

## 構造化プログラムの日本語ドキュメント作成システム†

金田 悠紀夫†† 前川 禎 男††  
吉田 満††† 川本 博††††

本論文は構造化されたプログラムに対する構造化されたドキュメントの概念を提案するとともに、漢字かな混り文による構造化ドキュメントを自動作成する支援システムに関して論じている。ここではブロック構造を持つ言語で記述された構造化プログラムを想定している。プログラムを根とし、プロシージャ宣言、データ宣言、本体、複文などブロックを構成するものを非終端節、単データ宣言、単文を終端節とするとプログラム全体は多レベルの木構造となる。本システムではこのプログラム木と同一構造を持つドキュメント木の考えを導入する。プログラム木の各節に対するドキュメントはドキュメント木の対応する節で与えられる。本システムではドキュメント木を depth first の順に探索し各節を、ドキュメント文が挿入されたブロックとして順次一次元に配置してゆき、それらを弧で結ぶことにより構成する一次元の流れ図をドキュメントとしている。ドキュメントはドキュメント記述用言語で記述されカード形式でコンピュータに入力される。ローマ字文の形で入力された日本語ドキュメント文は漢字かな混り文に変換され、ユーザからの出力指示言語を用いた出力指示により、ブロック内に漢字かな混り文が挿入された一次元の流れ図が出力される。

### 1. はじめに

汎用性のあるすぐれたソフトウェアを利用する場合においても提供者と緊密な情報交換がない場合には、ドキュメントの良否がソフトウェアの流通性に対して決定的な役割を演じる。本論文は構造化されたプログラムに対する構造化されたドキュメントの概念を提案し、日本語文構造化ドキュメントを自動生成するドキュメント作成支援システムに関して論じている。

### 2. 構造化プログラムと構造化ドキュメント

本論文では ALGOL, PL/I, PASCAL などのブロック構造を持つ言語で記述された構造化プログラムを主たる対象としている。

#### 2.1 プログラム木とドキュメント木の概念

プログラムは主プロシージャと複数個の外部プロシージャから構成されており、各プロシージャは宣言と本体から構成される。宣言はデータ宣言と内部プロシージャ宣言から構成され、本体は単文、複文、それらの連結から構成される。データ宣言は単データに対する宣言の集りと、本構造を持つ集合データの宣言（終

端節は単データ宣言）の集りから構成されている。複文は単文、宣言、入れ子された他の複文から構成されるブロック構造をしていて終端節は単文と単データ宣言となっている。したがってプログラムを根とし、プロシージャ宣言、データ宣言、本体、複文などブロックを構成するものを非終端節、単データ宣言、単文を終端節とするプログラム全体は多レベルの木構造をしていることが分かる。本システムではこのプログラムの静的な構造を示す木をプログラム木と呼んでおり、この木と同一構造を持つドキュメント木の考えを導入する。プログラム木の各節に対するドキュメントはドキュメント木の対応する節で与えられる。プログラム木を縦形に探索しながらプログラムテキストの先頭から順に対応をとっていくことにより容易にプログラム木およびドキュメント木とプログラムテキストの対応をとることができる。

#### 2.2 プログラムの持つ制御構造の記述

Dijkstra<sup>2)</sup> によると構造化されたプログラムの制御の流れは(1)文の連結、(2) if then 文、(3) if then else 文、(4) case 文、(5) while 文、(6) repeat 文、の6つの基本的な制御構造にしたがってブロックを結合することによって表現できるとしている。各ブロックは入れ子構造にすることを許しており、また制御の流れはブロック間を結ぶ有向弧（制御リンクと呼ぶ）で示される。本論文ではこれにループを示す for 文、プログラムの始点と終点を示す START 文、END 文、プロシージャ呼出しを示す文を付加することによ

† Japanese Documents Generator of Structured Programs by YUKIO KANEDA, SADA O MAEKAWA (Kobe University of Systems Engineering), MITSURU YOSHIDA (Sharp Corp.), and HIROSHI KAWAMOTO (Fujitsu, Ltd.).

†† 神戸大学工学部システム工学科

††† シャープ(株)

†††† 富士通(株)

り制御の流れが記述されると考える。

### 2.3 一次元流れ図を用いたドキュメント

本システムでは、ブロックを一次元に配置してゆきそれらを弧で結ぶことによって構成する一次元の流れ図をドキュメントとして採用している。

一次元流れ図は以下の手順にしたがって生成される。

1. ドキュメント木の各節をプログラム木の場合と同様な手法で制御リンクで結ぶ。
2. ドキュメント木を縦形探索し各節ごとに次の処置をとる。
  - a) その節に対するドキュメントを出力すると決めると記憶しておきその子孫の節の探索は行わない。
  - b) その節が終端節の場合はそのドキュメントは必ず出力することとして記憶しておく。
  - c) それ以外の場合は子孫の節に対する探索を続ける。
3. 全木の探索が終了すると出力するとして記憶されているドキュメントそれぞれを一つのブロックとして探索順にしたがって一次元に配置する。
4. 制御リンクにしたがってブロック間を結合する。
5. 結果の出力図が一次元の流れ図となる。

以上の手続で出力される流れ図はプログラムの各部分に関して任意のレベルのドキュメントを混合して一つの一次元流れ図として作成することができる。したがって部分的に詳細なドキュメントの含まれている流れ図を作ることも可能である。また木の任意の節を根として1~5の操作を行うことによりモジュールごとの一次元流れ図を生成することも可能である。

以上の手順で生成された一次元の流れ図はいわゆる二次元の流れ図に比して一見明解さに劣るとも考えられるが以下の利点がある。

1. プログラムテキストとの対応が極めてとり易い。
2. 各ブロックの大きさを比較的自由に選べるので各ブロックに十分な量の情報が記述できる。
3. 文献などに記述されているアルゴリズムも各ステップを文章で記述し、全体は一次元に並べられた文章群からなっている場合が多い。これらのアルゴリズムの記述と対応が取り易い。
4. プログラム作成後にドキュメントとして作成する場合二次元の流れ図に比して容易に作れる。
5. コンピュータを用いて流れ図を自動的に生成したり修正したりするのが比較的容易である。

### 2.4 ブロック構造を持たないプログラミング言語で記述された構造化プログラムへの適用

アセンブリ言語や FORTRAN などのいわゆるブロック構造を持たない言語で記述されたプログラムの場合には2.1節で述べているプログラムの構文構造とドキュメント構造の一対一対応はとれないが、2.3節で示した制御構造をプログラムの構造に反映させれば、プログラムとドキュメント構造との対応をとることが可能となる。

### 3. 構造化ドキュメント作成システム

構造化ドキュメントはドキュメント記述用言語で記述されカード形式で入力される。ドキュメントは一次元流れ図の形に構成されて記憶されており、ローマ字の形で記述されている日本語コメント文は漢字かな混り文の形に変換される。ユーザは出力形式指示コマンドを用いて出力形式を指定することにより、任意のレベルの一次元流れ図を出力させることができる。

#### 3.1 ドキュメント記述用言語

ドキュメント木を記述する言語をドキュメント記述用言語と呼ぶ。各節を D-BLOCK と呼ぶと、木は節となる全 D-BLOCK の本体と親子関係より記述される。D-BLOCK は START から END までの一連の文で定義され内部に子となる D-BLOCK 宣言の集合を含む。子の D-BLOCK で非終端節は親 D-BLOCK の外で宣言され親子関係は D-BLOCK 名の連結で示される。

例を図1に示す。プログラムはブロック構造をしており、 $B_0, B_1, B_2, B_3, B_4$  の5つのブロックからなる入れ子構造をしている。対応する D-BLOCK 名を  $B_0, B_1, B_2, B_3, B_4$  とすると図1(b)の形で表現される。

#### 3.2 ドキュメント記述言語の構成

ドキュメントは〈D-BLOCK 宣言の列〉で構成さ

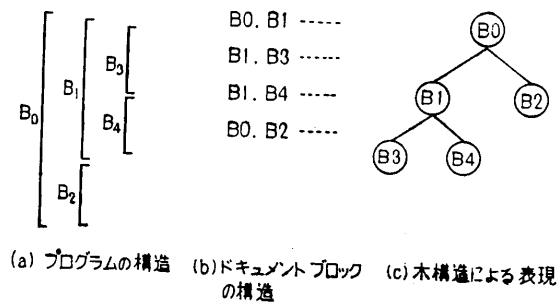


図1 ドキュメント木の構成例

Fig. 1 An example of a document tree.

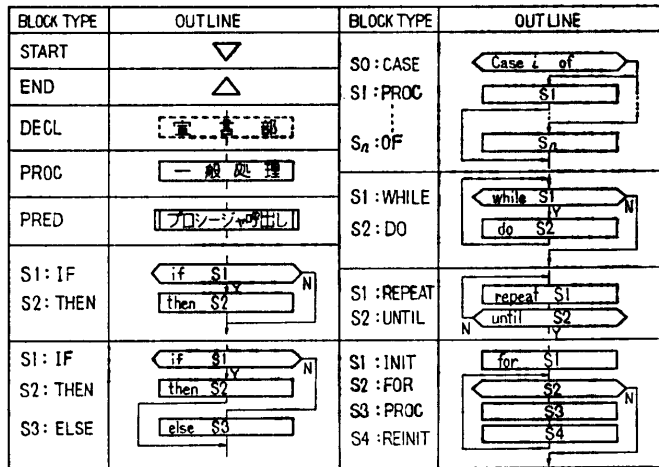


図 2 D-BLOCK と流れ図の対応

Fig. 2 Flow-chart structures and their D-BLOCK declarations.

れ、〈D-BLOCK 宣言の列〉は以下のような構成をしている。

- 〈D-BLOCK 宣言の列〉 ::= 〈D-BLOCK 宣言〉 | 〈D-BLOCK 宣言〉 〈D-BLOCK 宣言の列〉
- 〈D-BLOCK 宣言〉 ::= 〈D-BLOCK 名〉 〈D-BLOCK 名〉 :  
START 〈内部 D-BLOCK 宣言の列〉  
END
- 〈内部 D-BLOCK 宣言の列〉 ::= 〈内部 D-BLOCK 宣言〉 | 〈内部 D-BLOCK 宣言〉 〈内部 D-BLOCK 宣言の列〉
- 〈内部 D-BLOCK 宣言〉 ::= [ 〈D-BLOCK 名〉 : ] 〈D-BLOCK 本体〉
- 〈D-BLOCK 本体〉 ::= 〈D-BLOCK TYPE〉 〈日本語 コメント〉 /
- 〈D-BLOCK TYPE〉 ::= DECL | PROC | PRED | IF | THEN | WHILE | DO | REPEAT | UNTIL | INIT | FOR | OF | REINIT | CASE | ELSE

〈D-BLOCK 名〉は英文字で始まる 8 文字以内の文字列で D-BLOCK の名前を示す。“.” 記号を用いた連結により D-BLOCK 間の親子関係を表現している。

図 2 に START END および各 D-BLOCK TYPE とプログラムおよび一次元流れ図の概形を示している。

### 3.3 漢字かな混り文への変換

一次元流れ図およびプログラムテキスト中の漢字かな混り文では英数字列、特殊記号の出現頻度が高く、これに漢字、かたかな、ひらがなを加えた文に対処する汎用性が要求される。入力は一文字種の文字列を区切記号（制御文字）で囲み文字列ごとの変換を行う。

カードで入力されたデータは漢字列以外は制御文字の指定にしたがって一意的に変換される。漢字への変換は 2 レベルの辞書を用いて単語単位で行っている。

レベル 1 の辞書は、ユーザが自分自身で作成する辞

書で対象としている応用プログラム特有の用語を事前に登録しておくもので漢字への変換に際しては最初に検索される。レベル 1 で変換が不能の用語はレベル 2 の辞書を用いて変換が試みられる。この辞書はコンピュータ用語辞典や語彙調査より求めた専門用語 600 語（演算、変数など）に三省堂の新明解国語辞典の約 7 語の見出し語の内、最重要語、重要語としてあげられている約 5,000 語のなかから、正書法に漢字の含まれていない、ソフトウェアドキュメントに現われる可能性が極めて少ないと考えられる語（愛、愛情など）同音異義語などを選択的に取り除いたもの約 3,600 語を加え合わせて約 3,900 語からなる。変換は制御文字@で囲まれた文字列を単語とみなして一致する単語を辞書

を検索することによって探し、一致するものがあれば変換を行い、なければひらがなに変換する。ソフトウェアの日本語ドキュメントにあらわれる漢字を含む単語の内何%程度が上述したレベル 2 の辞書に含まれているかは興味ある問題である。ここでは一例として文献 9), 10) に示されているプログラムのコメント（日本語に翻訳されている）および文献 11) に示されている算法の記述部分を取り上げ漢字を含む単語を数えあげレベル 2 の辞書に含まれている数を求めた（表 1）。

漢字を含む単語の内出現頻度にして約 63% が（最）重要語に含まれ、26% が専門単語に含まれている。したがって全体として約 90% 近い単語が上記辞書に含まれていることになり専門単語の方が単語あたりの出現頻度は高くなっている。対象を算法のドキュメントという比較的小さい領域に限り、辞書をより充実していくことによりさらに高い変換率が期待できる。

なお、文献 9) のプログラムは実用されているプログラムの一部を抽出してきた例があげられているが、この場合は対象としている応用に依存した用語（給与俸給表など）が多く出現し上記辞書に含まれない単語が散見された。このような単語は事前にレベル 1 の辞書に登録しておく必要があるものである。

表 1 調査結果  
Table 1 Results of an investigation.

漢字を含む 単語の総数	異なり 語総数	（最）重要語に含 まれるもの		専門用語に含まれ るもの	
		総語数	異なり語数	総語数	異なり語数
2,967	633	1,869	350	769	103

### 3.4 ドキュメント出力用言語

ユーザがソフトウェアを理解する際に、最初から詳しいレベルのドキュメントを見ても容易に理解しにくい。またプログラムの隅々まで詳しく理解する必要がないかもしれない。このようなときにユーザの要求にあわせたレベルの詳しさをドキュメントを構成したり必要な部分だけのドキュメントを構成できれば便利である。ドキュメント木はその構造から終端節に近づくほど詳細なドキュメントとなり各節はプログラムのモジュールと対応している。このことから考えてプログラムの各部分に関して任意のレベルのドキュメントを混合して一次元の流れ図として出力したり、モジュールごとの一次元流れ図を指示できるようコマンドが設けられている。

どの D-BLOCK のドキュメントが必要か、また内部ブロックのネストがどの程度まで必要かを指定するために本システムでは OUTPUT と NEST という2つのコマンドを設けている。

OUTPUT 指示で指定された D-BLOCK は出力の対象となり、そのままの形で出力される。ただし、以下に NEST 指示された D-BLOCK に付随しておけば、その D-BLOCK を内部にネストして出力する。

### 3.5 ドキュメント作成システムの構成

本システムは神戸大学工学部システム工学科に設置された中形電子計算機システム (FACOM 230-38) で実現している。ドキュメントはユーザの指示により蓄積管形ディスプレイ端末、静電形プリンタに一次元形式の流れ図およびソースプログラムリストの形で出力される。文字パターンは JIS で定められた約 6,000 文字のパターンすべてを準備しているが、漢字以外の文字パターンは変換字に主記憶に常駐させ変換の高速化をはかっている。出力結果に対する編集はエディタにより行うことができる。

漢字変換の時間は、かなに換算して 1,000 文字の入力データを変換するのに約 30 秒の計算時間を要し、静電プリンタに 1 ページプリントするのに約 20 秒要している。ソフトウェアの規模は漢字変換部を含めて

```

マージソートのプログラム
PROCEDURE MARGESORT;
  TYPE ITEM=RECORD KEY: INTEGER END;
  TYPE INDEX=0..N;
  VAR A: ARRAY(1..2*N) OF ITEM;
  VAR I,J,K,L,T: INDEX;
  H,M,P,Q,R: INTEGER; UP: BOOLEAN;
  Aはインデックス1..N, 2*Nを指している。
  BEGIN UP:=TRUE; P:=1;
  REPEAT H:=1; M:=N;
  IF UP THEN
  BEGIN H:=1; J:=N; K:=N+1; L:=2*N
  END ELSE
  BEGIN K:=1; L:=N; L:=N+1; J:=2*N
  END;
  REPEAT
    A←ケブスをIとJからとりどしマージしてKへ
    Q:=I-RUNの長さ R=J-RUNの長さ
    Q:=P; R:=P; M:=M-2*P;
    WHILE (Q≠0) & (R≠0) DO
      マージ
    BEGIN
      IF A(I).KEY < A(J).KEY THEN
        BEGIN A(K):=A(I); K:=K+H; I:=I+1; Q:=Q-1
        END ELSE
        BEGIN A(K):=A(J); K:=K+H; J:=J-1; R:=R-1
        END;
      END;
      J-RUNの残り部分のコピー
      WHILE R≠0 DO
        BEGIN A(K):=A(J); K:=K+H; J:=J-1; R:=R-1
        END;
      I-RUNの残り部分のコピー
      WHILE Q≠0 DO
        BEGIN A(K):=A(I); K:=K+H; I:=I+1; Q:=Q-1
        END;
      H:=~H; T:=K; K:=L; L:=T
      UNTIL M=0;
      UP:=~UP; P:=2*P
      UNTIL P=N;
      IF ~UP THEN
        FOR I:=1 TO N DO A(I):=A(I+N)
        END;
      マージソート
  
```

図3 マージソートプログラムリスト

Fig. 3 Program list of merge sort program.

FORTRAN で約 4,500 行である。

### 3.6 構造化ドキュメントの生成例

$N$  個の整数要素  $A(1), A(2), \dots, A(N)$  のシーケンスをマージソートの手法を用いてソートするプログラムを例に挙げる。プログラムを PASCAL で記述したものを図3に示す。図4、図5はドキュメント記述用言語で記述された一次元流れ図を直接出力させたものである。図6は NEST したもので、

```

OUTPUT  MASORT
NEST    S0
NEST    S1
NEST    S2
NEST    S3

```

に対する出力例である。

## 4. 結 論

ソフトウェアを資源として有効利用していく上で、その機能、構造などを第三者に伝える高品質なドキュ

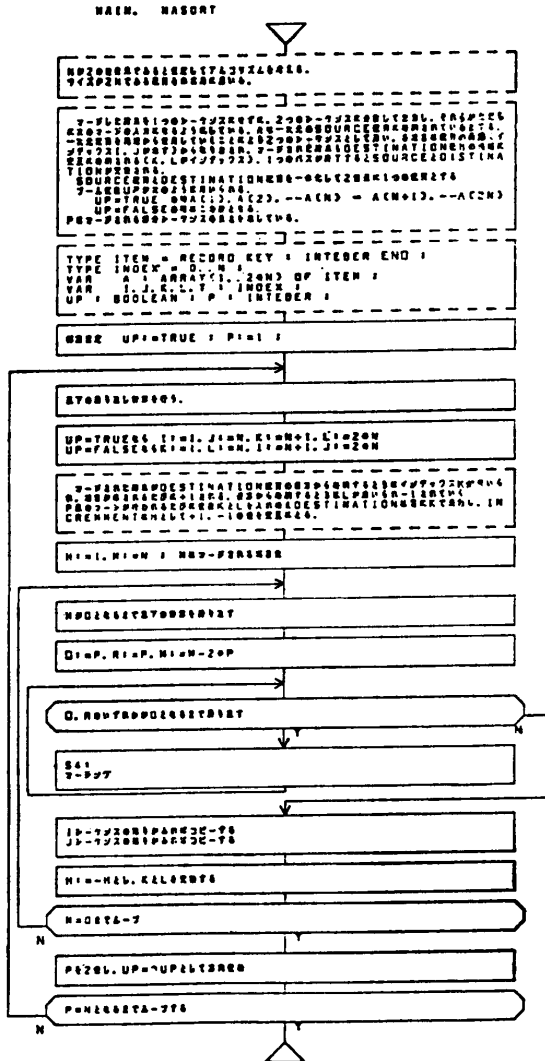


図 4 各 D-BLOCK の一次元流れ図 (I)  
 Fig. 4 One dimensional flow charts of each D-BLOCKS (I).

メントの存在は不可欠なものとなってくる。本システムにおける構造化ドキュメントの作成はプログラムリストだけを残しているプログラマにとっては労力の増加になる。しかしドキュメントが将来利用される有効な情報として残ることを考慮すればこの種の労力は必ず必要なものとなる。

本システムの特徴を列挙すると、

1. プログラマは構造化されたプログラムと分かり易いドキュメントを残すようになる。
2. 日本語文のドキュメントとして出力されるので理解し易い。
3. 任意のレベルの詳しさとドキュメントを構成できる。

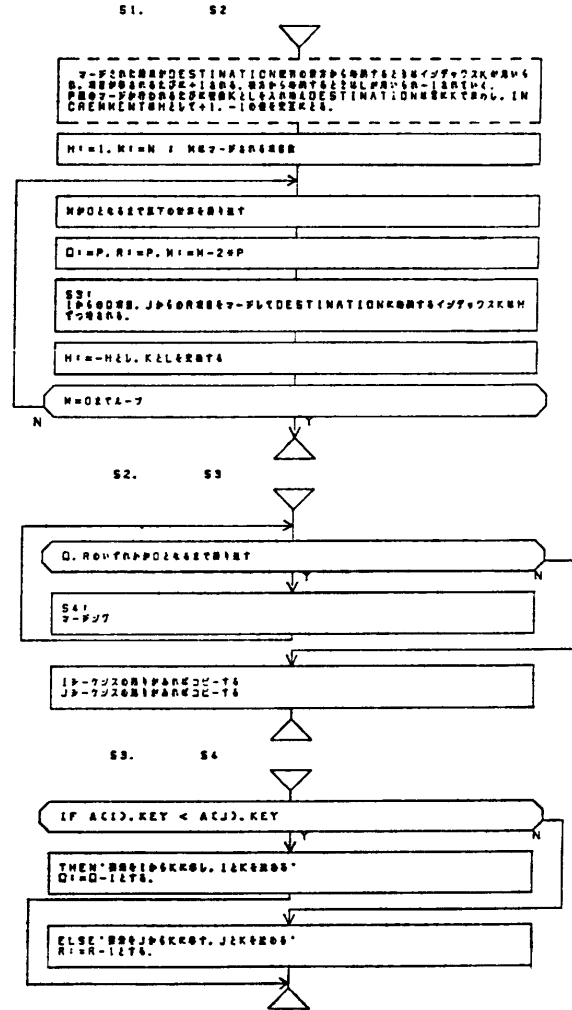


図 5 各 D-BLOCK の一次元流れ図 (II)  
 Fig. 5 One dimensional flow charts of each D-BLOCKS (II).

4. モジュール単位のドキュメントを構成できる。
  5. アルゴリズムに対する詳しい説明と、流れ図の持つ明解さを備えている。
  6. 計算機がブロックごとに管理しているので、プログラムの一部変更にもなうドキュメントの修正が容易である。
- などを挙げる事ができる。

今後は辞書および編集機能の充実を行うとともに文献 3) などにあげられている汎用性あるアルゴリズムのドキュメントおよびプログラムのデータベース化の研究を進めていくことを予定している。本システムは汎用性あるプログラムの時間的および空間的距離を隔てた利用、ソフトウェア情報のデータベース化、プログラミング教育の CAI など広い応用が期待できる。

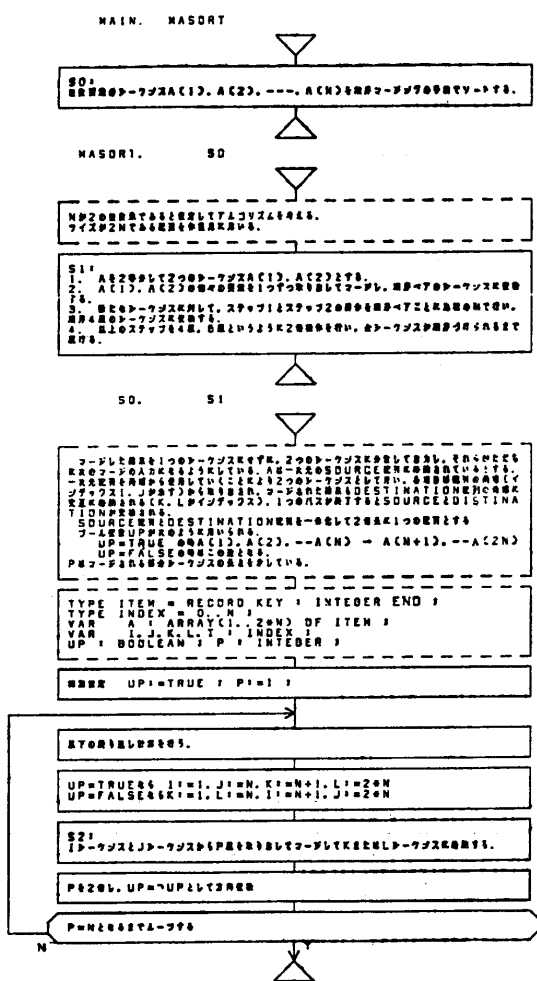


図 6 NEST された一次元流れ図  
Fig. 6 The nested one dimensional flow chart.

**謝辞** 本研究を推めるに当ってデータ提供など、御援助下さった電子技術総合研究所人間機械研究室渡辺定久室長、推論機構研究室淵一博室長に深謝します。

**参 考 文 献**

- 1) 鳥居, 杉藤, 真野, 二木: プログラム作成技術の現状に関する調査報告 [ I ], 電子技術総合研究所調査報告, 第 185 号 (1975).
- 2) Dahl, O. J., Dijkstra, E. W., Hoare, C. A. R.: Structured Programming, Academic Press (1972).
- 3) Knuth, D. E.: The Art of Computer Programming, Addison-Wesley (1975).
- 4) Wirth, N.: Algorithms+Data Structure=Programs, Prentice-Hall (1976).
- 5) Brooks, F. P., Jr.: The Mythical Man-Month, Addison-Wesley (1975).
- 6) Stevens, W. P. and Constantine, L. L.: Structured Design, IBM Syst. J., pp.115-139 (1974).
- 7) 松下, 山崎, 佐藤: 漢字カナ混り文変換システム, 情報処理, Vol. 15, No. 1, pp. 2-9 (1974).
- 8) 牧野, 勝部, 木澤: カナ漢字変換の一方法, 情報処理, Vol. 18, No. 7, pp. 656-663 (1977).
- 9) 木村 訳: プログラム書法, 共立出版 (1977).
- 10) 牛島 訳: コンパイラ作成の技法, 日本コンピュータ協会 (1977).
- 11) 米田, 寛 訳: 基本算法, サイエンス社 (1977).

(昭和 53 年 4 月 8 日受付)

(昭和 53 年 10 月 20 日採録)