B Type Promotion and Operator Mapping

B.1 Type Promotion

Under certain circumstances, an atomic value can be promoted from one type to another. The promotion rules listed below are used in basic type conversions (see 2.1.3.3 Type Conversions) and during processing of arithmetic expressions (see 2.5 Arithmetic Expressions) and value comparisons (see 2.6.1 Value Comparisons). The rules may be applied transitively. For example, since xs:integer is promotable to xs:decimal and xs:decimal is promotable to xs:float, it follows that xs:integer is promotable to xs:float, without necessarily passing through the intermediate xs:decimal type.

The following type promotions are permitted:

1. A value of type xs:decimal can be promoted to the type xs:float.
2. A value of type xs:float can be promoted to the type xs:double.
3. A value of a derived type can be promoted to its base type. As an example of this rule, a value of the derived type xs:integer can be promoted to its base type xs:decimal.

B.2 Operator Mapping

The tables in this section list the combinations of datatypes for which the various operators of XPath are defined. For each valid combination of datatypes, the table indicates the function(s) that are used to implement the operator and the datatype of the result. Definitions of the functions can be found in [XQuery 1.0 and XPath 2.0 Functions and Operators]. Note that in some cases the function does not implement the full semantics of the given operator. For a complete description of each operator (including its behavior for empty sequences or sequences of length greater than one), see the descriptive material in the main part of this document.

In the following tables, the term numeric refers to the types xs:integer, xs:decimal, xs:float, and xs:double. When the result type of an operator is listed as numeric, it means "same as the highest type of any input operand, in promotion order." For example, when invoked with operands of type xs:integer and xs:float, the binary + operator returns a result of type xs:float.

Binary Operators (extracted from XPath 2.0 Draft 19 April 2002)

Operator	Type(A)	Type(B)	Function	Result type
A + B	numeric	numeric	op:numeric-add(A, B)	numeric
A - B	numeric	numeric	op:numeric-subtract(A, B)	numeric
A * B	numeric	numeric	op:numeric-multiply(A, B)	numeric
A div B	numeric	numeric	op:numeric-divide(A, B)	numeric
A mod B	numeric	numeric	op:numeric-mod(A, B)	numeric
A eq B	numeric	numeric	op:numeric-equal(A, B)	xs:boolean
A eq B	xs:boolean	xs:boolean	op:boolean-equal(A, B)	xs:boolean
A eq B	xs:string	xs:string	op:numeric-equal(xf:compare(A, B), 1)	xs:boolean

A eq B	xs:date	xs:date	(missing)	xs:boolean
A eq B	xs:time	xs:time	(missing)	xs:boolean
A eq B	xs:dateTime	xs:dateTime	op:datetime-equal(A, B)	xs:boolean
A ne B	numeric	numeric	xf:not(op:numeric-equal(A, B))	xs:boolean
A ne B	xs:boolean	xs:boolean	xf:not(op:boolean-equal(A, B))	xs:boolean
A ne B	xs:string	xs:string	xf:not(op:numeric-equal(xf:compare(A, B), 1))	xs:boolean
A ne B	xs:date	xs:date	(missing)	xs:boolean
A ne B	xs:time	xs:time	(missing)	xs:boolean
A ne B	xs:dateTime	xs:dateTime	xf:not3(op:datetime-equal(A, B))	xs:boolean
A gt B	numeric	numeric	op:numeric-greater-than(A, B)	xs:boolean
A gt B	xs:boolean	xs:boolean	op:boolean-greater-than(A, B)	xs:boolean
A gt B	xs:string	xs:string	op:numeric-greater-than(xf:compare(A, B), 0)	xs:boolean
A gt B	xs:date	xs:date	(missing)	xs:boolean
A gt B	xs:time	xs:time	(missing)	xs:boolean
A gt B	xs:dateTime	xs:dateTime	op:datetime-greater-than(A, B)	xs:boolean
A lt B	numeric	numeric	op:numeric-less-than(A, B)	xs:boolean
A lt B	xs:boolean	xs:boolean	op:boolean-less-then(A, B)	xs:boolean
A lt B	xs:string	xs:string	op:numeric-less-than(xf:compare(A, B), 0)	xs:boolean
A lt B	xs:date	xs:date	(missing)	xs:boolean
A lt B	xs:time	xs:time	(missing)	xs:boolean
A lt B	xs:dateTime	xs:dateTime	op:datetime-less-than(A, B)	xs:boolean
A ge B	numeric	numeric	op:numeric-less-than(B, A)	xs:boolean
A ge B	xs:string	xs:string	op:numeric-greater-than(xf:compare(A, B), -1)	xs:boolean
A ge B	xs:date	xs:date	(missing)	xs:boolean
A ge B	xs:time	xs:time	(missing)	xs:boolean
A ge B	xs:dateTime	xs:dateTime	op:datetime-less-than(B, A)	xs:boolean
A le B	numeric	numeric	op:numeric-greater-than(B, A)	xs:boolean
A le B	xs:string	xs:string	op:numeric-less-than(xf:compare(A, B), 1)	xs:boolean
A le B	xs:date	xs:date	(missing)	xs:boolean
A le B	xs:time	xs:time	(missing)	xs:boolean
A le B	xs:dateTime	xs:dateTime	op:datetime-greater-than(B, A)	xs:boolean