
International Standard



5806

INTERNATIONAL ORGANIZATION FOR STANDARDIZATION • МЕЖДУНАРОДНАЯ ОРГАНИЗАЦИЯ ПО СТАНДАРТИЗАЦИИ • ORGANISATION INTERNATIONALE DE NORMALISATION

Information processing — Specification of single-hit decision tables

Traitement de l'information — Spécification des tables de décision à une réponse

First edition — 1984-12-15

STANDARDSISO.COM : Click to view the full PDF of ISO 5806:1984

UDC 681.3 : 16

Ref. No. ISO 5806-1984 (E)

Descriptors : data processing, information interchange, decision tables, specifications.

Foreword

ISO (the International Organization for Standardization) is a worldwide federation of national standards bodies (ISO member bodies). The work of preparing International Standards is normally carried out through ISO technical committees. Each member body interested in a subject for which a technical committee has been established has the right to be represented on that committee. International organizations, governmental and non-governmental, in liaison with ISO, also take part in the work.

Draft International Standards adopted by the technical committees are circulated to the member bodies for approval before their acceptance as International Standards by the ISO Council. They are approved in accordance with ISO procedures requiring at least 75 % approval by the member bodies voting.

International Standard ISO 5806 was prepared by Technical Committee ISO/TC 97, *Information processing systems*.

STANDARDSISO.COM : Click to view the full PDF of ISO 5806:1984

Information processing — Specification of single-hit decision tables

1 Scope and field of application

This International Standard specifies the basic format of single-hit decision tables and the relevant definitions, together with recommended conventions for preparation and use.

NOTES

1 This International Standard is concerned with the use of decision tables in the context of the documentation of computer-based information systems. It is not concerned with other uses, such as for the representation of program statements.

2 The format and conventions for the preparation and use of "multiple-hit" decision tables are outside the scope of this International Standard.

2 References

ISO 2382, *Data processing — Vocabulary —*

Part 1: Fundamental terms.

Part 7: Digital computer programming.

3 Definitions

For the purpose of this International Standard the following definitions apply.

3.1 decision table: A table of all contingencies that are to be considered in the description of a problem together with the action to be taken (from ISO 2382/1).

3.2 "single-hit" decision table: A decision table where any set of conditions will be satisfied by one, and only one, rule.

3.3 "multiple-hit" decision table: A decision table where at least one set of conditions will be satisfied by more than one rule (see note 2 to clause 1).

3.4 rule: A single column through the condition and action entry parts of the table, defining a unique set of conditions to be satisfied and the actions to be taken in consequence. A rule is satisfied if all conditions meet the condition entries of the rule.

3.5 "ELSE"-rule: The actions to be taken for all combinations of conditions not covered by the other rules in the table.

NOTE — The use of the ELSE-rule facility is optional.

3.6 condition: A description of a contingency to be considered in the representation of a problem, or a reference to other procedures to be considered as part of the condition.

3.7 action: A description of an operation to be taken in the formulation of a solution.

3.8 condition entry: An indication of the relevance of a condition to a particular rule.

3.9 action entry: An indication of the relevance of an action to a particular rule.

3.10 condition stub: A list of all the conditions to be considered in the description of a problem.

3.11 action stub: A list of all the actions to be taken in the solution of a problem.

3.12 table heading: The symbolic name or other means of referencing a decision table from other documents. Alternatively, or in addition, a clear description of the table.

3.13 initialisation section: An optional list of unconditional actions to be executed sequentially before the first condition is examined; it may be written in the row which follows that of the table heading.

3.14 limited entry table: A decision table where all the conditions and actions are completely described without reference to the rules (see annex B, example 1).

3.15 extended entry table: A decision table where the conditions and actions are generally described but are incomplete: the specifications are completed by the values specified in the rules (see annex B, example 2).

3.16 mixed entry table: A decision table whose stub consists of rows in which limited and extended entries are written (see annex B, example 4).

3.17 complete table: A decision table where for all combinations of condition entries there exists a satisfying rule.

NOTE — In practical terms extended entry tables will include limited entries and are therefore mixed entry tables. Any extended or mixed entry table may be transformed into a limited entry table (see annex B, example 3).

4 Format

4.1 Decision tables

The general representation of a decision table is given in figure 1.

The body of the table shall be divided into four parts by double lines, drawn close (or alternatively, single thick lines): this is to separate the condition parts from the action parts, and stubs from entries.

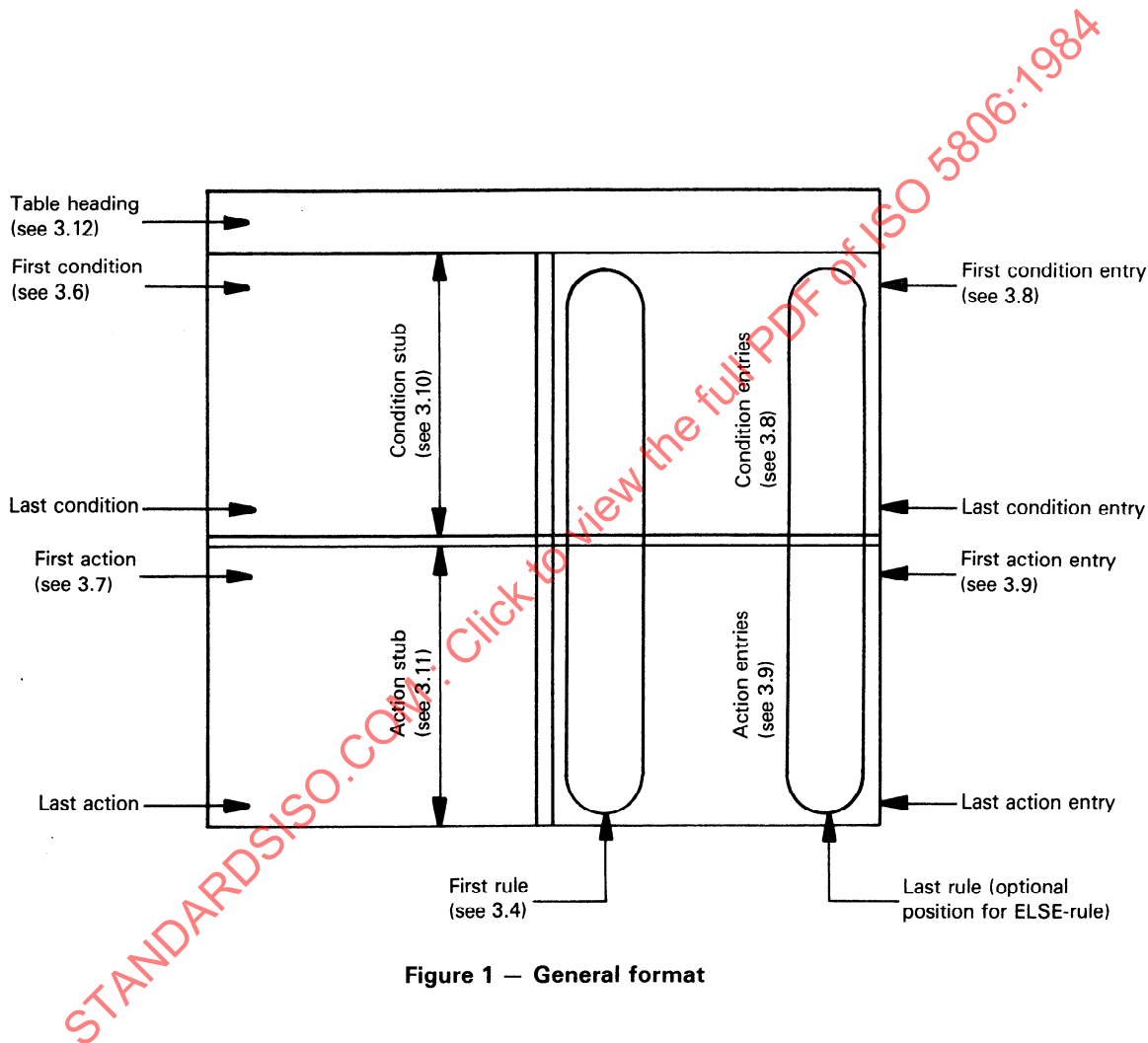


Figure 1 — General format

NOTE — The reading of a decision table may be facilitated by drawing: single thin horizontal lines between separate conditions, and similarly for separate actions; single thin vertical lines between separate rules. Conditions, actions and rules of a decision table may optionally be named, to allow a unique reference.

4.2 Condition entries

Form	Meaning within rule	Application
Y	The stated condition shall be fulfilled in satisfying this rule (Y = "Yes").	limited entries
N	The stated condition shall not be fulfilled in satisfying this rule (N = "No").	
Text, a value or a code	The text (or value or code) completes the specification of the otherwise incomplete condition for this rule; the condition shall then be fulfilled in satisfying the rule. If a code is used then it shall be described in a cross-referenced note.	extended entries
—	The stated condition is not relevant to the satisfaction of the rule; alternatively, the condition is logically impossible in the context of this rule; this may optionally be emphasized by the symbol "#" instead of "—".	any type of entry

NOTE — Any binary notation may be used to designate condition values.

4.3 Action entries

Form	Meaning within rule	Application
X	The stated action shall be taken when this rule is satisfied.	limited entries
Text, a value or a code	The text (or value or code) completes the specification of the otherwise incomplete action for this rule; the action shall be taken when the rule is satisfied. If a code is used then it shall be described in a cross-referenced note.	extended entries
—	The stated action shall not be taken when this rule is satisfied.	any type of entry

5 Relationships between table elements

5.1 Conditions

The relationship between successive conditions is the logical "AND": the first condition to be tested is assumed to be preceded by "IF". [Example: IF (first condition) AND (second condition), ..., AND (last condition)].

The order in which conditions are listed may be of importance. However, if the order is of no importance, tables may be easier to read if the more important, or "key" conditions are stated first: such a sequence might differ from the sequence preferred in programming.

5.2 Actions

The relationship between actions means successive execution: the first action to be taken is assumed to be preceded by "THEN", and the first action, the second action, ..., and the last action are successively executed.

Actions shall be stated in the order in which they are to be taken: where the execution sequence differs between rules then the actions shall be described as many times as necessary to show the various sequences. The use of sequence numbers is not recommended due to possible confusion with extended entry codes (see 4.3).

In any rule the last action to be taken should indicate where the next procedure is described unless the table is complete in itself.

5.3 Rules

The relationship between successive rules is the logical exclusive "OR".

The sequence of rules in a decision table is irrelevant: note, however, the convention that if an ELSE-rule is used then, to aid readability, it generally appears as the last rule in the table (see figure 1).

6 Relationships between decision tables

A large, and/or complex, problem may be described by a set of decision tables. There are four types of relationship, which may be combined:

- sequence;
- selection;
- repetition;
- nesting.

When decision tables are related then each shall be logically complete. The conditions in one table shall be tested independently of the results of condition tests in another: the effect of this requirement is that there is no relationship between the rules of related tables. This does not preclude such practices as the result of a condition test in one table being indicated through an action in that table (for example setting a flag) so that the indication may be inspected through a condition test in a subsequent table.

6.1 Sequence relationship

Two decision tables form a sequence if the first table has an immediate successor, as shown in figure 2. More than two decision tables may also form a sequence if the same general rule applies, that is, the n th is the only immediate successor to the $(n - 1)$ th.

It is recommended that in a sequence the preceding table shall include an action providing a pointer to the succeeding table.

This action will be the last to be taken in any rule where the succeeding table must be subsequently interpreted.

6.2 Selection relationship

Decision tables form a selection if the first table has more than one alternative immediate successor, as shown in figure 3.

It is recommended that in a selection the preceding table shall include actions providing pointers to the successive tables. The appropriate action will be the last to be taken in any rule where one of the succeeding tables must be subsequently interpreted.

6.3 Repetition relationship

A decision table may be interpreted by repetition if at least one rule requires re-examination of the condition in that table (see figure 4). Such a rule, or rules, require(s) to take as a last action some pointer to the same table.

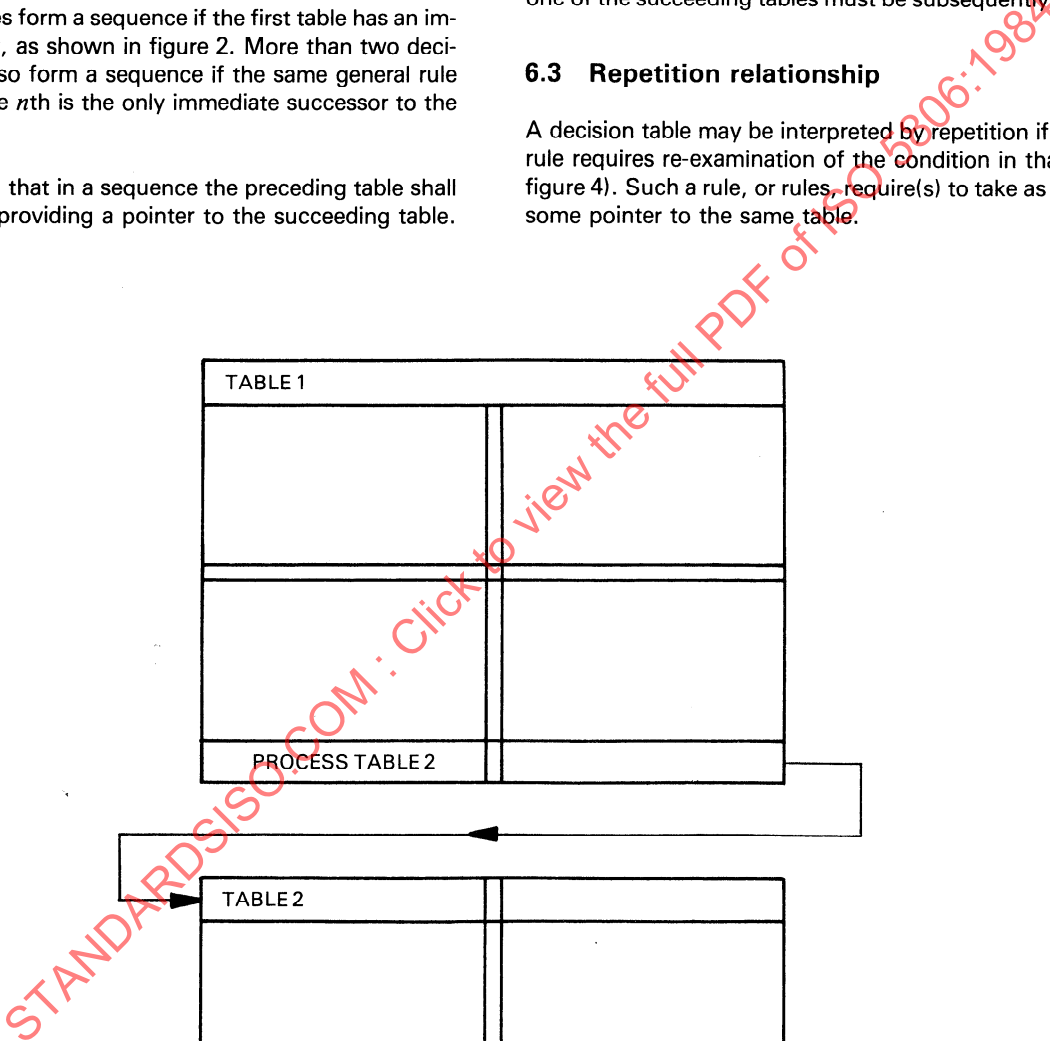


Figure 2 — Sequence of decision tables

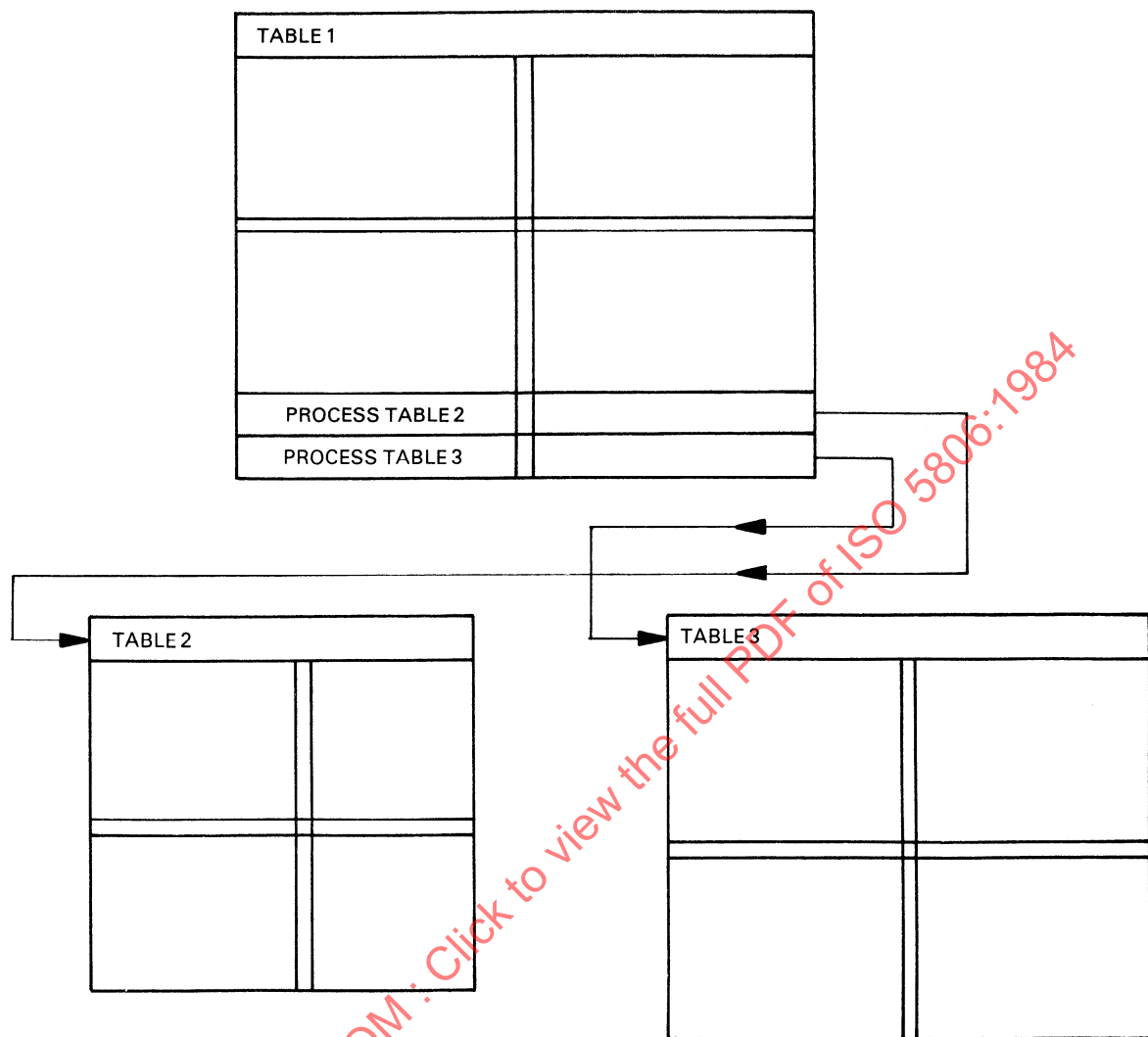


Figure 3 — Selection of decision tables

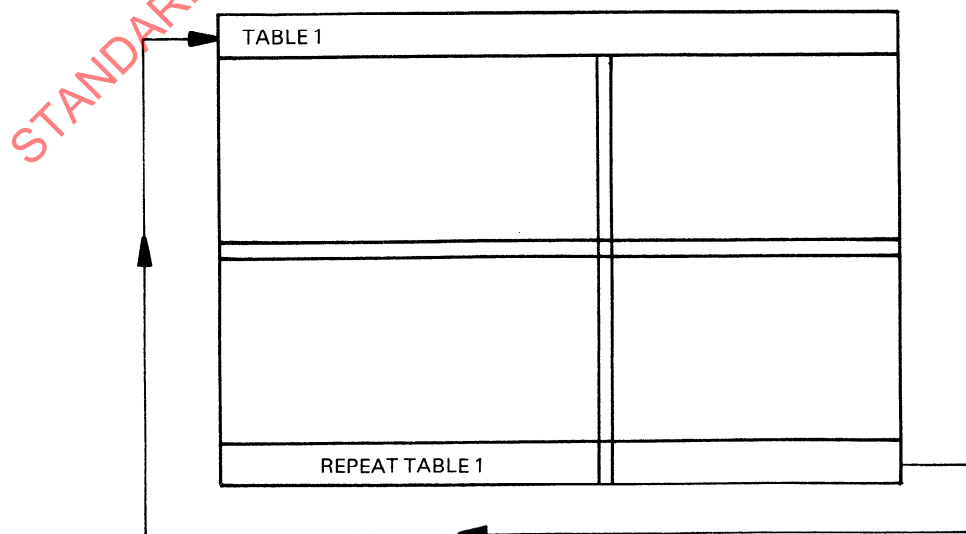


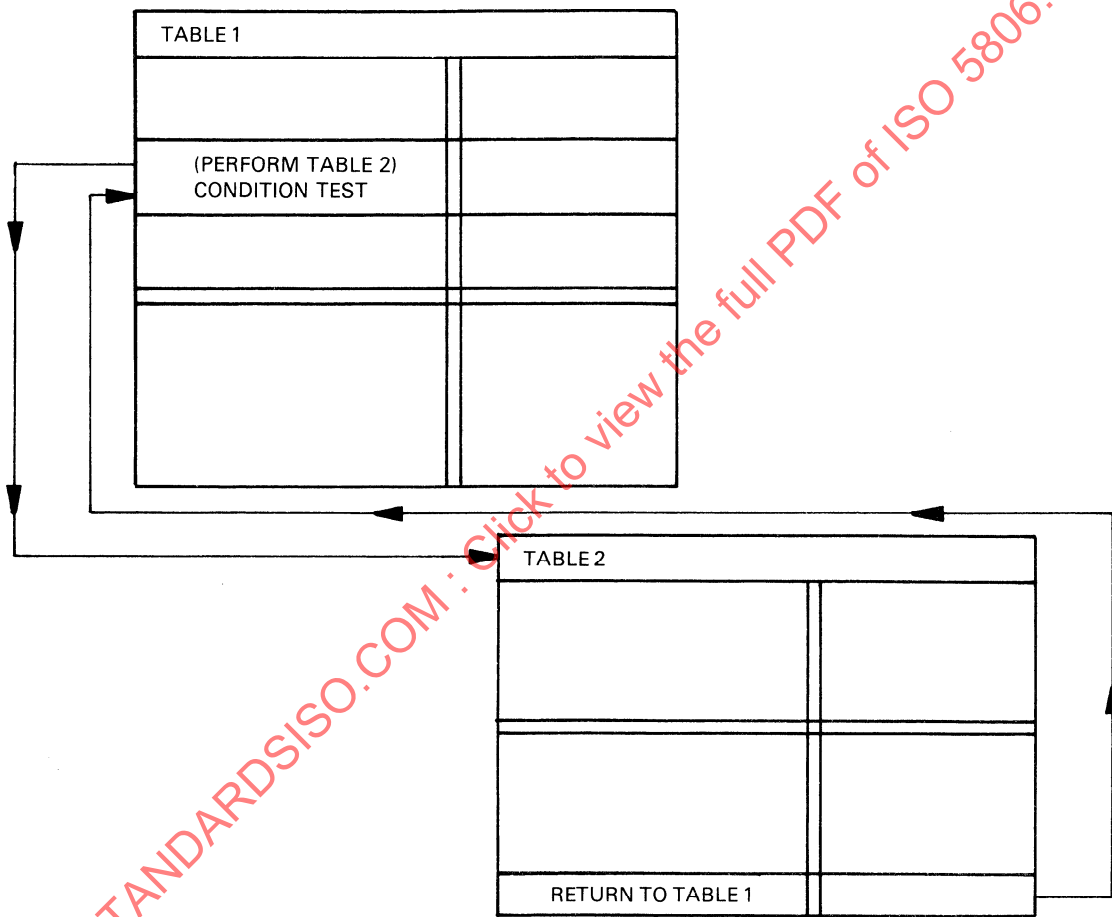
Figure 4 — Repetition of a decision table

6.4 Nesting relationship

Two decision tables have a nesting relationship if one table is completely interpreted whilst testing a condition (see figure 5) or taking an action (see figure 6) in the other. The relationship is as defined for nesting routines (see ISO 2382/7).

The nesting table will require some appropriate form of pointer in the relevant condition, or action, to the nested table. The

nested table will require an action which similarly points back to the nesting table. This action shall be the last taken for any rule in the nested table which is to continue the nesting relationship. The point indicated in the nesting table will be: for a condition, the condition from which the original exit was made, since the results of interpreting the nested table will be relevant to the test of that condition; for an action, the next relevant action.



NOTE — In this example, before CONDITION TEST in TABLE 1 is tested, TABLE 2 is executed and then CONDITION TEST in TABLE 1 is tested.

Figure 5 — Nested tables (exit at condition)

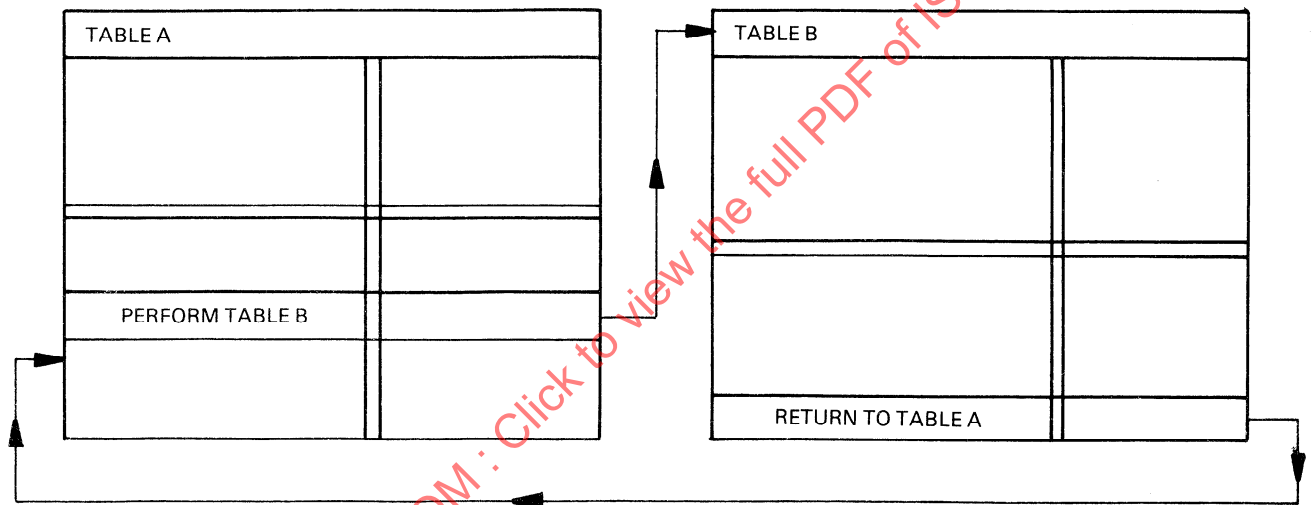


Figure 6 — Nested tables (exit at action)

6.5 Combinations of relationships

Any permutation of relationships may be used as necessary to describe the problem and its solution. Figure 7 illustrates a number of combined relationships.

TABLE 1 has two rules requiring repetition of that table. Two other rules sequence to TABLE 2. TABLE 2 has two rules sequencing further to TABLE 3 and two rules to TABLE 4.

TABLES 3 and 4 each have a nesting relationship with TABLE 5 in order to evaluate a condition.

The selection available from TABLE 1 is either

- repetition of TABLE 1;
- sequencing to TABLES 2, 3/nesting to 5; or
- sequencing to TABLES 2, 4/nesting to 5.

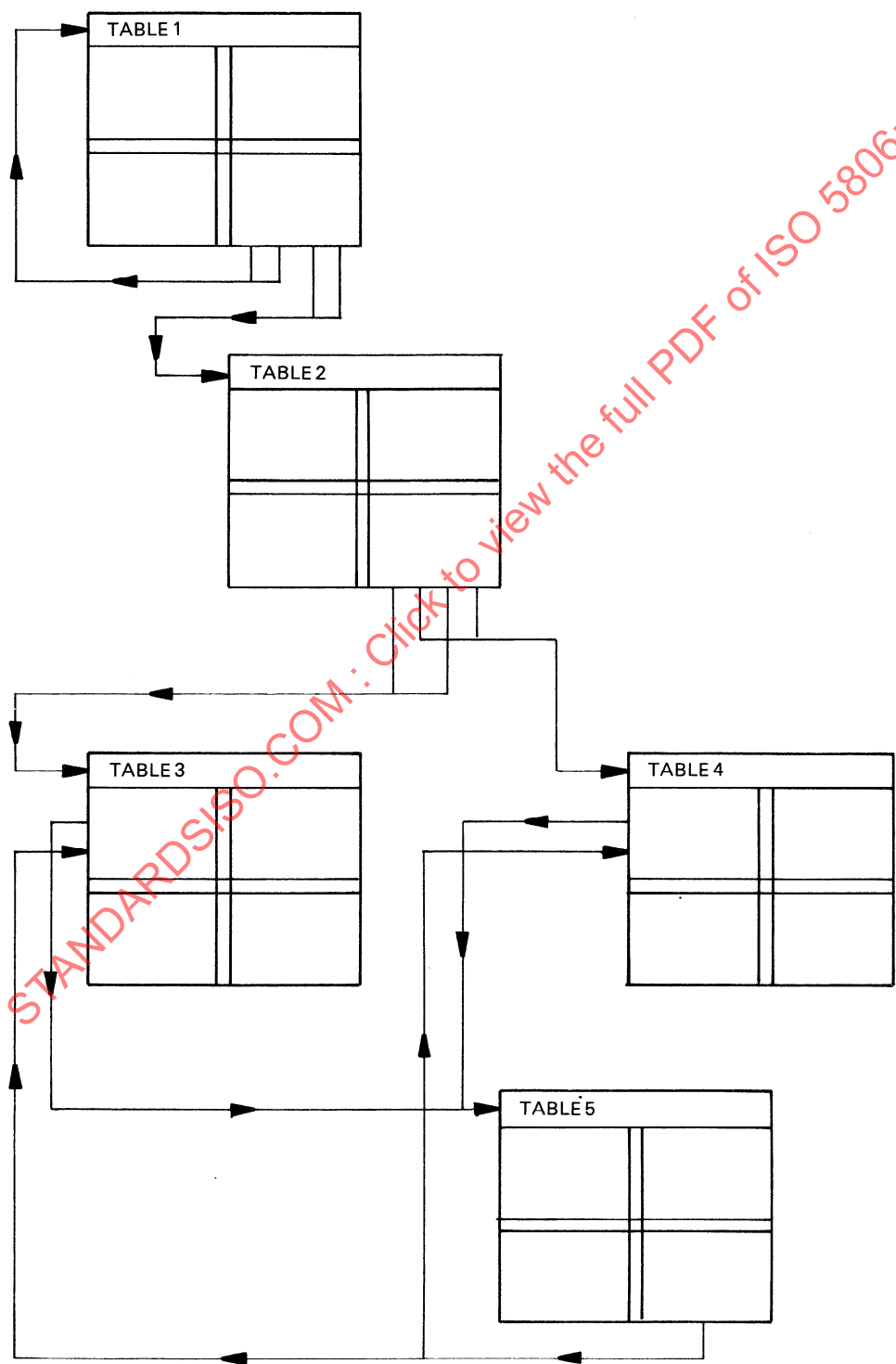


Figure 7 — Combined relationships

7 Interpretation of decision tables

7.1 Columnar method

The satisfied rule is found by determining the particular case and then comparing this case with each of the rules in turn. The steps required are

- a) test all conditions and identify their values for the particular case;
- b) compare the values found with each rule in turn until a single identical set of values is found and take all the actions specified for that rule, in sequence;
- c) if no rule is satisfied by the values for the particular case then all the actions specified for the ELSE-rule shall be taken in sequence.

7.2 Linear method

The satisfied rule is determined by a sequential test of each condition in turn. The steps required are

- a) test the first condition;
- b) ignore all those rules which are not satisfied by the result of this condition test;
- c) test the next condition which is relevant to the remaining rules: in effect this means ignoring any remaining conditions with only "—" condition entries (see 4.2). If the next

condition has "—" entries for some, but not all, of the remaining rules then the condition is only tested for rules specifying other than "—" entries;

d) repeat steps b) and c) until all conditions have been examined, i.e. being tested or ignored;

e) either a single rule is found which is satisfied by all the results of the condition tests or, if no rules remain then the ELSE-rule applies: whichever is true the actions specified for that rule are taken in sequence.

7.3 Completeness

By definition (see 3.2) the results of either method of interpretation must result in one, and only one, rule being satisfied. If an ELSE-rule is included in the table then, by definition (see 3.5), it cannot apply to a case which is satisfied by a specified rule.

Any table which includes an ELSE-rule is always complete. The ELSE-rule is, in effect, a default rule. The employment of an ELSE-rule requires care since it will be followed instead of rules omitted from the table by mistake.

If a table does not include an ELSE-rule then all logically possible permutations of conditions should be specified. For such a table care shall be taken in its preparation so that no permutation will be uncovered. The verification of completeness is an essential part of preparation.

Annex A

Recommendations for preparation

(This annex forms part of the standard.)

A.1 Condition entry construction

It is recommended that when a decision table is first drafted the full permutation of condition entries is listed before any attempt is made to reduce the size of the table. This will ensure that no combination of conditions is overlooked.

The complete number of rules in any table is always the product of the number of values permitted for each condition entry.

Example: A table has three conditions. For the entries

- a) condition 1 has two values;
- b) condition 2 has three values;
- c) condition 3 has four values.

Complete number of rules = $2 \times 3 \times 4 = 24$

The general procedure for constructing entries is therefore as follows:

- Step 1: Divide the total number of rules by the number of values permitted for the first condition entry. This will give the number of consecutive rules required for each of those values.

- Step 2: Divide the quotient from step 1 by the number of values for the next condition entry, to obtain the number of consecutive rules for each value.

- Step 3: Continue dividing each successive quotient by the number of values for successive conditions. The final quotient should be 1.

Example: An extended entry table has three conditions:

- a) condition 1 has two values: Y, N;
- b) condition 2 has three values: A, B, C;
- c) condition 3 has four values: 1, 2, 3, 4.

Complete number of rules = $2 \times 3 \times 4 = 24$

- Rules per value, condition 1 = $24 \div 2 = 12$
(i.e. $12 \times Y$'s, $12 \times N$'s).

- Rules per value, condition 2 = $12 \div 3 = 4$
(i.e. $4 \times A$'s, $4 \times B$'s, $4 \times C$'s).

- Rules per value, condition 3 = $4 \div 4 = 1$
(i.e. one each of 1, 2, 3, 4).

The complete permutation of condition entries is therefore as follows:

Condition 1	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	N	N	N	N	N	N	N	N	N	N
Condition 2	A	A	A	A	B	B	B	B	C	C	C	C	A	A	A	A	B	B	B	B	C	C
Condition 3	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2

NOTE — This method is cumbersome for large tables, and other methods of assuring completeness should be sought.