M2 – Data Science

Paris Saclay University

Semantic Web and Ontologies
F. Saïs (sais@lri.fr)

Practical session: RDF Data Validation

Intro to SHACL

SHACL[1] is a W3C recommendation aiming at validation of RDF data using so called shapes. Shapes are expressed in RDF, e.g.:

```
:MarriedManShape
    a sh:NodeShape ;
      sh:targetClass :MarriedMan ;
      sh:property [
            sh:path :hasWife ;
            sh:minCount 1 ;
            sh:maxCount 1 ;
     ].
```

Shapes are class-centric. Here, we define a shape for the RDFS class :MarriedMan. This shape checks that each instance of this class is explicitely related to exactly one other instance through the property :hasWife. Validating the RDF snippet

```
:John a :MarriedMan .
```

against the shape produces a validation error, as :John has no explicitely stated :hasWife relation, while validating the RDF snippet

```
:John a :MarriedMan ;
         :hasWife :Sue .
```

against the shape passes. You can test both examples e.g. at http://shacl.org/playground/.

Exercise 1 :

The following SHACL constraints (code 1) can be used for validating the RDF data graph (code2). However, there **are three bugs in the shapes**, find them and correct them.

Then test the correct version on http://shacl.org/playground/.

[1] https://www.w3.org/TR/shacl/

Code 1 : SHACL Constraints

```
@prefix dash: <http://datashapes.org/dash#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix schema: <http://schema.org/> .
@prefix sh: <http://www.w3.org/ns/shacl#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .

schema:PersonShape
   a sh:NodeShape ;
   sh:targetClass schema:Person ;
   sh:property [
      sh:path schema:givenName ;
      sh:datatype xsd:string ;
      sh:name "given name" ;
   ] ;
   sh:property [
      sh:path schema:birthDate ;
      sh:lessThan schema:deathDate ;
      sh:maxCount 1 ;
   ] ;
   sh:property [
      sh:path schema:gender ;
      sh:in ( "female" "male" ) ;
   ] ;
   sh:property [
      sh:path schema:address ;
      sh:node schema:AddressShape ;
   ] .

schema:AddressShape
   a sh:NodeShape ;
   sh:closed true ;
   sh:property [
      sh:path schema:streetAddress ;
      sh:datatype xsd:string ;
   ] ;
   sh:property [
      sh:path schema:postalCode ;
      sh:or ( [ sh:datatype xsd:string ] [ sh:datatype xsd:integer ] ) ;
      sh:minInclusive 10000 ;
      sh:maxInclusive 99999 ;
   ] .
```

Code 2 : RDF Graph

```
@prefix ex: <http://example.org/ns#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix schema: <http://schema.org/> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .

ex:Bob
   a schema:Person ;
   schema:givenName "Martin" ;
   schema:familyName "Dupont" ;
   schema:gender "m" ;
   schema:birthDate "1980-07-07"^^xsd:date ;
   schema:deathDate "1968-09-10"^^xsd:date ;
   schema:address ex:BobsAddress .

ex:BobsAddress
   schema:streetAddress "1600 Amphitheatre Pkway" ;
   schema:postalCode 9404 .
```

Exercise 2 (to deliver and marked)

Consider the ontology you created in Tab 2 on **master regulation.**

1- Open it on Protégé editor
2- Use SHACL editor tab (go to window) in Protégé to define SHACL constraints that allow to validate instances of this ontology.
3- For each class shape create at least one instance that violates a constraint you defined
4- Give the correct version of the instances violating the constraints.