

XXXXX

XXXXX¹ and XXXXX²

¹ XXXXX
XXXX,
² XXXXX
XXXX

Abstract. Keywords: ..

1 Introduction

2 Ideas

- sind alle constraints abgedeckt?
- kann man alle constraints in SPARQL definieren?
- sind alle constraints mit Logik ausdrückbar?
- vollständig mit Reasoning — OW
- vollständig ohne Reasoning — CW
- es gibt keinen query rewriting mechanismus für OWL 2, nur für OWL-QL
- constraints in einer anderen constraint language definieren wenn constraints nicht in OWL beschrieben werden können
- durch reasoning entstehen Probleme, auf die man nicht gekommen wäre –i sofort nachvollziehbar
- zeigen, dass OWL-QL und constraint language einer konkreten Domäne zusammen vollständig sind
- System entwickeln, das effizient ist / Experimente

Nehmen wir nun an, dass dein Framework welches entsprechende SPARQL Queries generiert diese auf einem SPARQL Endpoint evaluiert der zu der vorliegenden Ontologie bzw. des darin verwendeten OWL 2 Profils das entsprechende Entailment Regime realisiert, wären die zurückgegebenen Resultsets vollständig. Wie das Entailment Regime im Endpoint realisiert ist, also durch Query Rewriting oder durch Vervollständigung der ABox, ist dabei irrelevant.

Wie allerdings bspw. in https://www.uni-ulm.de/fileadmin/website_uni-ulm/iui.inst.090/Lehre/WS_2011-2012/SemWebGrundlagen/LectureNotes.pdf auf Seite 51 veranschaulicht, ist die Komplexität des Reasoning abhängig von der zugrunde gelegten Sprache und kann daher nur in bestimmten Fällen effizient durchgeführt werden. Wie in unserem letzten Paper beschrieben zielt unter anderem die Definition von DL-Lite gerade darauf ab Reasoning Aufgaben und Query Answering effizient zu ermöglichen und ist Grundlage des OWL 2 QL

Profils. Nun ist allgemein bekannt, dass die logische Konsistenz für diese Art von Sprachen effizient geprüft werden kann.

Allerdings wäre wie bspw. in http://www.aifb.kit.edu/images/d/d2/2005_925_Haase.Consistent.Evol.1.pdf beschrieben auch eine sogenannte 'User-defined Consistency' denkbar. Genau an dieser Stelle könnten wir ansetzen.

Thomas: should we provide an implementation of RDF Validation with OWL 2 QL in the RDF Validator (purl.org/net/rdfval-demo)?

3 research questions

- for which RDF validation requirement the expressivity of DL-LiteA respectively OWL 2 QL is sufficient?
- for which RDF validation requirement additional constraint languages are needed?
- which constraint languages are suitable to express remaining requirements?
- what are the effects of these constraints regarding complexity?

4 OWL 2 QL

OWL 2 profiles specification: [1]

- OWL 2 QL constructs
- Difference between OWL 2 DL and OWL 2 QL

5 RDF Validation Requirements Expressible By OWL 2 QL

6 RDF Validation Requirements Not Expressible By OWL 2 QL

6.1 class-specific disjoint data properties

constraint (ShEx):

```
1 (foaf:name xsd:string | foaf:givenName xsd:string+, foaf:familyName xsd:string)
```

6.2 class-specific disjoint object properties

6.3 Class-Specific Disjoint Group of Properties

requirements DB:

- ID: R-13-DISJOINT-GROUP-OF-PROPERTIES-CLASS-SPECIFIC
- URL: <http://lelystad.informatik.uni-mannheim.de/rdf-validation/?q=node/20>

A `<Human>` has either a name or at least 1 given name and 1 family name.
constraint (ShEx):

```
1 <Human> {  
2   (  
3     foaf:name xsd:string  
4     |  
5     foaf:givenName xsd:string+,  
6     foaf:familyName xsd:string  
7   )  
8 }
```

valid data (ShEx):

```
1 <Anakin>  
2   foaf:givenName "Anakin" ;  
3   foaf:familyName "Skywalker" .
```

```
1 <Anakin>  
2   foaf:name "Anakin Skywalker" .
```

invalid data (ShEx):

```
1 <Anakin>  
2   foaf:givenName "Anakin" ;  
3   foaf:familyName "Skywalker" ;  
4   foaf:name "Anakin Skywalker" .
```

7 Related Work

8 Evaluation

evaluation using practical data set

9 Conclusion and Future Work

References

1. Boris Motik, Bernardo Cuenca Grau, Ian Horrocks, Zhe Wu, Achille Fokoue, and Carsten Lutz. Owl 2 web ontology language: Profiles. Technical report, December 2008.