

8 Decision Table

8.1 Introduction

One of the ways to express the decision logic corresponding to the DRD decision artifact is a decision table. A decision table is a tabular representation of a set of related input and output expressions, organized into rules indicating which output entry applies to a specific set of input entries. The decision table contains all (and only) the inputs required to determine the output. Moreover, a complete table contains all possible combinations of input values (all the rules).

Decision tables and decision table hierarchies have a proven track record in decision logic representation. It is one of the purposes of DMN to standardize different forms and types of decision tables.

A decision table consists of:

- a name
- a set of inputs (0 or more). Each input is made of an *input expression* (the subject) and a number of *input entries*. The specification of input expression and all input entries is referred to as the input *clause*.
- a set of outputs (1 or more). Each output is made of an *output expression* (the name) and a number of *output entries*. The specification of output name and all output entries is referred to as the output clause.
- a list of rules (1 or more) in rows or columns of the table (depending on orientation), where each rule is composed of the specific input entries and output entries of the table row (or column). If the rules are expressed as rows, the columns are clauses, and vice versa.

Comment [JV1]: added

Comment [JV2]: Clause concept added

Comment [JV3]: Examples added

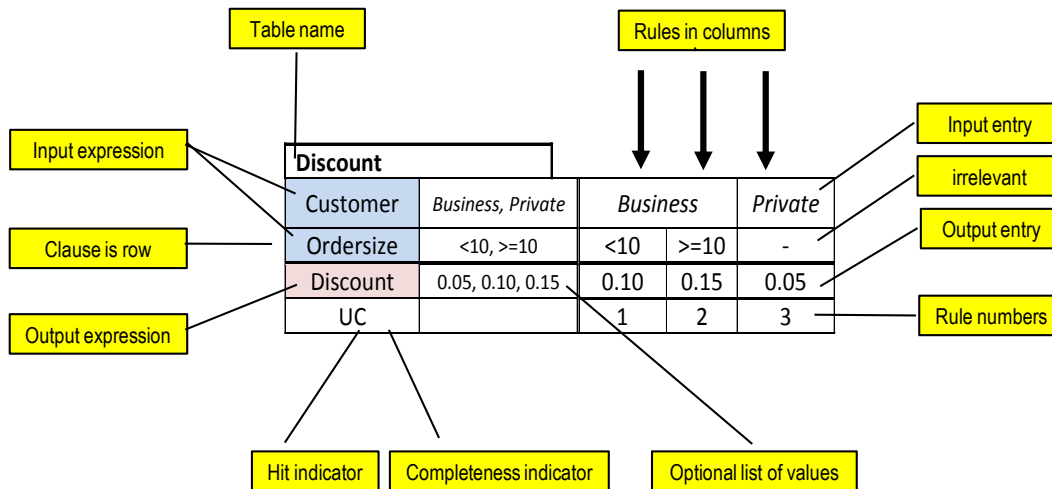


Figure x1: Decision table example (vertical orientation: rules in columns)

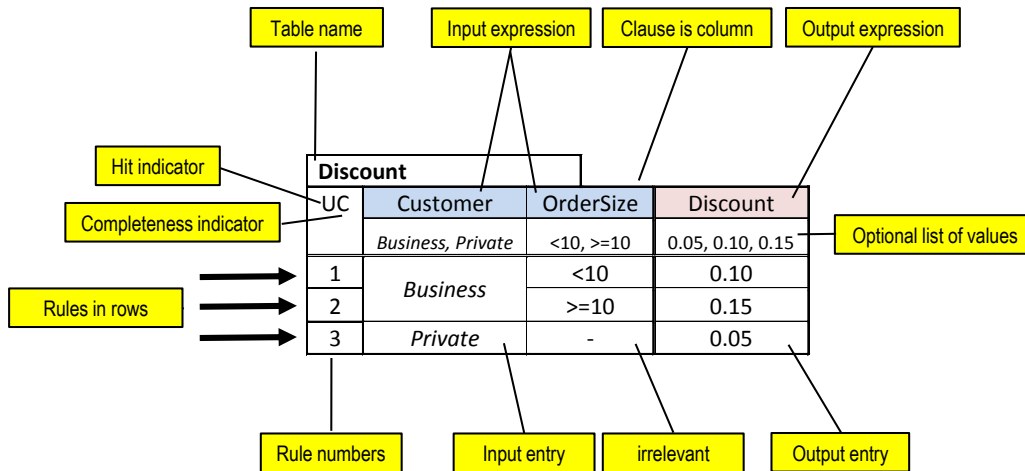


Figure x2: Decision table example (horizontal orientation: rules in rows)

The decision table shows the rules in a shorthand notation by arranging the entries in table cells. This shorthand notation shows all inputs in the same order in every rule and therefore has a number of readability and verification advantages.

Example:

Customer	OrderSize	Discount
<i>Business</i>	<10	0.10

reads as:

If Customer = “Business” **and** OrderSize < 10 **then** Discount = 0.10

In general, this is expressed as:

input expression 1	input expression 2	Output name
input entry a	input entry b	output entry c

The three highlighted cells in the decision table fragment above represent the following rule:

If the value of input expression 1 satisfies input entry a **and** the value of input expression 2 satisfies input entry b **then** the rule *matches* and the result for output name is output entry c.

Comment [JV4]: Example added

Comment [JV5]: Response to issue 67: The word "matches" is a bit vague - I think "satisfies" would be better

An input expression value *satisfies* an input entry if the value is equal to the input entry, or belongs to the list of values indicated by the input entry (e.g. a list or a range). If the input entry is '-' (meaning *irrelevant*), every value of the input expression satisfies the input entry and that particular clause is irrelevant in the specified rule.

Comment [JV6]: rephrased

A rule *matches* if the value of every input expression satisfies the corresponding input entry.

Comment [JV7]: Matching rephrased

If there are no input entries, any rule matches.

Comment [JV8]: added

The list of rules expresses the logic of the decision. For a given set of input values, the matching rule (or rules) indicate the resulting value for the output name. If rules *overlap*, multiple rules can match and a *hit policy* indicates how to handle the multiple matches.

If two input entries of the same input expression share no values, the entries (cells) are called *disjoint*. If there is an intersection, the entries are called *overlapping* (or even equal). 'Irrelevant' ('-') overlaps with any input entry of the input expression.

Comment [JV9]: Disjoint entries rephrased

Two rules are overlapping if all corresponding input entries are *overlapping*. A specific configuration of input data may then match the two rules.

Comment [JV10]: Response to issue 74: explain overlapping

Two rules are *disjoint* (non-overlapping) if at least one pair of corresponding input expressions is disjoint. No specific configuration of input data will match the two rules.

If tables are allowed to contain overlapping rules, the table hit policy indicates how overlapping rules have to be handled and which is the resulting value(s) for the output name, in order to avoid inconsistency.

The list of rules may contain all input entries to satisfy all expected combinations of input values, in which case the table is called *complete*.

8.2 Notation

This section builds on the generic notation for decision logic and boxed expressions defined in section 7.2.

A decision table representation standardizes:

- the orientation (rules as rows, columns or crosstab), as shown by the table
- placement of inputs, outputs and (optional) list of values in standard locations on a grid of cells. Each input expression is optionally associated with a list of input values. In this text the optional cells with lists of input values are indicated in *inverse*. Each output name is optionally associated with a list of output values. In this text the optional lists of output values are indicated in *inverse*.
- line style and optional use of color
- the contents of specific rule input and output entry cells
- the hit policy, indicating how to interpret overlapping rules
- the aggregation (optional), indicating how multiple hits are aggregated. The aggregation default is *collect*.
- placement of table name, hit policy (H), completeness indicator (C), aggregation and rule numbers as indicated in fig. 27-29. Rule numbers are consecutive natural numbers starting at 1. Rule numbering is required for tables with hit indicator F (first) or R (rule order), because the

Comment [JV11]: rephrased

Comment [JV12]: Response to issue 77: Specify how aggregation is specified in the written decision table (I only found out from the example on p84).

meaning depends on rule sequence. Crosstab tables have no rule numbers. Rule numbering is optional for other table types.

Comment [JV13]: Response to issue 69: Specify exactly how rules are numbered (integers? Natural numbers? Any sequence with a total ordering? Is numbering mandatory?)

8.2.1 Line style and color

Line style is normative. There is a double line between the inputs section and the outputs section, and there is a double line between inputs/output expressions and the rule entry cells. Other cells are separated by a single line.

Color is suggested, but does not influence the meaning. It is considered good practice to use different colors for the input expressions section and the output name section, and another (or no) color for the rule entries.

Comment [JV14]: rephrased

8.2.2 Table orientation

Depending on size, a decision table can be presented horizontally (rules as rows), vertically (rules as columns), or crosstab (rules composed from two input dimensions). Crosstab tables can only have the default hit policy (see later).

The table must be arranged in one of the following ways (see figure 27, 28, 29). Cells indicated in inverse are optional.

The input cell entry ‘-’ means ‘irrelevant’. HC is a placeholder for hit policy indicator (e.g. U, A, F, ...) and completeness indicator (see later).

Comment [JV15]: Response to issue 55: Also, we should not use italics to represent an optional cell because italics are used to represent a typographical string literal.

Comment [JV16]: Response to issue 70: The meaning of the letters "HC" in these tables is not introduced until p109. It's a long wait.

Comment [JV17]: Response to issue 70: The meaning of the letters "HC" in these tables is not introduced for another 30 pages. It's a long wait.

Comment [JV18]: Response to issue 77: Specify how aggregation is specified in the written decision table (I only found out from the example on p84).

table name			
HC	input expression 1	input expression 2	Output name
	Input value 1a, 1b	Input value 2a, 2b	Output value 1a, 1b
1	input entry 1.1	input entry 2.1	output entry 1.1
2		input entry 2.2	output entry 1.2
3	input entry 1.2	-	output entry 1.3
aggregation			

Figure 27a: Rules as rows: schematic layout

Discount				
UC	Customer	OrderSize	Delivery	Discount
	<i>Business, Private, Government</i>	<10, >=10	<i>sameday, slow</i>	0, 0.05, 0.10, 0.15
1	<i>Business</i>	<10	-	0.05
2		>=10	-	0.10
3	<i>Private</i>	-	<i>sameday</i>	0
4			<i>slow</i>	0.05
5	<i>Government</i>	-	-	0.15

Figure 28b: Rules as rows: example

Comment [JV19]: Response to issue 68: In several of these figures the "value 1a", "output entry 1a" etc labels are the same or different across rows and columns for no apparent reason. If there's no reason for them to match, please make them all different

table name				
input expression 1	Input value 1a, 1b	input entry 1.1		input entry 1.2
input expression 2	Input value 2a, 2b	input entry 2.1	input entry 2.2	-
Output name	Output value 1a, 1b	output entry 1.1	output entry 1.2	output entry 1.3
HC		1	2	3
aggregation				

Figure 29a: Rules as columns: schematic layout

Discount						
Customer	<i>Business, Private, Government</i>	<i>Business</i>		<i>Private</i>		<i>Government</i>
Ordersize	<10, >=10	<10	>=10	-		-
Delivery	<i>sameday, slow</i>	-	-	<i>sameday</i>	<i>slow</i>	-
Discount	0, 0.05, 0.10, 0.15	0.05	0.10	0	0.05	0.15
UC		1	2	3	4	5

Figure 30b: Rules as columns: example

table name			
Output name		input expression 1	
		input entry 1.1	input entry 1.2
input expression 2	input entry 2.1	output entry 1.1	output entry 1.3
	input entry 2.2	output entry 1.2	output entry 1.4

Figure 31a: Rules as crosstab: schematic layout (optional input and output values not shown)

Discount				
Discount		Customer		
		<i>Business</i>	<i>Private</i>	<i>Government</i>
Ordersize	<10	0.05	0	0.15
	>=10	0.10	0	0.15

Figure 32b: Rules as crosstab: simplified example with only two inputs

Discount		Customer, Delivery			
Discount		Business	Private		Government
		-	<i>sameday</i>	<i>slow</i>	-
Ordersize	<10	0.05	0	0.05	0.15
	>=10	0.10	0	0.05	0.15

Figure 33c: Rules as crosstab: the full example with three inputs

8.2.3 Input expressions

Input expressions are usually simple, for example, a name (e.g. CustomerStatus) or a test (e.g. Age<25). At conformance level 1, the input expression can be any text (e.g., natural language text), but shall not conflict with the look and feel of (S-)FEEL syntax. However, at conformance levels 2 and 3, expressions SHALL comply with (S-)FEEL syntax".

The order of input expressions is not related to any execution order in implementation.

8.2.4 Input values

Input expressions may be expected to result in a limited number or a limited range of values. It is important to model these expected input values, because a decision table will be considered complete if its rules cover all combinations of expected input values for all input expressions.

Regardless of how the expected input values are modeled, input values should be exclusive and complete. Exclusive means that input values are disjoint. Complete means that all relevant input values from the domain are present.

For example, the following two input value ranges overlap: <5, <10. The following two ranges are incomplete: <5, >5.

The list of input values is optional. If provided, it is a list of unary tests that must be satisfied by the corresponding input. At conformance level 1, the list elements can be any text (e.g., natural language text), but elements shall not conflict with the look and feel of (S-)FEEL syntax. However, at conformance levels 2 and 3, elements SHALL comply with (S-)FEEL syntax".

8.2.5 Table name and output name

The table name or the output name (or both) MUST be specified.

If the decision table is the value expression of a Decision or a Business Knowledge Model, then the table name and the output name SHALL be the name of the Decision or Business Knowledge Model.

If the decision table is contained in another boxed expression, then the table name SHALL be omitted and the output name SHALL be specified.

Comment [JV20]: Response to issue 66: Almost all the decision table examples and explanations have two input expressions

Comment [JV21]: Response to issue 71: Table entries "SHOULD NOT conflict with FEEL syntax" (4 places). OK, I know why you're saying this, but that "SHOULD NOT" is such a weak statement as to be useless. I suggest making it part of the conformance criteria

Comment [JV22]: added

Comment [JV23]: Response to issue 71: Table entries "SHOULD NOT conflict with FEEL syntax" (4 places). OK, I know why you're saying this, but that "SHOULD NOT" is such a weak statement as to be useless. I suggest making it part of the conformance criteria

8.2.6 Output values

The output entries of a decision table are often drawn from a list of output values.

The list of output values is optional. If provided, it is a list restricting output entries to the given list of values. At conformance level 1, the list elements can be any text (e.g., natural language text), but elements shall not conflict with the look and feel of (S-)FEEL syntax. However, at conformance levels 2 and 3, elements SHALL comply with (S-)FEEL syntax ".

When the hit policy is P (priority), meaning that multiple rules can match, but only one hit should be returned, the ordering of the list of output values is used to specify the (decreasing) priority.

The ordering of the list of output values is also used when the hit policy is output order.

8.2.7 Multiple outputs

The decision table can show a compound output (see Figure 34, Figure 35, Figure 36).

table name				
HC	input expression 1	input expression 2	output name	
	1	2	output 1	output 2
	input value 1a, input value 1b	input value 2a, input value 2b	output value 1a, output value 1b	output value 2a, output value 2b
1	input entry 1a	input entry 2a	output entry 1.1	output entry 2.1
2		input entry 2b	output entry 1.2	output entry 2.2
3	input entry 1b	-	output entry 1.3	output entry 2.3

Figure 34: Horizontal table with compound output

table name					
input expression 1		input value 1a, input value 1b	input entry 1a		input entry 1b
input expression 2		input value 2a, input value 2b	input entry 2a	input entry 2b	-
output name	output 1	output value 1a, output value 1b	output entry 1.1	output entry 1.2	output entry 1.3
	output 2	output value 2a, output value 2b	output entry 2.1	output entry 2.2	output entry 2.3
HC			1	2	3

Figure 35: Vertical table with compound output

Comment [JV24]: Response to issue 71: Table entries "SHOULD NOT conflict with FEEL syntax" (4 places). OK, I know why you're saying this, but that "SHOULD NOT" is such a weak statement as to be useless. I suggest making it part of the conformance criteria

Comment [JV25]: Response to issue 76: I don't understand how/where this list of output priorities is specified

Comment [JV26]: rephrased

table name			
output name		input expression 1	
output 1, output 2		input entry 1a	input entry 1b
input expression 2	input entry 2a	output entry 1.1, output entry 2.1	output entry 1.3 output entry 2.3
	input entry 2b	output entry 1.2, output entry 2.2	output entry 1.4, output entry 2.4

Figure 36: Crosstab with compound output

8.2.8 Input entries

Rule input entries are expressions. At conformance level 1, the expression can be any text (e.g., natural language text), but entries shall not conflict with the look and feel of (S-)FEEL syntax. However, at conformance levels 2 and 3, entries SHALL comply with (S-)FEEL syntax ".

A dash symbol ('-') can be used to mean any input value, *i.e.*, the input is irrelevant for the containing rule.

The input entries in a unary test should be '-' or a subset of the input values specified. For example, if the input values for input 'Age' are specified as [0..120], then an input entry of <0 SHOULD be reported as invalid.

Tables containing at least one '-' input entry are called *contracted* tables. The others are called *expanded*.

Tables where every input entry is *true*, *false*, or '-' are historically called *limited-entry* tables, but there is no need to maintain this restriction.

Evaluation of the input expressions in a decision table does not produce side-effects that influence the evaluation of other input expressions. This means that evaluating an expression or executing a rule should not change the evaluation of other expressions or rules of the same table. This is particularly important in first hit tables where the rules are evaluated in a predefined sequence: evaluating or executing a rule should not influence other rules.

The order of input entries is not related to any execution order in implementation.

8.2.9 Merged input entry cells

Adjacent input entry cells from different rules, with the same content and same (or no) prior cells can be merged, as shown in Figure 37 and Figure 38.

Comment [JV27]: Response to issue 71: Table entries "SHOULD NOT conflict with FEEL syntax" (4 places). OK, I know why you're saying this, but that "SHOULD NOT" is such a weak statement as to be useless. I suggest making it part of the conformance criteria

Comment [JV28]: rephrased

Comment [JV29]: Response to issue 72: "Where expressions in a decision table are expressed solely in FEEL, with no externally-defined functions, their evaluation does not produce side-effects. The behaviour of decision tables that call externally-defined functions with side-effects is undefined."

Comment [JV30]: Moved to input expressions

table name			
HC	input expression 1	input expression 2	Output name
	input value 1a, input value 1b	input value 2a, input value 2b	output value 1a, output value 1b
1	input entry 1a	input entry 2a	output entry 1.1
2		input entry 2b	output entry 1.2
3	input entry 1b	-	output entry 1.3

Figure 37: Merged rule input cells allowed

table name			
HC	input expression 1	input expression 2	Output name
	input value 1a, input value 1b	input value 2a, input value 2b	output value 1a, output value 1b
1	input entry 1a	input entry 2a	output entry 1.1
2		input entry 2b	output entry 1.2
3	input entry 1b	input entry 2a	output entry 1.3
4		input entry 2b	output entry 1.4

Figure 38: Merged rule input cells not allowed

8.2.10 Output entry

A rule output entry is an expression. At conformance level 1, the entry can be any text (e.g., natural language text), but entries shall not conflict with the look and feel of FEEL syntax. However, at conformance levels 2 or 3, entries SHALL comply with FEEL syntax".

Rule output cells cannot be merged (except in crosstabs, where adjacent output cells with the same content can be merged).

Shorthand notation

In vertical (rules as columns) tables with a single output name (equal to the table name), a shorthand notation may be used to indicate: output value applies ('X') or does not apply ('-'), as is common practice in decision tables.

Because there can be only one output entry for an output name, every rule must indicate no more than one 'X'. The other output entries must contain '-'.

The table in Figure 39 is shorthand notation for the table in Figure 40. It is called **shorthand**, because the output entries need not be (re-)written in every column, but are indicated with a one-character notation ('X' or '-'), thereby saving space in vertical tables, who tend to expand in width as the number of rules increases. The output values are written only once, before the rules, in the output expression part.

If a table name is provided, and there is only one output name (which has to be equal to the table name), the output name is optional.

Comment [JV31]: Response to issue 71: Table entries "SHOULD NOT conflict with FEEL syntax" (4 places). OK, I know why you're saying this, but that "SHOULD NOT" is such a weak statement as to be useless. I suggest making it part of the conformance criteria

Comment [GH32]: this makes me nervous because we give no rules for what can be merged. There could be many different ways to merge

Comment [JV33]: moved here

Comment [JV34]: Response to issue 53: Why call this format a 'shorthand'?

- Limited entry seems to work for both horizontal and vertical orientation. Why limit to vertical?
- There is nothing that states that every rule must have exactly 1 'X' in its limited output entry cells. I think we must say that, in Fig 35, 'output entry 1a' is the column of 3 cells containing, from top to bottom, 'X', '-', '-'. The constraint is that exactly 1 of the output entry's cells must contain an 'X', and the others must contain a '-'.

table name		Input value 1a, 1b, 1c	input entry 1.1		input entry 1.2	input entry 1.3
input expression 1		Input value 2a, 2b	input entry 2.1	input entry 2.2	-	-
Output name	output value 1a	Output value 1a, 1b, 1c	X	-	-	-
	output value 1b		-	X	-	-
	output value 1c		-	-	X	-
	output value 1d		-	-	-	X
HC			1	2	3	4

Figure 39: Shorthand notation for vertical tables (rules as columns)

table name		Input value 1a, 1b, 1c	input entry 1.1		input entry 1.2	input entry 1.3
input expression 1	Input value 2a, 2b	Input value 2a, 2b	input entry 2.1	input entry 2.2	-	-
input expression 2	Output value 1a, 1b, 1c	Output value 1a, 1b, 1c	output entry 1.1	output entry 1.2	output entry 1.3	output entry 1.4
Output name			1	2	3	4
HC						

Figure 40: Full notation for vertical tables (rules as columns)

8.2.11 Hit policy

A decision table normally has several rules.

As a default, rules do not overlap

If rules overlap, meaning that more than one rule may match a given set of input values, the hit policy indicator is required in order to recognize the table type and unambiguously understand the decision logic. The hit policy can be used to check correctness at design-time.

The hit policy specifies what the result of the decision table is in cases of overlapping rules, i.e. when more than one rule matches the input data. For clarity, the hit policy is summarized using a single character in a particular decision table cell. In horizontal tables this is the top-left cell (fig. 27) and in vertical tables this is the bottom-left cell (fig 28). The character is the initial letter of the defined hit policies (Unique, Any, Priority, First, Collect, Output Order, Rule order). Crosstab tables are always Unique and need no indicator.

The hit policy SHALL default to Unique, in which case the hit indicator is optional. Decision tables with the Unique hit policy SHALL NOT contain overlapping rules.

Comment [JV35]: Response to issue 53: It is misleading that the column with optional input/output values in Fig 36 is dropped in Fig 35. This column is optional independent of whether or not the 'shorthand' is employed. I think it should be possible to use the shorthand and also display the allowed values for the inputs (e.g. input value 1a, ...), but this might look ugly. Either way, we need to specify this clearly.

Comment [JV36]: added

Comment [JV37]: Response to issue 73: Please expand this sentence by saying precise which character (i.e. the initial letter of Unique, Any, Priority, First) and which cell.

Comment [JV38]: added

Comment [JV39]: Response to issue 74: "Decision tables with the Unique hit policy do not contain rules with overlapping input entries" should be "Decision tables with the Unique hit policy SHALL NOT contain rules with overlapping input entries".

Tools may support only a nonempty subset of hit policies, but the table type must be clear and therefore the hit policy indication is mandatory, except for the default unique tables. Unique tables must always be supported.

Single and multiple hit tables

A single hit table shall return the output of one rule only; a multiple hit table may return the output of multiple rules (or a function of the outputs, e.g. sum of values). If rules are allowed to overlap, the hit policy indicates how overlapping rules have to be interpreted.

The initial letter for hit policy also identifies if a table is single hit or multiple hit.

A single hit table may or may not contain overlapping rules, but returns the output of one rule only. In case of overlapping rules, the hit policy indicates which of the matching rules to select. Some restrictions apply to tables with compound outputs.

Single hit policies for single output decision tables are:

1. **Unique:** no overlap is possible and all rules are disjoint. Only a single rule can be matched. This is the default.
2. **Any:** there may be overlap, but all of the matching rules show equal output entries for each output, so any match can be used. If the output entries are non-equal, the hit policy is incorrect and the result is undefined.
3. **Priority:** multiple rules can match, with different output entries. This policy returns the matching rule with the highest output priority. Output priorities are specified in the ordered list of output values, in decreasing order of priority. Note that priorities are independent from rule sequence.
4. **First:** multiple (overlapping) rules can match, with different output entries. The first hit by rule order is returned (and evaluation can halt). This is still a common usage, because it resolves inconsistencies by forcing the first hit. However, first hit tables are not considered good practice because they do not offer a clear overview of the decision logic. It is important to distinguish this type of table from others because the meaning depends on the order of the rules. The last rule is often the catch-remainder. Because of this order, the table is hard to validate manually and therefore has to be used with care.

A multiple hit table may return output entries from multiple rules. The result will be a list of rule outputs or a simple function of the outputs.

Multiple hit policies for single output decision tables can be:

- ~~5. No order: returns all hits in a unique list in arbitrary order.~~
6. **Output order:** returns all hits in decreasing output priority order. Output priorities are specified in the ordered list of output values in decreasing order of priority.
7. **Rule order:** returns all hits in rule order. Note: the meaning may depend on the sequence of the rules.
8. **Collect:** returns all hits in arbitrary order. An operator ('+', '<', '#') can be added to apply a simple function to the outputs. If no operator is present, the result is the list of all the output entries.

Collect operators are::

- a) +: the result of the decision table is the sum of all the distinct outputs.
- b) <: the result of the decision table is the smallest value of all the outputs.
- e) ~~max. The result of the decision table is the largest value of all the outputs.~~

Comment [JV40]: Response to: issue 28. The empty subset is not considered valid.

Comment [JV41]: Response to issue 74: You should also specify absolutely precisely what "overlapping" means.

Comment [JV42]: added

Comment [JV43]: Response to issue 75: Please precisely specify what "same" means here (presumably all output entries are equal using the equal operator defined for their type). What happens if a table specifies "Any", but a pair of matching rules return non-equal outputs

Comment [JV44]: Response to issue 76: I don't understand how/where this list of output priorities is specified. I don't understand how this is different from the "First" single hit policy,

Comment [JV45]: Response to issue 76: I don't understand how/where this list of output priorities is specified.

Comment [JV46]: Response to issue 77: Specify how aggregation is specified in the written decision table (I only found out from the example on p84).

- d) #: the result of the decision table is the number of **distinct** outputs.
 - e) *mean*. The result of the decision table is the mean value of all the distinct outputs, defined as the sum divided by the count, where the semantics of *sum* and *count* are as specified above.
- Other policies, such as more complex manipulations on the outputs, can be performed by post-processing the output list (outside the decision table).

Comment [JV47]: Response to issue 77: For "count", you need to specify carefully whether the result is the number of rules that match or the number of distinct outputs (they're different if two or more rules return equal outputs).

The single letter for hit policy also identifies if a table is single hit or multiple hit.

To reduce complexity, decision tables with compound outputs support only the following hit policies: Unique, Any, First, **No order**, Rule order and Collect without operator, because the priority schema or collect operator over multiple outputs are undefined.

Comment [JV48]: In version 1

Note 1

Crosstab tables are unique and complete by definition and therefore do not need a hit policy or completeness indication.

Comment [JV49]: added

Note 2

The sequence of the rules in a decision table does not influence the meaning, except in **First** tables (single hit) and **Rule order** tables (multiple hit). These tables should be used with care.

8.2.12 Completeness indicator

Table completeness is an optional attribute. By default, tables are complete, producing a result for every expected case. If not, the indicator should read I(ncomplete).

Incomplete tables may specify a default output. The default value is underlined in the list of output values.

Comment [JV50]: We have no notation for this.

Comment [GEH51]: I suggest underlining the default output value, but this requires that we use some other notation to indicate optional cells, e.g. (output value 1a, output value 1b) instead of output value 1a, output value 1b uses parentheses to indicate the output values are optional and uses underline to indicate that output value 1b is the default

Comment [JV52]: added

8.2.13 Aggregation

Multiple hits must be aggregated into a single result. DMN 1.0 specifies six aggregation indicators, namely: collect, sum, min, max, count, average. Optionally, the aggregation indicator may be included in the table. The aggregation indicator is placed as an additional line at the bottom of the table in the form Aggregation=xyz (see p. 84 for example).

Comment [JV53]: Response to issue 77: Specify how aggregation is specified in the written decision table (I only found out from the example on p84).

The default aggregation indicator is *collect*.

Comment [GEH54]: Use of '=' here may be confused with a FEEL equality test. One alternative is to collapse with the multihit policy, described below. Another is to simply use the name of the built-in, e.g. sum, count, etc. Not clear we need an extra line at the bottom of the table (could we put this in the Hit cell, or next to it?)

Aggregation indicators have no incidence on decision tables with single hit policies.

In decision tables with multiple hit policies, the semantics of the aggregation indicators are:

Comment [GEH55]: SHALL NOT be used
This means that hit policy and aggregation are not independent. In fact, they can be combined. The list of multiple hit policies then becomes:
Rule order (R)
Output order (O)
Sum (+)
Minimum (<)
Maximum (>)
Count Distinct (C)
Mean (M)

- f) *collect*. The result of the decision table is the list of all the outputs, ordered or unordered per the hit policy.
- g) *sum*. The result of the decision table is the sum of all the distinct outputs.
- h) *min*. The result of the decision table is the smallest value of all the outputs.
- i) *max*. The result of the decision table is the largest value of all the outputs.
- j) *count*. The result of the decision table is the number of distinct outputs.
- k) *average*. The result of the decision table is the average value of all the distinct outputs, defined as the sum divided by the count, where the semantics of *sum* and *count* are as specified above.

Other policies, such as more complex manipulations on the outputs, can be performed by post-processing the output list (outside the decision table).

Comment [JV56]: Aggregation is now included in Collect

8.3 Metamodel

8.4 Examples

Table 1 provides examples for the various types of decision table discussed in this section. Further examples may be found in **Error! Reference source not found.**, in the context of a complete example of a DMN decision model.

Table 1: Examples of decision tables

Single Hit Unique	Applicant Risk Rating						
	U	Applicant Age		Medical History		Applicant Risk Rating	
	1	> 60		good		Medium	
	2			bad		High	
	3	[25..60]		-		Medium	
	4	< 25		good		Low	
	5			bad		Medium	
	Applicant Risk Rating						
	Applicant Age		< 25		[25..60]	> 60	
	Medical History		good	bad	-	good	bad
	Applicant Risk Rating		Low	Medium	Medium	Medium	High
	U		1	2	3	4	5
	Applicant Risk Rating						
	Applicant Age		< 25		[25..60]	> 60	
	Medical History		good	bad	-	good	bad
Low		X	-	-	-	-	
Medium		-	X	X	X	-	
High		-	-	-	-	X	
U		1	2	3	4	5	

Comment [GEH57]: Many of the entries here are string literals and should be italicized, i.e., *good*

Single Hit Any	Person Loan Compliance				
	A	Persons Credit Rating from Bureau	Person Credit Card Balance	Person Education Loan Balance	Person Loan Compliance
	1	A	< 10000	< 50000	Compliant
	2	not(A)	-	-	Not Compliant
	3	-	>= 10000	-	Not Compliant
4	-	-	>= 50000	Not Compliant	
<i>Example case: not A, >= \$10K, >= 50K -> Not Compliant (rules 2,3,4)</i>					
Single Hit Priority	Applicant Risk Rating				
	P	Applicant Age	Medical History	Applicant Risk Rating	
				<i>High, Medium, Low</i>	
	1	>= 25	good	Medium	
	2	> 60	bad	High	
3	-	bad	Medium		
4	< 25	good	Low		

Comment [GEH58]: Not legal syntax (except conformance level 1). Suggested: <10000

Comment [GEH59]: not(A)

Comment [JV60]: Double line changed

Single Hit First	<table border="1"> <thead> <tr> <th colspan="8">Special Discount</th> </tr> <tr> <th>F</th> <th>Type of Order</th> <th colspan="2">Customer Location</th> <th colspan="2">Type of Customer</th> <th colspan="2">Special Discount %</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Web</td> <td colspan="2">US</td> <td colspan="2">Wholesaler</td> <td colspan="2">10</td> </tr> <tr> <td>2</td> <td>Phone</td> <td colspan="2">-</td> <td colspan="2">-</td> <td colspan="2">0</td> </tr> <tr> <td>3</td> <td>-</td> <td colspan="2">Non-US</td> <td colspan="2">-</td> <td colspan="2">0</td> </tr> <tr> <td>4</td> <td>-</td> <td colspan="2">-</td> <td colspan="2">Retailer</td> <td colspan="2">5</td> </tr> </tbody> </table> <table border="1"> <thead> <tr> <th colspan="4">Special Discount</th> </tr> </thead> <tbody> <tr> <td>Type of Order</td> <td colspan="2">Web</td> <td>-</td> </tr> <tr> <td>Customer Location</td> <td colspan="2">US</td> <td>-</td> </tr> <tr> <td>Type of Customer</td> <td>Wholesaler</td> <td>Retailer</td> <td>-</td> </tr> <tr> <td>Special Discount %</td> <td>10</td> <td>5</td> <td>0</td> </tr> <tr> <td>F</td> <td>1</td> <td>2</td> <td>3</td> </tr> </tbody> </table> <p>Example case: Web, non-US, Retailer -> 0 (rule 3)</p>								Special Discount								F	Type of Order	Customer Location		Type of Customer		Special Discount %		1	Web	US		Wholesaler		10		2	Phone	-		-		0		3	-	Non-US		-		0		4	-	-		Retailer		5		Special Discount				Type of Order	Web		-	Customer Location	US		-	Type of Customer	Wholesaler	Retailer	-	Special Discount %	10	5	0	F	1	2	3
	Special Discount																																																																															
F	Type of Order	Customer Location		Type of Customer		Special Discount %																																																																										
1	Web	US		Wholesaler		10																																																																										
2	Phone	-		-		0																																																																										
3	-	Non-US		-		0																																																																										
4	-	-		Retailer		5																																																																										
Special Discount																																																																																
Type of Order	Web		-																																																																													
Customer Location	US		-																																																																													
Type of Customer	Wholesaler	Retailer	-																																																																													
Special Discount %	10	5	0																																																																													
F	1	2	3																																																																													
Multiple Hit Collect	<table border="1"> <thead> <tr> <th colspan="8">Holidays</th> </tr> </thead> <tbody> <tr> <td>Age</td> <td>-</td> <td><18</td> <td>>=60</td> <td>-</td> <td>[18..60)</td> <td>>=60</td> <td>-</td> </tr> <tr> <td>Years of Service</td> <td>-</td> <td>-</td> <td>-</td> <td>>=30</td> <td>[15..30)</td> <td>-</td> <td>>=30</td> </tr> <tr> <td>Holidays</td> <td>22</td> <td>5</td> <td>5</td> <td>5</td> <td>2</td> <td>3</td> <td>3</td> </tr> <tr> <td>C+</td> <td>1</td> <td>2</td> <td>3</td> <td>4</td> <td>5</td> <td>6</td> <td>7</td> </tr> <tr> <td>Aggregation=sum (Holidays)</td> <td colspan="7"></td> </tr> </tbody> </table> <p>Example case: Age=58, Service=31 -> Result=sum(22, 5, 3)=30</p>								Holidays								Age	-	<18	>=60	-	[18..60)	>=60	-	Years of Service	-	-	-	>=30	[15..30)	-	>=30	Holidays	22	5	5	5	2	3	3	C+	1	2	3	4	5	6	7	Aggregation=sum (Holidays)																															
Holidays																																																																																
Age	-	<18	>=60	-	[18..60)	>=60	-																																																																									
Years of Service	-	-	-	>=30	[15..30)	-	>=30																																																																									
Holidays	22	5	5	5	2	3	3																																																																									
C+	1	2	3	4	5	6	7																																																																									
Aggregation=sum (Holidays)																																																																																

Comment [GEH61]: I would put 'sum' here, and delete the following line containing Aggregation=...
As noted earlier, Sum implies No Order.

Comment [JV62]: Response to issue 77: Specify how aggregation is specified in the written decision table (I only found out from the example on p84).

Multiple Hit
Output order

Holidays			
O	Age	Years of Service	Holidays
			22, 5, 3, 2
1	-	-	22
2	>= 60	-	3
3	-	>= 30	3
4	< 18	-	5
5	>= 60	-	5
6	-	>= 30	5
7	[18..60)	[15..30)	2
8	[45..60)	< 30	2

Example case: Age=58, Service=31 -> Result=(22, 5, 3)

Multiple Hit
Rule order

Student Financial Package Eligibility				
R	Student GPA	Student Extra-Curricular Activities Count	Student National Honor Society Membership	Student Financial Package Eligibility List
1	> 3.5	>= 4	Yes	20% Scholarship
2	> 3.0	-	Yes	30% Loan
3	> 3.0	>= 2	No	20% Work-On-Campus
4	<= 3.0	-	-	5% Work-On-Campus

Example case: For GPA=3.6, EC Activities=4, NHS Membership -> result = (20% scholarship, 30% loan)

9 Simple Expression Language (S-FEEL)