### 142.1.1.3 Operations on variables

The ~~state diagram~~ operators used in state diagrams and in associated definitions of variables, constants, and functions are ~~shown~~defined in Table 142-1.  The operators are listed in decreasing order of precedence.

**Table 142-1—State diagram operators**

| Operator | Meaning |
|:---:|:---|
| (…) | Indicates precedence or a set of function arguments |
| […] | Array subscript |
| ++ | Unary opperator placed after a variable; increments the variable by 1 |
| -- | Unary opperator placed after a variable; decrements the variable by 1 |
| ! | Boolean NOT |
| * | Multiplication |
| / | Division |
| + | Addition |
| − | Subtraction |
| < | Less than (see 142.1.1.4) |
| > | More than (see 142.1.1.4) |
| ≤ | Less than or equal to (see 142.1.1.4) |
| ≥ | More than or equal to (see 142.1.1.4) |
| = | Equals (a test of equality) |
| != | Not equals |
| AND | Logical or bitwise AND. If one or both operands are defined as Boolean values, the operation is logical AND. Otherwise, the operation is considered the bitwise AND (each bit of the first operand is logically AND-ed with the corresponding bit of the second operand). |
| XOR | Logical or bitwise exclusive OR. If one or both operands are defined as Boolean values, the operation is logical XOR. Otherwise, the operation is considered the bitwise XOR (each bit of the first operand is logically XOR-ed with the corresponding bit of the second operand). |
| OR | Logical or bitwise OR. If one or both operands are defined as Boolean values, the operation is logical OR. Otherwise, the operation is considered the bitwise OR (each bit of the first operand is logically OR-ed with the corresponding bit of the second operand). |
| \| | Concatenation operation that combines several subfields or parameters into a single aggregated field or parameter |
| ∈ | Is a member of |
| ∉ | Is not a member of |
| ⇐ | Assignment operator |
| += | Increments left operand value by the value of the operand on the right ( x += y is equivalent to x ⇐ x+y) |
| −= | Decrements left operand value by the value of the operand on the right ( x −= y is equivalent to x ⇐ x-y) |