10.3.2.8 Decision Table

The normative notation for decision tables is specified in Clause 8. Each input expression SHALL be a textual expression (grammar rule 2). Each list of input values SHALL be an instance of unary tests (grammar rule 17). The value that is tested is the value of the input expression of the containing InputClause. Each list of output values SHALL be an instance of unary tests (grammar rule 17). The value that is tested is the value of a selected output entry of the containing OutputClause. Each input entry SHALL be an instance of unary tests (grammar rule 17).

The decision table components are shown in Figure 36: Rules as rows – schematic layout, and also correspond to the metamodel in 8.3 Metamodel For convenience, Figure 36 is reproduced here.

[editing note: insert copy of Figure 36 here]

The semantics of a decision table is specified by first composing its literal expressions and unary tests into Boolean expressions that are mapped to the semantic domain, and composed into rule matches then rule hits. Finally some of the decision table output expressions are mapped to the semantic domain and comprise the result of the decision table interpretation. Decision table components are detailed in Table 63.

Table 63: Semantics of decision table

Component name (* means optional)	**Description**
input expression	One of the N>=0 input expressions, each a literal expression
input values*	One of the N input values, corresponding to the N input expressions. Each is a unary tests literal (see below).
output values*	A unary tests literal for the output. (In the event of M>1 output components (see Figure 43), each output component may have its own output values)
rules	a list of R>0 rules. A rule is a list of N input entries followed by M output entries. An input entry is a unary tests literal. An output entry is a literal expression.
hit policy*	one of: "U", "A", “P”, “F”, "R", "O", "C", "C+", "C#", "C<", “C>” (default is "U")
default output value*	The default output value is one of the output values. If M>1, then default output value is a context with entries composed of output component names and output values.

Unary tests (grammar rule 17) are used to represent both input values and input entries. An input expression *e* is said to *satisfy* an input entry *t* (with optional input values *v*), depending on the syntax of *t*, as follows:

- grammar rule 17.a: FEEL(*e in (t)*)**=true**
- grammar rule 17.b: FEEL(*e in (t)*)**=false**
- grammar rule 17.c when *v* is not provided: **e != null**
- grammar rule 17.c when *v* is provided: FEEL(*e in (v)*)=**true**

A rule with input entries *t1,t2,…,tN* is said to *match* the input expression list *[e1,e2,…,eN]* (with optional input values list *[v1,v2,…vN]*) if *ei satisfies ti* (with optional input values *vi*) for all *i* in 1..N.

A rule is *hit* if it is matched and the hit policy indicates that the matched rule's output value should be included in the decision table result. Each hit results in one output value (multiple outputs are collected into a single context value). Therefore, multiple hits require aggregation.

The hit policy is specified using the initial letter of one of the following boldface policy names.

Single hit policies:

- **Unique** – only a single rule can be matched
- **Any** – multiple rules can match, but they all have the same output
- **Priority** – multiple rules can match, with different outputs. The output that comes first in the supplied *output values* list is returned
- **First** – return the first match in rule order

Multiple hit policies:

- **Collect** – return a list of the outputs in arbitrary order
- **Rule order** – return a list of outputs in rule order
- **Output order** – return a list of outputs in the order of the *output values* list

The Collect policy may optionally specify an *aggregation*, as follows:

- **C**+ – return the sum of the outputs
- **C**# – return the count of the outputs
- **C**< – return the minimum-valued output
- **C>** – return the maximum-valued output

The *aggregation* is defined using the following built-in functions specified in 10.3.4.4 List functions: *sum, count, minimum, maximum.* To reduce complexity, decision tables with compound outputs do not support aggregation and support only the following hit policies: *Unique*, *Any*, *Priority, First*, *Collect without operator*, and *Rule order.*

A decision table may have no rule hit for a set of input values. In this case, the result is given by the default output value,

or **null** if no default output value is specified. A complete decision table SHALL NOT specify a default output value.

The semantics of a decision table invocation **DTI** are as follows:

1. Every rule in the rule list is matched with the input expression list. Matching is unordered.

2. If no rules match,

a. if a default output value *d* is specified, **DTI=**FEEL(*d*)

b. else **DTI=null**.

3. Else let *m* be the sublist of rules that match the input expression list. If the hit policy is "First" or "Rule order", order *m* by rule number.

a. Let *o* be a list of output expressions, where the expression at index *i* is the output expression from rule *m[i].*

The output expression of a rule in a single output decision table is simply the rule's output entry. The output expression of a multiple output decision table is a context with entries composed from the output names and the rule's corresponding output entries. If the hit policy is "Output order", the decision table SHALL be single output and *o* is ordered consistent with the order of the *output values.*

b. If a multiple hit policy is specified, DTI=FEEL(aggregation(o)), where aggregation is one of the built-in functions *sum, count, minimum* as specified in 10.3.4.4 List functions.

c. else **DTI=**FEEL(*o[1]*).