APÊNDICE A – XLPATH EBNF

```
LocationPath                    ::= RelativeLocationPath
                                | AbsoluteLocationPath
AbsoluteLocationPath            ::= '/' RelativeLocationPath?
                                | AbbreviatedAbsoluteLocationPath
RelativeLocationPath            ::= Step
                                | RelativeLocationPath '/' Step
                                | AbbreviatedRelativeLocationPath
Step                            ::= AxisSpecifier NodeTest Predicate*
                                | LevelSpec AxisSpecifier NodeTest Predicate*
                                | AbbreviatedStep
LevelSpec
    ::=             (‘HL’ | ‘LL’) ‘::’
AxisSpecifier                   ::= AxisName '::'
                                | AbbreviatedAxisSpecifier
AxisName                        ::= 'ancestor'
                                | 'ancestor-or-self'
                                | 'attribute'
                                | 'child'
                                | 'descendant'
                                | 'descendant-or-self'
                                | 'following'
                                | 'following-sibling'
                                | 'namespace'
                                | 'parent'
                                | 'preceding'
                                | 'preceding-sibling'
                                | 'self'
                                | 'link-destination'
                                | 'link-source'

| ‘arc-destination’
                                | ‘arc-source’
                                | ‘linked’
                                | ‘link-part-child’
                                | ‘link-part-descendant’
NodeTest                        ::= NameTest
                                | NodeType '(' ')'
                                | 'processing-instruction' '(' Literal ')'
                                | ElementSpec
ElementSpec                                                     ::=
    ElementName ‘(‘ QName? ‘)’
ElementName                                                     ::=
    ‘arc’
                                | ‘element’
                                | ‘locator’
                                | ‘remote’
                                | ‘resource’
Predicate                       ::= '[' PredicateExpr ']'
                                | ‘[[‘ PredicateExpr ']]'
PredicateExpr                   ::= Expr
                                | OperatorSpec
```

OperatorSpec ::=
AttributeSpec
| LinkSpec
| TextSpec

AttributeSpec ::=
'attribute(' QName ')=' Value

LinkSpec
::= 'link(' QName ')'

TextSpec ::= 'text()=' Value

AbbreviatedAbsoluteLocationPath ::= '//' RelativeLocationPath
| '///' RelativeLocationPath

AbbreviatedRelativeLocationPath ::= RelativeLocationPath '//' Step

AbbreviatedStep ::= '.'
| '..'
| '...'

AbbreviatedAxisSpecifier ::= '@'?

Expr ::= OrExpr

PrimaryExpr ::= VariableReference
| '(' Expr ')'
| Literal
| Number
| FunctionCall

FunctionCall ::= FunctionName '(' (Argument (',' Argument)*)? ')'

Argument ::= Expr

UnionExpr ::= PathExpr
| UnionExpr '|' PathExpr

PathExpr ::= LocationPath
| FilterExpr
| FilterExpr '/' RelativeLocationPath
| FilterExpr '//' RelativeLocationPath

FilterExpr ::= PrimaryExpr
| FilterExpr Predicate

OrExpr ::= AndExpr
| OrExpr 'or' AndExpr

AndExpr ::= EqualityExpr
| AndExpr 'and' EqualityExpr

EqualityExpr ::= RelationalExpr
| EqualityExpr '=' RelationalExpr
| EqualityExpr '!=' RelationalExpr

RelationalExpr ::= AdditiveExpr
| RelationalExpr '<' AdditiveExpr
| RelationalExpr '>' AdditiveExpr
| RelationalExpr '<=' AdditiveExpr
| RelationalExpr '>=' AdditiveExpr

AdditiveExpr ::= MultiplicativeExpr
| AdditiveExpr '+' MultiplicativeExpr
| AdditiveExpr '-' MultiplicativeExpr

MultiplicativeExpr ::= UnaryExpr
| MultiplicativeExpr MultiplyOperator UnaryExpr
| MultiplicativeExpr 'div' UnaryExpr
| MultiplicativeExpr 'mod' UnaryExpr

```
UnaryExpr                    ::=  UnionExpr
                                | '-' UnaryExpr
ExprToken                    ::=  '(' | ')' | '[' | ']' | '.' | '..' | '@' | ',' | '::' | ‘…’ | ‘[[‘ | ’]]’
                | NameTest
                | NodeType
                | Operator
                | FunctionName
                | AxisName
                | Literal
                | Number
                | VariableReference
Literal         ::=          '"' [^"]* '"'
                | "'" [^']* "'"
Number          ::=          Digits ('.' Digits?)?
                | '.' Digits
Digits          ::=          [0-9]+
Operator        ::=  OperatorName
                | MultiplyOperator
                | '/' | '//' | '///' | '|' | '+' | '-' | '=' | '!=' | '<' | '<='
                | '>' | '>='
OperatorName                 ::=  'and' | 'or' | 'mod' | 'div'
MultiplyOperator             ::=  '*'
FunctionName                 ::=  QName - NodeType
VariableReference            ::=  '$' QName
NameTest        ::=          '*'
                | NCName ':' '*'
                | QName
NodeType        ::=          'comment'
                | 'text'
                | 'processing-instruction'
                | 'node'
ExprWhitespace               ::=  S
```