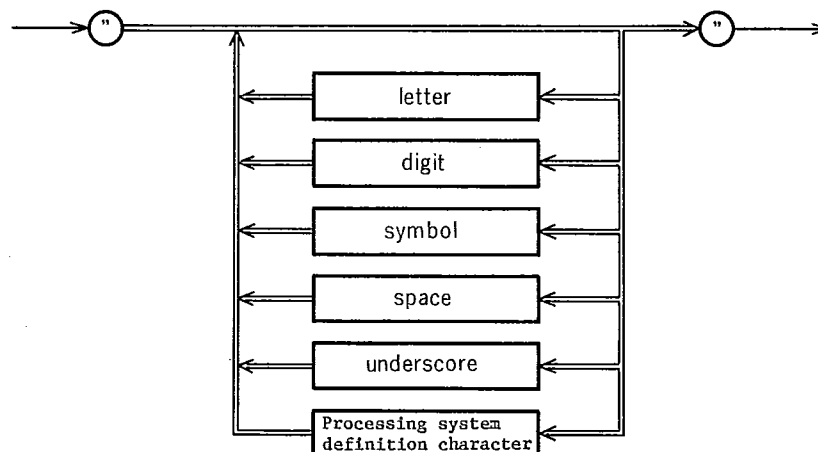
- (8) Positive constant The syntax of the positive constant shall be in accordance with Fig. 9.

Fig. 9. Positive constant



- (9) String The syntax of the string shall be in accordance with Fig. 10.

Fig. 10. String

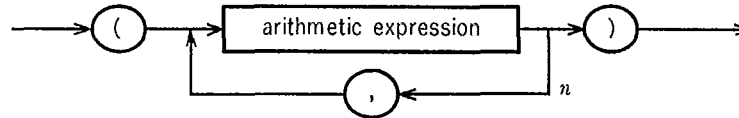


Remarks: When " is used as the element of a string, two " are followed in succession (See Example).

Example: An example indicating the string EXAMPLE OF "A" of characters: "EXAMPLE OF " "A""

- (10) Pose constant The syntax of a pose constant shall be in accordance with Fig. 11.

Fig. 11. Pose constant



6.2 Semantics The semantics of constants shall be in accordance with the following:

- (1) The constants of arithmetic type [6.1 (1) to (8)] shall take the numbers being expressed by the constant thereof as the values. The letter E of the unsigned integer shall indicate the power exponent of 10. When the sign immediately after E has been omitted, it shall be considered that the positive sign has been omitted. As regards the number of places of a significant figure, the processing system definitions apply. However, it shall be the accuracy of not less than 6 places of decimals.
- (2) The maximum value and minimum value of the permissible constant of each processing system shall be called as the machine maximum value and machine minimum value. As regards the actual numerical values of these, the processing system definitions apply.
- (3) The angle shall be expressed by radian. Where it is expressed by degree, the degree number of 6.1 (3) shall be used.
- (4) The value of string shall be the string of all characters interposed by symbol ". In this case, a space also has a meaning. The length of the string shall be the length of the string interposed by a symbol ". As regards the maximum length of string permitted by each processing system, the processing system definitions apply.
- (5) The pose constants shall be the constants expressing the position, posture and configuration of a robot. The arithmetic formula constituting the pose constants shall not include variables.

In addition, as regards the number of arithmetic formulas n and meanings, the processing system definitions apply.

6.3 Exceptional conditions The exceptional conditions of constants shall be in accordance with the following:

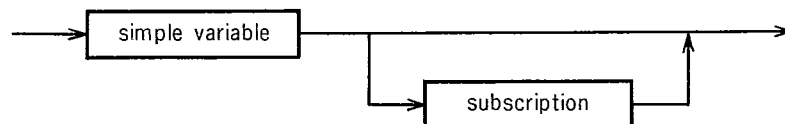
- (1) When a numerical value constant is outside the range of the positive machine maximum value or the negative machine minimum value. In this case, processing may be continued considering each to be the machine maximum value or machine minimum value.
- (2) When a string is exceeding its limit length.
- (3) When the number of element expressions of pose constant is not sufficient.

7. Variables

7.1 Syntax The syntax of variables shall be in accordance with (1) to (9) of the following:

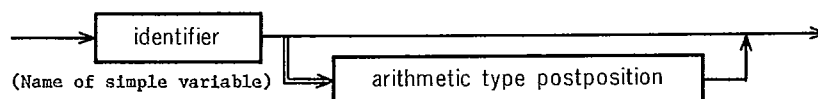
- (1) Arithmetic variable The syntax of the arithmetic variable shall be in accordance with Fig. 12.

Fig. 12. Arithmetic variable



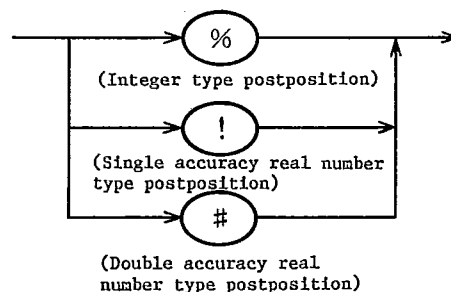
- (2) Simple variable The syntax of the simple variable shall be in accordance with Fig. 13.

Fig. 13. Simple variable



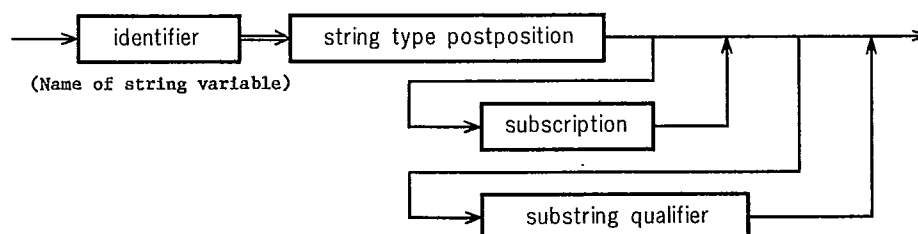
- (3) Arithmetic type postposition The syntax of the arithmetic type postposition shall be in accordance with Fig. 14.

Fig. 14. Arithmetic type postposition



- (4) String variable The syntax of the string variable shall be in accordance with Fig. 15.

Fig. 15. String variable



- (5) Substring qualifier The syntax of the substring qualifier shall be in accordance with Fig. 16.

Fig. 16. Substring qualifier



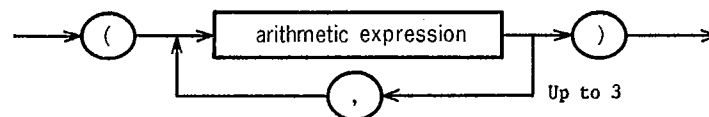
- (6) String type postposition The syntax of the string type postposition shall be in accordance with Fig. 17.

Fig. 17. String type postposition



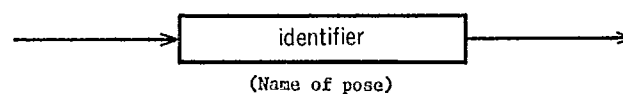
- (7) Subscription The syntax of the subscription shall be in accordance with Fig. 18.

Fig. 18. Subscription



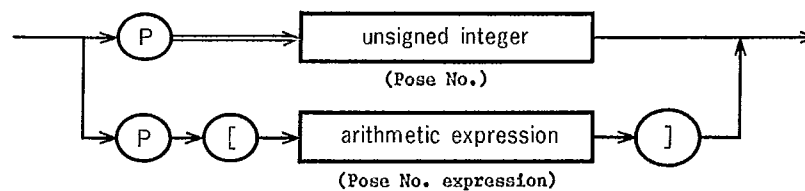
- (8) Pose variable The syntax of the pose variable shall be in accordance with Fig. 19.

Fig. 19. Pose variable



- (9) Numbered pose variable The syntax of the numbered pose variable shall be in accordance with Fig. 20.

Fig. 20. Numbered pose variable



7.2 Semantics The semantics of the variables shall be in accordance with the following:

Remarks: During execution of a program, an arithmetic variable is always connected with a numerical value, a string variable with a string value, and a pose variable and numbered pose variable with a pose value each. The value which has been connected with the variable may vary in positive or in negative according to the executions of the statement of the program.

- (1) The pose name shall have been declared by the pose variable declaration [See 10.1 (6)] in advance.
- (2) The lower limit of the pose No. shall be 1, and as to the upper limit the processing system definitions apply.
- (3) The numbered pose variable using the pose number expression shall be considered to be the numbered pose variable having the same pose number as its evaluation value. For example, in the case of P [I] where the value of I is 5, it shall be considered to be P5.
- (4) Such as P005, P05 and P5 shall express the identical numbered pose variables.
- (5) The variable name may be allowed to include a key word, however, the variable name of a key word only shall not be permissible.
- (6) Since the identifier begins with FN is considered to be a function name, it shall not be used as a variable name.
- (7) The variables given in Table 2 are built-in variables having particular meanings, and these shall not be used for other purposes.

Table 2. Built-in variables

Variable	Type	Meaning
DATE \$	String	It indicates the date in the form of YY/MM/DD.
TIME \$	String	It indicates the time in the form of HH:MM:SS.
TIMER	Integer	It indicates the time by seconds from the reference time. Remarks: As regards the reference time, the processing system definitions apply.

- (8) The identification of a variable shall be made only by an identifier. The type postposition shall not concern with the identification.
- (9) Each element of an array shall be referred to as the element of the array defined by the value of subscription expression. The subscription expression shall be rounded off to the integer nearest to the obtained value.

- (10) The length of the string being connected to a string variable can vary from 0 character to the maximum string length. As regards the maximum string length, the processing system definitions apply, however, it should preferably be 255 characters.
- (11) The substring qualifiers $a\$ (m:n)$ of string $a\$$ shall designate the substrings from the m -th character to the n -th character of string values connected with $a\$$. The characters in the string shall be numbered from the left to the right respectively in order from 1. If m or n does not come in the range of from 1 to $LEN(a\$)$, m is regarded as $MAX(m, 1)$ and n as $MIN[n, LEN(a\$)]$ (LEN , MAX and MIN see 16.).

Furthermore, if $m > n$, it shall be in accordance with the following:

- (a) In the case where $n < m < LEN(a\$)$, $a\$ (m:n)$ shall be the empty string at immediately before the m -th string.
- (b) In the case where $LEN(a\$) < m$, $a\$ (m:n)$ shall be the empty string at immediately after the $a\$$.
- (12) As regards the initial value which is connected with each variable at the time of start of execution, the processing system definitions apply.

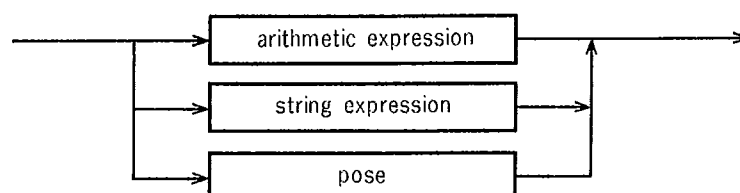
7.3 Exceptional conditions The exceptional conditions of variables shall be in accordance with the following:

- (1) When the value of subscription expression does not come in the range of the upper limit and lower limit of the array elements indicated clearly. In this case, continuation of a program shall not be performed.
- (2) When the postposition is in contradiction. This exceptional condition should preferably be detected at the time of input.
- (3) When the pose number does not come within a permissible range. This exceptional condition should preferably be detected at the time of input.
- (4) When the value of pose number expression does not come within a permissible range. In this case, the continuation of a program shall not be performed.

8. Expressions

8.1 Syntax of expression The syntax of the expressions shall be in accordance with Fig. 21.

Fig. 21. Expression

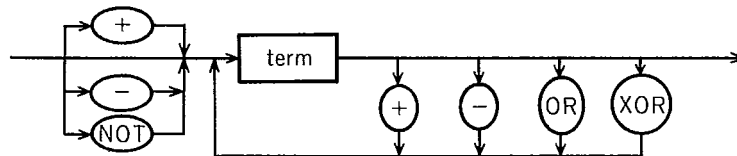


8.2 Arithmetic expression

8.2.1 Syntax The syntax of the arithmetic expression shall be in accordance with (1) to (6) of the following:

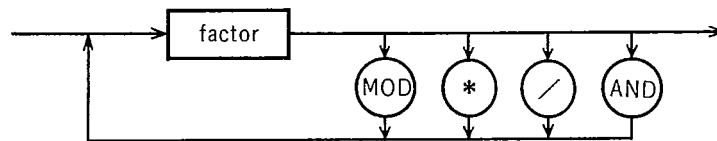
- (1) Arithmetic expression The syntax of the arithmetic expression shall be in accordance with Fig. 22.

Fig. 22. Arithmetic expression



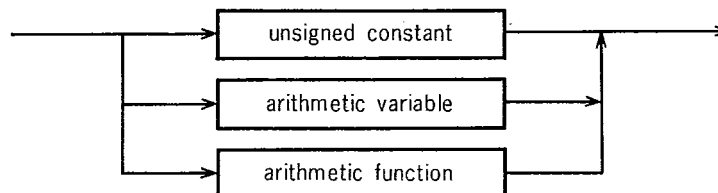
- (2) Term The syntax of the term shall be in accordance with Fig. 23.

Fig. 23. Term



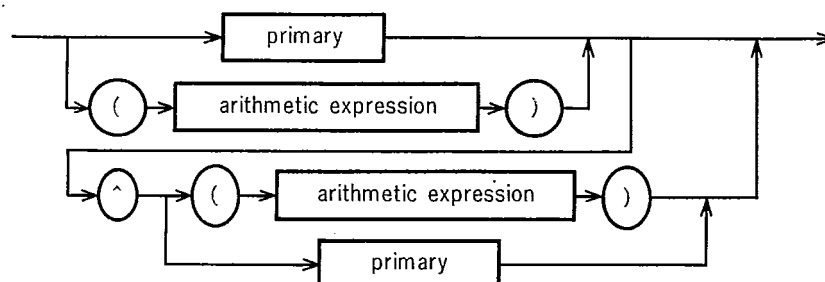
- (3) Primary The syntax of the primary shall be in accordance with Fig. 24.

Fig. 24. Primary



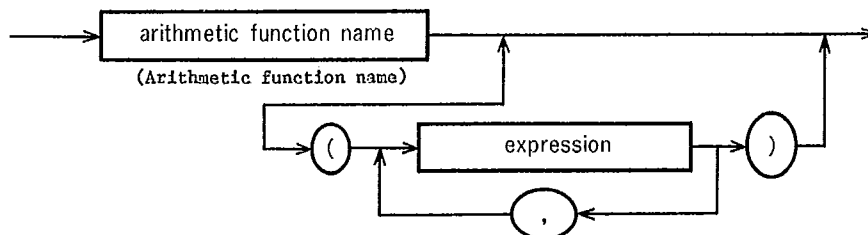
- (4) Factor The syntax of the factor shall be in accordance with Fig. 25.

Fig. 25. Factor



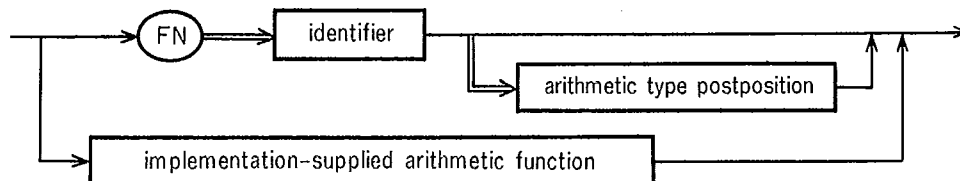
- (5) Arithmetic function The syntax of the arithmetic function shall be in accordance with Fig. 26.

Fig. 26. Arithmetic function



- (6) Arithmetic function name The syntax of the arithmetic function name shall be in accordance with Fig. 27.

Fig. 27. Arithmetic function name



8.2.2 Semantics The semantics of the arithmetic elements shall be in accordance with the following:

- (1) The functions other than implementation-supplied function shall have been defined in advance by DEF statement.
- (2) +, -, *, /, MOD and ^ shall be the arithmetic operators, and these perform the numeric operation indicated by the operator. Herein, ^ shall be the power multiple operator. NOT, AND, OR and XOR shall be the bit-operation operators, and these perform the operations of respective bits.
- (3) 0^0 shall be 1.
- (4) The accuracy of calculation shall be not less than 6 places of the decimal, and the maximum and minimum should preferably be not more than 10^{39} and not less than 10^{-39} .

8.2.3 Exceptional conditions The exceptional conditions of arithmetic expression shall be in accordance with the following:

- (1) When a division by 0 has been performed. In this case, it should be preferable to consider the machine maximum value having the sign of numeral to be divided to be the quotient.
- (2) When an overflow has generated in the evaluation of an expression. In this case, it should be preferable to consider the machine maximum value having positive sign to be the value of expression.

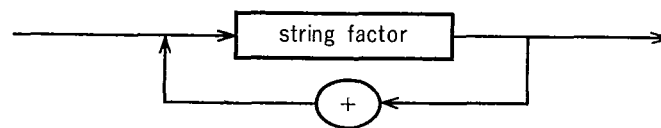
- (3) When the numerical value of negative has been power multiplied by a non integer. In this case, the continuation of a program shall not be performed.
- (4) When 0 has been power-exponent multiplied by the numerical value of negative. In this case, it should preferably be considered to be the machine maximum value of positive.
- (5) When an underflow has generated in the evaluation of an expression. In this case, it should be preferable to report that it is an exceptional condition and to continue a program considering the result to be 0.
- (6) When the numeral of the actual argument of function disagrees with the numeral of dummy argument. In this case, the continuation of a program shall not be performed.

8.3 String expressions

8.3.1 Syntax The syntax of the string expressions shall be in accordance with (1) to (5) of the following:

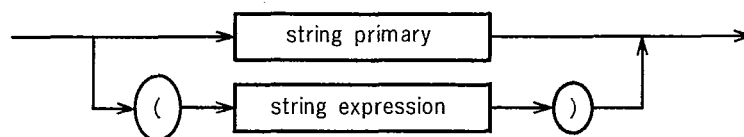
- (1) String expression The syntax of the string expression shall be in accordance with Fig. 28.

Fig. 28. String expression



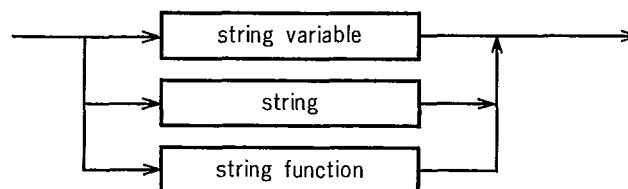
- (2) String factor The syntax of the string factor shall be in accordance with Fig. 29.

Fig. 29. String factor



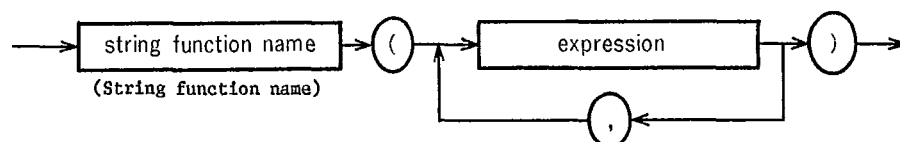
- (3) String primary The syntax of the string primary shall be in accordance with Fig. 30.

Fig. 30. String primary



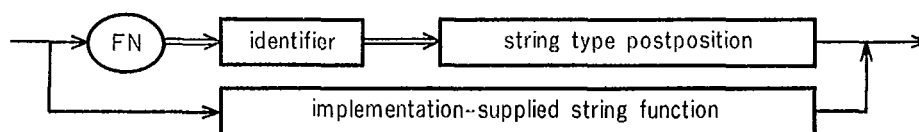
- (4) String function The syntax of the string function shall be in accordance with Fig. 31.

Fig. 31. String function



- (5) String function name The syntax of the string function name shall be in accordance with Fig. 32.

Fig. 32. String function name



8.3.2 Semantics The semantics of the string expressions shall be in accordance with the following:

- (1) The functions other than the implementation-supplied function shall have been defined by the function declaration specified in 10. in advance.
- (2) + shall be the operator which performs the concatenation operation of a string.

8.3.3 Exceptional condition The exceptional condition of string expression shall be as follows:

- (1) When the numeral of the actual argument of the string function disagrees with the numeral of the dummy argument. In this case, the continuation of a program shall not be performed.

8.4 Pose

8.4.1 Syntax The syntax of the pose shall be in accordance with (1) and (2) of the following:

- (1) Pose The syntax of the pose shall be in accordance with Fig. 33.

Fig. 33. Pose

