

FACOM 222 による HARP ソースプログラムの 形式処理と簡単なエラーチェックについて*

柳 沢 健** 当 麻 喜 弘**

1. まえがき

本報告は、東京大学大形計算センターの計算機（以下大形機と称する）を利用する際、手近にある計算機を用いて大形機のソースプログラム（HARP ソースプログラム）の形式処理および簡単なエラーチェックができれば便利であるという考えのもとに、東工大に設置されている FACOM 222 に上記の作業を行なわせる処理プログラムを作成する研究を行なったので、その結果をまとめたものである。

2. 基本的な考え方

FACOM 222 のプログラム言語 FAST は FORTRAN II と類似なものであり、FAST を会得した者が HARP を理解することは容易であると思われる。そこで大形機利用者は HARP を用いるものとし、処理プログラムは以下の操作を行なう。

(1) 形式処理：従来、東工大で FAST ソースプログラムを書いてきた形式に類似の形式で HARP ソースプログラムを書くものとする。こうして書かれたプログラムを大形機にかけ得る形式に変換する。その際コード変換の問題も処理する。

(2) エラーチェック：前項の形式処理を行なう際、簡単に発見できるエラーをチェックする。

3. 形式処理

東工大の FACOM 222 は紙テープ入力であり、大形機はカード入力である。東大で紙テープからカードに変換するプログラムを開発したので、ここではその利用を考慮して形式処理を行なう。形式処理には次の二つの操作が含まれる。

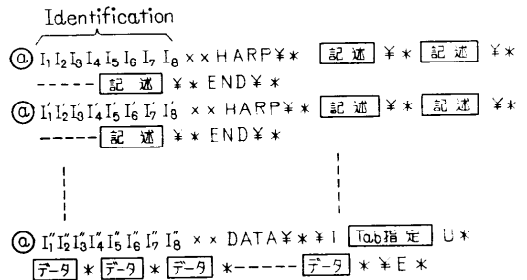
(1) コード変換表の作成

FACOM 222 のコードを大形機のコードに変換するために、ソースプログラムの前にコードを定義づけるコード変換表をつける。実際の変換表は一定の順序でコードを紙テープにパンチしたものとなる。

いくつかの job があるときは最初の job の前のみ変換表がつき、すべての job にいちいち変換表がつくことは避けるようになっている。

(2) 紙テープ形式からカード形式への変換

紙テープにパンチされたプログラムをカードにパンチし直すにはいくつかの Tab 指定を行なわなければならない。東大の紙テープ → カード変換プログラムが要求する形式に従って Tab 記号を挿入するために、処理プログラムは第 1 図の形式のプログラムを第 2 図のように形式処理する。図で頭部形式とは第 3 図のようなものをいう。



- 注 (1) X および記述の空白箇所はスペース記号
- (2) 記述番号は必ず 4 桁使用する
- (3) Continuation は考えていない
- (4) Tab 指定はカードにパンチされるデータの頭の桁位置を指定するもので、該当する位置に 1 を打ち、他はすべて 0 を打ったもの、1 のあとにすべて 0 が続くときは後続の 0 は省略可

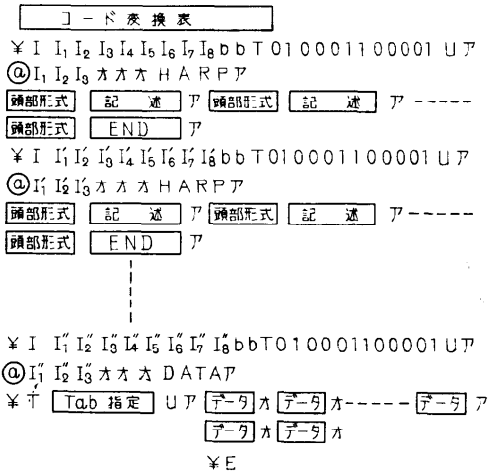
第 1 図 大形機利用者が紙テープにパンチする形式

こうして処理されたプログラムは東大の紙テープ → カード変換プログラムでカードにパンチし直され、その後大形機で実行されることになる*。この経過を

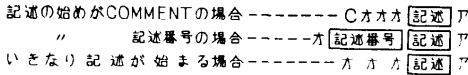
* 最近東大の紙テープ → カード変換プログラムが用い得なくなったので、本処理プログラムも実用に供し得なくなった。

* Rearrangement and Error Checking of HARP Source Program by FACOM 222, by Takeshi Yanagisawa and Yoshihiro Tohma (Tokyo Institute of Technology)

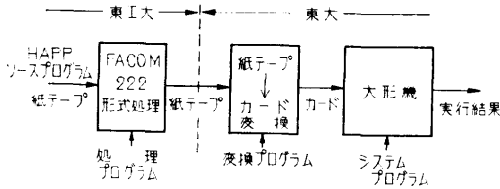
** 東京工業大学理工学部



(注) ¥ (*), ア (CRLF), オ (Tab) は () 内のアテ字
第2図 処理プログラムが紙テープにパンチする形式



第3図 頭部形式



第4図 処理経過

図示すると第4図のようになる。

4. エラーチェック

実際に形式処理を行なう前に、次の点を監視してエラーをチェックする。

- (1) 開カッコと閉カッコの数が一致するか (ERROR TYPE 1).
- (2) 開カッコの数が、 ≤32 か (ERROR TYPE 2).
- (3) FORMAT 記述中の H field 以外は (A, L, O, P, Q, X, I, E, F, D, H, 数字, ., ', /, スペース,) の記号か (ERROR TYPE 3).
- (4) FORMAT 記述中の H field 以外で (E の

数)+(Fの数)+(Dの数)=(.の数)か (ERROR TYPE 4).

エラーが検出されると記述の通し番号(コメントから数える)と ERROR TYPE を印字し、作業は続ける。なお閉カッコを忘れると ERROR TYPE 1 の外に ERROR TYPE 3 を 80 桁目に達するまで印字し続ける。

5. 処理プログラム

処理プログラムを作成するのに FAST 言語を用いるか機械語を用いるかが問題となったが、FACOM 222 は桁単位処理が可能な入出力バッファ(シンクロナイザと称している)を3台持つので、これを直接利用すること、FAST 言語にはロジック(特に部分抽出など)がないこと、などを考慮して機械語を用いることにした。記述単位で処理を行ない、またデータは一括して処理する。

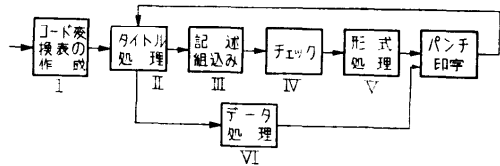
(1) 処理プログラムの構成

処理プログラムは次の七つのセグメントよりなる。

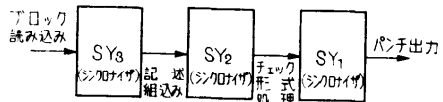
- I コード変換表作成部 (80 語)
 - II タイトル処理部 (85 語)
 - III 記述組込み部 (49 語)
 - IV チェック部 (191 語)
 - V 形式処理部 (48 語)
 - VI データ処理部 (82 語)
 - VII レジスタ・パラメータ部 (44 語)
- 合計 579 語となった。

各セグメントの相互関係を示すと第5図のようになる。また処理経過に応じた情報の流れを示すと第6図のようになる。

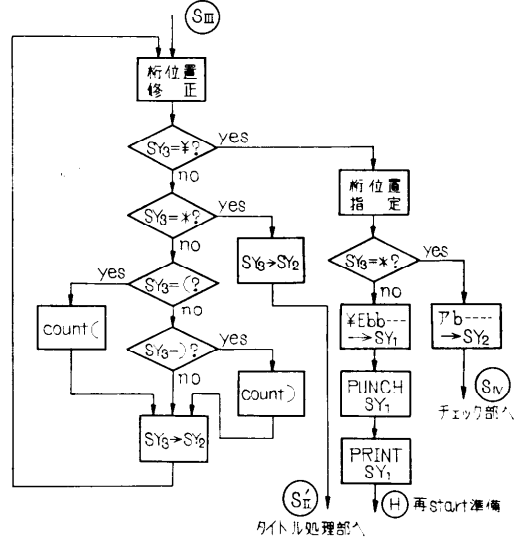
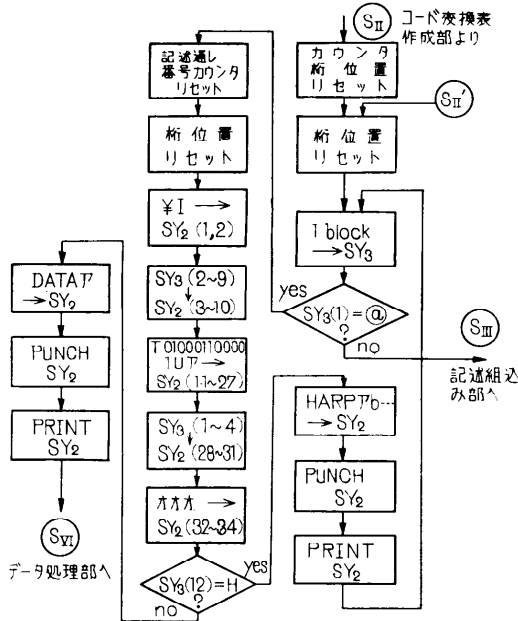
(2) 処理プログラムのフローチャート



第5図 各セグメントの相互関係



第6図 情報の流れ

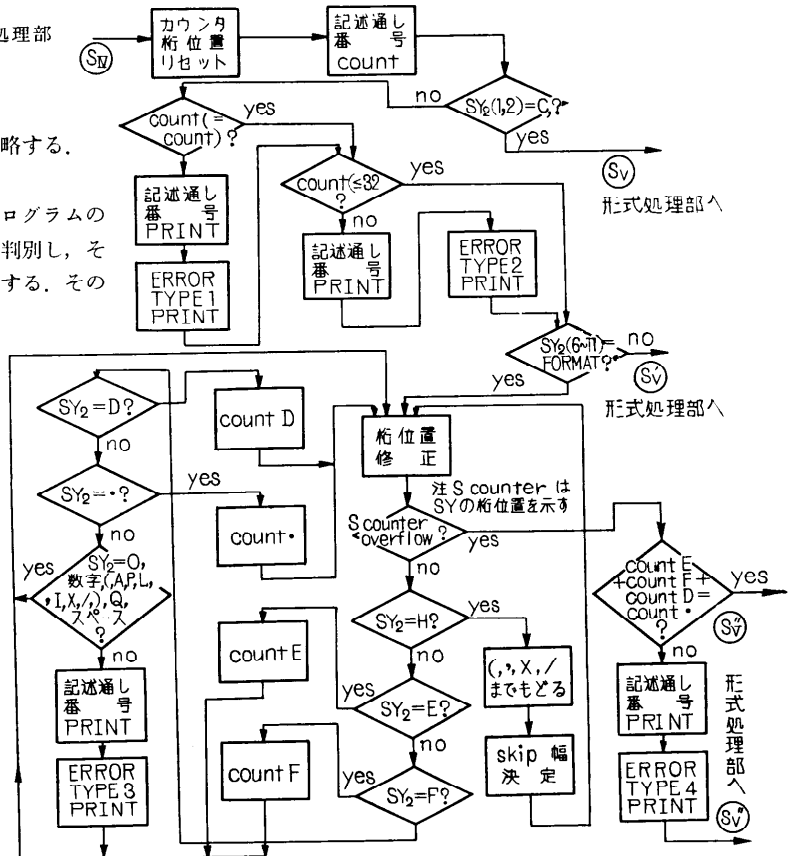


1. コード変換表作成部
記号列をパンチするだけゆえ省略する。

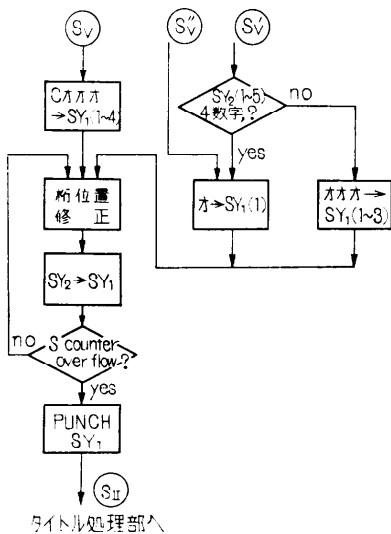
2. タイトル処理部 (第7図)
このセグメントは、HARP プログラムのタイトルかデータのタイトるかを判別し、それぞれのタイトルをパンチ、印字する。そのほか、次の記述組み込みの際のブロック読み込みの機能も含んでいる。

なおパンチはシンクロナイザ SY₂ の 80 桁を一度に行なうので、SY₂ の残余桁には b (ブランク) を書き込むようにしている。

3. 記述組み込み部 (第8図)
紙テープの読み込み速度を速めるためにブロック読み込みを用いた。このセグメントは、ブロックごとに SY₃ に読み込まれたものを SY₂ に一つの記述としてまとめるためのものである。ブロックのエンドマークは乗算記号 * と同じであるため、ブ



第9図 チェック部



第10図 形式処理部

ロックエンドに際し次の操作を行なう。

(i) —*, すなわち単に乗算記号の場合は SY₂ への組み込みを続ける。

(ii) —¥*, すなわち記述エンドの場合は SY₂ への組み込みは完了したことになる。

(iii) —YE*, すなわち形式処理すべき全てのプログラムが終了した場合は ¥E 記号をパンチ、印字し、動作は停止する。

なお、次のチェックに役立てるためこのセグメントで開カッコと閉カッコの数を数しておく。

4. チェック部 (第9図)

SY₂ に組み込まれた記述について、4で述べたチェックを行なう。FORMAT 中の H field に相当する桁は skip する。

5. 形式処理部 (第10図)

チェックを受けた記述を SY₂ から SY₁ へ移しながら、3で述べた頭部形式の処理を行なう。

6. データ処理部 (第11図)

このセグメントは第2図のようにデータの形式処理を行なうものであるが、特に

(i) 1枚のカードに収まるデータの数が (Tab 指定中の1の数) + 1 と等しくなるようにするために、データの数をかぞえながらオ (Tab) 記号、ア (CRLF) 記号をデータ

間に挿入する。

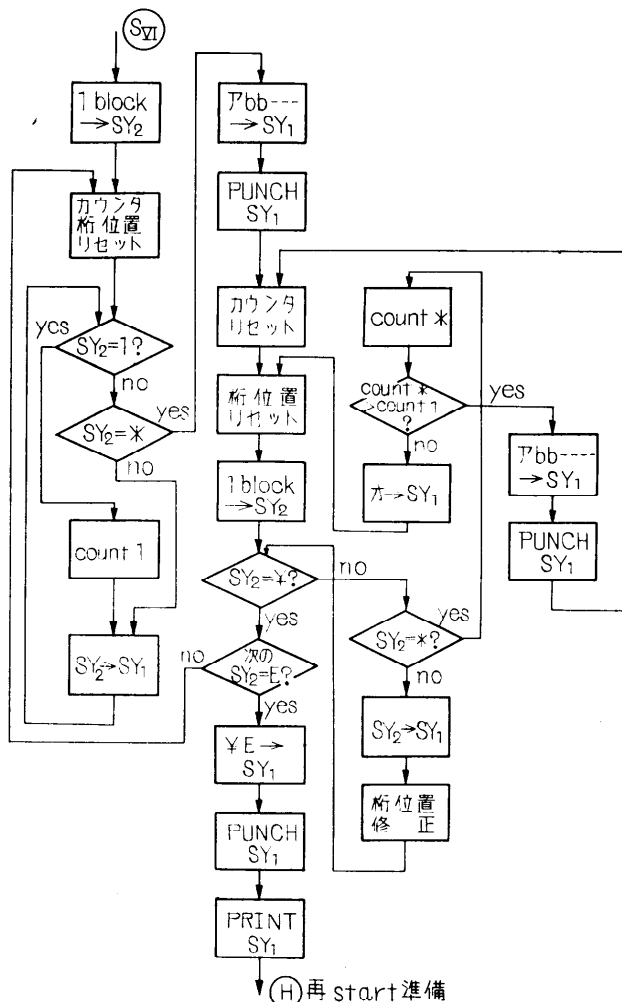
(ii) データの途中で新たな Tab 指定ができる、などの機能を持たせてある。

6. 実例

実例を第12~14図に示す。

第12図下は正しい HARP ソースプログラムを形式処理した紙テープ出力の印字結果である。ア (CRLF), オ (Tab) 記号は、活字がないので印字されず、手で C_R, Tab と記入してある。

第13図下は形式処理が完了し、誤りが検出されなかった場合の印字結果である。



第11図 データ処理部

```

HARP**
C, QUADRATIC EQUATIONS—SUBROUTINE WITH ARGUMENTS**
C, EQUATIONS ARE OF FORM A*X**2+B*X+C=0**
C, N SETS OF COEFFICIENTS ARE GIVEN, 1 SET ON EACH CARD**
WRITE(6,100)**
0100, FORMAT(1H1,19#QUADRATIC EQUATIONS)**
READ(5,101)**
0101, FORMAT(14)**
DO 10 I=1,N**
READ(5,102)A,B,C**
0102, FORMAT(3F12.0)**
WRITE(6,103)I,A,B,C**
0103, FORMAT(1H0,6#NUMBER,14,10X,3E15.7)**
CALL QUAD(A,B,C,X1,X2,IND)**
GO TO(11,12),IND**
0011, WRITE(6,104)X1,X2**
0104, FORMAT(1H,6#ROOTS=E15.7,6H,
X2, 10H)
0012, WRITE(6,105)X1,X2**
0105, FORMAT(1H,6#ROOTS=E15.7,6H +,E15.7,2H I)**
0010, CONTINUE**
STOP**
END**
EQUAD HARP**
SUBROUTINE QUAD(A,B,C,X1,X2,IND)**
C, SOLVES EQUATION A*X**2+B*X+C=0**
C, IF REAL ROOTS, THEY ARE IN X1,X2, WITH IND=1**
C, IF COMPLEX ROOTS, REAL PART IS IN X1, IMAG.PT. IN X2, WITH IND=2**
D=B**2-4*A*C**
IF(D.LT.0.0)GO TO 1**
S=-B/A**
T=SQRT(D)/A**
X1=S+T**
X2=S-T**
IND=1**
RETURN**
0001, X1=-B/A**
X2=SQRT(D)/A**
IND=2**
RETURN**
END**
DATA**
TUP
0005*
#10000000000000000000000000001U*
0.2+0.3*0.4#0.2#-0.3#0.4#0.3#0.4#0.6#2.0*1.0#2.0*1.0#-1.0#1.0#1.0#
**

```

```

#1EX5
T010001100001U GEVS
#1EX5
T010001100001U GEVS
#1EX5
T010001100001U GEVS
DATA

```

第12 図 例題

第13 図 例題

第12 図 例題

第13 図 例題

