B.4 Precedence and Associativity

Expression operators bind most tightly, in the following precedence order, tightest first:

- unary operators: ~, ^ and *;
- dot join: **.** ;
- box join: *[]*;
- restriction operators: *<:* and *:>*;
- arrow product: ->;
- intersection: *&*;
- override: ++;
- cardinality: *#*;
- union and difference: + and -;
- expression quantifiers and multiplicities: *no*, *some*, *lone*, *one*, *set*;
- comparison negation operators: *!* and *not*;
- comparison operators: *in*, =, <, >, =<, =>.

Note, in particular, that dot join binds more tightly than box join, so *a.b[c]* is parsed as *(a.b)[c]*.

Logical operators are bound at lower precedence, as follows:

- negation operators: *!* and *not*;
- conjunction: *&&* and *and*;
- implication: =>, *implies*, and *else*;
- bi-implication: <=>, *iff*;
- disjunction: || and *or*;
- let and quantification operators: *let*, *no*, *some*, *lone*, *one* and *sum*.

All binary operators associate to the left, with the exception of implication, which associates to the right. So, for example, *p* => *q* => *r* is parsed as *p* => *(q* => *r)*, and *a.b.c* is parsed as *(a.b).c*.

In an implication, an else-clause is associated with its closest then-clause. So the constraint

```
p => q => r else s
```

for example, is parsed as

```
p => (q => r else s)
```