

## 単純型流れ図の構造とその分類

著者	富樫 昭
雑誌名	鹿児島大学理学部紀要. 数学・物理学・化学
巻	7
ページ	3-10
別言語のタイトル	ON STRUCTURE AND CLASSIFICATION OF FLOW CHARTS OF SIMPLE TYPE
URL	<a href="http://hdl.handle.net/10232/00001751">http://hdl.handle.net/10232/00001751</a>

# 単純型流れ図の構造とその分類

富 樫 昭

## ON STRUCTURE AND CLASSIFICATION OF FLOW CHARTS OF SIMPLE TYPE

By

Akira TOGASI

(1974年9月30日受理)

### Abstract

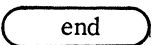
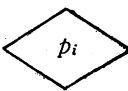
In this paper we shall develop a theory of flow charts of simple type, a basic scheme for more general flow charts. After studying the relationship between flow chart of Scott type and incomplete finite automaton, we define a flow chart of simple type by using the terminology of the underlying finite automaton, and clarify the characteristics of the flow chart. It is shown that the set of flow charts having a predicate element and a decision element is equal to a set of flow chart of simple type.

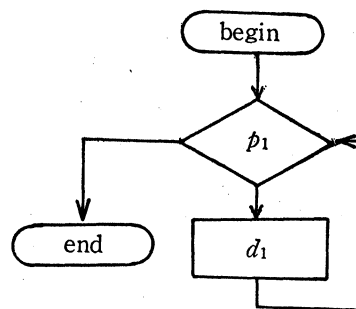
### 1. はじめに

流れ図 (flow chart) はコンピュータのプログラミング上非常に重要な役割を果すもので、ほとんどのプログラムは、あらかじめ作られた流れ図に従って作成される。

本論文では、この流れ図をモデル化し、そのうちで最も基本的なタイプについてその構造の解明と分類をおこなうことを試みる。

一般的には流れ図は次の要素間を方向をもった線で結んだものである。

要素名	表現	意味
B 要素		流れ図の始まり
E 要素		流れ図の終り
$p$ 要素		判定
$d$ 要素		処理
A 線		流れ線 (要素間の連結: 流れの方向が明らかなきときは矢印は省略)



第1図

**定義 1.** 流れ図とは各々1個のB要素, E要素とそれぞれ有限個(空を含む)の  $p$  要素および  $d$  要素の間を, 次の規則にしたがって A 線で結んでできる有向グラフをいう。

Akira TOGASI (Department of Mathematics, Kagoshima University.)

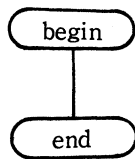
- 1 A線は必ず要素から出て( (要素) → ; outAとよぶ )要素に入る( ← (要素) ; inAとよぶ ),
- 2 B要素は1つ, かつただ1つの out A をもつ,
- 3 E要素は空でない有限個の in A をもつが out A はない,
- 4 p要素は空でない有限個の in A と条件によって行先の異なる2個の out A をもつ,
- 5 d要素は空でない有限個の in A と1個の out A をもつ,
- 6 どの out A もその行先が自分自身となることはない,
- 7 B要素から E要素に到達する path が少なくとも一つ必ず存在する,  
ここで path とは A線をたどって各要素を通過する道のことである,
- 8 どの p, d 要素も B要素からの path の途中にある,  
たとえば第1図は流れ図の簡単な例である,

定義 2.  $n$  個の  $p$  要素,  $m$  個の  $d$  要素からなる流れ図全体の集合を  $F_{n,m}$  で表わす.  
 $F$  をすべての流れ図の集合とすれば,

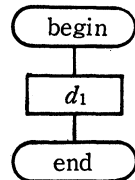
$$F = \bigcup_{n,m=0}^{\infty} F_{n,m}$$

である。

例 1  $F_{0,0}$  (したがって単一要素集合である)

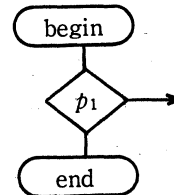


例 2  $F_{0,1}$



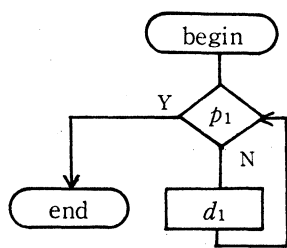
例 3  $F_{1,0}$

$\therefore F_{1,0} = \phi$

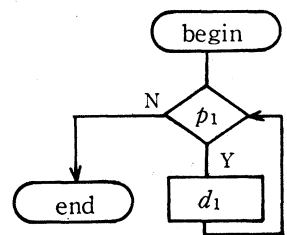


例 4  $F_{1,1}$  の全体

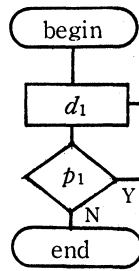
Y: YES, N: NO



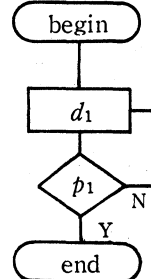
(1)



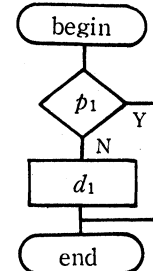
(2)



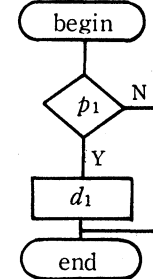
(3)



(4)



(5)



(6)

2 流れ図と有限オートマトンの関連

流れ図の  $p$  要素を  $p_i$  : predicate symbol,

$d$  要素を  $d_j$  : decision symbol

とし,  $F_{n,m}$  の要素を

$$f(p_1, p_2, \dots, p_n; d_1, d_2, \dots, d_m)$$

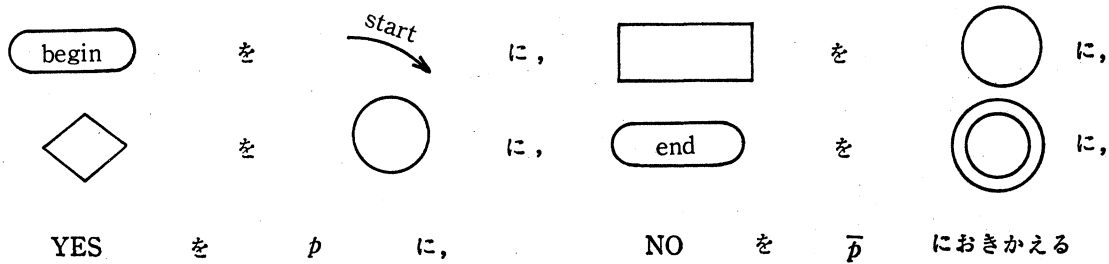
で表わす。すると流れ図  $f$  の流れは  $\{p_i, d_j\}$  ( $i = 1, 2, \dots, n; j = 1, 2, \dots, m$ ) の relation algebra として有限オートマトンで表現できる。

たとえば例 4 の (2) の流れ図は Scott の表現を用いると

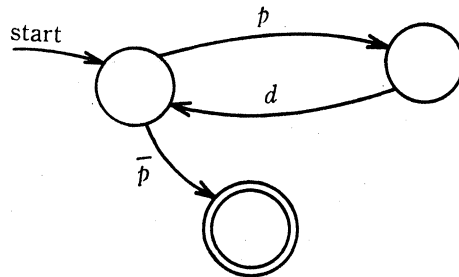
$$(p_1; d_1)^* ; \bar{p}_1$$

で表わせる。ここで  $*$  は Kleene の演算,  $;$  は  $;$  の前の演算のあとに次の演算をおこなう記号である。

ここで流れ図の各表現をつぎのようにおきかえる。



すると第 2 図例 4 の (2) の流れ図は



第 3 図

と表わせる。そして明らかに  $\forall f \in F_{1,1}$  に対して一意的に対応する有限オートマトンの表現ができる。

いま  $\Sigma = \{p, \bar{p}, d\}$  とすると

$$T(f) \equiv \{\bar{p}, pd\bar{p}, pdpd\bar{p}, \dots, (pd)^n \bar{p}, \dots\}$$

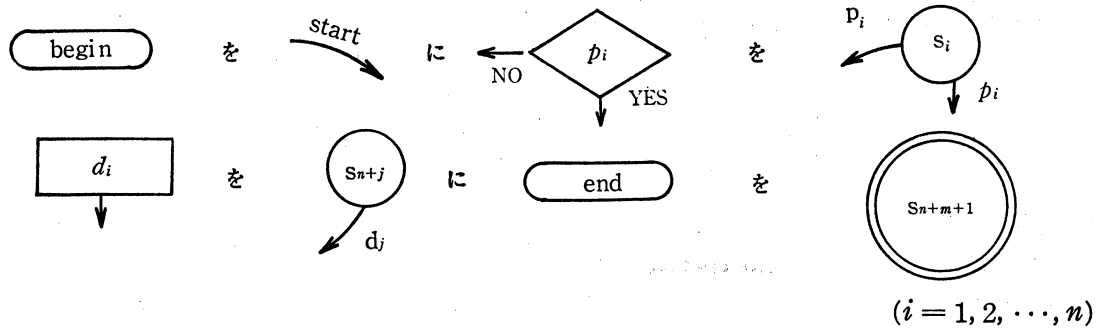
は  $(p; d)^* ; \bar{p}$

に対応する。

これは流れ図が不完全有限オートマトンの正則言語で表現できることを示している。

流れ図に対し 1 意的に定まる有限オートマトンを  $f$  と書き, これを base automaton とよぶ, すなわち,

定義 3 Base automaton  $f$  とは



におきかえてできる有限オートマトンをいう。

**定理 1**  $\forall f \in F$  に対する Scott の特性表現は  $p_i, d_j, \bar{p}_i$  を記号とみた場合正則表現である。

証明  $\chi(f)$  を  $f$  に対する Scott の特性表現とする。 $f$  の base automaton  $f = (S, M, s_0, D)$  を考える。ただし  $\Sigma = \{p_1, p_2, \dots, p_n, \bar{p}_1, \bar{p}_2, \dots, \bar{p}_n, d_1, \dots, d_n\}$ 。

すると  $T(f) \equiv \{w \mid w \in \Sigma \text{ かつ } M^*(s_0, w) \in D\}$

は正則である ([2])。

$\hat{T}(f) \equiv \{\bar{w} \mid w \in T(f) \text{ で, } w \text{ 中の } p_i \text{ を } \bar{p}_i \text{ に, } \bar{p}_i \text{ を } \bar{p}_j \text{ に, } d_j \text{ を } d_j \text{ に置きかえてできる語全体}\}$

(たとえば  $p d \bar{p} \in T(f)$  は  $\bar{p}; d; \bar{p}; \in \hat{T}(f)$ ) とすると

$\chi(f)\{;\} = \hat{T}(f)$  である。

ところが  $T(f)$  は正則, したがって  $\hat{T}(f)$  は

$\hat{\Sigma} = \{p_1, \dots, p_n, \bar{p}_1, \dots, \bar{p}_n, d_1, \dots, d_n\} = \Sigma \cup \{;\}$

の上で正則である。

故に  $\chi(f)$  は正則表現である (証明終)。

一般に  $F \equiv \bigcup_{n,m=0}^{\infty} F_{n,m}$  の構造は複雑であるので, 一気に  $\forall f \in F$  について構造, 分類をおこなうことは困難である。

そこで  $f$  のうち, 次のような単純型とよばれる型のものについて考察をすすめることにする。

**定義 4**  $f \in F_{n,m}$  が単純型 (simple type) であるとは,  $f$  に対応する base automaton

$f = (S, M, s_0, D)$  において

$s \in S \quad w \in \Sigma^* \quad (\Sigma = \{p_1, \dots, p_n, \bar{p}_1, \dots, \bar{p}_n, d_1, \dots, d_n\})$

が存在して

(i)  $M^*(s_0, w) = s,$

(ii)  $\forall \bar{s} \in S \quad (D \neq \bar{s})$  に対して (1) の  $w$  の head (これを  $h(w)$  とする) が一意に定まって,  $M^*(s_0, h(w)) = \bar{s}$

であることである。

いいかえれば  $f$  で  $D$  の状態を除くすべての状態をただ一度通過するような長さ  $n+m$  の語が存在するとき  $f$  を単純型とよぶ。

$F_{n,m}$  の要素で単純型のものの全体を  $S_{n,m}$ , 単純型流れ図全体の集合を  $S$  とすると, あきらかに

$S = \bigcup_{n,m=0}^{\infty} S_{n,m}$  である。

また次のことが成り立つ。

定理 2  $\forall l (= 0, 1, 2, \dots)$  に対し、

$$F_{0,l} = S_{0,l}$$

$$F_{l,0} = S_{l,0}$$

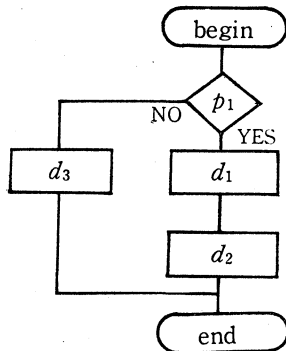
例  $F_{0,0} = S_{0,0}, F_{1,0} = S_{1,0} = \phi, F_{0,1} = S_{0,1}$

また  $F_{1,1} = S_{1,1}$

しかし任意の  $n, m$  に対しては必ずしも

$$F_{n,m} = S_{n,m} \text{ とはならない。}$$

例  $f = p_1; d_1; d_2 \cup \bar{p}_1; d_3$  は



$f \in F_{1,3}$  であるが  
 $f \notin S_{1,3}$  である (第4図)。

第4図

$F_{n,m}$  は  $S_{n,m}$  の要素の合成でできていると考えることができるので、その意味で  $S_{n,m}$  の解明は基本的に重要なものである。

### 3 単純型流れ図の基本型

定義4は流れ図そのものについて言えば、B要素からはじめてすべての  $p, d$  要素を1回ずつ通過する path が存在するような型のものをさしている。したがって path をたどって現れる各要素を縦に並べると第5図のような  $p, d$  の列ができる。このような  $p, d$  要素の列を単純型流れ図の基本型とよぶことにする。

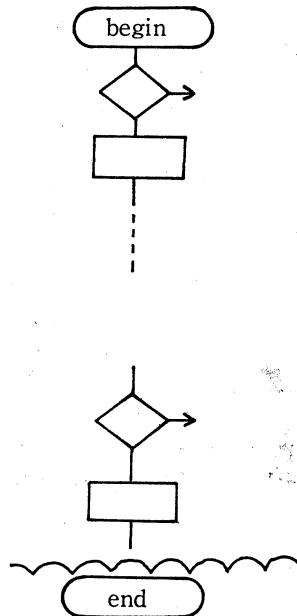
$S_{n,m}$  の基本型の全体を  $B_{n,m}$  であらわし、 $S$  のすべての基本型の集合を  $B$  とすれば明らかに

$$B \equiv \bigcup_{n,m=0} B_{n,m} \text{ である。}$$

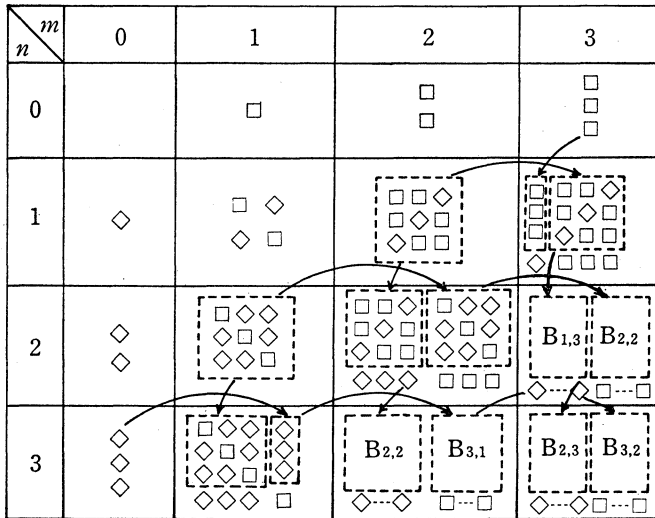
いま  $B_{n,m}$  のパターンを  $n, m = 1, 2, 3$  について図示したのが第6図である。

この図からもわかるように  $B_{n,m}$  はその一つ前のパターン  $B_{n-1,m}$  と  $B_{n,m-1}$  のそれぞれの下に前者に  $p$  要素( $\diamond$ ), 後者に  $d$  要素( $\square$ )を付加したものとなっている。

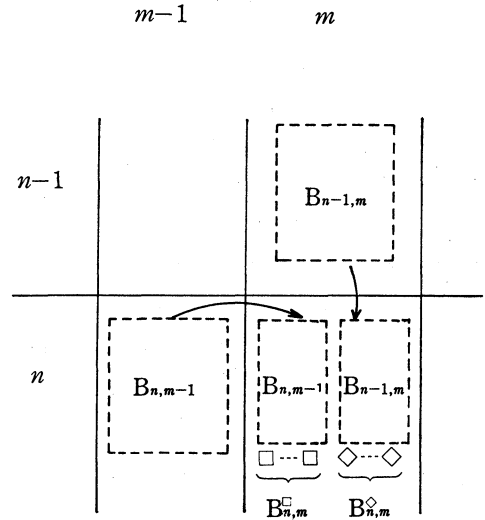
$B_{n-1,m}$  の下に  $p$  要素( $\diamond$ ) 1個をつけたパターン全体を  $B_{n,m}^\diamond$ ,  $B_{n,m-1}$  の下に  $d$  要素( $\square$ )を1個つけた



第5図



第6図



第7図

パターン全体を  $B_{n,m}^\square$  で表わすと

$$B_{n,m} = B_{n,m}^\diamond \cup B_{n,m}^\square \tag{1}$$

である (第7図参照)。

また,  $B_{n,m}$  の要素の数  $\#(B_{n,m})$  についてしらべると  $\#(B_{1,1}) = 2$ ,  $\#(B_{2,1}) = 3$ ,  $\#(B_{2,2}) = 6$  で, 一般には  $p$  要素  $n$  個,  $d$  要素  $m$  個の重複を許す順列であるから

$$\#(B_{n,m}) = \frac{(n+m)!}{n!m!}$$

となる。また,

$$\begin{aligned} \#(B_{n,m}^\diamond) &= \#(B_{n-1,m}), \\ \#(B_{n,m}^\square) &= \#(B_{n,m-1}). \end{aligned}$$

したがって, これと (1) の関係より次の定理が得られる。

**定理 3**  $\#(B_{n,m}) = \#(B_{n-1,m}) + \#(B_{n,m-1})$ ,  $n, m = 1, 2, \dots$  ただし,  $\#(B_{0,l}) = \#(B_{l,0}) = 1$  とする

証明 上の図の関係からも明らかであるが, 各々の順列を計算してもよい。

$$\begin{aligned} &\#(B_{n-1,m}) + \#(B_{n,m-1}) \\ &= \frac{(n+m-1)!}{(n-1)!n!} + \frac{(n+m-1)!}{n!(n-1)!} = \frac{(n+1)!}{n!m!} = \#(B_{n,m}) \end{aligned}$$

#### 4 単純型流れ図 $S_{n,m}$ について

$B_{n,m}$  において縦に通る1本の path の他に, 各  $p$  要素のもう一つの out A の行先や (end) につながる path をつけてできるすべての有向グラフの集合が  $S_{n,m}$  である。

**定理 4**  $n, m = 1, 2, \dots$  のとき

$$\#(S_{n,m}) = 2^{n-1} \frac{(n+m-1)!}{(n-1)!(m-1)!} \left[ \frac{2(n+m)}{nm} (n+m-1)^n \right]$$

$$+ \frac{2(n+m-1)}{n} \sum_{i=1}^n \binom{n}{i} (n+m-2)^{n-i}$$

$$+ \frac{n+m-1}{m} \sum_{i=1}^{n-1} \binom{n-1}{i} (n+m-2)^{n-i}]$$

証明 単純型流れ図の全体を次の2通りの場合に分類して考える。

(A)  $B_{n,m}$  のパターンの最後の要素が直接 (end) につながるような path がある場合 (第8図)。

この場合の流れ図の総数はパターン数が  $\#(B_{n,m}) = \frac{(n+m)!}{n!m!}$  個、 $p$  要素の他の1つの outA の行先としてとり得る場合の数が各々  $(n+m-1)$  個、 $p$  要素の2つの outA を入れかえる (yes と no の交換) ことにより各2通り増加するので、あわせて

$$\frac{(n+m)!}{n!m!} 2^n (n+m-1)^n \tag{2}$$

通りである。

(B)  $B_{n,m}$  のパターンの最後の要素が直接 (end) につながっていない場合 (第9, 10図)。この場合は更に次の2通りの場合に細別される。

(i)  $B_{n,m-1}^{\square}$  のパターンの場合 (第9図)。

この場合は  $\#(B_{n,m-1}^{\square}) = \frac{(n+m-1)!}{n!(m-1)!}$  である。

最後の  $d$  要素からの outA の行先の場合の数が  $n+m-1$  通り、 $n$  個の  $p$  要素のうち  $i$  個のもう一つの outA が (end) へ直接到達している場合、他の  $(n-i)$  個の各  $p$  要素の他の一つの outA の行先の場合の数は、自分自身、すぐ下の要素、(end) の3個を除いて各  $(n+m-2)$  通り、それに  $p$  要素の2つの outA の入れかえで各2通りで、これを合すると

$$\frac{(n+m-1)!}{n!(m-1)!} 2^n (n+m-1) \sum_{i=1}^n \binom{n}{i} (n+m-2)^{n-i} \tag{3}$$

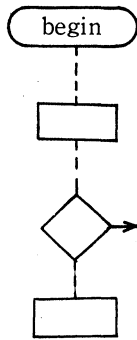
通りである。

(ii)  $B_{n-1,m}^{\diamond}$  のパターンの場合 (第10図)。

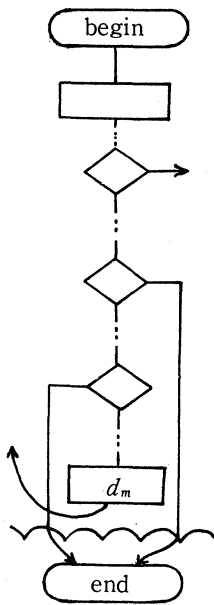
この場合は  $\#(B_{n-1,m}^{\diamond}) = \frac{(n+m-1)!}{(n-1)!m!}$  である。

また、最後の  $p$  要素  $p_n$  の2つの outA の行先としてとり得る場合の数が  $(n+m-1) \times (n+m-2)$  通り、

他の  $n-1$  個の  $p$  要素のうち  $i$  個のもう一つの outA が直接 (end) へ到達したときの残りの  $(n-1-i)$  個の  $p$  要素の各々のもう一つの outA の行先としてとり得る場合の数が  $(n+m-2)$  通り、それに  $p$  要素の2つの outA の入れかえが2通りである。これをあわせて

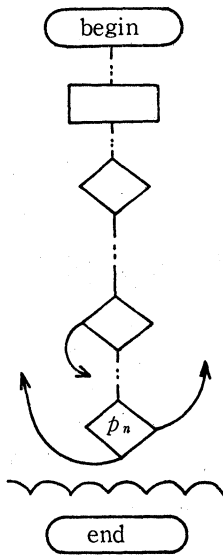


第8図



第9図





第10図

$$\frac{(n+m-1)!}{(n-1)!m!} 2^{n-1}(n+m-1)(n+m-2) \sum_{i=1}^{n-1} \binom{n-1}{i} (n+m-2)^{n-1-i} \quad (4)$$

通りである。

(2), (3), (4) をあわせて

$$\begin{aligned} \#(S_{n,m}) &= \frac{(n+m)!}{n!m!} 2^n(n+m-1)^n \\ &+ \frac{(n+m-1)!}{n!(m-1)!} 2^n(n+m-1) \sum_{i=1}^n \binom{n}{i} (n+m-2)^{n-i} \\ &+ \frac{(n+m-1)!}{(n-1)!m!} 2^{n-1}(n+m-1)(n+m-2) \sum_{i=1}^{n-1} \binom{n-1}{i} \times \\ &\quad (n+m-2)^{n-1-i} \end{aligned}$$

である。これを整頓すれば定理の結果の式がでる。

例  $\#(S_{1,1}) = 6$ ,  $\#(S_{1,2}) = 20$ ,  $\#(S_{2,1}) = 80$

ただし、この数は単に場合の数だけから求めてあるので、実際のプログラミングには意味のないものも含まれていることに注意しなければならない。

参 考 文 献

- [1] J.W. de Bakker and L.G.L. Th. Meertens: Simple recursive program schemes and inductive assertions, MR 142/72 Dec Tech Rep. 41 pp.
- [2] M.O. Rabin and D. Scott: Finite automata and their decision problems, IBM J. Res. Develop., 3 (1959), 115-125.