

(19) 日本国特許庁(JP)

(12) 公開特許公報(A)

(11) 特許出願公開番号

特開2005-149194
(P2005-149194A)

(43) 公開日 平成17年6月9日(2005.6.9)

(51) Int. Cl. ⁷	F I	テーマコード (参考)
G06F 9/45	G06F 9/44 322E	5B081
G05B 19/408	G05B 19/408 Z	5H269

審査請求 未請求 請求項の数 4 O L (全 16 頁)

(21) 出願番号	特願2003-386660 (P2003-386660)	(71) 出願人	000005326 本田技研工業株式会社 東京都港区南青山二丁目1番1号
(22) 出願日	平成15年11月17日(2003.11.17)	(74) 代理人	100077665 弁理士 千葉 剛宏
		(74) 代理人	100116676 弁理士 宮寺 利幸
		(74) 代理人	100077805 弁理士 佐藤 辰彦
		(72) 発明者	吉田 亮 埼玉県狭山市新狭山1-10-1 ホンダ エンジニアリング株式会社内
		Fターム(参考)	5B081 AA10 CC41 5H269 AB01 BB16 EE27 EE29

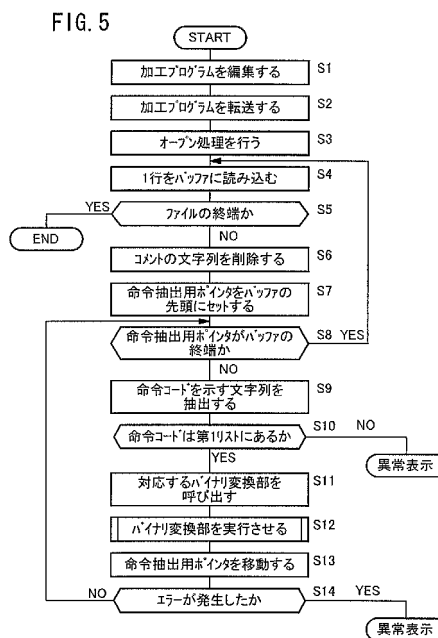
(54) 【発明の名称】 ソースプログラム変換システム

(57) 【要約】

【課題】 加工プログラム等のソースプログラムに対して新規の命令コードを適用する際に、該命令コードを簡便に適用させる。

【解決手段】 命令文字列が記録された第1リストと、命令文字列に従属する引数が記録された第2リストとを設ける。加工プログラムから1行をバッファに読み込む(ステップS4)。バッファから命令コードを抽出し、該命令コードが第1リストに登録されていることを確認する(ステップS9及びステップS10)。該命令コードに対応したバイナリ変換部を呼び出して実行する(ステップS11及びS12)。バイナリ変換部では、命令コードに続く文字列から引数を抽出し、該引数が第2リストにおける命令コードに対応する箇所に登録されていることを確認する。命令コードを引数とともに変換して実行コードを生成する。第1リスト及び第2リストには新規の命令コード及びその引数が追加登録可能である。

【選択図】 図5



【特許請求の範囲】**【請求項 1】**

テキスト形式のソースプログラムをコンピュータが読み取り実行可能な実行プログラムに変換するためのソースプログラム変換システムにおいて、

前記ソースプログラムに記述されている文字列群を入力する入力手段と、

あらかじめ適用された命令コード及び該命令コード毎に従属する引数が記録されたリストと、

前記入力手段で入力した文字列群から命令コードを抽出し、該命令コードが前記リストに登録されていることを確認する命令コード抽出手段と、

前記命令コードが引数をとるものである場合に、前記命令コードに続く文字列から引数を抽出し、該引数が前記リストにおける前記命令コードに対応する箇所に登録されていることを確認する引数抽出手段と、

前記命令コード抽出手段によって抽出された命令コードを、前記引数抽出手段によって抽出された引数とともに変換し、実行可能な実行コードを生成する変換手段と、

を有し、

前記リストには、命令コード及び引数が追加登録可能であることを特徴とするソースプログラム変換システム。

【請求項 2】

請求項 1 記載のソースプログラム変換システムにおいて、

前記引数抽出手段は、前記命令コードに続く文字列を順に調べて引数を順次抽出するとともに、抽出済みの引数を記憶し、該抽出済みの引数と同じ文字列があるとき、その時点における命令コードに対応した引数の抽出を終了することを特徴とするソースプログラム変換システム。

【請求項 3】

請求項 1 記載のソースプログラム変換システムにおいて、

前記リストは、命令コードが記録されている第 1 リストと、

各命令コード毎にその従属する引数が記録されている第 2 リストと、

に分けて構成されていることを特徴とするソースプログラム変換システム。

【請求項 4】

請求項 1 記載のソースプログラム変換システムにおいて、

前記変換手段は、前記命令コード毎に対応して実行コードを生成するバイナリ変換部を有し、

前記変換手段には、命令コードに対応したバイナリ変換部を追加登録可能であることを特徴とするソースプログラム変換システム。

【発明の詳細な説明】**【技術分野】****【0001】**

本発明は、テキスト形式のソースプログラムをコンピュータが読み取り実行可能な実行プログラムに変換するためのソースプログラム変換システムに関する。

【背景技術】**【0002】**

数値制御式の加工機械（NC 装置）を動作させる加工プログラムはテキスト形式で編集されており、該ソースプログラムには命令コードとして動作の種類の設定コード、目標位置の設定コード及び動作速度の設定コード等が記録される。NC 装置では、所定の変換部によってこれらの命令コードを認識、解釈してバイナリ形式の実行プログラムに変換した後、この実行プログラムを実行させてモータ等の制御を行う。

【0003】

加工プログラムの記述方法は NC 装置の変換部の機能に対応しており、1 行に 1 つの命令コードを記述する方法の他に、1 行に複数の命令コードを記述する方法や、行の概念が

10

20

30

40

50

なく連続して命令コードを記述する方法もある。

【0004】

1行に複数の命令コードを記述する方法では、NC装置の変換部に各命令コードに対応する処理部を設けて、該処理部において複数の命令コードが併記可能であることを認識するようにしている(例えば、特許文献1参照)。

【0005】

また、行の概念がなく連続して命令コードを記述する方法では、加工プログラムを命令コード毎に分割して、その後の分割した文字列が併記可能なものであるか否かを認識しながら変換を行っている。

【0006】

【特許文献1】特開平10-133826号公報

【発明の開示】

【発明が解決しようとする課題】

【0007】

上記のとおり、NC装置間では加工プログラムに互換性がないため、所定の加工プログラムを他のNC装置に移植しようとする場合、移植先のNC装置における変換部の仕様に合わせて加工プログラムを再編集する必要がある。

【0008】

また、1行に複数の命令コードを記述する方法や、行の概念がなく連続して命令コードを記述する方法では、従来適用されていなかった新規の命令コードが適用される場合には、該命令コードに対応した処理部を追加するだけではならず、従来の命令コードに対応する処理部も変更する必要がある。従って、新規の命令コードを追加するためには、処理部の内容を熟知している者でなければ追加の操作を行うことができず、該操作自体にも時間がかかる。

【0009】

本発明はこのような課題を考慮してなされたものであり、加工プログラム等のソースプログラムに対して新規の命令コードを適用する際に、該命令コードを簡便に適用させることができるソースプログラム変換システムを提供することを目的とする。

【課題を解決するための手段】

【0010】

本発明に係るソースプログラム変換システムは、テキスト形式のソースプログラムをコンピュータが読み取り実行可能な実行プログラムに変換するためのソースプログラム変換システムにおいて、前記ソースプログラムに記述されている文字列群を入力する入力手段と、あらかじめ適用された命令コード及び該命令コード毎に従属する引数が記録されたりリストと、前記入力手段で入力した文字列群から命令コードを抽出し、該命令コードが前記リストに登録されていることを確認する命令コード抽出手段と、前記命令コードが引数をとるものである場合に、前記命令コードに続く文字列から引数を抽出し、該引数が前記リストにおける前記命令コードに対応する箇所に登録されていることを確認する引数抽出手段と、前記命令コード抽出手段によって抽出された命令コードを、前記引数抽出手段によって抽出された引数とともに変換し、実行可能な実行コードを生成する変換手段と、を有し、前記リストには、命令コード及び引数が追加登録可能であることを特徴とする。

【0011】

このように、リストを参照しながら命令コード及び引数を抽出することにより、ソースプログラムに対して新規の命令コードを適用する際にはリストに命令コード及び引数を追加登録すればよく、該命令コードを簡便に適用させることができる。

【0012】

この場合、前記引数抽出手段は、前記命令コードに続く文字列を順に調べて引数を順次抽出するとともに、抽出済みの引数を記憶し、該抽出済みの引数と同じ文字列があるとき、その時点における命令コードに対応した引数の抽出を終了することにより、命令コードと引数とを正確に区別することができる。

10

20

30

40

50

【 0 0 1 3 】

また、前記リストは、命令コードが記録されている第1リストと、各命令コード毎にその従属する引数が記録されている第2リストとに分けて構成することにより、新規の登録が容易になる。

【 0 0 1 4 】

さらに、前記変換手段は、前記命令コード毎に対応して実行コードを生成するバイナリ変換部を有し、前記変換手段には、命令コードに対応したバイナリ変換部を追加登録可能とするとよい。

【 発明の効果 】

【 0 0 1 5 】

本発明に係るソースプログラム変換システムによれば、加工プログラム等のソースプログラムに対して新規の命令コードを適用する際に、該命令コードを簡便に適用させることができる。これにより、例えば、それまでの追加登録の履歴を把握していない者であっても、容易且つ短時間に新規の命令コードを追加し、適用させることができる。

【 発明を実施するための最良の形態 】

【 0 0 1 6 】

以下、本発明に係るソースプログラム変換システムについて実施の形態を挙げ、添付の図1～図7を参照しながら説明する。

【 0 0 1 7 】

図1に示すように、本実施の形態に係るソースプログラム変換システム10は、加工部12の数值制御を行うNC装置14と、該NC装置14を動作させるための加工プログラム(ソースプログラム)16、第1リスト18及び第2リスト20の編集を行う編集端末22とを有する。加工プログラム16、第1リスト18及び第2リスト20は、編集端末22のエディタ24によってテキスト形式で編集される。

【 0 0 1 8 】

NC装置14は、編集端末22から転送された加工プログラム16、第1リスト18及び第2リスト20をそれぞれ加工プログラム16a、第1リスト18a及び第2リスト20aとして記録するメモリ26と、第1リスト18a及び第2リスト20aを参照しながら加工プログラム16aを実行プログラム28に変換する変換部30と、実行プログラム28を読み取り実行して前記加工部12を動作させる実行処理部32とを有する。実行処理部32はアンプ34を介してモータ36、ソレノイド37等を作用させ、工具やテーブル等を駆動することによりワークに対する加工を行う。

【 0 0 1 9 】

変換部30は、加工プログラム16aから所定の長さの文字列を順に読み出す入力部40と、第1リスト18aを参照しながら、入力部40によって読み出された文字列に記述されている命令コードの種類を識別する命令コード抽出部42と、該命令コード抽出部42で識別した命令コードの種類に応じた引数を、第2リスト20aを参照しながら抽出し、実行可能なバイナリ形成に変換する変換処理部(引数抽出手段)44とを有する。該変換処理部44は、各命令コード毎に対応したバイナリ変換部46a、46b、46c、46d、46e及び46fを有する。

【 0 0 2 0 】

図2に示すように、加工プログラム16及び16aは、テキスト形式で編集されており、命令コード(「N」、「E」、「G00」等)と、命令コードに従属する引数(「X100.0」、「Y200.0」等)と、コメントが記述されている。

【 0 0 2 1 】

NC装置14で適用可能な命令コードは「G00」、「G01」、「G02」、「E」、「F」及び「N」の6つの命令コードである。引数は省略可能である。命令コード「G00」は変換処理部44におけるバイナリ変換部46aによって変換処理される。同様に、命令コード「G01」、「G02」、「E」、「F」及び「N」は、それぞれバイナリ変換部46b、46c、46d、46e及び46fによって変換処理される。

10

20

30

40

50

【0022】

なお、後述するように、変換処理部44には、新規の命令コードに対応する変換部を追加することが可能であり、例えば、新規の命令コード「G103」に対応するバイナリ変換部46gを追加設定することができる。

【0023】

實際上、NC装置14にはより多くの命令コードが存在するが、理解を容易にするために代表的な6つの命令コードと、新規の命令コード「G103」に限って説明する。

【0024】

変換部30で変換可能な加工プログラム16aの書式は、1行に複数の命令コードを記述することが可能であって、命令コード毎に他の命令コードとの併記が許可されるか又は排他的であるかが規定されている。併記が許可される命令コードは、1行に複数記述可能である。排他的な命令コード同士を1行に記述することは禁止されており、併記が許可される1以上の命令コードと1つの排他的な命令コードを記述することは可能である。

10

【0025】

「G00」の命令コードは、非補間の位置決めコードであって、排他的な命令コードである。また、引数は3次元座標としての「X」、「Y」及び「Z」と、姿勢を示す「A」、「B」及び「C」である。

【0026】

「G01」の命令コードは、直線補間コードであり、排他的な命令コードである。また、引数は「X」、「Y」、「Z」、「A」、「B」及び「C」である。

20

【0027】

「G02」の命令コードは、時計回りの円弧補間コードであり、排他的な命令コードである。また、引数は「X」、「Y」、「Z」及び円弧の半径としての「R」である。

【0028】

「E」の命令コードは、各軸速度の指定コードであり、他命令コードとの併記は許可されている。また、引数はない。

【0029】

「F」の命令コードは、補間動作速度指定コードであり、他命令コードとの併記は許可されている。また、引数はない。

【0030】

「N」の命令コードは、ラベル番号指定コードであり、他命令コードとの併記は許可されている。また、引数はない。この「N」の命令コードは、所定の分岐命令（いわゆるGO TO文等）に対応する分岐先として利用可能である。

30

【0031】

例えば、「N10」という記述は、ラベル番号「10」を指定する命令であり、「N」の直後に対応する数値を記述する。「F」及び「N」も同様である。また、行番号を10番として採番し、動作として $X = 100.0$ [mm]、 $Y = 200.0$ [mm]の位置へ各軸速度を100 [%]で移動させる場合には、

「N10 G00 X100.0 Y200.0 E100」

という文字列を1行に記述すればよい。

40

【0032】

なお、これらの命令のうちいくつかは、JIS B 6315-2及びISO/DIS 6983-2にも規定されている。

【0033】

図3Aに示すように、第1リスト18及び18aには各命令コードが記録されている。

【0034】

図4Aに示すように、第2リスト20及び20aには各命令コード毎にその従属する引数が記録されている。このうち、引数を有する命令コードの行には、最後の引数の欄の右側欄にそれ以上の引数がないことを示す「-」が記録されている。また、命令コード「E」、「F」及び「N」には、従属する引数がないことから、最初の欄に「-」が記録され

50

ており、引数がないことを示している。

【0035】

第1リスト18に命令コード、第2リスト20に引数をそれぞれ分けて記録することにより適用済みの命令コード及び引数が明瞭になり、後述するように新規の命令コードを追加登録する際にも編集が容易である。

【0036】

次に、このように構成されるソースプログラム変換システム10によって、加工プログラム16を実行プログラム28に変換する手順について、図5～図7を参照しながら説明する。図5に示す処理のうち、ステップS1～S4は、主に前記入力部40によって実行される処理であり、ステップS5～S14は主に前記命令コード抽出部42によって実行される処理である。また、図6に示す処理は、主に変換処理部44のバイナリ変換部46aによって実行される処理である。

10

【0037】

まず、ステップS1において、編集端末22のエディタ24によって加工プログラム16を編集する。この編集作業は、いわゆるプログラミング作業であり、NC装置14が所定の動作を行うことができるように加工プログラム16を作成・編集する。また、第1リスト18及び第2リスト20は、あらかじめ図3A及び図4Aに示すように作成されているものとする。

【0038】

加工プログラム16は、テキスト形式で編集され、第1リスト18及び第2リスト20はテキスト形式又はバイナリ形式で編集されている。

20

【0039】

次に、ステップS2において、加工プログラム16、第1リスト18及び第2リスト20をNC装置14のメモリ26に転送し、加工プログラム16a、第1リスト18a及び第2リスト20aとして記録する。

【0040】

次いで、ステップS3において、所定の変換開始指示操作により、入力部40がメモリ26内の加工プログラム16aに対するアクセスを可能にするためのオープン処理を行う。加工プログラム16aは、図2に示すように編集されているものとする。

【0041】

さらに、ステップS4において、入力部40は加工プログラム16aから1行を所定のバッファに読み込む。このステップS4の処理は、実行される度に先頭から順に1行ずつを読み込むことができるように、加工プログラム16aにおける行を示す所定の行ポインタにより制御される。

30

【0042】

次に、ステップS5において、読み込んだ行がファイルの終端を示す「EOF」コードであるか否かを確認する。バッファ中の文字列が「EOF」コードであるときには処理を終了し、それ以外ときにはステップS6へ進む。

【0043】

ステップS6においては、バッファ中の文字列からコメントの文字列を削除する。つまり、括弧“(”及び閉じ括弧“) ”で囲まれる範囲の文字列をその括弧を含んで削除し、さらに、その行におけるセミコロン“ ; ”以降の文字列を削除する。

40

【0044】

これにより、例えば、加工プログラム16の第1行目の文字列(図2参照)の場合、「(10番のタップ位置へ移動)」という文字列と、「 ; 速度設定」という文字列が削除され、

「G00 X100.0 Y200.0 E100」

という文字列がバッファ中に残る。

【0045】

次に、ステップS7において、読み取り位置を示す命令抽出用ポインタをバッファの先

50

頭にセットする。つまり、上記の文字列の場合には、先頭の文字である「G」の部分に所定の命令抽出用ポインタがセットされる。この命令抽出用ポインタはステップS8以降の処理においてバッファにおける位置が移動する。

【0046】

次いで、ステップS8において、その時点の命令抽出用ポインタの位置を調べ、該命令抽出用ポインタがバッファの終端にある場合には前記ステップS4へ戻り、それ以外の場合には次のステップS9へ移る。ステップS4へ戻る場合には加工プログラム16aから次の行の文字列をバッファに順次読み込み、該文字列に対して処理を続行することになる。

【0047】

ステップS9において、その時点における命令抽出用ポインタの位置から次のスペース位置までに含まれる文字列を命令コードとして抽出する。上記の文字列の場合には、当初「G00」という命令コードの文字列が抽出される。

【0048】

ステップS10において、抽出された命令コードを示す文字列が第1リスト18aにあるか否かを確認する。該文字列が第1リスト18aに存在する場合には、次のステップS11へ移り、存在しない場合には所定の異常表示を行い、処理を終了する。

【0049】

ステップS11において、抽出された命令コードに対応する所定のバイナリ変換部を変換処理部44から呼び出す。具体的には、抽出された命令コードが「G00」である場合には、対応するバイナリ変換部46aを呼び出す。

【0050】

次に、ステップS12において、呼び出されたバイナリ変換部を実行させる。このとき、バイナリ変換部に対して、抽出された命令コードに続く文字列を引き渡す。つまり、上記の文字列の場合には、「G00」に続く文字列である

「X100.0 Y200.0 E100」

という文字列をバイナリ変換部46aに引き渡す。このバイナリ変換部46aにおける処理については後述する。

【0051】

次いで、ステップS13において、バイナリ変換部によって認識された引数を確認し、バッファ上において該引数の文字列に続く次の文字の位置に命令抽出用ポインタを移動する。つまり、上記の文字列における最初の命令コード「G00」に対応したバイナリ変換部46aを呼び出した場合には、引数が「X100.0 Y200.0」であると認識され、命令抽出用ポインタはそれに続く文字である「E」の位置にセットされる。

【0052】

さらにステップS14において、バイナリ変換部における処理中にエラーが発生したか否かを確認し、エラーがある場合には所定の異常表示を行い、処理を終了する。それ以外の場合、つまり正常である場合には前記ステップS8へ戻りバッファ中の次の文字列に対して処理を続行する。

【0053】

このようにして、加工プログラム16aに記述された文字列を、ステップS4～S8において1行ずつバッファに順次読み込むとともに、ステップS9～S14においてバッファ内の文字列を分解して、バイナリ変換部46a～46fによって順次バイナリ形式に変換することができる。

【0054】

次に、前記ステップS12におけるバイナリ変換部の処理について図6を参照しながら説明する。なお、本説明では、代表例として、バイナリ変換部のうち命令コード「G00」に対応するバイナリ変換部46aが、命令コード抽出部42から文字列「X100.0 Y200.0 E100」の引き渡しを受けた場合について説明する。

【0055】

10

20

30

40

50

まず、ステップ S 1 0 1 において、引き渡しを受けた文字列における先頭の文字「X」の位置に引数抽出用ポインタをセットしておく。

【0056】

次に、ステップ S 1 0 2 において、対応する命令コードである「G 0 0」は引数をとるか否かを確認する。具体的には、第 2 リスト 2 0 a を参照して、命令コード「G 0 0」の行を参照し、この行に引数を示す文字がなく、文字「-」のみが記録されているときにはステップ S 1 0 8 へ移り、引数を示す文字列が記録されている場合には、ステップ S 1 0 3 へ移る。

【0057】

また、引数がない命令コードは第 2 リスト 2 0 a に登録しないこととし、該第 2 リスト 2 0 a を参照したときに対応する文字列がない場合にステップ S 1 0 8 へ移るようにしてもよい。

10

【0058】

次に、ステップ S 1 0 3 において、第 2 リスト 2 0 a における「G 0 0」の行を参照し、その時点における引数抽出用ポインタが示す文字（例えば、「X」）が存在するか否かを調べる。対応する文字がないときにはステップ S 1 0 8 へ移り、該文字が存在するときにはステップ S 1 0 4 へ移る。

【0059】

なお、上記のとおり、命令コード「G 0 0」は、引数として「X」、「Y」、「Z」、「A」、「B」及び「C」の 6 つをとり得るが、引数の数は 6 以下の任意の数でよく、例えば、引数が全くなってもよい。

20

【0060】

ステップ S 1 0 4 において、その時点における引数抽出用ポインタが示す文字がすでに処理済みであるか否かを確認する。つまり、後述するように、ある引数が所定の命令コードと同じ文字列に適用されている場合であっても、引き渡された文字列のうち先頭側に存在する文字列を引数として識別し、後側に存在する文字列を「G 0 0」に続く次の命令コードであると識別することができる。

【0061】

引数抽出用ポインタが示す文字がすでに処理済みである場合には、ステップ S 1 0 8 へ移り、それ以外の場合にはステップ S 1 0 5 へ移る。

30

【0062】

ステップ S 1 0 5 においては、その時点における引数抽出用ポインタの位置から次のスペース位置までに含まれる数値を示す文字列を抽出し、引数抽出用ポインタの示す文字と組み合わせて、その引数が示す数値データとして認識する。

【0063】

具体的には、引数抽出用ポインタが「X」を示す場合には、引数 X を、 $X = 100.0$ として認識し、メモリ 26 上の所定の変数エリア（又は所定のバッファ）に記録する。この場合、「100.0」という文字列としてではなく数値データとして記録してもよい。

【0064】

次に、ステップ S 1 0 6 において、前記ステップ S 1 0 5 において認識した数値データがあらかじめ規定された適正範囲の値であるか否かを確認する。適正である場合には、ステップ S 1 0 7 へ移り、不適正である場合には所定の異常表示を行い、処理を終了する。数値データが不適正である場合には、所定のワーニング表示を行った後にステップ S 1 0 7 へ移って処理を続行してもよい。

40

【0065】

ステップ S 1 0 7 においては、その時点まで認識した引数と該引数に付属する数値を示す文字列の次の文字に引数抽出用ポインタをセットする。つまり、引数を示す文字列「X」に対して、 $X = 100.0$ と認識した後は、引数抽出用ポインタを次の文字列「Y」の位置にセットする。この後、前記ステップ S 1 0 2 へ移る。

【0066】

50

このようにして、ステップ S 1 0 2 ~ S 1 0 7 を繰り返し行うことにより、文字列「X 1 0 0 . 0 Y 2 0 0 . 0 E 1 0 0」に対して、命令コード「G 0 0」に対応する引数を X = 1 0 0 . 0、Y = 2 0 0 . 0 として認識することができる。この後、引数抽出用ポインタが文字列「E」の位置にセットされたときには、前記ステップ S 1 0 3 により、該文字列「E」が引数でないことが認識されてステップ S 1 0 8 へ移る。

【0067】

ステップ S 1 0 8 (変換手段)においては、その時点までに認識された引数の数(0 ~ 6)と各引数の数値データを反映させて「G 0 0」をバイナリ形式に変換する。

【0068】

ここでいう変換とは、単に「G 0 0」という文字列をバイナリ形式に表現形式だけを変えるのではなく、NC装置 1 4 において実行処理部 3 2 が読み取り実行可能な形式(つまり実行コード)に変換することをいう。変換したコードは、実行プログラム 2 8 を記録する領域に対して、所定のアドレス順に書き込む。

【0069】

この後、引き渡しを受けた文字列「X 1 0 0 . 0 Y 2 0 0 . 0 E 1 0 0」のうち、この時点における引数抽出用ポインタの位置である「E」の位置に基づき、「X 1 0 0 . 0 Y 2 0 0 . 0」までが認識済みであり、これに続く文字列「E 1 0 0」が未処理である旨を命令コード抽出部 4 2 に通知する。命令コード抽出部 4 2 では、この通知に基づき前記ステップ S 1 3 において命令抽出用ポインタを移動させる。

【0070】

ステップ S 1 0 8 の処理の後、バイナリ変換部 4 6 a における今回の処理、つまり図 6 に示す処理を終了する。加工プログラム 1 6 a に命令コードとしての文字列「G 0 0」が複数記述されている場合には、バイナリ変換部 4 6 a が再度呼び出されることはもちろんである。

【0071】

ここまで命令コード「G 0 0」に対応したバイナリ変換部 4 6 a の処理について説明したが、他の命令コード「G 0 1」、「G 0 2」、「E」、「F」及び「N」についても、バイナリ変換部 4 6 b ~ 4 6 f (図 1 参照)によって同様に処理される。バイナリ変換部 4 6 a ~ 4 6 f は、ステップ S 1 0 1 ~ S 1 0 7 に相当する部分は共通にして、ステップ S 1 0 8 に相当する部分だけを個別に設けるようにしてもよい。

【0072】

また、上記の例では、1行に2つの命令コード「G 0 0」及び「E 1 0 0」が含まれている例について説明したが、1行に3つ以上の命令コードが含まれている場合も同様に処理可能である。

【0073】

ところで、加工プログラム 1 6 a に記述される命令コードは、動作命令と非動作命令に区別される。動作命令とは、モータ 3 6 やソレノイド 3 7 等のアクチュエータを実際に動作させる命令であり、非動作命令は、速度設定や送り設定等のパラメータ設定用の命令であり、実際の動作を伴うことがない。

【0074】

動作命令と非動作命令を組み合わせ、例えば、各軸の速度を 1 0 0 [%] に設定し、X = 1 0 0 . 0、Y = 2 0 0 . 0 の座標に位置決めするように加工プログラム 1 6 a に記述する際、

「G 0 0 X 1 0 0 . 0 Y 2 0 0 . 0 E 1 0 0」

という記述と、

「E 1 0 0 G 0 0 X 1 0 0 . 0 Y 2 0 0 . 0」

という記述の両方が許容されている場合がある。

【0075】

このような場合には、前記ステップ S 1 0 8 において変換したコードを、実行プログラム 2 8 を記録する領域に対して直接に書き込むのではなく、一度、動作命令用の記録領域

10

20

30

40

50

と、非動作命令用の記録領域に分けて書き込み、加工プログラム 16 a の全て（又は所定の部分）まで変換が終わった後にコードを並び替えて実行プログラム 28 として記録してもよい。

【0076】

以上の手順によって加工プログラム 16 a を変換することによって作成された実行プログラム 28 は、実行処理部 32 のコンピュータによって読み取り実行され、アンプ 34 を介してモータ 36 及びソレノイド 37 等を制御する。

【0077】

なお、上記の手順は、加工プログラム 16 a に基づいて実行プログラム 28 を一度作成した後に、該実行プログラム 28 を実行させるといふ、いわゆるコンパイラ型の変換システムとしてもよいし、また、加工プログラム 16 a を実行プログラム 28 に変換しながら、変換した部分を直ちに実行させるといふ、いわゆるインタプリタ型の変換システムとしてもよい。

【0078】

次に、加工プログラム 16 に新規の命令コードを適用させる場合に、該加工プログラム 16 を実行プログラム 28 に変換する手順について説明する。

【0079】

ここで説明する新規の命令コードは、従来適用されていなかったものであり、「G103」と表現されるものとする。また、この命令コード「G103」は、タップサイクルを設定するコードであって、排他的な命令コードであり、引数には工具の回転数を示す「N」と、ピッチを示す「F」と、タイマ設定用の「P」とをとり得るものとする。

【0080】

この命令コード「G103」を用いて、加工プログラム 16 に

```
「G103 N8 F0.8 P1.0 F600」
```

という文字列を記述した場合について以下に説明する。

【0081】

この場合、2つの文字列「F」のうち先頭側の「F0.8」は「G103」の引数で、ピッチが0.8 [mm]であることを示す。一方、後側の「F600」は、「G103」とは別の命令コードであって、補間動作速度が600 [mm/s]であることを示す。また、「N8」は、「G103」の引数であって、工具の回転数が8回であることを示し、前記の命令コード「N」とは異なる。「P1.0」は、「G103」の引数であって、1.0 [s]のタイマ機能を示す。

【0082】

加工プログラム 16 に新規の命令コード「G103」を記述し、該加工プログラム 16 を実行プログラム 28 に変換する際、準備として図 7 に示す作業を行う。

【0083】

先ず、ステップ S201 において、変換処理部 44（図 1 参照）に新規の命令コード「G103」をバイナリコードに変換するためのバイナリ変換部 46g（図 1 参照）を追加登録する。この場合、他のバイナリ変換部 46a ~ 46f 及び命令コード抽出部 42 は変更する必要がない。

【0084】

バイナリ変換部 46g は、図 6 に基づいて説明したバイナリ変換部 46a と基本的に同じ構成であり、前記ステップ S108 に相当する部分が異なる。つまり、バイナリ変換部 46a は命令コード「G00」をバイナリ形式に変換することに対して、バイナリ変換部 46g におけるステップ S108 に相当する部分では命令コード「G103」をバイナリ形式に変換する。以下の説明では、バイナリ変換部 46g に対して、図 6 と同じステップ番号を用いて説明する。

【0085】

次に、ステップ S202 において、第 1 リスト 18 に新規の命令コード「G103」を追加登録し、更新する（図 3B 参照）。

10

20

30

40

50

【0086】

さらに、ステップS203において、第2リスト20に命令コード「G103」と、そのとり得る引数の「N」、「F」及び「P」を追加登録し、更新する(図4B参照)。

【0087】

第1リスト18及び第2リスト20に対する追加登録は、例えば、エディタ24を用いればよい。

【0088】

このようにステップS203までの準備が終了した後、図5及び図6に示す処理を実行する。

【0089】

この場合、前記ステップS9及びS10において、更新された第1リスト18(図3B参照)を参照することにより「G103」という文字列が命令コードであることが認識可能であり、しかも最初の1文字が同一である他の命令コードの「G00」、「G01」及び「G02」との区別が可能である。

【0090】

また、この後、ステップS11により、命令コード「G103」に対応するバイナリ変換部46gが適切に呼び出され、「G103」に続く文字列である「N8 F0.8 P1.0 F600」がバイナリ変換部46gに引き渡される。

【0091】

バイナリ変換部46gにおける処理では、ステップS102及びS103において、更新された第2リスト20(図4B参照)を参照することにより、文字列「N」が「G103」の引数であることが認識可能であり、ラベル番号指定コードである「N」とは区別できる。

【0092】

さらに、ステップS105及びS106において、文字列「N」に続く文字列である「8」を認識することにより、N=8、つまり工具の回転数が8回であることが認識可能となり、命令コード「G103」の最初の引数である[N]についての処理が終了する。

【0093】

次に、ステップS107において引数抽出用ポインタを2つの文字列「F」のうち先頭側の位置にセットする。その後、文字列「N」の場合と同様に、F=0.8、つまりピッチが0.8[mm]であることを認識する。

【0094】

さらに、引数抽出用ポインタを文字列「P」の位置にセットして同様の処理を行い、P=1.0、つまり1.0[s]のタイマ機能を認識する。

【0095】

このようにして、新規の命令コードである「G103」に従属する引数である「N8」、「F0.8」及び「P1.0」を正しく認識することができる。

【0096】

さらにまた、この後ステップS107において引数抽出用ポインタを2つの文字列「F」のうち後側の位置にセットし、ステップS103を介してステップS104に移る。このステップS104においては、前記のとおり、その時点における引数抽出用ポインタが示す文字列「F」が処理済みであるか否かを確認するが、この文字列「F」は2つ存在するうちの後側であることからすでに1度処理を行っているとは判断されて、ステップS108へ移ることとなる。

【0097】

従って、後側の「F」及びこれに続く「600」という文字列は、命令コードである「G103」の引数ではなく、異なる次の命令コードであることを正しく判断することができる。

【0098】

この後、ステップS108では、認識された各引数及びその数値データを反映させて新

10

20

30

40

50

規の命令コードである「G103」をバイナリ形式に変換する。

【0099】

上述したように、本実施の形態に係るソースプログラム変換システム10によれば、「G103」等の新規の命令コードを用いる場合、これに対応するバイナリ変換部46gを変換処理部44に追加登録するとともに、第1リスト18及び第2リスト20に「G103」の命令コードとその引数「N」、「F」及び「P」を追加登録すればよい。この場合、他のバイナリ変換部46a~46f及び命令コード抽出部42には影響がなく、修正を行う必要がない。

【0100】

次に、本実施の形態に係るソースプログラム変換システム10と対比するために、従来技術について説明する。 10

【0101】

従来技術においては、加工プログラム16aからバッファに読み込んだ文字列のうち、「E」、「F」、「G」又は「N」のいずれかの文字から次に現れる「E」、「F」、「G」又は「N」までの間を1つの命令として認識している。従って、前記の「G103」のような新規の命令コードが「F」や「N」を引数としてとるものと規定されている場合には、そのままでは命令コードと引数との区別ができず、専用の判断処理を設ける必要がある。

【0102】

つまり、命令コード抽出部42に相当する処理部又は変換処理部44に相当する処理部において次のような意味の例外処理を追加、変更する必要がある。 20

【0103】

すなわち、この場合の例外処理は、“ただし、切り分け予定部分の先頭の文字列が「G103」であったら、1回目の「F」及び「N」は無視する”という意味のものである。

【0104】

また、このような追加、変更の操作は、新規の命令コードが適用される度に行う必要がある。命令コード抽出部42に相当する処理部が次第に複雑となり、追加、変更の操作の際に人為的なミスを誘発することがある。また、追加、変更の操作を担当した以外の者には処理構造の理解が困難になる。さらに、いわゆるバージョン管理が繁雑となる。

【0105】

一方、本実施の形態に係るソースプログラム変換システム10によれば、命令コード抽出部42や変換処理部44は、第1リスト18及び第2リスト20を参照しながら命令コードの抽出を行うため、新規の命令コードを適用する際に、命令コード抽出部42や従来の命令コードに対応するバイナリ変換部46a~46fに対して変更を加える必要がない。この場合、第1リスト18及び第2リスト20には、新規の命令コードに対応して、その命令コード自体と、引数とを追加登録するが、この手順は極めて容易であって、しかも追加登録の数が増えても第1リスト18及び第2リスト20の基本構造は変化がない。従って、それまでの追加登録の履歴を把握していない者であっても、容易且つ短時間に新規の命令コードを追加適用させることができる。 30

【0106】

このように、本実施の形態に係るソースプログラム変換システム10によれば、加工プログラム16及び16aに対して新規の命令コードを適用する際に、該命令コードを簡単に適用させることができる。 40

【0107】

また、上記の説明では、命令コードとして「E」、「G00」、「G103」等の動作や条件を設定するための命令コードについて例示したが、これ以外に、分岐処理、演算処理及び代入処理等に関する命令コードについても適用可能であることはもちろんである。例えば、第2リスト20における登録データを引数なしとし、前記ステップS108の処理を分岐命令、演算命令及び代入命令に置き換えることにより、これらの処理に対して適用可能となる。つまり、図6に示す処理は各命令コード毎に用意されるものであり、全 50

体の仕組みや他の命令処理に影響されずに作成可能である。

【0108】

さらに、本実施の形態に係るソースプログラム変換システム10は、NC装置14を制御するための加工プログラム16に限らず、他の形式のプログラムにも適用可能である。

【0109】

上記の説明では、新規の命令コードとして1つの命令コード「G103」に対して適用する例について説明したが、新規の命令コードが複数である場合にも同様の手順により対応可能であることはもちろんである。

【0110】

さらにまた、上記の説明では、第1リスト18と第2リスト20とを参照しながら命令コード及び引数を抽出する例について説明したが、第2リスト20のうち図4Aにおける最も左側の列に相当する部分を第1リスト18として兼用することとし、該第1リスト18自体を省略してもよい。

【0111】

実行プログラム28は、1つの加工プログラム16から変換されるものとして説明したが、加工プログラム16は複数のファイルから構成されていてもよい。

【0112】

本発明に係るソースプログラム変換システムは、上述の実施の形態に限らず、本発明の要旨を逸脱することなく、種々の構成をとり得ることはもちろんである。

【図面の簡単な説明】

【0113】

【図1】本実施の形態に係るソースプログラム変換システムの構成ブロック図である。

【図2】加工プログラムの内容を示す図である。

【図3】図3Aは、第1リストの内容を示す図であり、図3Bは、新規の命令コードを追加登録して更新された第1リストの内容を示す図である。

【図4】図4Aは、第2リストの内容を示す図であり、図4Bは、新規の命令コードを追加登録して更新された第2リストの内容を示す図である。

【図5】入力部及び命令コード抽出部が行う処理の手順を示すフローチャートである。

【図6】バイナリ変換部が行う処理の手順を示すフローチャートである。

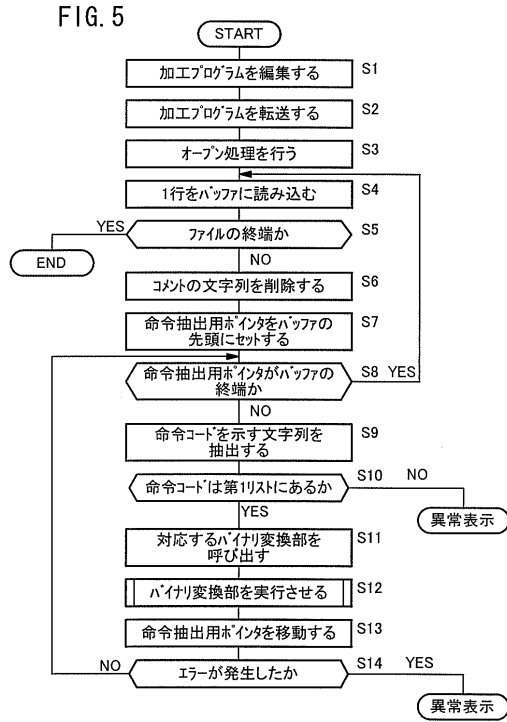
【図7】新規の命令コードを適用させる際における準備の手順を示すフローチャートである。

【符号の説明】

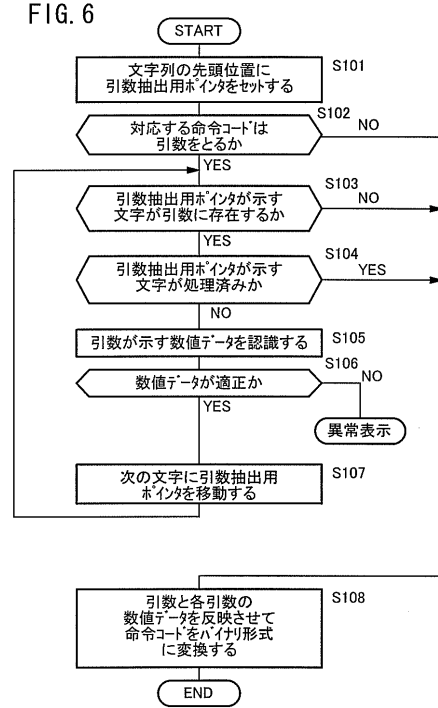
【0114】

10 ... ソースプログラム変換システム	14 ... NC装置
16、16a ... 加工プログラム	18、18a ... 第1リスト
20、20a ... 第2リスト	22 ... 編集端末
24 ... エディタ	26 ... メモリ
28 ... 実行プログラム	30 ... 変換部
32 ... 実行処理部	40 ... 入力部
42 ... 命令コード抽出部	44 ... 変換処理部
46a ~ 46g ... バイナリ変換部	

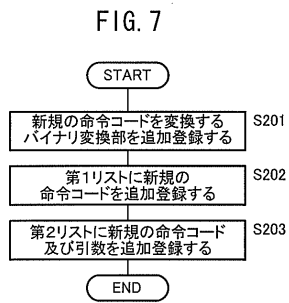
【 図 5 】



【 図 6 】



【 図 7 】



フロントページの続き

【要約の続き】