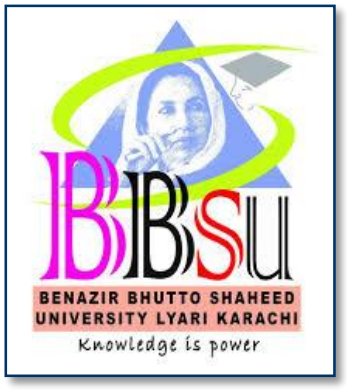


BENAZIR BHUTTO SHAHEED UNIVERSITY

LYARI, KARACHI

WEB SEMANTICS ASSIGNMENT # 04

Submitted to:

Sir Anwar Ali Sathio

Submitted by:

Name: Salman Abdul Rahim

Department: Computer Science

Semester: 7th – B

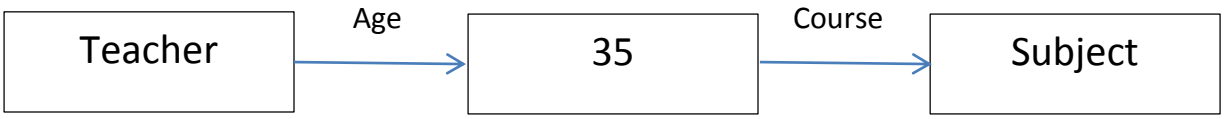
Roll #: 616

Q1: Discuss RDF vocabulary?

Answer: The language defined in this specification consists of a collection of RDF resources that can be used to describe properties of other RDF resources including properties which define application specific RDF vocabularies. The vocabulary is defined in a namespace informally called 'RDFS' here, and identified by the URI.

Q2: How can we maintain consistency and inconsistency between classes and properties? Support your answer through different example.

Answer: Consistency & Inconsistency : consistency is a flow information suppose we design a class so every next class is depended upon previous class or every next class is reflecting the information about every previous class. Every next class is derived almost previous classes. If any inconsistency comes in different classes or different properties, ontologies or data sources may be contradictory so we maintain and set.



the first query is to collect all the teachers aged 37. The second query asks for all the teachers that have declared an age. The third query, instead, requires collecting all the teachers that teach some courses but not Database, where dashed nodes and lines are introduced to allow a form of negation.

We can see a W-graph model as a Kripke structure and a semantically equivalency query as a formula of the temporal logic CTL. In this way, the inconsistency checking problem is reduced to the problem of finding out the states of the model.

Q3: How do all discussed layering issues will be solved?

The relationship between the various layers in this Semantic Web tower well this does depend somewhat on which layers are being considered, but there are some principles and some basic kinds of relationships that can be considered.

1- **Same-syntax semantic extension:** In this proposed layering, this is the same layering relationship as that between RDF and RDF Schema.
2- **Syntax and semantic extension:** In this proposed layering the semantics of OWL is defined as an extension of the semantics of RDF Schema.
3- **Same-syntax but diverging semantics:** In this proposal layering, OWL syntax would again be RDF syntax, or a subset of RDF syntax, but the meaning of some constructs would be different from their meaning in RDF or RDF Schema

4- **Differing syntax and semantics:** In this proposal layering, OWL differs from RDF and RDF Schema in both syntax and semantics. Again, although the formalisms would diverge, considerable overlap is possible and even desirable.

Solution For layering: One approach would be to change RDF or RDF Schema in some way, perhaps by removing some of the parts of RDF Schema that participate in the paradoxes. However, if we want to keep all of RDF and RDF Schema, it is necessary to pick one of the other solutions

The natural way of defining OWL in this layering proposal is to make the restrictions of OWL be new syntax. These restrictions would have a separate meaning defined by the OWL model theory. For example, a possible syntax for a maximum cardinality restriction like the one above could be:

<owl:cardinality maximum=''0'' property=''friend''> Person </owl:cardinality>

Q4: What are major requirements for OWL and how do they support our work?

Answer:

The Working Group currently feels that the requirements described below are essential to the language.

Explicit ontology extension: Ontologies must be able to explicitly extend other ontologies in order to reuse concepts while adding new classes and properties.

Ontologies as distinct resources: Ontologies must be resources that have their own unique identifiers, such as a URI reference.

Ontology metadata: It must be possible to provide meta-data for each ontology, such as author, publish-date, etc.

Class and property equivalence: The language must include features for stating that two classes or properties are equivalent

Classes as instances: The language must support the ability to treat classes as instances. This is because the same concept can often be seen as a class or an individual

Cardinality constraints: The language must support the specification of cardinality restrictions on properties.

XML syntax: The language should have XML serialization syntax. XML has become widely accepted by industry and numerous tools for processing XML have been developed.

Q5: Write expressions which can be driven from following:

Answer:

Statement of RDF:

1- Data Analysis with Python Expert.

Domain: Data Analysis

Relation: with

Range: Python

Super Class: Expert

Class: Data Analysis

Subclass: Python