



TECHNISCHE  
UNIVERSITÄT  
DRESDEN

# FOUNDATIONS OF SEMANTIC WEB TECHNOLOGIES

## Semantics of RDF(S)

Sebastian Rudolph

Dresden, 16 April 2013



DRESDEN  
SCHOOL  
OF INFORMATICS

# Agenda

- 1 Motivation and Considerations
- 2 Simple Entailment
- 3 RDF Entailment
- 4 RDFS Entailment
- 5 Downsides of RDF(S)

# Agenda

- 1 Motivation and Considerations
- 2 Simple Entailment
- 3 RDF Entailment
- 4 RDFS Entailment
- 5 Downsides of RDF(S)

## Why Formal Semantics?

- after introduction of RDF(S), criticism of tool developers: different tools were incompatible (despite the existing specification)
- e.g. triple stores:
  - same RDF document
  - same SPARQL query
  - different answers
- thus a model-theoretic formal semantics was defined for RDF(S)

## How is RDF(S) Linked to a Logic?

- to start with: what are the sentences in RDF(S)?
  - basic elements (vocabulary  $V$ ): IRIs, bnodes and literals (these are not sentences themselves)
  - every triple

$$(s, p, o) \in (\text{IRI} \cup \text{bnode}) \times \text{IRI} \times (\text{IRI} \cup \text{bnode} \cup \text{literal})$$

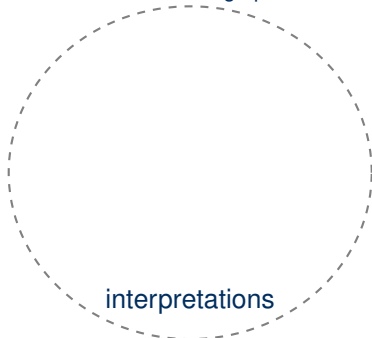
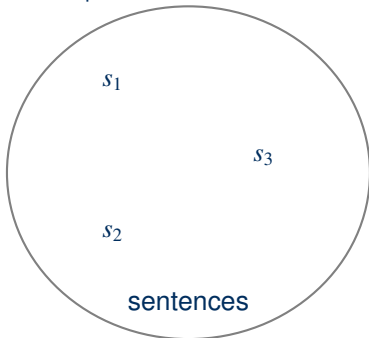
is a sentence

- every finite set of triples (denoted: graph) is a sentence

# How is RDF(S) Linked to a Logic?

What is the semantics?

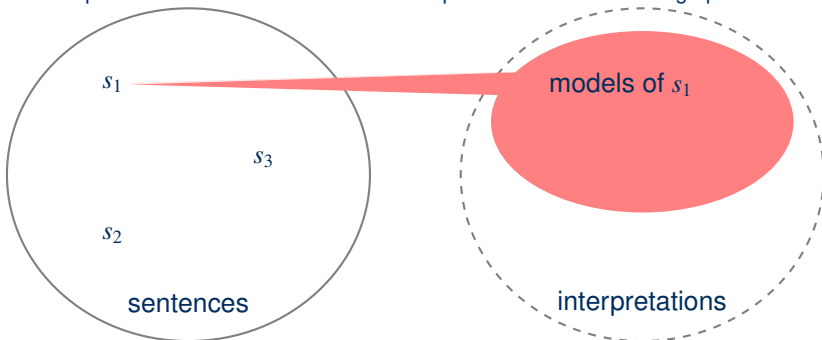
- consequence relation that defines when an RDF(S) graph  $G'$  logically follows from an RDF(S) graph  $G$ , i.e.  $G \models G'$
- model-theoretic semantics: we define a set of interpretations and stipulate under which conditions an interpretation is a model of a graph



# How is RDF(S) Linked to a Logic?

What is the semantics?

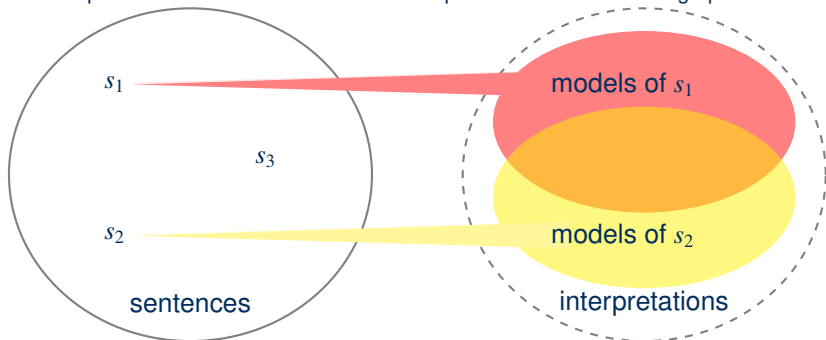
- consequence relation that defines when an RDF(S) graph  $G'$  logically follows from an RDF(S) graph  $G$ , i.e.  $G \models G'$
- model-theoretic semantics: we define a set of interpretations and stipulate under which conditions an interpretation is a model of a graph



# How is RDF(S) Linked to a Logic?

What is the semantics?

- consequence relation that defines when an RDF(S) graph  $G'$  logically follows from an RDF(S) graph  $G$ , i.e.  $G \models G'$
- model-theoretic semantics: we define a set of interpretations and stipulate under which conditions an interpretation is a model of a graph

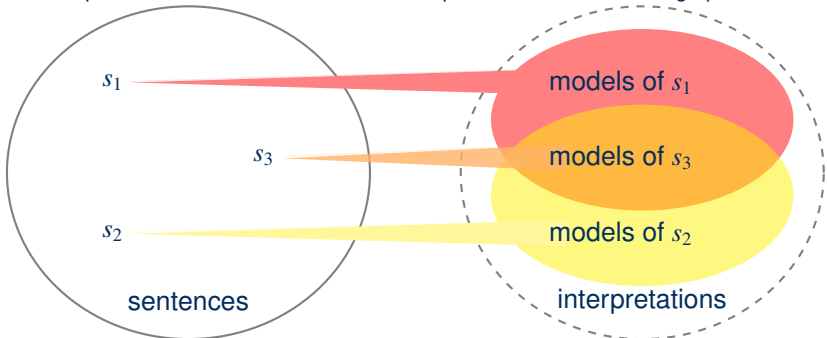




# How is RDF(S) Linked to a Logic?

What is the semantics?

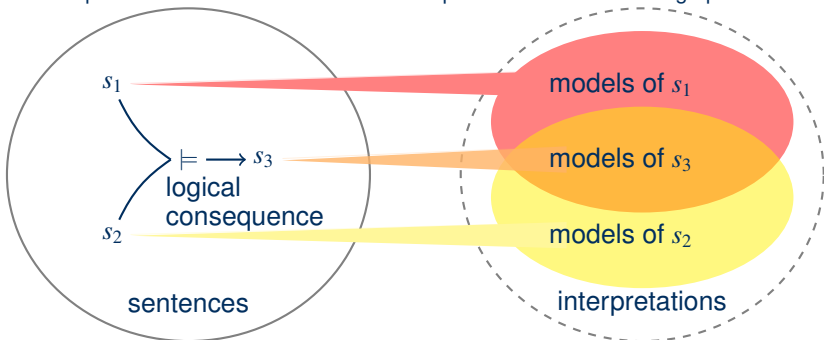
- consequence relation that defines when an RDF(S) graph  $G'$  logically follows from an RDF(S) graph  $G$ , i.e.  $G \models G'$
- model-theoretic semantics: we define a set of interpretations and stipulate under which conditions an interpretation is a model of a graph



# How is RDF(S) Linked to a Logic?

What is the semantics?

- consequence relation that defines when an RDF(S) graph  $G'$  logically follows from an RDF(S) graph  $G$ , i.e.  $G \models G'$
- model-theoretic semantics: we define a set of interpretations and stipulate under which conditions an interpretation is a model of a graph



# Semantics of RDF(S)

- we proceed stepwise:

simple interpretations

# Semantics of RDF(S)

- we proceed stepwise:

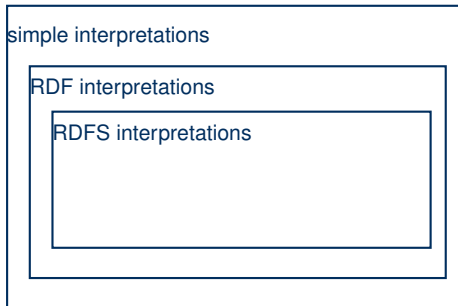


simple interpretations

RDF interpretations

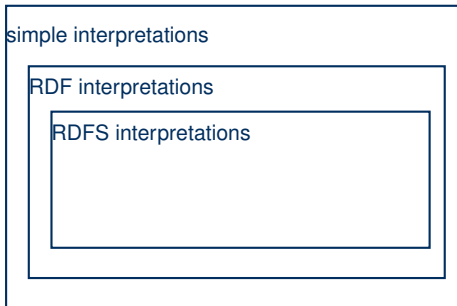
# Semantics of RDF(S)

- we proceed stepwise:



# Semantics of RDF(S)

- we proceed stepwise:



- the more we restrict the set of interpretations, the stronger the consequence relation becomes

# Agenda

- 1 Motivation and Considerations
- 2 Simple Entailment**
- 3 RDF Entailment
- 4 RDFS Entailment
- 5 Downsides of RDF(S)

# Semantics of the Simple Entailment

## Definition (Simple Interpretation)

A simple Interpretation  $\mathcal{I}$  for a vocabulary  $V$  consists of

- $IR$ , a non-empty set of resources, also referred to as domain, with
- $LV \subseteq IR$  the set of literal values, that contains (at least) all untyped literals from  $V$ , and
- $IP$ , the set of properties of  $\mathcal{I}$ ;
- $I_S$ , a function, mapping IRIs from  $V$  to the union of the sets  $IR$  and  $IP$ , i.e.,  
 $I_S: V \rightarrow IR \cup IP$ ,
- $I_{EXT}$ , a function, mapping every property to a set of pairs from  $IR$ , i.e.,  
 $I_{EXT}: IP \rightarrow 2^{IR \times IR}$  and
- $I_L$ , a function mapping typed literals from  $V$  into the set  $IR$  of resources.



## Semantics of the Simple Entailment

- IR is also called domain or universe of discourse of  $\mathcal{I}$
- $I_{\text{EXT}}(p)$  is also referred to as the extension of the property  $p$

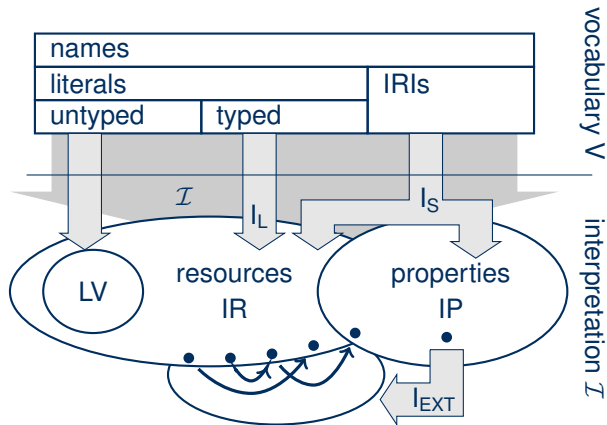
### Definition (interpretation function)

based on  $I_L$  and  $I_S$ , we define  $\cdot^{\mathcal{I}}$  as follows:

- every untyped literal "a" is mapped to  $a$  :  $(\text{"a"})^{\mathcal{I}} = a$
- every untyped literal with language information "a"@t is mapped to the pair  $\langle a, t \rangle$ , that is:  $(\text{"a"@t})^{\mathcal{I}} = \langle a, t \rangle$ ,
- every typed literal  $l$  is mapped to  $I_L(l)$ , that is:  $l^{\mathcal{I}} = I_L(l)$  and
- every IRI  $i$  is mapped to  $I_S(i)$ , hence:  $i^{\mathcal{I}} = I_S(i)$ .

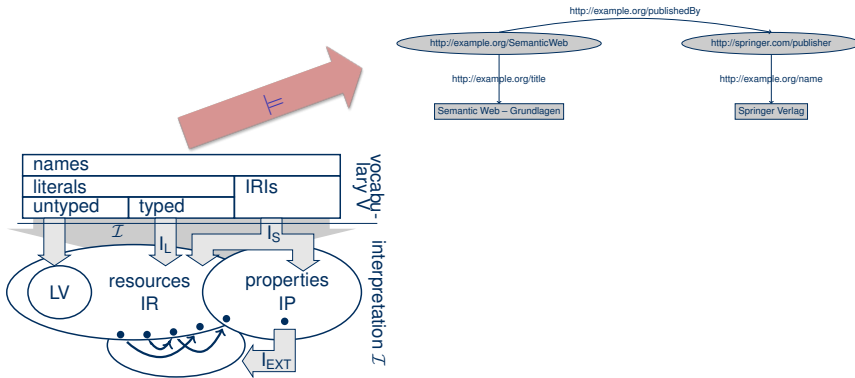
# Semantics of the Simple Entailment

Interpretation (schematic):



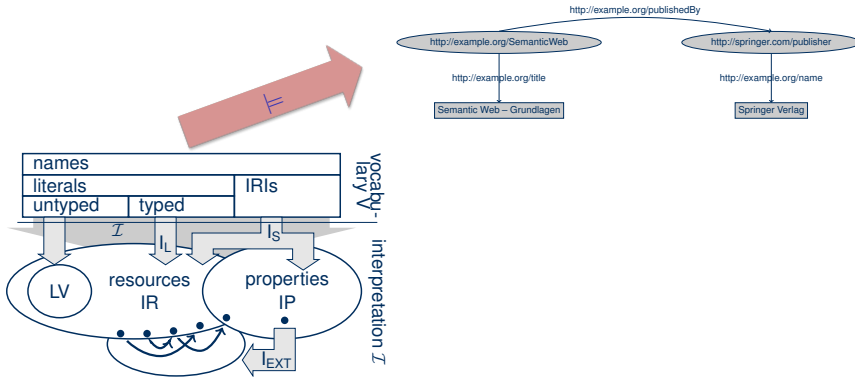
# Semantics of the Simple Entailment

- Question: When is a given interpretation a model of a graph?



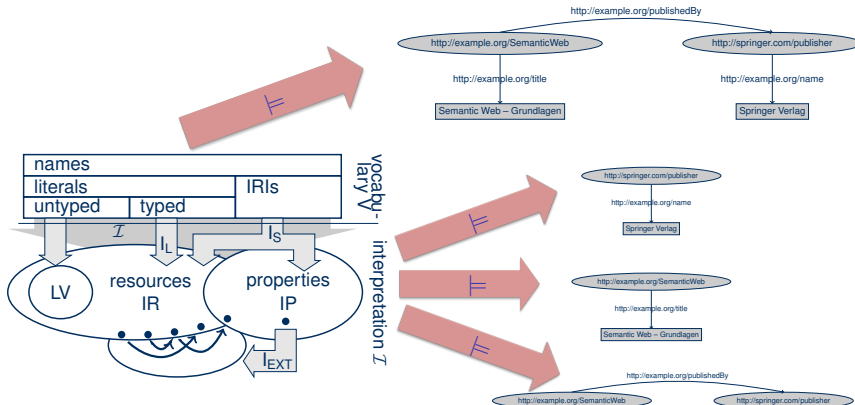
# Semantics of the Simple Entailment

- Question: When is a given interpretation a model of a graph?
- ... if it is a model for every triple of the graph!



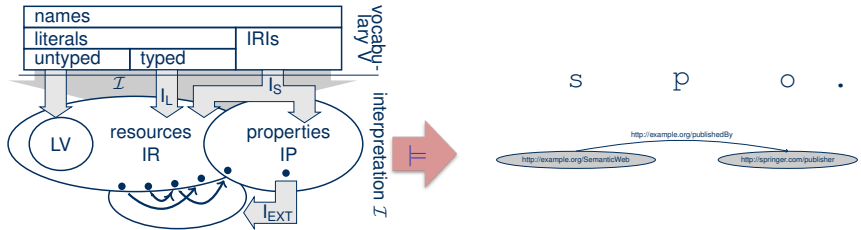
# Semantics of the Simple Entailment

- Question: When is a given interpretation a model of a graph?
- ... if it is a model for every triple of the graph!



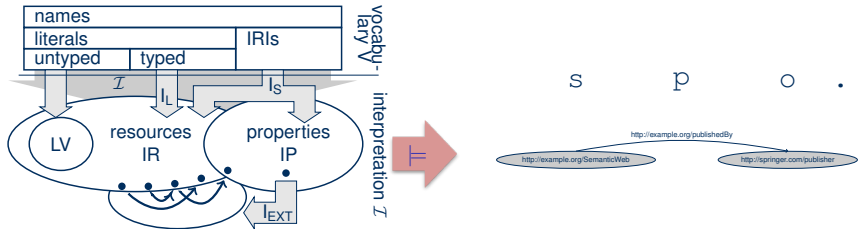
# Semantics of the Simple Entailment

- Question: When is a given interpretation a model of a triple?



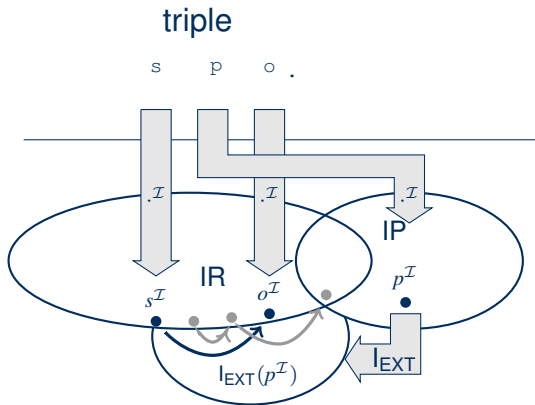
# Semantics of the Simple Entailment

- Question: When is a given interpretation a model of a triple?
- ... if all subject, predicate, and object are contained in  $V$  and additionally  $\langle s^{\mathcal{I}}, o^{\mathcal{I}} \rangle \in I_{EXT}(p^{\mathcal{I}})$  holds



# Semantics of Simple Entailment

schematically:



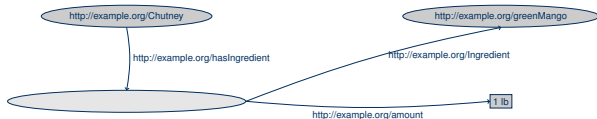


## Semantics of Simple Entailment

- ...oops, we forgot the bnodes!
- let  $A$  be a function mapping all bnodes to elements of  $IR$
- given an interpretation  $\mathcal{I}$ , let  $\mathcal{I} + A$  behave just like  $\mathcal{I}$  on the vocabulary, and additionally for every bnode  $_:label$  let  $(_:label)^{\mathcal{I}+A} = A(_:label)$
- now, an interpretation  $\mathcal{I}$  is a model of an RDF graph  $G$ , if there exists an  $A$  such that all triples are satisfied w.r.t.  $\mathcal{I} + A$

# Simple Interpretations: Example

given graph  $G$ :

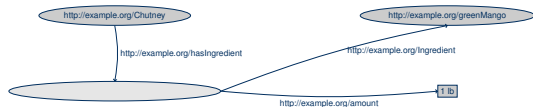


and interpretation  $\mathcal{I}$ :

$$\begin{array}{lll}
 \text{IR} = \{c, g, h, z, l, m, 1 \text{ lb}\} & \text{I}_S = \text{ex:Chutney} & \mapsto c \\
 \text{IP} = \{h, z, m\} & \text{ex:greenMango} & \mapsto g \\
 \text{LV} = \{1 \text{ lb}\} & \text{ex:hasIngredient} & \mapsto h \\
 \text{I}_{\text{EXT}} = h \mapsto \{\langle c, l \rangle\} & \text{ex:ingredient} & \mapsto z \\
 & \text{ex:amount} & \mapsto m \\
 & & \text{I}_L \text{ is the "empty function"} \\
 & & z \mapsto \{\langle l, g \rangle\} \\
 & & m \mapsto \{\langle l, 1 \text{ lb} \rangle\}
 \end{array}$$

Is  $\mathcal{I}$  a model of  $G$ ?

# Simple Interpretations: Example



$$\begin{array}{lll}
 \text{IR} = \{c, g, h, z, l, m, 1 \text{ lb}\} & \text{I}_S = \text{ex:Chutney} & \mapsto c \\
 \text{IP} = \{h, z, m\} & \text{ex:greenMango} & \mapsto g \\
 \text{LV} = \{1 \text{ lb}\} & \text{ex:hasIngredient} & \mapsto h \\
 \text{I}_{\text{EXT}} = h \mapsto \{\langle c, l \rangle\} & \text{ex:ingredient} & \mapsto z \\
 & \text{ex:amount} & \mapsto m \\
 & m \mapsto \{\langle l, 1 \text{ lb} \rangle\} & \text{I}_L \text{ is the "empty function"}
 \end{array}$$

- If we pick  $A: \_:\text{id1} \mapsto l$ , then we get

$$\begin{array}{lll}
 \langle \text{ex:Chutney}^{\mathcal{I}+A}, \_:\text{id1}^{\mathcal{I}+A} \rangle & = \langle c, l \rangle & \in \text{I}_{\text{EXT}}(h) = \text{I}_{\text{EXT}}(\text{ex:hasIngredient}^{\mathcal{I}+A}) \\
 \langle \_:\text{id1}^{\mathcal{I}+A}, \text{ex:greenMango}^{\mathcal{I}+A} \rangle & = \langle l, g \rangle & \in \text{I}_{\text{EXT}}(z) = \text{I}_{\text{EXT}}(\text{ex:ingredient}^{\mathcal{I}+A}) \\
 \langle \_:\text{id1}^{\mathcal{I}+A}, "1 \text{ lb}"^{\mathcal{I}+A} \rangle & = \langle l, 1 \text{ lb} \rangle & \in \text{I}_{\text{EXT}}(m) = \text{I}_{\text{EXT}}(\text{ex:amount}^{\mathcal{I}+A})
 \end{array}$$

- Therefore,  $\mathcal{I}$  is a model of  $G$ .

# Simple Entailment

- definition of simple interpretations fixes the notion of simple entailment for RDF graphs
- question: how can this (abstractly defined) semantics be turned something computable
- answer: deduction rules

# Simple Entailment

deduction rules for simple entailment:

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

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

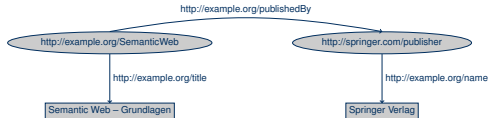
- precondition for applying this rule: the bnode has not already been associated with another IRI or literal

# Simple Entailment

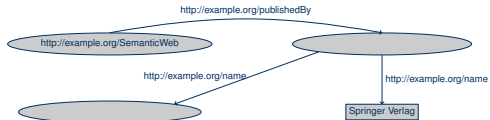
## Theorem

A graph  $G_2$  is simply entailed by a graph  $G_1$  if  $G_1$  can be extended to a graph  $G'_1$  by applying the rules se1 and se2 such that  $G_2$  is contained in  $G'_1$ .

Example.: the  
graph



simply entails



# Agenda

- 1 Motivation and Considerations
- 2 Simple Entailment
- 3 RDF Entailment**
- 4 RDFS Entailment
- 5 Downsides of RDF(S)

## RDF interpretations

RDF interpretations are specific simple interpretations, where additional conditions are imposed on the URIs of the RDF vocabulary

```
rdf:type rdf:Property rdf:XMLLiteral rdf:nil  
rdf:List rdf:Statement rdf:subject rdf:predicate  
rdf:object rdf:first rdf:rest rdf:Seq rdf:Bag  
rdf:Alt rdf:_1 rdf:_2 ...
```

inorder to realize their intended semantics.



# Conditions for RDF Interpretations

An RDF interpretation for a vocabulary  $V$  is a simple interpretation for the vocabulary  $V \cup V_{\text{RDF}}$  that additionally satisfies the following conditions:

1.  $x \in \text{IP}$  exactly if  $\langle x, \text{rdf:Property}^{\mathcal{I}} \rangle \in \text{I}_{\text{EXT}}(\text{rdf:type}^{\mathcal{I}})$ .

# Conditions for RDF Interpretations

An RDF interpretation for a vocabulary  $V$  is a simple interpretation for the vocabulary  $V \cup V_{\text{RDF}}$  that additionally satisfies the following conditions:

1.  $x \in \text{IP}$  exactly if  $\langle x, \text{rdf:Property}^{\mathcal{I}} \rangle \in \text{I}_{\text{EXT}}(\text{rdf:type}^{\mathcal{I}})$ .

“For every triple predicate we can infer that it is an member of the class of all properties.”

## Conditions for RDF Interpretations

An RDF interpretation for a vocabulary  $V$  is a simple interpretation for the vocabulary  $V \cup V_{\text{RDF}}$  that additionally satisfies the following conditions:

1.  $x \in \text{IP}$  exactly if  $\langle x, \text{rdf:Property}^{\mathcal{I}} \rangle \in \text{I}_{\text{EXT}}(\text{rdf:type}^{\mathcal{I}})$ .

“For every triple predicate we can infer that it is an member of the class of all properties.”

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

## Conditions for RDF Interpretations

2. If  $"s" \hat{=} \text{rdf:XMLLiteral}$  is contained in  $V$  and  $s$  is a well-formed XML literal, then
- $I_L("s" \hat{=} \text{rdf:XMLLiteral})$  is the XML value of  $s$ ;
  - $I_L("s" \hat{=} \text{rdf:XMLLiteral}) \in LV$ ;
  - $\langle I_L("s" \hat{=} \text{rdf:XMLLiteral}), \text{rdf:XMLLiteral}^I \rangle \in I_{\text{EXT}}(\text{rdf:type}^I)$

$$\frac{u \ a \ l}{l \ \text{rdf:type} \ \text{rdf:XMLLiteral}} \quad ??? \quad \text{für } l \text{ a well-formed XML literal}$$

## Conditions for RDF Interpretations

2. If  $"s" \text{^^} \text{rdf:XMLLiteral}$  is contained in  $V$  and  $s$  is a well-formed XML literal, then
- $I_L("s" \text{^^} \text{rdf:XMLLiteral})$  is the XML value of  $s$ ;
  - $I_L("s" \text{^^} \text{rdf:XMLLiteral}) \in LV$ ;
  - $\langle I_L("s" \text{^^} \text{rdf:XMLLiteral}), \text{rdf:XMLLiteral}^I \rangle \in I_{\text{EXT}}(\text{rdf:type}^I)$

$$\frac{\text{u a l}}{l \text{ rdf:type rdf:XMLLiteral}} \quad ??? \quad \text{für } l \text{ a well-formed XML literal}$$

Oops, literals must not occur in subject position!

## Conditions for RDF Interpretations

2. If `"s"^^rdf:XMLLiteral` is contained in  $V$  and  $s$  is a well-formed XML literal, then
  - $I_L("s"^^rdf:XMLLiteral)$  is the XML value of  $s$ ;
  - $I_L("s"^^rdf:XMLLiteral) \in LV$ ;
  - $\langle I_L("s"^^rdf:XMLLiteral), rdf:XMLLiteral^{\mathcal{I}} \rangle \in I_{EXT}(rdf:type^{\mathcal{I}})$

## Conditions for RDF Interpretations

2. If  $"s" \hat{=} \text{rdf:XMLLiteral}$  is contained in  $V$  and  $s$  is a well-formed XML literal, then
- $I_L("s" \hat{=} \text{rdf:XMLLiteral})$  is the XML value of  $s$ ;
  - $I_L("s" \hat{=} \text{rdf:XMLLiteral}) \in LV$ ;
  - $\langle I_L("s" \hat{=} \text{rdf:XMLLiteral}), \text{rdf:XMLLiteral}^I \rangle \in I_{\text{EXT}}(\text{rdf:type}^I)$

$\frac{u \ a \ l}{u \ a \ _:n}$	$lg$	$l$ a literal, $_:n$ not bound otherwise
$\frac{u \ a \ _:n}{_:n \ \text{rdf:type} \ \text{rdf:XMLLiteral}}$	$rdf2$	If rule $lg$ has assigned $_:n$ to the XML Literal $l$

## Conditions for RDF Interpretations

3. If  $"s" \hat{=} \text{rdf:XMLLiteral}$  is contained in  $V$  and  $s$  is an ill-formed XML literal, then
  - $I_L("s" \hat{=} \text{rdf:XMLLiteral}) \notin LV$  and
  - $\langle I_L("s" \hat{=} \text{rdf:XMLLiteral}), \text{rdf:XMLLiteral}^{\mathcal{I}} \rangle \notin I_{\text{EXT}}(\text{rdf:type}^{\mathcal{I}})$ .



## RDF Interpretations

- Note:  $x$  is a property exactly if it is linked to the resource denoted by `rdf:Property` via the `rdf:type` property (this has the direct consequence that in every RDF interpretation holds  $IP \subseteq IR$ ).
- The value space of the `rdf:XMLLiteral` datatype contains for every well-formed XML string exactly one so-called XML value. The RDF specs only stipulate that this value is neither an XML string itself nor a data value of any XML Schema datatype nor a Unicode string.

# RDF Interpretations

- additional requirement: every RDF interpretation must be a model of the following “axiomatic” triples:

<code>rdf:type</code>	<code>rdf:type</code>	<code>rdf:Property</code>	<code>.</code>
<code>rdf:subject</code>	<code>rdf:type</code>	<code>rdf:Property</code>	<code>.</code>
<code>rdf:predicate</code>	<code>rdf:type</code>	<code>rdf:Property</code>	<code>.</code>
<code>rdf:object</code>	<code>rdf:type</code>	<code>rdf:Property</code>	<code>.</code>
<code>rdf:first</code>	<code>rdf:type</code>	<code>rdf:Property</code>	<code>.</code>
<code>rdf:rest</code>	<code>rdf:type</code>	<code>rdf:Property</code>	<code>.</code>
<code>rdf:value</code>	<code>rdf:type</code>	<code>rdf:Property</code>	<code>.</code>
<code>rdf:_1</code>	<code>rdf:type</code>	<code>rdf:Property</code>	<code>.</code>
<code>rdf:_2</code>	<code>rdf:type</code>	<code>rdf:Property</code>	<code>.</code>
<code>...</code>	<code>rdf:type</code>	<code>rdf:Property</code>	<code>.</code>
<code>rdf:nil</code>	<code>rdf:type</code>	<code>rdf:List</code>	<code>.</code>

$$\frac{\quad}{u \ a \ x}$$
**rdfax**

every axiomatic triple “`u a x .`”  
can always be derived

## RDF Entailment

- Theorem: A graph  $G_2$  is RDF-entailed by a graph  $G_1$ , if there is a graph  $G'_1$ , such that
  - $G'_1$  can be derived from  $G_1$  via lg, rdf1, rdf2 and rdfax and
  - $G_2$  is simply entailed by  $G'_1$ .
  
- note: two-stage deduction process

# Agenda

- 1 Motivation and Considerations
- 2 Simple Entailment
- 3 RDF Entailment
- 4 RDFS Entailment**
- 5 Downsides of RDF(S)

## RDFS Interpretations

... RDFS interpretations are specific RDF interpretations, where additional constraints are imposed for the URIs of the RDFS vocabulary

<code>rdfs:domain</code>	<code>rdfs:range</code>	<code>rdfs:Resource</code>
<code>rdfs:Literal</code>	<code>rdfs:Datatype</code>	<code>rdfs:Class</code>
<code>rdfs:subClassOf</code>	<code>rdfs:subPropertyOf</code>	<code>rdfs:Container</code>
<code>rdfs:member</code>	<code>rdfs:ContainerMembershipProperty</code>	
<code>rdfs:comment</code>	<code>rdfs:seeAlso</code>	<code>rdfs:isDefinedBy</code>
<code>rdfs:label</code>		

such that the intended semantics of these URIs is realized.

## RDFS Interpretations

- for the sake of easier representation, we introduce – given an interpretation  $\mathcal{I}$  – a function  $I_{\text{CEXT}}$  that maps resources to sets of resources (thus:  $I_{\text{CEXT}}: \text{IR} \rightarrow 2^{\text{IR}}$ ) by letting  $I_{\text{CEXT}}(y)$  contain exactly those elements  $x$ , for which  $\langle x, y \rangle$  is contained in  $I_{\text{EXT}}(\text{rdf:type}^{\mathcal{I}})$ . We call  $I_{\text{CEXT}}(y)$  the (class) extension of  $y$ .
- moreover, we let  $\text{IC}$  be the extension of the specific IRI `rdfs:Class`, hence:  $\text{IC} = I_{\text{CEXT}}(\text{rdfs:Class}^{\mathcal{I}})$ .
- note: both  $I_{\text{CEXT}}$  as well as  $\text{IC}$  are fully determined by  $\cdot^{\mathcal{I}}$  and  $I_{\text{EXT}}$ .

## RDFS Interpretations

An RDFS interpretation for a vocabulary  $V$  is an RDF interpretation for the vocabulary  $V \cup V_{\text{RDFS}}$ , that additionally satisfies the following criteria:

- $IR = I_{\text{CEXT}}(\text{rdfs:Resource}^{\mathcal{I}})$   
Every resource is of type `rdfs:Resource`.
- $LV = I_{\text{CEXT}}(\text{rdfs:Literal}^{\mathcal{I}})$   
Every untyped and every well-formed typed literal is of type `rdfs:Literal`.
- If  $\langle x, y \rangle \in I_{\text{EXT}}(\text{rdfs:domain}^{\mathcal{I}})$  and  $\langle u, v \rangle \in I_{\text{EXT}}(x)$ , then  $u \in I_{\text{CEXT}}(y)$ .  
If the property `rdfs:domain` connects  $x$  with  $y$  and the property  $x$  connects the resources  $u$  and  $v$ , then  $u$  is of type  $y$ .

## RDFS Interpretations

- If  $\langle x, y \rangle \in I_{\text{EXT}}(\text{rdfs:range}^{\mathcal{I}})$  and  $\langle u, v \rangle \in I_{\text{EXT}}(x)$ , then  $v \in I_{\text{CEXT}}(y)$ .  
If the property `rdfs:range` connects  $x$  with  $y$  and the property  $x$  connects the resources  $u$  and  $v$ , then  $v$  is of type  $y$ .
- $I_{\text{EXT}}(\text{rdfs:subPropertyOf}^{\mathcal{I}})$  is reflexive and transitive on IP.  
The `rdfs:subPropertyOf` property connects every property with itself. Moreover, if `rdfs:subPropertyOf` connects a property  $x$  with a property  $y$  and additionally  $y$  with a property  $z$ , then `rdfs:subPropertyOf` also connects  $x$  directly with  $z$ .



## RDFS Interpretations

- If  $\langle x, y \rangle \in \text{I}_{\text{EXT}}(\text{rdfs:subPropertyOf}^{\mathcal{I}})$ ,  
then  $x, y \in \text{IP}$  and  $\text{I}_{\text{EXT}}(x) \subseteq \text{I}_{\text{EXT}}(y)$ .  
If `rdfs:subPropertyOf` connects  $x$  with  $y$ , then both  $x$  and  $y$  are properties every pair of resources contained in the extension of  $x$  is also contained in the extension of  $y$ .
- If  $x \in \text{IC}$ , then  $\langle x, \text{rdfs:Resource}^{\mathcal{I}} \rangle \in \text{I}_{\text{EXT}}(\text{rdfs:subClassOf}^{\mathcal{I}})$ .  
If  $x$  represents a class, then it has to be a subclass of the class of all resources, i.e., the pair containing  $x$  and `rdfs:Resource` is in the extension of `rdfs:subClassOf`.

## RDFS Interpretations

- If  $\langle x, y \rangle \in I_{\text{EXT}}(\text{rdfs:subClassOf}^{\mathcal{I}})$ , then  $x, y \in \text{IC}$  and  $I_{\text{CEXT}}(x) \subseteq I_{\text{CEXT}}(y)$ .  
If  $x$  and  $y$  are connected via the `rdfs:subClassOf` property, then both  $x$  and  $y$  are classes and the (class) extension of  $x$  is a subset of the (class) extension of  $y$ .
- $I_{\text{EXT}}(\text{rdfs:subClassOf}^{\mathcal{I}})$  is reflexive and transitive on  $\text{IC}$ .  
The `rdfs:subClassOf` property connects every class to itself.  
Moreover, whenever this property connects a class  $x$  with a class  $y$  and a class  $y$  with a class  $z$ , then it also directly connects  $x$  with  $z$ .

## RDFS Interpretations

- If  $x \in I_{\text{CEXT}}(\text{rdfs:ContainerMembershipProperty}^{\mathcal{I}})$ , then  $\langle x, \text{rdfs:member}^{\mathcal{I}} \rangle \in I_{\text{EXT}}(\text{rdfs:subPropertyOf}^{\mathcal{I}})$ .  
If  $x$  is a property of the type `rdfs:ContainerMembershipProperty`, then it is `rdfs:subPropertyOf`-connected with the property `rdfs:member`.
- If  $x \in I_{\text{CEXT}}(\text{rdfs:Datatype}^{\mathcal{I}})$ , then  $\langle x, \text{rdfs:Literal}^{\mathcal{I}} \rangle \in I_{\text{EXT}}(\text{rdfs:subClassOf}^{\mathcal{I}})$ .  
If some  $x$  is typed as element of the class `rdfs:Datatype`, then it must be a subclass of the class of all literal values (denoted by `rdfs:Literal`).
- ... additionally we require satisfaction of the following axiomatic triples:

## RDFS Interpretations

<code>rdf:type</code>	<code>rdfs:domain</code>	<code>rdfs:Resource</code> .
<code>rdfs:domain</code>	<code>rdfs:domain</code>	<code>rdf:Property</code> .
<code>rdfs:range</code>	<code>rdfs:domain</code>	<code>rdf:Property</code> .
<code>rdfs:subPropertyOf</code>	<code>rdfs:domain</code>	<code>rdf:Property</code> .
<code>rdfs:subClassOf</code>	<code>rdfs:domain</code>	<code>rdfs:Class</code> .
<code>rdf:subject</code>	<code>rdfs:domain</code>	<code>rdf:Statement</code> .
<code>rdf:predicate</code>	<code>rdfs:domain</code>	<code>rdf:Statement</code> .
<code>rdf:object</code>	<code>rdfs:domain</code>	<code>rdf:Statement</code> .
<code>rdfs:member</code>	<code>rdfs:domain</code>	<code>rdfs:Resource</code> .
<code>rdf:first</code>	<code>rdfs:domain</code>	<code>rdf&gt;List</code> .
<code>rdf:rest</code>	<code>rdfs:domain</code>	<code>rdf&gt;List</code> .
<code>rdfs:seeAlso</code>	<code>rdfs:domain</code>	<code>rdfs:Resource</code> .
<code>rdfs:isDefinedBy</code>	<code>rdfs:domain</code>	<code>rdfs:Resource</code> .
<code>rdfs:comment</code>	<code>rdfs:domain</code>	<code>rdfs:Resource</code> .
<code>rdfs:label</code>	<code>rdfs:domain</code>	<code>rdfs:Resource</code> .
<code>rdf:value</code>	<code>rdfs:domain</code>	<code>rdfs:Resource</code> .

## RDFS Interpretations

<code>rdf:type</code>	<code>rdfs:range</code>	<code>rdfs:Class .</code>
<code>rdfs:domain</code>	<code>rdfs:range</code>	<code>rdfs:Class .</code>
<code>rdfs:range</code>	<code>rdfs:range</code>	<code>rdfs:Class .</code>
<code>rdfs:subPropertyOf</code>	<code>rdfs:range</code>	<code>rdf:Property .</code>
<code>rdfs:subClassOf</code>	<code>rdfs:range</code>	<code>rdfs:Class .</code>
<code>rdf:subject</code>	<code>rdfs:range</code>	<code>rdfs:Resource .</code>
<code>rdf:predicate</code>	<code>rdfs:range</code>	<code>rdfs:Resource .</code>
<code>rdf:object</code>	<code>rdfs:range</code>	<code>rdfs:Resource .</code>
<code>rdfs:member</code>	<code>rdfs:range</code>	<code>rdfs:Resource .</code>
<code>rdf:first</code>	<code>rdfs:range</code>	<code>rdfs:Resource .</code>
<code>rdf:rest</code>	<code>rdfs:range</code>	<code>rdf:List .</code>
<code>rdfs:seeAlso</code>	<code>rdfs:range</code>	<code>rdfs:Resource .</code>
<code>rdfs:isDefinedBy</code>	<code>rdfs:range</code>	<code>rdfs:Resource .</code>
<code>rdfs:comment</code>	<code>rdfs:range</code>	<code>rdfs:Literal .</code>
<code>rdfs:label</code>	<code>rdfs:range</code>	<code>rdfs:Literal .</code>
<code>rdf:value</code>	<code>rdfs:range</code>	<code>rdfs:Resource .</code>

# RDFS Interpretations

```
rdfs:ContainerMembershipProperty
    rdfs:subClassOf      rdf:Property .
rdf:Alt                rdfs:subClassOf      rdfs:Container .
rdf:Bag                rdfs:subClassOf      rdfs:Container .
rdf:Seq                rdfs:subClassOf      rdfs:Container .

rdfs:isDefinedBy      rdfs:subPropertyOf  rdfs:seeAlso .

rdf:XMLLiteral        rdf:type             rdfs:Datatype .
rdf:XMLLiteral        rdfs:subClassOf      rdfs:Literal .
rdfs:Datatype         rdfs:subClassOf      rdfs:Class .

rdf:_1                rdf:type             rdfs:ContainerMembershipProperty .
                        rdfs:ContainerMembershipProperty .
rdf:_1                rdfs:domain           rdfs:Resource .
rdf:_1                rdfs:range           rdfs:Resource .
rdf:_2                rdf:type             rdfs:ContainerMembershipProperty .
                        rdfs:ContainerMembershipProperty .
```



## RDFS Entailment

$\frac{u \text{ a } x .}{u \text{ rdf:type rdfs:Resource } .}$	rdfs4a	the subject of every triple is a resource
$\frac{u \text{ a } v .}{v \text{ rdf:type rdfs:Resource } .}$	rdfs4b	objects that are not literals are resources as well
$\frac{u \text{ rdfs:subPropertyOf } v . v \text{ rdfs:subPropertyOf } x .}{u \text{ rdfs:subPropertyOf } x .}$	rdfs5	transitivity
$\frac{u \text{ rdf:type rdf:Property } .}{u \text{ rdfs:subPropertyOf } u .}$	rdfs6	reflexivity
$\frac{a \text{ rdfs:subPropertyOf } b . u \text{ a } y .}{u \text{ b } y .}$	rdfs7	subproperty inferences for instances
$\frac{u \text{ rdf:type rdfs:Class } .}{u \text{ rdf:subClassOf rdfs:Resource } .}$	rdfs8	classes contain only resources



## RDFS Entailment

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

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

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

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

$$\frac{u \text{ rdf:type } \text{rdfs:Datatype} .}{u \text{ rdfs:subClassOf } \text{rdfs:Literal} .} \text{ rdfs10}$$
 every datatype is a  
subclass of `rdfs:Literal`

## RDFS Entailment

- important definition: XML clash

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

```
ex:evilRemark  ex:hasSmiley  ">:->"^^rdf:XMLLiteral .
```

- occurs if a node of type `rdfs:Literal` gets assigned an ill-formed literal value

## RDFS Entailment

### Theorem:

A graph  $G_2$  is RDFS entailed by  $G_1$ , if there is a graph  $G'_1$  obtained by applying the rules lg, gl, rdfsax, rdf1, rdf2, rdfs1 – rdfs13 and rdfsax to  $G_1$ , such that

- $G_2$  is simply entailed by  $G'_1$  or
- $G'_1$  contains an XML clash.

# Agenda

- 1 Motivation and Considerations
- 2 Simple Entailment
- 3 RDF Entailment
- 4 RDFS Entailment
- 5 Downsides of RDF(S)**

## What RDF(S) Cannot Do

- Certain seemingly sensible consequences are not RDFS-entailed, e.g.

```
ex:talksTo    rdfs:domain      ex:Homo .  
ex:Homo       rdfs:subClassOf  ex:Primates .
```

should imply

```
ex:talksTo    rdfs:domain      ex:Primates .
```

- possible solution: use a stronger, so-called “extensional” semantics (but this would be outside the standard)
- no possibility to express negation