

Benchmark Queries:

Simple Queries: Simple Queries (SQ) are about checking OWL2 constructs from structure point of view. These queries include finding distinct property characteristics (SQ1), all the object properties using property chain axiom (SQ2), domain and ranges of all the object and data properties (SQ3) and classes associated with disjoint union construct (SQ4).

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl:<http://www.w3.org/2002/07/owl#>
PREFIX xsd:< http://www.w3.org/2001/XMLSchema#>
PREFIX benchmark:<http://localhost/owl2bechmark/univ.owl#>
PREFIX rdfs:< http://www.w3.org/2000/01/rdf-schema#>
PREFIX dc:< http://purl.org/dc/elements/1.1/>
```

- a) SELECT distinct ?Proptype
WHERE
{
?Class rdf:type Owl:ObjectProperty.
?Class rdf:type ?Proptype
}
- b) SELECT ?ObjectProperty
WHERE
{
?ObjectProperty Owl:propertyChainAxiom ?node
}
- c) SELECT ?domain ?p ?range
WHERE
{
?p rdfs:domain ?domain.
?p rdfs:range ?range
}
- d) SELECT ?subject ?object
WHERE
{
?subject owl:disjointUnionOf ?object
}

Complex Queries: Complex Queries (CQ) include bushy patterns, long chains, large size, irregular patterns to check the responses of KBS systems on OWL2 ontologies. For instance, (CQ5) is used to find all the authors, their publications, friends, universities from_where_obtained masters degree, teaching courses and research interests (Bushy Pattern). Similarly (CQ6) is used to find the Long Chain Pattern. For large size results (CQ7) is used. (CQ8) is used to select all the instances of (x, y) of all the disjoint object properties.

e) SELECT ?Publication ?Pub_Author ?friend_name ?university_degree_from
?teaching_course ?ResearchArea
WHERE
{
?Publication ub:publicationAuthor ?PubAuthor.
?PubAuthor ub:isFriendOf ?friendname.
?PubAuthor ub:hasMasterDegreeFrom
?university_degree_from.
?PubAuthor ub:teacherOf ?teaching_course.
?PubAuthor ub:researchInterest ? ResearchArea
}

f) SELECT ?Publication ?Pub_Author ?supervisor_name ?membership_institute
?organization
WHERE
{
?Publication ub:publicationAuthor ?PubAuthor. ?PubAuthor
ub:isAdvisedBy ?supervisor_name. ?supervisor_name,
ub:isMemberOf ?membership_institute.
?membership_institute ub:subOrganization ?organization
}

g) SELECT ?student
WHERE
{
?student ub:takesCourse ?course
}

```

h) SELECT ?property1 ?property2 ?instanceX1 ?instanceY1 ?instanceX2 ?instanceY2
WHERE
{
  ?property1 owl:propertyDisjointWith ?property2.
  ?instanceX1 ?property1 ?instanceY1.
  ?instanceX2 ?property2 ?instanceY2
}

```

Object Property characteristics pattern based assertion queries: The object property characteristics pattern based queries (OPQ) are used for obtaining the object property characteristics patterns from the assertion data. There are nine queries in the OPQ to cover inverse of, Inverse functional, Asymmetric, Functional, Transitive, Disjoint, Symmetric, Irreflexive and Reflexive object properties (OPQ9 to OPQ17)

```

i) SELECT ?a ?p ?b
WHERE
{
  ?a ?p ?b.
  ?p rdf:type owl:ObjectProperty.
  ?b ?q ?a
  filter(?a!=?x)
}

j) SELECT ?a ?p ?b
WHERE
{
  ?a ?p ?b.
  ?p rdf:type owl:ObjectProperty
  filter not EXISTS
  {
    ?x ?p ?y
    filter(?a!=?x)
  }
}

```

k) SELECT ?a ?p ?b
WHERE
{
?a ?p ?b.
?p rdf:type owl:ObjectProperty
Filter not EXISTS
{
?b ?p ?a
}
}

l) SELECT ?a ?p ?b
WHERE
{
?a ?p ?b.
?p rdf:type owl:ObjectProperty
filter not EXISTS
{
?a ?p ?x
filter(?b!=?x)
}
}

m) SELECT distinct ?a ?p ?b ?c
WHERE
{
 ?a ?p ?b.
 ?b ?p ?c.
 ?a ?p ?c.
 ?p rdf:type owl:ObjectProperty.
 ?p rdfs:domain ?k.
 ?p rdfs:range ?k
 filter(?b!=?a && ?b!=?c)
}

n) SELECT distinct ?a ?p ?b ?p2
WHERE
{
 ?a ?p ?b.
 ?p rdf:type owl:ObjectProperty
 filter not exists
 {
 ?a ?p2 ?b
 filter(?p!=?p2)
 }
}

o) SELECT distinct ?a ?p ?b ?c
WHERE
{
 ?a ?p ?b.
 ?b ?p ?c.
 ?p rdf:type owl:ObjectProperty
 filter(?a=?c)
}

p) SELECT distinct ?a ?p ?b
WHERE
{
?a ?p ?b.
?p rdf:type owl:ObjectProperty.
?p rdfs:domain ?k.
?p rdfs:range ?k
filter not EXISTS
{
?a ?p ?a
}
}

q) SELECT distinct ?a ?p ?b
WHERE
{
?a ?p ?b.
?p rdf:type owl:ObjectProperty
filter(?a=?b)
}

