

DIN DEUTSCHES INSTITUT FÜR NORMUNG E.V.

DIN 66 246 Teil 1

Programmierung numerisch gesteuerter Arbeitsmaschinen

Prozessor-Eingabesprache

Grundlagen und mögliche Geometriedefinitions-
und Ausführungsanweisungen

Normenausschuß Informationsverarbeitungssysteme (NI)
im DIN Deutsches Institut für Normung e. V.

Normenausschuß Maschinenbau (NAM) im DIN

BEUTH VERLAG GMBH · BERLIN

Programmierung numerisch gesteuerter Arbeitsmaschinen
Prozessor-Eingabesprache
Grundlagen und mögliche Geometriedefinitions-
und Ausführungsanweisungen

DIN
66 246
Teil 1

Numerical control of machines; NC processor input; basic part program reference language

Zusammenhang mit dem Internationalen Norm-Entwurf ISO/DIS 4342 — 1980, siehe Erläuterungen.

Inhalt

	Seite		Seite
1 Zweck und Anwendungsbereich	5	6 Programmdefinitionsanweisungen	10
2 Allgemeines	5	6.1 Synonymdefinitionsanweisung (SYN)	10
3 Koordinatensystem	5	6.2 Reservierungsanweisung für indizierte Symbole (RESERV)	10
4 Strukturelement	5	6.3 Definition eines Makros (MACRO)	10
4.1 Buchstaben	5	7 Programmausführungsanweisungen	11
4.2 Ziffern	5	7.1 Beginn und Ende eines Teileprogrammes	11
4.3 Sonderzeichen	5	7.1.1 Teileprogrammanfangsanweisung (PARTNO)	11
4.4 Zeichen	6	7.1.2 Teileprogrammabschlußanweisung (FINI)	11
4.5 Begrenzungszeichen für Textzeichenfolgen	6	7.2 Aufruf und Aufrufunterdrückung von Postprozessoren	11
4.6 Textzeichenfolgen	6	7.2.1 Aufruf eines Postprozessors (MACHIN)	11
4.7 Vorzeichenlose Zahlen	6	7.2.2 Aufrufunterdrückung von Postprozessoren (NOPOST)	11
4.8 Schlüsselwörter	6	7.3 Eingabe — Ausgabeanweisungen	11
4.9 Nicht indizierte Symbole	6	7.3.1 Druckanweisung (PRINT)	11
4.10 Indizierte Symbole	7	7.3.2 Allgemeine Ausgabeanweisung (PUNCH)	12
4.11 Marken	7	7.3.3 Leseanweisung (READ)	12
4.12 Anweisungen	7	7.3.4 CLDATA-Druckanweisung (CLPRNT)	12
4.13 Geschachtelte Anweisungen	7	7.3.5 Anweisung zur Kennzeichnung der gedruckten Ausgabe (TITLES)	12
4.14 Teileprogramm	7	7.4 Sprunganweisungen	13
5 Arithmetische Anweisungen	8	7.4.1 Schleifenanfangs- und Schleifenend- anweisung (LOOPST, LOOPND)	13
5.0.1 Allgemeine Semantik der arithmetischen Anweisungen	8	7.4.2 Sprunganweisung (JUMPTO)	13
5.0.2 Syntax der arithmetischen Anweisungen	8	7.4.3 Bedingte Sprunganweisung (IF)	13
5.1 Arithmetische Operatoren	8	7.4.4 Verzweigungsanweisung (TRANTO)	14
5.2 Arithmetische Funktionen	9	7.5 COPY- und Index-Anweisung	14
5.2.1 Algebraische Funktionen	9	7.5.1 COPY-Anweisung (COPY)	14
5.2.2 Trigonometrische und geometrische Funktionen	9	7.5.2 INDEX-Anweisung (INDEX)	14
5.2.3 Exponentialfunktionen	9	7.6 Makroaufruf (CALL)	15
5.2.4 Vektorfunktionen	9		

Fortsetzung Seite 2 bis 70

Normenausschuß Informationsverarbeitungssysteme (NI) im DIN Deutsches Institut für Normung e.V.
Normenausschuß Maschinenbau (NAM) im DIN

Nachdruck, auch auszugsweise, nur mit Genehmigung des DIN Deutsches Institut für Normung e. V., Berlin, gestattet.

7.7	Kommentaranweisung (REMARK)	15
7.8	Postprozessordruckanweisung (PPRINT)	15
8	Geometrische Definitionsanweisungen	16
8.0	Allgemeine Bemerkungen zu geometrischen Definitionsanweisungen	16
8.0.1	Einschränkungen	16
8.0.2	Koordinatensystem und Dimensionen	16
8.0.3	Häufig benutzte Modifikatoren	16
8.0.4	Spezifikatoren	16
8.1	Lokales Bezugssystem (REFSYS)	16
8.1.1	Festlegung eines lokalen Bezugssystems	17
8.1.2	Ende der Gültigkeit eines lokalen Bezugssystems	17
8.2	Festlegung eines impliziten Z-Wertes (ZSURF) durch :	17
8.2.1	– eine Ebene parallel zur XY-Ebene im Abstand h	17
8.2.2	– eine Ebene	17
8.2.3	– eine Zylinderfläche	17
8.2.4	– eine Kugelfläche	18
8.2.5	– eine Kegelfläche	18
8.2.6	– eine allgemeine Fläche 2. Ordnung	18
8.3	Definition eines Punktes (POINT)	18
8.3.1	– durch seine kartesischen Koordinaten	18
8.3.2	– als Schnittpunkt zweier Geraden	19
8.3.3	– als Schnittpunkt einer Geraden und eines Kreises	19
8.3.4	– als Schnittpunkt zweier Kreise	19
8.3.5	– durch seine Polarkoordinaten in der von der Anweisung genannten Koordinatenebene	20
8.3.6	– auf einem Kreis durch den Winkel zwischen Radiusvektor und X-Achse	20
8.3.7	– als n-ter Punkt eines Punktmusters	20
8.3.8	– als Kreismittelpunkt	21
8.3.9	– als Schnittpunkt einer Geraden und eines Kegels	21
8.3.10	– als Schnittpunkt einer Geraden und eines tabellierten Zylinders in der Nähe eines Definitionspunktes des tabellierten Zylinders	21
8.3.11	– als Schnittpunkt dreier Ebenen	21
8.3.12	– durch die Angabe der X- und Y-Inkrementen, die an einen gegebenen Punkt anzutragen sind	22
8.3.13	– durch seinen Abstand von einem gegebenen Punkt und dem Winkel, den die Gerade durch den gegebenen und den definierten Punkt mit der X-Achse bildet	22
8.3.14	– auf einer Geraden durch seine X- oder Y-Koordinate	22
8.3.15	– auf einem tabellierten Zylinder durch seine Entfernung von einem gegebenen Punkt des tabellierten Zylinders	22
8.4	Definition eines Punktmusters (PATTERN)	22
8.4.0	Modifizierte Punktmuster	23
	Definition eines Punktmusters	
8.4.1	– auf einer Geraden durch Anfangs- und Endpunkt und die Punktzahl	23
8.4.2	– durch den Anfangspunkt, einen Vektor, und die Gesamtzahl der Punkte im Punktmuster	23
8.4.3	– durch Anfangspunkt, Richtungsvektor und Inkrement entlang des Vektors	23

8.4.4	– durch den Anfangspunkt, einen Richtungsvektor und eine jeweils spezifizierte Anzahl von Punkten mit gleichem Abstand untereinander längs des Vektors	23
8.4.5	– durch den Anfangspunkt, einen Richtungswinkel und eine jeweils spezifizierte Anzahl von Punkten mit gleichem Abstand untereinander in dieser Richtung	24
8.4.6	– auf einem Kreis durch Anfangs- und Endpunkt, Definitionsrichtung und Punktzahl	24
8.4.7	– auf einem Kreis durch Anfangswinkel, Definitionsrichtung und Punktzahl	24
8.4.8	– auf einem Kreis durch einen Anfangswinkel, die Definitionsrichtung und Winkelinkremente	25
8.4.9	– auf einem Kreis durch Anfangswinkel, Definitionsrichtung und eine jeweils spezifizierte Anzahl von Punkten mit gleichem Abstand untereinander auf dem Kreisumfang	25
8.4.10	– durch direkte Kombination von zwei Punktmustern	25
8.4.11	– durch ein Punktmuster, einen Verschiebungsvektor und die Gesamtzahl der Reihen	26
8.4.12	– durch ein Punktmuster, einen Richtungsvektor und durch Inkremente entlang des Vektors	26
8.4.13	– durch ein Punktmuster, einen Richtungsvektor und eine jeweils spezifizierte Anzahl von Punkten mit gleichem Abstand untereinander entlang des Vektors	27
8.4.14	– durch die Kombination zweier Muster mit der Möglichkeit der Drehung des zweiten Musters	27
8.4.15	– durch die Verkettung von Punkten und Punktmustern	27
8.4.16	– durch ein Punktmuster und eine Spiegelachse	28
8.4.17	– durch die Kombination zweier Muster, wobei das zweite Muster ein kreisförmiges Punktmuster darstellt	28
8.5	Definition einer Geraden (LINE)	28
8.5.1	– durch zwei Punkte	28
8.5.2	– durch die X- oder Y-Achse	29
8.5.3	– parallel zu einer Geraden durch einen Punkt	29
8.5.4	– senkrecht zu einer Geraden durch einen Punkt	29
8.5.5	– durch einen gegebenen Abstand und parallel zu einer Geraden	29
8.5.6	– durch einen Winkel zu einer Geraden und durch einen Punkt	30
8.5.7	– durch den Tangens des Schnittwinkels mit einer gegebenen Geraden und durch einen Punkt	30
8.5.8	– durch den Winkel zur X-Achse und den Achsenabschnitt	30
8.5.9	– durch ihre Steigung gegenüber der X-Achse und den Achsenabschnitt	30
8.5.10	– als Tangente an einen Kreis und durch einen Punkt	31
8.5.11	– als Tangente an zwei Kreise	31
8.5.12	– als Tangente an einen Kreis unter einem bestimmten Winkel zu einer Geraden	31

Seite	Seite
8.5.13 – durch zwei sich schneidende Ebenen	32
8.5.14 – als Tangente an einen Kegelschnitt durch einen Punkt	32
8.5.15 – als Normale in einem Punkt eines Kegelschnittes	32
8.5.16 – als Tangente an einen tabellierten Zylinder in der Nähe eines Definitionspunktes des tabellierten Zylinders und durch einen Punkt	32
8.5.17 – als Senkrechte in der Nähe eines bestimmten Definitionspunktes auf einem tabellierten Zylinder und durch einen Punkt	32
8.5.18 – als Normale auf einem Kegelschnitt durch einen Punkt	33
8.6 Definition einer Ebene (PLANE)	33
8.6.1 – durch die Koeffizienten der für die Ebene geltenden Gleichung $ax + by + cz - d = 0$. . .	33
8.6.2 – durch drei nicht auf einer Geraden liegende Punkte	34
8.6.3 – durch einen Punkt und parallel zu einer gegebenen Ebene	34
8.6.4 – durch eine parallele Ebene in gegebenem Abstand	34
8.6.5 – durch einen Punkt und senkrecht zu einem Vektor	34
8.6.6 – durch zwei Punkte senkrecht zu einer gegebenen Ebene	35
8.6.7 – durch einen Punkt und senkrecht zu zwei sich schneidenden Ebenen	35
8.6.8 – durch einen Punkt und tangential zu einem Zylinder	35
8.7 Definition eines Vektors (VECTOR)	35
8.7.1 – durch seine Komponenten	35
8.7.2 – durch seine Endpunkte	35
8.7.3 – senkrecht zu einer gegebenen Ebene	36
8.7.4 – durch einen Skalar multipliziert mit einem Vektor	36
8.7.5 – durch das Vektorprodukt von zwei gegebenen Vektoren	36
8.7.6 – durch Normalisierung eines gegebenen Vektors	37
8.7.7 – durch seine Länge und einen Winkel in einer gegebenen Koordinatenebene	37
8.7.8 – als Einheitsvektor, parallel zur Schnittgeraden zweier gegebener Ebenen	37
8.7.9 – durch Addition oder Subtraktion von zwei gegebenen Vektoren	37
8.7.10 – als Einheitsvektor in der XY-Ebene, durch einen Winkel zu einer Geraden	38
8.8 Definition eines Kreises (CIRCLE)	38
8.8.1 – durch die Koordinaten seines Mittel- punktes und den Radius	38
8.8.2 – durch seinen Mittelpunkt und den Radius	38
8.8.3 – durch seinen Mittelpunkt und eine Tangente	38
8.8.4 – durch seinen Mittelpunkt und einen Punkt auf dem Kreisumfang	39
8.8.5 – durch drei Punkte auf dem Kreisumfang	39
8.8.6 – durch seinen Mittelpunkt und einen Tangentialkreis	39
8.8.7 – durch zwei Tangenten und einen Radius	39
8.8.8 – durch eine Tangente, einen Punkt auf dem Kreisumfang und einen Radius	40
8.8.9 – durch zwei Tangentialkreise und einen Radius	40
8.8.10 – durch eine Tangente, einen Tangentialkreis und einen Radius	41
8.8.11 – durch eine Tangente, einen tangierenden Tabcyl und einen Radius	41
8.8.12 – durch drei Tangenten	41
8.8.13 – durch zwei Punkte auf dem Kreisumfang und einen Radius	42
8.8.14 – durch einen Punkt auf dem Kreisumfang, einen Tangentialkreis und einen Radius	42
8.8.15 – durch eine Tangente, einen tangierenden Kegelschnitt und einen Radius	42
8.8.16 – durch zwei tangierende Tabcyl's und einen Radius	42
8.8.17 – durch einen Tangentialkreis, einen tangierenden Tabcyl und einen Radius	43
8.8.18 – durch einen Punkt auf dem Kreisumfang, einen tangierenden Tabcyl und einen Radius	43
8.8.19 – durch eine Mittelpunktgerade, einen tangierenden Tabcyl und einen Radius	43
8.9 Definition eines Zylinders (CYLNDR)	43
8.9.1 – durch einen Punkt auf der Zylinderachse, den Einheitsvektor in Richtung der Zylinderachse und den Radius	43
8.9.2 – durch seinen Radius und zwei tangierende Ebenen	44
8.9.3 – durch seinen Radius, eine tangierende Ebene und einen tangierenden Zylinder	44
8.9.4 – durch seinen Radius und zwei tangierende Zylinder	45
8.10 Definition einer Kugel (SPHERE)	45
8.10.1 – durch die Koordinaten ihres Mittelpunktes und ihren Radius	45
8.10.2 – durch ihren Mittelpunkt und ihren Radius	46
8.10.3 – durch ihren Mittelpunkt und einen Punkt auf der Kugel	46
8.10.4 – durch ihren Mittelpunkt und eine tangierende Ebene	46
8.10.5 – durch vier Punkte auf der Kugel	46
8.11 Definition eines Kegels (CONE)	46
8.11.1 – durch seinen Scheitelpunkt, seinen Achs- vektor und seinen Halbwinkel	46
8.11.2 – durch die Koordinaten seines Scheitel- punktes, die Komponenten eines Vektors in Achsrichtung und den Kosinus seines Halbwinkels	47
8.12 Definition einer Ellipse (ELLIPS)	47
8.12.1 – durch ihren Mittelpunkt, die Länge ihrer großen Halbachse, die Länge ihrer kleinen Halbachse und den Winkel, den die große Halbachse mit der X-Achse bildet	47
8.13 Definition einer Hyperbel (HYPERB)	47

	Seite
8.13.1 – durch den Mittelpunkt, die halbe Länge der reellen Achse, die halbe Länge der imaginären Achse, und den Winkel zwischen reeller Achse und X-Achse	47
8.14 Definition eines Kegelschnittes durch Punkte oder Tangenten des Kegelschnittes (LCONIC)	47
Definition eines Kegelschnittes	48
8.14.1 – durch fünf Punkte	48
8.14.2 – durch vier Punkte und eine Steigung	48
8.14.3 – durch drei Punkte und zwei Steigungen	48
8.15 Definition eines Kegelschnittes durch seine mathematische Gleichung (GCONIC)	48
8.15.1 – durch die allgemeine Gleichung der Kurve 2. Ordnung	48
8.15.2 – durch eine Gleichung der Kurve 2. Ordnung, in der y als Funktion von x ausgedrückt ist	49
8.15.3 – durch eine Gleichung der Kurve 2. Ordnung, in der x als Funktion von y ausgedrückt ist	49
8.16 Definition einer allgemeinen Fläche 2. Ordnung (QADRIC)	49
8.16.1 – Definition einer Fläche durch die Koeffizienten der allgemeinen Gleichung 2. Ordnung	49
8.17 Definition eines tabellierten Zylinders (TABCYL)	50
Definition eines Tabcyls	50
8.17.1 – durch eine Folge von Punkten oder deren kartesische Koordinaten in einer Hauptebene, wobei die Erzeugende eine Parallele zu einer Koordinatenachse ist	51
8.17.2 – durch eine Folge von Punkten, angegeben in ihren Polarkoordinaten, wobei die Erzeugende eine Parallele zur Z-Achse ist	51
8.17.3 – durch eine Folge von Punkten, wobei die Erzeugende eine Parallele zu einem angegebenen Vektor ist	52
8.17.4 – als Äquidistante zu einem angegebenen Tabcyl	52
8.18 Definition einer Matrix (MATRIX)	52
8.18.1 – durch zwölf Parameter	52

	Seite
8.18.2 – durch Angabe des Verschiebungsvektors	53
8.18.3 – durch Angabe des Drehwinkels in einer Ebene	53
8.18.4 – durch Angabe eines Verzerrungsfaktors	53
8.18.5 – durch Angabe einer Spiegelungsachse, oder einer Spiegelungsebene	54
8.18.6 – durch die Angabe der Hauptebenen, an denen gespiegelt wird	54
8.18.7 – durch die Angabe von drei Ebenen	54
8.18.8 – durch Angabe eines Punktes und zweier Vektoren	54
8.18.9 – durch das Produkt zweier Matrizen	55
8.18.10 – als die Inverse einer gegebenen Matrix	55
8.19 Definition einer Regelfläche (RLDSRF)	55
8.19.1 – durch die Vorgabe zweier ebener Raumkurven	55
8.19.2 – durch die Vorgabe einer ebenen Raumkurve und eines Punktes	55
9 Geometrische Ausführungsanweisungen	56
9.0.1 Allgemeine Semantik	56
9.1 Bewegungsanweisungen	63
9.1.1 FROM	63
9.1.2 GOTO	64
9.1.3 GODLTA	64
9.1.4 GO	65
9.1.5 OFFSET	65
9.1.6 GO *** (GOFWD, usw.)	65
9.2 Werkzeuglage-Anweisungen	66
9.2.1 Werkzeuglage zur Führungsfläche	66
9.2.2 Modale Angabe zur Werkzeuglage zur Teilefläche	66
9.2.3 Dicken-Spezifikator	67
9.2.4 Toleranz-Anweisung; Toleranzband, Innentoleranz, Außentoleranz	67
9.2.5 Werkzeugmaße und -art	68
9.2.6 Teileflächen-Spezifikator (PSIS)	68
9.2.7 Teileflächen-Spezifikator (NOPS)	68
9.2.8 Teileflächen-Spezifikator (AUTOPS)	69
9.3 Anfahr-Richtungssteuerung	69
9.3.1 Anfahr-Richtungssteuerung in Richtung auf Punkt(e)	69
9.3.2 Anfahr-Richtungssteuerung in Richtung des Vektors (der Vektoren)	69
9.3.3 Anfahr-Richtungssteuerung durch Flächenvektor(en)	69
9.4 Transformation der berechneten Werkzeugbahn	69
9.5 Unterdrücken der Werkzeugpositionen	70

1 Zweck und Anwendungsbereich

Die Norm dient dazu, die Sprache für NC-Prozessor-Eingabetexte zur rechnergestützten Programmierung von numerisch gesteuerten Arbeitsmaschinen festzulegen. 1)

Mit jedem NC-Prozessor soll es möglich sein, zumindest eine Untermenge der in dieser Norm festgelegten Eingabesprache zu verarbeiten.

Ein NC-Prozessor ist ein Programm, das Kompilier-, Übersetzer- und verwandte Funktionen von Programmiersprachen für numerisch gesteuerte Arbeitsmaschinen beschreibt.

Jeder NC-Prozessor soll CLDATA-Texte erzeugen, wie in DIN 66 215 Teil 1 und Teil 2 festgelegt.

Die Teile der Eingabesprache, die Postprozessoranweisungen sind, sind in DIN 66 215 Teil 2 festgelegt.

Für die Teile der Eingabesprache, die technologische Definitions- und Ausführungsanweisungen sind, ist eine Norm in Vorbereitung.

2 Allgemeines

2.1 Die Eingabesprache für NC-Prozessoren ist eine problemorientierte Sprache zur Beschreibung von Bearbeitungsaufgaben für numerisch gesteuerte Arbeitsmaschinen. Grundlage dieser Norm bilden die APT-ähnlichen Sprachen (APT = Automatically Programmed Tools).

Alle Schlüsselwörter (siehe Abschnitt 4.8) sind aus der englischen Sprache abgeleitet.

Viele stammen aus der Terminologie der Metallbearbeitung.

Ein in der Eingabesprache geschriebener Text heißt Teileprogramm und ist eine vollständige und geordnete Folge von Anweisungen.

2.2 Anweisungen werden in Zeilen geschrieben.

Es gibt kein Sonderzeichen für die Beendigung einer Anweisung, jedoch ein Sonderzeichen, das anzeigt, daß eine Anweisung auf der nächsten Zeile fortgesetzt wird. Durch dieses Konzept kann die Eingabesprache beschrieben werden, ohne auf die Vorschriften über die Darstellung auf verschiedene Medien, beispielsweise Lochkarten oder Lochstreifen eingehen zu müssen.

2.3 In der Eingabesprache können Symbole verwendet werden, d. h. einer Untermenge kann ein symbolischer Name zugeordnet werden. Auf diese Untermenge kann im Teileprogramm später unter ihrem symbolischen Namen zurückgegriffen werden.

Es gibt keine explizite oder implizite Typenvereinbarung. Ein Symbol wird bestimmt durch den Typ der Anweisung, in der es definiert ist.

2.4 Die verschiedenen Eigenschaften der Sprache werden durch „Semantik“ und „Syntax“ beschrieben und gegebenenfalls durch „Zusammenhang“, „Beispiel“, „Einschränkung“ und „Querbezug“ ergänzt.

1) NC ist die Abkürzung des englischen Ausdrucks 'Numerical Control' (Numerische Steuerung).

2) Dieser Bezugspunkt ist für jeden NC-Prozessor zu definieren.

3 Koordinatensystem

3.1 Grundlage für die Definition des Koordinatensystems der Eingabesprache ist die Norm DIN 66 217.

3.2 Das Koordinatensystem ist ein rechtshändiges kartesisches System, bezogen auf ein auf eine Arbeitsmaschine gespanntes Werkstück. Die Achsen des Koordinatensystems verlaufen gleichsinnig parallel zu den linearen Bewegungsrichtungen der Arbeitsmaschine und heißen X, Y und Z.

3.3 Es wird davon ausgegangen, daß das Werkstück ortsfest ist und das Werkzeug sich relativ zum Koordinatensystem bewegt.

3.4 Die Prozessorausgabe (CLDATA) unterliegt hinsichtlich des Koordinatensystems denselben Regeln wie die Eingabesprache. Die ausgegebenen Koordinatenwerte bezeichnen den Bezugspunkt des Werkzeuges im Koordinatensystem des Werkstückes. 2)

4 Strukturelemente

4.1 Buchstaben

Semantik:

Buchstaben haben keine individuelle Bedeutung. Sie werden zur Bildung von Schlüsselwörtern, Symbolen, Marken und Textzeichenfolgen benutzt.

Syntax:

< Buchstabe > : : = 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

4.2 Ziffern

Semantik:

Ziffern haben keine individuelle Bedeutung. Sie werden zur Bildung von Zahlen, Symbolen, Marken und Textzeichenfolgen benutzt.

Syntax:

< Ziffer > : : = 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9

4.3 Sonderzeichen

Semantik:

Sonderzeichen werden als Operatoren zur Bildung von arithmetischen Ausdrücken und Trennungszeichen in Anweisungen benutzt.

+ – * / arithmetische Operatoren (siehe Abschnitt 5.1)

) Das Zeichen „Klammer zu“ grenzt eine Marke (siehe Abschnitt 4.11) von der zugehörigen Anweisung ab oder wird in Verbindung mit dem Zeichen „Klammer auf“ für die dort beschriebenen Fälle bei Indizierung (siehe Abschnitt 4.10) und geschachtelten Anweisungen (siehe Abschnitt 4.13) verwendet.

(Das Zeichen „Klammer auf“ wird in Verbindung mit dem Zeichen „Klammer zu“ bei Indizierung (siehe Abschnitt 4.10), geschachtelten Anweisungen (siehe Abschnitt 4.13), algebraischen Anweisungen (siehe Abschnitt 5.0.1) und Argumenten von Funktionen (siehe Abschnitt 5.2) verwendet.

Das Zeichen „Punkt“ wird als Dezimalpunkt verwendet.

- = Das „Gleichheitszeichen“ grenzt ein Symbol von der zugehörigen Anweisung ab und wird in arithmetischen Anweisungen (siehe Abschnitt 5.0.1) sowie in Makrodefinitionsanweisungen (siehe Abschnitt 6.3) und Makroaufrufanweisungen (siehe Abschnitt 7.6) benutzt.
- / Der „Schrägstrich“ grenzt das Hauptwort von dem Rest der Anweisung ab.
- , Das Zeichen „Komma“ grenzt Anweisungselemente von einander ab.
- \$ Das Dollarzeichen kennzeichnet die Fortsetzung einer Anweisung auf einer neuen Zeile. Durch zwei Dollarzeichen wird eine Anweisung von nachfolgendem Kommentar getrennt. 3)
- ␣ Das Zeichen „Zwischenraum“ hat nur in Textzeichenfolgen (siehe Abschnitt 4.6) Bedeutung.
- ' Das Zeichen „Apostroph“ wird als Begrenzungszeichen für Textzeichenfolgen verwendet.

Syntax:

< Sonderzeichen > : : =
 . | + | - | * | / | | (|) | \$ | ␣ | ' | = | 3)

4.4 Zeichen

Semantik:

Ein Zeichen kann ein Buchstabe, eine Ziffer oder ein Sonderzeichen sein.

Syntax:

< Zeichen > : : =
 < Buchstabe > | < Ziffer > | < Sonderzeichen >

4.5 Begrenzungszeichen für Textzeichenfolgen

Semantik:

Das Zeichen „Apostroph“ kennzeichnet den Anfang und das Ende einer Textzeichenfolge

Beispiel:

PARTNO 'Ventil Gehäuse'

Syntax:

< Begrenzungszeichen für Textzeichenfolgen > : : = '

4.6 Textzeichenfolgen

Semantik:

Textzeichenfolgen werden verwendet bei erläuternden Anweisungen (z. B. REMARK siehe Abschnitt 7.7), bei Anweisungen für Textzeilen (z. B. TITLES siehe Abschnitt 7.3.5) um Sonderinformationen in die Postprozessorausgabe zu schreiben (z. B. INSERT siehe DIN 66 215 Teil 2).

Syntax:

< Textzeichenfolge > : : =
 $\begin{matrix} k \\ 0 \end{matrix} [< \text{Zeichen} >] |$
 < Begrenzungszeichen für Textzeichenfolgen >
 $\begin{matrix} k \\ 0 \end{matrix} [< \text{Zeichen} >]$
 < Begrenzungszeichen für Textzeichenfolgen > 4)

- 3) Das Dollarzeichen wird als Beispiel für ein Währungszeichen angegeben.
- 4) Die Definition der Textzeichenfolge impliziert, daß auch die Angabe keines Textes zulässig ist.
- 5) DIN 66 215 Teil 1 und Teil 2.

4.7 Vorzeichenlose Zahlen

Semantik:

Zahlen haben ihre übliche Bedeutung und werden als Dezimalzahlen aufgefaßt. Wenn eine Zahl keinen Dezimalpunkt enthält, wird angenommen, daß er auf der äußersten rechten Stelle steht.

Einschränkungen:

Es gibt keine Festlegungen über die Anzahl der Stellen einer Zahl und ihre Rundung; sie sind implementierungsabhängig. Unterscheidungen zwischen ganzen Zahlen und realen Zahlen werden nicht gemacht, da intern nur Realzahlen verwendet werden. Immer wenn eine ganze Zahl benötigt wird (beispielsweise für einen Index) wird der Bruchteil des realen Wertes nach Rundung unterdrückt.

Beispiel:

47711

58.

3.14

Syntax:

< Vorzeichenlose Zahl > : : =
 $\begin{matrix} k \\ 1 \end{matrix} [< \text{Ziffer} >] [.] |$
 $\begin{matrix} k \\ 0 \end{matrix} [< \text{Ziffer} >] . \begin{matrix} k \\ 1 \end{matrix} [< \text{Ziffer} >]$

4.8 Schlüsselwörter

Semantik:

Schlüsselwörter haben eine bestimmte Bedeutung innerhalb der Eingabesprache.

Die Anzahl der Schlüsselwörter ist nicht festgelegt, sie kann bei späteren Änderungen dieser Norm erweitert werden.

Schlüsselwörter übernehmen nicht die Aufgabe von Begrenzungszeichen, da zwei Schlüsselwörter durch Sonderzeichen voneinander zu trennen sind.

Schlüsselwörter werden in zwei Gruppen eingeteilt, die Hauptwörter, die die Art der Anweisung definieren, und die Nebewörter, die zusätzliche Information liefern. Diese Gruppen können jeweils in zwei weitere Untergruppen aufgeteilt werden: die Prozessorschlüsselwörter und die Postprozessorschlüsselwörter. Postprozessorschlüsselwörter sind in einem eigenen Dokument aufgelistet und definiert. 5)

Einschränkungen:

Schlüsselwörter bestehen mindestens aus zwei Zeichen und werden nur aus Buchstaben gebildet.

Beispiel:

POINT

INTOF

Syntax:

< Schlüsselwort > : : = $\begin{matrix} 6 \\ 2 \end{matrix} [< \text{Buchstabe} >]$

4.9 Nicht indizierte Symbole

Semantik:

Nicht indizierte Symbole haben im Gegensatz zu den Schlüsselwörtern keine eigene Bedeutung.

Sie erhalten ihre zeitlich begrenzte Bedeutung erst aus dem Zusammenhang. Sie werden benutzt als Namen, Marken oder arithmetische Symbole.

Einschränkungen:

Mit Ausnahme der arithmetischen Symbole darf ein nicht indiziertes Symbol innerhalb eines Teileprogramms nicht mehrfach definiert werden. Prozessoren sollten Schlüsselwörter als nicht indizierte Symbole nicht zulassen.

Beispiel:

P2
L1
BOBBY

Syntax:

< nicht indiziertes Symbol > : : = < Buchstabe >
 $\begin{matrix} 5 \\ 0 \end{matrix}$ [< Buchstabe > | < Ziffer >]

4.10 Indizierte Symbole

Ein indiziertes Symbol entspricht in seiner Bedeutung, abgesehen vom Index, dem nicht indizierten Symbol. Der Index kann in implementierungsabhängigen Ausnahmefällen entfallen.

Beispiel:

P2(1)
P2(5)

Syntax:

< Indiziertes Symbol > : : =
 < nicht indiziertes Symbol > [(< Skalar >)]

4.11 Marken

Semantik:

Einer Anweisung kann zur Kennzeichnung eine Marke zugeordnet werden. Auf diese Marke kann in einer Sprunganweisung Bezug genommen werden.

Beispiel:

A1) GOTO/P1
Wobei A1 die Marke ist.

Syntax:

< Marke > : : =
 $\begin{matrix} 6 \\ 1 \end{matrix}$ [< Ziffer >] | < nicht indiziertes Symbol >

4.12 Anweisungen

Semantik:

Eine Anweisung ist eine vollständige Instruktion oder Informationseinheit, vergleichbar mit einem Satz in einer natürlichen Sprache. Eine Anweisung besteht aus einer syntaktisch definierten Folge von Elementen, die durch Sonderzeichen voneinander getrennt sind. Diese Elemente sind Schlüsselwörter, Symbole, Zahlen und Textzeichenfolgen.

Es gibt mehrere Arten von Anweisungen, die zur einfacheren Beschreibung in die folgenden fünf Abschnitte dieser Norm eingeordnet werden:

- Arithmetische Anweisungen
- Programmdefinitionsanweisungen
- Programmausführungsanweisungen
- Geometrische Definitionsanweisungen
- Bewegungsanweisungen.

Anweisungen können markiert oder nicht markiert sein. Markierte Anweisungen können innerhalb einer Schleife

oder eines Makros vorkommen. Außerhalb von Schleife und Makro können markierte Anweisungen nur noch in der Verbindung mit der Fähigkeit, eine Kontrollfläche unter zwei möglichen auszuwählen, auftreten.

Schlüsselwörter innerhalb einer Anweisung können durch zuvor definierte Synonyme ersetzt werden (siehe Abschnitt 6.1).

In Makroanweisungen können Schlüsselwörter oder Symbole durch Formalparameter ersetzt werden (siehe Abschnitt 6.3).

Syntax:

< Anweisung > : : = [< Marke >]
 < nicht markierte Anweisung >
 < nicht markierte Anweisung > : : =
 < arithmetische Anweisung > |
 < Programmdefinitionsanweisung > |
 < Programmausführungsanweisung > |
 < geometrische Definitionsanweisung > |
 < Bewegungsanweisung >

4.13 Geschachtelte Anweisungen

Semantik:

Definitionsanweisungen können, mit Ausnahme der Makrodefinitionsanweisung, geschachtelt werden.

Geschachtelte Anweisungen werden zwischen Klammern geschrieben. Immer wenn ein Symbol auftreten kann, kann die volle Anweisung mit Symbol (in diesem Fall kann später im Teileprogramm auf dieses Symbol Bezug genommen werden), oder ohne Symbol geschachtelt werden.

Da ein Sachverhalt durch ein Symbol, eine geschachtelte Anweisung mit Symbol oder eine geschachtelte Anweisung ohne Symbol dargestellt werden kann, wird für die Summe dieser Definitionsmöglichkeiten der Ausdruck „Spezifikator“ eingeführt (siehe Abschnitt 8.0.4).

Beispiel:

Die Anweisungen

PT4 = POINT/3,6
 PT6 = POINT/8,9
 L2 = LINE/PT4, PT6

können wie folgt geschrieben werden

L2 = LINE/(PT4 = POINT/3,6), (POINT/8,9)

4.14 Teileprogramm

Semantik:

Ein Teileprogramm ist eine vollständige und geordnete Folge von Anweisungen. Das Ergebnis der Verarbeitung eines Teileprogramms ist ein entsprechend vollständiges und geordnetes Maschinensteuerprogramm.

Ein Teileprogramm besteht aus einer Anweisung zur Kennzeichnung des Teileprogrammanfangs, einer Folge weiterer Anweisungen und einer Anweisung zur Kennzeichnung des Teileprogrammendes.

Syntax:

< Teileprogramm > : : =
 < Teileprogrammanfangsanweisung >
 $\begin{matrix} k \\ 0 \end{matrix}$ [< Anweisung >]
 < Teileprogrammabschlußanweisung >

5 Arithmetische Anweisung

5.0 Syntax und Semantik der arithmetischen Anweisung

5.0.1 Allgemeine Semantik der arithmetischen Anweisungen

Aufgrund der arithmetischen Eigenschaften der Sprache kann ein großer Bereich algebraischer Ausdrücke in einem Teileprogramm verwendet werden. Im allgemeinen werden in der Teileprogrammarithmetik die gleichen Operatoren benutzt wie in der konventionellen Algebra. Das algebraische Multiplikationszeichen $\times$ wird durch $*$ ersetzt.

Der waagerechte Bruchstrich wird ersetzt durch $/$.

Die Hochstellung des Exponenten wird durch $**$ ersetzt.

Die normalen mathematischen Einschränkungen gelten wie überall; beispielsweise muß bei der Verwendung von $\text{SQRT}(A)$ (siehe Abschnitt 5.2.1) A einen positiven Wert haben oder bei der Verwendung von $\text{ASIN}(B)$ (siehe Abschnitt 5.2.2) der Wert von B zwischen ± 1 liegen.

Das Zeichen $=$ wird in arithmetischen Anweisungen benutzt, braucht aber nicht die Bedeutung „gleich“ zu haben wie in der Algebra. In der Teileprogrammarithmetik sollte dieses Zeichen gelesen werden als „wird ersetzt durch“ oder „erhält den Wert von“. Deshalb ist es zulässig $A = A + 1$ zu schreiben, was bedeutet, daß Eins dem Wert von A hinzuaddiert werden muß und A dann den Wert des Ergebnisses erhält.

Es gelten die normalen Prioritäten wie bei algebraischen Operationen. Beispielsweise wird in der Anweisung $A = 2 + 3 * 4$ die $*$ Operation vor der $+$ Operation ausgeführt. Klammern können in algebraischen Anweisungen benutzt werden wie in der Algebra.

Die implizite Multiplikationsvereinbarung der Algebra ist nicht zulässig.

$A = 5 (B + 2)$ beispielsweise in der Algebra zulässig, muß als Anweisung geschrieben werden

$A = 5 * (B + 2)$

Hierarchie der Prioritäten, nach denen die Operationen auszuführen sind:

Priorität	1	()
	2	**
	3	* oder /
	4	+ oder -

In den Beispielen der folgenden Abschnitte sind nur numerische Näherungswerte angegeben.

5.0.2 Syntax der arithmetischen Anweisungen

$\langle \text{Additionsoperator} \rangle : : = + \mid -$

$\langle \text{Multiplikationsoperator} \rangle : : = * \mid /$

$\langle \text{Exponentialoperator} \rangle : : = **$

$\langle \text{Primärausdruck} \rangle : : = \langle \text{Vorzeichenlose Zahl} \rangle \mid$

$\langle \text{arithmetisches Symbol} \rangle \mid \langle \text{Skalarfunktion} \rangle \mid$

$\langle \text{arithmetischer Ausdruck} \rangle$

$\langle \text{Faktor} \rangle : : = \langle \text{Primärausdruck} \rangle \mid$

$\langle \text{Primärausdruck} \rangle \langle \text{Exponentialoperator} \rangle \langle \text{Primärausdruck} \rangle$

$\langle \text{Term} \rangle : : = \langle \text{Faktor} \rangle \mid \langle \text{Term} \rangle \langle \text{Multiplikationsoperator} \rangle \langle \text{Faktor} \rangle$

$\langle \text{arithmetischer Ausdruck} \rangle : : = \langle \text{Term} \rangle \mid$

$\langle \text{Additionsoperator} \rangle \langle \text{Term} \rangle \mid \langle \text{arithmetischer Ausdruck} \rangle \langle \text{Additionsoperator} \rangle \langle \text{Term} \rangle$

$\langle \text{Skalar} \rangle : : = \langle \text{Primärausdruck} \rangle \mid \langle \text{Additionsoperator} \rangle \langle \text{Primärausdruck} \rangle \mid \langle \text{arithmetisches Symbol} \rangle = \langle \text{Skalar} \rangle$

$\langle \text{arithmetische Anweisung} \rangle : : =$

$\langle \text{arithmetisches Symbol} \rangle = \langle \text{Skalar} \rangle$

$\langle \text{algebraische Funktion} \rangle : : =$

$\sqrt[n]{\text{ABS} \mid \text{SQRT}} \langle \text{Skalar} \rangle$

$\langle \text{trigonometrische Funktion} \rangle : : =$

$\sqrt[n]{\text{SIN} \mid \text{COS} \mid \text{TAN} \mid \text{ASIN} \mid \text{ACOS} \mid \text{ATAN}} \langle \text{Skalar} \rangle$

$\langle \text{geometrische Funktion} \rangle : : =$

$\text{ANGL} \langle \text{Kreisspezifikator} \rangle, \langle \text{Punktspezifikator} \rangle \mid$

$\text{DIST} \langle \text{Punktspezifikator} \rangle, \langle \text{Punktspezifikator} \rangle \mid$

$\text{NUM} \langle \text{Punktmusterspezifikator} \rangle$

$\langle \text{Exponentialfunktion} \rangle : : =$

$\sqrt[n]{\text{EXP} \mid \text{LOG}} \langle \text{Skalar} \rangle$

$\langle \text{Vektorfunktion} \rangle : : =$

$\text{LNTH} \langle \text{Vektorspezifikator} \rangle \mid$

$\text{DOT} \langle \text{Vektorspezifikator} \rangle, \langle \text{Vektorspezifikator} \rangle$

$\langle \text{Skalarfunktion} \rangle : : =$

$\langle \text{trigonometrische Funktion} \rangle \mid \langle \text{algebraische Funktion} \rangle \mid \langle \text{geometrische Funktion} \rangle \mid$

$\langle \text{Exponentialfunktion} \rangle \mid \langle \text{Vektorfunktion} \rangle$

5.1 Arithmetische Operatoren

Semantik:

Die Additionsoperatoren $+$ und $-$ können in einer der folgenden Weisen benutzt werden:

- als monadischer Operator; in diesem Fall bezeichnet der Operator das Vorzeichen, das dem folgenden Term zu geben ist:
 - $+$ besagt, daß der folgende Term positiv sein muß,
 - $-$ besagt, daß der folgende Term negativ sein muß.
- als Binäroperator; in diesem Fall besagt der Operator, dem ein arithmetischer Ausdruck voraus- und ein Term nachgestellt ist, wie der Term mit dem arithmetischen Ausdruck additiv verknüpft werden muß:
 - $+$ besagt, daß der Term dem arithmetischen Ausdruck hinzuaddiert werden muß
 - $-$ besagt, daß der Term vom arithmetischen Ausdruck zu subtrahieren ist.

Die Multiplikationsoperatoren $*$ und $/$ besagen, wie der vorangehende Term mit dem folgenden Faktor multiplikativ verknüpft werden soll:

- $*$ der Term ist mit dem Faktor zu multiplizieren.
- $/$ der Term ist durch den Faktor zu dividieren.

Der Exponentialoperator $**$ besagt, daß der vorangehende Primärausdruck in die vom folgenden Primärausdruck angegebene Potenz zu erheben ist.

Beispiele:

$A = B + C$ A erhält den Wert der Summe von B und C

$A = B * C$ A erhält den Wert des Produktes von B und C

$A = B ** C$ A erhält den Wert von B zur Potenz C

Syntax:

< Additionsoperator > : : = + | -

< Multiplikationsoperator > : : = * | /

< Exponentialoperator > : : = **

5.2 Arithmetische Funktionen

Allgemeines

Im gesamten folgenden Abschnitt wurde die bevorzugte Form der Funktionsnamen benutzt, d. h. SIN, ABS usw.

Es wird jedoch zugestanden, daß die älteren „F“-Formen, nämlich SINF, ABSF usw. in einigen Fällen noch eine Zeitlang in Gebrauch sein können. Die Erläuterungen und Bemerkungen zu den bevorzugten Formen gelten deshalb genau so für die „F“-Versionen.

5.2.1 Algebraische Funktionen

Semantik:

Der Name für die algebraische Funktion bezeichnet die Art des arithmetischen Prozesses, der an dem nachfolgenden in Klammern geschriebenen arithmetischen Ausdruck durchzuführen ist.

ABS es muß der absolute Wert des arithmetischen Ausdrucks gebildet werden

SQRT es muß die Quadratwurzel des arithmetischen Ausdrucks gebildet werden.

Beispiele:

A = ABS (-2) A erhält den absoluten Wert von (-2), d. h. 2.

A = SQRT (9) A erhält den Wert der Quadratwurzel von (9), d. h. 3

Syntax:

< algebraische Funktion > : : =

$\frac{1}{1}$ [ABS | SQRT] (< Skalar >)

5.2.2 Trigonometrische und geometrische Funktionen

Semantik:

Der Name der Funktion bezeichnet die Art des arithmetischen Prozesses, der an dem nachfolgenden in Klammern stehenden arithmetischen Ausdruck durchzuführen ist.

SIN es ist der Sinus des folgenden arithmetischen Ausdrucks zu bilden.

COS es ist der Kosinus des folgenden arithmetischen Ausdrucks zu bilden.

TAN es ist der Tangens des folgenden arithmetischen Ausdrucks zu bilden.

ASIN bezeichnet den Winkel, dessen Sinus durch den folgenden arithmetischen Ausdruck gegeben ist.

ACOS bezeichnet den Winkel, dessen Kosinus durch den folgenden arithmetischen Ausdruck gegeben ist.

ATAN bezeichnet den Winkel, dessen Tangens durch den folgenden arithmetischen Ausdruck gegeben ist.

ANGL bezeichnet den gegen den Drehsinn des Uhrzeigers gemessenen Winkel zwischen der positiven x-Achse und der Geraden, die den Mittelpunkt des spezifizierten Kreises mit dem spezifizierten Punkt verbindet.

DIST bezeichnet den Abstand zwischen den spezifizierten Punkten.

NUM bezeichnet die Anzahl der Punkte im spezifizierten Muster.

Bei trigonometrischen Funktionen ist das Argument ein Winkel im Gradmaß. Bei den Umkehrfunktionen bedeutet das Ergebnis einen Winkel im Gradmaß und zwar in den folgenden Bereichen:

A = ASIN (. . .) $-90^\circ \leq A \leq 90^\circ$

A = ACOS (. . .) $0^\circ \leq A \leq 180^\circ$

A = ATAN (. . .) $-90^\circ \leq A \leq 90^\circ$

Beispiele:

A = SIN (30) A erhält den Wert 0,5

A = COS (30) A erhält den Wert 0,866

A = TAN (45) A erhält den Wert 1,000

A = ASIN (.5) A erhält den Wert 30

A = ASIN (-.5) A erhält den Wert -30

A = ACOS (.866) A erhält den Wert 30

A = ACOS (-.866) A erhält den Wert 150

A = ATAN (1) A erhält den Wert 45

A = ATAN (-1) A erhält den Wert -45

Syntax:

< trigonometrische Funktion > : : =

$\frac{1}{1}$ [SIN | COS | TAN | ASIN | ACOS | ATAN] (< Skalar >)

< geometrische Funktion > : : =

ANGL (< Kreisspezifikator >, < Punktspezifikator >) |

DIST (< Punktspezifikator >, < Punktspezifikator >) |

NUM (< Punktmusterspezifikator >)

5.2.3 Exponentialfunktionen

Semantik:

Der Name für die Exponentialfunktion bezeichnet die Art des Exponentialvorganges, der an dem nachfolgenden in Klammern geschriebenen arithmetischen Ausdruck durchgeführt werden soll.

EXP es wird der Wert von „e“ zu einer Potenz erhoben, die durch den Wert des arithmetischen Ausdrucks gegeben ist.

LOG es wird der natürliche Logarithmus des arithmetischen Ausdrucks gebildet. Der arithmetische Ausdruck muß in diesem Falle einen positiven Wert haben.

Beispiele:

A = EXP (2) A erhält den Wert von e^2 , d. h. 7,3891

A = LOG (2) A erhält den Wert von $\log_e 2$, d. h. 0,6931

Syntax:

< Exponentialfunktion > : : =

$\frac{1}{1}$ [EXP | LOG] (< Skalar >)

5.2.4 Vektorfunktionen

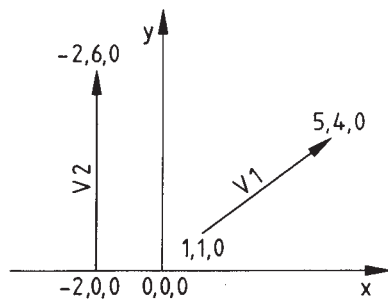
Semantik:

Der Name der LNTH-Vektorfunktion gibt an, daß die Länge des Vektors, dargestellt durch den nachfolgenden in Klammern geschriebenen Vektorspezifikator, zu berechnen ist.

Der Name für die DOT-Vektorfunktion besagt, daß das Skalarprodukt des nachfolgend in Klammern stehenden Paares von Vektorspezifikatoren zu bilden ist. Algebraisch heißt das (Länge des ersten Vektors) x (Länge des zweiten Vektors) x COS (Winkel zwischen den Vektoren).

Das Ergebnis dieser Operation ist ein Skalar.

Beispiel:



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

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

A1 = LNTH (V1) A1 erhält den Wert 5
 A2 = LNTH (V2) A2 erhält den Wert 6
 A3 = DOT (V1, V2) A3 erhält den Wert 18

Syntax:

< Vektorfunktion > : :=

LNTH (< Vektorspezifikator >) |

DOT (< Vektorspezifikator >, < Vektorspezifikator >)

6 Programmdefinitionsanweisungen

6.1 Synonymdefinitionsanweisung

SYN/Folge von Synonym-Schlüsselwortpaaren

Semantik:

Die Synonymdefinitionsanweisung wird in einem Teileprogramm benutzt, um Alternativformen für die Buchstabenfolge von Schlüsselwörtern zu haben. Das in einer SYN-Anweisung gegebene Synonym wird, wenn es später in einem Teileprogramm erscheint, verarbeitet wie das dazugehörige Schlüsselwort. Wenn ein Symbol als Synonym festgelegt ist, kann es nicht in einem anderen Zusammenhang neu definiert werden.

Beispiele:

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

damit ist die folgende Anweisung möglich:

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

Sie wird interpretiert als:

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

Die folgende Anweisung ist nicht mehr erlaubt, da P schon als Synonym definiert wurde:

P = POINT/0, 1,2

Syntax:

< Synonymdefinitionsanweisung > : :=

SYN/ < Synonymparameterliste >

< Synonymparameterliste > : :=

 $\{^k_1 [\text{< nicht indiziertes Symbol >}, \text{< Schlüsselwort >}] \}$

6.2 Reservierungsanweisung für indizierte Symbole

RESERV/Folge von Paarungen zu indizierender Symbole und maximalem Index

Semantik:

Die Reservierungsanweisung gibt an, welche Symbole indiziert sind und wie groß der Index maximal werden kann.

Die Anweisung RESERV muß jedem indizierten Symbol vorausgehen.

Beispiel:

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

Damit ist die folgende Anweisung möglich:

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

Nicht erlaubt ist aber, da P2 der maximale Index 7 zugeordnet wurde,

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

Einschränkungen:

Die Typen der zu indizierenden Symbole und die Anzahl der Paarungen (Symbol und maximaler Index) können implementierungsabhängig sein.

Syntax:

< Reservierungsanweisung > : :=

RESERV/ < Reservierungsparameterliste >

< Reservierungsparameterliste > : :=

 $\{^k_1 [\text{< nicht indiziertes Symbol >}, \text{< Skalar >}] \}$

6.3 Definition eines Makros

Semantik:

Ein Makro wird verwendet, wenn eine Gruppe von Teileprogrammanweisungen mehr als einmal in einem Teileprogramm benutzt werden soll. Die Makrodefinition ermöglicht es, dieser Anweisungsgruppe für eine Bezugnahme durch spätere Aufrufe (CALL) ein Symbol zuzuweisen.

Eine Anweisung in der Form

a. MAC1 = MACRO

oder b. MAC2 = MACRO/A, B, C

oder c. MAC3 = MACRO/A = 1.2, B = P1, C = 5

bezeichnet den Anfang einer Folge von Teileprogrammanweisungen, die zu einem Makro zusammengefaßt werden.

Die Makroanweisung kann eine Parameterliste (wie in b. oder c.) enthalten und jeder Parameter dieser Liste kann einen Normal-Wert erhalten (wie in c.). Parameter können überall innerhalb eines Makros benutzt werden und erhalten nur Werte, wenn das Makro aufgerufen wird. Normal-Werte werden nur bei Fehlen eines tatsächlichen Wertes im Aufruf benutzt.

Das Ende der Makrofolge von Teileprogrammanweisungen wird durch die Anweisung TERMAC gekennzeichnet.

Beispiele:

MAC1 = MACRO

MAC2 = MACRO/A, B

GOTO/P1

GOTO/A

GODLTA/0.5

GODLTA/B

GODLTA/-0.5

GODLTA/-B

TERMAC

TERMAC

Einschränkungen:

Die folgenden Teileprogrammanweisungen dürfen nicht in einer Makrofolge erscheinen.

PARTNO

FINI

LOOPST

LOOPND

MACRO

Außerdem kann implementierungsabhängig die Verwendung von SYN/ und RESERV/ innerhalb eines Makros verboten sein.

Syntax:

< Makrodefinitionsanweisung > : : =
 < nicht indiziertes Symbol >
 = MACRO [/< Parameterliste >]
 < Parameterliste > : : =
 $\{ \{ \{ \text{< nicht indiziertes Symbol >, | < nicht indiziertes Symbol > = < Normalwert >, | \}$
 < Normalwert > : : = < nicht indiziertes Symbol > |
 < Skalar > | < arithmetischer Ausdruck > | < Schlüsselwort >

Querbezüge:

PARTNO
 FINI
 LOOPST
 LOOPND
 CALL

7 Programmausführungsanweisungen

7.1 Beginn und Ende eines Teileprogrammes

7.1.1 Teileprogrammanfangsanweisung

PARTNO/Textzeichenfolge
 PARTNO 'Textzeichenfolge'

Semantik:

Die PARTNO-Anweisung wird für die Teileprogramm-kennzeichnung benutzt und bezeichnet den Beginn des Teileprogrammes.

Beispiel:

PARTNO 'Ventilgehäuse, Zeichnung Nr 72445'

Syntax:

< Teileprogrammanfangsanweisung > : : = PARTNO
 [/< Textzeichenfolge > | '< Textzeichenfolge >']

7.1.2 Teileprogrammabschlußanweisung

FINI

Semantik:

Die Anweisung FINI bezeichnet das Ende eines Teileprogrammes

Syntax:

< Teileprogrammabschlußanweisung > : : = FINI

7.2 Aufruf und Aufrufunterdrückung von Postprozessoren

7.2.1 Aufruf eines Postprozessors

MACHIN/Parameterliste

Mit der MACHIN-Anweisung wird ein Postprozessor aufgerufen.

Jeder Postprozessor ist gekennzeichnet durch einen Namen und gegebenenfalls durch eine Kennzeichnungsnummer. Zusätzliche Parameter, die spezielle Postprozessormöglichkeiten auslösen, sind implementierungsabhängig und können gegebenenfalls angegeben werden.

In einem Teileprogramm sind mehrere Postprozessoraufrufe erlaubt.

Zusammenhang:

Der Aufruf von Postprozessoren wird unterdrückt durch eine NOPOST-Anweisung.

Beispiel:

MACHIN/EXPOST, 0

Syntax:

< Postprozessoraufwurf > : : = MACHIN/< nicht indiziertes Symbol > [< Parameterliste >]⁶⁾

Querbezug:

NOPOST

7.2.2 Aufrufunterdrückung von Postprozessoren

NOPOST

Semantik:

Es werden alle Postprozessoraufrufe eines Teileprogrammes unterdrückt.

Syntax:

< Aufrufunterdrückung von Postprozessoren > : : = NOPOST

Querbezug:

MACHIN

7.3 Eingabe-/Ausgabeeanweisungen

7.3.1 Druckanweisung

PRINT/ON

PRINT/OFF

PRINT/Ø

PRINT/PAGE

PRINT/2, S₁, . . . , S_n

PRINT/NONAME, S₁, . . . , S_n

PRINT/S₁, . . . , S_n

PRINT/3, ALL

PRINT/ALL

Semantik:

Die Druckanweisungen steuern das Ausdrucken der intern gespeicherten Werte definierter Größen (kanonische Form).

Mit der Anweisung PRINT/ON kann der Programmierer den Ausdruck der kanonischen Form erhalten für alle Größen, die nach dieser Anweisung und vor dem Gegenbefehl PRINT/OFF definiert werden.

Die Anweisung PRINT/Ø oder PRINT/PAGE bewirkt Seitenwechsel.

Die Anweisung PRINT/2, . . . oder PRINT/NONAME, . . . bewirkt den Ausdruck der kanonischen Form oder des Skalarwertes der spezifischen Größen ohne deren Symbol.

Die Anweisung PRINT/3, . . . bewirkt den Ausdruck der kanonischen Form oder des Skalarwertes der spezifizierten Größen mit ihrem Symbol.

Die Anweisung PRINT/3, ALL hat dieselbe Wirkung wie die Anweisung PRINT/3, . . . betrifft aber sämtliche zuvor definierten Größen.

Beispiel:

PRINT/3, L1, C2, ANGEL4, PT2

⁶⁾ Die Syntax des implementierungsabhängigen Teiles der Parameterliste ist vom Postprozessorersteller festzulegen.

Syntax:

< Druckenweisung > : := PRINT/< Druckparameterliste >

< Druckparameterliste > : :=

$\frac{1}{1}$ [ON | OFF] | $\frac{1}{1}$ [Ø | PAGE] |

$\frac{1}{1}$ [2 | 3], $\left\{ \frac{k}{1} \right\}$ [< Symbol >,] } | [3,] ALL |

NONAME, $\left\{ \frac{k}{1} \right\}$ [< Symbol >,] }

7.3.2 Allgemeine Ausgabeanweisung

PUNCH/m, $S_1, \dots, S_n$

PUNCH/ALL

PUNCH/m, ALL

PUNCH/ $S_1, \dots, S_n$

PUNCH/ON

PUNCH/OFF

Semantik:

Die allgemeine Ausgabeanweisung steuert die Ausgabe von intern gespeicherten Werten definierter Größen (kanonische Form) in einer für den Prozessor lesbaren Form. Das Ausgabemedium kann durch den Parameter m gesteuert werden und ist implementierungsabhängig.

$S_1, \dots, S_n$ sind die auszugebenden Größen

ON bewirkt den Beginn der Ausgabe

OFF beendet die Ausgabe

Die Anweisung PUNCH/ALL bewirkt die Ausgabe sämtlicher Größen, die vor dieser Anweisung definiert werden.

Syntax:

< Allgemeine Ausgabeanweisung > : := PUNCH/

< Ausgabeparameterliste >

< Ausgabeparameterliste > : :=

$\frac{1}{1}$ [ON | OFF] | [< Ziffer >,] ALL |

[< Ziffer >,] $\left\{ \frac{k}{1} \right\}$ [< Symbol >,] }

Querbezug:

READ

7.3.3 Leseanweisung

READ/1

READ/ALL

READ/1, $S_1, \dots, S_n$

READ/ $S_1, \dots, S_n$

READ/2, Segment-Name

READ/Segment-Name

Semantik:

Die Leseanweisung steuert das Einlesen von Daten, die in vom Prozessor lesbarer Form (kanonischer Form) vorliegen. Diese Daten können vorher durch die allgemeine Ausgabeanweisung (PUNCH) erzeugt worden sein.

READ/1, ... und READ/ALL sind allgemeine Leseanweisungen. Durch die Anweisung READ/1, $S_1, \dots, S_n$ wird jeder gelesenen kanonischen Form oder jedem gelesenen Skalarwert ein neues Symbol zugewiesen. $S_1, \dots, S_n$ sind die neuen Symbole.

Die Anweisung READ/2, Segment-Name oder READ/Segment-Name

ist implementierungsabhängig und dient dazu, zuvor extern gespeicherte Anweisungen in das Teileprogramm

einzuführen. Implementierungsabhängig ist ferner, wo und wie die Anweisungen, die mit READ/2, ... gelesen werden sollen, zuvor extern zu speichern sind.

Einschränkungen:

Die Anweisungen, die durch READ/2, ... eingefügt werden, dürfen die Anweisung READ/2, ... nicht enthalten.

Der erste Aufruf eines Systemmakros (extern gespeicherte Anweisungsfolge) darf nicht durch eine Anweisung erfolgen, die durch READ/2, ... eingefügt wurde.

Ein Systemmakro darf keine READ/2, ... Anweisung enthalten.

Syntax:

< Leseanweisung > : := READ/< Leseparameterliste >

< Leseparameterliste > : := $\frac{1}{1}$ [1 | ALL] |

[1,] $\left\{ \frac{k}{1} \right\}$ [< Symbol >,] } | [2,] $\frac{6}{1}$ [< Zeichen >]

Querbezug:

PUNCH

7.3.4 CLDATA-Druckenweisung

CLPRNT

CLPRNT/ON

CLPRNT/OFF

CLPRNT/OMIT, n

Semantik:

Die Anweisung CLPRNT bewirkt den Ausdruck des gesamten CLDATA-Textes.

ON bewirkt den Beginn des Ausdrucks.

OFF beendet das Ausdrucken.

Die Anweisung CLPRNT/OMIT, n bewirkt, daß nach jeder Satzausgabe n Sätze unterdrückt werden. Der erste und letzte Satz werden stets gedruckt.

Syntax:

< CLDATA-Druckenweisung > : :=

CLPRNT [/< CLDATA-Druckparameterliste >]

< CLDATA-Druckparameterliste > : :=

$\frac{1}{1}$ [ON | OFF | OMIT, n]

7.3.5 Anweisung zur Kennzeichnung der gedruckten Ausgabe

TITLES/Textzeichenfolge (bis zu 65 Zeichen)

TITLES/RIGHT Textzeichenfolge (bis zu 60 Zeichen)

TITLES/RIGHT 'Textzeichenfolge'

Semantik:

Die Anweisung TITLES/. ... bewirkt, daß die folgende Textzeichenfolge unmittelbar, beginnend am linken Druckrand ausgedruckt wird und sich maximal über 65 Druckpositionen erstreckt.

Die Anweisung TITLES/RIGHT ... bewirkt, daß die folgende Textzeichenfolge unmittelbar, beginnend ab der 66. Druckposition ausgedruckt wird und sich maximal bis zur 125. Druckposition erstreckt.

Beispiel:

PRINT/PAGE

TITLES

PRINT/S1, S2, S3, S4

Einschränkungen:

Um eine Kennzeichnung zu erhalten, die sich von der 1. bis einschließlich zur 126. Druckposition erstreckt, ist die Anweisung TITLES/RIGHT zu benutzen auf die unmittelbar die Anweisung TITLES/ folgt.

Syntax:

< Anweisung zur Kennzeichnung der gedruckten Ausgabe > : : = TITLES/< Textzeichenfolge > |
TITLES< Begrenzungszeichen für Textzeichenfolge >
< Textzeichenfolge > < Begrenzungszeichen für Textzeichenfolge > | TITLES/RIGHT, < Textzeichenfolge > |
TITLES/RIGHT< Begrenzungszeichen für Textzeichenfolge > < Textzeichenfolge > < Begrenzungszeichen für Textzeichenfolge >

Querbezug:

PRINT

7.4 Sprunganweisungen**7.4.1 Schleifenanfangs- und Schleifenendeanweisung**

LOOPST

LOOPND

Semantik:

Die Anweisungen LOOPST und LOOPND bestimmen den Anfang und das Ende einer Schleife. Die Anweisungen zwischen LOOPST und LOOPND sind Teil der Schleife.

Innerhalb einer Schleife kann von der normalen sequentiellen Ausführung des Teileprogramms durch Sprungbefehle (IF (siehe Abschnitt 7.4.3), JUMPTO (siehe Abschnitt 7.4.2)) abgewichen werden.

Zusammenhang:

Die Anweisung MACRO impliziert LOOPST und die Anweisung TERMAC impliziert LOOPND, d. h. daß ein MACRO definitionsgemäß alle Eigenschaften einer Schleife besitzt.

Beispiel:

J = 1

LOOPST

A)

.....

IF (J - 100) B, C, C

B) J = J + 1

JUMPTO/A

C) LOOPND

Einschränkungen:

Die Anweisungen LOOPST und LOOPND müssen immer paarweise benutzt werden. Ein Makro kann innerhalb einer Schleife nicht definiert werden und umgekehrt.

Syntax:

< Schleifenanfangsanweisung > : : = LOOPST

< Schleifenendeanweisung > : : = LOOPND

Querbezug:

IF

JUMPTO

MACRO

TERMAC

7.4.2 Sprunganweisung

JUMPTO/Marke

Semantik:

Durch die Anweisung JUMPTO/Marke wird die normale sequentielle Ausführung des Teileprogrammes unterbrochen, als nächste wird die Anweisung, die die Marke der Sprunganweisung trägt, ausgeführt.

Vor- und Rückverzweigungen sind erlaubt.

Die Anweisungen LOOPND und TERMAC können Sprungziele sein, jedoch nicht die Anweisungen MACRO und LOOPST.

Zusammenhang:

Die Sprunganweisung und die Anweisung, die die Marke der Sprunganweisung trägt, müssen in demselben Schleifen- oder Makrobereich liegen.

Beispiel:

LOOPST

....

A)

....

JUMPTO/A

LOOPND

LOOPST

....

A)

....

IF (WERT-2) A, B, C

....

C)

....

B) LOOPND

Einschränkungen:

Verzweigungen auf Anweisungen außerhalb des Schleifenbereiches oder des Makrobereiches sind nicht erlaubt.

Die Sprunganweisung selbst kann für diese Verzweigung nicht Sprungziel sein.

Syntax:

< Sprunganweisung > : : = JUMPTO/< Marke >

Querbezug:**Markendefinition**

LOOPST

LOOPND

MACRO

TERMAC

7.4.3 Bedingte Sprunganweisung

IF(n) Marke 1, Marke 2, Marke 3

Semantik:

Die normale sequentielle Ausführung eines Teileprogrammes kann durch die bedingte Sprunganweisung in Abhängigkeit von dem Wert eines arithmetischen Ausdruckes oder eines Skalars (n) geändert werden. Ist der Wert von (n) kleiner als null, wird zu der Anweisung mit der Marke 1 verzweigt. Ist der Wert von (n) gleich null, wird zu der Anweisung mit der Marke 2, und ist der Wert von (n) größer als null, wird zu der Anweisung mit der Marke 3 verzweigt. Vor- und Rückverzweigungen sind erlaubt. Die Anweisungen LOOPND und TERMAC

können Sprungziel sein, jedoch nicht die Anweisungen MACRO und LOOPST.

Zusammenhang:

Siehe JUMPTO

Einschränkungen:

Verzweigungen auf eine Anweisung außerhalb des Schleifen- oder Makrobereiches, der den bedingten Sprungbefehl enthält, sind nicht gestattet.

Syntax:

< Bedingter Sprungbefehl > : := IF (< arithmetischer Ausdruck >) < Marke >, < Marke >, < Marke >

Querbezug:

Markendefinition

Arithmetischer Ausdruck

LOOPST

LOOPND

MACRO

TERMAC

7.4.4 Verzweigungsanweisung

TRANTO/Marke

Semantik:

Als nächstes wird die Anweisung, die die in der Verzweigungsanweisung angegebene Marke trägt, ausgeführt.

Zusammenhang:

Die Anweisung TRANTO/ ... wird nur in Verbindung mit Werkzeugbewegungen, die durch mehrere Kontrollflächen (multiple check surfaces) begrenzt sind, benutzt.

Einschränkungen:

Die Anweisung TRANTO/ ... läßt nur eine Vorwärtsverzweigung zu, auch dann, wenn sie innerhalb einer Schleife oder eines Makros benutzt wird. Ziel einer TRANTO-Verzweigung darf nur eine Bewegungsanweisung oder eine Postprozessoranweisung sein.

Syntax:

< Verzweigungsanweisung > : := TRANTO/< Marke >

Querbezug:

LOOPST

LOOPND

MACRO

TERMAC

Markendefinition

7.5 COPY- und Index-Anweisung

7.5.1 COPY-Anweisung

COPY/i, SAME, n

COPY/i, TRANSL, x, y, z, n

XYROT

COPY/i, YZROT, Winkel, n

XZROT

COPY/i, MODIFY, Matrixsymbol, n.

Semantik:

Durch die Anweisung COPY/ ... werden die zwischen den Anweisungen INDEX/i und INDEX/i, NOMORE oder INDEX/i und COPY/i, ... liegenden CLDATA-Sätze n mal kopiert und die Werkzeugkoordinaten gemäß den angegebenen Modifikatoren transformiert.

Bedeutung der Modifikatoren:

SAME

keine Transformation

TRANSL, x, y, z

Koordinatentranslation

XYROT, α

Drehung in der XY-Ebene um den Winkel α

YZROT, α

Drehung in der YZ-Ebene um den Winkel α

XZROT, α

Drehung in der XZ-Ebene um den Winkel α

MODIFY, Matrixsymbol

Allgemeine Transformation

Zusammenhang:

Der Wirkungsbereich einer COPY-Anweisung kann eingebettet sein in den Wirkungsbereich einer TRACUT-Anweisung oder umgekehrt. In jedem Fall werden durch die TRACUT-Anweisung nur die ursprünglichen Werkzeugkoordinaten transformiert. Die TRACUT-Anweisung wirkt nicht kumulativ.

Beispiel:

....

INDEX/7

....

GOTO/x, y, z

....

COPY/7, MODIFY, MATR, 2

....

COPY/7, TRANSL, x₁, y₁, z₁, 1

....

J = 5

INDEX/J

....

INDEX/J, NOMORE

....

COPY/J, XYROT, 90, 3

....

COPY/7, SAME, 2

....

Einschränkungen:

COPY-Schleifen dürfen sich nicht überlappen, können jedoch geschachtelt sein.

Syntax:

< COPY-Anweisung > : := COPY/< COPY-Parameterliste >

< COPY-Parameterliste > : := < Marke >,

[¹ [SAME|TRANSL, < Skalar >, < Skalar >, < Skalar >]]

[¹ [XYROT|YZROT|XZROT], < Skalar >|MODIFY, < Matrix Spezifikator >], < Skalar >

Querbezug:

INDEX

TRACUT

MATRIX

7.5.2 INDEX-Anweisung

INDEX/i

INDEX/i, NOMORE

Semantik:

Durch die Anweisungen INDEX/i und INDEX/i, NOMORE wird eine Gruppe von Anweisungen begrenzt, auf die durch eine COPY-Anweisung Bezug genommen werden kann. Diese Gruppe von Anweisungen wird COPY-Schleife genannt.

Zusammenhang:

Die Anweisung INDEX/i, NOMORE kann durch eine oder mehrere COPY/i ... Anweisungen ersetzt werden. Im letzten Fall wird durch jede COPY-Anweisung eine COPY-Schleife definiert. Die so definierten Schleifen haben einen gemeinsamen Ausgangspunkt.

Beispiel:

siehe COPY- Anweisung

Einschränkungen:

COPY-Schleifen dürfen sich nicht überlappen, können jedoch geschachtelt werden. Der gleiche Indexwert i, darf nur in einem Paar von INDEX-Anweisungen oder nur in den von INDEX-COPY-Anweisungen begrenzten Schleifen mit gemeinsamen Ausgangspunkt verwendet werden.

Syntax:

< INDEX-Anweisung > : : =
INDEX/< Skalar > [, NOMORE]

Querbezug:

COPY

7.6 Makroaufruf

CALL/Symbol

CALL/Symbol, Parameterliste

Semantik:

Durch die CALL-Anweisung gelangt ein im Teileprogramm definiertes Makro zur Ausführung. Auf das Makro, dem das im Makroaufruf genannte Symbol zugeordnet ist, wird Bezug genommen und die CALL-Anweisung durch die Folge von Anweisungen, die ein Makro bilden (zwischen MACRO und TERMAC), ersetzt. Die Formalparameter innerhalb der Makroanweisung erhalten die Werte der Aktualparameter, die in der CALL-Anweisung spezifiziert sind. Beim Fehlen von aktuellen Werten im Aufruf werden die Normalparameter, die in der MACRO-Anweisung spezifiziert sind, benutzt.

Die CALL-Anweisung kann auch benutzt werden, um ein außerhalb des Teileprogrammes definiertes Makro (Systemmakro) aufzurufen.

Die CALL-Anweisung kann auch zum Aufruf eines Sonderprogrammes benutzt werden. In diesem Fall ist die Parameterliste den Anforderungen des Sonderprogrammes anzupassen.

Zusammenhang:

Die aktuellen Parameter einer CALL-Anweisung gelten nur für den speziellen Aufruf und verändern in keiner Weise die Anweisungen des gespeicherten Makros.

Die CALL-Anweisung kann an jeder Stelle eines Teileprogrammes auftreten, sogar in einer Schleife (zwischen LOOPST und LOOPND) und auch in einem Makro (zwischen MACRO und TERMAC). Letzteres ist eine Makro-Schachtelung.

Beispiel:

....

....

MACNAM = MACRO/A, B = 10, PL, H, P

....

....

CALL/BSURP, ... (Parameterliste) ...

....

....

TERMAC

....

....

PLS = PLANE/ ...

HS = arithmetischer Ausdruck

....

....

CALL/MACNAM, A = 5, PL = PLS, H = HS, P = POINT

Einschränkungen:

In einer CALL-Anweisung innerhalb eines Makros dürfen sich die Parameter der Parameterliste von den im Makro definierten Parameternamen unterscheiden.

Syntax:

< Makroaufruf > : : = CALL/< nicht indiziertes
Symbol > [, < Parameterliste >]

< Parameterliste > : : =

{₁^k [< nicht indiziertes Symbol >, | < nicht indiziertes
Symbol > = < Aktualwert, >]}

< Aktualwert > : : = < nicht indiziertes Symbol > |

< Skalar > | < arithmetischer Ausdruck > | < Schlüssel-
wort >

Querbezug:

MACRO

TERMAC

LOOPST

LOOPND

7.7 Kommentaranweisung

REMARK/Textzeichenfolge

REMARK 'Textzeichenfolge'

Semantik:

Der in dieser Anweisung angegebene Text erscheint als Kommentar in dem Kontrollausdruck des Teileprogramms.

Beispiel:

REMARK/ES FOLGEN GEOMETRISCHE
DEFINITIONEN

Syntax:

< Kommentaranweisung > : : = REMARK
[/< Textzeichenfolge > | '< Textzeichenfolge >']

7.8 Postprozessordruckanweisung

PPRINT 'Textzeichenfolge'

PPRINT/Textzeichenfolge

Semantik:

Der Text dieser Anweisung wird vom Prozessor an den Postprozessor übergeben. Dadurch wird es dem Teileprogrammierer ermöglicht, die vom Postprozessor erstellten Listen mit einem Kommentar zu versehen.

Beispiel:

PPRINT/WERKZEUGWECHSEL

Syntax:

< Postprozessordruckanweisung > : : = PPRINT
[/ < Textzeichenfolge > | ' < Textzeichenfolge > ']

8 Geometrische Definitionsanweisungen

8.0 Allgemeine Bemerkungen zu geometrischen Definitionsanweisungen

8.0.1 Einschränkungen

Eine syntaktisch gültige Anweisung kann eine geometrische sinnlose Situation darstellen. Es ist nicht möglich, alle diese Situationen bzw. deren korrekte Behandlung durch den Prozessor abzudecken. Einschränkungen zur Definition und Verwendung geometrischer Anweisungen sind nur in Sonderfällen gegeben. Ein Beispiel ist der Radius in der Kreisdefinitionsanweisung. Wenn der Radius durch einen geschachtelten arithmetischen Ausdruck gegeben ist, kann er einen negativen Wert erhalten. Dieser Wert kann von einigen Prozessoren zurückgewiesen werden, während andere den absoluten Wert nehmen.

8.0.2 Koordinatensystem und Dimensionen

X, Y und Z in geometrischen Definitionsanweisungen beziehen sich auf Abstände im zugrunde liegenden Koordinatensystem (siehe Abschnitt 3). Es können auch lokale Bezugssysteme benutzt werden (siehe REFSYS, Abschnitt 8.1).

Das Maßsystem ist mm, Grad orientiert. Grundsätzlich werden Dezimalteile dieser Basismaße angegeben

Beispiel:

8.75° ± 8° 45'

8.0.3 Häufig benutzte Modifikatoren

8.0.3.1 Für die eindeutige Definition eines geometrischen Elements müssen in zahlreichen Definitionen folgende Modifikatoren benutzt werden:

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

8.0.3.2 Die Modifikatoren IN oder OUT vor einem Symbol besagen, daß das zu bestimmende Element gänzlich innerhalb bzw. außerhalb des genannten Elementes liegt.

8.0.3.3 Die Modifikatoren XLARGE, XSMALL, ... bezeichnen die Position des zu bestimmenden Elementes gegenüber dem nächsten genannten Element. Zum Beispiel zeigt XLARGE, daß das zu bestimmende Element in bezug auf X größere Werte hat als eine andere sonst mögliche Lösung. In vielen Fällen kann durch verschiede-

dene Modifikatoren die gleiche Lösung ausgewählt werden.

8.0.3.4 Die Modifikatoren RIGHT und LEFT werden benutzt, um zwischen zwei sonst möglichen Lösungen auswählen zu können. Die Modifikatoren gelten in Blickrichtung vom ersten zum zweiten genannten Element.

8.0.3.5 Die Modifikatoren POSX und NEGX bezeichnen die positive oder negative X-Richtung. Dasselbe gilt für POSY und NEGY in der Y-Richtung.

8.0.3.6 Die Modifikatoren LARGE und SMALL werden benutzt, um unter zwei möglichen Lösungen die größere oder kleinere auszuwählen.

8.0.3.7 Die Modifikatoren CLW und CCLW bezeichnen den Uhrzeiger- bzw. Gegenuhrzeigersinn.

8.0.4 Spezifikatoren

Der Ausdruck Spezifikator oder 'Spez' wird gebraucht in Verbindung mit einem Symbol, das ein bestimmtes geometrisches Element bezeichnet.

Die folgenden Ausdrücke werden verwendet:

Punkt-Spezifikator

Punktmuster-Spezifikator

Ebenen-Spezifikator

Vektor-Spezifikator

Kreis-Spezifikator

Zylinder-Spezifikator

Kugel-Spezifikator

Kegel-Spezifikator

Ellipsen-Spezifikator

Hyperbel-Spezifikator

Kegelschnitt-Spezifikator

Flächen 2. Ordnung-Spezifikator

Tabcyl-Spezifikator

Matrix-Spezifikator

Regelflächen-Spezifikator

Der Ausdruck Flächen-Spezifikator wird als Sammelbegriff verwendet.

8.1 Lokales Bezugssystem

REFSYS/Bezugssystemparameterliste

Semantik:

Alle in einem Teileprogramm definierten Elemente werden auf das zugrunde liegende Koordinatensystem bezogen. Es kann aber ein lokales Bezugssystem festgelegt werden in Form eines dreiachsigen kartesischen Koordinatensystems, das gegenüber dem zugrunde liegenden System gedreht und/oder verschoben ist. Die Gültigkeit eines lokalen Bezugssystems wird durch eine REFSYS-Anweisung eingeleitet und erstreckt sich bis zur nächsten REFSYS-Anweisung. Alle Arten von Anweisungen sind innerhalb des Gültigkeitsbereiches zulässig. Es sind nur geometrische Definitionen betroffen. Die Ergebnisse der geometrischen Definitionen sind kanonische Formen im Basis-Koordinatensystem.

Syntax:

< Bezugssystemerklärung > : : = REFSYS/
< Bezugssystemparameterliste >

8.1.1 Festlegung eines lokalen Bezugssystems

REFSYS/Matrix

Semantik:

Die Matrix ist eine zuvor definierte Transformationsmatrix, die geometrische Definitionen vom lokalen Bezugssystem auf das zugrunde liegende Koordinatensystem transformiert.

Zusammenhang:

Ein noch geltendes lokales Bezugssystem verliert durch eine neue REFSYS/Matrix-Anweisung seine Gültigkeit.

Syntax:

< Bezugssystemparameterliste > : := < Matrix Spezifikator >

8.1.2 Ende der Gültigkeit eines lokalen Bezugssystems

REFSYS/NOMORE

Semantik:

Beendet die Gültigkeit eines lokalen Bezugssystems

Syntax:

< Bezugssystemparameterliste > : := NOMORE

Querbezug:

REFSYS/Matrix

8.2 Festlegung eines impliziten Z-Wertes

ZSURF/Flächenparameterliste

Semantik:

Mit Hilfe einer ZSURF-Anweisung wird nachfolgenden Punkt- und Punktmusterdefinitionen, die keinen expliziten Wert der Z-Koordinate haben, ein Z-Wert zugewiesen.

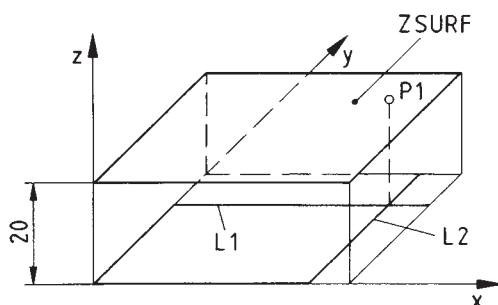
Syntax:

< Festlegung eines impliziten Z-Wertes > : := ZSURF/
< Flächenparameterliste >

8.2.1 Festlegung eines impliziten Z-Wertes durch eine Ebene parallel zur XY-Ebene im Abstand h

ZSURF/h

Beispiel:



ZSURF/20

P1 = POINT/INTOF, L1, L2

Syntax:

< Flächenparameterliste > : := < Skalar >

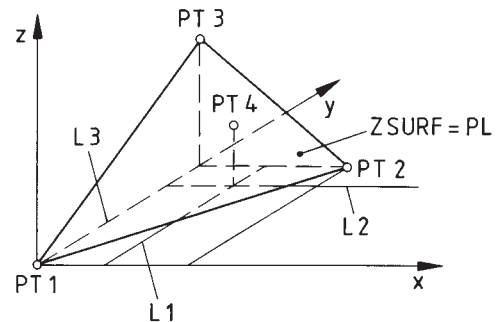
8.2.2 Festlegung eines impliziten Z-Wertes durch eine Ebene

ZSURF/Ebene

Semantik:

Der implizite Z-Wert ist eine lineare Funktion der X- und Y-Koordinaten

Beispiel:



PL = PLANE/PT1, PT2, PT3

ZSURF/PL

PT4 = POINT/INTOF, L1, L2

Syntax:

< Flächenparameterliste > : := < Ebenen-Spezifikator >

8.2.3 Festlegung eines impliziten Z-Wertes durch eine Zylinderfläche

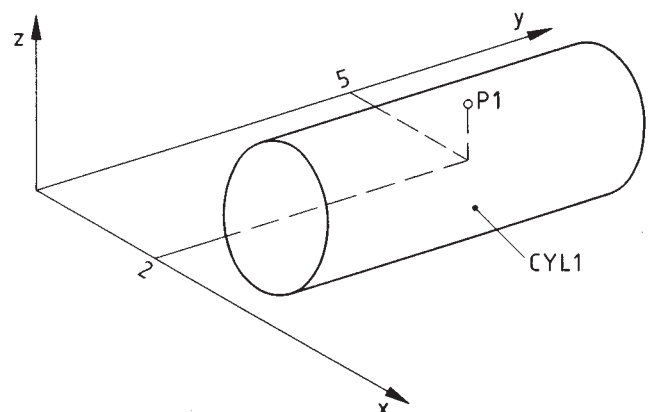
ZSURF/Zylinder, ZLARGE
ZSMALL

ZSURF/Zylinder

Semantik:

Der implizite Z-Wert ist der den X-, Y-Koordinaten zugeordnete Z-Wert des Zylinders. Mit Hilfe der Modifikatoren wird zwischen zwei möglichen Z-Werten unterschieden. Wenn kein Modifikator angegeben ist, wird ZLARGE angenommen.

Beispiel:



ZSURF/CYL1, ZLARGE

P1 = POINT/2, 5

Syntax:

< Flächenparameterliste > : := < Zylinder-Spezifikator >
[,ZLARGE | ,ZSMALL]

8.2.4 Festlegung eines impliziten Z-Wertes durch eine Kugelfläche

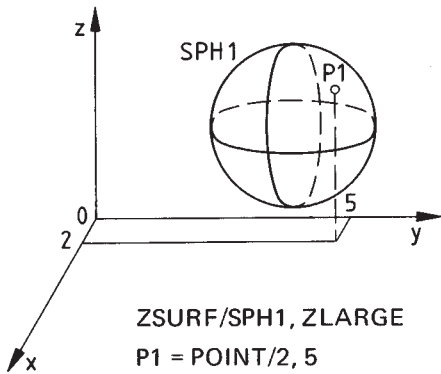
ZSURF/Kugel, ZLARGE
ZSMALL

ZSURF/Kugel

Semantik:

Der implizite Z-Wert ist der den XY-Koordinaten zugeordnete Z-Wert der Kugel. Mit Hilfe der Modifikatoren wird zwischen zwei möglichen Z-Werten unterschieden. Wenn kein Modifikator angegeben ist, wird ZLARGE angenommen.

Beispiel:



Syntax:

< Flächenparameterliste > : : = < Kugel-Spezifikator >
[,ZLARGE | ,ZSMALL]

8.2.5 Festlegung eines impliziten Z-Wertes durch eine Kegelfläche

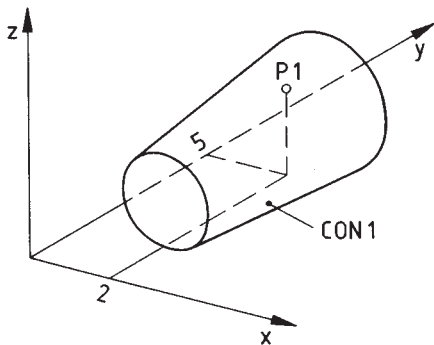
ZSURF/Kegel, ZLARGE
ZSMALL

ZSURF/Kegel

Semantik:

Der implizite Z-Wert ist der den XY-Koordinaten zugeordnete Z-Wert des Kegels. Mit Hilfe der Modifikatoren wird zwischen zwei möglichen Z-Werten unterschieden. Wenn kein Modifikator angegeben ist, wird ZLARGE angenommen.

Beispiel:



ZSURF/CON1, ZLARGE
P1 = POINT/2, 5

Syntax:

< Flächenparameterliste > : : = < Kegel-Spezifikator >
[,ZLARGE | ,ZSMALL]

8.2.6 Festlegung eines impliziten Z-Wertes durch eine allgemeine Fläche 2. Ordnung

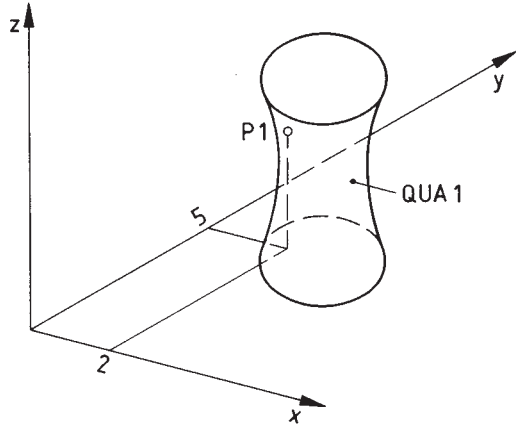
ZSURF/Fläche 2. Ordnung, ZLARGE
ZSMALL

ZSURF/Fläche 2. Ordnung

Semantik:

Der implizite Z-Wert ist der den XY-Koordinaten zugeordnete Z-Wert der allgemeinen Fläche 2. Ordnung. Mit Hilfe der Modifikatoren wird zwischen zwei möglichen Z-Werten unterschieden. Wenn kein Modifikator angegeben ist, wird ZLARGE angenommen.

Beispiel:



ZSURF/QUA1, ZLARGE
P1 = POINT/2, 5

Syntax:

< Flächenparameterliste > : : = < Fläche 2. Ordnung-Spezifikator >
[,ZLARGE | ,ZSMALL]

8.3 Definition eines Punktes (POINT)

POINT/Punktparameterliste

Einschränkungen:

Der Punkt hat drei Koordinaten X, Y, Z. Wenn die Z-Koordinate nicht explizit angegeben ist, wird sie aus der gültigen ZSURF-Anweisung berechnet. Ist keine ausdrückliche ZSURF-Anweisung gegeben, wird Z = 0 angenommen.

Zusammenhang:

Die Punktdefinition ist abhängig von der ZSURF-Anweisung, wenn der Punkt nicht durch seine drei Koordinaten oder als Schnittpunkt dreier Ebenen definiert ist.

Syntax:

< Punktdefinitionsanweisung > : : = < Symbol > =
POINT/< Punktparameterliste >

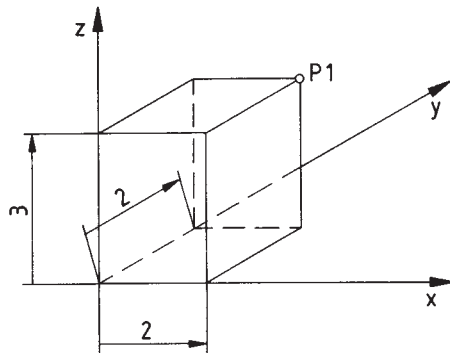
8.3.1 Definition eines Punktes durch seine kartesischen Koordinaten

POINT/X, Y, Z
POINT/X, Y

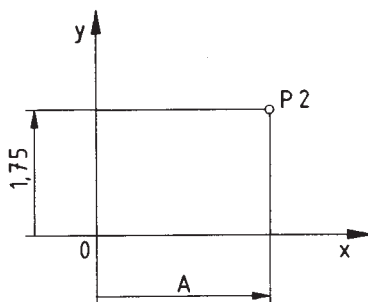
Semantik:

Die Koordinatenwerte des Punktes sind in der Reihenfolge X, Y, Z anzugeben.

Beispiel:



$$P1 = \text{POINT}/2, 2, 3$$



$$A = 2.25$$

$$P2 = \text{POINT}/A, 1.75$$

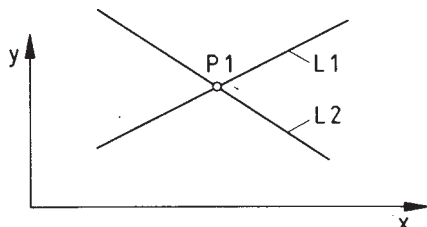
Syntax:

< Punktparameterliste > : := < Skalar >, < Skalar >
[, < Skalar >]

8.3.2 Definition eines Punktes als Schnittpunkt zweier Geraden

POINT/INTOF, Gerade 1, Gerade 2

Beispiel:



$$P1 = \text{POINT}/\text{INTOF}, L1, L2$$

Einschränkungen:

Die beiden Geraden müssen sich voneinander unterscheiden und dürfen nicht parallel sein.

Syntax:

< Punktparameterliste > : := INTOF,
< Geraden-Spezifikator >, < Geraden-Spezifikator >

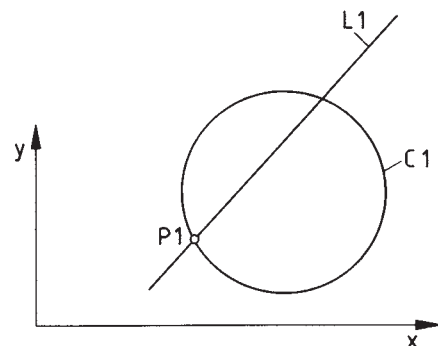
8.3.3 Definition eines Punktes als Schnittpunkt einer Geraden und eines Kreises

XLARGE
POINT/XSMALL, INTOF, Gerade, Kreis
YLARGE
YSMALL

Semantik:

Von den zwei möglichen Lösungen wird durch den Modifikator XLARGE, (XSMALL, ...) eine ausgewählt.

Beispiel:



$$P1 = \text{POINT}/\text{XSMALL}, \text{INTOF}, L1, C1$$

Einschränkungen:

Die Gerade und der Kreis müssen sich schneiden oder berühren. Wenn die Gerade parallel zu einer Koordinatenachse verläuft, können nur die mit dieser Achse in Beziehung stehenden Modifikatoren benutzt werden. YLARGE, YSMALL dürfen beispielsweise nicht benutzt werden, wenn die Gerade parallel zur X-Achse verläuft.

Syntax:

< Punktparameterliste > : :=
¹ [XLARGE | XSMALL | YLARGE | YSMALL], INTOF,
< Geraden-Spezifikator >, < Kreis-Spezifikator >

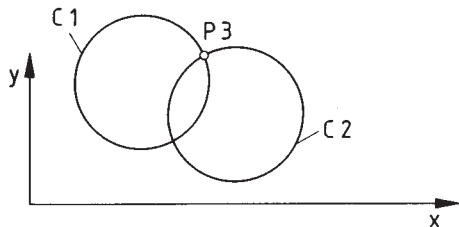
8.3.4 Definition eines Punktes als Schnittpunkt zweier Kreise

XLARGE
POINT/XSMALL, INTOF, Kreis 1, Kreis 2
YLARGE
YSMALL

Semantik:

Von den möglichen Lösungen wird durch den Modifikator XLARGE, (XSMALL, ...) eine ausgewählt.

Beispiel:



P3 = POINT/YLARGE, INTOF, C1, C2

Einschränkungen:

Die beiden Kreise müssen sich schneiden oder berühren.

Wenn die durch die Mittelpunkte der beiden Kreise definierten Gerade parallel zu einer Koordinatenachse verläuft, können nur die zu der anderen Achse gehörenden Modifikatoren benutzt werden. YLARGE oder YSMALL müssen beispielsweise benutzt werden, wenn die beiden Mittelpunkte die gleiche Y Koordinate haben.

Syntax:

< Punktparameterliste > : : =

¹₁ [XLARGE | XSMALL | YLARGE | YSMALL], INTOF,

< Kreis-Spezifikator >, < Kreis-Spezifikator >

8.3.5 Definition eines Punktes

durch seine Polarkoordinaten in der von der Anweisung genannten Koordinatenebene

XYPLAN

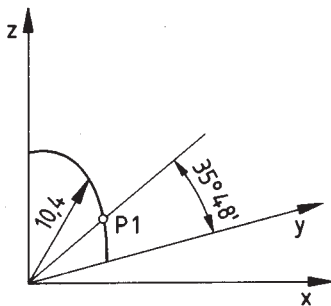
POINT/RTHETA, YZPLAN, Radius, Winkel

ZXPLAN

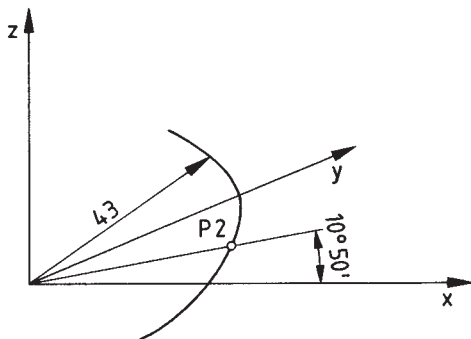
Semantik:

Radius bezeichnet den Betrag des Radiusvektors. Winkel bezeichnet den algebraischen Winkel von der ersten Achse, auf die durch den Modifikator Bezug genommen wird, zum Radiusvektor.

Beispiel:



P1 = POINT/RTHETA, YZPLAN, 10.4, 35.8



P2 = POINT/RTHETA, XYPLAN, 43, 10.833

Syntax:

< Punktparameterliste > : : = RTHETA,

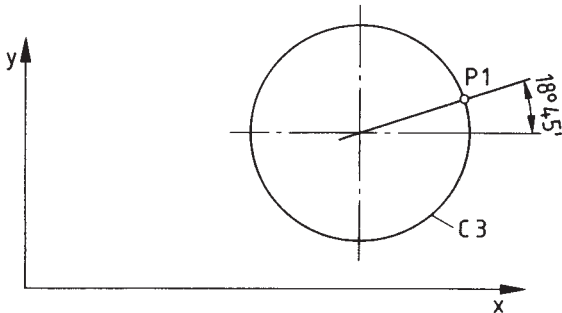
¹₁ [XYPLAN | YZPLAN | ZXPLAN], < Skalar >, < Skalar >

8.3.6 Definition eines Punktes

auf einem Kreis durch den Winkel zwischen Radiusvektor und X-Achse

POINT/Kreis, ATANGL, Winkel

Beispiel:



P1 = POINT/C3, ATANGL, 18.75

Syntax:

< Punktparameterliste > : : = < Kreis-Spezifikator >, ATANGL, < Skalar >

8.3.7 Definition eines Punktes

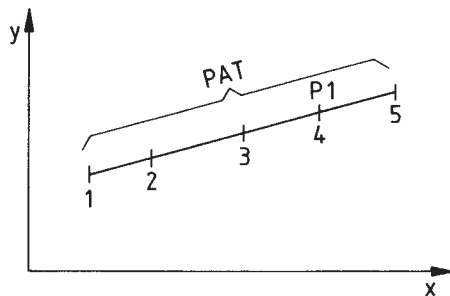
als n-ter Punkt eines Punktmusters

POINT/Punktmuster, Punktnummer

Semantik:

Punktnummer ist die Ordnungszahl des Punktes im Punktmuster.

Beispiel:



P1 = POINT/PAT, 4

Einschränkungen:

Die Punktnummer muß ganzzahlig sein. Sie darf nicht größer sein als die Gesamtzahl der Punkte im Punktmuster und nicht kleiner als eins sein.

Syntax:

< Punktparameterliste > : : = < Punktmuster-Spezifikator >, < Skalar >