# **Proposal:**

**On page 74, section 8.1; (DMN 1.2\_ballot13 clean 70.pdf: page 77, section 8.1)**

**REPLACE the following sentence:**

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).

BY THE FOLLOWING:

An input expression value *satisfies* an input entry if the value is equal to the input entry, or it belongs to the list of values indicated by the input entry (e.g., a list or a range), or one of the expressions in the input entry evaluates to true. For the complete specification of the input entry satisfaction conditions, please refer to section ‘8.3.3 – Decision Rule Metamodel’.

**On page 80, section 8.2.8; (DMN 1.2\_ballot13 clean 70.pdf: page 83, section 8.2.7)**

 **REPLACE the following sentence:**

Rule input entries are expressions.

BY THE FOLLOWING:

Rule input entries are unary tests (grammar rule 17).

**On page 87, section 8.3.3; (DMN 1.2\_ballot13 clean 70.pdf: page 90, section 8.3.3)**

**REPLACE the following sentence:**

Otherwise, an instance of DecisionRule is said to be *applicable* if and only if, at least one of the rule's inputEntrys match their corresponding inputExpression value. The inputEntrys are matched in arbitrary order.

BY THE FOLLOWING:

Otherwise, an instance of DecisionRule is said to be *applicable* if and only if, all of the DecisionTable’s inputExpression values satisfy their corresponding inputEntry.

An inputExpression satisfies its corresponding inputEntry if and only if one of the following alternatives is true:

1. One of the expressions in the inputEntry evaluates to a value, and the inputExpression value is equal to that value;
2. One of the expressions in the inputEntry evaluates to a list of values, and the inputExpression value is equal to at least one of the values in that list;
3. One of the expressions in the inputEntry is a unary test, and the unary test evaluates to true when the inputExpression value is applied to it
4. One of the expressions in the inputEntry is a boolean expressions using the special ‘?’ variable and that expression evaluates to true when the inputExpression value is assigned to ‘?’

The inputEntrys are matched in arbitrary order.

**On page 108, section 10.3.1.2; (DMN 1.2\_ballot13 clean 70.pdf: page 111, section 10.3.1.2)**

**REPLACE the following GRAMMAR RULE:**

1. positive unary test = simple positive unary test | "null" ;

BY THE FOLLOWING:

1. positive unary test = expression

**DMN 1.2\_ballot13 clean 70.pdf: page 126, section 10.3.2.12**

**REMOVE the following sentence:**

In Grammar Rule **51c**, the qualified name must evaluate to a comparable constant value at modeling time, *i.e.* the endpoint must be a literal or a named constant.

**DMN 1.2\_ballot13 clean 70.pdf: page 126, table 47, section 10.3.2.12**

**ADD the following rows:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Grammar Rule**  | **FEEL Syntax**  | **Equivalent FEEL Syntax**  | **Applicability**  |
| 51.c | e1 in e2 | e1 = e2 | e2 is a qualified name that does **not** evaluate to a list |
| 51.c | e1 in e2 | list contains( e2, e1 ) | e1 is a simple value that is not a list and e2 is a qualified name that evaluates to a list |
| 51.c | e1 in e2 | { ? : e1, r : e2 }.r | e2 is a boolean expression that uses the special variable “?” |