英文・図表清書システム FROFF について†

安部憲広†† 東出正裕††† 辻　三郎††

近年，国際学会が頻繁に開催され，私達も英論文発表の機会が多くなっており，手軽に品質の良い英文書の作成できるシステムが必要となっている．現在までのところ，アメリカではベル研の TROFF，東大大型計算センターの PROFF のように写植機を出力装置とする良いシステムが作られているが，文書に不可欠の図表，数式等の処理までも含めたものはあまりない．

本論文は，図表も含めて英文を清書するシステム FROFF (Flexible ROFF) について報告する．FROFF は写植機よりポピュラなドットプリンタを出力装置とし，英文とともに図表をも同時に出力できる上，フォントの指定，字体の変形（イタリック化）等が動的にでき，さらに基本コマンドを用いて複雑な処理を行わせるためのルールの定義が許されていて，印刷文書に劣らない品質の文書が可能になっている．

なお，システムの大部分は FLISP で書かれていて，新機能の追加が容易なシステム構成となっている．

1. まえがき

学術交流の進展とともに，品質の高い英文書作成システムが必要となっている．タイプライタで英文を打ってみると右端がふぞろいとなってしまうが，印刷製本された文書と同等なものが手軽に利用できることが望ましい．こうした状況をもとに文書処理プログラム ROFF が開発され，いくつかのシステムが稼動中である．

私達の研究室でも ASSEMBLER で ROFF[8] を作成し，現在当研究室および東大生研で稼動中である．しかし，この ROFF は従来の ROFF と同様出力媒体をタイプライタとしていたため，出力文字の大きさ（ピッチ）が一定であり，単語間のばらつきが目立ち，写植機や組み版によって得られた文書に比して劣る点が少なくなかった．

そこで最近ポピュラとなりつつあるドットプリンタの特徴を活かして，各文字に固有のピッチを与え出力様式の美化を計るとともに，図表なども容易に出力しうるシステム FROFF (Flexible ROFF) を作成した．私達の必要とする文書には図や表は欠かせぬものであり，英文と同様に取り扱いうる必要がある．

ところで各文字に固有のピッチを与え，より高品質の文書を作成するために写植機を用いたシステムが，ベル研 (TROFF)[5]，東大大型計算機センター (PROFF)[6] 等で作成されているが，我々のシステムは出力媒体としてよりポピュラなドットプリンタを使用している点と，図表などが扱い得るという点で新たな機能を有している．なお，システムはその大部分が FLISP[10),11)] で，そして画像の操作，文書の出力等は FORTRAN，ASSEMBLER で書かれている*．

2. FROFF のハードウェア構成

図1にハードウェアの構成を示す．使用計算機はミニコン HP 2108 A (68 kB) であり，これに 256 kB の IC メモリ (FLISP の自由リスト領域，および画像メモリとして使用される)，ドットプリンタ，テレビカメラ，ジョイスティック等より成り，FLISP システム[11)]と動画像処理システム[12)]とが協調して稼動できるようになっている．それぞれの詳細に関しては既発表の各論文に譲る．

3. FROFF の主特徴

通常の ROFF の機能としての右端調整 (justification)，ページ，段落，行間隔制御などは当然のものとして，FROFF は以下の特徴を有している．

1）豊富なコマンドとユーザ定義によるルールの使用にもとづく自由な書式の実現，

2）フォント，文字の大きさの動的な選択が可能，

3）小文字のない端末からもテキストが入力できる，

† On FROFF: Text-Processing System for English Texts and Graphs by NORIHIRO ABE (Faculty of Engineering Science, Osaka University), MASAHIRO HIGASHIDE (Nippon Electric Co., Ltd.) and SABURO TSUJI(Faculty of Engineering Science, Osaka University).

†† 大阪大学基礎工学部制御工学科
††† 日本電気(株)

* 現在 FLISP は，他言語のスケジュール機能，画像処理関数を有している．

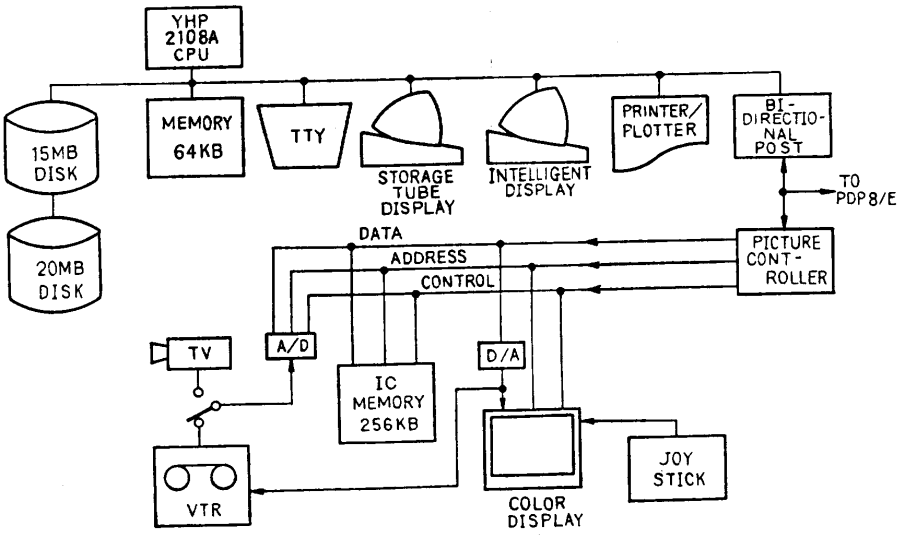

図 1 FROFF のシステム構成
Fig. 1 Configuration of FROFF system.

4) 出力位置の動的な指定が可能,

5) 画像データの出力が行える,

6) 種々の直線の表示, 表の作成が行える,

7) 入力装置としてジョイスティックやTVカメラなど, 種々のものが使用できる,

各項目について順次詳述することにする.

1) 上記2)～7) の機能をはたすため, 現在60種のコマンドが用意されていて, 柔軟できめの細かい出力が得られるようになっている. しかしコマンドのみでは使用経験の少ない初心者は使いにくいため, コマンド使用の簡易化を目的として, ルールの定義と使用とが可能になっている. ルールを用いることで出力の一定形式(タイトル, 目次, 著者名, 文献など), あるいは特殊形式(数式, 特殊文字, 表など), および各自の希望に応じた書式が手軽に実現できる. またこうしたルールの内で標準的な様式のもの, 特に有効なものはシステム組込みのルールとして宣言するだけで使用できるようにもなっている. 具体例については後述する.

2) 文字のピッチを各文字固有のものとして, 単語のばらつきを良くしている. 図2-(a) にはIとWとの文字パターンが示されているが, それぞれ 13×32, 29×32 のようなピッチを有している (単位はドットである). 図2-(b) がタイプライタ, 図2-(c) が FROFF の出力の一例であるが, 明らかに後者の方がすぐれている. また例からもわかるように ROMAN, BOLD, など各種のフォントの動的な選択, 字の斜体化 (イタリック化) が可能である. 現在フォントは 14 種用意されているが, 60種まで設定可能である. また字の大きさも基本パターンの拡大, 圧縮によって任意の大きさに選ぶことができる. もちろん, あまりに拡大された文字は量子化誤差が無視できず読みずらいものとなるが, 通常の使用には不都合はあまりない (図3参照).

3) 入力テキストの作成が小文字のない端末でも行える, すなわち入力文字の大小にかかわらず出力は原則としてすべて小文字で出力する. しかし改行時 (段落指定などによる), あるいはピリオドの直後は文の先頭と判断して, 最初の単語の1字は大文字化させる. ただし, "i.e.", "e.g." などは例外として扱っている. ところがこのようにすると, たとえば FLISP SYSTEM のようにすべてを大文字にしたり, An Example of the Output や McCarthy, 参考文献の印字の場合など, 大文字, 小文字の混在する場合うまくゆかないので, 文字サイズを変更したり, 記号列を連接 (concatenate), 分離するコマンド, ルールが用意されている (5, 6 で詳述する).

4) 文字の出力位置を直前の出力位置に対して相対的に上下左右に移動させることができる. この場合の移動量はダイナミックに決定することが可能である. これは出力テキストの空間的にしめる大きさによって次の出力位置が決まる場合には不可欠である. たとえば図4のように式を箱で囲むようなことは, 実際に囲まれる式が出力された後に初めて可能である. また FROFF の処理結果は印字前にまずディスクファイル

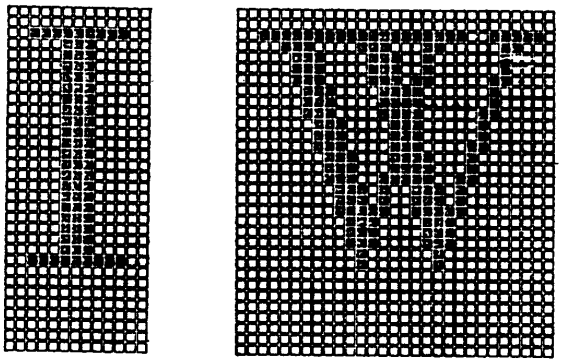

図 2 (a) I と W の文字パターン
Fig. 2-(a) Character pattern I and W.

In order to meet the needs of a publication of papers in English, many systems to run off texts have been developed. In this paper, we report a system FROFF which can make a fair copy of not only texts but also graphs and tables indispensable to our papers. Its selection of fonts, specification of character size are dinamically changeable, and the typing location can be also changed in lateral or longitudinal directions. Each character has its own width and a line length is counted by the sum of each character. By using commands or rules which are defined to facilitate the construction of format expected or some mathematical expressions, elaborate and pretty documents can be successfully obtained.

図 2 (b) タイプライタによる英文例
Fig. 2-(b) English text by a typewriter.

In order to meet the needs of a publication of papers in English, many systems to run off texts have been developed. In this paper, we report a system **FROFF** which can make a fair copy of not only texts but also graphs and tables indispensable to our papers. Its selection of *fonts*, specification of *character size* are dinamically changeable, and the typing location can be also changed in lateral or longitudinal directions. Each character has its own width and a line length is counted by the sum of each character. By using commands or rules which are defined to facilitate the construction of format expected or some *mathematical expressions*, elaborate and pretty documments can be successfully obtained.

図 2 (c) FROFF の出力例
Fig. 2-(c) An example of FROFF output.

FROFFLISP

図 3 利用可能な字のサイズの一例
Fig. 3 An example of available character sizes.

$$\frac{\mu^{\frac{1}{2}}}{\frac{1}{\sqrt{3}}+\left|\frac{x}{y}\right|-n_0}$$

図 4 箱で囲んだ数式の一例
Fig. 4 A mathematical expression enclosed by a box.

に出されるため(後述4参照)，入力テキストが必ずしも出力順に並ぶ必要はなく，1つのページ内ではその出力行を2列に分割したり，図表番号をあとから記入することが許されている．たとえば表1はまず表の枠を書き，その後表の内容を書き込んで作ったものである．

5) 図のない文献はないといって良いほど，文献に図は不可欠である．特に私達の研究室のように画像関係の研究を行っている所では，処理例のサイズを指定して直接文書中に結果が挿入できることは非常に便利である．この場合，画像の二値化，反転，一部分の切出し等が可能である．**図5**(a)(b) は同図形の出力サイズ

表 1 コマンドとルールの例
Table 1 An exampl of commands and rules.

TYPE	SYMBOL	DEFINITION
COMMAND	.A *n*	Selection of font of characters
	.X *n*	Alter a width of characters into *n*
	.L *n*	Space out *n* lines
	.Pl *n*	Set a length of page to *n*
	.B *n*	Definition of character size: big or small letter etc.
	.Ox *n*	Sift an output location to horizontal direction
	.Joy	Accept a coordinate by the joystic
	.Pic	Output a picture
	.Line	Drow a line
	.Box	Output a box
RULE	*It	Print an expression in italic form
	*Sf *a b*	Make an expression with a suffix: a_b
	*Sp *a b*	Make an expression a^b
	*Fra *a b*	Print a fractional expression a/b
	*Sigma 3 *a b c*	Make an expression $\sum_a^b C$
	*Theta	Print a Greek letter θ
	*Phd osaka	Output a string Ph.D. Thesis, Osaka University,
	*Joho 5	Print a string Information Processing of Japan, 5

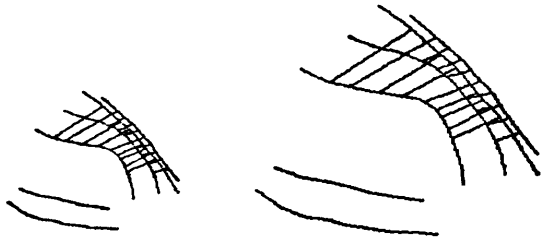

図 5 (a) (b) 図形の出力例
Fig. 5-(a), (b) Picture output from a disc file.

を変化させた例である（例は心臓の解析結果である）.

6）水平，垂直に限られてはいるが，種々の直線の出力ができる．直線は，出力位置，長さ，太さ，実・破線，一点鎖線等自由な指定が許されている．**図 6** には直線の例を，**図 7** にはそれをもとに書いた Box の例を示す.

7）入力テキストは通常は HP 2108 A のテキストエディタによってディスクファイルに入れ，そこから受け取るが，ジョイスティックやTVカメラより入力することもできる．**図 8** の図は**図 9** の線画をTVカメラより入力し，記号をジョイスティックによって与え

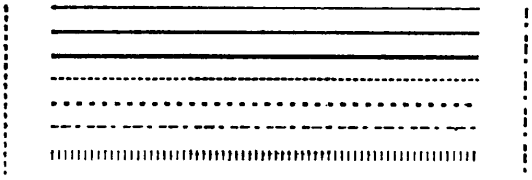

図 6 種々の直線の出力
Fig. 6 An output of lines.

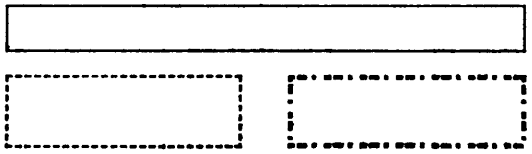

図 7 コマンド .Box によるボックスの出力
Fig. 7 Boxes drawn by the command .Box.

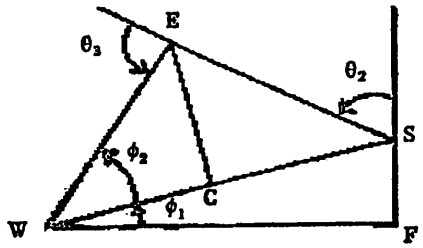

図 8 TV カメラ，ジョイスティックによる図形の入力
Fig. 8 An example of the picture given by TV-camera and joystick.

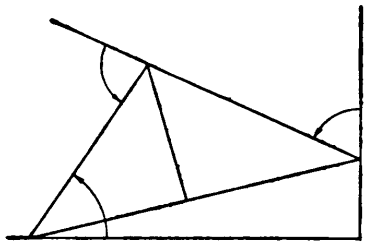

図 9 図 8 の原画像
Fig. 9 An original picture for Fig. 8.

たものである．現在のところTV入力された図の整形プログラムは作られてはいないが，直線，円弧の整形ルーチンを加えることにより，将来レタリングも不要になるであろう.

4. FROFF の構成

ルール，コマンドについて詳述する前に，FROFF の全体的構成について概説しておく．**図 10** に示すように入力テキストが読み込まれると，まずルールの定義・解釈が行われる．すなわち入力テキスト中にルールとして定義された記号があるか否か調べ，あればそ

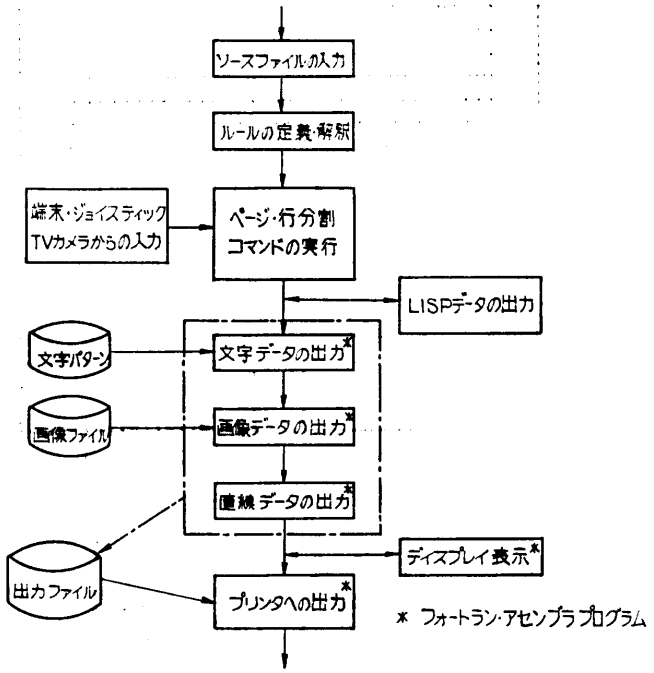

図 10 FROFF のシステム・フロチャート
Fig. 10 A system flowchart of FROFF.

れをその定義体で置換し，次にコマンドを解釈・実行することによってページ・行分割を行い，結果として得られる FLISP のリストをディスクファイルへ出力する．この場合，ジョイスティック等が利用できる．

ディスクファイル化されたデータはバイナリ変換された後，FORTRAN, ASSEMBLER で書かれたプログラムによって文字データ，画像，直線などが出力される．この場合，ドットプリンタに出力するかわりに，ディスプレイに表示することにより，図表の配置等の是非をチェックすることができる．

ところで，流れ図中で途中結果をディスクへ一時確保しているが，これはたとえばスペル等のミスを修正する場合，最初からやり直すのではなく，この途中結果から修正した方が良いためである．

5. コマンドとその例

入力テキストは，テキスト行とコマンド行からなり，いずれも空白をデリミタとする．コマンド行はピリオドで始まる行であり，コマンドの引数には整数(cm, mm の単位をつけることができ，ない場合はドット数を表す)，変数(ユーザ定義の変数あるいはシステム変数) が使え，さらに加減乗除の演算子が使える．簡単にこれらを説明する．まず，変数についてであるが，**表 1** のコマンド例中，.A n は n で示されるフォントを選択することを示しているが，この n はシステム変数 %A の値として保持されている．そして，ある部分だけをほかのフォントの文字で印字して，その後再び元のフォントに戻す場合には，.ST %A によって n の値をスタックに保存した後，コマンド .A n' によってフォントを変更して印字し，最後に .RS %A によってフォントを回復させることができる．しかし，より細かな書式を設定するためには，上のようなシステム変数以外に，適当な補助変数も必要であり，それがコマンド .SET によって導入できるようにもなっている．形式は .SET arg value であり，変数 arg に値 (または変数値) value を設定することができる．その際，value 部には演算子を含む表現が許されている．たとえば .SET Y /-XZ 2* によって変数 Y は $(X-Z)/2$ という値にセットされる．表 1 の線びき，項目の配置等は，こうした変数を使用したルールを用いて実現されており，コマンド列を工夫することによって非常にきめの細かな様式の設定が可能となっている．また，図 5, 6, 7 では図，直線の例を示したが，そのサイズや位置はドット単位，cm, mm 単位のいずれでも，また混在したものでも良いようになっている．

6. ルールとその例

現在，システム中にはすでに 100 程度のルールが組み込まれていて，その使用を宣言することによってただちに使うことができるが，ユーザは各自の希望する様式を，コマンドを組み合わせたルールを定義することによって実現することができる．

ルールを定義するためにはコマンド .. を使用し，次の . だけの行までが 1 つのルールであると解釈される．**図 11** にはルールの例が 2 つ示されているが，それぞれについて簡単に説明してみよう．まず，＊MAC という名の引数なしのルールが定義されている．その意味は，%B の値 (コマンド .B m の m を保持しているシステム変数) を保護した後，各ユニット**の先頭文字を大文字で書くことを指定し (.B 2)，記号 Mc を印字し続いて Carthy と書く (.BI が Mc と Carthy を連接させる)，そして最後にコマンド .RS %B によってルール適用前の状態を回復させることである．す

* 演算はポーランド記法による．
** ルール，コマンド以外の記号列のことである．

```
.. *MAC
.ST %B
.B 2
MC
.BI
CARTHY
.RS %B
.
.. #0 X
.ST %B
.B 0
X
.RS %B
.
```

図 11 コマンドを用いたルールの例

Fig. 11 Two rules defined by using commands.

なわち，テキスト中に記号 *MAC が出現すればそれを McCarthy と置換して出力させることをこのルールは意味している．2番目のルール #0 は引数を1つもつルールであるが，これはたとえば #0 FLISP を前後の出力とは独立に大文字で表示する機能をもっている*．ここに示したルールはきわめて簡単なものであるが，実際に用いられているルールの中には20〜30ほどのコマンドを組み合わせた複雑なものも多く，これらはシステム組込みとして利用され，大へん役に立っている．そのごく一部ではあるが，二，三の例を表1に示しておくことにする．表中 *PHD, *JOHO 等は，私達の良く利用する文献名などを出力させるためのルールであり，現在 25 種の文献名に対してルールが組み込まれている．

7. ルールの数式，図表への適用

6では記号列印字へのルールの適用について述べたが，文書に不可欠の数式，表なども容易に作成できなければならない．その中でも，サフィックス，スーパサフィックス，分数，平方根，絶対値，シグマ形式などは基本的に必要なものであるし，また θ, π, ∞ といった記号がよく数式に使われることを忘れてはならない．FROFF はこれらを組込みのルールとして有しており，その一例を表1に示してある．そして，複雑な数式は，これらのルールを用いて，数式をポーランド記法化することによって表現するようにしている．たとえば，図12の(2)に示されている分数式は次の表現

*FRA D *2 $A+$ *FRA C B

によって書かれている．ここで，*FRA は表1にあるように引数を2つもつ分数表記のためのルールであり，*2 は同様に引数を2つもつ引数結合ルール（*MAC で用いたのと同じコマンド .BI で2つの引数を結合する）である．上式の意味は，Dを分子とし，分母を $A+$ と C/B とを結合したユニット $A+C/B$ とする分数を出力せよということになる．最初，不慣れな間は，この記法にはなじめぬことも考えられるが，少しの練習によって容易に希望する数式が得られるようになっている．

$$\frac{\sqrt{2}-1}{\sqrt{2}} \qquad \frac{D}{A+\frac{C}{B}} \qquad \sum_{i=1}^{\infty} a_i + \sum_{i=1}^{k} \frac{|a_i+a_{i+1}|}{a_{i-1}}$$

(1) (2) (3)

図 12 数式の例

Fig. 12 Some mathematical expressions.

表に関しても，表1のように，各項目の縦幅が一定の表，種々のサイズのもの，また表の項目の書込みの位置などを必要に応じて指定することのできるルールが組み込まれている．現在のところ，まだ n 乗根，$\tilde{x}, \tilde{y}$ などに関するルールは組み込まれてはいないが，既存のルールと同様に，3の4)の機能を用いることによって，定義・利用することができる．

8. 文字パターン生成について

以上でコマンド，ルールの詳細について述べてきたが，出力の基本単位である文字パターンに関して少しふれておくことにする．すでに述べたように，現在14種のフォントが利用可能であり，さらに60種まで拡張の余地がある．このフォントの設定には文字パターン生成用の FLISP プログラムが開発されていて，新しい文字パターンの TV カメラからの入力，二値化，位置補正，パターンの修正，ピッチの指定，および文字パターンのファイルへの書込みが，カラーディスプレイ上でジョイスティックを用いることによって，容易に行うことができるようになっており，字種の増設に何の問題点もない．問題点があるとすれば，それは文字を拡大して出力した場合* であり，特に拡大・斜体化（イタリック化）したとき，パターンの補正を行って出力していないため，少し読みづらいものとなる点である．この点は，文字の拡大コマンドにパターンの補正を行わせる機能を加えて改善する必要がある．

* .B 0 は以後，すべてを大文字で，.B 2 は先頭文字を大文字で印字するコマンドである．

* たとえば，表題文の印字などでは，文字の拡大が必要となることが少なくない．

9. 結果および問題点

FROFF の具体例として長い英文を示すのが最適ではあるが，紙面の都合上不可能である．しかしながら，ここまでに引用した例を見ていただければ，得られる文書の質は高く，決して写植機を用いたシステムに劣るものではないことは明らかであろう．

問題があるとすれば，それはシステムのスピードとルールの仕様に関してであろう．まず，速度の点であるが，アセンブラで書かれた ROFF に比して4倍程度の時間が必要であり，入力テキストのさ細なミスを修正して完全な出力を得るまでには少なからぬ時間を要する．特に，比較的システムに経験の乏しいユーザが，組込みルールを使わず，各自が好みの書式をルール化しようとする場合，希望の文書作成にはかなりの時間と努力を要するようであり，組込みルールの強化が不可欠である．ところが，そのルールの形式に現在問題がある．それは，ルールの引数条件が限定されすぎている点である．たとえば表1のルール ＊SIGMA 3 は "＊SIGMA 3 a b c is given…" なるテキストに対して "$\sum_{a}^{b}$C is given…" を出力するが，もし "＊SIGMA 3 a b is given…" と引数が欠けると "$\sum_{a}^{b}$ is given…" となってしまい，"$\sum_{a}$ b is given…" のように引数が2個ならば上限を略すといったような仕様ができない(3つめの引数として is をとってしまう)．この点を，たとえば入力テキストを $[＊SIGMA 3 a b c $] is given…, $[＊SIGMA 3 a b $] is given… のように，ルールとその引数を $[, $] という特別なカッコで囲む LISP 的記法を導入し，ルールを FLISP の lexpr* で記述することによって，引数条件をゆるめルールをより使いやすくすることも必要であろう．

また，別の問題点として，図表の配置，スペルのミスの取扱いがある．図表出力のためには，図表をコマンドやルールで出力させる場合と，図表を挿入するための余白をとる場合とがあるが，これがページの切れ目にぶつかることが少なくない．そうしたときには，図表の一部が欠けたり，その空白に図を挿入できなくなってしまい，コマンドやルールを改めて指示しなくてはならない．この問題を解決するには，最初に図表のサイズを宣言しておき，それを初めて引用する文と同じページ（または，次のページ）に挿入する自動配置を考えなければいけないだろう**．また，ミス・スペリングはより重要な問題であろう．これには辞書を用いる方法と，そうでない方法とが考えられるが，現状ではいずれも容易ではない．まず後者から考えてみよう．辞書を持つと言葉でいうことはやさしいが，実際実用レベルの辞書を持つことは，現在のところ特に限られたアトム領域しかもたない LISP にとっては非現実的なことである．そうすると，ミス・スペリングは，各単語が以前に出現した語とどう違っているかという比較によって処理せざるを得ない．しかしながら，動詞のように活用したりする語が正しい語か否か(work に対して works, worked は正しいが，take に対して taked は誤りなど)，その語が動詞として使われていることがわからない限り，誤りの指摘はできない (taked という語のないことも指摘できない)．辞書があれば，上の問題の大部分は解決するが，よりレベルの高い誤りの発見については，自然言語処理の発展を待たなくてはならない．もちろん，この辞書は完全なものでなくても良いが，システムは辞書内の単語にだけ責任をもつものであり，ユーザの誤りが標準的な語彙に多発するものか否かを調査しなければ，この方法が実用的なものかどうか不明といわざるを得ない．

* LISP の定義関数の一種であり，任意の個数の引数が許されるタイプの関数である．

** 現在これを可能とするルールを作成中である．

10. おわりに

文書清書システム FROFF について述べた．すでに本システムによって10編の論文が編集され，好評を得ている．システムの大部分が FLISP で書かれているため，機能の追加・変更は容易であり，今後さらに使い良いシステムへと改良してゆきたいと考えている．

末尾ながら，文字パターン生成などで協力をいただいた本研究室の森明雅之君に感謝いたします．

参考文献

1) Shermer, Jack E. and Cherry, Lorinda L.: The Soft Display Word Processor, Computer, 11, No. 12, p. 39 (Dec. 1978).
2) Hartke, David H., Sterling, Warren M. and Shermer Jack E.: Design of a Raster Display Processor for Office Applications, IEEE Trans. on Computers, C-27, No. 4, pp. 337-348 (Apr. 1978).
3) Ida Tetsuo: A Program for Run-off, Roff(1) and Roff(2), University of Tokyo.
4) Tsukuba Computer Center: Tsukuba Run-off

System (1976).
5) Lesk, M. E. and Kernighan, B. W.: Computer Typesetting of Technical Journals on UNIX, Proc. of Ncc, pp. 879-888 (1977).
6) Hasebe, K., Nomoto, S. and Ishida, H.: An On Line Phototypesetter Support System, Computer Center, Univ. of Tokyo (Mar. 1979).
7) Kernigham, Brian W. and Cherry, Lorinda L.: A System for Typesetting Mathematics, Commun. ACM., Vol. 18, No. 3, p. 151 (Mar. 1975).
8) Higashide, M.: Roff User's Manual, Osaka University (Apr. 1978).
9) Higashide, M.: FLISP System and its Application to Run Off, Master Thesis, Osaka University (Feb. 1979).
10) Abe, N. Higashide, M. and Tsuji, S.: An Improvement of FLISP System by Micro Programming, Trans. Inst. Elect. Commun. Engineers of Japan, 62-D (1979).
11) Higashide, M., Konishi, K., Abe, N. and Tsuji, S.: The FLISP System for Mini Computer using B-frame Technique and M-frame Technique, Information Processing of Japan, Vol. 20, No. 1, pp. 8-16 (1979).
12) Yachida, M., Kitani, S., Osada, M. and Tsuji, S.: An Interactive System for Digital Processing of Moving Images, Trans. Inst. Elect. Commun. Engineers of Japan, Vol. 61-D, No. 10, pp. 775-782 (1978).
13) Higashide, M.: FROFF Memo, Osaka University (Mar. 1979).
14) Abe, N.: The FROFF Manual, Osaka University (in preparation).

(昭和54年5月7日受付)
(昭和55年11月20日採録)