UNIVERSITÀ
DEGLI STUDI
FIRENZE

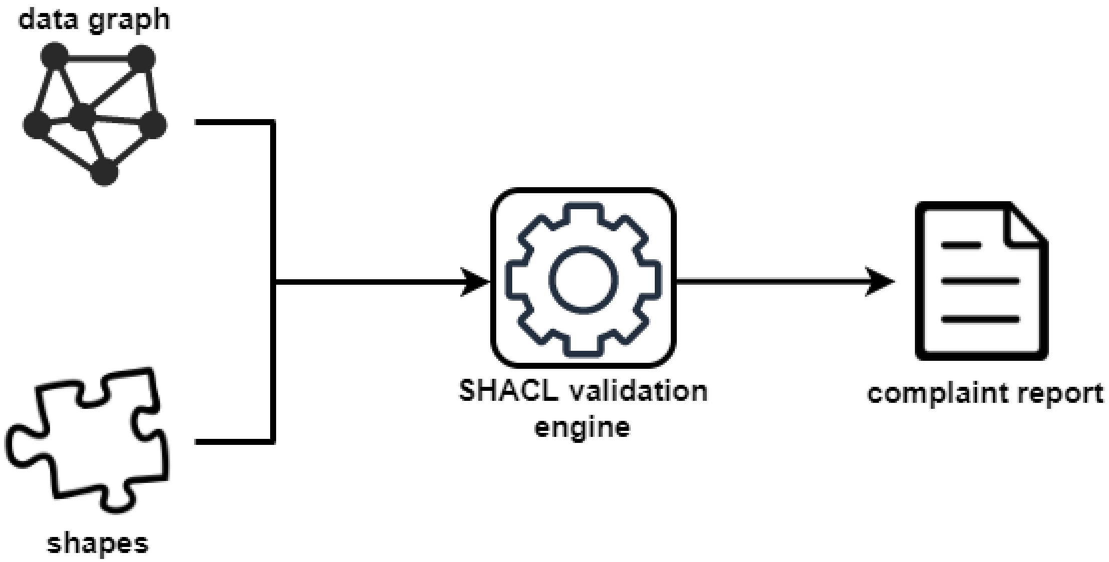
Evaluation and analysis of SHACL support by validation tools

Cosimo Giani

Firenze, 2021

SHACL

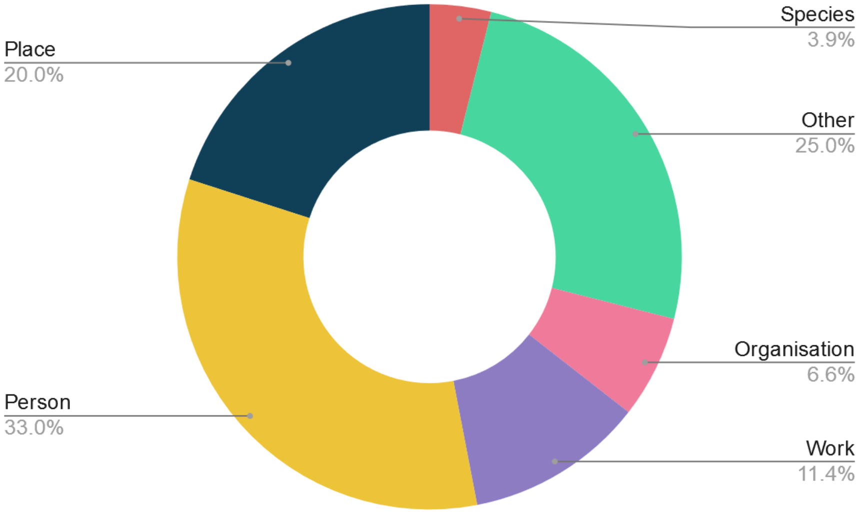
- **SHACL** is a standard provided by W3C consortium that stands for **Sha**pe **C**onstraint **L**anguage and is a specific for validating graph-based data against a set of conditions – called *shapes*.



Dataset and shape construction

- For the **dataset** used during the evaluation of the SHACL support was used **DBpedia[1].**
- For evaluation purposes, several subsets of different sizes of this graph have been created.

- The **shapes**, which declare constraints upon the given data from the data graph, were created following the main occurences for each DBpedia classes.



[1] DBpedia: *https://www.dbpedia.org/*

Tools

- The tools selected for the SHACL validation are:
  - **TopBraid**[2] by TopQuadrant
  - **RDF4J**[3] by Eclipse Foundation
  - **Neosemantics**[4] (Neo4J) by Neo Technology

[2] TopBraid: *https://github.com/TopQuadrant/shacl*
[3] RDF4J: *https://rdf4j.org*
[4] Neosemantics: *https://neo4j.com/labs/neosemantics/*

Implementation details

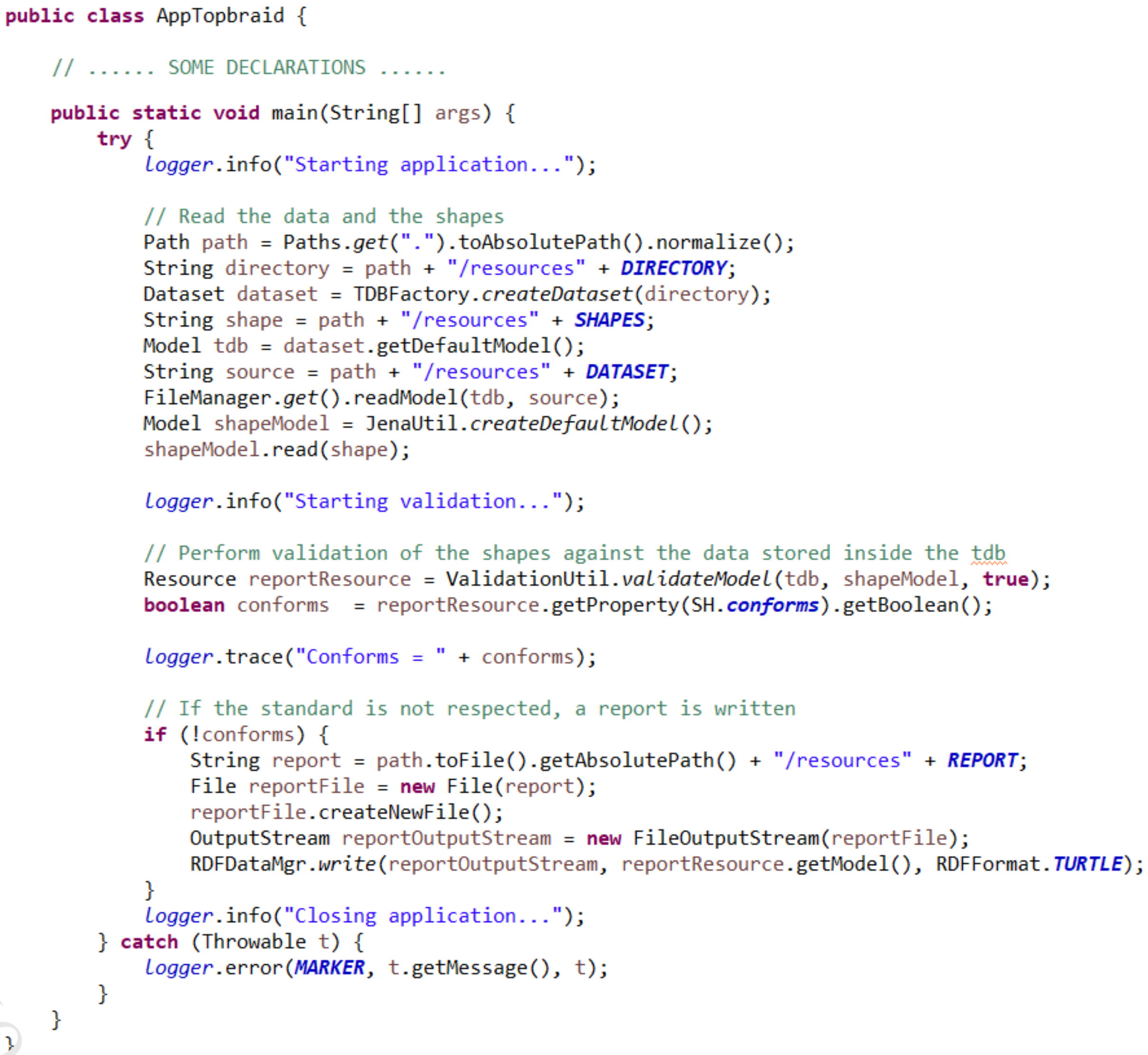
- **TopBraid framework**

– TopBraid is a solution based on the Apache Jena[5] framework that was used in its form of API to validate data contained within a triplestore.

– The triplestore was implemented with **Jena TDB**, a component of Jena for RDF storage and query.

[5] Apache Jena: *https://jena.apache.org*

Implementation details



```
public class AppTopbraid {

    // ...... SOME DECLARATIONS ......

    public static void main(String[] args) {
        try {
            logger.info("Starting application...");

            // Read the data and the shapes
            Path path = Paths.get(".").toAbsolutePath().normalize();
            String directory = path + "/resources" + DIRECTORY;
            Dataset dataset = TDBFactory.createDataset(directory);
            String shape = path + "/resources" + SHAPES;
            Model tdb = dataset.getDefaultModel();
            String source = path + "/resources" + DATASET;
            FileManager.get().readModel(tdb, source);
            Model shapeModel = JenaUtil.createDefaultModel();
            shapeModel.read(shape);

            logger.info("Starting validation...");

            // Perform validation of the shapes against the data stored inside the tdb
            Resource reportResource = ValidationUtil.validateModel(tdb, shapeModel, true);
            boolean conforms  = reportResource.getProperty(SH.conforms).getBoolean();

            logger.trace("Conforms = " + conforms);

            // If the standard is not respected, a report is written
            if (!conforms) {
                String report = path.toFile().getAbsolutePath() + "/resources" + REPORT;
                File reportFile = new File(report);
                reportFile.createNewFile();
                OutputStream reportOutputStream = new FileOutputStream(reportFile);
                RDFDataMgr.write(reportOutputStream, reportResource.getModel(), RDFFormat.TURTLE);
            }
            logger.info("Closing application...");
        } catch (Throwable t) {
            logger.error(MARKER, t.getMessage(), t);
        }
    }
}
```

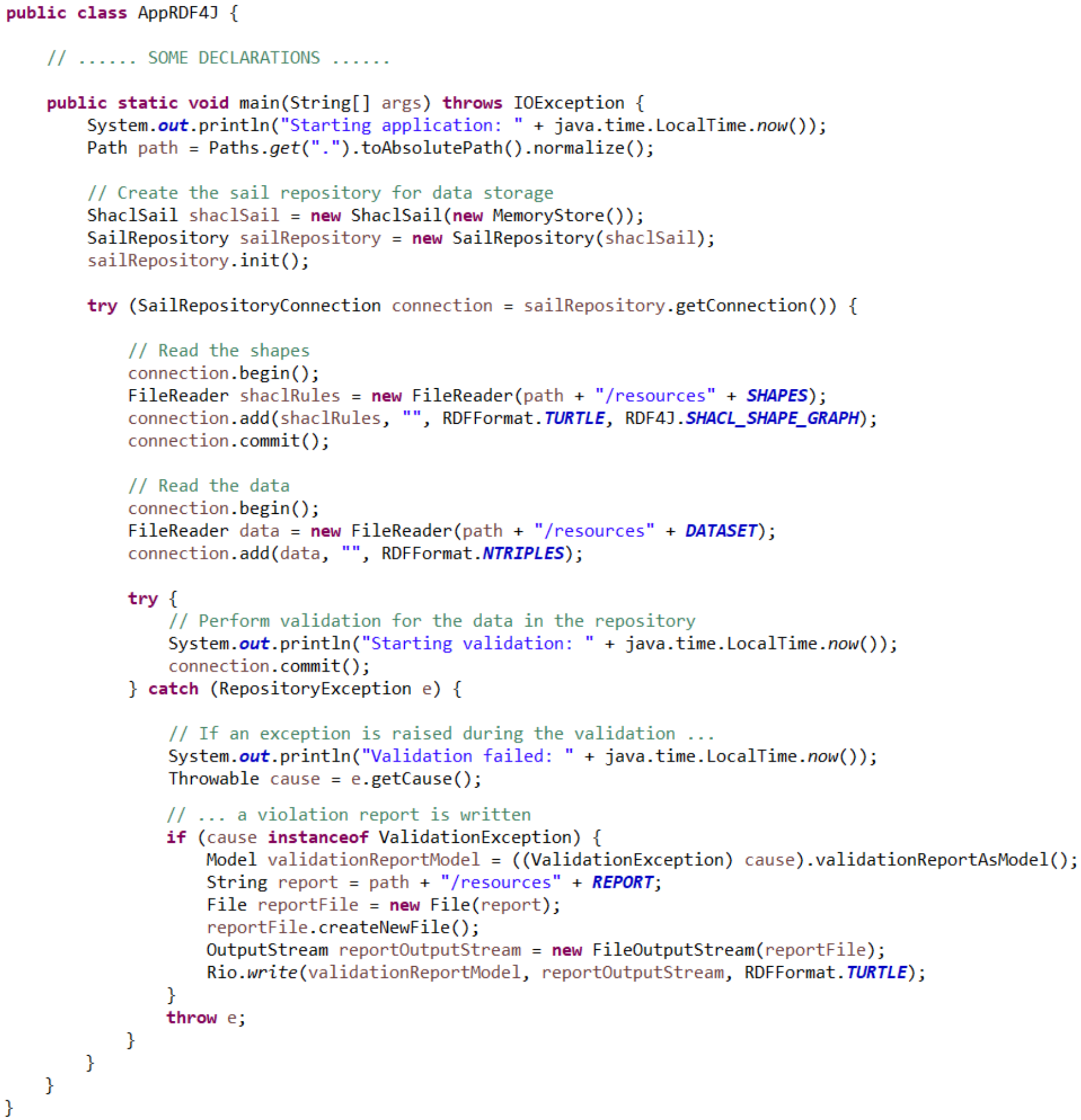
Implementation details

- **RDF4J framework**

– This approach uses a triplestore called **SailRepository**[6], which is a repository that operates directly on top of a **Sail**, i.e. a particular database.

– The application connects to the *SailRepository,* loads the SHACL shapes and perform the validation in a transactional manner.

Implementation details



```
public class AppRDF4J {

    // ...... SOME DECLARATIONS ......

    public static void main(String[] args) throws IOException {
        System.out.println("Starting application: " + java.time.LocalTime.now());
        Path path = Paths.get(".").toAbsolutePath().normalize();

        // Create the sail repository for data storage
        ShaclSail shaclSail = new ShaclSail(new MemoryStore());
        SailRepository sailRepository = new SailRepository(shaclSail);
        sailRepository.init();

        try (SailRepositoryConnection connection = sailRepository.getConnection()) {

            // Read the shapes
            connection.begin();
            FileReader shaclRules = new FileReader(path + "/resources" + SHAPES);
            connection.add(shaclRules, "", RDFFormat.TURTLE, RDF4J.SHACL_SHAPE_GRAPH);
            connection.commit();

            // Read the data
            connection.begin();
            FileReader data = new FileReader(path + "/resources" + DATASET);
            connection.add(data, "", RDFFormat.NTRIPLES);

            try {
                // Perform validation for the data in the repository
                System.out.println("Starting validation: " + java.time.LocalTime.now());
                connection.commit();
            } catch (RepositoryException e) {

                // If an exception is raised during the validation ...
                System.out.println("Validation failed: " + java.time.LocalTime.now());
                Throwable cause = e.getCause();

                // ... a violation report is written
                if (cause instanceof ValidationException) {
                    Model validationReportModel = ((ValidationException) cause).validationReportAsModel();
                    String report = path + "/resources" + REPORT;
                    File reportFile = new File(report);
                    reportFile.createNewFile();
                    OutputStream reportOutputStream = new FileOutputStream(reportFile);
                    Rio.write(validationReportModel, reportOutputStream, RDFFormat.TURTLE);
                }
                throw e;
            }
        }
    }
}
```

Implementation details

- **Neosemantics**

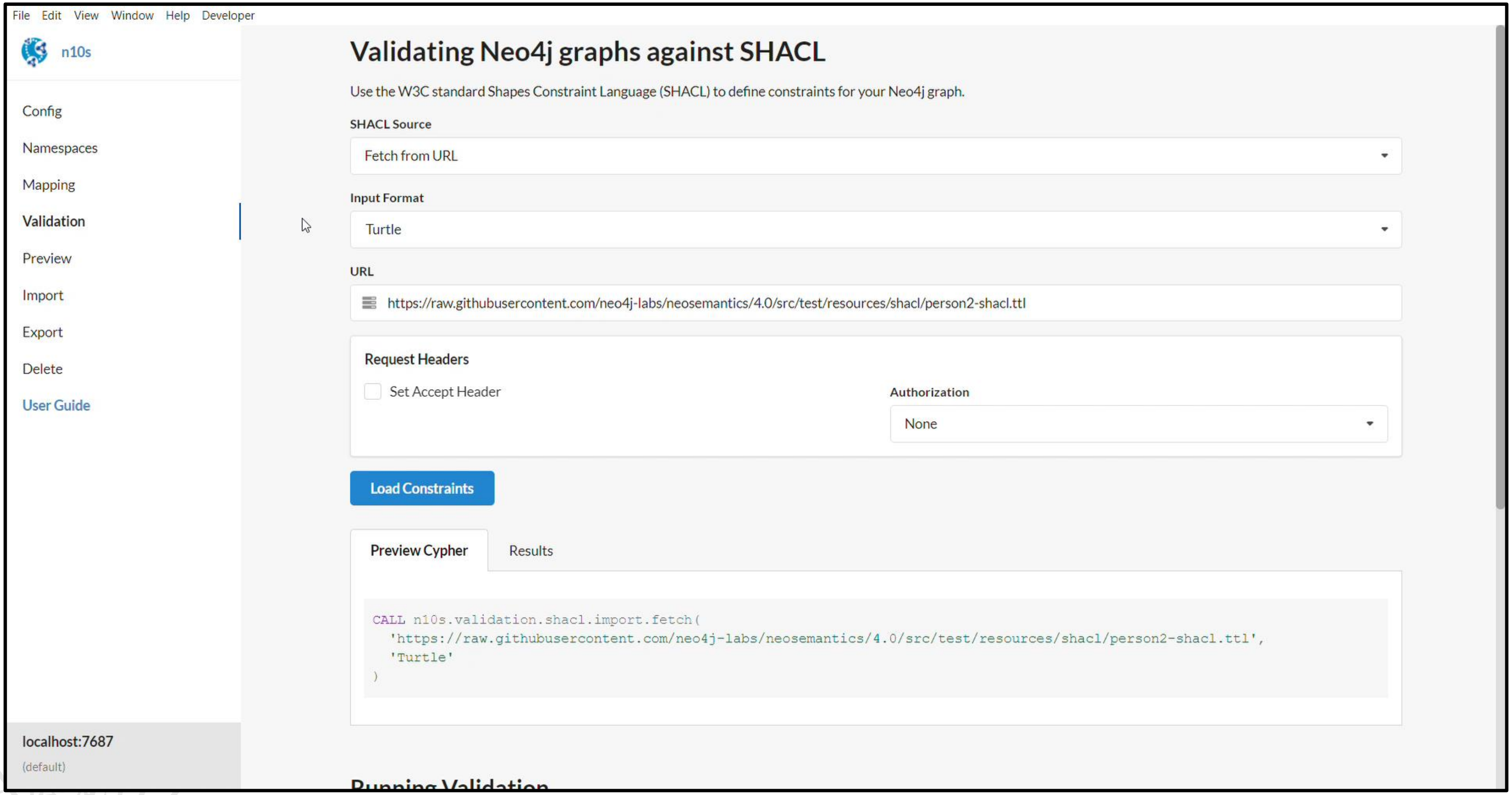
– Neosemantics is a plugin of the database management system Neo4J.

– For data storage is necessary to create a local database. It was used in its version 4.2.5.

– Two ways of performing validation:

1. **Neo4J Browser UI**
2. **GUI** or **GraphApp** of the plugin itself

Implementation details



Validating Neo4j graphs against SHACL

Use the W3C standard Shapes Constraint Language (SHACL) to define constraints for your Neo4j graph.

SHACL Source: Fetch from URL

Input Format: Turtle

URL: https://raw.githubusercontent.com/neo4j-labs/neosemantics/4.0/src/test/resources/shacl/person2-shacl.ttl

Request Headers

□ Set Accept Header

Authorization: None

```
CALL n10s.validation.shacl.import.fetch(
  'https://raw.githubusercontent.com/neo4j-labs/neosemantics/4.0/src/test/resources/shacl/person2-shacl.ttl',
  'Turtle'
)
```

Running Validation

Video: Neosemantics plugin in action.

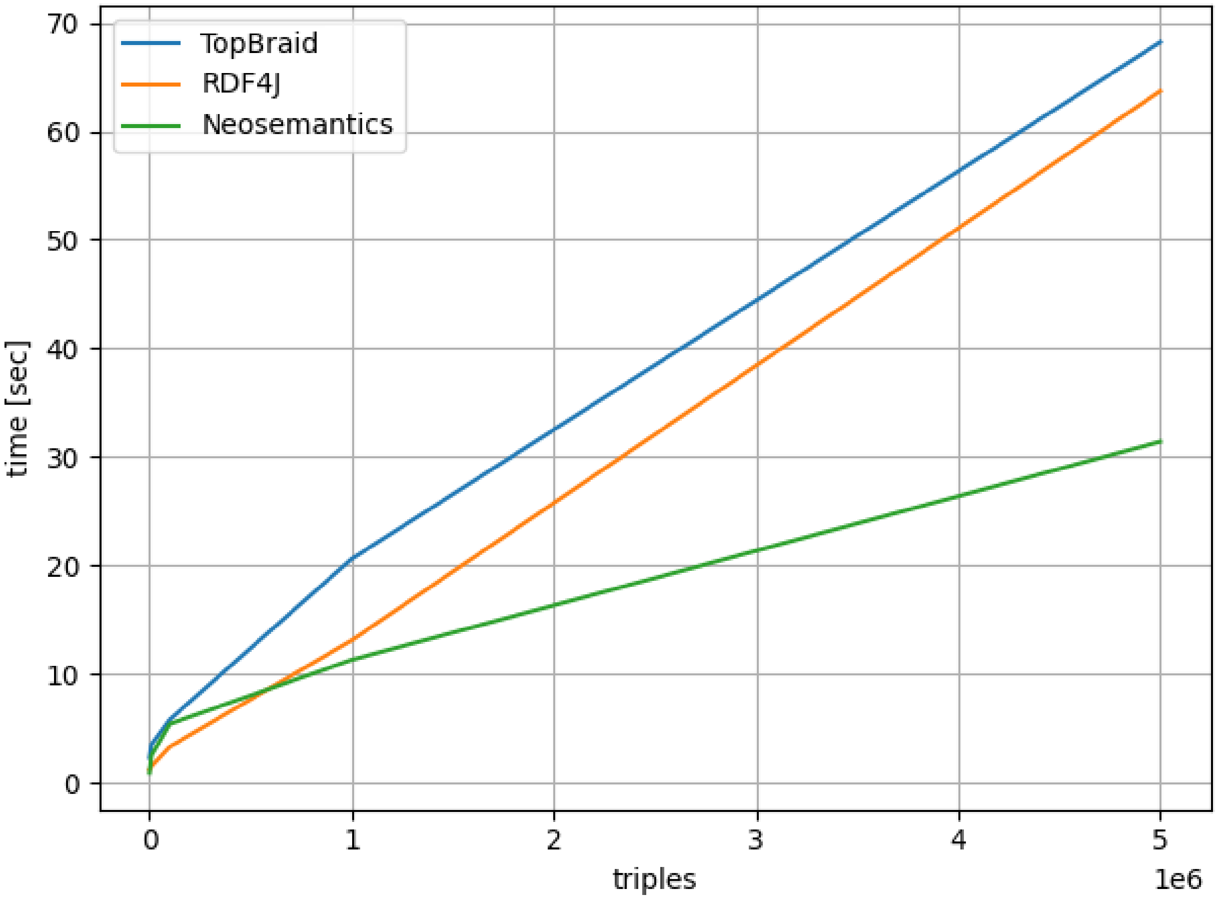
Experimental results

- To determine the quality of the SHACL support, two types of tests were performed:
  - **Average time**: computational time required and related memory use
  - **Feature support**: analysis of the constraint components support
- The experiments took place on a *Intel Core i7-8750H @ 2.20GHz* CPU and *16GB* of RAM.

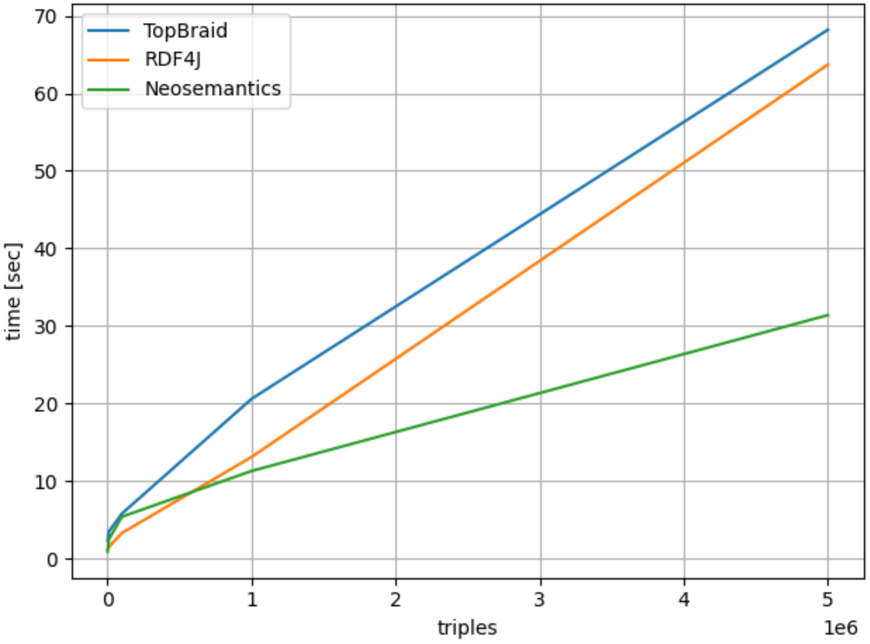
Average time

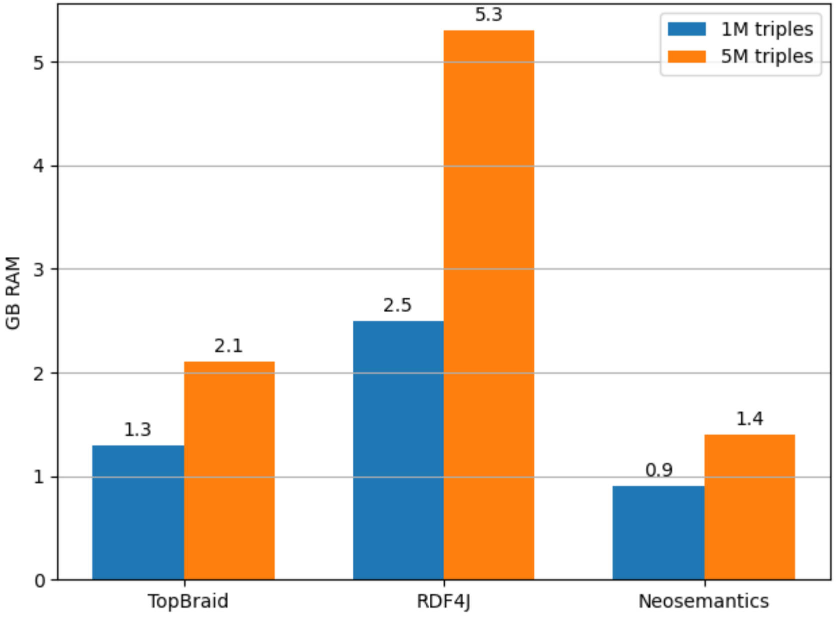
- Tests were performed with increasing fragments dataset: 1000, 10.000, 100.000, 1M and 5M triples.
- For each subset 10 measurements were carried out.
- It was also monitored the related amount of memory necessary for each tool.

Average time



Average time





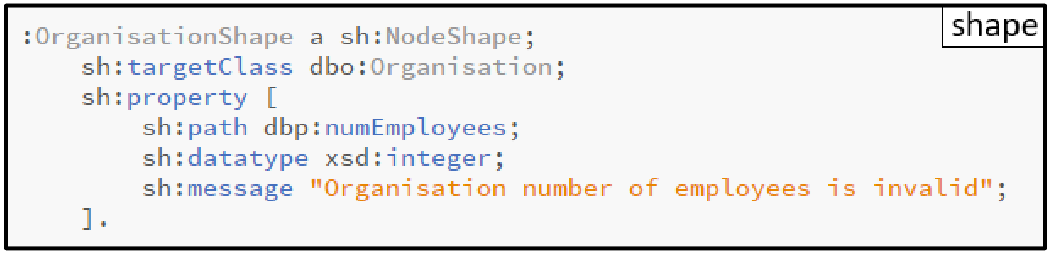
Feature support

- This type of tests have been performed with the aim to analyze the support of SHACL **features**, i.e. the constraints which the shapes are made of.

- The experiments consisted in piloted violations to verify the actual abilty of the tools in detecting and managing these features.

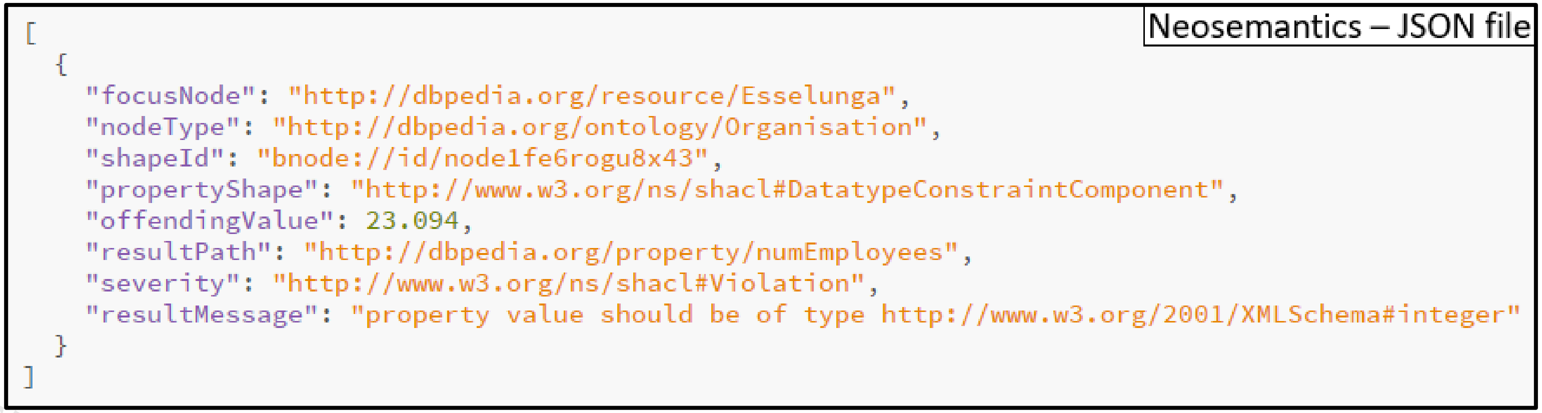
sh:datatype

- It is a *value type constraint* and it restricts the datatype of all value nodes to a given value.



shape

```
:OrganisationShape a sh:NodeShape;
    sh:targetClass dbo:Organisation;
    sh:property [
        sh:path dbp:numEmployees;
        sh:datatype xsd:integer;
        sh:message "Organisation number of employees is invalid";
    ].
```

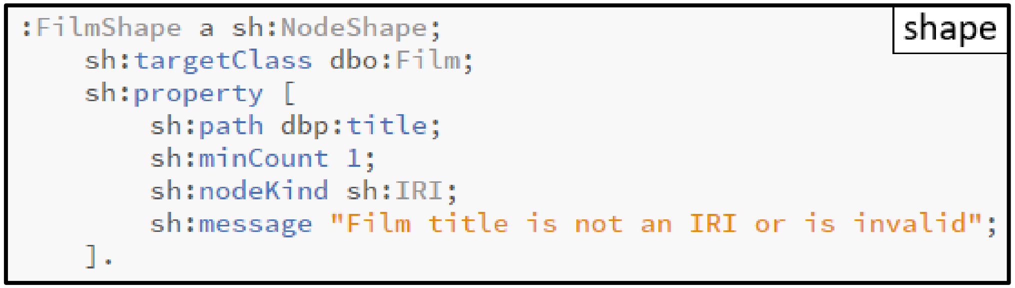


Neosemantics – JSON file

```
[
  {
    "focusNode": "http://dbpedia.org/resource/Esselunga",
    "nodeType": "http://dbpedia.org/ontology/Organisation",
    "shapeId": "bnode://id/node1fe6rogu8x43",
    "propertyShape": "http://www.w3.org/ns/shacl#DatatypeConstraintComponent",
    "offendingValue": 23.094,
    "resultPath": "http://dbpedia.org/property/numEmployees",
    "severity": "http://www.w3.org/ns/shacl#Violation",
    "resultMessage": "property value should be of type http://www.w3.org/2001/XMLSchema#integer"
  }
]
```

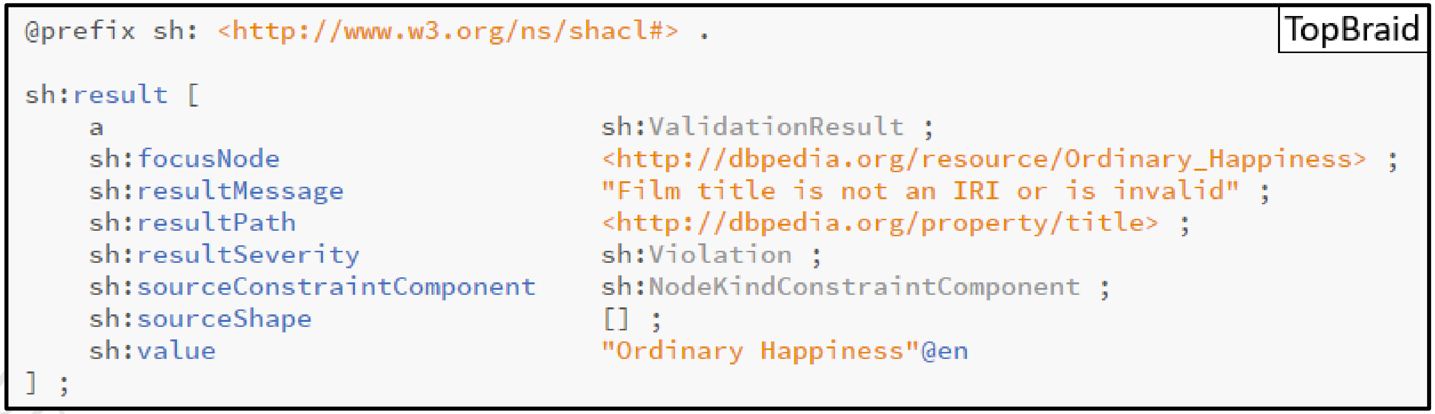
sh:nodeKind

- It is a *value type constraint* and it specifies a condition to be satisfied by the RDF node kind of each value node.



shape

```
:FilmShape a sh:NodeShape;
    sh:targetClass dbo:Film;
    sh:property [
        sh:path dbp:title;
        sh:minCount 1;
        sh:nodeKind sh:IRI;
        sh:message "Film title is not an IRI or is invalid";
    ].
```

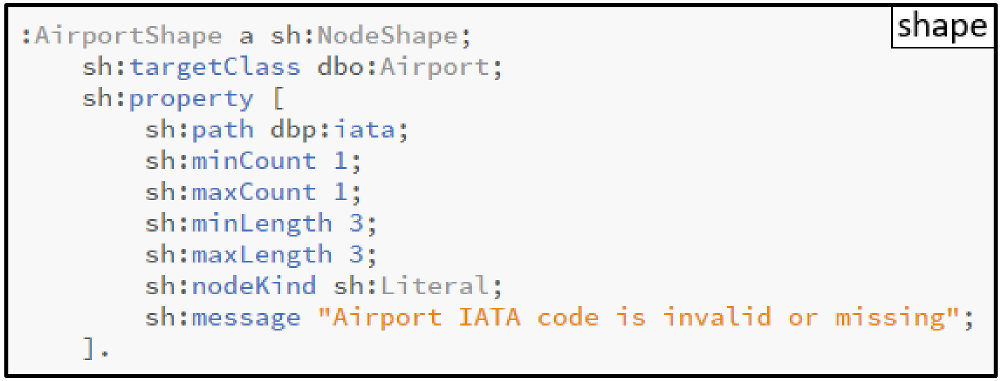


TopBraid

```
@prefix sh: <http://www.w3.org/ns/shacl#> .

sh:result [
    a                                sh:ValidationResult ;
    sh:focusNode                     <http://dbpedia.org/resource/Ordinary_Happiness> ;
    sh:resultMessage                 "Film title is not an IRI or is invalid" ;
    sh:resultPath                    <http://dbpedia.org/property/title> ;
    sh:resultSeverity                sh:Violation ;
    sh:sourceConstraintComponent     sh:NodeKindConstraintComponent ;
    sh:sourceShape                   [] ;
    sh:value                         "Ordinary Happiness"@en
] ;
```

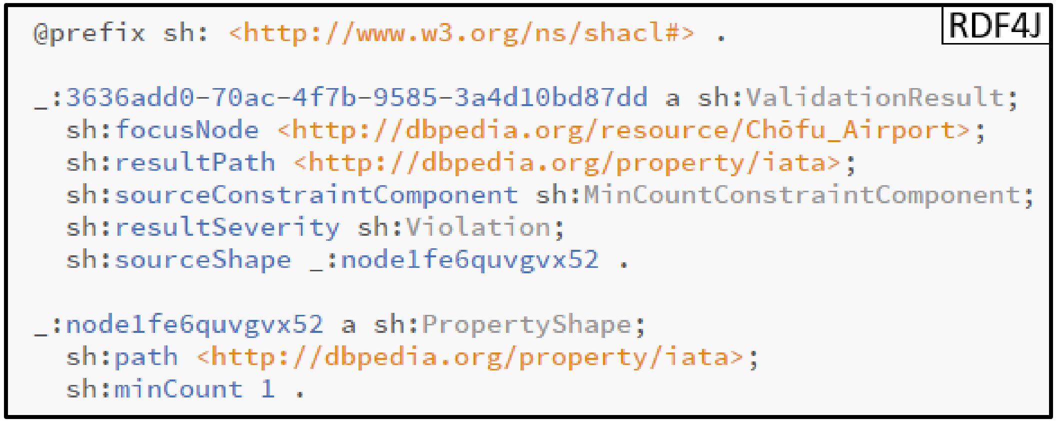
sh:minCount & sh:maxCount

- These are *cadinality constraints* and represent constrictions on the number of value nodes for the given focus node.



shape

```
:AirportShape a sh:NodeShape;
    sh:targetClass dbo:Airport;
    sh:property [
        sh:path dbp:iata;
        sh:minCount 1;
        sh:maxCount 1;
        sh:minLength 3;
        sh:maxLength 3;
        sh:nodeKind sh:Literal;
        sh:message "Airport IATA code is invalid or missing";
    ].
```

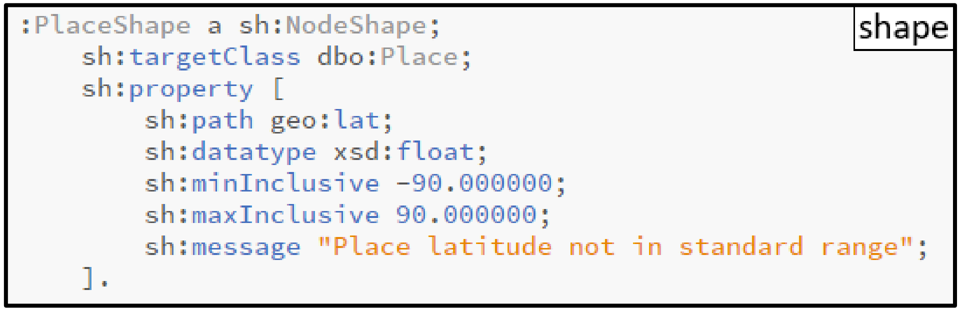


RDF4J

```
@prefix sh: <http://www.w3.org/ns/shacl#> .

_:3636add0-70ac-4f7b-9585-3a4d10bd87dd a sh:ValidationResult;
  sh:focusNode <http://dbpedia.org/resource/Chōfu_Airport>;
  sh:resultPath <http://dbpedia.org/property/iata>;
  sh:sourceConstraintComponent sh:MinCountConstraintComponent;
  sh:resultSeverity sh:Violation;
  sh:sourceShape _:node1fe6quvgvx52 .

_:node1fe6quvgvx52 a sh:PropertyShape;
  sh:path <http://dbpedia.org/property/iata>;
  sh:minCount 1 .
```

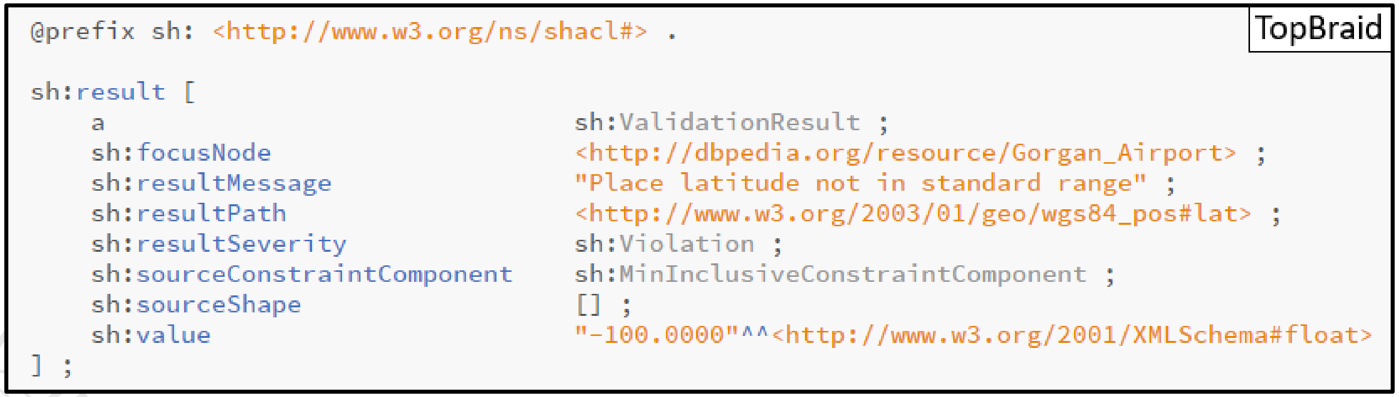
sh:min/max Inclusive & Exclusive

- These are *value range constraints* and specify value range conditions to be satisfied by comparable value nodes.



shape

```
:PlaceShape a sh:NodeShape;
    sh:targetClass dbo:Place;
    sh:property [
        sh:path geo:lat;
        sh:datatype xsd:float;
        sh:minInclusive -90.000000;
        sh:maxInclusive 90.000000;
        sh:message "Place latitude not in standard range";
    ].
```

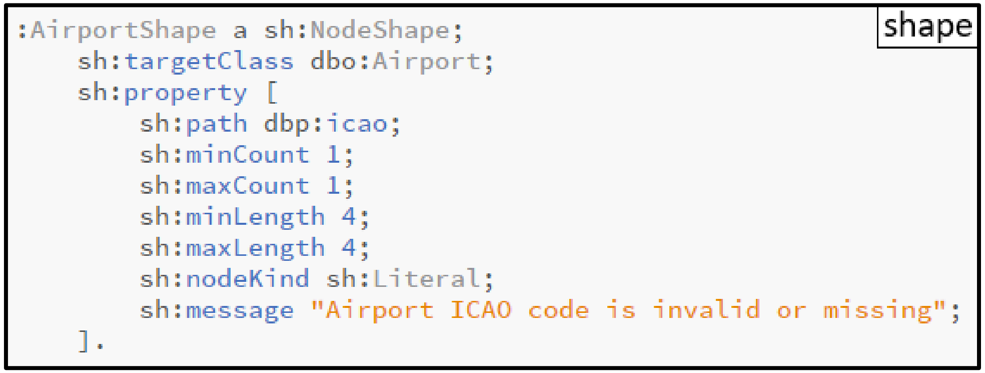


TopBraid

```
@prefix sh: <http://www.w3.org/ns/shacl#> .

sh:result [
    a                               sh:ValidationResult ;
    sh:focusNode                    <http://dbpedia.org/resource/Gorgan_Airport> ;
    sh:resultMessage                "Place latitude not in standard range" ;
    sh:resultPath                   <http://www.w3.org/2003/01/geo/wgs84_pos#lat> ;
    sh:resultSeverity               sh:Violation ;
    sh:sourceConstraintComponent    sh:MinInclusiveConstraintComponent ;
    sh:sourceShape                  [] ;
    sh:value                        "-100.0000"^^<http://www.w3.org/2001/XMLSchema#float>
] ;
```

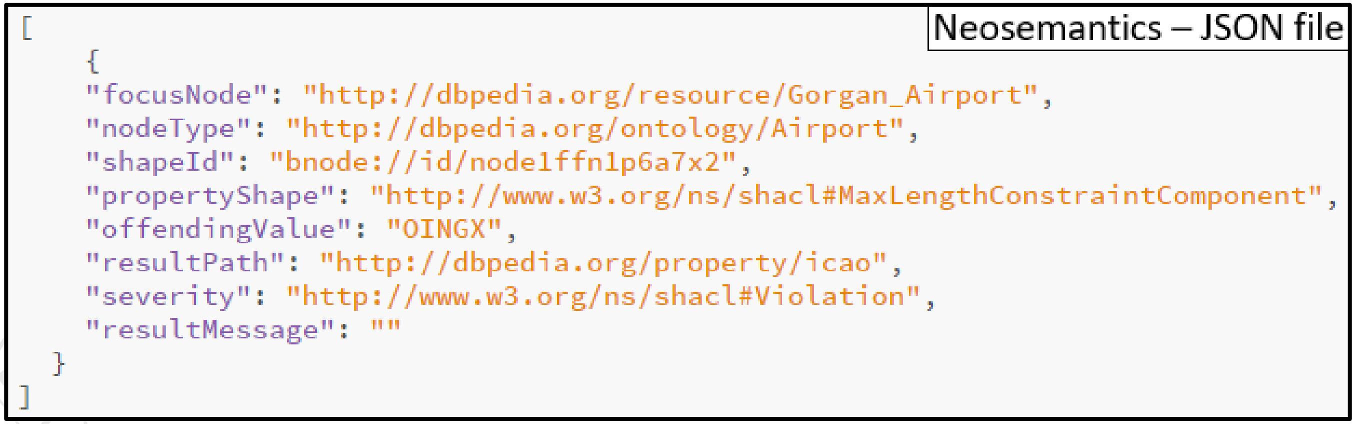
sh:minLength & sh:maxLength

- These are *string-based constraints* and specify the string lentgh of a value node.



shape

```
:AirportShape a sh:NodeShape;
    sh:targetClass dbo:Airport;
    sh:property [
        sh:path dbp:icao;
        sh:minCount 1;
        sh:maxCount 1;
        sh:minLength 4;
        sh:maxLength 4;
        sh:nodeKind sh:Literal;
        sh:message "Airport ICAO code is invalid or missing";
    ].
```

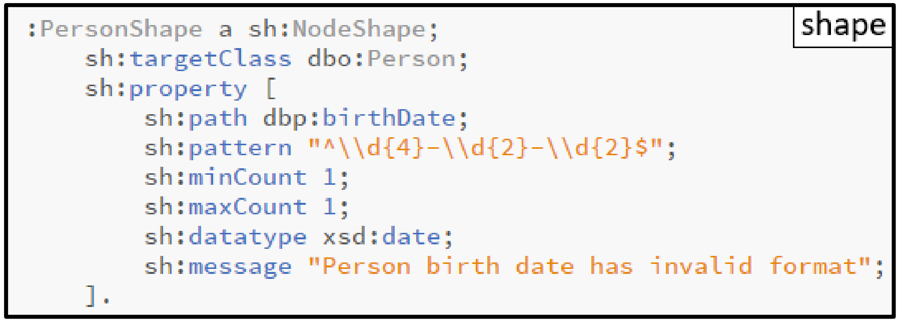


Neosemantics – JSON file

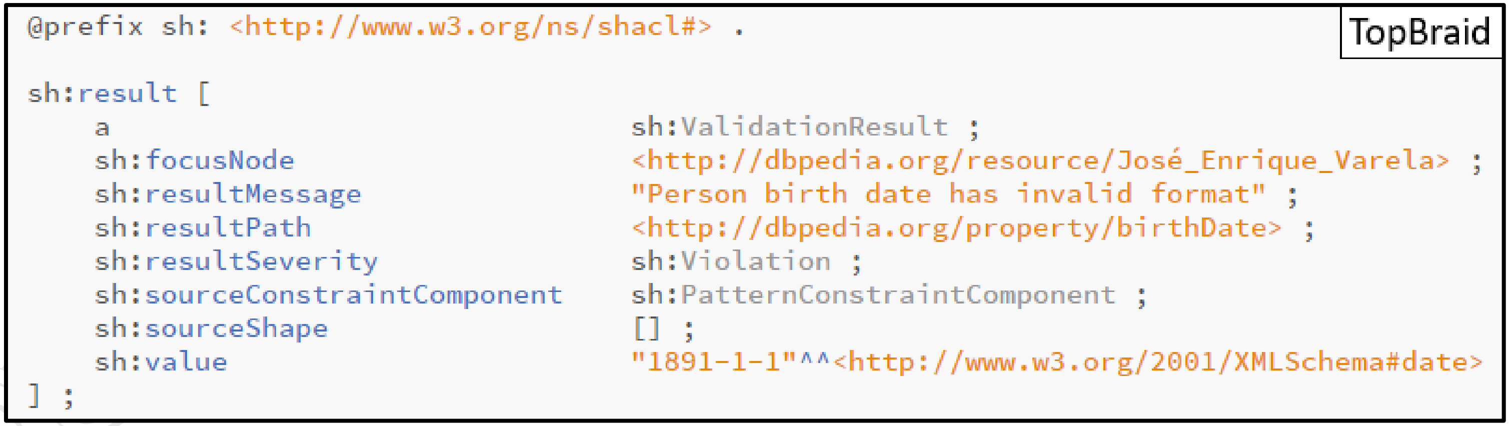
```
[
    {
    "focusNode": "http://dbpedia.org/resource/Gorgan_Airport",
    "nodeType": "http://dbpedia.org/ontology/Airport",
    "shapeId": "bnode://id/node1ffn1p6a7x2",
    "propertyShape": "http://www.w3.org/ns/shacl#MaxLengthConstraintComponent",
    "offendingValue": "OINGX",
    "resultPath": "http://dbpedia.org/property/icao",
    "severity": "http://www.w3.org/ns/shacl#Violation",
    "resultMessage": ""
  }
]
```

sh:pattern

- It is a *string-based constraint* and it specifies a regular expression that a value node need to match to satisfy the condition.



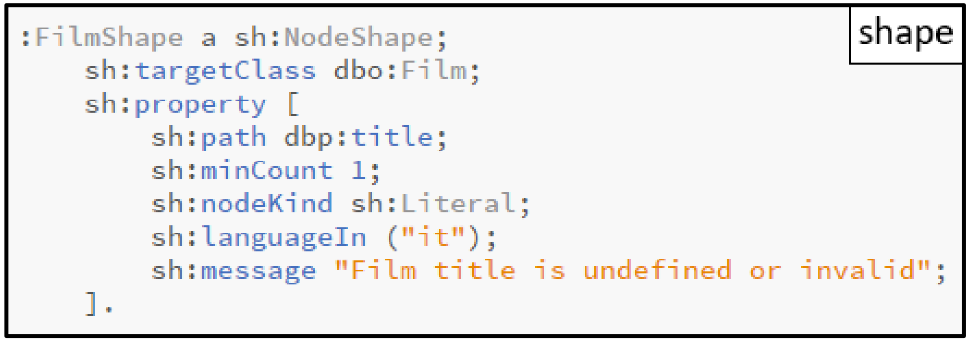
```
:PersonShape a sh:NodeShape;
    sh:targetClass dbo:Person;
    sh:property [
        sh:path dbp:birthDate;
        sh:pattern "^\\d{4}-\\d{2}-\\d{2}$";
        sh:minCount 1;
        sh:maxCount 1;
        sh:datatype xsd:date;
        sh:message "Person birth date has invalid format";
    ].
```



```
@prefix sh: <http://www.w3.org/ns/shacl#> .

sh:result [
    a                               sh:ValidationResult ;
    sh:focusNode                    <http://dbpedia.org/resource/José_Enrique_Varela> ;
    sh:resultMessage                "Person birth date has invalid format" ;
    sh:resultPath                   <http://dbpedia.org/property/birthDate> ;
    sh:resultSeverity               sh:Violation ;
    sh:sourceConstraintComponent    sh:PatternConstraintComponent ;
    sh:sourceShape                  [] ;
    sh:value                        "1891-1-1"^^<http://www.w3.org/2001/XMLSchema#date>
] ;
```

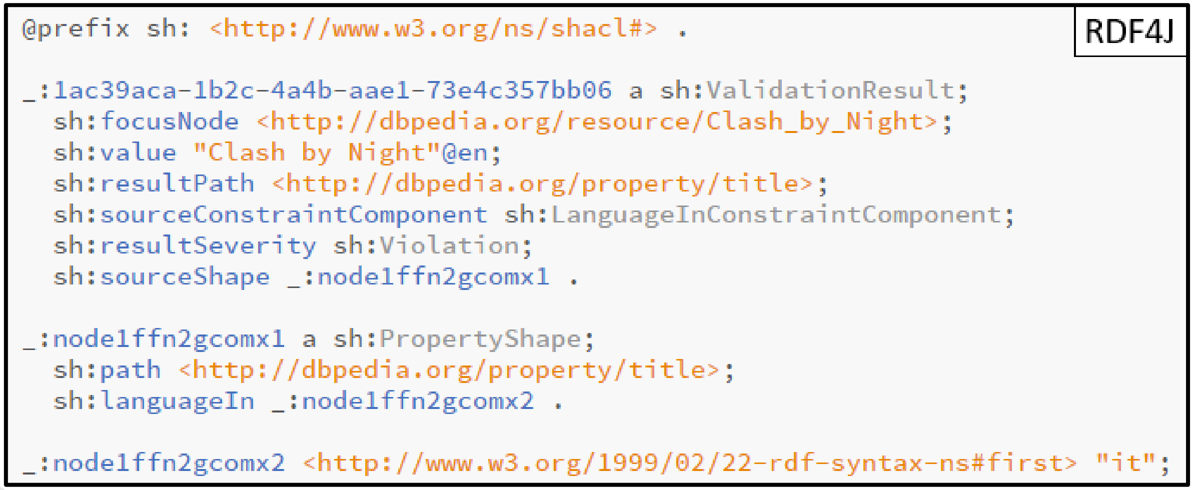
sh:languageIn

- It is a *string-based constraint* and it specifies the allowed language tags.



shape

```
:FilmShape a sh:NodeShape;
    sh:targetClass dbo:Film;
    sh:property [
        sh:path dbp:title;
        sh:minCount 1;
        sh:nodeKind sh:Literal;
        sh:languageIn ("it");
        sh:message "Film title is undefined or invalid";
    ].
```

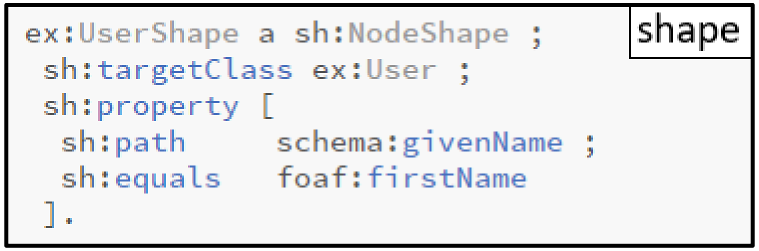


RDF4J

```
@prefix sh: <http://www.w3.org/ns/shacl#> .

_:1ac39aca-1b2c-4a4b-aae1-73e4c357bb06 a sh:ValidationResult;
  sh:focusNode <http://dbpedia.org/resource/Clash_by_Night>;
  sh:value "Clash by Night"@en;
  sh:resultPath <http://dbpedia.org/property/title>;
  sh:sourceConstraintComponent sh:LanguageInConstraintComponent;
  sh:resultSeverity sh:Violation;
  sh:sourceShape _:node1ffn2gcomx1 .

_:node1ffn2gcomx1 a sh:PropertyShape;
  sh:path <http://dbpedia.org/property/title>;
  sh:languageIn _:node1ffn2gcomx2 .

_:node1ffn2gcomx2 <http://www.w3.org/1999/02/22-rdf-syntax-ns#first> "it";
```

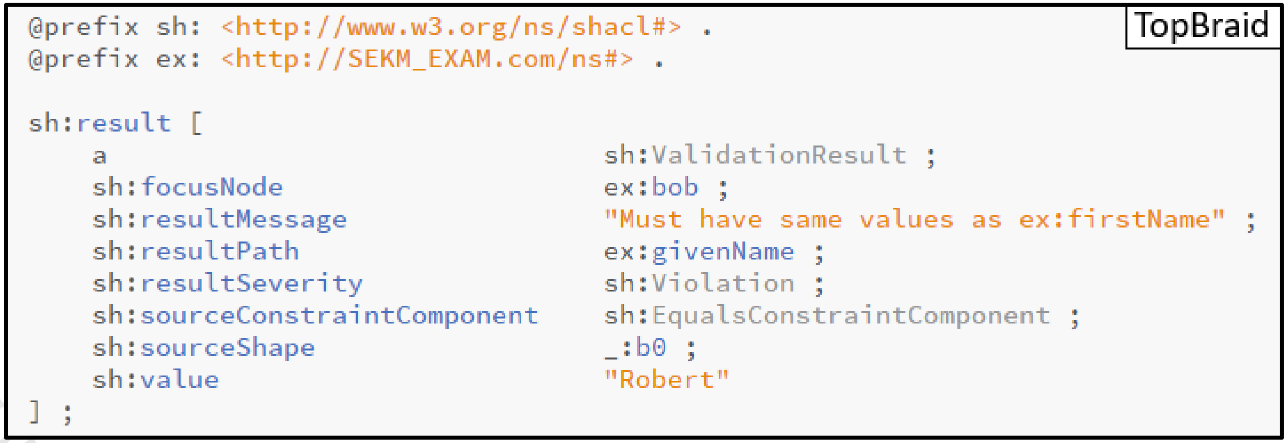
sh:equals

- It is a *property pair constraint* and it specifies the condition that the set of values of both properties at a given focus node must be **equal**.



shape

```
ex:UserShape a sh:NodeShape ;
 sh:targetClass ex:User ;
 sh:property [
  sh:path      schema:givenName ;
  sh:equals    foaf:firstName
 ].
```

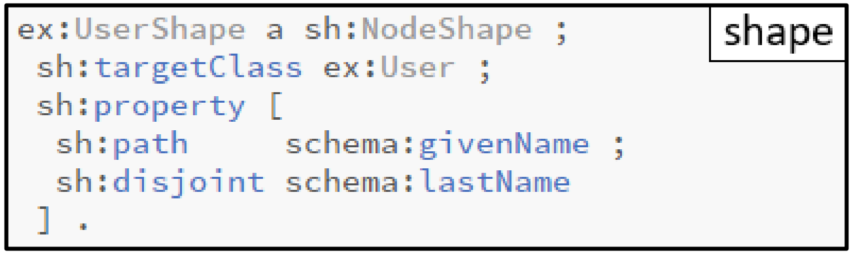


TopBraid

```
@prefix sh: <http://www.w3.org/ns/shacl#> .
@prefix ex: <http://SEKM_EXAM.com/ns#> .

sh:result [
    a                               sh:ValidationResult ;
    sh:focusNode                    ex:bob ;
    sh:resultMessage                "Must have same values as ex:firstName" ;
    sh:resultPath                   ex:givenName ;
    sh:resultSeverity               sh:Violation ;
    sh:sourceConstraintComponent    sh:EqualsConstraintComponent ;
    sh:sourceShape                  _:b0 ;
    sh:value                        "Robert"
] ;
```

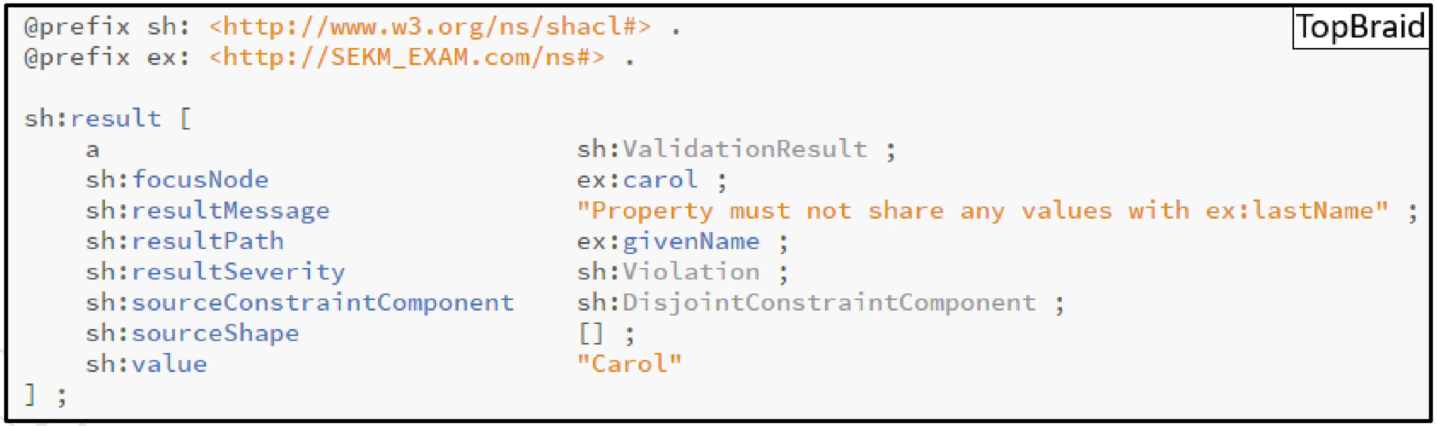
sh:disjoint

- It is a *property pair constraint* and it specifies the condition that the set of values of both properties at a given focus node must be **different**.



shape

```
ex:UserShape a sh:NodeShape ;
 sh:targetClass ex:User ;
 sh:property [
  sh:path     schema:givenName ;
  sh:disjoint schema:lastName
 ] .
```

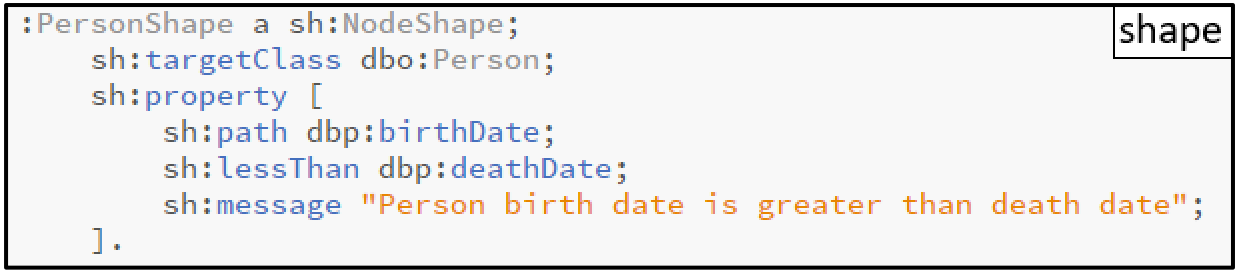


TopBraid

```
@prefix sh: <http://www.w3.org/ns/shacl#> .
@prefix ex: <http://SEKM_EXAM.com/ns#> .

sh:result [
    a                              sh:ValidationResult ;
    sh:focusNode                   ex:carol ;
    sh:resultMessage               "Property must not share any values with ex:lastName" ;
    sh:resultPath                  ex:givenName ;
    sh:resultSeverity              sh:Violation ;
    sh:sourceConstraintComponent   sh:DisjointConstraintComponent ;
    sh:sourceShape                 [] ;
    sh:value                       "Carol"
] ;
```

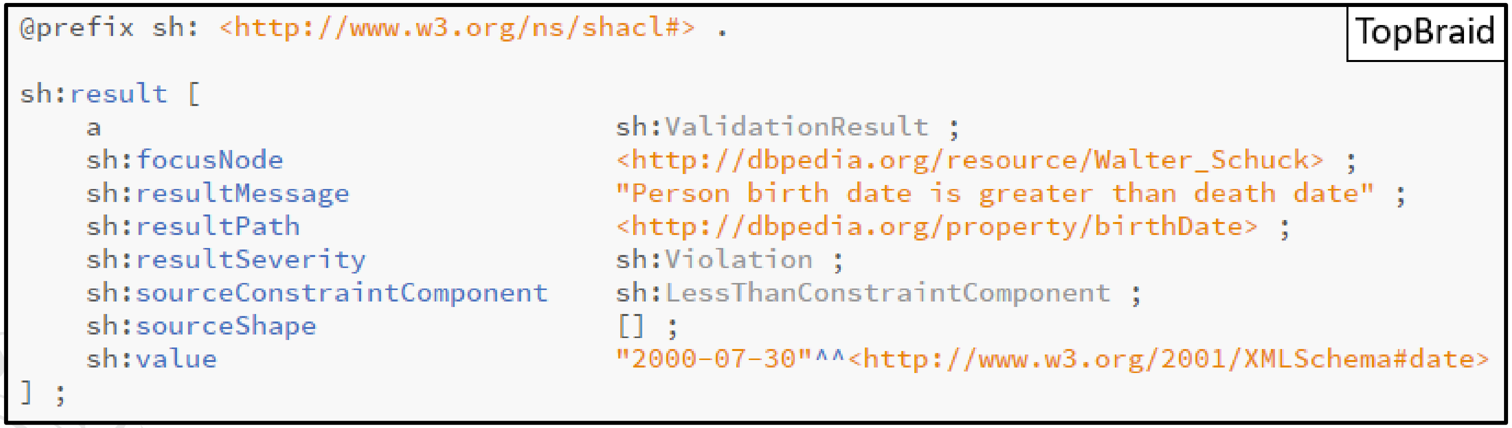
sh:lessThan

- It is a *property pair constraint* and it specifies the condition that the values must be smaller than the values of another property.



shape

```
:PersonShape a sh:NodeShape;
    sh:targetClass dbo:Person;
    sh:property [
        sh:path dbp:birthDate;
        sh:lessThan dbp:deathDate;
        sh:message "Person birth date is greater than death date";
    ].
```

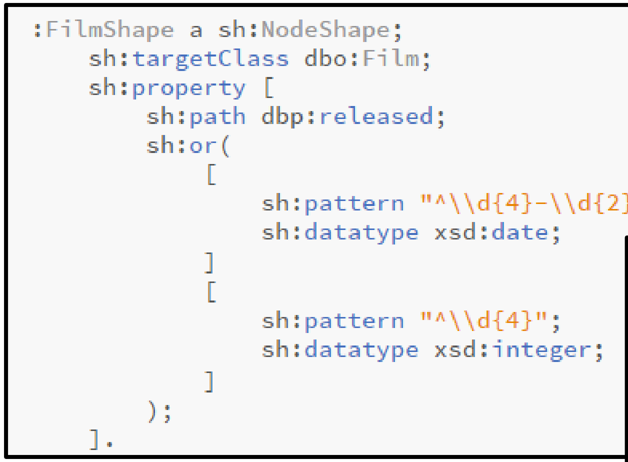


TopBraid

```
@prefix sh: <http://www.w3.org/ns/shacl#> .

sh:result [
    a                              sh:ValidationResult ;
    sh:focusNode                   <http://dbpedia.org/resource/Walter_Schuck> ;
    sh:resultMessage               "Person birth date is greater than death date" ;
    sh:resultPath                  <http://dbpedia.org/property/birthDate> ;
    sh:resultSeverity              sh:Violation ;
    sh:sourceConstraintComponent   sh:LessThanConstraintComponent ;
    sh:sourceShape                 [] ;
    sh:value                       "2000-07-30"^^<http://www.w3.org/2001/XMLSchema#date>
] ;
```

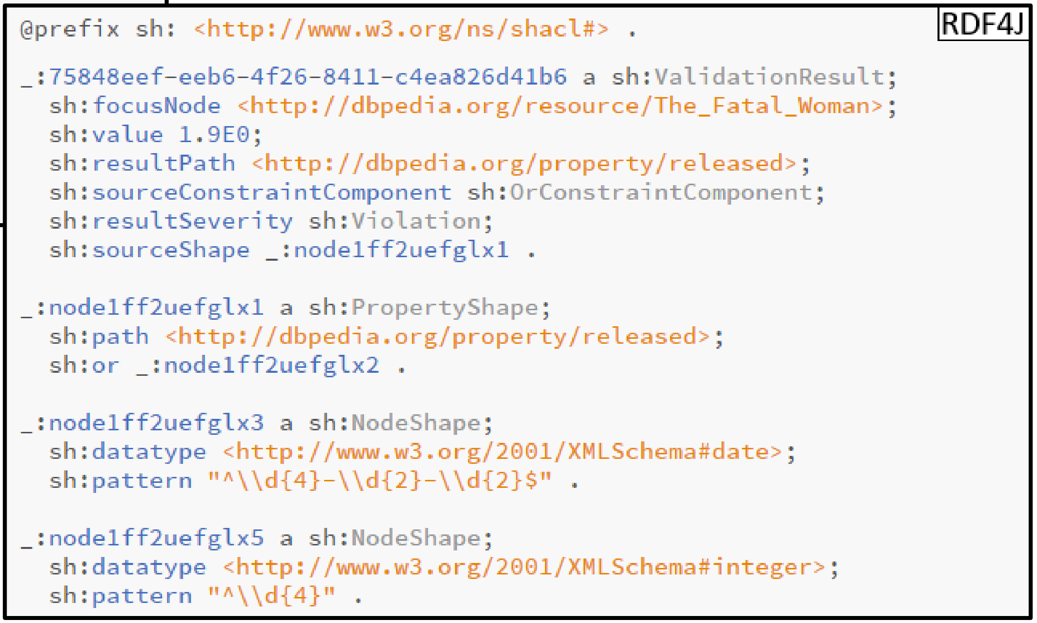
sh:or

- It is a *logical constraint* and it specifies the condition that each value node conforms to **at least one** of the provided shapes.



shape

```
:FilmShape a sh:NodeShape;
    sh:targetClass dbo:Film;
    sh:property [
        sh:path dbp:released;
        sh:or(
            [
                sh:pattern "^\\d{4}-\\d{2}-\\d{2}$";
                sh:datatype xsd:date;
            ]
            [
                sh:pattern "^\\d{4}";
                sh:datatype xsd:integer;
            ]
        );
    ].
```

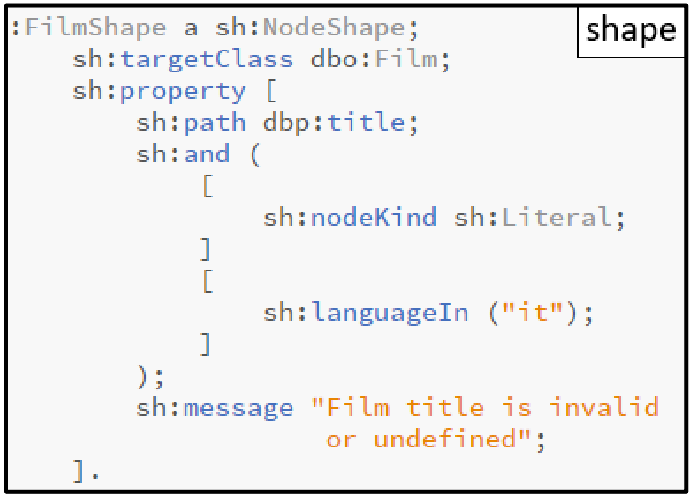


RDF4J

```
@prefix sh: <http://www.w3.org/ns/shacl#> .

_:75848eef-eeb6-4f26-8411-c4ea826d41b6 a sh:ValidationResult;
  sh:focusNode <http://dbpedia.org/resource/The_Fatal_Woman>;
  sh:value 1.9E0;
  sh:resultPath <http://dbpedia.org/property/released>;
  sh:sourceConstraintComponent sh:OrConstraintComponent;
  sh:resultSeverity sh:Violation;
  sh:sourceShape _:node1ff2uefglx1 .

_:node1ff2uefglx1 a sh:PropertyShape;
  sh:path <http://dbpedia.org/property/released>;
  sh:or _:node1ff2uefglx2 .

_:node1ff2uefglx3 a sh:NodeShape;
  sh:datatype <http://www.w3.org/2001/XMLSchema#date>;
  sh:pattern "^\\d{4}-\\d{2}-\\d{2}$" .

_:node1ff2uefglx5 a sh:NodeShape;
  sh:datatype <http://www.w3.org/2001/XMLSchema#integer>;
  sh:pattern "^\\d{4}" .
```

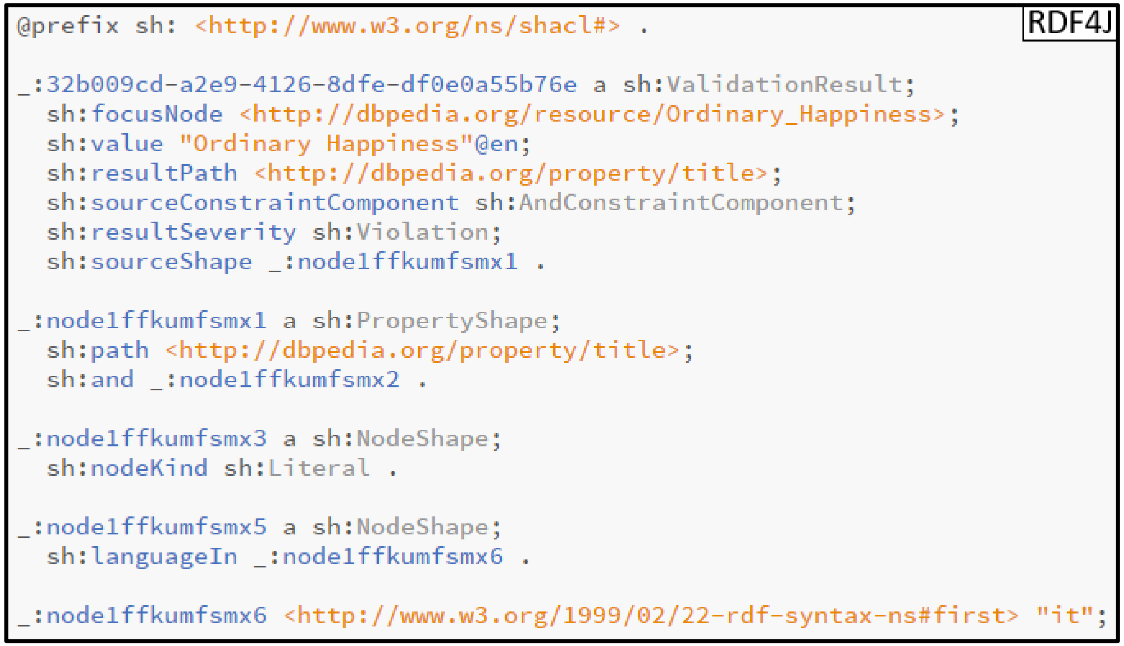
sh:and

- It is a *logical constraint* and it specifies the condition that each value node conforms to **all** the provided shapes.



shape

```
:FilmShape a sh:NodeShape;
    sh:targetClass dbo:Film;
    sh:property [
        sh:path dbp:title;
        sh:and (
            [
                sh:nodeKind sh:Literal;
            ]
            [
                sh:languageIn ("it");
            ]
        );
        sh:message "Film title is invalid
                    or undefined";
    ].
```

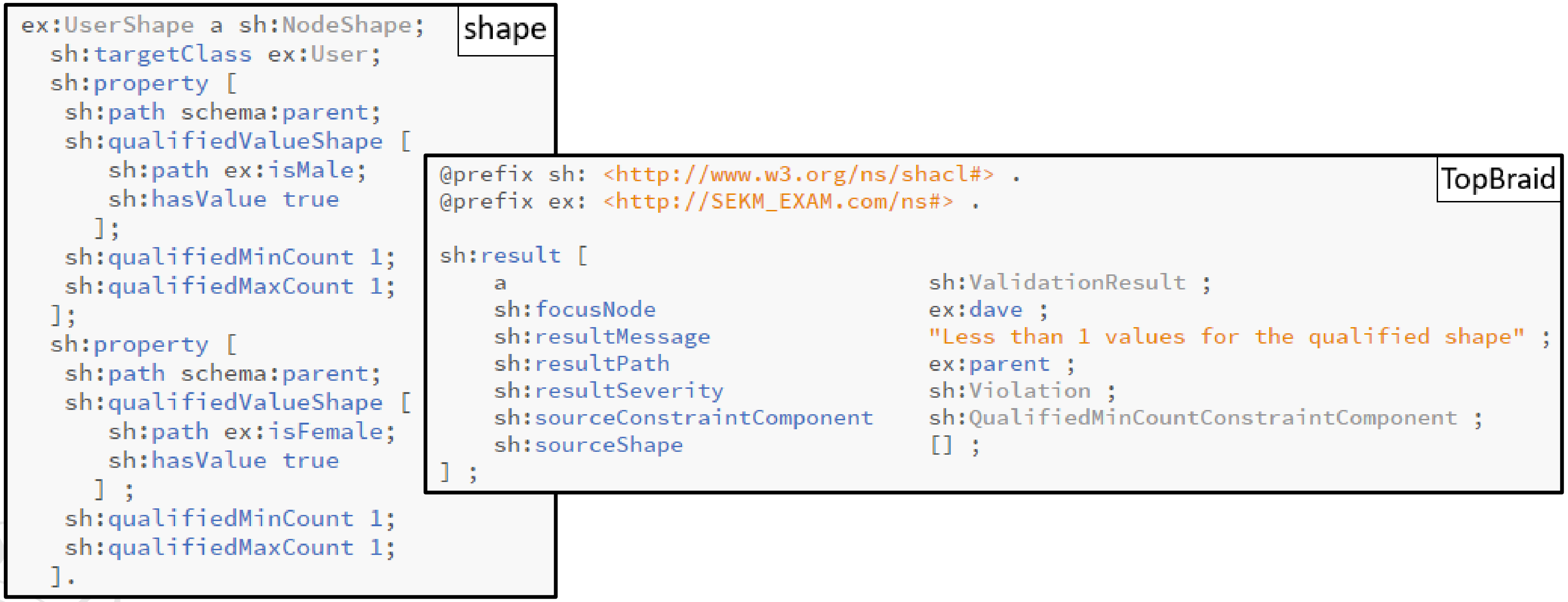


RDF4J

```
@prefix sh: <http://www.w3.org/ns/shacl#> .

_:32b009cd-a2e9-4126-8dfe-df0e0a55b76e a sh:ValidationResult;
  sh:focusNode <http://dbpedia.org/resource/Ordinary_Happiness>;
  sh:value "Ordinary Happiness"@en;
  sh:resultPath <http://dbpedia.org/property/title>;
  sh:sourceConstraintComponent sh:AndConstraintComponent;
  sh:resultSeverity sh:Violation;
  sh:sourceShape _:node1ffkumfsmx1 .

_:node1ffkumfsmx1 a sh:PropertyShape;
  sh:path <http://dbpedia.org/property/title>;
  sh:and _:node1ffkumfsmx2 .

_:node1ffkumfsmx3 a sh:NodeShape;
  sh:nodeKind sh:Literal .

_:node1ffkumfsmx5 a sh:NodeShape;
  sh:languageIn _:node1ffkumfsmx6 .

_:node1ffkumfsmx6 <http://www.w3.org/1999/02/22-rdf-syntax-ns#first> "it";
```

sh:qualifiedValueShape & sh:qualified(Min/Max)Count

- It is a *shape-based constraint* and it specifies the condition that a specified number of value nodes conforms to a given shape.



shape

```
ex:UserShape a sh:NodeShape;
  sh:targetClass ex:User;
  sh:property [
   sh:path schema:parent;
   sh:qualifiedValueShape [
      sh:path ex:isMale;
      sh:hasValue true
    ];
   sh:qualifiedMinCount 1;
   sh:qualifiedMaxCount 1;
  ];
  sh:property [
   sh:path schema:parent;
   sh:qualifiedValueShape [
      sh:path ex:isFemale;
      sh:hasValue true
    ] ;
   sh:qualifiedMinCount 1;
   sh:qualifiedMaxCount 1;
  ].
```

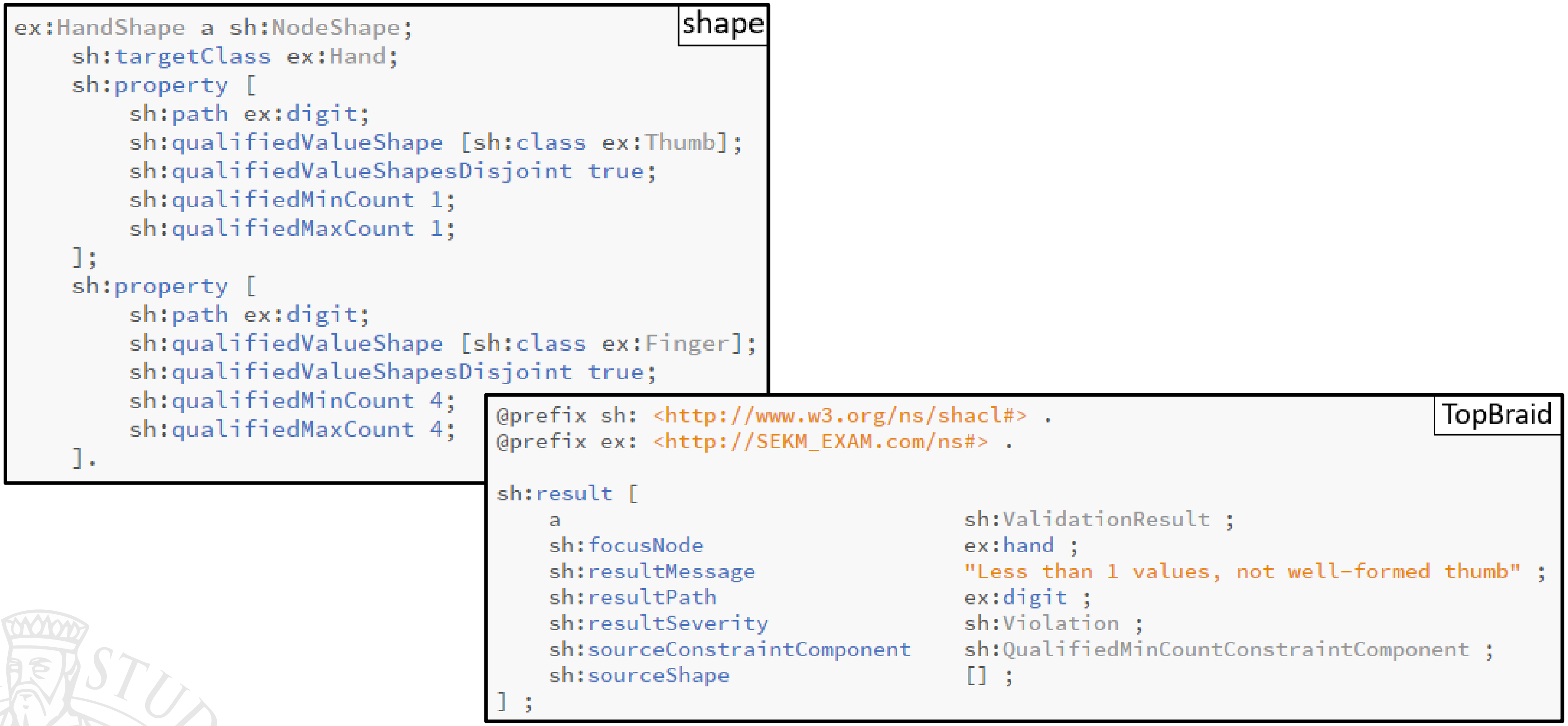
TopBraid

```
@prefix sh: <http://www.w3.org/ns/shacl#> .
@prefix ex: <http://SEKM_EXAM.com/ns#> .

sh:result [
    a                                   sh:ValidationResult ;
    sh:focusNode                        ex:dave ;
    sh:resultMessage                    "Less than 1 values for the qualified shape" ;
    sh:resultPath                       ex:parent ;
    sh:resultSeverity                   sh:Violation ;
    sh:sourceConstraintComponent        sh:QualifiedMinCountConstraintComponent ;
    sh:sourceShape                      [] ;
] ;
```

UNIVERSITÀ DEGLI STUDI FIRENZE

sh:qualifiedValueShapesDisjoint

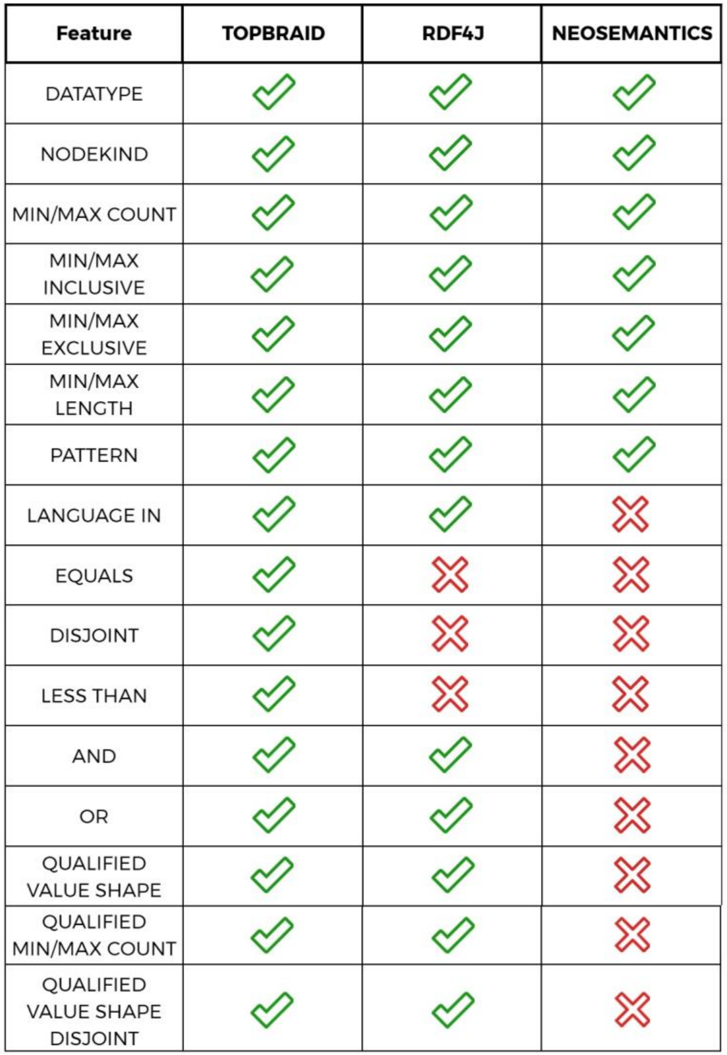
- This is not technically a feature, but an optional parameter of the previous ones. If set to *true* then the value nodes must not conform to any of the sibling shapes.



```
ex:HandShape a sh:NodeShape;
    sh:targetClass ex:Hand;
    sh:property [
        sh:path ex:digit;
        sh:qualifiedValueShape [sh:class ex:Thumb];
        sh:qualifiedValueShapesDisjoint true;
        sh:qualifiedMinCount 1;
        sh:qualifiedMaxCount 1;
    ];
    sh:property [
        sh:path ex:digit;
        sh:qualifiedValueShape [sh:class ex:Finger];
        sh:qualifiedValueShapesDisjoint true;
        sh:qualifiedMinCount 4;
        sh:qualifiedMaxCount 4;
    ].
```

```
@prefix sh: <http://www.w3.org/ns/shacl#> .
@prefix ex: <http://SEKM_EXAM.com/ns#> .

sh:result [
    a                               sh:ValidationResult ;
    sh:focusNode                    ex:hand ;
    sh:resultMessage                "Less than 1 values, not well-formed thumb" ;
    sh:resultPath                   ex:digit ;
    sh:resultSeverity               sh:Violation ;
    sh:sourceConstraintComponent    sh:QualifiedMinCountConstraintComponent ;
    sh:sourceShape                  [] ;
] ;
```

Summary support table



Feature	TOPBRAID	RDF4J	NEOSEMANTICS
DATATYPE	✓	✓	✓
NODEKIND	✓	✓	✓
MIN/MAX COUNT	✓	✓	✓
MIN/MAX INCLUSIVE	✓	✓	✓
MIN/MAX EXCLUSIVE	✓	✓	✓
MIN/MAX LENGTH	✓	✓	✓
PATTERN	✓	✓	✓
LANGUAGE IN	✓	✓	✗
EQUALS	✓	✗	✗
DISJOINT	✓	✗	✗
LESS THAN	✓	✗	✗
AND	✓	✓	✗
OR	✓	✓	✗
QUALIFIED VALUE SHAPE	✓	✓	✗
QUALIFIED MIN/MAX COUNT	✓	✓	✗
QUALIFIED VALUE SHAPE DISJOINT	✓	✓	✗

Conclusions

- Even though Neosemantics seems the most advantageous in terms of data scalability and execution speed, in relation to the quality of the support the other tools behave better.
- The choice of one tool over another is to be weighted considering the user needs and the availability of resources.
- Future works:
  - Write more shapes, even with more constraints
  - Try the validation on different datasets

Thanks for the attention