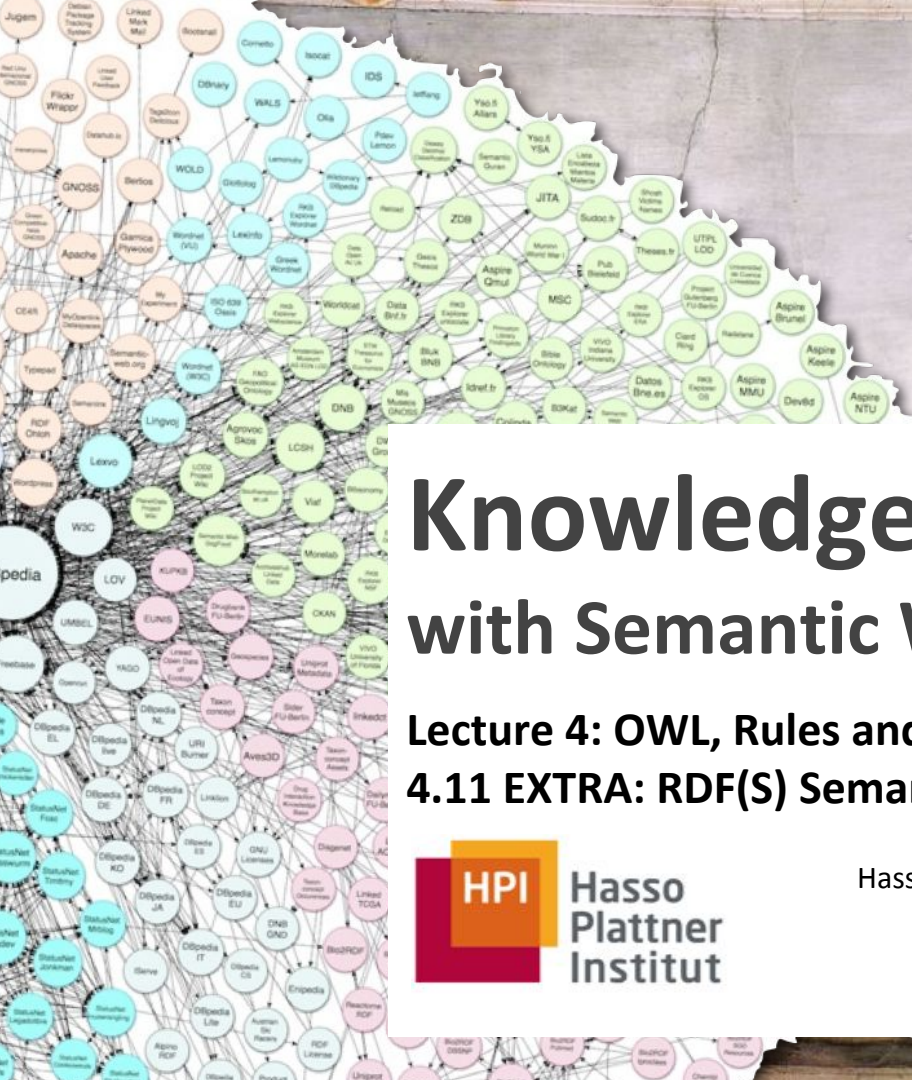




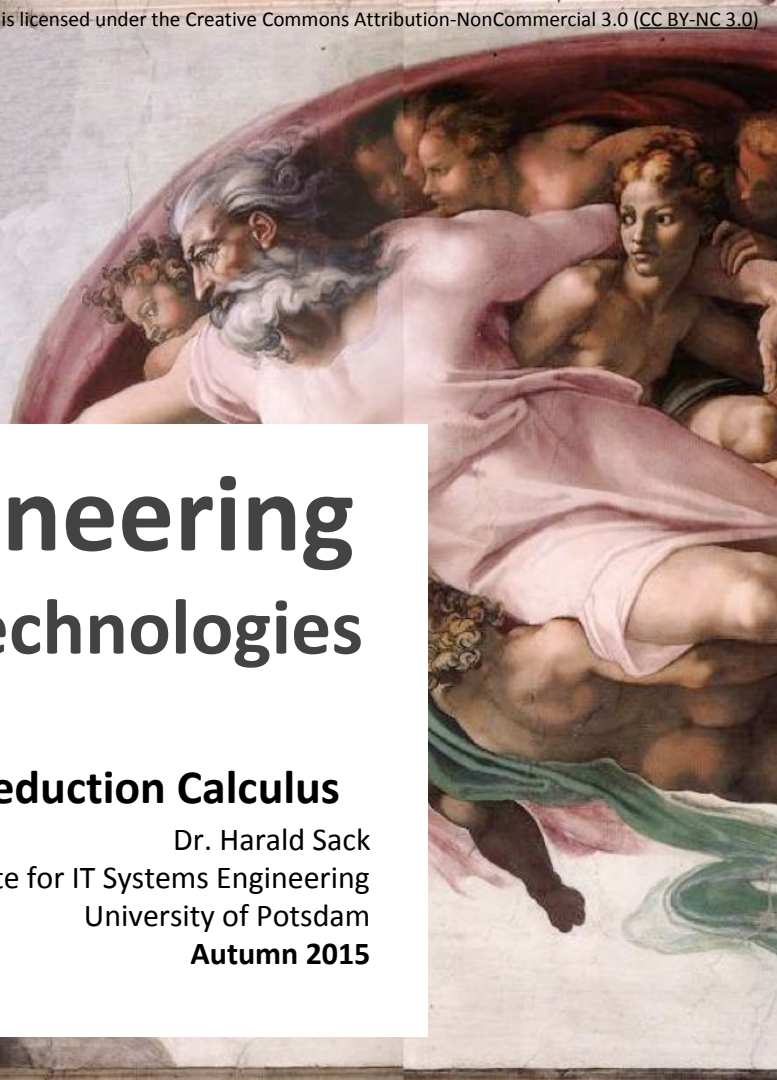
Knowledge Engineering with Semantic Web Technologies

Lecture 4: OWL, Rules and Reasoning

4.11 EXTRA: RDF(S) Semantics (4) - Deduction Calculus



Dr. Harald Sack
Hasso Plattner Institute for IT Systems Engineering
University of Potsdam
Autumn 2015



- **Model-theoretic Semantics** describes the behavior of a Logic wrt. *correct entailments*, but is **not** well suited for *algorithmic implementations*
- To proof $G_1 \models G_2$ via model-theoretic semantics
ALL (RDFS)-Interpretations would have to be considered
- Use **Algorithms to decide the validity of reasoning in a syntactic way**
(*Algorithms only apply assertions of logic without using the interpretation*)
- Proof of **correctness** is mandatory (!), i.e. operational (proof-theoretic) semantics and model-theoretic semantics are congruent
- Proof theory reduces model-theoretic semantics to symbol manipulation (=a calculus of deduction rules).

- General Form of **Deduction Rules**:
$$\frac{s_1 \dots s_n}{s}$$
- If statements $s_1, \dots, s_n$ are contained in the set of known valid assertions (are valid), then also the statement s can be added (is also valid).
- The whole set of deduction rules for a logic is called **Deduction Calculus**

General Notation for RDF(S) Deduction Rules

- **u** and **v** refer to arbitrary URIs or IDs of blank nodes
(any possible **Subject** of a triple)
- **a** and **b** refer to arbitrary URIs
(anything admissible for the **Property** of a triple)
- **x** and **y** refer to arbitrary URIs, IDs of blank nodes or literals
(any possible **Object** of a triple)
- **_:n** will be used for the ID of a **bnode**
- **l** may be any **Literal**

Deduction Rules for Simple Entailment

- all URIs are treated likewise

$$\frac{u \text{ a } x .}{u \text{ a } _ : n .} \quad \text{se1}$$

$$\frac{u \text{ a } x .}{_ : n \text{ a } x .} \quad \text{se2}$$

$_ : n$ must not be contained in the graph the rule is applied to!

- Theorem:**

A Graph G_1 **simply entails** a Graph G_2 ,
if G_1 via deduction rules se1 and se2 can be transformed
to a Graph G_1' in a way that G_2 is contained in G_1' .

Deduction Rules for RDF Entailment

$$\frac{}{u \text{ a } x .} \quad \text{rdfax}$$

every **axiomatic triple** „ $u \text{ a } x .$ “ can
always be deduced

$$\frac{u \text{ a } l .}{u \text{ a } _ : n .} \quad \text{lg}$$

literals can be substituted by **bnodes**
that do not already occur in the graph

$$\frac{u \text{ a } y .}{a \text{ rdf:type rdf:Property}} \quad \text{rdf1}$$

for all **properties** in a triple can be deduced
that it is an entity of the class of properties

$$\frac{u \text{ a } l .}{_ : n \text{ rdf:type rdf:XMLLiteral}} \quad \text{rdf2}$$

where $_ : n$ does not yet occur in the graph
unless it has been introduced by a preceding
application of the lg rule

- **Theorem:** A Graph G_1 **RDF entails** a Graph G_2 exactly if there exists a Graph G_1' that can be derived from G_1 via the rules rdfax, lg, rdf1 und rdf2 and that simply entails G_2 .

Deduction Rules for RDFS Entailment

$$\frac{a \text{ rdfs:domain } x . \quad u \text{ a } y .}{u \text{ rdf:type } x .} \quad \text{rdfs2}$$

$$\frac{a \text{ rdfs:range } x . \quad u \text{ a } y .}{y \text{ rdf:type } x .} \quad \text{rdfs3}$$

$$\frac{u \text{ a } x .}{u \text{ rdf:type rdfs:Resource } .} \quad \text{rdfs4a}$$

$$\frac{u \text{ a } x .}{x \text{ rdf:type rdfs:Resource } .} \quad \text{rdfs4b}$$

- Domain and Range Property Restrictions

- Everything is a resource

Deduction Rules for RDFS Entailment

- Subclasses

$$\frac{u \text{ rdf:type rdfs:Class .}}{u \text{ rdfs:subClassOf rdfs:Resource .}} \quad \text{rdfs8}$$

$$\frac{u \text{ rdfs:subClassOf } x . \quad v \text{ rdf:type } u .}{v \text{ rdf:type } x .} \quad \text{rdfs9}$$

$$\frac{u \text{ rdf:type rdfs:Class .}}{u \text{ rdfs:subClassOf } u .} \quad \text{rdfs10}$$

$$\frac{u \text{ rdfs:subClassOf } v . \quad v \text{ rdfs:subClassOf } x .}{u \text{ rdfs:subClassOf } x .} \quad \text{rdfs11}$$

Deduction Rules for RDFS Entailment

- Subproperties

$$\frac{u \text{ rdfs:subPropertyOf } v . \quad v \text{ rdfs:subPropertyOf } x .}{u \text{ rdfs:subPropertyOf } x .} \quad \text{rdfs5}$$

$$\frac{u \text{ rdf:type rdf:Property .}}{u \text{ rdfs:subPropertyOf } u .} \quad \text{rdfs6}$$

$$\frac{a \text{ rdfs:subPropertyOf } b . \quad u a y .}{u b y .} \quad \text{rdfs7}$$

Deduction Rules for RDFS Entailment

- Container

$$\frac{u \text{ rdf:type rdfs:ContainerMembershipProperty .}}{u \text{ rdfs:subPropertyOf rdfs:member .}} \quad \text{rdfs12}$$

- Literals

$$\frac{u \text{ rdf:type rdfs:Datatype .}}{u \text{ rdfs:subClassOf rdfs:Literal .}} \quad \text{rdfs13}$$

$$\frac{u \text{ a } _ : n .}{u \text{ a l .}} \quad \text{gl}$$

Deduction Rules for RDFS Entailment

- An inconsistent Graph G entails every arbitrary Graph
- **Inconsistency:** there is no interpretation I with $G^I = \text{true}$
- But with RDF(S) there are only restricted possibilities to create inconsistencies
- Example: „XML-Clash“:

```
ex:hasSmiley rdfs:range rdf:Literal .
```

```
rx:meanRemark ex:hasSmiley ">:->“^^XMLLiteral .
```

- **Theorem:**

A Graph G_1 **RDFS entails** a Graph G_2 exactly if there exists a Graph G_1' that can be derived from G_1 via the rules `rdfax`, `lg`, `rdf1`, `rdf2`, `rdfs1-rdfs13`, `gl`, and `rdfsax` with

- G_1' simply entails G_2 , or
- G_1' contains an XML-Clash (is inconsistent)



photo: [Matthew McVickar, flickr](#)

12 EXTRA - OWL Profiles

OpenHPI - Course Knowledge Engineering with Semantic Web Technologies

Lecture 4: OWL, Rules, and Reasoning