K-1718

AEC RESEARCH AND DEVELOPMENT REPORT

DECISION TABLES FOR DOCUMENTATION AND SYSTEM ANALYSIS

AUTHOR:

H. R. Gregg

UNION CARBIDE CORPORATION
NUCLEAR DIVISION

Operating the

OAK RIDGE GASEOUS DIFFUSION PLANT • OAK RIDGE NATIONAL LABORATORY
OAK RIDGE Y-12 PLANT • PADUCAH GASEOUS DIFFUSION PLANT

For the Atomic Energy Commission
Under U.S. Government Contract W7405 eng 26

DISCLAIMER

Printed in the United States of America. Available from Clearinghouse for Federal
Scientific and Technical Information, National Bureau of Standards,
U.S. Department of Commerce, Springfield, Virginia 22151
Price: Printed Copy $3.00; Microfiche $0.65

K-1718

Date of Issue: October 3, 1967

MATHEMATICS AND COMPUTERS

DECISION TABLES

FOR

DOCUMENTATION AND SYSTEM ANALYSIS

Author:

H. R. Gregg

Union Carbide Corporation
Nuclear Division
Computing Technology Center
Oak Ridge, Tennessee

THIS PAGE

WAS INTENTIONALLY

LEFT BLANK

TABLE OF CONTENTS

		Page
I	INTRODUCTION	5
II	BASIC CONCEPTS OF DECISION TABLES	7
	Limited Entry Table	12
	Extended Entry Table	13
	Mixed Entry Table	16
	Which Fort to Use	16
	How to Use Tables	17
	Manual Programming from Tables	19
III	SUMMARY	21

THIS PAGE

WAS INTENTIONALLY

LEFT BLANK

INTRODUCTION

The purpose of this manual is to describe the fundamental principles of decision tables and give some indication of the power and applicability of this technique as a programming tool. Relatively little use has been made of decision table techniques, particularly when compared to flow-charting.

One reason for this situation is that the decision table technique is generally considered as being an "advanced" tool. The tabular form of recording, however, has been known and used for many years as a means of summarizing facts.

The basic principles of decision tables are quite simple and one of the major advantages of the tables is that they provide a powerful means for expressing complex problems in a greatly simplified form.

This is not to say that decision tables should completely supplant flow charts. Functionally, the two are quite different. Decision tables are primarily an analytical technique, while flow charts are a documentation technique.

A decision table may be either an action or a result table. Basically, an action table is a compact representation of a procedure or system in which alternative courses of action are specified for various combinations of conditions. The table states what action (decision) should be taken for a given combination of conditions. This action (which actually may be several actions) is a decision rule which basically states that if such and such happens, then this and this should be done. This type of decision table may be viewed as a set of rules designed to tell what to do for given circumstances.

Tables have gained widespread recognition for such diverse purposes as income tax tables, railroad time tables, stock market quotations, race track results, economic forecasts, and many other areas. The purpose of such tabular presentations is to provide information in an easy to read and understand format.

BASIC CONCEPTS OF DECISION TABLES

The format and the compactness of tables allow complex situations to be more easily grasped and force the analyst to make a complete and accurate statement of the problem logic.

Consider the following examples taken from "Tabular Form in Decision Logic", Datamation, July, 1961, and "Decision Tables", General Information Tables - IBM:

> "Some orderly arrangement of data is necessary for the logical solution of any complex problem. At first glance, for example, the puzzle below does not lend itself to easy solution:
>
> Andy dislikes the catcher. Ed's sister is engaged to the second baseman. The center fielder is taller than the right fielder. Harry and the third baseman live in the same building. Paul and Allen each won $20 from the pitcher at pinochle. Ed and the outfielders play poker during their free time. The pitcher's wife is the third baseman's sister. All the battery and infield, except Allen, Harry, and Andy are shorter than Sam. Paul, Andy, and the shortstop lost $50 each at the racetrack. Paul, Harry, Bill, and the catcher took a trouncing from the second baseman at pool. Sam is undergoing a divorce suit. The catcher and the third baseman each have two children. Ed, Paul, Jerry, the right fielder, and the center fielder are bachelors. The others are married. The shortstop, the third baseman, and Bill each cleaned up $100 betting on the fight. One of the outfielders is either Mike or Andy. Jerry is taller than Bill. Mike is shorter than Bill. Each of them is heavier than the third baseman. With these facts, determine the names of the men playing the various positions on the baseball team."

"The puzzle is, however, far less complex than it appears when the problem is broken down into a series of steps based on the facts which determine the solution. For example, the first sentence of the puzzle implies that Andy is not the catcher; the fourth sentence, that Harry is not third baseman; the fifth, that neither Paul nor Allen is pitcher. These negative facts may be recorded as follows:

	P	C	1st	2nd	3rd	SS	LF	RF	CF
Andy		N							
Ed				N					
Harry					N				
Paul	N								
Allen	N								
Sam									
Bill									
Jerry									
Mike									

Figure 1. Data Organization

The recording of all such implied facts leads to the puzzle's solution:"

	P	C	1st	2nd	3rd	SS	LF	RF	CF
Andy	N	N	N	N	Y	N	N	N	N
Ed	N	N	N	N	N	Y	N	N	N
Harry	Y	N	N	N	N	N	N	N	N
Paul	N	N	Y	N	N	N	N	N	N
Allen	N	Y	N	N	N	N	N	N	N
Sam	N	N	N	N	N	N	Y	N	N
Bill	N	N	N	N	N	N	N	N	Y
Jerry	N	N	N	Y	N	N	N	N	N
Mike	N	N	N	N	N	N	N	Y	N

"Glancing around the office, I can see three young women busily engaged in the various duties of a typical work day. Let me tell you about them. Blond Marilyn is a chatterbox. Penelope and Theresa enjoy going to the movies. Marilyn is married, but the other two are single. Penelope has an attractive figure, while Marilyn is somewhat on the plump side. Theresa's quiet moods contrast to Penelope's happy ones, but they both seem to enjoy life in native Manhattan. Marilyn has dimples; Theresa may be recognized by her amber eyes and red hair. Unlike the others, Marilyn prefers Shakespeare and country living in Chappaqua.

Without looking back, can you recall all of Penelope's characteristics? Do you have a clear image of each girl and know what data is missing or where there are inconsistencies? To help answer these questions, let's rearrange the information. Displayed in tabular form, it would appear as in Figure 2:"

Name	Marilyn	Penelope	Theresa
Marital Status	Married	Single	Single
Hair Color	Blond		Red
Figure	Plump	Attractive	
Enjoys Movies		Yes	Yes
Prefers Shakespeare	Yes	No	No
Residence	Chappaque	Manhattan	Manhattan
Features	Dimples		Amber Eyes
Characteristics	Chatterbox	Happy	Quiet

Figure 2

> "From this illustration (Figure 2), some of the advantages of tables over narrative style for comparative data display can be readily appreciated. Conciseness and clarity are achieved by classifying data; Completeness is insured by revealing areas where information is missing; Meaningful relationships are recognized quickly and easily with the two dimensional structure."

While recognizing these advantages, many will point out that tables are merely a systematic way to present static data. Do they have a worthwhile function in a more dynamic situation - that of decision making? Would tables be valuable in systems analysis and computer programming?

A decision table is a tabular representation of a particular set of:

(1) Conditions: Variables that must be considered in reaching a decision.

(2) Actions: Operations that must be carried out when a given set of conditions exist.

(3) Rules: Specific sets of conditions and the actions dictated by these conditions.

(4) Entries: Additional information about either a condition or action pertinent to a particular rule.

These elements are defined in a decision table as:

Rules

Condition Statement	Condition Entry
Action Statement	Action Entry

Vertical Rule Orientation

Stub	Rule 1	Rule 2	Rule 3
Over 18 years of Age	T	T	F
Under 26 years of Age	T	F	T
Draft Eligible	T	F	F

Figure 3

This table could also be written as horizontal rule orientation:

Rule No.	Over 18 Yrs. of Age	Under 26 Yrs. of Age	Draft Eligible
1	Y	Y	Y
2	Y	N	N
3	N	Y	N

Figure 4

Decision tables may take one of three forms:

(1) Limited entry table

(2) Extended entry table

(3) Mixed entry table

Limited Entry Table

The entire condition or action is written in the stub. The condition entry area contains representations of the values true or false or yes or no or something to that effect (See Figures 3 and 4).

Although it is standard to use horizontal rule format, the vertical rule format may be superior if space limitation to a page is critical (See Figure 5).

	1	2	3	4	5	6	7
Male	F	T	T	T	T	T	T
Healthy		F	T	T	T	T	T
Over 18			F	T	T	T	T
Under 26				F	T	T	T
Married					F	T	T
Children						F	T
Reject	X	X	X	X			X
Priority 1					X		
Priority 2						X	

Figure 5

The action statements in a limited entry table completely define the actions to be taken for the sets of conditions shown. The action entries merely approve or disapprove an action statement for the particular set of conditions shown in the column. The action entry per rule, therefore, is limited to execute or do not execute.

Extended Entry Table

The condition stub statements identifying the variables to be considered together with the information expressed in the condition entry blocks go to make up the statement condition variable.

	Rule 1	Rule 2	Rule 3	Rule 4
Age	<25	<25	≥25	≥25
Sex	"M"	"F"		
Accident	-	-	= 0	>0
Set Rate = Rate +	Risk Factor		Special Rate	
Set New Rate =	Rate	Rate	Rate	Rate

Figure 6

So many variations of extended entry decision tables are in common use today that it is impractical to describe the possible formats. These tables appear to have their greatest application in manually operated systems, such as the calculation of personal income taxes (See Figure 7). The widespread use of extended entry tables and the relative ease with which they are used by the general public is an indication of their value.

Decision tables are a good means of communication. Figure 8 was developed to determine what action should be taken for various transaction codes in the A.E.C. Funds Accounting System. The problem definition used six pages to describe transaction code requirements. A clear, concise, and logical presentation of the same facts are contained on only one page in a decision table (See Figure 8).

1966 TAX TABLE A—FOR SINGLE PERSONS

Read down the income columns below until you find the line covering the total income (page 1, line 9, Form 1040). Then read across to the appropriate column headed by the number corresponding to the number of your exemptions, this is your tax.

If your total income is—		And the number of exemptions is—			If your total income is—		And the number of exemptions is—					
At least	But less than	1	2	3 If 4 or more there is no tax	At least	But less than	1	2	3	4	5	6 If 7 or more there is no tax
		Your tax is—					Your tax is—					
$0	$900	$0	$0	$0	$2,450	$2,475	$236	$124	$23	$0	$0	$0
900	925	2	0	0	2,475	2,500	240	128	26	0	0	0
925	950	5	0	0	2,500	2,525	244	132	30	0	0	0
950	975	9	0	0	2,525	2,550	248	136	33	0	0	0
975	1,000	12	0	0	2,550	2,575	253	139	37	0	0	0
1,000	1,025	16	0	0	2,575	2,600	257	143	40	0	0	0
1,025	1,050	19	0	0	2,600	2,625	261	147	44	0	0	0
1,050	1,075	23	0	0	2,625	2,650	265	151	47	0	0	0
1,075	1,100	26	0	0	2,650	2,675	270	155	51	0	0	0
1,100	1,125	30	0	0	2,675	2,700	274	159	54	0	0	0
1,125	1,150	33	0	0	2,700	2,725	278	163	58	0	0	0
1,150	1,175	37	0	0	2,725	2,750	282	167	61	0	0	0
1,175	1,200	40	0	0	2,750	2,775	287	171	65	0	0	0
1,200	1,225	44	0	0	2,775	2,800	291	175	68	0	0	0
1,225	1,250	47	0	0	2,800	2,825	295	179	72	0	0	0
1,250	1,275	51	0	0	2,825	2,850	299	183	76	0	0	0
1,275	1,300	54	0	0	2,850	2,875	304	187	79	0	0	0
1,300	1,325	58	0	0	2,875	2,900	308	191	83	0	0	0
1,325	1,350	61	0	0	2,900	2,925	312	195	87	0	0	0
1,350	1,375	65	0	0	2,925	2,950	317	199	91	0	0	0
1,375	1,400	68	0	0	2,950	2,975	322	203	94	0	0	0
1,400	1,425	72	0	0	2,975	3,000	327	207	98	0	0	0
1,425	1,450	76	0	0	3,000	3,050	333	213	104	4	0	0
1,450	1,475	79	0	0	3,050	3,100	342	221	111	11	0	0
1,475	1,500	83	0	0	3,100	3,150	350	229	119	18	0	0
1,500	1,525	87	0	0	3,150	3,200	359	238	126	25	0	0
1,525	1,550	91	0	0	3,200	3,250	367	246	134	32	0	0
1,550	1,575	94	0	0	3,250	3,300	376	255	141	39	0	0
1,575	1,600	98	0	0	3,300	3,350	385	263	149	46	0	0
1,600	1,625	102	2	0	3,350	3,400	393	272	157	53	0	0
1,625	1,650	106	5	0	3,400	3,450	402	280	165	60	0	0
1,650	1,675	109	9	0	3,450	3,500	410	289	173	67	0	0
1,675	1,700	113	12	0	3,500	3,550	419	297	181	74	0	0

Schedule III. Unmarried (or legally separated) taxpayers who qualify as HEAD OF HOUSEHOLD

If the amount on line 11d, page 1, is:		*Enter on line 12, page 1:*	
Not over $1,000		14% of the amount on line 11d.	
Over—	*But not over—*		*of excess over—*
$1,000	— $2,000	$140, plus 16%	— $1,000
$2,000	— $4,000	$300, plus 18%	— $2,000
$4,000	— $6,000	$660, plus 20%	— $4,000
$6,000	— $8,000	$1,060, plus 22%	— $6,000
$8,000	— $10,000	$1,500, plus 25%	— $8,000
$10,000	— $12,000	$2,000, plus 27%	— $10,000
$12,000	— $14,000	$2,540, plus 31%	— $12,000
$14,000	— $16,000	$3,160, plus 32%	— $14,000
$16,000	— $18,000	$3,800, plus 35%	— $16,000
$18,000	— $20,000	$4,500, plus 36%	— $18,000
$20,000	— $22,000	$5,220, plus 40%	— $20,000
$22,000	— $24,000	$6,020, plus 41%	— $22,000
$24,000	— $26,000	$6,840, plus 43%	— $24,000
$26,000	— $28,000	$7,700, plus 45%	— $26,000
$28,000	— $32,000	$8,600, plus 46%	— $28,000
$32,000	— $36,000	$10,440, plus 48%	— $32,000
$36,000	— $38,000	$12,360, plus 50%	— $36,000
$38,000	— $40,000	$13,360, plus 52%	— $38,000
$40,000	— $44,000	$14,400, plus 53%	— $40,000
$44,000	— $50,000	$16,520, plus 55%	— $44,000
$50,000	— $52,000	$19,820, plus 56%	— $50,000
$52,000	— $64,000	$20,940, plus 58%	— $52,000
$64,000	— $70,000	$27,900, plus 59%	— $64,000
$70,000	— $76,000	$31,440, plus 61%	— $70,000
$76,000	— $80,000	$35,100, plus 62%	— $76,000
$80,000	— $88,000	$37,580, plus 63%	— $80,000
$88,000	— $100,000	$42,620, plus 64%	— $88,000
$100,000	— $120,000	$50,300, plus 66%	— $100,000
$120,000	— $140,000	$63,500, plus 67%	— $120,000
$140,000	— $160,000	$76,900, plus 68%	— $140,000
$160,000	— $180,000	$90,500, plus 69%	— $160,000
$180,000		$104,300, plus 70%	— $180,000

Figure 7

Developed by:
Business Applications Research Section
Business Systems Department

AEC-ORO FUNDS ACCOUNTING

TRANSACTION CODE DECISION TABLE

Legend:
Y = YES
N = NO
X = TAKE THIS ACTION
- = NOT APPLICABLE

Conditions	Transaction Codes																						
	01	07-08 09-10	15	16	16	17	18	18	19-20-21 24-36-39	19	19	19	20-21	20-21	20-21	20-21	24	24	24	24	36-39	36-39	ELSE
Initial Obligation > 0	-	-	-	Y	N	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
LOC > PV/LOC	-	-	-	-	-	-	Y	N	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
AEC Direct (Contractor 99999)	-	-	-	-	-	-	-	-	-	Y	N	-	-	-	-	-	-	-	Y	N	-	-	-
Payments > Obligations	-	-	-	-	-	-	-	-	-	-	-	-	Y	N	N	N	Y	-	N	N	Y	N	-
Funds Code = 06 or 37 or 39	-	-	-	-	-	-	-	-	-	-	-	-	-	Y	N	N	-	Y	N	N	-	-	-
Obligations > 0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	Y	N	-	-	Y	Y	-	-	-
Payments ≠ Obligations	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	Y	Y	-	-	-
GLS Contra-Entry	-	-	-	-	-	-	-	-	N	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Final Payment > 0	-	-	-	-	-	-	-	-	-	Y	Y	N	-	-	-	-	-	-	-	-	-	-	-
Compute Prior Year's Totals	X	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Compute Quarterly & FY Allotment	-	X	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Compute Current & FY Obligation	-	-	X	X	-	-	-	-	-	-	-	-	-	-	-	-	-	-	X	-	-	-	-
Compute Current & FY Collection	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Reject Transaction	-	-	-	-	X	-	-	X	X	-	-	-	X	-	-	X	-	-	-	-	X	-	X
Issue Rejection Notice	-	-	-	-	X	-	-	X	X	-	-	-	X	-	-	X	-	-	-	-	X	-	X
Compute Current & FY LOC	-	-	-	-	-	X	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Compute Current & FY PV/LOC	-	-	-	-	-	-	X	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Issue Notice of Difference	-	-	-	-	-	-	-	-	-	-	X	-	-	-	-	-	-	-	-	X	-	-	-
Adjust Obligations to Payments	-	-	-	-	-	-	-	-	-	X	-	-	-	-	-	-	X	-	-	-	-	-	-
Issue Notice of Adjustment	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	X	-	X	-	-	-	-
Adjust Payments	-	-	-	-	-	-	-	-	-	-	X	X	-	-	-	-	-	-	-	-	-	-	-
Compute Current & FY Payments	-	-	-	-	-	-	-	-	-	-	-	-	-	X	X	-	-	X	X	X	-	-	-
Punch Payment Card	-	-	-	-	-	-	-	-	-	-	-	-	-	X	X	-	-	-	-	-	-	-	-
Compute Current & FY Non-Rec Trans	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	X	-

Figure 8

Mixed Entry Table

These tables incorporate some of the characteristics from both the limited entry table and the extended entry table. Although the extended and limited entry forms can be mixed within a table, one form must be used exclusively with a horizontal row of the table.

	Rule 1	Rule 2	Rule 3	Rule 4
Request is for	1st Class	1st Class	Tourist	Tourist
1st Class Available	Y	N		
Tourist Available			Y	N
Issue	1st Class		Tourist	
Place on Wait List		X		X

Figure 9

We have a mixed entry table here because there are relationships and commands and also yes or no and execution statements for entries.

Which Form to Use

The particular form used depends upon the preference of the table developer. Ease of understanding should be a major consideration. Except for a few fundamentals, the rules for constructing decision tables are not too explicit. The situation which the table describes and the use to which the table is put are factors that influence the table structure. The important thing is that the table be easily read.

As pointed out earlier, one of the primary advantages of decision tables is that they permit a complex decision making process to be portioned into various component parts, each by an individual table. This means that the

table must allow for a transfer of control from one table to another. This is accomplished with appropriate action entries and the else rule (See Figure 10), i.e., indicates either true or false - take action as indicated.

	Rule 1	Rule 2	Rule 3	Rule 4	Else
Condition Variable 1	T	T	T	F	-
Condition Variable 2	F	F	T	F	-
Condition Variable 3	T	F		F	-
Condition Variable 4	F	F	F	T	-
Action 1	1	2	1		-
Action 2	2	1	1	X	-
GO TO	Table 3	Table 3	Table 6	Table 3	Table 2

Figure 10

The numbers in the action portion of Figure 10 indicate the order in which the actions are to be performed on each rule. When two actions have the same number on the same rule, it indicates that either may be performed first or they may be performed simultaneously. This could be most important for third generation computers with the ability to process in line and out of line operations simultaneously.

How to Use Tables

Tables should be relatively small in the number of conditions, number of rules, and the number of actions. The nature of such tables helps break the applications or system into logical modules. This will allow programming and checking the modules out independently. It also helps in program maintenance, since a change is more likely to be confined to a

program module. This step may be particularly important in real time systems for segmentation and overlay.

The analyst can make out rough tables during the interviews to help guide him in the interviews. He can then formalize the tables at his desk at which time questions usually come up. He can clear up these questions and check the accuracy of the tables with re-interviews. But as the person being interviewed states, "When such and such occurs, we do this," the analyst is in a position to make an entry into the table.

This method of building up the table fits in very well with the way the interviews normally are conducted. It is very difficult for the person being interviewed to stick to the so called "main line" procedure. Frequently, the person finds it necessary to say, "We do this, but of course, if such and such occurs, we do that." With a decision table being developed, the person can jump around almost as much as he wants. But if the analyst is trying to flow chart the procedure, the person is pressured to stick to one path at a time.

The next step is to develop the decision tables for the new system, based on what the analyst has learned about the current system. The attempt here is to simplify and streamline as much as possible. There should be an attempt to combine rules that lead to the same actions. For example, if the conditions for one rule are YNNY, and the conditions for the second rule are YNYY, and if both result in the same action, the two rules should be combined with the use of - (don't care) YN-Y, since the third condition in this case is doesn't matter.

The analyst, of course, should go further than just combining rules within a table. He should try to eliminate whole tables by looking for different and better ways to accomplish the same end results.

Manual Programming from Tables

Before programming begins, it is desirable (and usually necessary) for the programmer to analyze the decision table for completeness, lack of redundancy, and lack of contradiction.

Pollack has developed two algorithms for condition testing which will pick a good sequence to minimize execution time for the tests and the other for minimizing program storage space. Pollack's methods are based on minimizing a "dash count" or "weighted dash count" where the dashes represent indifference in condition testing.[1]

Pollack's Algorithm No. 1 Minimize Space:

1. Calculate column count, dash count and delta.
2. Select minimum dash count. If equals, select maximum delta.
3. Generate branch and partition.
4. Repeat if necessary.

Frequency X DC = WDC		Frequency	50	20	20	10	Σ Dashes	T-F
		Dash Count to 2^n	8	4	4	1	DC	Δ
			1	2	3	4		
80		C.V.1	T	-	F	F	4	3
400		C.V.2	-	T	F	F	8	1
560		C.V.3	-	-	-	F	16	1
80		C.V.4	F	T	-	F	4	5
400		C.V.5	-	F	T	F	8	1

Figure 11

DC = Σ dashes
WDC = Frequency .DC
Δ = True - False

[1] Pollack, S., "Analysis of Decision Rules in Decision Tables", May, 1963.

Pollack's Algorithm No. 2 Minimize Time:

1. Calculate Column Count WDC + DELTA
2. Select Minimum WDC. If equal, select minimum delta.
3. Generate branch and partition.
4. Repeat if necessary.

A more simple method has been proposed by Egler[2]. All that is required is to arrange the conditions in descending order from most general to least general. That is, the condition with the most Y's and N's (and least blanks or dashes) is placed first, the next most general is second, etc. When the conditions have been arranged in this sequence, the rules are then sorted. Blanks sort before N's which sort before Y's. This procedure does not in general minimize the number of executed tests, but it does provide a fairly effective and simple scheme. Richfield Oil has used this procedure and reports that while programmers can develop better programs for small tables, the procedure gives better results than the programmers usually achieve for larger tables.

As mentioned, some attention has been paid to the frequency that rules are used (or their probability of usage) in order to minimize the number of executed tests. The programmer might consider putting "rule counters" in the program to keep track of the times each rule is used, then rearrange his table in a least time priority.

Program segmentation seems to follow quite naturally from the use of tables. A program breaks down into tables and subroutines whose execution is controlled by the tables. The actions called for in a table can be other tables, formulae, or other straight line coding. This segmentation facilitates program changing.

Tables can be used recursively, that is, an action refers the program back into the entry point of the same table.

---

[2]Egler, J. F., "A Procedure for Converting Logic Table Conditions Into An Efficient Sequence of Test Instructions," Communications of the ACM, September, 1963.

SUMMARY

People have difficulty both in developing and using logic, mainly due to inability of the human mind to simultaneously consider more than a few alternatives and their relationships. The logic of a problem can always be written out in sentences, but this necessitates a sequential consideration of alternative paths rather than the more desirable simultaneous consideration. The written definition may be exact, but is prone to misinterpretation and erroneous conclusions. For this reason, techniques such as flow charts, symbolic logic, and algebra are used, but again, they are primarily sequential methods of expressing logic.

Decision tables are two-dimensional in nature and therefore enable us to fully express and consider both the sequential and parallel aspects of logic. This makes them uniquely suited to logic expression and development.