Boolean implication and equivalence - Changes are relative to P1800-2008 Draft 3a

Two new Boolean operators implication **->** and equivalence **<->** are introduced. The operators can be used in any expressions. In fact, in constraints the operator **->** 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

Between Syntax 11-7 and Table 11-1

Insert

The operator `expression1 -> expression2` is a shorthand for writing
`!(expression1) || expression2`
and
`expression1 <-> expression2` is a shorthand for
`(expression1 -> expression2) && (expression2 -> expression1)`

A.8.6

Replace

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

With

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