# A Declarative Semantics for the ShEx grammar

August 11, 2015

## Contents

# 1 Interpretation of ShEx Document

## 1.1 i_shExDoc

shExDoc : directive* ((shape | start | startActions) statement*)?

## 1.2 i_statement

statement : directive |start |shape ;

$$
\begin{aligned}
&\mathit{statement} ::= \\
&\qquad \mathit{directive\_statement}\langle\langle \mathit{directive} \rangle\rangle \mid \\
&\qquad \mathit{start\_statement}\langle\langle \mathit{start} \rangle\rangle \mid \qquad \mathit{start\_shape}\langle\langle \mathit{shape} \rangle\rangle
\end{aligned}
$$

directive : baseDecl |prefixDecl ;

baseDecl : KW_BASE IRIREF ;

prefixDecl : KW_PREFIX PNAME_NS IRIREF ;

start : KW_START '=' ( shapeLabel |shapeDefinition semanticActions) ;

shape : KW_VIRTUAL? shapeLabel shapeDefinition semanticActions ;

shapeDefinition : (includeSet |inclPropertySet |KW_CLOSED)* '' oneOfShape? '' ;

includeSet : '&' shapeLabel ;

inclPropertySet : KW_EXTRA predicate+ ;

oneOfShape : someOfShape ( '|' someOfShape )* ;

someOfShape : groupShape ( '||' groupShape )* ;

groupShape : unaryShape ( ',' unaryShape )* ','? ;

unaryShape : shapeid? ( tripleConstraint |include |'(' oneOfShape ')' cardinality? semanticActions) ;

include : '&' shapeLabel ;

shapeid : '$' shapeLabel ;

shapeLabel : iri |blankNode ;

tripleConstraint : senseFlags? predicate valueClass cardinality? annotation* semanticActions;

senseFlags : '!' ''? |'' '!'? ;

predicate : iri ;

valueClass : KW_LITERAL xsFacet* # valueClassLiteral

|(KW_IRI |KW_BNODE |KW_NONLITERAL) groupShapeConstr? stringFacet* # valueClassNonLiteral

|datatype xsFacet* # valueClassDatatype

|groupShapeConstr # valueClassGroup

|valueSet # valueClassValueSet

|'.' # valueClassAny

;

groupShapeConstr : shapeOrRef (KW_OR shapeOrRef)* ;

shapeOrRef : ATPNAME_LN |ATPNAME_NS |'@' shapeLabel |shapeDefinition ;

xsFacet : stringFacet |numericFacet;

stringFacet : KW_PATTERN string |' ' string |stringLength INTEGER ;

stringLength : KW_LENGTH |KW_MINLENGTH |KW_MAXLENGTH;

numericFacet : numericRange INTEGER |numericLength INTEGER ;

numericRange : KW_MININCLUSIVE |KW_MINEXCLUSIVE |KW_MAXINCLUSIVE |KW_MAXEXCLUSIVE ;

numericLength : KW_TOTALDIGITS |KW_FRACTIONDIGITS ;

datatype : iri ;

annotation : ';' iri (iri |literal) ;

```
cardinality : '*' |'+' |'?' |repeatRange ;
```

```
repeatRange : '' INTEGER (',' (INTEGER |'*')?)? '' ;
```

```
valueSet : valueSetList |valueSetRef ;
```

```
valueSetList : '(' value* ')' ;
```

```
valueSetRef : '*' iri ;
```

```
value : iriRange |literal ;
```

```
iriRange : iri (' ' exclusion*)? |'.' exclusion+ ;
```

```
exclusion : '-' iri ' '? ;
```

```
literal : rdfLiteral |numericLiteral |booleanLiteral ;
```

```
numericLiteral : INTEGER |DECIMAL |DOUBLE ;
```

```
rdfLiteral : string (LANGTAG |'g datatype)? ;
```

```
booleanLiteral : KW_TRUE |KW_FALSE ;
```

```
string : STRING ;
```

```
iri : IRIREF # IRIREF
```

```
|prefixedName # PREFIXED_NAME
```

```
|RDF_TYPE # RDF_TYPE
```

```
;
```

```
prefixedName : PNAME_LN |PNAME_NS ;
```

```
blankNode : BLANK_NODE_LABEL ;
```

```
codeDecl : '%' codeLabel? iri? CODE? ;
```

```
codeLabel : UCASE_LABEL ;
```

```
startActions : codeDecl+ ;
```

```
semanticActions : codeDecl* ;
```