## 19.1 Assignments

Values or templates may be assigned to variables or template variables (see clause 11). This is indicated by the symbol ":=".

***Syntactical Structure***

*VariableRef* ":=" ( *Expression* | *TemplateBody* )

***Semantic Description***

During execution of an assignment, the right-hand side of the assignment shall evaluate to a value or template. The effect of an assignment is to bind the variable to the value of the expression or to a template. The expression shall contain no unbound variables. Assignments are processed from left to right, i.e. expressions in the left hand side are evaluated before those in the right hand side. The evaluations obey the operator precedence defined in table 6. Unless the assignment is to a lazy or fuzzy variable or parameter, the right hand side is evaluated completely before the resulting value or template is bound to the evaluated left-hand side of the assignment. Whenever assignments are used within the right hand side of an assignment (due to assignment notation), these rules apply recursively.

When a direct or indirect element or field of a lazy or fuzzy variable is assigned, the variable is also evaluated as much as necessary before assignment, i.e. if an ancestor of that element or field is initialized with a function call, it shall be evaluated. Thus, if the variable is fully assigned, it does not need to be evaluated before assignment.

NOTE: If a sub-field or sub-element of a fuzzy variable is assigned that has an ancestor which was formerly assigned a function call, this function call will be evaluated once before the assignment and replaced by its result inside the variable. Thus, the other sub-fields and sub-elements of that ancestor, apart from the field or element being assigned become non-fuzzy.

***Restrictions***

In addition to the general static rules of TTCN‑3 given in clause 5, the following restrictions apply:

1. The right‑hand side of an assignment shall evaluate to a value or template, which is type compatible with the variable at the left-hand side of the assignment.
2. When the right‑hand side of the assignment evaluates to a template (global or local template, in-line template, template variable or a matching mechanism), the variable at the left hand side shall be a template variable.
3. If the left-hand side of the assignment is a reference to a non-optional value object (i.e. a value definition, a mandatory field, a record/set of/array element, a union alternative, a value parameter), the right-hand side shall not be a reference to an omitted field or the omit symbol.
4. Using a reference to an omitted field in the right-hand side of the assignment has the same effect as using the **omit** keyword.