

テープ FORTRAN とカード FORTRAN の変換

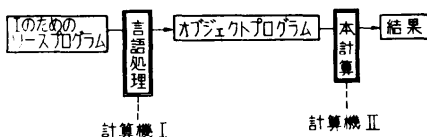
類似プログラミング言語 NARC II より HARP 5020 への翻訳*

高橋 理** 大泉 充郎** 桂 重俊***

1. まえがき

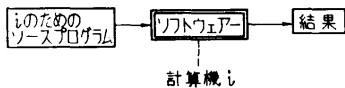
プログラムは数回の試行ののち、完成してから本計算に掛ることがほとんどであり、大形高速機を要するプログラムも中小形機で試行できれば地理的条件を考慮に入れて、利用者の側から見ても大形機の有効な利用率から見ても好都合である。もともと中小形機で開始した計算も実行途中から大形に乗り移る場合も多い実情にある。

記憶装置内のイメージを直接書く素朴なプログラム記法を除いてソースプログラムは一般に多かれ少なかれ言語処理を受けるものと見なされる。



第 1 図

多くの場合、計算機 I と計算機 II に同じ計算機が使用される（これは必要条件ではない）ので、これを



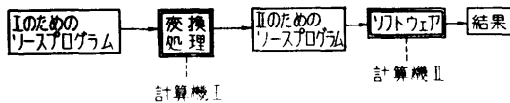
第 2 図

と表す。

$i=I$ で試算が済んで本計算を $i=II$ で実行したいとき

* Conversion of Programs Written in a Tape-oriented FORTRAN type Language into a Card-oriented FORTRAN (Translation of Programming Language NARC II into a Similar Language HARP 5020), by Tadashi Takahashi, Juro Oizumi (Research Institute of Electrical Communication, Tohoku University) and Shigetoshi Katsura (Faculty of Engineering Technology, Tohoku University)

** 東北大学電気通信研究所 *** 東北大学工学部
本研究は科学研究補助金（総合研究一代表者 後藤以紀）で行なったものである。



第 3 図

計算機 II の入力言語を中間言語として計算機 i により変換する。本論文では $i=I$ として、 $I=NEAC 2230$ 、その入力言語を NARC II、 $II=HITAC 5020$ 、その入力言語を HARP 5020 としてある。

中形機がそれ自身のために有する自動プログラミング言語が、大形機のそれと表現上近似している場合は、利用者のプログラムを多少修正し大形機向き外部表現を得ることができるとの見透しにより変換法を採ることとした。

プログラムの外部表現が近似しているといっても、計算機的な約束の点では一見些細であっても、中にはほとんど形式的に対応させられない場合も少なくない。結果的に見て、一応実用に供することができたが、予め互換性を計画的に与えられていない言語間の変換は意外に困難で、一般論としての可能性が直ちに実用性とならない点も少なくない。

2. NARC II ソースプログラムから HARP 5020 ソースプログラムへの変換について

NARC II は紙テープ入力による FORTRAN II 型に近いものであり、HARP 5020 はカード入力による FORTRAN IV 型に近いものである。一般に、FORTRAN II の文法は FORTRAN IV の文法に包含されているといわれており、お互いに共通な事項が多い。われわれは NARC II の中で HARP 5020 に包含されていない記述を中心に、変換したプログラムに HARP 5020 に対する実用上の適合性を与えることをねらった。

基本的な方針は：

(1) あらかじめ NEAC 2230 によって試計算まで済んだ NARC II ソースプログラム紙テープから記法を中心に交換して、HARP 5020 ソースプログラム紙テープ(6単位 NEAC コード)を作成する。

(2) その作成された紙テープからテープツーカードコンバータを介して HIJOB モンターランにのせられる HARP 5020 ソースプログラムのカードデッキを得る。

(3) 記法上 HARP 5020 に包含されず交換を必要とする NARC II の事項を次の三つに分類し、交換プログラムで処理する。

HARP 5020 に対し

(a) はじめから適合しているもの。これらに対しては交換をしないという処理を行なう。

(b) 計算機を用いた形式的交換で、適合できるもの。交換表にもとづき必要な置換、挿入、削除を行なう。

第1表

1-1 表記に関するもの

1-1-1) プログラムの配列の順序

NARC II	HARP 5020
(1) main program と sub program は program の最後につく	main program は program の先頭につく あるいは ENTRY card で指定する
(2) 内部記憶を越える大規模な program の処理 segmentation	chain job

1-1-2) parameter の指定が違うもの

NARC II での記述	HARP 5020 での記述
REWI ND <i>j</i> BACK SPACE <i>j</i> READ TAPE <i>j</i> , list WRITE TAPE <i>j</i> , list	使用できる MT の unit 番号は3だけである。
FORMAT (A 2, ..., 5 B, ...) A: 改頁の指定 2: この FORMAT に対する list の内容を印刷したのも2行改行する。0~9まで指定できる。 B: ブランクの指定	

1-1-3) HARP 5070 に同じ記述の statement がないもの

NARC II	statement の機能
TITLE ××××××	segment の開始を示す×××××××× はプログラム名を表す。
JOBEND	job の終りを示す。
IF (OVERFLOW) <i>n</i> ₁ , <i>n</i> ₂	overflow の検知はすべてこの命令で行なう。

(c) 機械的交換は困難であるが、人手による交換で適合できるもの。交換中プログラムで検知できるものについて警告する。

((d) 昭和41年11月から大形機に紙テープリーダーが設置され、直接紙テープの入力が可能となったので、交換したプログラムの7単位 HITACHI コードの紙テープ出力を追加した。)

1. NARC II と HARP 5020 との比較

NARC II で HARP 5020 からはみでている部分の中で、機械的なものとしては、取扱う数の範囲コードの違い、入力媒体の違いなどがあり、プログラミングに関するものとしては processor 自身の能力差、たとえばソースプログラムのエラーの摘発の差、あるいは見解の相違による object program の働きに関する違い、入力機器構成その他から生ずる文法的な記述上の違いなどが見られる。それを第1表に示す。

NARC II	Statement の機能
RESERVE list	segment 間の変数に共通領域を割り当てる。
READ PAPER TAPE <i>j</i> , list PUNCH PAPER TAPE <i>j</i> , list	紙テープに対する入出力を行なう FORMAT を使用していない (<i>j</i> =1~2; 機種番号)
PRINT <i>n</i> , list FORWARD <i>j</i>	

1-1-4) HARP 5020 にあっても使用が禁じられているもの

NARC II	HARP 5020
HALT, HALT <i>n</i> PAUSE PAUSE <i>n</i> IF (SENSE SWITCH) <i>n</i> ₁ , <i>n</i> ₂ STOP <i>n</i>	これらの statement の使用が、operating system の都合で禁止されている

1-1-5) comment と statement number

C) 123) } の)	comment statement No. } カードの欄指定で入力するため) は使用していない
---------------	--

1-1-6) NARC II にはなく追加しなければならないもの

\$ HARP
\$ DATA
READ (5, *n*), WRITE (7, *n*) の FORMAT
EXTERNAL

(注) NARC II における segmentation は次のようなものである。一つの job は 1~11 の segment から構成することができ一つの segment は内部記憶で一度に処理できる大きさのプログラム単位である。プログラマはプログラムを segment に分けておけば、その演算の制御は NARC II Processor により run time に run routine が自動的に行なう。

1-2 表記以外のもの	
1-2-1) 使用しているプログラムおよびデータの入力媒体	
NARC II	HARP 5020
紙テープ	カード
1-2-2) コードおよび活字の違い	
NEAC コード	IBM コード乃至 HITACHI コード
※ * @ ¥	* = \$
1-2-3) single precision で取り扱える整数および実数の範囲	
NARC II	HARP 5020
10進12桁の整数 10進10桁, 指数-50~49	2進31桁の整数 2進24桁, 指数-38~38
1-2-4) 文法上の誤りの指摘	
NARC II	HARP 5020
例(1) variable name の未定義, 重複定義の指摘を しない	する
(2) プログラムの流れのチェック なし	あり
1-2-5) オブジェクトプログラムの機能の違い	
例 DO 100 100 CONTINUE DO loop を1回も通らない	I=11, 12 (I1>I2) DO loop を1回通る

2. 変換方法

第1表の比較にもとづいて主要な変換事項について述べる。なお変換対照表および変換プログラムの概略を第2表と第4図に示す。

2-1 表記に関するもの

2-1-1) segment から subroutine への変換

segment の変換は性質上 chain job か HARP 5020 で使用可能な内部記憶の大きさから subroutine に変換することが考えられる。NARC II で使用できる segment の数は最大11個で、一つの segment で使用する最大の内部記憶は2,000語以内である。

また、NARC II は1語2命令が多いので1語1命令の HARP でおおよそプログラムの長さは2倍となりデータも倍長計算とすると、内部記憶が約2倍必要になる。共同利用大形機の主組織で user の利用しうる内部記憶容量を約48KWと見積り、これにおさまる segment 数 n は

$$n \times 2K \times 2 \leq 48K \quad (n \leq 12)$$

となり segments をすべて内部記憶に収容できる。

また chain job にすると subroutine の場合に比して、外部記憶装置の使用、計算機の占有時間の増加ということが生じるので subroutine に変換する。なお

segment の中の sub program と main program の配列は main program を先に置くことにした(2-3-1プログラムの配列参照)。

2-1-2) MT に関する命令については MT 1本の場合は HARP 5020 のソースプログラムで使用できる unit 番号を3とする。2本使用している場合は警告する。

2-1-3) LP 出力 FORMAT

ブランク要素の B を X にし頁切換えの FORMAT (A, ……) を FORMAT (1H1, ……) に変換する。しかし 1-1-2) で記したように頁切換え以外の改行動作が NARC II と HARP 5020 とで異なっている。

NARC II では、list の内容を印刷したあとで改行し、HARP 5020 ではその逆に list の内容を印刷する前に改行する。それでも list と format が対応して最後まで実行される場合には FORMAT (2, ……) という NARC II での改行指定の数(この場合は2)だけ / を format の一番外側の右カッコの直前に挿入し、最初の左カッコ直前の改行指定は 1H+ とする。すなわち FORMAT (1H+, ……, /) とすることですむ。

しかしこの場合に困ることは、format に対応している list がつきて format が途中で打ち切られたときに改行が実行されなくなる。特に不定回数の繰り返えしになっている list のときには実行されないという事態がしばしば生ずる恐れがある。これを避けるために format の改行の部分だけ分離してそれだけ行なう write statement を追加する。たとえば

```
PRINT 10, A……
```

```
10) FORMAT (2, ……( ))
```

に対し

```
WRITE (6,10) A……
```

```
10) FORMAT (1H+, ……( ))
```

```
WRITE (6,10000)
```

```
10000) FORMAT (1H+, /)
```

とする。これだと印刷後の改行が可能になるが、しかし write 命令がふえて印刷に要する時間が約倍くらいに増加すると思われる。しかし現在は最初の方法で 1H+ の代わりに 1Hb として、重ね打ちの生じることをさけるようにしている。この / の挿入で 72 欄以上に延びたときには continuation 記号を付し 1 line 分を挿入する。

2-1-4) TITLE ABCD → \$ ABCD HARP 64

Segment を subroutine に対応づけることからその

2-1-6) IF(OVERFLOW) n_1, n_2

$$\rightarrow \left\{ \begin{array}{l} \text{IF ACCUMULATOR} \\ \text{OVERFLOW } n_1, n_3 \\ n_3 \text{ IF QUOTIENT} \\ \text{OVERFLOW } n_1, n_2 \end{array} \right.$$

NARC II で OVERFLOW JUMP の命令は、これ一つなので、二つの命令の組み合わせとする。

2-1-7) RESERVE list

→ COMMON list

変数を segment 間で共通に使用するためにはそれを segment の RESERVE で宣言すればよい。

また COMMON で宣言されていない変数でも、RESERVE で宣言されていれば、その segment 内の subprogram 間でも共通に使用することができる。そのため RESERVE を COMMON で置きかえ、その list をつける。さらにその segment 内の subprogram のすべてに同一の COMMON list をつける (2-3-1 のプログラムの配列参照)。

2-1-8) READ PAPER TAPE j, list

$$\rightarrow \left\{ \begin{array}{l} \text{READ (5, n)} \\ n \text{ FORMAT (}\dots\dots\text{)} \end{array} \right.$$

NARC II では、データを紙テープから FORMAT を用いずに入力する。データに対しては欄指定がなく 1 個のデータの終りを、で区切り、そこまで読んで 1 個の数としての処理を完了する。そのため変換に際して次のような問題を考慮する必要が生じる。

(a) データについて

- 1) カード 1 枚にデータをいくつ含ませるか。
- 2) データのフィールドの長さをいくらにするか。
- 3) READ (5, n) の list の中で最初の変数に対応するデータを見出し、それを新規のカードに対して割り当てる。

(b) FORMAT の作成

list とカード 1 枚に編集されているデータの状況に合致した FORMAT を作成しなければならない。そのためには

- 1) list の変数の型に応じて real, integer の FORMAT の数要素を選択しなければならない。
- 2) 個々のデータの長さに合ったフィールド指定をその数要素へ与えなければならない。
- 3) list の中で繰り返し欄があればそれを満たす FORMAT を作らなければならない。

これらを共に満足させるためには変換するプログラムを NARC II の下で実行させ、データが読み込ま

れる度にその list とデータを参照して、それにあう FORMAT を作らなければならない。しかし、それでもなお

```
READ PAPER TAPE 1, K, A, ((B(I, J), L(I, J),
I=1, K), J=1, K), M, C
```

に対する HARP 5020 の FORMAT を作成することは list を変更しないでは作れない。NARC II の場合には FORMAT の使用がなくプログラマが繰り返えしの回数 K を知っていることから、紙テープ上でのデータの配列にだけ気をつければ上のような命令も使用できる。このため次のような処理を行なう。

(c) データに対して

- 1) データはカード 1 枚に一つとする。

(d) FORMAT に対して

- 1) 数要素は次の二つとする。

E 16.10, I 12

- 2) list の変数の type が切り換わるたびに以降を list とする。read statement を追加する。

- 3) list の繰り返えしがあるときには DO loop に組む。その loop の中に繰り返えしを示す左カッコから、終端の右カッコまでを list とする新しい read statement を置く。

この処理によれば FORMAT は list を参照するだけで作成できデータはデータのみで変換処理できる。たとえば

```
READ PAPER TAPE 1, K, N, ((B(I, J), L(I,
J), I=1, K), J=1, K), M, C
```

は次のように変換される。

```
READ (5,10100) K, N
DO 12001 J=1, K
DO 12000 I=1, K
READ (5,10101) B(I, I)
READ (5,10100) L(I, J)
12000 CONTINUE
12001 CONTINUE
READ (5, 10100) M
READ (5, 10101) C
10100 FORMAT (I 12)
10101 FORMAT (E 16.10)
```

2-1-9)

PUNCH PAPER TAPE j, list

$$\rightarrow \left\{ \begin{array}{l} \text{WRITE (7, n) list} \\ n \text{ FORMAT (}\dots\dots\text{)} \end{array} \right.$$

HARP 5020 に変換したプログラムには list に合った FORMAT が必要になる。それは READ PAPER TAPE の FORMAT を作成する場合と類似な問題になる。また、出力したカードは次に入力されることが予想されるので入出力の FORMAT が共通である

方が都合よい。

以上のことから READ PAPER TAPE の FORMAT を作成する場合 2-1-8 と同じ取扱いをし、命令を WRITE (7, n) に変える。

2-1-10)

```

    HALT
    HALT n
    PAUSE
    PAUSE n
    IF(SENSE SWITCH) n1, n2
  }
  → {
    C HALT
    C HALTn
    C PAUSE
    C PAUSEn
    C IF(SENSE SWITCH) n1, n2
  
```

以上の命令の使用が、大形機の運用上禁止されているので、プログラムから省く。その際 NARC II のプログラムイメージを残す意味で comment 扱いとする。

2-1-11)

1234) → 1234 を 2 col. から 5 col. までに割り当て)を除く。

NARC II ではカードの 1~5 col. に相当する部分では欄指定を用いておらず、そのため)で statement No. の終端を示している。変換に当たっては、statement No. (NARC II では4桁まで)を 5 col. 目につめて上位桁を 2 col. までの間に割り当てる。この変換プログラムで附加した statement No. には 32767 以内の5桁の数を用いる。

2-1-12) \$ DATA

NARC II ではデータ群の頭には何も命令はつけず直接データから始まっている。変換プログラムでは変換するプログラムの開始が数字の場合にはデータであると考え自動的に \$ DATA を設けたのち、カード1枚に対しデータ1個を割り当てる。

2-1-13) EXTERNAL

NARC II では subprogram name を call statement のパラメータとして以前に何ら宣言することなく書くことができる。HARP 5020 ではその場合、External 命令の list で宣言することが必要である。変換に際しては call statement のパラメータに subprogram name があるかどうか検査することを避けて、便宜的にどの subprogram の中にも external 命令を挿入し、そのリストで job 内の subprogram

name をすべて宣言する。

2-2 表記以外のもの

2-2-1) 紙テープからカードへの変換およびコード変換については、東北大学の NEAC 2230 計算機システムはカード機器がついていないので、オフラインのテープとカードコンバータを用いて入力媒体およびコード変換を行なった。これは6単位 NEAC コードの紙テープを読み込んでそのコードに対応するコードを80欄カードへせん孔し、それをパッチボードプログラムにより制御できる機能を持ったものである。変換せん孔速度は毎秒 5~6 枚程度である。

現在の大型機に紙テープリーダがつき紙テープから直接入力することが可能となったので、NEAC-2230 の紙テープ出力機器を若干改造し、偶数パリティ7単位の紙テープの作成を可能とした。そのため変換プログラムには新たに NEAC コードから HITACHI コードへのコンバージョンプログラムを増設した。

2-2-2) 数の取扱い範囲

数の桁数および指数部の大きさは single precision の場合、整数および実数共に NARC II が HARP 5020 に比して大きい。そのため HARP 5020 では、double precision で計算を行なうことにし、出力は NARC II の10桁までとする。有効桁数の増長による精度は良くなると思われるが、指数部の大きさは、double precision にしても single precision の場合にくらべて変わらないので、NARC II で生じなかった overflow の起こる危険性は依然として残る。

第2表

2-3 変換対照一覧表

2-3-1) 置換および挿入するもの

a) プログラムの配列

NARC II	変換した HARP 5020
例 (1) main program と subprogram の配列	
TITLE ABC	\$ ABC HARP 64
SUBROUTINE A	C MAIN
⋮	⋮
END	END
FUNCTION B(X)	\$ B HARP 64
⋮	⋮
END	FUNCTION B(X)
C) MAIN	END
END	\$ A HARP 64
JOBEND 2	SUBROUTINE A
	END
例 (2) segment の配列から subroutine の配列へ	
TITLE A	\$ B HARP 64
RESERVE V1, V2 (10)	COMMON V1, V2 (10)
SUBROUTINE S1	CALLA
⋮	⋮
END	END

CALL S1	\$ A HARP 64
END	SUBROUTINE A
TITLE B	COMMON V1, V2 (10)
RESERVE V1, V2 (10)	CALL S1
	END
CALL A	\$ S1 HARP 64
	SUBROUTINE S1
END	COMMON V1, V2 (10)
JOBEND 2	END

b)

REWIND j	} j=12 or 13	REWIND 3
BACK SPACE j		BACK SPACE 3
READ TAPE j, list		READ TAPE 3, list
WRITE TAPE j, list		WRITE TAPE 3, list
FORMAT (A 2, ..., 5 B, ...)		FORMAT (1H1, ..., 5 X, .../)
TITLE xxxxxx		\$ xxxxxx HARP 64
IF (OVERFLOW) n1, n2		IF ACCUMULATOR OVERFLOW n1, n2
		n1 IF QUOTIENT OVERFLOW n1, n2
RESERVE list		COMMON list
		一つの segment 内のすべての subroutine, function に, COMMON list を挿入する。 (参照 3-3-1 プログラムの配列)
READ PAPER TAPE j, list		READ (5, n) list
		n FORMAT (.....)
PUNCH PAPER TAPE j, list		WRITE (7, n) list
		n FORMAT (.....)
PRINT n, list		WRITE (6, n) list
FORWARD j (j=12 or 13)		READ TAPE 3
HALT		C HALT
HALT n		C HALT n
PAUSE		C PAUSE
PAUSE n		C PAUSE n
IF (SENSE SWITCH) n1, n2		C IF (SENSE SWITCH) n1, n2
		(comment とする)
FUNCTION		\$ HARP 64
		FUNCTION
SUBROUTINE		\$ HARP 64
		SUBROUTINE

c) 組み込み関数サブプログラムの名前

EXPF	DEXP
ALOGF	DLOG
ATANF	DATAN
SINF	DSIN
COSF	DCOS
SQRTF	DSQRT

2-3-2) 削除するもの

NARC II	変換した HARP 5020
xxxxx)	xxxxx 1 1 col. 2 col. 5
STOP n	STOP
JOBEND	

3. あとがき

以上詳述したように、われわれは NARC II より、HARP 5020 への翻訳プログラムを作成し、一応所期の目的を達成し、NARC II の紙テープより HARP 5020 の紙テープへの変換約 50 件、NARC II のテープより HARP 5020 のカードへの変換約 20 件の実績を有し、仙台・東京間 1~2 回の往復で半数は実用になっている。1回で通らないことは翻訳プログラムがまだ debug の必要があることによる場合と本質的な機械の差による場合(入出力の問題)があるが、前者に対しては今後の虫取りにより完成を期したい。始めから HARP 5020 だけを相手にした program の場合には、syntax checker で十分かつその方が有利であるかもしれないが、本方式は試験計算の一部を小形地方機で行ってから長時間の実計算を大形機に切換えようという特徴を持つものである。

なお、この変換プログラムの実施に当たっては、東北大学計算センターの猪苗代勉、小畑征二郎、武田敏夫の諸君に負うところが多大であり、ここに厚く感謝する。

参考文献

- 1) NEAC-2230 NARC II 説明書 (東北大学計算センター第3版, 昭和40年, 第4版, 昭和41年)
- 2) NEAC-2230 NARC-II 使用法説明書(改第2版) 日本電気株式会社(昭和40年)
- 3) 高橋, 大泉: カード FORTRAN と紙テープ FORTRAN の諸問題点, 情報処理学会全国大会講演予稿集, 昭和39年
- 4) HITAC 5020 FORTRAN (HARP) プログラムマニュアル(40年7月版, 40年11月版) 株式会社 日立製作所
- 5) 森口: FORTRAN IV 入門 (HARP 5020 に即して)(1965年 東京大学出版会)
- 6) 小畑, 大泉, 高橋, 桂: NARC II 言語プログラムから HARP 5020 言語プログラムの変換, 情報処理学会全国大会講演予稿集, 昭和41年
- 7) 大泉, 高橋, 猪苗代, 小畑, 武田: NARC-II から FORTRAN-IV へ変換する PROGRAM, NEAC-Journal No. 15, 昭和41年
- 8) 日本工業規格原案 電子計算機プログラム用言語 FORTRAN 3000, 5000, 7000 日本電子工業振興協会(昭和41年)

(昭和42年4月18日受付)