論　文

Decision Grid Chart のあいまいさとループの表現について*

守屋慎次**　平松啓二**

Abstract

Ambiguity in decision grid chart is pointed out and methods to solve the ambiguity are presented. It is shown that program loops are naturally expressed in decision grid chart, and several properties of program loops in desion grid chart are investigated.

1. はじめに

複雑な判定状況を記述する際，goto もしくは perform 文で連結されたいくつかのデシジョンテーブルが有効な場合が多い。その際，複雑な判定状況をより小さな単位に分析する過程に，しばしば困難さが伴う[1]．Strunz[1] はそのような分析を遂行するための有用な手段として，Decision Grid Chart（以後 DGC と略記）を紹介している．**Fig. 1** に DGC の例を示す．例えば，a_2 は，c_2 が N。であれば c_1, c_3 には無関係に実行される．DGC 上のアクションは左から右の順に実行されるが，条件の判定には順序性はない．

文献 1) では，書かれた DGC を，何個かのデシジョンテーブルへ人手によって変換する方法が例示されている．この考え方が示唆するものは有用であるが，変換の客観的な基準が述べられていないために，変換後の意味が異なる場合も生じ得る．

本論文ではまず，DGC が表わす論理に「あいまいさ」があることを指摘する．次いで，このあいまいさは DGC が持って生まれたものであり，何らかの約束を設定しない限り，DGC を書く者の意図が読む者（物）に，一般には正しく伝達されないことを示す．そして，1) の考え方に添った，簡潔で適切と思われる二つの約束を提案する．この約束により，DGC のあいまいさは解消される．さらに本論文では，DGC が簡潔な形でループを表わす能力を有することを示す．また，DGC によるループの表現が正しく行われるための必要十分条件など，この能力を利用するのに必要不可欠な性質を導いている．ループの表現という，DGCの新しい能力は，DGC の適用範囲を少なからず拡大するものであると思われる．

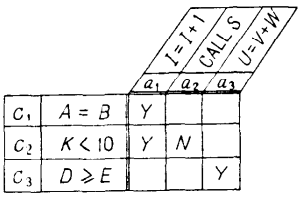

Fig. 1 A decision grid chart

2. DGC のあいまいさとその解消案

2.1 DGC が表わす論理のあいまいさ

DGC をデシジョンテーブルへ人手によって変換する Strunz[1] の考え方を，**Fig. 2** の例に基づいて簡単に述べる．まず，Fig. 2 上に Y, N を枠で囲んで示すようにアクションをグループ分けする．次いで1つのグループを1つのデシジョンテーブルに，というや

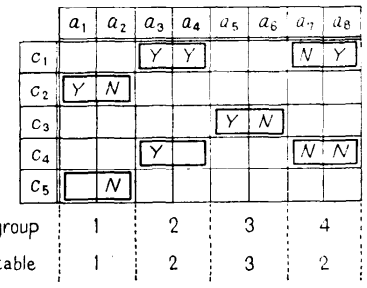

Fig. 2 Decision grid chart for illustrative example

* Ambiguity in Decision Grid Chart and Expression of Program Loops by Shinji MORIYA and Keiji HIRAMATSU (Department of Electrical Communication Engr. Tokyo Electrical Engineering College)

** 東京電機大学電気通信工学科

T_1				
c_2	Y	Y	N	N
c_5	Y	N	Y	N
a_1	X	X		
a_2				X
gots T_2	X	X	X	X

T_2				
c_1	Y	Y	N	N
c_4	Y	N	Y	N
a_3	X			
a_4	X	X		
perform T_3	X	X	X	X
a_7				X
a_8		X		

T_3		
c_3	Y	N
a_5	X	
a_6		X
return	X	X

Fig. 3 Decision tables converted from the decision grid chart shown in Fig. 2

り方で変換し，各テーブルを適切に連結する．ただし，グループ2とグループ4のように同一の条件群を要素とするグループは1つのテーブルにまとめる．その際，アクションの順序を保つために，グループ3はグループ2内で呼出される closed 型テーブルになる．**Fig. 3** が変換の結果である．このように1つの DGC が複数個のデシジョンテーブルに変換されるから，DGC のすべての条件の判定が終了する以前に，いくつかのアクションが実行されることになる．従って，Strunz の手順には，次に述べる2つの仮定のうち少なくもいずれか一方が満たされていなければならない．

〔仮定 1〕 DGC をデシジョンテーブルへ変換する者(物)は，その DGC が表わす意味を熟知している．

〔仮定 2〕 DGC の実行直前までには，すべての条件の真理値が決定しており，DGC 内のいかなるアクションの実行によっても，それらの真理値が影響を受けることはない．

以上である．しかし，機械的に変換を行う場合，また，DGC を communication tool として用いる場合，仮定1は成立しない．さらに，一般の計算機プログラムにおいて，仮定2が成立するのはむしろ特殊の場合と考える方がより自然であろう．一方，DGC の使用範囲を仮定2が成立する範囲に限定する方法も考えられる．しかし，仮定2の成立と不成立のいずれも，DGC が本来有する自然の性質であることを考慮すれば，上の制限は不自然で DGC の可能性を限定することにもなりかねない．

以上の理由から，DGC の機械的な解釈は一般的には無理であるといえる．つまり，DGC が表わす論理にはあいまいさがある．しかし，DGC の優れた特長を広く活用するためには，DGC の意味があいまいのままであってはならない．そこで，あいまいさを解消する方法について次に述べる．

2.2 DGC に対する新しい約束

Fig. 2 は3つのデシジョンテーブルに変換された．これを，$T_1=[\{c_2, c_5\}, \{a_1, a_2\}]$, $T_2=[\{c_1, c_4\}, \{a_3, a_4,$ perfrom $T_3, a_7, a_8\}]$, $T_3=[\{c_3\}, \{a_5, a_6\}]$ と書き，(Fig. 2 の) サブ DGC と呼ぶことにする．与えられた DGC から，このようなサブ DGC 群が一意に決定できれば，DGC が表わす論理にはもはやあいまいさはない[2]．サブ DGC は条件とアクションの2つの部分からなるから，あいまいさの解消案も2段構えになる．なお，以後の議論は制限エントリ DGC に限って行われる．

〔**提案 C 1**〕 条件のグループを明示するための記号 "—" をエントリ内に書入れる．"—" は，b (ブランク) と同様 Yes または No (つまり don't care) と解釈される．

Fig. 2 に — 記号を書入れた例を **Fig. 4** に示す．

DGC に書かれる条件の集合を，$C=\{c_1, \cdots, c_n\}$, アクションの集合を $A=\{a_1, \cdots, a_m\}$ と表わす．$c_i \in C$, $a_j \in A$ の添数 i, j はそれぞれ，条件スタブの行番号及びアクションの列番号に対応するものとする．また c_i 行 a_j 列に書かれた記号 (Y, N, b, — のいずれか1つ)を e_{ij} と表わし，$C(a_j)=\{c_i \in C \mid e_{ij} \neq b\}$ と定める．

〔**提案 C 2**〕 DGC を分解して得られるそれぞれのサブ DGC の条件スタブは，$\{C(a_j) \mid a_j \in A, j=1, \cdots,$

	a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8
c_1			Y	Y	—		N	Y
c_2	Y	N						
c_3			—		Y	N		
c_4			Y		—		N	N
c_5	—	N						
group	1		2	3	4	5	6	
table	1		2	3	2	5	6	

Fig. 4 Decision grid chart entered grouping information (mark "—") in the entry

$n\}$ の各元によって構成される．

Fig. 4 の場合，$\{\{c_2, c_5\}, \{c_1, c_3, c_4\}, \{c_1\}, \{c_3\}, \{c_1, c_4\}\}$ の各元が分解されたサブ DGC の条件部となる．

〔**提案 A 1**〕 $C(a_{k-1}) \neq C(a_k) = C(a_{k+1}) = \cdots = C(a_{k+l}) \neq C(a_{k+l+1})$ のとき，$A_p = \{a_k, a_{k+1}, \cdots, a_{k+l}\}$ と置く．この方法で，$A = \{a_1, \cdots, a_m\}$ を $A_1, \cdots, A_p, \cdots, A_q$ に直和分割する．

Fig. 4 の場合，$A_1 = \{a_1, a_2\}$, $A_2 = \{a_3\}$, $A_3 = \{a_4\}$, $A_4 = \{a_5\}$, $A_5 = \{a_6\}$, $A_6 = \{a_7, a_8\}$. A_p は Strunz[1] におけるグループに相当する．そこで $S = A_1, A_2, \cdots, A_q$ をグループ列と呼ぶ．

さて，任意の $a \in A_i$, $a' \in A_j$ に対し，$C(a) = C(a') \neq \phi$ のとき，$A_i \equiv A_j$ と表わし，$C(a) \neq C(a')$ または $C(a) = C(a') = \phi$ のとき，$A_i \not\equiv A_j$ と表わすことにする．明らかに $A_i \not\equiv A_{i+1}$ である．Fig. 4 の場合 $A_2 \equiv A_4$ で，この A_2 と A_4 の間に置かれた A_3 をアクションとするサブ DGC の T_3' が，closed 型になっている．しかし，別のグループ列 $S' = A_1A_2A_3A_4$ で，$A_1 \equiv A_3$, $A_2 \equiv A_4$ の場合，closed 型 DGC になるのが $T_2 = [C(a), A_2]$ (ただし $a \in A_2$) か $T_3 = [C(a'), A_3]$ (ただし $a' \in A_3$) か，依然としてあいまいがさ残る．つまり，$\equiv$ で結ばれる A_i, A_j ($i \neq j$) の対が入れ子にならない場合にあいまいである．しかし，このあいまいさを解消するための確固たる理由を有する方法が見当らない．そこで，一応，次の基本方針を定めておく．即ち，$A_i \equiv A_j$, $A_k \equiv A_l$, $i < k < j < l$ のように互に入れ子にならない場合，添字の順序の上で後に現われた方を closed 型と考える．しかし，$S'' = A_1A_2A_3A_4A_5A_6$ で，$A_1 \equiv A_3$, $A_2 \equiv A_5$, $A_4 \equiv A_6$ のような場合，この基本方針だけでは包みきれない状況を含んでいる．そこで，DGC をサブ DGC に分解する次のアルゴリズムを提案する．

アルゴリズムは DGC で表わされる．例えば Fig. 1 の DGC は，実際には **Fig. 5** (a) の横型[1]，または，Fig. 5 (b) のような縦型に書けばよい．なお，$T_i = [C(a), A_j]$ (ただし $a \in A_j$) なる記法は $T_i = [A_j]$ と書いても情報は失われない．

〔**提案 A 2**〕 分解アルゴリズム． **Fig. 6** 参照．〔例〕 $S = A_1A_2A_3A_4A_5A_6A_7A_8A_9$ で，

a_1	X		
a_2		X	
a_3			X
c_1	Y	—	—
c_2	Y	N	—
c_3	—	—	Y

(a) holizontal type

c_1	X		
c_2		X	
c_3			X
a_1	Y	Y	—
a_2	—	N	—
a_3	—	—	Y

(b) vertical type

Fig. 5 Two types for writing decision grid chars

$A_1 \equiv A_8$, $A_2 \equiv A_4$, $A_3 \equiv A_6$, $A_5 \equiv A_7$, 他は $\not\equiv$ とする．

a_1, a_2:

$i=4$ $S \Rightarrow A_1^{(1)}A_2^{(2)}A_5^{(1)}A_6^{(1)}A_7^{(1)}A_8^{(1)}A_9^{(1)}$ $(S_1^{(2)} = A_2^{(1)}B_3^{(1)}A_4^{(1)})$

$i=7$ $\Rightarrow A_1^{(1)}A_2^{(2)}A_5^{(2)}A_8^{(1)}A_9^{(1)}$ $(S_5^{(2)} = A_5^{(1)}B_6^{(1)}A_7^{(1)})$

$i=8$ $\Rightarrow A_1^{(2)}A_9^{(1)}$ $(S_1^{(2)} = A_1^{(1)}B_2^{(2)}B_5^{(2)}A_8^{(1)})$

$a_{3.1}$: $T_1 = [A_1^{(2)}]$, $T_2[A_9^{(1)}]$; $S' = T_1T_9$

$a_{3.2}$: $T_3 = [A_3^{(1)}]$, $T_6 = [A_6^{(1)}]$, $T_2 = [S_2^{(2)}]$, T_5

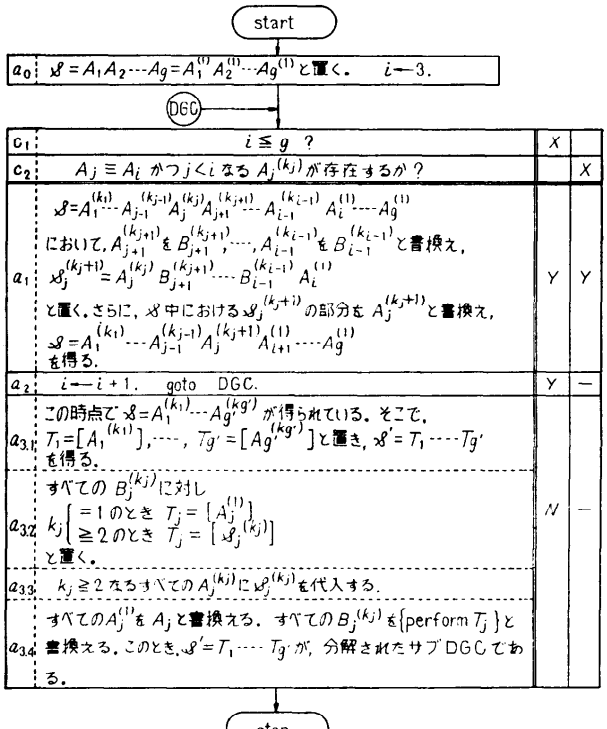

Fig. 6 Decomposition algorithm of decision grid chars

$=[S_5^{(2)}]$

$a_{3,3}$: $T_1=[S_1^{(2)}]$

$a_{3,4}$: $T_1=[A_1$ {perform T_2} {perform T_5} $A_8]$, $T_2=[A_2$ {perform T_3} $A_4]$, $T_3=[A_3]$, $T_5=[A_5$ {perform T_6} $A_7]$, $T_6=[A_6]$, $T_9=[A_9]$; $S'=T_1$ T_9.

3. DGC とループ

Strunz[1] は DGC によるループの表現には何らふれていない．しかし，デシジョンテーブルの場合もそうであったように[3] DGC は，ループを簡潔に表現する能力を備えている．

Fig. 7 の流れ図において，a_2 から①及び②への矢印をそれぞれ GO TO 1 及び GO TO 2 に，a_7 から③への矢印を GO TO 3 に書換えて，同じ論理を表現したものが **Fig. 8** である．ただし，c_1, c_2, c_3 及び a_5 にはそれぞれ，ラベル 1, 2, 3 及び 5 が付与されている．例えばラベル 2 に制御が移った場合，c_2 が No ならば a_5, a_6 を実行し，Yes ならば c_3 を判断する．c_3 が Yes ならば a_1, a_2, GOTO 1 を実行し，No ならば a_2, GOTO 2 を実行する．つまり，ラベル 2 に制御が移った後，判断の可能性がある条件の集合は $\{c_2, c_3\}$ $(=R_C(c_2))$，実行の可能性あるアクションの集合は $\{a_1, a_2,$ GOTO 1, GOTO 2, $a_5, a_6\}$ $(=R_A(c_2))$ である．Fig. 7 の S_2 に囲まれた条件とアクションが，この 2 つの集合に対応している．Fig. 8 では，S_2' に囲まれたエントリの各行と各列に対応

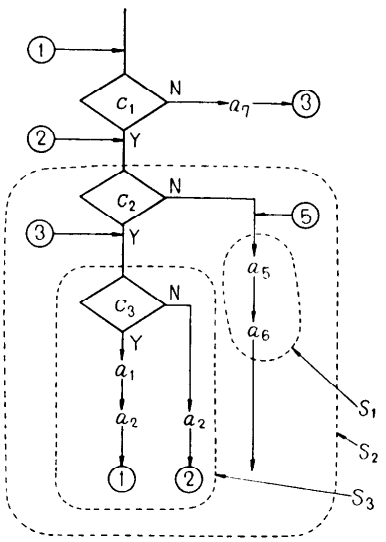

Fig. 7 Flowchart for illustrative example

					5			
		a_1	a_2	GOTO 1	GOTO 2	a_5	a_6	a_7	GOTO 3
1	c_1	Y	Y	Y	Y	Y	Y	N	N
2	c_2	Y	Y	Y	Y	N	N	—	—
3	c_3	Y	—	Y	N	—	—	—	—

S_3'　S_2'　S_1'

Fig. 8 Decision grid chart with statement labels, 1, 2, 3 and 5, expressing the same algorithm as that of Fig. 7

した条件とアクションが，この 2 つの集合に対応している．このように，Fig. 7 上の S_1, S_2, S_3 は Fig. 8 上の S_1', S_2', S_3' に対応している．以上の考察から，Fig. 7 と Fig. 8 が同一のアルゴリズムを表現していることが確認できる．このように通常のプログラム同様，ラベル付けと GOTO 文によって，DGC が有するループの表現能力を利用可能である．しかし，この能力を活用するためには，Fig. 8 の S_1', S_2', S_3' が示すようなある範囲を知る必要がある．つまり，ラベルに付けられた文に制御が移行した後，判断し実行される可能性のある文の集合が，一意に確定できなければならない．

本章では，ラベル付けによってこのような集合が一意に確定できるのに必要十分な条件を求め，判断と実行の可能性ある文の集合を知る方法を導く．これらの結果は，DGC のループの表現能力を利用するために必要不可欠な事項である．なお，以後の議論は，制限エントリのサブ DGC に関して進められる．特にことわりのない限り，DGC といえばサブ DGC を指すものとする．

3.1 条件へのラベル付け

3.1.1 概念の形式化

DGC 上の条件の集合を改めて $C=\{c_1, \cdots, c_n\}$，アクションの集合を $A=\{a_1, \cdots, a_m\}$ と表わす．c_i 及び a_j の添数 i 及び j はそれぞれ，DGC 上の行番号及び列番号を意味する．i 行 j 列のエントリに書かれた記号 (Y, N, — のいずれか 1 つ) を e_{ij} と書く．

〔**定義 1**〕 DGC 上のラベル付けられた条件 c_i (またはアクション a_j) へ制御が移行した場合，移行した時点から制御が DGC の外に出るまで，あるいは，移行した時点から再びその DGC 内の文へ制御が移るまでに，判断される可能性のある条件の集合を $R_C(c_i)$ (または，$R_A(a_j)$)，また，実行される可能性のあるア

Table 1 ∗-product

∗	Y	N	—	φ
Y	Y	φ	Y	φ
N	φ	N	N	φ
—	Y	N	—	φ
φ	φ	φ	φ	φ

∗--- product

クションの集合を $R_A(c_i)$ (または $R_A(a_j)$) と表わす. なお, $R_C(a_j)=\phi$ と考える. ϕ は空集合を表わす.

〔**定義 2**〕 $e_j{}^p=(e_{1j}, e_{2j}, \cdots, e_{pj})$, $1\leqq p\leqq n$.

〔**定義 3**〕 $W=\{Y, N, —, \varphi\}$, $V=\{Y, N, —\}$ と置き, W 及び V それぞれの p 個 ($1\leqq p\leqq n$) の直積を $W^p=W\times\cdots\times W$, $V^p=V\times\cdots\times V$ と表わす.

〔**定義 4**〕 $x, y\in W$ とするとき2項関係 $\leqq$ を, $x*y=x\Leftrightarrow x\leqq y$ と定める. ここに, 2項演算∗ (∗積と呼ぶ) は **Table 1** で与えられる.

例えば, $Y\leqq Y$, $Y\leqq —$, $\varphi\leqq Y$, $Y\not\leqq N$ である.

〔**定義 5**〕 $\boldsymbol{x}, \boldsymbol{y}\in W^p$ とする. $\boldsymbol{x}=(x_1, \cdots, x_p)$, $\boldsymbol{y}=(y_1, \cdots, y_p)$ と表わすとき, $\boldsymbol{x}*\boldsymbol{y}=\boldsymbol{x}\Leftrightarrow\boldsymbol{x}\leqq\boldsymbol{y}$. ここに $\boldsymbol{x}*\boldsymbol{y}=(x_1*y_1, \cdots, x_p*y_p)$.

$\boldsymbol{x}=(Y, N)$, $\boldsymbol{y}=(Y, —)$ のとき, $\boldsymbol{x}*\boldsymbol{y}=(Y, N)=\boldsymbol{x}$. $\boldsymbol{y}'=(N, —)$ のとき, $\boldsymbol{x}*\boldsymbol{y}'=(\varphi, N)\notin V^p$ で, $\boldsymbol{x}\neq\boldsymbol{x}*\boldsymbol{y}'\neq\boldsymbol{y}'$. ∗積は, 巾等律, 交換律, 結合律, が成立する.

[定義 6] $\boldsymbol{x}\in V^p$ とする. $Q(\boldsymbol{x})=Q(x_1, \cdots, x_p)=\{a_j\in A\,|\,\boldsymbol{x}\leqq\boldsymbol{e}_j{}^p\}$.

Fig. 8 の場合, $Q(—)=\phi$, $Q(Y)=\{a_1, a_2,$ GOTO 1, GOTO 2, $a_5, a_6\}$ ($=R_A(c_2)$ が後に示される. 以下同様), $Q(N)=\{a_7,$ GOTO 3$\}$, $Q(Y, Y)=\{a_1, a_2,$ GOTO 1, GOTO 2$\}$ ($=R_A(c_3)$), $Q(Y, N)=\{a_5, a_6\}$ ($=R_A(a_5)$).

〔**定義 7**〕 $A(c_{p+1})=\{a_j\,|\,e_{p+1,j}\neq —,\ j=1, \cdots, m\}$, $0\leqq p\leqq n-1$.

Fig. 8 の場合, $A(c_2)=\{a_1, a_2,$ GOTO 1, GOTO 2, $a_5, a_6\}$ ($\subseteq Q(Y)$). $A(c_3)=\{a_1,$ GOTO 1, GOTO 2$\}$ ($\subseteq Q(Y, Y)$).

$A(c_{p+1})=\phi$ のとき, c_{p+1} はアクションの実行に何ら影響を与えない. つまり, c_{p+1} の有無は DGC が表わす論理に無関係である. 従って, 以後の議論では $A(c_{p+1})\neq\phi$ と仮定する.

$e_{p+1,j}$ が Y (または N) であれば, c_{p+1} の判定が

		a_1	a_2	a_3	a_4
	c_1	Y	Y	—	N
2	c_2	Y	—	—	—
3	c_3	—	Y	N	—

Fig. 9 A decision grid chart

Yes (または No) のときに a_j を実行せよ, という論理を表わしている. つまり, そのような a_j は, c_{p+1} に制御が移行した後に, 実行される可能性がある. 従って, $a_j\in A(c_{p+1})$ ならば $a_j\in R_A(c_{p+1})$ である. これを「規則」として述べておこう.

〔**規則 1**〕 $A(c_{p+1})\subseteq R_A(c_{p+1})$. $0\leqq p\leqq n-1$.

DGC にラベル付けるということは, その DGC が表現するアルゴリズムを構成する文の1つにラベル付けることである, と考えるのが自然であろう. 従って, DGC の条件にラベル付けた場合, 条件の判定には順序性が存在する. この順序性は, 次のように定めるのが実際上適切である.

〔**規則 2**〕 DGC 上の条件から成る集合 $C'=\{c_p, c_{p+1}, \cdots, c_{p+i}, \cdots, c_{p+q}\}$ の元 c_{p+i} にラベル付けられているとする. このとき, C' の判断は, $\{c_p, \cdots,$

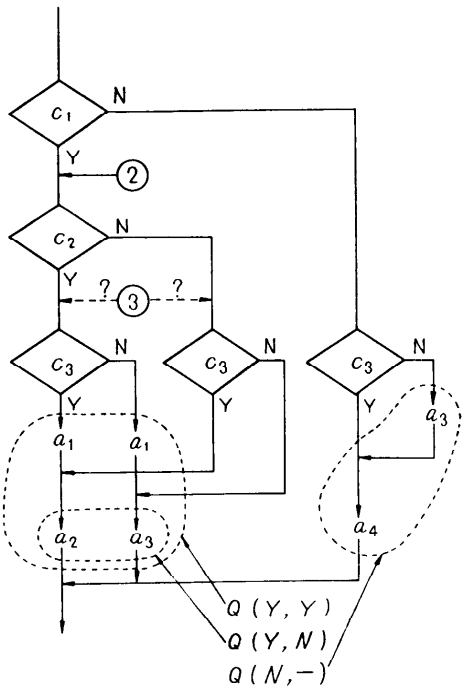

Fig. 10 Flowchart expressing the same algorithm as that of Fig. 9

$c_{p+i-1}\}$ 次いで $\{c_{p+i}, \cdots, c_{p+q}\}$ の順に行わねばならない．ラベル付けられた元が存在しないとき，判断の順序は任意である $(1\leqq p\leqq i\leqq q\leqq n)$.

ここで **Fig. 9** (前頁参照) について考える．ラベル 2, 3 がないものとして，それと等価な論理を流れ図で表わせば **Fig. 10** (前頁参照) が得られる．Fig. 9 上のラベル 3 は c_3 に付けられている．ところが Fig. 10 には c_3 が 3 つある最も左側の c_3 にラベル 3 が付くと解釈すれば，$R_A'(c_3)=\{a_1, a_2, a_3\}$ $(=Q(Y, Y))$，真中と解釈すれば $R_A''(c_3)=\{a_2, a_3\}$ $(=Q(Y, N))$，右側なら $R_A'''(c_3)=\{a_3, a_4\}$ $(=Q(N, —))$ である．しかし，$R_A'''(c_3)\not\ni A(c_3)=\{a_2, a_3\}$ だから，右側の c_3 には付かない(規則 1)．残りの $R_A'(c_3)$ と $R_A''(c_3)$ はいずれも $A(c_3)$ を含む．しかし，$R_A'(c_3)\not\ni R''_A(c_3)$ である．つまり，Fig. 9 のラベル 3 に制御を移そうとしても，その後の流れを一意に定めることができない．従って，Fig. 9 の c_3 へのラベル付けは論理上意味を持たないといえる．DGC にラベル付けるということは，その DGC が表現するアルゴリズムを構成する文の 1 つにラベル付けること，という考え方を基礎にして，上記のことを形式的に表現するために，まず次の定義をする．

〔**定義 8**〕 $\Omega_{c^p}=\{Q(\boldsymbol{x})\,|\,Q(\boldsymbol{x})\supseteq A(c_{p+1}), \boldsymbol{x}\in V^p\}$, $1\leqq p\leqq n-1$.

$\boldsymbol{x}=(x_1, x_2, \cdots, x_p)$ とすれば，c_1 の判定が x_1, c_2 の判定が $x_2, \cdots, c_p$ の判定が x_p の場合に，実行の可能性を残すアクションの集合が $Q(\boldsymbol{x})$ である．Ω_{c^p} は，そのような $Q(\boldsymbol{x})$ のうち，$A(c_{p+1})$ を含むものの集合である．従って，先に述べたことは，次のように言い換えられる．即ち，ラベル付けられた条件 c_{p+1} に制御を移した場合，その後の流れが一意に定まる (これを「ラベル付けが一意である」と略称する) ためには $|\Omega_{c^p}|=1$* でなければならない．また，$|\Omega_{c^p}|=|\{Q(\boldsymbol{x})\}|=1$ であれば，そのような $Q(\boldsymbol{x})$ を $R_A(c_{p+1})$ と考えることにより，その後の流れは一意に定まる $(Q(\boldsymbol{x}_1)=Q(\boldsymbol{x}_2), \boldsymbol{x}_1\not\ni\boldsymbol{x}_2$ なる $\boldsymbol{x}_1, \boldsymbol{x}_2$ が存在したときは，そのいずれを $R_A(c_{p+1})$ と考えてもよい)．そこで，次のように定める．

〔**定義 9**〕 $|\Omega_{c^p}|=1$ であるとき，またそのときのみ，$c_{p+1}\in C$ $(1\leqq p\leqq n-1)$ へのラベル付けは一意である，という．

3.1.2 ラベル付けが一意であるための必要十分条件

〔**定義 10**〕 (i) $\boldsymbol{k}_j{}^p$ は，$a_j\in A(c_{p+1})$ なる $\boldsymbol{e}_j{}^p$ を表わす：(ii) $\boldsymbol{k}_*{}^p=\boldsymbol{k}_1*\boldsymbol{k}_2*\cdots*\boldsymbol{k}_q$，ただし，$\{\boldsymbol{k}_j{}^p\}=\{\boldsymbol{k}_1, \boldsymbol{k}_2, \cdots\boldsymbol{k}_q\}$，なお，$1\leqq p\leqq n$.

Fig. 9 の場合，$\boldsymbol{k}_1{}^1=(Y)$, $\boldsymbol{k}_*{}^1=(Y)$; $\boldsymbol{k}_2{}^2=(Y, —)$, $\boldsymbol{k}_3{}^2=(—, —)$, $\boldsymbol{k}_*{}^2=(Y, —)$.

〔**定義 11**〕 $\boldsymbol{x}, \boldsymbol{y}\in W^p$ とする．このとき，(i) $\boldsymbol{x}<\boldsymbol{y}\Leftrightarrow\boldsymbol{x}\leqq\boldsymbol{y}$ かつ $\boldsymbol{x}\not\ni\boldsymbol{y}$; (ii) $\boldsymbol{x}$ と $\boldsymbol{y}$ は第 1 種比較不可能$\Leftrightarrow\boldsymbol{x}\not\ni\boldsymbol{x}*\boldsymbol{y}\not\ni\boldsymbol{y}$ かつ $\boldsymbol{x}*\boldsymbol{y}\in V^p$; (iii) $\boldsymbol{x}$ と $\boldsymbol{y}$ は第 2 種比較不可能 $\Leftrightarrow\boldsymbol{x}\not\ni\boldsymbol{x}*\boldsymbol{y}\not\ni\boldsymbol{y}$ かつ$\boldsymbol{x}*\boldsymbol{y}\notin V^p$; なお，$x, y\in W$ についても同様な記法を用いる．

例えば，$(Y, —)$ と $(—, N)$ とは第 1 種，$(Y, —)$ と (N, N) とは第 2 種比較不可能である．

〔**定理 1**〕 $c_{p+1}\in C$ $(1\leqq p\leqq n-1)$ へのラベル付けが一意であるための必要十分条件は次の (i)(ii) である．(i) $\boldsymbol{k}_*{}^p\in V^p$; (ii) $\boldsymbol{e}_j{}^p<\boldsymbol{k}_*{}^p$ なる $\boldsymbol{e}_j{}^p$ または，$\boldsymbol{k}_*{}^p$ と第 1 種比較不可能な $\boldsymbol{e}_j{}^p$ は存在しない．

〔**例**〕 Fig. 9 の場合，$\boldsymbol{k}_*{}^2=(Y, —)\in V^p$. $\boldsymbol{e}_1{}^2=(Y, Y)<\boldsymbol{k}_*{}^2$ で (ii) を満足しないから，c_3 へのラベル付けは一意でない．c_2 へのラベル付けは一意，Fig. 8 の場合，c_1, c_2, c_3 へのラベル付けはすべて一意．

(**証明**) 十分性．定義 7 の注釈より $A(c_{p+1})\not\ni\phi$ だから $\boldsymbol{k}_*{}^p$ が存在する $(1\leqq p\leqq n-1)$. すべての $\boldsymbol{k}_j{}^p$ $(a_j\in A(c_{p+1}))$ に対して $\boldsymbol{k}_*{}^p\leqq\boldsymbol{k}_j{}^p$ である (定義 10) から，$Q(\boldsymbol{k}_*{}^p)\supseteq A(c_{p+1})$. 以上と (i) の $\boldsymbol{k}_*{}^p\in V^p$ とから $Q(\boldsymbol{k}_*{}^p)\in\Omega_{c^p}$. 従って，$\Omega_{c^p}=\{Q(\boldsymbol{k}_*{}^p)\}$ であることを示せば十分である (定義 9)． そこで $Q(\boldsymbol{x})\supseteq A(c_{p+1})$ なる $\boldsymbol{x}\in V^p$ が存在したと仮定して，場合を，①$\boldsymbol{k}_*{}^p=\boldsymbol{x}$, ②$\boldsymbol{k}_*{}^p<\boldsymbol{x}$, ③$\boldsymbol{k}_*{}^p>\boldsymbol{x}$, ④$\boldsymbol{k}_*{}^p$ と $\boldsymbol{x}$ とは第 1 種比較不可能, ⑤$\boldsymbol{k}_*{}^p$ と $\boldsymbol{x}$ とは第 2 種比較不可能，の 5 つに分けて考える．$\boldsymbol{x}\in V^p$ がこれらのいずれか 1 つに落ちることは明らかであろう． ①の場合. $Q(\boldsymbol{k}_*{}^p)=Q(\boldsymbol{x})$ となってこのような $\boldsymbol{x}$ の存在は Ω_{c^p} には無関係. ②の場合. $\boldsymbol{k}_*{}^p=(k_1{}^*, \cdots, k_p{}^*)$, $\boldsymbol{x}=(x_1, \cdots, x_p)$, $\boldsymbol{k}_j{}^p=(k_{j1}, \cdots, k_{jp})$ と置く．(i) の $\boldsymbol{k}_*{}^p\in V^p$ より $k_i{}^*$ は $Y, N, —$ のいずれか 1 つであるから，$k_i{}^*=k_{ji}$ なる $j(a_j\in A(c_{p+1}))$ が少なくも 1 つは存在する．一方，$\boldsymbol{k}_*{}^p<\boldsymbol{x}$ だから，$k_i{}^*<x_i$ なる i $(1\leqq i\leqq p)$ が少なくとも 1 つある．従って上記の j に対し $\boldsymbol{k}_j{}^p\not\ni\boldsymbol{x}$ だから $a_j\notin Q(\boldsymbol{x})$. これは $Q(\boldsymbol{x})\supseteq A(c_{p+1})\ni a_j$ なる仮定に矛盾する． 従って，仮定が正しいとき②の場合はあり得ない. ③の場合. $\boldsymbol{k}_*{}^p>\boldsymbol{x}$ より $Q(\boldsymbol{k}_*{}^p)\subseteq Q(\boldsymbol{x})$. $Q(\boldsymbol{k}_*{}^p)=Q(\boldsymbol{x})$ の場合については①の場合と同様．そこで，$a_j\in Q(\boldsymbol{x})-Q(\boldsymbol{k}_*{}^p)$ なる $\boldsymbol{e}_j{}^p$ が存在したとする．$a_j\notin Q(\boldsymbol{k}_*{}^p)$ より $\boldsymbol{k}_*{}^p\not\ni\boldsymbol{e}_j{}^p$. 従って，

---

* 有限集合 S の元の数を $|S|$ と表わす．

$\boldsymbol{k}_*^p$ と $\boldsymbol{e}_j^p$ の関係は，（イ）$\boldsymbol{k}_*^p > \boldsymbol{e}_j^p$，（ロ）$\boldsymbol{k}_*^p$ と $\boldsymbol{e}_j^p$ は第1種比較不可能，（ハ）$\boldsymbol{k}_*^p$ と $\boldsymbol{e}_j^p$ は第2種比較不可能，のいずれかである．条件（ii）から（イ）（ロ）ではない．一方，$Q(\boldsymbol{x}) \supseteq A(c_{p+1}) \cup \{a_j\}$ だから，$\boldsymbol{x} \leqq \boldsymbol{k}_*^p$ かつ $\boldsymbol{x} \leqq \boldsymbol{e}_j^p$，即ち $\boldsymbol{x} \leqq \boldsymbol{k}_*^p * \boldsymbol{e}_j^p$ である．$\boldsymbol{x} \in V^p$ だから $\boldsymbol{k}_*^p * \boldsymbol{e}_i^p \in V^p$．従って，このとき（ハ）の状態はあり得ない．つまり，$Q(\boldsymbol{x}) \supseteq A(c_{p+1})$ なる仮定と（ii）が成立するとき，③の場合はあり得ない．④の場合．②の場合の記法を用いる．第1種比較不可能の条件と $\boldsymbol{k}_*^p \in V^p$，$\boldsymbol{x} \in V^p$ とから，「$k_i^* = —$ かつ $x_i \neq —$」かつ「$k_i^* \neq —$ かつ $x_i = —$」なる s，t $(s \neq t)$ が存在する．一方，$\boldsymbol{k}_*^p \in V^p$ であるから $k_i^* = k_{jt}$ なる $\boldsymbol{k}_j^p$ が存在する．またこのとき $x_t = —$ であるから $\boldsymbol{x} \nleqq \boldsymbol{k}_j^p$．従って $a_j \notin Q(\boldsymbol{x})$ となって $Q(\boldsymbol{x}) \supseteq A(c_{p+1})$ の仮定に反する．つまり，この仮定の下に④の場合はあり得ない．⑤の場合．第2種比較不可能の条件から（イ）$k_i^* = Y$ かつ $x_i = N$，（ロ）$k_i^* = N$ かつ $x_i = Y$，のいずれか一方の成立する i が存在する．任意の $a_j \in A(c_{p+1})$ に対して $a_j \in Q(\boldsymbol{k}_*^p) \cap Q(\boldsymbol{x})$ と仮定しているから，すべての $\boldsymbol{k}_j^p$ $(a_j \in A(c_{p+1}))$ に対し，$\boldsymbol{k}_*^p \leqq \boldsymbol{k}_j^p$ かつ $\boldsymbol{x} \leqq \boldsymbol{k}_j^p$．この2つの関係式が成立するためには，（イ）または（ロ）の場合の i におけるすべての j $(a_j \in A(c_{p+1}))$ に対し，$k_{ji} = —$ でなければならない．また，そのようなすべての j に対する k_{ji} の $*$ 積が k_i^* に等しいから，$k_i^* = —$ となって（イ）（ロ）と矛盾する．従って⑤の場合もあり得ない．

以上のように，（i）（ii）が成立して，かつ，$Q(\boldsymbol{x}) \supseteq A(c_{p+1})$ なる $\boldsymbol{x} \in V^p$ が存在すれば，$Q(\boldsymbol{x}) = Q(\boldsymbol{k}_*^p)$ である．従って $|\Omega_{c^p}| = |\{Q(\boldsymbol{k}_*^p)\}| = 1$.

必要性．一意性より $Q(\boldsymbol{x}) \supseteq A(c_{p+1})$ なる $\boldsymbol{x} \in V^p$ が存在する．$A(c_{p+1}) \neq \phi$ なる仮定から $Q(\boldsymbol{x}) \neq \phi$ である．十分性の証明より，このような $\boldsymbol{x}$ は①と③の場合に限り存在する．つまり $\boldsymbol{x} \leqq \boldsymbol{k}_*^p$．従って，$\boldsymbol{x} \in V^p$ だから（i）の $\boldsymbol{k}_*^p \in V^p$ が成立．十分性の証明における③の $\boldsymbol{x} < \boldsymbol{k}_*^p$ の場合，$a_j \in Q(\boldsymbol{x}) - Q(\boldsymbol{k}_*^p)$ としたとき，$\boldsymbol{k}_*^p$ と $\boldsymbol{e}_j^p$ との関係は（イ）（ロ）（ハ）の3種類であった．また，この場合（ハ）の状態はあり得ない．そして，（イ）または（ロ）ならば上記の a_j が存在して $Q(\boldsymbol{x}) \neq Q(\boldsymbol{k}^*)$．この対偶と，一意性より（ii）が成立する．（証終）．

定理1は c_1 へのラベル付けについては述べていない，c_1 へ制御が移るということは，その DGC 全体を実行することに等しい．つまり，c_1 へのラベル付けは常に一意，と考えるのが自然である．

〔規則 3〕 $c_1 \in C$ へのラベル付けは一意である．

3.1.3 実行すべきアクション，判断すべき条件

〔定理 2〕 $c_{p+1} \in C$ $(1 \leqq p \leqq n-1)$ へのラベル付けが一意であるとき，$R_A(c_{p+1}) = Q(\boldsymbol{k}_*^p)$.

〔証明〕 定理1の証明より明らか．（証終）．

Fig. 9 の場合，$R_A(c_2) = Q(\boldsymbol{k}_*^1) = Q(Y) = \{a_1, a_2, a_3\}$．定義6の例参照．

定理2は $R_A(c_1)$ を含んでいない．規則3と同じ理由により，次のように考えるのが自然である．

〔規則 4〕 $R_A(c_1) = A$

次に，c_{p+1} へ制御が移行した後に判断の可能性ある条件の集合 $R_C(c_{p+1})$ について述べる．$R_C(c_{p+1})$ は C の部分集合であるが，規則2や今までの議論から，$R_C(c_{p+1}) \subseteq \{c_{p+1}, c_{p+2}, \cdots, c_n\} = D$ と考えるのが妥当である．D の元の中には，すべての $a_j \in R_A(c_{p+1})$ にわたって $e_{ij} = —$ なる $c_i \in D$ 存在し得る．このような c_i は，c_{p+1} へ制御が移行した後の流れに無関係である．従って，

〔規則 5〕 $R_C(c_{p+1}) = \{c_i \in D \mid e_{iu} * e_{iv} * \cdots * e_{iw} \neq —\}$，ただし $R_A(c_{p+1}) = \{a_u, a_v, \cdots, a_w\}$，$0 \leqq p \leqq n-1$.

3.2 アクションへのラベル付け

$a_i \in A$ の添数 i は DGC 上の列番号を表わす．

〔定義 12〕 $A \supseteq B \neq \phi$ とするとき，$[B]_q = \{a_i \in B \mid q \leqq i \leqq m\}$，$1 \leqq q \leqq m$.

アクションの実行には，左から右への順序性が存在した．従って，ラベル付けられた a_q へ制御が移った後に実行の可能性あるアクション（以後，実行されるアクション，と呼ぶ）の集合 $R_A(a_q)$ は，a_q と，$[A]_{q+1}$ のいくつかの元とからなる．DGC は，条件，c_1, $\cdots$, c_n を一括判定した後，判定結果を満足するアクションの列を実行することを意味している．ここにいう判定結果は $\boldsymbol{x} \in V^n$ により，また，アクションの列は $Q(\boldsymbol{x})$ によって表わされる．$Q(\boldsymbol{x})$ のうち，a_q を元に持つものを $Q(\boldsymbol{x}_1)$, $Q(\boldsymbol{x}_2)$, $\cdots$ とすれば，a_q へ制御が移行した後に実行されるアクションの列は，$[Q(\boldsymbol{x}_1)]_q$，または $[Q(\boldsymbol{x}_2)]_q$，または $\cdots$ である．従って，移行後の流れが一意に定まるためには，$[Q(\boldsymbol{x}_1)]_q = [Q(\boldsymbol{x}_2)]_q = \cdots$ でなければならない．

〔定義 13〕 $\Omega_{A^q} = \{[Q(\boldsymbol{x})]_q \mid a_q \in Q(\boldsymbol{x}),\ \boldsymbol{x} \in V^n\}$ と置く．このとき，$|\Omega_{A^q}| = 1$ のとき，またそのときのみ，$a_q \in A$ へのラベル付けは一意である，という．ただし，$1 \leqq q \leqq m$.

〔定理 3〕 $a_q \in A$ へのラベル付けが一意であるた

めの必要十分条件は，$e_j^n < e_q^n$ なる e_j^n，または，e_q^n と第1種比較不可能な e_j^n が，存在しないことである．ただし，$1 \leqq q < j \leqq m$．（証明略）

Fig. 8 の場合，a_5 へのラベル付けは一意．Fig. 9 のアクションにラベルを付ける場合，a_1, a_3 へのラベル付けは一意でなく，a_2, a_4 へのラベル付けは一意．

[定理 4] $a_q \in A$ へのラベル付けが一意であるとき，$R_A(a_q) = [Q(e_q^n)]_q$; $1 \leqq q \leqq m$．（証明略）．

Fig. 8 の場合，$R_A(a_5) = [Q(Y, N, —)]_5 = \{a_5, a_6\}$．

4. おわりに

次の諸点について論じた．① DGC が持つあいまいさが指摘され，新しい約束の必要性が示された．新しい約束として，②条件のグループを明示するために記号 "—" の記入が提案され，③アクションのグループを明示するために，DGC をサブ DGC 群に分解するアルゴリズムが示された．次いで，新しい性質として，④DGC にはループを表現する能力が備わっていることが明らかになり，⑤ラベル付けの一意性の必要十分条件や，ラベル付きの文へ制御が移行した後に判断もしくは実行の対象となる文がどれであるかを導いた．さて，②の — 記号記入に伴うわずらわしさはむろん短所である．しかし，記入により，DGC のあいまいさが解消され，DGC の文書性が高まるなど，少なくとも短所を補う長所があるものと考えられる．③の分解アルゴリズムは，記述の都合上一見複雑であるが，考え方は直観的で簡単であり，通常の場合に対してはごく当然の考え方であると思われる．従って，この考え方が DGC 作成時の負担になることはないであろう．④のループの性質は，DGC の表現力と適用範囲と興味とを，少なからず拡大するものであることを確信する．ただ一つ欠点らしく見えるのは，ラベル付けが条件の判定に部分ごとの順序性をもたらすことである．しかし，これは欠点と考えるべきものでなく，当然の性質と見るべきものであろう．なぜならば，第一に，ループの表現力は DGC が持って生まれた自然の性質であること．第二に，ラベル欄を持つ DGC は，それを持たない（従来の）DGC のすべてを保存していること．最後に，ループを表現するには文の位置を明示する必要があり，位置の存在は順序を意味するつまり，ループの表現に順序性の存在は当然である．さて，⑤で得られた結果は，DGC を書く者と読む者（物）の双方に，ループに関する重要な情報を与えるものである．また，定義9と 13，定理1と3，定理2と4のそれぞれの類似性は，条件とアクションへのラベル付けが同質のものであることを示唆しているが，その具体的な意味は紙数の都合により割愛した．

なお，混合エントリ DGC の性質，定理1～4を利用した DGC の変換アルゴリズム，DGC とデシジョンテーブルとの関係などについては別に報告したい．

DGC のあいまいさが解消され，新たにループの表現能力が見出された．DGC で表わされた論理は，整理されて簡潔である．本論文の冒頭で述べた用途の他に，Fig. 5 のような形式による二次元言語として，また，プログラムの新しいモデルとしての DGC の利用が期待できる．

最後に本稿をまとめるに当って適切なご指示を頂いた査読者に御礼を申上げる．

参考文献

1) H. Strunz: The Developement of Decision Tables via Parsing of Complex Decision Situations, C. ACM. Vol. 16, No. 6, (1973)
2) King, P. J. H.: The Interpretation of Limited Entry Decision Table Format and Relationships among Conditions. Comp. J. Vol. 12, No. 4, (1969)
3) 守屋，平松：表言語とその処理法，情報処理，Vol. 13, No. 1, (1972)

（昭和 49 年 9 月 6 日受付）
（昭和 50 年 9 月 3 日再受付）