

JIS

JAPANESE INDUSTRIAL STANDARD

**NC Processor Input—Basic Part
Program Reference Language**

JIS B 6327—1985

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.

Contents

	Page
1. Scope	1
1.1 Application of Standard	1
1.2 Conformity to Standard	1
2. Definitions	1
3. Method for Specification of Language	2
4. Coordinate System	2
5. Characters, Statements and Part Program	2
5.1 Letters	2
5.2 Digits	2
5.3 Special Characters	2
5.4 Characters	3
5.5 Literal Character String	3
5.6 Unsigned Number	4
5.7 Keywords	4
5.8 Simple Identifiers	4
5.9 Identifiers	5
5.10 Labels	5
5.11 Statements	5
5.12 Nesting	6
5.13 Part Program	6
6. Arithmetic Statement	6
6.1 Semantics and Syntax	6
6.2 Operators	8
6.3 Scalar Functions	8
6.3.1 Algebraic Functions	9
6.3.2 Trigonometric Functions	9
6.3.3 Exponential Functions	10
6.3.4 Vector Functions	11
6.3.5 Miscellaneous Functions	11
7. Program Definition Statements	12
7.1 Semantic and Syntax	12
7.2 Synonym Declaration Statement	13
7.3 Reservation Declaration Statement	14
7.4 Macrodefinition Statement and Macrotermination Statement	14

B 6327-1985

	Page
8. Execution Control of Part Program	16
8.1 Start and End of Part Program	16
8.1.1 Part Program Identification Statement	16
8.1.2 Part Program Termination Statement	16
8.2 Program Control Statement	16
8.2.1 Semantics and Syntax	16
8.2.2 Postprocessing Statement	17
8.2.3 No-postprocessing Statement	17
8.2.4 Print Statement	17
8.2.5 Punch Statement	19
8.2.6 Read Statement	19
8.2.7 Cutter Location Print Statement	20
8.2.8 Titles Statement	20
8.2.9 Loop Start and End Statements	21
8.2.10 Unconditional Transfer Statement	22
8.2.11 Conditional Transfer Statement	22
8.2.12 Geometric Transfer Statement	23
8.2.13 Copy Statement and Index Specification	24
8.2.14 Macroexecution Statement	27
8.2.15 Remarks Statement	28
9. Geometrical Definition Statement	28
9.1 Semantics and Syntax	28
9.2 Local Reference System	32
9.2.1 Semantics and Syntax	32
9.2.2 Declaration of Local Reference System	32
9.2.3 End of Local Reference System	32
9.3 Declaration of Z Surface	32
9.3.1 Semantics and Syntax	32
9.3.2 Declaration of Z-surface by Its Height from the XY Plane	33
9.3.3 Declaration of Z Surface by Plane	33
9.3.4 Declaration of Z Surface by Cylinder	34
9.3.5 Declaration of Z Surface by Sphere	35
9.3.6 Declaration of Z Surface by Cone	35
9.3.7 Declaration of Z Surface by Quadric	36
9.4 Point Definition Statement	37
9.4.1 Semantics and Syntax	37
9.4.2 Definition of a Point by Its Rectangular Coordinates	38

	Page
9.4.3 Definition of a Point as Intersection of Two Lines	38
9.4.4 Definition of a Point as Intersection of a Line and a Circle	39
9.4.5 Definition of a Point as Intersection of Two Circles	40
9.4.6 Definition of a Point by its Polar Coordinates	40
9.4.7 Definition of a Point on a Circle at an Angle with the X Axis	41
9.4.8 Definition of a Point in a Pattern	42
9.4.9 Definition of a Point as the Center of a Circle	42
9.4.10 Definition of a Point as the Intersection of a Line and a Circle	43
9.4.11 Definition of a Point as the Intersection of a Line and Tabulated Cylinder	43
9.4.12 Definition of a Point as the Intersections of Three Planes	44
9.4.13 Definition of a Point as Increments from a Defined Point ...	45
9.4.14 Definition of a Point at a Radial Distance and at an Angle from a Previously Defined Point	45
9.4.15 Definition of a Point on a Line and with a Given X or Y Coordinate Value	46
9.4.16 Definition of a Point at a Given Distance along a Tabulated Cylinder from a Point on the Tabulated Cylinder	47
9.5 Pattern Definition Statement	47
9.5.1 Semantics and Syntax	47
9.5.2 Definition of a Pattern by the Starting and Ending Points and the Total Number of Points in the Pattern	49
9.5.3 Definition of a Pattern by the Starting Point, a Vector, and the Total Number of Points in the Pattern	49
9.5.4 Definition of a Pattern by the Starting Point, a Vector, and the Total Number of Spacing and Increments	50
9.5.5 Definition of a Pattern by the Starting Point, Angle, and the Combination of the Total Number of Spacing and Increments	51
9.5.6 Definition of a Circular Arc Pattern by a Circle, the Starting and Ending Angles, Direction and the Total Number of Points	51
9.5.7 Definition of a Circular Arc Pattern by a Circle, an Angle, Direction and Total Number of Equally Spaced Points along the Circle	52

B 6327-1985

	Page
9.5.8 Definition of a Circular Arc Pattern by a Circle, an Angle, Direction and the Combination of Total Number of Spacing and Increments of Angle	53
9.5.9 Definition of a Grid Pattern by Two Patterns	54
9.5.10 Definition of a Grid Pattern by a Pattern, a Vector and the Total Number	53
9.5.11 Definition of a Grid Pattern by a Pattern, a Vector and the Combination of Total Number of Spacing and Increments	56
9.5.12 Definition of a Grid Pattern by Two Patterns	58
9.5.13 Definition of a Pattern by Two Patterns	59
9.5.14 Definition of a Pattern by Points and Patterns	60
9.5.15 Definition of a Mirroring Pattern to Line	62
9.6 Line Definition Statement	63
9.6.1 Semantics and Syntax	63
9.6.2 Definition of a Line Passing Two Points	64
9.6.3 Definition of a Line as a Coordinate Axis	64
9.6.4 Definition of a Line by a Point and a Parallel Line	65
9.6.5 Definition of a Line by a Point and a Perpendicular Line	65
9.6.6 Definition of a Line by a Parallel Line at a Given Distance	66
9.6.7 Definition of a Line by a Point and the Angle between This Line and Another One	67
9.6.8 Definition of a Line by a Point and Its Slope Relative to a Line	67
9.6.9 Definition of a Line by Its Angle with X axis and Its Intersection with the Named Coordinate Axis	68
9.6.10 Definition of a Line by the Slope of Its Angle with the X axis and Its Intersection with the Named Coordinate Axis	69
9.6.11 Definition of a Line by a Point and as a Tangent to a Circle	69
9.6.12 Definition of a Line as a Tangent to Two Circles	70
9.6.13 Definition of a Line by Its Angle with a Line and as a Tangent to a Circle	71
9.6.14 Definition of a Line as an Intersection of Two Planes	71
9.6.15 Definition of a Line by a Point and as a Tangent to a Conic	72
9.6.16 Definition of a Line by a Point Lying on a Conic, and Perpendicular to the Conic at This Point	73

	Page
9.6.17 Definition of a Line by a Point, and Perpendicular to a Conic	73
9.6.18 Definition of a Line by a Point and as a Tangent to a Tabulated Cylinder	74
9.6.19 Definition of a Line by a Point and Perpendicular to a Tabulated Cylinder	75
9.7 Plane Definition Statement	75
9.7.1 Semantics and Syntax	75
9.7.2 Definition of a Plane by the Coefficients of the Plane Equation	76
9.7.3 Definition of a Plane Passing through Three Points	77
9.7.4 Definition of a Plane Passing through a Point and Parallel to a Given Plane	77
9.7.5 Defintiion of a Plane Parallel to a Given Plane at a Given Distance	78
9.7.6 Definition of a Plane Passing through a Point and Perpendicular to a Vector	79
9.7.7 Definition of a Plane Passing through Two Points and Perpendicualr to a Given Plane	80
9.7.8 Defintiion of a Plane Passing through a Point and Perpendicular to Two Planes	80
9.7.9 Defintiion of a Plane Passing through a Point and Tangential to a Cylinder	81
9.8 Vector Definition Statement	82
9.8.1 Semantics and Syntax	82
9.8.2 Definition of a Vector by Its Components	83
9.8.3 Definition of a Vector by Two Points	83
9.8.4 Definition of a Unit Vector Perpendicular to a Given Plane	84
9.8.5 Definition of a Vector by a Scalar Times a Vector	85
9.8.6 Definition of a Vector by the Cross Product of Two Given Vectors	86
9.8.7 Definition of a Unit Vector by Normalizing a Given Vector	86
9.8.8 Definition of a Vector by a Length and a Given Angle in a Given Plane	87
9.8.9 Definition of a Unit Vector Parallel to the Intersection of Two Given Planes	88
9.8.10 Definition of a Vector by the Addition or Subtraction of Two Given Vectors	89
9.8.11 Definition of a Unit Vector in the XY-Plant Making a Specified Angle with a Given Line	90

B 6327-1985

	Page
9.9 Circle Definition Statement	90
9.9.1 Semantics and Syntax	90
9.9.2 Definition of a Circle by the Coordinates of the Center and Its Radius	92
9.9.3 Definition of a Circle by Its Center Point and Its Radius	93
9.9.4 Definition of a Circle by Its Center Point and a Tangential Line	93
9.9.5 Definition of a Circle by Its Center Point and a Point on the Circumference	94
9.9.6 Definition of a Circle by Three Points on the Circumference	94
9.9.7 Definition of a Circle by Its Center Point and a Tangential Circle	95
9.9.8 Definition of a Circle by Two Tangential Lines and Its Radius	96
9.9.9 Definition of a Circle by a Tangential Line, a Point on the Circumference and Its Radius	96
9.9.10 Definition of a Circle Tangential to Two Circles and Having a Given Radius	97
9.9.11 Definition of a Circle by a Tangential Line, a Tangential Circle and Its Radius	98
9.9.12 Definition of a Circle by a Tangential Line, a Tangential Tabulated Cylinder and Its Radius	99
9.9.13 Definition of a Circle by Three Tangential Lines	99
9.9.14 Definition of a Circle by Two Points on Its Circumference and Its Radius	100
9.9.15 Definition of a Circle by a Point on Its Circumference, a Tangential Circle and Its Radius	101
9.9.16 Definition of a Circle by a Tangential Line, a Tangential Conic and Its Radius	102
9.9.17 Definition of a Circle by Two Tangential Tabulated Cylinders and Its Radius	102
9.9.18 Definition of a Circle by a Tangential Circle, a Tangential Tabulated Cylinder and Its Radius	103
9.9.19 Definition of a Circle by a Point on Its Circumference, a Tangential Tabulated Cylinder and Its Radius	104
9.9.20 Definition of a Circle by a Line through Its Center, a Tangential Tabulated Cylinder and Its Radius	105
9.9.21 Definition of a Circle by the Coordinates of Its Center, the Direction Cosines of the Axis of Its Right Vertical Cylinder and Its Radius	106
9.10 Cylinder Definition Statement	106
9.10.1 Semantics and Syntax	106

	Page
9.10.2 Definition of a Cylinder by a Point, a Vector, and Its Radius	107
9.10.3 Definition of a Cylinder by Two Tangential Planes and Its Radius	108
9.10.4 Definition of a Cylinder by a Tangential Plane, a Tangential Cylinder and Its Radius	109
9.10.5 Definition of a Cylinder by Two Tangential Cylinders and Its Radius	110
9.11 Sphere Definition Statement	111
9.11.1 Semantics and Syntax	111
9.11.2 Definition of a Sphere by the Coordinates of Its Center and Its Radius	112
9.11.3 Definition of a Sphere by Its Center Point and Its Radius	112
9.11.4 Definition of a Sphere by its Center Point and a Point on the Sphere	113
9.11.5 Definition of a Sphere by Its Center Point and a Tangential Plane	113
9.11.6 Definition of a Sphere by Four Points on the Sphere	114
9.12 Cone Definition Statement	114
9.12.1 Semantics and Syntax	114
9.12.2 Definition of a Cone by Its Vertex, Axis Vector and Half-angle	115
9.12.3 Definition of a Cone by the Coordinates of Its Vertex, the Vector Components of Its Axis and the Cosine or Its Half-angle	115
9.13 Ellipse Definition Statement	116
9.13.1 Semantics and Syntax	116
9.13.2 Definition of an Ellipse by Its Center Point, Length of Its Semimajor Axis, Length of Its Semiminor Axis and the Angle Made by the Major axis with the X axis	117
9.14 Hyperbola Definition Statement	117
9.14.1 Semantics and Syntax	117
9.14.2 Definition of a Hyperbola by the Center, the Length of Half the Transverse Axis, the Length of Half the Conjugate Axis and the Angle between the Transverse Axis and the X-axis of the Coordinate System	118
9.15 Lofted Conic Definition Statement	119
9.15.1 Semantics and Syntax	119
9.15.2 Definition of a Lofted Conic by Five Points	119
9.15.3 Definition of a Lofted Conic by Four Points and One Slope	120

	Page
9.15.4 Definition of a Lofted Conic by Three Points and Two Slopes	120
9.16 General Conic Definition Statement	121
9.16.1 Semantics and Syntax	121
9.16.2 Definition of a Conic by the Conic Equation Represented as an Implicit Function	122
9.16.3 Definition of a Conic by the Conic Equation Represented as the Explicit Function of X	122
9.16.4 Definition of a Conic by the Conic Equation Represented as the Explicit Function of Y	123
9.17 Quadric Definition Statement	123
9.17.1 Semantics and Syntax	123
9.17.2 Definition of a Quadric Surface by the Quadric Equation	124
9.18 Tabulated Cylinder Definition Statement	128
9.18.1 Semantics and Syntax	128
9.18.2 Definition of a Tabulated Cylinder by a Series of Points	129
9.18.3 Definition of a Tabulated Cylinder by Polar Coordinates of a Series of Points	130
9.18.4 Definition of a Tabulated Cylinder by a Series of Points and a Vector	131
9.18.5 Definition of a Tabulated Cylinder at a Distance from a Given Parallel Tabulated Cylinder	132
9.19 Matrix Definition Statement	133
9.19.1 Semantics and Syntax	133
9.19.2 Definition of a Matrix by Twelve Parameters	134
9.19.3 Definition of a Matrix by Defining the Translation of the Old Coordinate System into the New System	134
9.19.4 Definition of a Matrix by Defining the Rotation in the Referenced Plane of the Old Coordinate System into the New System	135
9.19.5 Definition of a Matrix by Defining the Scaling of the Old Coordinate System into the New System	136
9.19.6 Definition of a Matrix by Defining a Symmetry with Respect to a Line or a Plane	136
9.19.7 Definition of a Matrix by Defining a Symmetry with Respect to the Referenced Plane	137
9.19.8 Definition of a Matrix by Three Planes	138
9.19.9 Definition of a Matrix by a Point and Two Vectors	138
9.19.10 Definition of a Matrix as the Product of Two Matrices	139
9.19.11 Definition of a Matrix as the Inverse of a Given Matrix..	140

	Page
9.20 Ruled Surface Definition Statement	140
9.20.1 Semantics and Syntax	140
9.20.2 Definition of a Ruled Surface by Two Curved Surfaces and Two Planes Given Either by Three Points or by Two Points and a Vector	140
9.20.3 Definition of a Ruled Surface by a Curved Surface, the Plane Given Either by Three Points or by Two Points and One Vector, and a Point.....	141
10. Geometric Execution Statement	142
10.1 Semantics and Syntax	142
10.2 Motion Statement	143
10.2.1 Semantics and Syntax	143
10.2.2 Cutter Location Initialization Statement	144
10.2.3 Point-to-point Motion Absolute and Multiple Point- to-point Statement	145
10.2.4 Point-to-point Motion Relative	147
10.2.5 Initial Continuous Motion Statement	148
10.2.6 Special Initial Continuous Motion Statement	149
10.2.7 Intermediate Continuous Motion Statement and Terminal Continuous Motion Statement	150
10.3 Tool Path Control Statement	151
10.3.1 Semantics and Syntax	151
10.3.2 Tool-to-drive Surface Orientation	152
10.3.3 Modal Tool-to-part Surface Orientation	153
10.3.4 Thick Specification	153
10.3.5 Tolerance Selector	154
10.3.6 Cutter Specification	155
10.3.7 Part Surface Specification	156
10.4 Startup Direction Control Statement	156
10.4.1 Semantics and Syntax	156
10.4.2 Startup Direction Control Statement by Points	157
10.4.3 Startup Direction Control Statement by Vectors	157
10.4.4 Startup Direction Control Statement by Surface Vectors	158
10.5 Transform Cutter Location Statement	159
10.6 Processor Output File Control Statement	160

B 6327-1985

	Page
Appendix 1. Detailed Semantics of Continuous Motion	162
1. Classification of Continuous Motion Statements	162
2. Initial Continuous Motion Statement	162
2.1 Semantics	162
2.2 One Surface Startup, Non-Directed, with Effective Part Surface	162
2.3 One Surface Startup, Non-Directed, without Effective Part Surface	163
2.4 One Surface Startup, Non-Directed, Independent of Part Surface	164
2.5 One Surface Startup, Directed, with Effective Part Surface	164
2.6 One Surface Startup, Directed, Independent of Part Surface	165
2.7 Two Surface Startup, Non-Directed	166
2.8 Two Surface Startup, Directed	166
2.9 Three Surface Startup, Non-Directed	167
2.10 Three Surface Startup, Directed	167
3. Intermediate Continuous Motion Statement and Terminal Continuous Motion Statement	167
3.1 Semantics	167
3.2 Tolerance	168
3.3 Direction of Motion	169
3.4 Omission of Check Surface	170
3.5 Multiple Intersection	172
3.6 Multiple Check Surfaces	172
Appendix 2. Rules for Representing NC Processor Input — Basic Part Program Reference Language on Medium	174
1. Description of Statement	174
2. Continuation of Column	174
3. Multiple Description of Statements	174
4. Remarks	174
5. Spaces	174
Appendix 3. Rules for Representing General Form of NC Processor Input — Basic Part Program Reference Language	175
1. General Form	175
2. Keywords	175
3. Specifications	175
4. Variables	175

	Page
5. Surface Specifiers	175
6. Braces	175
7. Brackets	175
8. Omission of Element	175
9. Repetition	175
10. Yen Symbol	176
11. Omission of Left Side of Equal Mark	176
Appendix 4. Syntax Description of the Reference Language	177
1. Syntax Description Conventions	177
2. Metalinguistic Symbols	177
3. Metavariable Names	177
4. Spaces	177
5. Linguistic Symbols	177
6. Production Rules	178
7. Metavariables Omitting Construction	178
Appendix 5. Keywords List	179
Appendix 6. Recommended Synonyms List	189
Appendix 7. Recommended Standard Forms	190
1. Standard Form	190
2. Graphics	190

JAPANESE INDUSTRIAL STANDARD

J I S

NC Processor Input — Basic Part
Program Reference Language

B 6327-1985

1. Scope

1.1 Application of Standard This Japanese Industrial Standard specifies the language to produce an NC processor input — basic part program. The NC processor processes the part program, prepares CLDATA containing the tool path information of a standard state not depending on a specific NC device, the command information to various functions of machine, etc., and outputs the information for NC depending on a specific NC device from this CLDATA.

1.2 Conformity to Standard If a part program is written by the language specified herein and has the interpretation specified herein, the part program conforms to this standard. If an NC processor executes the process conforming to the interpretation specified herein to prepare CLDATA, the NC processor conforms to the Standard. The NC processor is not required to be able to process all the languages contained in this Standard. That is, though the NC processor can process a part of the languages contained in this Standard, the NC processor thereof conforms to the standard.

Further, if a NC processor is not contradictory to the type and the interpretation of the language specified herein, though the extension of the type and the interpretation are contained, and otherwise though the NC processor processes the language not contained in this standard, the NC processor thereof conforms to the Standard.

2. Definitions

The definitions of principal terms used in this Standard shall be as specified in JIS B 0181, JIS X 0001, JIS B 6325, JIS B 6326, and JIS B 6310.

Applicable Standards:

JIS B 0181-Vocabulary for Numerically Controlled Machine Tools

JIS B 6310-Axis and Motion Nomenclature for Numerically Controlled
Machine Tools

JIS B 6325-Numerical Control Processor Output-CLDATA

JIS B 6326-Numerical Control Processor Output-Minor Elements of
2000-Type RecordsJIS X 0001-Glossary of Terms Used in Information Processing
(Fundamental Terms)

Corresponding International Standard:

ISO/DIS 4342-Numerical control of machines - NC processor input -
Basic part program reference language

3. Method for Specification of Language

This language is specified by using definitions, the general form described in the form to be used for part program (refer to Appendix 3) and syntax by metalanguage form (refer to Appendix 4).

4. Coordinate System

The coordinate system used for this language shall be as follows:

- (1) The coordinate system is classified into reference coordinate system and local coordinate system.
- (2) The reference coordinate system is the right-handed rectangular Cartesian coordinate system fixed on the workpiece.
- (3) The reference axis is of the positive direction of each coordinate axis of reference coordinate system, which is respectively called reference axis X, reference axis Y and reference axis Z.
- (4) The local coordinate system is defined in 9.2.
- (5) The shape of a workpiece and the motion of a tool to the workpiece are programmed by using the language specified herein on those two coordinate systems.

5. Characters, Statements and Part Program

5.1 Letters (alphabet)

5.1.1 Semantics Letters have no individual meaning, being used for forming keywords, simple identifiers, character strings or labels.

5.1.2 Syntax The syntax is as follows:

$$\langle \text{letter} \rangle ::= A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | \\ T | U | V | W | X | Y | Z$$

5.2 Digits

5.2.1 Semantics Digits have no individual meaning, being used for forming simple identifiers, unfigned numbers, character strings or labels.

5.2.2 Syntax The syntax is as follows:

$$\langle \text{digit} \rangle ::= 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9$$

5.3 Special Characters

5.3.1 Semantics The special characters shall be as follows:

- + positive sign
- negative sign
- * asterisk
- / slant

-) closing parenthesis
- (opening parenthesis
- . period
- = equal sign
- , comma
- ¥ yen sign
- \$ dollar sign
- space
- ' apostrophe
- : colon
- ; semicolon

5.3.2 Syntax The syntax is as follows:

<special character> ::= + | - | * | / |) | (| . | = | , | ¥ | \$ | Δ | ' | : | ;

Remark: The sign of Δ represents the space of special character.

5.4 Characters

5.4.1 Semantics A character is a letter, digit or special character or other valid character.

5.4.2 Syntax The syntax is as follows:

<character> ::= <letter> | <digit> | <special character>

5.5 Literal Character String

5.5.1 Semantics The literal character string is the string of character describing the title and the note in a title statement and a note statement (refer to 8.2.8 and 8.2.15).

- Remarks
1. The apostrophe is used at the beginning and at the end of a literal character string to indicate the extent of the literal character string.
 2. The space in a literal character string has the semantic as one character.

5.5.2 Syntax The syntax is as follows:

<literal character string> ::= <symbol for literal delimiter>

^k₀ [<character>] <symbol for literal delimiter>

<symbol for literal delimiter> ::= ' '

5.6 Unsigned Number

5.6.1 Semantics The unsigned number is the decimal real number not having positive or negative signs. In the number not including a decimal point, it is assumed to appear after the rightmost digit.

Remark: The number is of a real number when used internally and no distinctions are made between integer and real numbers.

Example: Examples of Unsigned Numbers

47711

58.

3.14

5.6.2 Syntax The syntax is as follows:

$$\langle \text{unsigned number} \rangle ::= \overset{k}{1} [\langle \text{digit} \rangle]_0^1 [\cdot] | \overset{k}{0} [\langle \text{digit} \rangle] . \overset{n}{1} [\langle \text{digit} \rangle]$$

5.7 Keywords

5.7.1 Semantics Keywords are divided into the major keywords and the minor keywords and they are the words having fixed semantics. The major keyword is the word to represent the principal semantics of the statement. When the semantics of the statement are not thoroughly represented by the major keyword alone, the major keyword is followed by a slant/and thereafter, joined by the minor keyword as additional information. The minor keyword is the word representing the semantics of operation, conditions, the display of state in the minor part. The keywords are given in Appendix 5 and Appendix 3 of JIS B 6326. The keywords can be replaced by the synonyms preliminarily defined (refer to 7.2 and Appendix 6).

5.8 Simple Identifiers

5.8.1 Semantics Simple identifiers have no inherent meaning and are used as specifiers [refer to 9.1.1 (3)], labels, variable names (refer to 6.1.1) and macronames (refer to 8.4.1).

Example: Examples of Simple Identifiers

P

L1

BOBBY

5.8.2 Syntax The syntax is as follows:

$$\langle \text{simple identifier} \rangle ::= \langle \text{letter} \rangle \overset{5}{0} [\langle \text{letter} \rangle | \langle \text{digit} \rangle]$$

5.9 Identifiers

5.9.1 Semantics An identifier is a simple identifier or the simple identifier carrying a subscript and has no inherent meanings. The identifier with subscript is used as a specifier and a variable name.

Remark: The simple identifier carrying a subscript is called the identifier with subscript.

Example: Examples of Identifiers

P 1

P 2 (5)

5.9.2 Syntax The syntax is as follows:

$\langle \text{identifier} \rangle ::= \langle \text{simple identifier} \rangle \mid [(\langle \text{scalar} \rangle)]$

5.10 Labels

5.10.1 Semantics A part programmer may attach a label to a statement for statement identification. The label may be referred to in a conditional or unconditional transfer statement or a geometric transfer statement.

Remark: A label always pairs with a closing parenthesis or the label carrying a colon to be used (refer to 5.11.2).

Example: Examples of Labels

TRANTO/A1

...

...

A1) GOTO/P1

5.10.2 Syntax The syntax is as follows:

$\langle \text{label} \rangle ::= {}^6_1[\langle \text{digit} \rangle] \mid \langle \text{simple identifier} \rangle$

5.11 Statements

5.11.1 Semantics A statement is a complete instruction or unit of information. The statement is classified into 6 classes of arithmetic statement, program definition statement, program control statement, geometrical definition statement, geometrical execution statement, and postprocessor statement. Statements may be labeled.

Remark: The postprocessor statement is as specified in JIS B 6326.

Example: Examples of Statements

A = ABS (Y)

FINI

P1 = POINT/INTOF, L1, L2

5.11.2 Syntax The syntax is as follows:

$\langle \text{statement} \rangle ::= {}_0^1 [\langle \text{label} \rangle {}_1^1 [] | :] \langle \text{unlabeled statement} \rangle$
 $\langle \text{unlabeled statement} \rangle ::= \langle \text{arithmetic statement} \rangle | \langle \text{program definition statement} \rangle | \langle \text{program control statement} \rangle | \langle \text{geometrical definition statement} \rangle | \langle \text{geometrical execution statement} \rangle | \langle \text{post-processor statement} \rangle$

5.12 Nesting

5.12.1 Semantics A nesting is the geometrical definition statement other than the declaration of local coordinate system and the declaration of Z plane in the geometrical definition statement or the right side of its equal sign written between the opening parenthesis and the closing parenthesis (refer to 9.1.2). The geometrical definition statement of nesting may be referred to by its identifier thereafter.

Example: Examples of Nesting

L2 = LINE/(PT4 = POINT/3, 6), PT1

L3 = LINE/PT4, (POINT/5, 5)

5.13 Part Program

5.13.1 Semantics A part program starts from a part program identification statement, is followed by a logically complete and ordered sequence of statement and completed by a part program termination statement (refer to 8.1).

5.13.2 Syntax The syntax is as follows:

$\langle \text{part program} \rangle ::= \langle \text{part program identification statement} \rangle$
 ${}_0^k [\langle \text{statement} \rangle] \langle \text{part program termination statement} \rangle$

6. Arithmetic Statements

6.1 Semantics and Syntax

6.1.1 Semantics The arithmetic statement is composed of identifier, equal sign and arithmetic expression following them, and the value of arithmetic expression is substituted for the identifier. This name is called the variable name. The arithmetic operation in arithmetic expression is composed of adding operation, multiplying operation and exponential operation.

- (1) Operator The operator indicates the kinds of operations among the organized elements of arithmetic expression. As the signs representing operation, + for adding operator, - for subtraction operator, * for multiplication operator, / for division operator, and ** for exponential operator are used.

- (2) The Hierarchy of Operational Priorities The hierarchy of operational priorities is as follows:

<u>Priority</u>	<u>Operator</u>
1	()
2	* *
3	*, /
4	+, -

For instance, in the arithmetic statement of $A = 2 + 3 * 4$, the operator $*$ is prior to the operator $+$ and the value of A becomes 14. Further, in the arithmetic statement of $A = (2 + 3) * 4$, the operation in brackets has priority and the value of A becomes 20.

- (3) Substitution of Value The equal sign has the meaning that the value of the arithmetic expression on its right side is substituted for the name on its left side. Therefore, $A = A + 1$ means that the value obtained by adding 1 to A becomes the new value of A .
- (4) Scalar Functions The scalar function used in arithmetic expression is composed of algebraic function, trigonometrical function, exponential function, vector function, and miscellaneous function. They are called scalar functions.
- (5) Citation of Scalar Amount The scalar amount of arithmetic statement may be cited by a variable name. The variable name thereof is the name of the left side of the equal sign in arithmetic statement.

6.1.2 Syntax The syntax is as follows:

$\langle \text{adding operator} \rangle ::= + \mid -$
 $\langle \text{multiplying operator} \rangle ::= * \mid /$
 $\langle \text{exponential operator} \rangle ::= **$
 $\langle \text{primary} \rangle ::= \langle \text{unsigned number} \rangle \mid \langle \text{arithmetic symbol} \rangle \mid \langle \text{scalar function} \rangle \mid (\langle \text{arithmetic expression} \rangle)$
 $\langle \text{factor} \rangle ::= \langle \text{primary} \rangle \mid [\langle \text{exponential operator} \rangle \langle \text{primary} \rangle]$
 $\langle \text{term} \rangle ::= \langle \text{factor} \rangle \mid [\langle \text{multiplying operator} \rangle \langle \text{factor} \rangle]$
 $\langle \text{arithmetic expression} \rangle ::= \langle \text{term} \rangle \mid [\langle \text{adding operator} \rangle \langle \text{term} \rangle]$
 $\langle \text{arithmetic expression} \rangle \langle \text{adding operator} \rangle \langle \text{term} \rangle$
 $\langle \text{scalar} \rangle ::= \langle \text{primary} \rangle \mid [\langle \text{adding operator} \rangle \langle \text{primary} \rangle] \mid (\langle \text{arithmetic symbol} \rangle \langle \text{arithmetic expression} \rangle)$

$\langle \text{arithmetic statement} \rangle ::= \langle \text{arithmetic symbol} \rangle = \langle \text{scalar} \rangle$
 $\langle \text{algebraic function} \rangle ::= \frac{1}{2} [\text{ABS} \mid \text{SQRT}] (\langle \text{scalar} \rangle)$
 $\langle \text{trigonometric function} \rangle ::= \frac{1}{2} [\text{SIN} \mid \text{COS} \mid \text{TAN} \mid \text{ASIN} \mid \text{ACOS} \mid \text{ATAN}]$
 $(\langle \text{scalar} \rangle)$
 $\langle \text{exponential function} \rangle ::= \frac{1}{2} [\text{EXP} \mid \text{LOG}] (\langle \text{scalar} \rangle)$
 $\langle \text{vector function} \rangle ::= \text{LNTH}(\langle \text{vector specifier} \rangle) \mid \text{DOT}(\langle \text{vector specifier} \rangle, \langle \text{vector specifier} \rangle)$
 $\langle \text{miscellaneous functions} \rangle ::= \text{ANGL}(\langle \text{circle specifier} \rangle,$
 $\langle \text{point specifier} \rangle) \mid \text{DIST}(\langle \text{point specifier} \rangle, \langle \text{point specifier} \rangle) \mid \text{NUM}(\langle \text{pattern specifier} \rangle)$
 $\langle \text{scalar function} \rangle ::= \langle \text{algebraic function} \rangle \mid \langle \text{trigonometric function} \rangle \mid$
 $\langle \text{exponential function} \rangle \mid \langle \text{vector function} \rangle \mid \langle \text{miscellaneous function} \rangle$

6.2 Operators

6.2.1 Semantics The semantics of operator is as follows:

- (1) Adding Operator The adding operator is composed of a unary operator and a binary operator.
 - (a) Unary Operator A term is placed after the operator.
 - + The sign of the following term is not changed.
 - The sign of the following term is changed.
 - (b) Binary Operator An arithmetic expression is placed before the operator and a term is placed after it.
 - + A term is added to the arithmetic expression.
 - A term is subtracted from the arithmetic expression.
- (2) Multiplying Operator A term is placed before the multiplying operator and a factor is placed after it.
 - * The item is multiplied by a factor.
 - / The item is divided by a factor.

Remarks 1. $A = 5 (3 + 2)$ shall be rewritten by $A = 5 * (3 + 2)$

2. $A = \frac{1 + 2 + 3}{4}$ shall be rewritten by $A =$
 $(1 + 2 + 3) / 4.$

- (3) Exponential Operator A primary is placed before the exponential operator and a primary is also placed after it.

* * The preceding primary is raised to a power given by the following primary.

Example: Examples of Operators

$$A = B + C$$

$$A = B * C$$

$$A = B ** C$$

$$A = (B + C) / D$$

$$A = (B - C) * D$$

6.3 Scalar Functions

6.3.1 Algebraic Functions

6.3.1.1 Semantics The algebraic function designator denotes the type of arithmetic process to be performed using the following bracketed arithmetic expression.

ABS The absolute value of the arithmetic expression is to be taken.

SQRT The nonnegative square root of the arithmetic expression is to be taken.

Remark: The value of arithmetic expression shall be nonnegative real number.

Example: Examples of Algebraic Function

$$A = \text{ABS} (-2)$$

$$A = \text{SQRT} (9)$$

6.3.2 Trigonometric Functions

6.3.2.1 Semantics The trigonometric function designation denotes the type of trigonometric process to be performed on the following bracketed arithmetic expression.

SIN The sine of the following arithmetic expression is to be taken.

COS The cosine of the following arithmetic expression is to be taken.

TAN The tangent of the following arithmetic expression is to be taken.

ASIN The angle whose sine is given by the following arithmetic expression is to be taken.

ACOS The angle whose cosine is given by the following arithmetic expression is to be taken.

ATAN The angle whose tangent is given by the following arithmetic expression is to be taken.

Remark: The values of arithmetic expression processed by direct functions SIN, COS, and TAN are given by angles in degree. The values of arithmetic expression processed by reverse functions ASIN, ACOS, and ATAN are respectively given by $[-1, 1]$, $[-1, 1]$, and $[-\infty, \infty]$ and the values obtained as the processed results are angles in degree respectively within ranges of $[-90, 90]$, $[0, 180]$, and $[-90, 90]$.

Example: Examples of Trigonometric Functions

A = SIN (30)

A = COS (30)

A = TAN (45)

A = ASIN (0.5)

A = ASIN (-0.5)

A = ACOS (0.866)

A = ACOS (-0.866)

A = ATAN (1)

A = ATAN (-1)

6.3.3 Exponential Functions

6.3.3.1 Semantics The exponential function designator denotes the type of exponential process to be performed on the following bracketed arithmetic expression.

EXP The value of e to a power given by the value of the following arithmetic expression is taken.

LOG The natural logarithm of the following arithmetic expression is taken.

Remark: The value of arithmetic expression as the object of LOG shall be positive real number.

Example: Examples of Exponential Functions

A = EXP (2)

A = LOG (2)

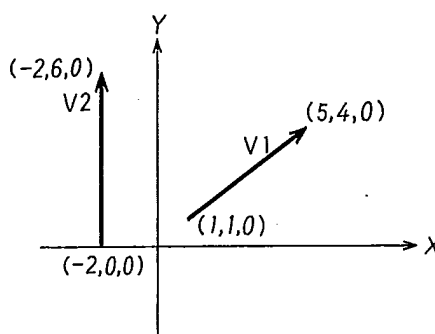
6.3.4 Vector Functions

6.3.4.1 Semantics The vector function designator denotes the type of vector process to be performed on the following bracketed vector.

LNTH The length of the bracketed vector is taken.

DOT The inner product of bracketed two vectors is taken.

Example: Examples of Vector Functions



V1 = VECTOR/1, 1, 0, 5, 4, 0

V2 = VECTOR/ -2, 0, 0, -2, 6, 0

A1 = LNTH (V1)

A2 = LNTH (V2)

A3 = DOT (V1, V2)

6.3.5 Miscellaneous Functions

6.3.5.1 Semantics The miscellaneous function designator denotes the type of process to be performed on the following bracketed circle, point, or pattern.

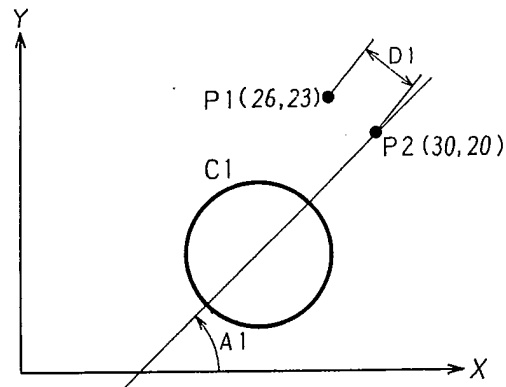
ANGL The angle in a unit of degree counterclockwise measured between the positive X axis and the line joining the center of the specified circle with the specified point is to be taken.

DIST The distance between the specified two points is to be taken.

NUM The number of points in the specified pattern is to be taken.

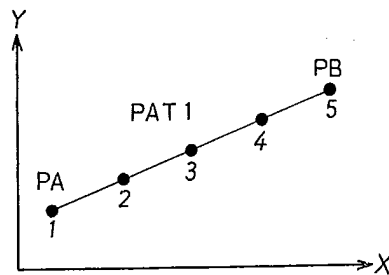
12
B 6327-1985

Exmaple: Examples of Miscellaneous Functions



A1 = ANGL (C1, P2)

D1 = DIST (P1, P2)



PAT1 = PATTERN/LINEAR, PA, PB, 5

N1 = NUM (PAT 1)

7. Program Definition Statements

7.1 Semantics and Syntax

7.1.1 Semantics Program definition statements preliminarily define the synonyms of keywords, the storage region of data, and a sequence of statements used repeatedly.

7.1.2 Syntax The syntax is as follows:

<program definition statement> ::= <synonym declaration> | <reserve declaration> | <macroblock>

7.2 Synonym Declaration Statement

7.2.1 Semantics A synonym declaration statement defines an alternative identifier giving the same semantics as the keyword as a synonym. The General Form of the synonym-keyword pairs is as follows:

SYN/synonym 1, keyword 1, synonym 2, keyword 2 . . . , synonym n, keyword n

Remarks 1. A synonym is a simple identifier.

2. Synonym 1, . . . , synonym n respectively have the same meanings as keyword 1, . . . , keyword n.
3. Keywords are given in Appendix 5 and Appendix 3 of JIS B 6326.
4. For specific keywords among keywords, the synonyms given in Appendix 6 are recommended.
5. On and after a synonym declaration statement, synonyms receive the same process as keywords.
6. The simple identifier once declared as a synonym can not be redefined in any other statement.

Example: Examples of Synonym Declaration Statements

SYN/P, POINT, C, CIRCLE, TT, TANTO, L, LINE

This declaration allows the following statement.

L1 = L/RIGHT, TT, C1, RIGHT, TT, C2

which is taken as

L1 = LINE/RIGHT, TANTO, C1, RIGHT, TANTO, C2

The followign type of statement is not allowed because P was defined as synonym.

P = POINT/0, 1, 2

7.2.2 Syntax The syntax is as follows:

<synonym parameter list>::= <simple identifier>,< keyword>^k₀[, <simple idnetifier>,< keyword>]

<synonym declaration>::= SYN/<synonym parameter list>

7.3 Reservation Declaration Statement

7.3.1 Semantics A reservation declaration statement reserves the data region having a suitable size by the name and the maximum number of subscripts. The General Form is as follows:

RESERVE/simple identifier 1, maximum value 1, simple identifier 2, maximum value 2 . . . , simple identifier n, maximum value n.

- Remarks 1. Simple identifier 1, simple identifier 2, . . . , simple identifier n are the identifiers with subscript.
2. Maximum value 1, maximum value 2, . . . , maximum value n are the maximum number of subscripts.
3. A reservation declaration statement shall be specified before the identifier with subscript is used (refer to 5.9.1).

Example: Example of Reservation Declaration Statement

RESERVE/P1, 10, P2, 7, PAT1, 12

By this statement, simple identifiers P1, P2, and PAT 1 become identifiers with subscript and the maximum number of subscripts become respectively 10, 7, and 12.

Therefore, this allows the statement

P1 (8) = POINT/INTOF, L1, L2

but would not allow

P2 (8) = POINT/INTOF, L1, L2.

7.3.2 Syntax The syntax is as follows:

<reserve parameter list>:: = <simple identifier>, <unsigned numeral>

k_0 [, <simple identifier>, <unsigned numeral> | <reserve declaration> ::=

RESERVE/<reserve parameter list>

7.4 Macrodefinition Statement and Macrotermination Statement

7.4.1 Semantics A macrodefinition statement and a macrotermination statement, to use several times a group of the sequence of statements in a part program, instruct to make the group of the sequence of statements thereof be a macroregion.

The General Form is as follows:

- (1) Macroidentifier MACRO [/parameter 1 [= normal value 1], parameter 2 [= normal value 2], . . . ¥ parameter n [normal value n]]

- (2) TERMAC

- Remarks 1. The start of macrosequence of part program statements is the MACRO statement of General Form (1).
2. The end of macrosequence of part program statements is TERMAC of General Form (2).
3. A macroidentifier is the identifier set to the macro.
4. Parameter 1, parameter 2, ..., parameter n are simple identifiers.
5. Normal value 1, normal value 2, ..., normal value n are the values preliminarily given to the parameters.
6. When a macrodefinition is called by a macroexecution statement, an effective value is given to the parameter not given a normal value (refer to 8.2.14).
7. PARTNO, FINI, LOOPST, LOOPND and other macrodefinition statements are not written within the macrosequence of part program statements.

Example: Examples of Macrodefinition Statement and Macrotermination Statement

```
MAC1 = MACRO
GO
GOTO/P1
GODLTA/0.5
TERMAC
MAC3 = MACRO/A, B = -15
GOTO/A
GODLTA/B
GODLTA/-B
TERMAC
```

7.4.2 Syntax The syntax is as follows:

```
<normal macroitem>:: = <keyword> | <simple identifier> | <unsigned
number> | (<arithmetic expression>)
<macroitem list>:: = <simple identifier> | <simple identifier> = <normal
macroitem> | <macroitem list>, <simple identifier> | <macroitem list>, <simple
identifier> = <normal macroitem>
```

<macroassignment statement> :: = <simple identifier> = MACRO $\frac{1}{0}$ [/ <macro-item list>]

<macrotermination statement> :: = TERMAC

<macroblock> :: = <macroassignment statement> $\frac{k}{0}$ [<statement> | <macro-termination statement>]

8. Execution Control of Part Program

8.1 Start and End of Part Program

8.1.1 Part Program Identification Statement

8.1.1.1 Semantics The part program identification statement gives the character string to identify the part program and denotes the start of part program.

The General Form is as follows:

PARTNO/'character string'

Example: Example of Part Program Identification Statement

PARTNO/'VALVE HOUSING, DRAWING NO 72445'

8.1.1.2 Syntax The syntax is as follows:

<part program identification statement> :: = PARTNO/ <character string>

8.1.2 Part Program Termination Statement

8.1.2.1 Semantics The part program termination statement defines the end of part program.

The General Form is as follows:

FINI

8.1.2.2 Syntax The syntax is as follows:

<part program termination statement> :: = FINI

8.2 Program Control Statement

8.2.1 Semantics and Syntax

8.2.1.1 Semantics The program control statement controls the input and output of data and the flow of program.

8.2.1.2 Syntax The syntax is as follows:

<program control statement> :: = <postprocessing statement> | <no-postprocessing statement> | <print statement> | <punch statement> | <read statement> | <cutter location print statement> | <titles statement> | <loop start statement> | <loop end statement> | <unconditional transfer statement> | <conditional transfer

statement>|<geometric transfer statement>|<copy statement>|<index specification>|<index specification>|<macroexecution statement>|<remark statement>

8.2.2 Postprocessing Statement

8.2.2.1 Semantics The postprocessing statement calls post processor. The General Form is as follows:

MACHIN/identifier [, numeral] [, { keyword 1 } , ... , { keyword n }]

Remarks 1. Keywords are given in Appendix 5 and Appendix 3 of JIS B 6326.

2. When no-postprocessing statement exists, the postprocessor is not called in spite of the presence of postprocessing statement.
3. As many postprocessing statements as possible may be written to call the postprocessor in the part program.

Example: Example of Postprocessing Statement

MACHIN/EXPOST, 0

8.2.2.2 Syntax The syntax is as follows:

<postprocessing statement> :: = MACHIN/<postprocessor parameter list>

<postprocessor parameter list>] $\begin{smallmatrix} k \\ 0 \end{smallmatrix}$ [, <keyword> | , <scalar >]

8.2.3 No-postprocessing Statement

8.2.3.1 Semantics The no-postprocessing statement prohibits the calling of postprocessor. Even if a postprocessing statement is specified, the calling is neglected.

The General Form is as follows:

NOPOST

8.2.3.2 Syntax The syntax is as follows:

<no-postprocessing statement>:: = NOPOST

8.2.4 Print Statement

8.2.4.1 Semantics The print statement instructs the print of the normal form of graphic definition statement [refer to 9.1.1 (2)] and of the value of variable of arithmetic statement.

The General Form is as follows:

(1) PRINT/ { ON
OFF }

(2) PRINT/ { 0
PAGE }

- (3) PRINT/ { 2, identifier 1, ..., identifier n
NONAME, identifier 1, ..., identifier n }
- (4) PRINT/ { 3, identifier 1, ..., identifier n
identifier 1, ..., identifier n }
- (5) PRINT/ { 3, ALL
ALL }

- Remarks 1. PRINT/ON causes the printing of normal form graphic definition statement just thereafter and the printing of the value of the variable of arithmetic statement just thereafter to be started on and after this statement.
2. PRINT/OFF causes the printing to be stopped on and thereafter.
3. PRINT/0 and PAGE cause the page to be ejected.
4. PRINT/2, identifier 1, ..., identifier n and NONAME. identifier 1, ..., identifier n cause the normal form of the graphic and the value of the variable of specified identifier to be printed without their identification.
5. PRINT/3, identifier 1, ..., identifier n and identifier 1, ..., identifier n cause the normal form of the graphic and the value of the variable of specified identifier to be printed with their identification.
6. PRINT/3, ALL and ALL cause the normal form of and the value of the variable of all graphics to be printed with their identification.

Example: Example of PRINT STATEMENT

PRINT/3, L1, C2, ANGL4, PT2

8.2.4.2 Syntax The syntax is as follows:

<print statement>:: = PRINT/<print parameter list>

<print parameter list>:: = $\frac{1}{2}$ [ON|OFF]|

$\frac{1}{2}$ [0|PAGE]|

$\frac{1}{6}$ [NONAME | 2 | 3], <identifier> $\frac{k}{0}$ [, <identifier>]|

$\frac{1}{6}$ [3,] ALL

8.2.5 Punch Statement

8.2.5.1 Semantics The punch statement causes the normal form of the graphic of graphic definition statement and the value of the variable of arithmetic statement to be punched.

The General Form is as follows:

- (1) PUNCH/ $\left\{ \begin{array}{l} \text{ON} \\ \text{OFF} \end{array} \right\}$
- (2) PUNCH/ $\left\{ \begin{array}{l} 1, \text{identifier } 1, \dots, \text{identifier } n \\ \text{identifier } 1, \dots \text{identifier } n \end{array} \right\}$
- (3) PUNCH/ $\left\{ \begin{array}{l} 1, \text{ALL} \\ \text{ALL} \end{array} \right\}$

Remarks 1. PUNCH/ON causes the punch of the normal form of graphic definition statement just thereafter and the punch of the value of the variable of arithmetic statement just thereafter to be started on and after this statement.

2. PUNCH/OFF causes the punch to be stopped thereafter.

3. PUNCH/1, identifier 1, ..., identifier n and identifier 1, ..., identifier n cause the normal form of the graphic of and the value of the variable of specified identifier to be punched.

4. PUNCH/1, ALL and ALL cause the normal form of all graphics and the value of variable to be punched.

Example: Example of Punch Statement

PUNCH/1, A, B, C

8.2.5.2 Syntax The syntax is as follows:

<punch statement>:: = PUNCH/<punch parameter list>

<punch parameter list> :: = $\frac{1}{i}$ [ON|OFF]|

$\frac{1}{i}$ [1,] <identifier> $\frac{k}{0}$ [, <identifier>] |

$\frac{1}{0}$ [1,] ALL

8.2.6 Read Statement

8.2.6.1 Semantics The read statement instructs the reading of the normal form of the graphic of punched graphic definition statement and the reading of the value of the variable of arithmetic statement.

The General Form is as follows:

- (1) READ/ $\left\{ \begin{array}{l} 1 \\ \text{ALL} \end{array} \right\}$
- (2) READ/1, identifier 1, ..., identifier n

- Remarks 1. READ/1 instructs the reading.
2. READ/ALL causes the normal form and the value of the variables of all the punched graphics to be read together with their identifiers
 3. READ/1, identifier 1, ..., identifier n assigns new identifiers to the standard form and the value of the variable of all the punched graphics to be read.

8.2.6.2 Syntax The syntax is as follows:

<read statement> :: = READ/<read parameter list>

<read parameter list> :: = $\frac{1}{2}$ [1 | ALL] |
 1, <identifier> $\frac{k}{0}$ [, <identifier>]

8.2.7 Cutter Location Print Statement

8.2.7.1 Semantics The cutter location print statement instructs the printing of CLDATA.

The General Form is as follows:

CLPRNT $\left[/ \left\{ \begin{array}{l} \text{ON} \\ \text{OFF} \\ \text{OMIT, } n \end{array} \right\} \right]$

- Remarks 1. CLPRNT calls for a print of all cutter location data.
2. CLPRNT/ON starts the printing of cutter location data.
 3. CLPRNT/OFF stops the printing of cutter location data.
 4. CLPRNT/OMIT, n prints the first point, then points $n + 1$, $2n + 1$, $3n + 1$, ..., and the last point.

8.2.7.2 Syntax The syntax is as follows:

<cutter location print statement> :: = CLPRNT $\frac{1}{0}$ [/ <clprint parameter list>]

<clprint parameter list> :: = $\frac{1}{2}$ [ON | OFF | OMIT, <scalar>]

8.2.8 Titles Statement

8.2.8.1 Semantics The titles statement instructs the labeling of the printed output.

The General Form is as follows:

TITLES/'character string'

Immediately after executing this statement, the character string is printed.

Example: Examples of Title Print Statements

PRINT/PAGE

TITLES/'...'

PRINT/3, S1, S2, S3, S4

8.2.8.2 Syntax The syntax is as follows:

< titles statement > :: = TITLES/< literal character string>

8.2.9 Loop Start and End Statements

8.2.9.1 Semantics The loop start statement and the loop end statement define the bounds of the group of statements which is a loop.

The General Form is as follows:

(1) LOOPST

(2) LOOPND

Within a loop, the normal sequential execution of the part program may be altered by transfer statements.

Example: Examples of Loop Start and End Statements

J = 1

LOOPST

A) . . .

. . .

1F (J - 100) B, C, C

B) J = J + 1

JUMPTO/A

C) LOOPND

8.2.9.2 Syntax The syntax is as follows:

< start of a loop > :: = LOOPST

< end of a loop > :: = LOOPND

8.2.10 Unconditional Transfer Statement

8.2.10.1 Semantics The unconditional transfer statement instructs the normal sequential execution of the part program to be interrupted and the control to be continued at the statement with the specified label.

The General Form is as follows:

JUMPTO/label

Remark: The unconditional transfer statement and the statement associated with the label referred to by this statement shall occur within the same loop, or otherwise, within the same macroregion.

Example: Examples of Unconditional Transfer Statements

LOOPST

. . .

. . .

A) . . .

JUMPTO/A

. . .

LOOPND

8.2.10.2 Syntax The syntax is as follows:

<unconditional transfer statement> :: = JUMPTO/<label>

8.2.11 Conditional Transfer Statement

8.2.11.1 Semantics The conditional transfer statement instructs the normal sequential execution of a part program to be interrupted and the control to continue at the labeled statement selected by the value of arithmetic expression.

The General Form is as follows:

1 F (arithmetic expression) label 1, label 2, label 3

- Remarks
1. Control is transferred to the statement with the first label if the value of arithmetic expression is less than zero, to the second label if it is zero, and to the third label if it is greater than zero.
 2. The unconditional transfer statement and the statement associated with the label referred to by this statement shall occur within the same loop region, or otherwise, within the same macroregion.

Example: Examples of Conditional Transfer Statements

LOOPST

...
A) ...

...
IF (VALUE - 2) A, B, C

C) ...

...

...

B) LOOPND

8.2.11.2 Syntax The syntax is as follows:

<conditional transfer statement>:: = 1 F (<arithmetic expression>)

<label> $\frac{2}{2}$ [, <label>]

8.2.12 Geometric Transfer Statement

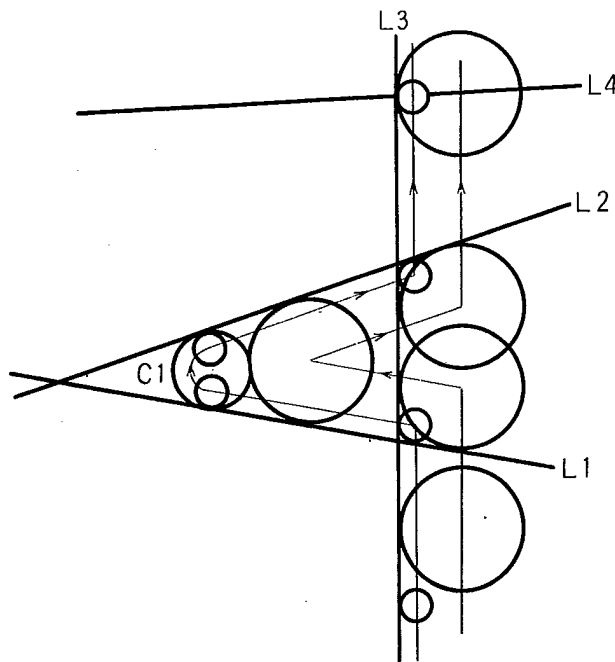
8.2.12.1 Semantics The geometric transfer statement instructs the normal sequential execution of a part program to be interrupted and the control to be transferred to the labeled statement.

The General Form is as follows:

TRANTO/label

- Remarks 1. The geometric transfer statement is used in conjunction with multiple check surface motion statements (refer to 3.6 of Appendix 1).
2. The geometric transfer statement can transfer control only to geometric execution statement and postprocessor statement.

Example: Example of Geometric Transfer Statement



TLRGT, GORGT/L3, PAST, L1
 GOLFT/L1, TO, L2, ID1, TANTO, C1, ID2
 ID1) GORGT/L2, PAST, L3
 TRANTO/ID3
 ID2) GOFWD/C1, TANTO, L2
 GOFWD/L2, PAST, L3
 ID3) GOLFT/L3, ON, L4

8.2.12.2 Syntax The syntax is as follows:

<geometric transfer statement>:: = TRANTO/ <label>

8.2.13 Copy Statement and Index Specification

8.2.13.1 Semantics The copy statement and the index specification instruct the group of statements corresponding to the sequence of tool paths to be a copy region and the tool path to be transferred specified number of times.

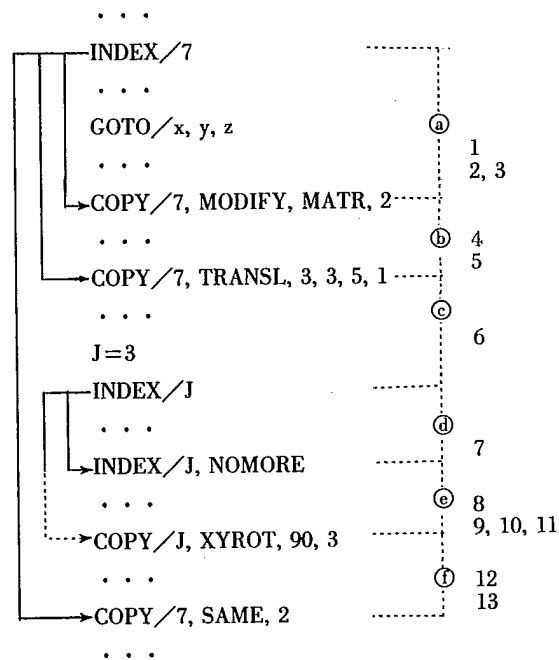
The General Form is as follows:

(1) INDEX/i [, NOMORE]

(2) COPY/i, $\left\{ \begin{array}{l} \text{SAME} \\ \text{TRANSL, a, b, c} \\ \left\{ \begin{array}{l} \text{XYROT} \\ \text{YZROT} \\ \text{ZXROT} \end{array} \right\}, \text{angle} \\ \text{MODIFY, Matrix} \end{array} \right\}, n$

The statement of INDEX/*i* is placed at the top of the sequence of statements as copy region and the statement of COPY/*i*, ... is placed at the end of copy region. When the tool path is transferred, the tool path can be converted and the conversion is assigned by the information written in COPY/*i*, ... When the tool path is transferred *n* times, the tool path before conversion is firstly outputted and the tool path just therebefore is outputted *n* times by the specified conversion. Therefore, the conversion of the tool path on the *n* times is converted to that on the *n*-1st time. INDEX/*i*, NOMORE is placed between INDEX/*i* and COPY/*i*, ... Only the tool path of the range from INDEX/*i* to INDEX/*i*, NOMORE is converted to be outputted.

- Remarks 1. *i* is the scalar to identify the copy region.
2. 1 is the number of times of copy.
 3. SAME is no conversion.
 4. *a*, *b*, *c* are respectively the values of parallel displacement of coordinate axes *Y*, *Y*, and *Z*.
 5. Angle is the rotation angle on a coordinate plane [refer to 9.1.1 (5)].
 6. MODIFY, matrix is the coordinate transformation matrix (refer to 9.19).
 7. A copy region may be within the range assigned by transform cutter location statement (refer to 10.5) or on the contrary, the range assigned by the transform cutter location statement may be within a copy region.
 8. A copy region may be contained within another copy region.
 9. When plural copy regions exist, the same one can not be used as the scalar for identification.

Example: Example of Copy Statement and Index Specification

The above-mentioned tool path is as follows:

- 1: output (a)
- 2: convert (a) by MODIFY to be outputted.
- 3: convert 2 [(a) just therebefore] by MODIFY to be otuputted.
- 4: output (b).
- 5: convert 1, 2, 3, and 4 by TRANSL to be outputted.
- 6: output (c).
- 7: output (d).
- 8: output (e).
- 9: convert (d) by XYROT to be outputted.
- 10: convert 9 by XYROT to be outputted.
- 11: convert 10 by XYROT to be outputted.
- 12: output (f).
- 13: further successively output the tool path from 1 to 12 two more times.

8.2.13.2 Syntax The syntax is as follows:

<index specification>:: = INDEX/ <scalar> $\frac{1}{0}$ [, NOMORE]

<copy statement>:: = COPY/ <copy parameter list>

<copy parameter list>:: = <scalar> , $\frac{1}{1}$ [SAME | TRANSL $\frac{3}{3}$ [, <scalar>]

$\frac{1}{1}$ [XYROT | YZROT | ZXROT | , <scalar>] MODIFY, <matrix spec> , <scalar>

8.2.14 Macroexecution Statement

8.2.14.1 Semantics The macroexecution statement instructs the macro-definition statement in a part program to be executed.

The General Form is as follows:

CALL/Macroidentifier [[, parameter 1 = effective value 1], ... [,
parameter n = effective value n]]

- Remarks
1. When an effective value is specified, the parameter is replaced thereby and when the effective value is not specified, the normal value in macrodefinition statement is taken as the parameter.
 2. The macroidentifier shall be the macroidentifier defined in the part program.
 3. The effective value of parameter given in a macroexecution statement is applied only to its execution.
 4. A macroexecution statement may be placed at any position of a part program. It may be placed either within a loop region or within a macroregion.
 5. When a macroexecution statement is placed within another macroregion, the parameter in the macroexecution statement shall not accord with the parameter in the macroregion.

Example: Examples of Macroexecution Statements

```

. . .
MACNAM=MACRO/A, B = 10, PL, H, P
. . .
CALL/POCKET, D=10, E=3
. . .
TERMAC
. . .
PL5=PLANE/. . .
. . .
CALL/MACNAM, A=S, PL=PL5, H=H5, P=POINT

```

8.2.14.2 Syntax The syntax is as follows:

< macroexecution statement > :: = CALL/ < simple identifier >

$\frac{1}{0}$ [< formal macroassignment list >]

< formal macroassignment list > :: = $\frac{k}{0}$ [, < formal macroitem >]

< formal macroitem > :: = < simple identifier > = < normal macroitem >

8.2.15 Remarks Statement

8.2.15.1 Semantics The remarks statement instructs remarks to be indicated in the print of a part program.

The General Form is as follows:

REMARK/ 'character string'

Example: Example of Remarks Statement

REMARK/'GEOMETRICAL DEFINITIONS FOLLOW'

8.2.15.2 Syntax The syntax is as follows:

<remark statement>:: = REMARK/ <literal character string>

9. Geometrical Definition Statement

9.1 Semantics and Syntax

9.1.1 Semantics The geometrical definition statement defines the graphic necessary for describing the shape of workpiece or the like, vector and matrix, and local reference system.

In the geometrical definition statement, a syntactically valid statement may represent a geometrically nonsensical situation. For instance, the circle the radius of which is negative, can not be defined and the intersection point of the circle having no intersection point can not be defined.

- (1) Coordinate System A graphic is defined either under coordinate system of basic coordinate system (refer to 4.) or under local reference system (refer to 9.2).
- (2) Normal Form Graphics such as point, straight line, etc., to make the processing of NC processor easy, determine the expression form of each graphic datum inside the processor. This expression form of graphic data is called a normal form (refer to Appendix 7).
- (3) Specifier The specifier is the language to indicate the defined graphic. The definition statement of a nested graphic is also included.

For instance, in the definition statement P1 = POINT/2,5 of the point having coordinator (2,5), identifier P1 is a point specifier. In the definition statement of the circle C1 = CIRCLE/CENTER, (POINT/3,8) RADIUS, r having the coordinates (3, 8) of its center and its radius r, (POINT/3,8) is a point specifier and identifier C1 is a circle specifier.

Specifiers are classified as follows:

- Point specifier
- Pattern specifier
- Line specifier
- Plane specifier
- Vector specifier

Circle specifier
 Cylinder specifier
 Sphere specifier
 Cone specifier
 Ellipse specifier
 Hyperbola specifier
 Lofted conic specifier
 General conic specifier
 Quadric specifier
 Tabulated cylinder specifier
 Matrix specifier
 Ruled surface specifier
 Surface specifier
 Conic section specifier

The Point, Pattern, Line, Plane, Vector, Circle, Cylinder, Sphere, Cone, Ellipse, Hyperbola, Lofted Conic, General Conic, Quadric, Tabulated Cylinder, Matrix, Ruled Surface, Surface, and Conic described in the general form of geometrical definition statement shall be respective specifiers.

Further, those carrying numeric or letter like examples of Point 1, Point 2, ... Point n mean respective specifiers.

- (4) Modifiers for Unambiguous Definition of Graphic Modifiers are used to select one graphic among some graphics becoming candidates for the definition of graphics. Modifiers are as follows:

NEGX	IN	XLARGE	RIGHT	LARGE	CLW
POSX	OUT	XSMALL	LEFT	SMALL	CCLW
NEGY		YLARGE			
POSY		YSMALL			
NEGZ		ZLARGE			
POSZ		ZSMALL			

where

- (a) NEGX, . . . , POSZ: They respectively indicate a positive or negative direction as to the coordinate axes specified by the last letter of each modifier and are used to select the desired direction of vector in the definition of vector.
- (b) IN, OUT: They indicate the inclusive relation between the desired circle or cylinder and the specified graphic, IN indicating inscribing relation and OUT indicating circumscribing relation. They are used for selection of the desired circle or cylinder from the inclusive relation to the graphic preliminarily specified in geometrical definition statement when a circle or a cylinder is defined.

- (c) XLARGE, . . . , ZSMALL: They indicate the large and small relation of each coordinate as to the coordinate axes specified by the first letter of each modifier and are used for the selection of the desired graphic in the definition of graphic. Its method used is given in Table 9.1.1. For instance, XLARGE indicates that the graphic on the larger coordinate side as to X axis is selected.

Table 9.1.1. Use Method of XLARGE, . . . , ZSMALL

Type of graphic	Method for selection
Z plane	It is selected according to the large or small of the coordinate of intersection point of the specified coordinate axes and two candidates Z planes.
Point	It is selected according to the large or small of the coordinate when two candidate points are projected onto the specified axes.
Line	It is selected according to the large or small of the coordinate of the intersection point of the specified coordinate axes and two candidate lines.
Plane	It is selected according to the large or small of the coordinate of the intersection point of the specified coordinate axes and two candidate planes.
Circle	(1) It is selected according to the large or small of the coordinate of two candidate regions divided by a plane as to the specified coordinate axes. (2) It is selected according to the large or small of the coordinate of the center of two candidate circles projected onto the specified coordinate axes.
Cylinder	(1) It is selected according to the large or small of the coordinate of two candidate regions divided by a plane as to the specified coordinate axes. (2) It is selected according to the large or small of the coordinate of the center axis of two candidate cylinders as to the specified coordinate axes.

- (d) RIGHT, LEFT: The desired graphic is selected according to either of right or left seeing the second specified graphic from the initial specified graphic.
- (e) LARGE, SMALL: The circle the radius of which is large or small is respectively selected among candidate circles.
- (f) CLW, CCLW: The graphic the direction of which is clockwise or counterclockwise is respectively selected.

- (5) Angle The unit of angle shall be degree. The angle is measured counterclockwise from the basic axes on the specified coordinate plane. The basic axes are the axes having a positive direction on each of the coordinate axes. The relation between the coordinate plane and the basic axes is given in Table 9.1.2. When, for instance, XYPLAN or XYROT is used, the counterclockwise direction from the basic axes X on XY plane is made positive.

Table 9.1.2. Coordinate Plane and Basic Axes

Coordinate plane	Basic axes	Keyword specifying coordinate plane	
XY	X	XYPLAN	XYROT
YZ	Y	YZPLAN	YZROT
ZX	Z	ZXPLAN	ZXROT

9.1.2 Syntax The syntax is as follows:

<geometrical definition statement> :: = < declaration of local reference system> | < declaration of Z plane> | < point definition statement> |
 < pattern definition statement> | < line definition statement> | < plane definition statement> | < vector definition statement> | < circle definition statement> | < cylinder definition statement> | < sphere definition statement> | < cone definition statement> | < ellipse definition statement> |
 < hyperbola definition statement> | < lofted conic definition statement> |
 < general conic definition statement> | < quadric definition statement> |
 < tabulated cylinder definition statement> | < matrix definition statement> |
 < ruled surface definition statement>
 < surface specification> :: = < line spec> | < plane spec> | < circle spec> | < cylinder spec> | < sphere spec> | < cone spec> | < conic spec> | < quadric spec> |
 < tabulated cylinder spec> | < ruled surface spec>
 < conic spec> :: = < ellipse spec> | < hyperbola spec> | < lofted conic spec> |
 < general conic spec>

9.2 Local Reference System

9.2.1 Semantics and Syntax

9.2.1.1 Semantics The local reference system is the coordinate system rotated and/or translated with respect to the basic coordinate system. A graphic may be defined by using local reference system in a part program. The local reference system is indicated by the declaration of local reference system and exact effect on the graphic definition to the end of local reference system. The graphic within local reference system is processed by NC processor and converted to the graphic in basic coordinate system.

9.2.1.2 Syntax The syntax is as follows:

<reference system declaration >:: = REFSYS/<reference system parameter list>

<reference system parameter list> :: = <matrix spec>|NOMORE

9.2.2 Declaration of Local Reference System

9.2.2.1 Semantics The start of local reference system is indicated. The General Form is as follows:

REFSYS/Matrix

Remark: The matrix is the matrix to convert the graphic on local reference system to the basic coordinate system.

9.2.3 End of Local Reference System

9.2.3.1 Semantics The end of local reference system is indicated. The General Form is as follows:

REFSYS/NOMORE

9.3 Declaration of Z Surface

9.3.1 Semantics and Syntax

9.3.1.1 Semantics If Z surface is declared, when the point defined after this declaration and the point in a pattern do not have unambiguous Z coordinate, Z coordinate of this surface is assigned to those points. In the case of no-declaration of Z surface, the value of Z coordinate of those points shall be zero.

9.3.1.2 Syntax The syntax is as follows:

<Z-surface declaration> ::= ZSURF/<Z-surface parameter list>

<Z-surface parameter list> ::= < scalar > |

<plane spec> |

<cylinder spec> $\frac{1}{2}$ [, ZLARGE | , ZSMALL] |

<sphere spec> $\frac{1}{2}$ [, ZLARGE | , ZSMALL] |

<cone spec> $\frac{1}{2}$ [, ZLARGE | , ZSMALL] |

<quadric spec> $\frac{1}{4}$ [, ZLARGE | , ZSMALL]

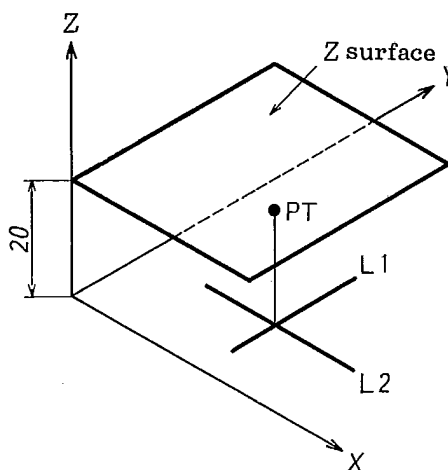
9.3.2 Declaration of Z Surface by Its Height from XY Plane

9.3.2.1 Semantics Z surface is declared by the height from XY plane.
The General Form is as follows:

ZSURF/height

Remark: The height means the height from XY plane.

Example: Example of Declaration of Z Surface by the height from XY Plane



ZSURF/20

PT=POINT/INTOF, L1, L2

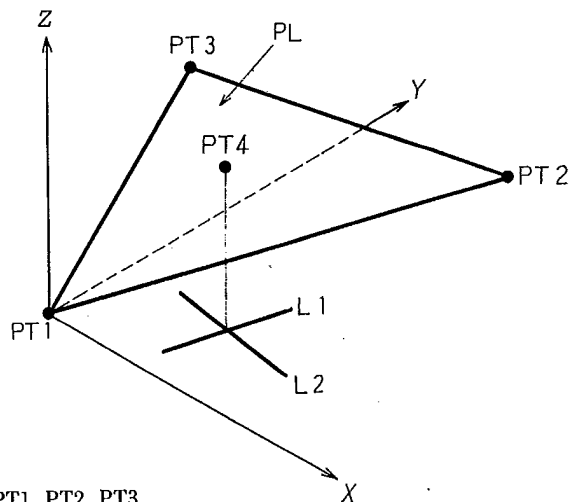
9.3.3 Declaration of Z Surface by Plane

9.3.3.1 Semantics Z surface is declared by plane.
The general Form is as follows:

ZSURF/Plane

Remarks 1. If a plane is parallel to XY plane, the Z coordinate shall be the height from XY plane. If the plane is not parallel to XY plane, Z coordinate value of the point varies with X-and Y coordinate values.

2. The plane shall not be perpendicular to XY plane.

Example: Example of Declaration of Z Surface by Plane

PL=PLANE/PT1, PT2, PT3
 ZSURF/PL
 PT4=POINT/INTOF, L1, L2

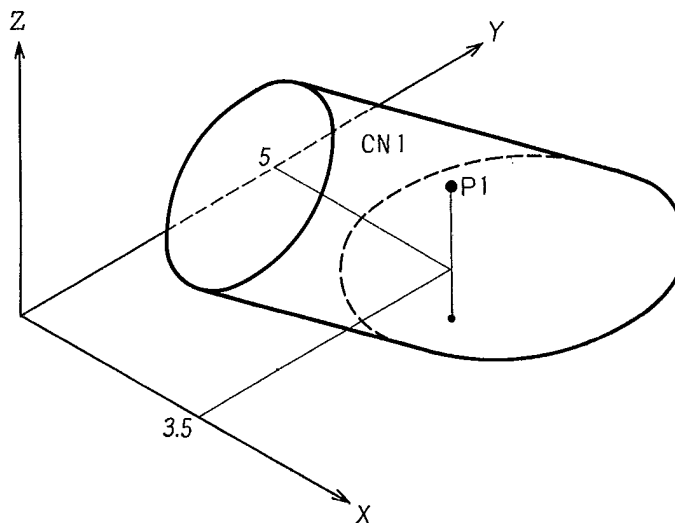
9.3.4 Declaration of Z Surface by Cylinder

9.3.4.1 Semantics Z surface is declared by cylinder.

The General Form is as follows:

ZSURF/Cylinder $\left[, \left\{ \frac{\text{ZLARGE}}{\text{ZSMALL}} \right\} \right]$

Remark: After this declaration of Z surface, the line passing the defined point or each point of pattern, parallel to Z axis shall either intersect the cylinder or contact therewith.

Example: Example of Declaration of Z Surface by Clinder

ZSURF/CN1, ZLARGE
 P1=POINT/3.5, 5

9.3.5 Declaration of Z Surface by Sphere

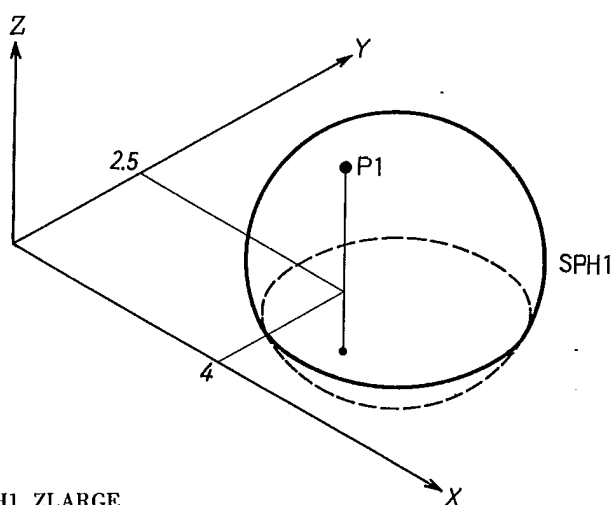
9.3.5.1 Semantics Z surface is declared by sphere.

The General Form is as follows:

ZSURF/Sphere $\left[, \left\{ \frac{\text{ZLARGE}}{\text{ZSMALL}} \right\} \right]$

Remark: After this declaration of Z surface, the line passing the defined point or each point of pattern, parallel to the Z axis shall either intersect the sphere or contact therewith.

Example: Example of Declaration of Z Surface by Sphere



ZSURF/SPH1, ZLARGE
P1=POINT/4, 2.5

9.3.6 Declaration of Z Surface by Cone

9.3.6.1 Semantics Z surface is declared by a cone.

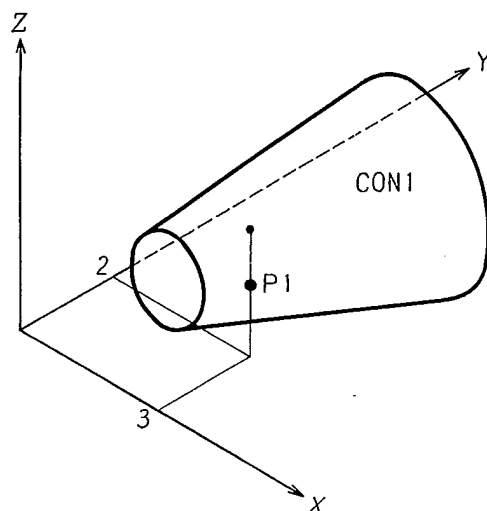
The General Form is as follows:

ZSURF/Cone $\left[, \left\{ \frac{\text{ZLARGE}}{\text{ZSMALL}} \right\} \right]$

Remark: After the declaration of Z surface, the line passing the defined point or each point of a pattern, parallel to the Z axis shall either intersect or touch a cone.

36
B 6327-1985

Example: Example of Declaration of Z Surface by Cone



ZSURF/CON1, ZSMALL

P1=POINT/3, 2

9.3.7 Declaration of Z Surface by Quadric

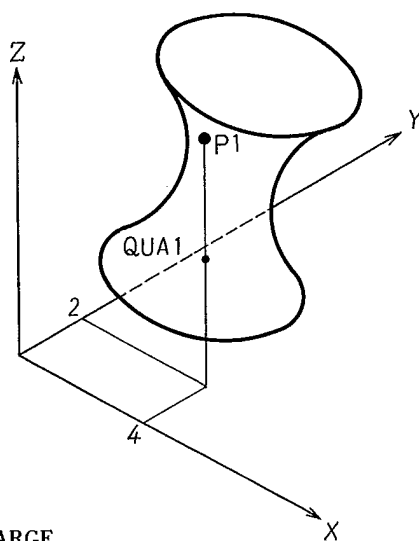
9.3.7.1 Semantics Z surface is declared by a quadric.

The General Form is as follows:

ZSURF/Quadric $\left[, \begin{Bmatrix} \text{ZLARGE} \\ \text{ZSMALL} \end{Bmatrix} \right]$

Remark: After this declaration of Z surface, the line passing the defined point or each point of a pattern, parallel to Z axis shall either intersect or touch a quadric.

Example: Example of Declaration of Z Surface by Quadric



ZSURF/QUA1, ZLARGE

P1=POINT/4, 2

9.4 Point Definition Statement

9.4.1 Semantics and Syntax

9.4.1.1 Semantics A point definition statement defines a point. The point has three coordinates. The Z coordinate of the point the Z coordinate of which is not clearly indicated is obtained from a Z surface when the Z surface is given and is made to be zero when Z surface is not given (refer to 9.3). When the point is defined by three coordinates, or otherwise when the point is defined as the intersecting point of three planes, the Z surface is neglected.

9.4.1.2 Syntax The syntax is as follows:

<point definition statement>:: = <identifier> = POINT / <point parameter

list>

<point spec>:: = <identifier> ($\frac{1}{2}$ [< identifier> =] POINT / <point parameter list >)

<point option>:: = <point spec> | <scalar> , < scalar >

< point parameter list > :: = <scalar> $\frac{2}{3}$ [, < scalar >] |

INTOF $\frac{2}{2}$ [, < line spec >] |

$\frac{1}{2}$ [XLARGE | XSMALL | YLARGE | YSMALL] , INTOF , < line spec > ,

< circle spec > |

$\frac{1}{2}$ [XLARGE | XSMALL | YLARGE | YSMALL , INTOF $\frac{2}{2}$ [, < circle spec >] |

RTHETA , $\frac{1}{2}$ [XYPLAN | YZPLAN | ZXPLAN] $\frac{2}{2}$ [, < scalar >] |

<circle spec > , ATANGL , <scalar>

<pattern spec > , < scalar >

CENTER , <circle spec> |

$\frac{1}{2}$ [XLARGE | XSMALL | YLARGE | YSMALL | , INTOF , < line spec > ,

< conic section spec > |

INTOF , < line spec > , < tabulated cylinder spec > , < point spec > |

INTOF $\frac{3}{3}$ [, < plane spec >] |

<point spec> , DELTA $\frac{2}{2}$ [, < scalar >] |

<point spec> , $\frac{1}{2}$ [THETAR | RTHETA] $\frac{2}{2}$ [, < scalar >] |

< line spec > , $\frac{1}{2}$ [XCOORD | YCOORD] , < scalar > |

< point spec > , $\frac{1}{2}$ [XLARGE | XSMALL | YLARGE | YSMALL | , < tabulated

cylinder spec > , < scalar >