B4M36OSW – Short Test 1

12. 11. 2020

Name:

Score: / 5

Download the input data from `Files/Tests/Data/2020-11-12/input-data.ttl` and import them to your GraphDB repository. Version shown here has shortend coordinates and lowered precision:

```
@prefix geo:  <http://www.opengis.net/ont/geosparql#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix ex:   <http://onto.fel.cvut.cz/ontologies/test/2020-11-12/> .
@prefix skos: <http://www.w3.org/2004/02/skos/core#> .
@prefix sf: <http://www.opengis.net/ont/sf#> .

ex:Prague a geo:Feature ;
      skos:prefLabel "Prague"@en, "Praha"@cs;
      geo:hasGeometry ex:PraguePolygon, ex:PragueReferencePoint .


ex:PraguePolygon a sf:Polygon, geo:Geometry ;
      rdfs:label "Geometrie Prahy vyjádřená polygonem" ;
      geo:asWKT "<http://www.opengis.net/def/crs/OGC/1.3/CRS84>
                 POLYGON((14.22 50.10, ... ,
                 14.31 50.11, 14.31 50.13,
                 14.22 50.10))"^^geo:wktLiteral .

ex:PragueReferencePoint a sf:Point, geo:Geometry ;
      rdfs:label "Geometrie definičního bodu Prahy" ;
      geo:asWKT "<http://www.opengis.net/def/crs/OGC/1.3/CRS84>
                 POINT(14.448 50.079)"^^geo:wktLiteral .

ex:Brno a geo:Feature ;
      skos:prefLabel "Brno"@cs;
      geo:hasGeometry ex:BrnoReferencePoint .

ex:BrnoReferencePoint a sf:Point, geo:Geometry ;
      rdfs:label "Geometrie definičního bodu Brna" ;
      geo:asWKT "<http://www.opengis.net/def/crs/OGC/1.3/CRS84>
                 POINT(16.610 49.198)"^^geo:wktLiteral .
```

1. (2 points) Use GeoSPARQL (do not forget to enable GeoSPARQL plugin in GraphDB) to compute the distance between Prague and Brno in kilometers. Use point geometry of each feature.

2. (2 points) Write a SHACL constraints to validate GeoSPARQL ontology rules. Include shape constrains to validate geo:Feature and geo:Geometry, while:

- geo:Feature must have at least one geo:hasGeometry property and its value must have geo:Geometry type,
- geo:Geometry must have at least one geo:asWKT property with geo:wktLiteral datatype.

Validate input data in https://shacl.org/playground/ against your SHACL, it shall pass.

3. (1 point) Download data from Files/Tests/Data/2020-11-12/gml-example.ttl. They are very similar to the original data, only ex:BrnoReferencePoint has now geo:asGML property with value of type geo:gmlLiteral. Validate it against SHACL you have created in previous task. It shall not pass, because geo:Geometry must have geo:asWKT with geo:wktLiteral.

   Rewrite the SHACL to make it pass:

   - geo:Geometry must have either geo:asWKT property with value of type geo:wktLiteral, or geo:asGML property with value of type geo:gmlLiteral.

   HINT: look at the sh:or property in SHACL.