

Boolean implication and equivalence - Changes are relative to P1800-2008 Draft 4

Two new Boolean operators implication \rightarrow and equivalence \leftrightarrow are introduced. The operators can be used in any expressions. In fact, in constraints the operator \rightarrow already exists with the same meaning as the one to be introduced.

Syntax 11-7—Operator syntax (excerpt from Annex A)

Replace

```

assignment_operator ::=                                     //from A.6.2
    = | += | -= | *= | /= | %= | &= | |= | ^= | <<= | >>= | <<<= | >>>=
conditional_expression ::=                                 //from A.8.3
    cond_predicate ? { attribute_instance } expression : expression
unary_operator ::=                                       //from A.8.6
    + | - | ! | ~ | & | ~& | | | ~| | ^ | ~^ | ^~
binary_operator ::=
    + | - | * | / | % | == | != | === | !== | ==? | !=? | && | || | **
    | < | <= | > | >= | & | | | ^ | ^~ | ~^ | >> | << | >>> | <<<
inc_or_dec_operator ::= ++ | --
unary_module_path_operator ::=
    ! | ~ | & | ~& | | | ~| | ^ | ~^ | ^~
binary_module_path_operator ::=
    == | != | && | || | & | | | ^ | ^~ | ~^

```

With

```

assignment_operator ::=                                     //from A.6.2
    = | += | -= | *= | /= | %= | &= | |= | ^= | <<= | >>= | <<<= | >>>=
conditional_expression ::=                                 //from A.8.3
    cond_predicate ? { attribute_instance } expression : expression
unary_operator ::=                                       //from A.8.6
    + | - | ! | ~ | & | ~& | | | ~| | ^ | ~^ | ^~
binary_operator ::=
    + | - | * | / | % | == | != | === | !== | ==? | !=? | && | || | **
    | < | <= | > | >= | & | | | ^ | ^~ | ~^ | >> | << | >>> | <<<
    | -> | <->
inc_or_dec_operator ::= ++ | --
unary_module_path_operator ::=
    ! | ~ | & | ~& | | | ~| | ^ | ~^ | ^~
binary_module_path_operator ::=
    == | != | && | || | & | | | ^ | ^~ | ~^

```

Table 11-1: Operators and data types

Replace

&&	other binary logical operators	integral, real,
----	--------------------------------	--------------------

shortreal

With

&& || other binary logical operators
-> <->

integral,
real,
shortreal

Table 11-2—Legal operators for use in real expressions

Replace

! && || logical operators

With

! && || -> <-> logical operators

Table 11-4—Operator precedence and associativity

Replace

() [] :: .	left
+ - ! ~ & ~& ~ ^ ~^ ^~ ++ -- (unary)	
**	left
* / %	left
+ - (binary)	left
<< >> <<< >>>	left
< <= > >= inside dist	left
== != === !== ==? !=?	left
& (binary)	left
^ ~^ ^~ (binary)	left
(binary)	left
&&	left
	left
?: (conditional operator)	right
->	right
= += -= *= /= %= &= ^= =	none
<<= >>= <<<= >>>= := :/ <=	
{ } { }	concatenation

With

() [] :: .	left
+ - ! ~ & ~& ~ ^ ~^ ^~ ++ -- (unary)	
**	left
* / %	left
+ - (binary)	left
<< >> <<< >>>	left
< <= > >= inside dist	left
== != === !== ==? !=?	left
& (binary)	left
^ ~^ ^~ (binary)	left
(binary)	left
&&	left
	left
?: (conditional operator)	right

```

-> <->
= += -= *= /= %= &= ^= |=
<<= >>= <<<= >>>= := :/ <=
{} {{}}

```

```

right
none
concatenation

```

11.4.7 Logical operators

Replace

The operators *logical and* (&&) and *logical or* (||) are logical connectives. The result of the evaluation of a logical comparison shall be 1 (defined as true), 0 (defined as false), or, if the result is ambiguous, the unknown value (x). The precedence of && is greater than that of ||, and both are lower than relational and equality operators.

A third logical operator is the unary *logical negation* operator (!). The negation operator converts a nonzero or true operand into 0 and a zero or false operand into 1. An ambiguous truth value remains as x.

With

The operators *logical and* (&&), ~~and~~ *logical or* (||), *logical implication* (->) and *logical equivalence* (<->) are logical connectives. The result of the evaluation of a

logical ~~comparison operation~~ shall be 1 (defined as true), 0 (defined as false), or, if the result is ambiguous, the

unknown value (x). The precedence of && is greater than that of ||, and both are lower than relational and equality operators. ~~The logical implication expression1 -> expression2 is a shorthand for writing !(expression1) || expression2,~~

~~and the logical equivalence expression1 <-> expression2 is a shorthand for writing (expression1 -> expression2) && (expression2 -> expression1).~~

~~A third logical operator is the unary logical negation operator (!).~~ The unary *logical negation* operator (!) ~~negation operator~~ converts a nonzero

or true operand into 0 and a zero or false operand into 1. An ambiguous truth value remains as x.

A.8.6

Replace

binary_operator ::=

```

+ | - | * | / | % | == | != | === | !== | ===? | !==? | && | || | **
| < | <= | > | >= | & | | | ^ | ^~ | ~^ | >> | << | >>> | <<<

```

With

binary_operator ::=

```

+ | - | * | / | % | == | != | === | !== | ===? | !==? | && | || | **
| < | <= | > | >= | & | | | ^ | ^~ | ~^ | >> | << | >>> | <<<
| -> | <->

```