<table>
<thead>
<tr>
<th>Course</th>
<th>Dates</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overview &amp; XML</td>
<td>11 APR DS5</td>
<td></td>
</tr>
<tr>
<td>Introduction into RDF</td>
<td>11 APR DS6</td>
<td></td>
</tr>
<tr>
<td>RDFS – Syntax &amp; Intuition</td>
<td>16 APR DS6</td>
<td></td>
</tr>
<tr>
<td>Tutorial 1</td>
<td>23 APR DS6</td>
<td></td>
</tr>
<tr>
<td>RDFS – Semantics</td>
<td>25 APR DS5</td>
<td></td>
</tr>
<tr>
<td>RDFS Rule-based Reasoning</td>
<td>25 APR DS6</td>
<td></td>
</tr>
<tr>
<td>Tutorial 2</td>
<td>30 APR DS6</td>
<td></td>
</tr>
<tr>
<td>SPARQL – Syntax &amp; Intuition</td>
<td>02 MAY DS5</td>
<td></td>
</tr>
<tr>
<td>SPARQL – Semantics</td>
<td>02 MAY DS6</td>
<td></td>
</tr>
<tr>
<td>SPARQL Algebra</td>
<td>09 MAY DS5</td>
<td></td>
</tr>
<tr>
<td>Tutorial 3</td>
<td>09 MAY DS6</td>
<td></td>
</tr>
<tr>
<td>OWL – Syntax &amp; Intuition</td>
<td>14 MAY DS6</td>
<td></td>
</tr>
<tr>
<td>OWL &amp; Description Logics</td>
<td>16 MAY DS5</td>
<td></td>
</tr>
<tr>
<td>OWL 2</td>
<td>16 MAY DS6</td>
<td></td>
</tr>
<tr>
<td>Tutorial 4</td>
<td>23 MAY DS5</td>
<td></td>
</tr>
<tr>
<td>Tableau I</td>
<td>23 MAY DS6</td>
<td></td>
</tr>
<tr>
<td>Tableau II</td>
<td>30 MAY DS5</td>
<td></td>
</tr>
<tr>
<td>Tutorial 5</td>
<td>30 MAY DS6</td>
<td></td>
</tr>
<tr>
<td>Hypertableau I</td>
<td>4 JUN DS6</td>
<td></td>
</tr>
<tr>
<td>Hypertableau II</td>
<td>6 JUN DS5</td>
<td></td>
</tr>
<tr>
<td>Tutorial 6</td>
<td>6 JUN DS6</td>
<td></td>
</tr>
<tr>
<td>SPARQL 1.1</td>
<td>18 JUN DS6</td>
<td></td>
</tr>
<tr>
<td>SPARQL Entailment</td>
<td>20 JUN DS5</td>
<td></td>
</tr>
<tr>
<td>Tutorial 7</td>
<td>20 JUN DS6</td>
<td></td>
</tr>
<tr>
<td>OWL &amp; Rules</td>
<td>25 JUN DS6</td>
<td></td>
</tr>
<tr>
<td>Ontology Editing</td>
<td>27 JUL DS5</td>
<td></td>
</tr>
<tr>
<td>Ontology Engineering</td>
<td>27 JUL DS6</td>
<td></td>
</tr>
<tr>
<td>Tutorial 8</td>
<td>2 JUL DS6</td>
<td></td>
</tr>
<tr>
<td>Linked Data &amp; Applications</td>
<td>4 JUL DS5</td>
<td></td>
</tr>
<tr>
<td>Q&amp;A Session</td>
<td>9 JUL DS6</td>
<td></td>
</tr>
<tr>
<td>Q&amp;A Session</td>
<td>11 JUL DS5</td>
<td></td>
</tr>
</tbody>
</table>
RDF Schema
RDF Schema
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:

  simple interpretations

  RDF interpretations

  RDFS interpretations
Semantics of RDF(S)

- we proceed stepwise:
  - simple interpretations
  - RDF interpretations
    - RDFS interpretations
  - 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 $\mathcal{I}$ as follows:

- Every untyped literal "a" is mapped to $a$: $("a")^\mathcal{I} = a$
- Every untyped literal with language information "a"@t is mapped to the pair $\langle a, t \rangle$, that is: $("a"@t)^\mathcal{I} = \langle a, t \rangle$,
- 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):

- Names
  - Literals
    - Untyped
    - Typed
  - IRIs

Vocabulary V

Interpretation I

Resources IR

Properties IP

I_E

TU Dresden, 25 April 2014  Foundations of Semantic Web Technologies
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!
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?

\[ \text{interpretation } \mathcal{I} = s \mathcal{I}_L p \mathcal{I}_S o \mathcal{I}_{\text{EXT}}. \]
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:

triple

\[ s \rightarrow p \rightarrow o \]
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 $_:\text{label}$ let
  
  $\text{(#label)}^{\mathcal{I}+A} = A(\text{_:label})$

- 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 $\mathcal{I}$:

\[
\begin{align*}
\text{IR} & = \{c, g, h, z, l, m, 1 \text{ lb}\} & \text{IS} & = \{ \text{ex:Chutney} \mapsto c, \text{ex:greenMango} \mapsto g, \text{ex:hasIngredient} \mapsto h, \text{ex:ingredient} \mapsto z, \text{ex:amount} \mapsto m \} \\
\text{IP} & = \{h, z, m\} \\
\text{LV} & = \{1 \text{ lb}\} \\
\text{I}_{\text{EXT}} & = h \mapsto \{\langle c, l \rangle\} \quad z \mapsto \{\langle l, g \rangle\} \\
& \quad m \mapsto \{\langle l, 1 \text{ lb} \rangle\} \\
\text{IL} & \text{ is the “empty function”}
\end{align*}
\]

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

TU Dresden, 25 April 2014    Foundations of Semantic Web Technologies
Simple Interpretations: Example

IR = \{c, g, h, z, l, m, 1 lb\}  \quad IS = \begin{array}{l}
\text{ex:Chutney} \mapsto c \\
\text{ex:greenMango} \mapsto g \\
\text{ex:hasIngredient} \mapsto h \\
\text{ex:ingredient} \mapsto z \\
\text{ex:amount} \mapsto m
\end{array}

LV = \{1 lb\}

I_{EXT} = h \mapsto \{c, l\} \quad z \mapsto \{l, g\} \quad m \mapsto \{l, 1 lb\}

I_L is the “empty function”

• If we pick \( A: \text{_:idl} \mapsto l \), then we get

\( \langle \text{ex:Chutney}^{\mathcal{I}+A}, \text{_:idl}^{\mathcal{I}+A} \rangle = \langle c, l \rangle \in I_{EXT}(h) = I_{EXT}(\text{ex:hasIngredient}^{\mathcal{I}+A}) \)

\( \langle \text{_:idl}^{\mathcal{I}+A}, \text{ex:greenMango}^{\mathcal{I}+A} \rangle = \langle l, g \rangle \in I_{EXT}(z) = I_{EXT}(\text{ex:ingredient}^{\mathcal{I}+A}) \)

\( \langle \text{_:idl}^{\mathcal{I}+A}, "1 \ lb"^{\mathcal{I}+A} \rangle = \langle l, 1 \ lb \rangle \in I_{EXT}(m) = I_{EXT}(\text{ex:amount}^{\mathcal{I}+A}) \)

• 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:

\[
\begin{array}{c}
\text{se1} \\
\frac{u \ a \ x .}{u \ a \ _\text{_:n} .}
\end{array}
\]

\[
\begin{array}{c}
\text{se2} \\
\frac{u \ a \ x .}{_\text{_:n} \ a \ x .}
\end{array}
\]

- 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

TU Dresden, 25 April 2014    Foundations of Semantic Web Technologies
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            ... 

In order to realize their intended semantics.
Conditions for RDF Interpretations

An RDF interpretation for a vocabulary V is a simple interpretation for the vocabulary V ∪ V_{RDF} that additionally satisfies the following conditions:

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

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

“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 \( V \cup V_{RDF} \) that additionally satisfies the following conditions:

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

“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"^^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;
   - $I_L("s"^^rdf:XMLLiteral) \in LV$;
   - $\langle I_L("s"^^rdf:XMLLiteral), rdf:XMLLiteral^I \rangle \in I_{EXT}(rdf:type^I)$
Conditions for RDF Interpretations

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

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;
   - $I_L("s"^^rdf:XMLLiteral) \in LV$;
   - $\langle I_L("s"^^rdf:XMLLiteral), rdf:XMLLiteral^I \rangle \in I_{EXT}(rdf:type^I)$
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;
   - $I_L("s"^^rdf:XMLLiteral) \in LV$;
   - $\langle I_L("s"^^rdf:XMLLiteral), rdf:XMLLiteral^I \rangle \in I_{EXT}(rdf:type^I)$

\[
\begin{align*}
\text{u a l} & \quad \text{lg} \quad \text{l a literal, } _::n \\
\text{u a }_::n & \quad \text{not bound otherwise}
\end{align*}
\]

\[
\begin{align*}
\text{u a }_::n & \quad \text{rdf2} \quad \text{If rule lg has assigned } _::n \text{ to the XML Literal l}
\end{align*}
\]
Conditions for RDF Interpretations

3. If "s"^^\texttt{rdf:XMLLiteral} is contained in V and s is an ill-formed XML literal, then
   - $I_L("s"^^\texttt{rdf:XMLLiteral}) \not\in LV$ and
   - $\langle I_L("s"^^\texttt{rdf:XMLLiteral}), \texttt{rdf:XMLLiteral}^I \rangle \not\in I_{\text{EXT}}(\texttt{rdf:type}^I)$. 
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 $\text{IP} \subseteq \text{IR}$).

- 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:

```
rdf:type            rdf:type            rdf:Property .
rdf:subject         rdf:type            rdf:Property .
rdf:.predicate      rdf:type            rdf:Property .
rdf:object          rdf:type            rdf:Property .
rdf:first           rdf:type            rdf:Property .
rdf:rest            rdf:type            rdf:Property .
rdf:value           rdf:type            rdf:Property .
rdf:_1               rdf:type            rdf:Property .
rdf:_2               rdf:type            rdf:Property .
...                  rdf:type            rdf:Property .
rdf:nil              rdf:type            rdf:List .
```

\[
\text{每一个阶义三元式“}\ u \ a \ x \text{.”}
\]

TU Dresden, 25 April 2014 — Foundations of Semantic Web Technologies
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

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

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: $I_{\text{CEXT}} : \mathbb{IR} \rightarrow 2^{\mathbb{IR}}$) by letting $I_{\text{CEXT}}(y)$ contain exactly those elements $x$, for which $(x, y)$ 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 $IC$ be the extension of the specific IRI $\text{rdfs:Class}$, hence: $IC = I_{\text{CEXT}}(\text{rdfs:Class}^\mathcal{I})$.

- note: both $I_{\text{CEXT}}$ as well as $IC$ are fully determined by $\mathcal{I}$ and $I_{\text{EXT}}$. 
RDFS Interpretations

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

- $IR = I_{CEXT}(rdfs:Resource^I)$
  Every resource is of type $rdfs:Resource$.

- $LV = I_{CEXT}(rdfs:Literal^I)$
  Every untyped and every well-formed typed literal is of type $rdfs:Literal$.

- If $\langle x, y \rangle \in I_{EXT}(rdfs:domain^I)$ and $\langle u, v \rangle \in I_{EXT}(x)$, then $u \in I_{CEXT}(y)$.
  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 \( \langle x, y \rangle \in I_{\text{EXT}}(\text{rdfs:range}^I) \) and \( \langle u, v \rangle \in I_{\text{EXT}}(x) \), then \( v \in I_{\text{CEXT}}(y) \). If the property \text{rdfs:range} connects \( x \) with \( y \) and the property \( x \) connects the resources \( u \) and \( v \), then \( v \) is of type \( y \).

- \( I_{\text{EXT}}(\text{rdfs:subPropertyOf}^I) \) is reflexive and transitive on IP. The \text{rdfs:subPropertyOf} property connects every property with itself. Moreover, if \text{rdfs:subPropertyOf} connects a property \( x \) with a property \( y \) and additionally \( y \) with a property \( z \), then \text{rdfs:subPropertyOf} also connects \( x \) directly with \( z \).
RDFS Interpretations

- If \( \langle x, y \rangle \in I_{\text{EXT}}(\text{rdfs:subPropertyOf}^\mathcal{I}) \),
  then \( x, y \in IP \) and \( I_{\text{EXT}}(x) \subseteq I_{\text{EXT}}(y) \).
  If \( \text{rdfs:subPropertyOf} \) 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 \( x \in IC \), then \( \langle x, \text{rdfs:Resource}^\mathcal{I} \rangle \in I_{\text{EXT}}(\text{rdfs:subClassOf}^\mathcal{I}) \).
  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 \( \text{rdfs:Resource} \) is in the extension of \( \text{rdfs:subClassOf} \).
RDFS Interpretations

- If \( \langle x, y \rangle \in I_{\text{EXT}}(\text{rdfs:subClassOf}^I) \), then \( x, y \in \text{IC} \) and 
  \( I_{\text{CEXT}}(x) \subseteq I_{\text{CEXT}}(y) \).
  If \( x \) and \( y \) are connected via the \text{rdfs:subClassOf} property, then both \( x \) and \( y \) are classes and the (class) extension of \( x \) is a subset of the (class) extension of \( y \).

- \( I_{\text{EXT}}(\text{rdfs:subClassOf}^I) \) is reflexive and transitive on \( \text{IC} \).
  The \text{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 \( x \in I_{\text{CEXT}}(\text{rdfs:ContainerMembershipProperty}^I) \), then \( \langle x, \text{rdfs:member}^I \rangle \in I_{\text{EXT}}(\text{rdfs:subPropertyOf}^I) \).

If \( x \) is a property of the type \text{rdfs:ContainerMembershipProperty}, then it is \text{rdfs:subPropertyOf}-connected with the property \text{rdfs:member}.

• If \( x \in I_{\text{CEXT}}(\text{rdfs:Datatype}^I) \), then \( \langle x, \text{rdfs:Literal}^I \rangle \in I_{\text{EXT}}(\text{rdfs:subClassOf}^I) \).

If some \( x \) is typed as element of the class \text{rdfs:Datatype}, then it must be a subclass of the class of all literal values (denoted by \text{rdfs:Literal}).

• ... additionally we require satisfaction of the following axiomatic triples:
### RDFS Interpretations

<table>
<thead>
<tr>
<th>Property</th>
<th>Domain</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>rdf:type</td>
<td>rdfs:domain</td>
<td>rdfs:Resource</td>
</tr>
<tr>
<td>rdfs:domain</td>
<td>rdfs:domain</td>
<td>rdf:Property</td>
</tr>
<tr>
<td>rdfs:range</td>
<td>rdfs:domain</td>
<td>rdf:Property</td>
</tr>
<tr>
<td>rdfs:subPropertyOf</td>
<td>rdfs:domain</td>
<td>rdf:Property</td>
</tr>
<tr>
<td>rdfs:subClassOf</td>
<td>rdfs:domain</td>
<td>rdfs:Class</td>
</tr>
<tr>
<td>rdf:subject</td>
<td>rdfs:domain</td>
<td>rdf:Statement</td>
</tr>
<tr>
<td>rdf:predicate</td>
<td>rdfs:domain</td>
<td>rdf:Statement</td>
</tr>
<tr>
<td>rdf:object</td>
<td>rdfs:domain</td>
<td>rdf:Statement</td>
</tr>
<tr>
<td>rdfs:member</td>
<td>rdfs:domain</td>
<td>rdfs:Resource</td>
</tr>
<tr>
<td>rdf:first</td>
<td>rdfs:domain</td>
<td>rdf:List</td>
</tr>
<tr>
<td>rdf:rest</td>
<td>rdfs:domain</td>
<td>rdf:List</td>
</tr>
<tr>
<td>rdfs:seeAlso</td>
<td>rdfs:domain</td>
<td>rdfs:Resource</td>
</tr>
<tr>
<td>rdfs:isDefinedBy</td>
<td>rdfs:domain</td>
<td>rdfs:Resource</td>
</tr>
<tr>
<td>rdfs:comment</td>
<td>rdfs:domain</td>
<td>rdfs:Resource</td>
</tr>
<tr>
<td>rdfs:label</td>
<td>rdfs:domain</td>
<td>rdfs:Resource</td>
</tr>
<tr>
<td>rdf:value</td>
<td>rdfs:domain</td>
<td>rdfs:Resource</td>
</tr>
</tbody>
</table>
RDFS Interpretations

<table>
<thead>
<tr>
<th>rdf:type</th>
<th>rdfs:range</th>
<th>rdfs:Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>rdfs:domain</td>
<td>rdfs:range</td>
<td>rdfs:Class</td>
</tr>
<tr>
<td>rdfs:range</td>
<td>rdfs:range</td>
<td>rdfs:Class</td>
</tr>
<tr>
<td>rdfs:subPropertyOf</td>
<td>rdfs:range</td>
<td>rdf:Property</td>
</tr>
<tr>
<td>rdfs:subClassOf</td>
<td>rdfs:range</td>
<td>rdfs:Class</td>
</tr>
<tr>
<td>rdf:subject</td>
<td>rdfs:range</td>
<td>rdfs:Resource</td>
</tr>
<tr>
<td>rdf:predicate</td>
<td>rdfs:range</td>
<td>rdfs:Resource</td>
</tr>
<tr>
<td>rdf:object</td>
<td>rdfs:range</td>
<td>rdfs:Resource</td>
</tr>
<tr>
<td>rdfs:member</td>
<td>rdfs:range</td>
<td>rdfs:Resource</td>
</tr>
<tr>
<td>rdf:first</td>
<td>rdfs:range</td>
<td>rdfs:Resource</td>
</tr>
<tr>
<td>rdf:rest</td>
<td>rdfs:range</td>
<td>rdf:List</td>
</tr>
<tr>
<td>rdfs:seeAlso</td>
<td>rdfs:range</td>
<td>rdfs:Resource</td>
</tr>
<tr>
<td>rdfs:isDefinedBy</td>
<td>rdfs:range</td>
<td>rdfs:Resource</td>
</tr>
<tr>
<td>rdfs:comment</td>
<td>rdfs:range</td>
<td>rdfs:Literal</td>
</tr>
<tr>
<td>rdfs:label</td>
<td>rdfs:range</td>
<td>rdfs:Literal</td>
</tr>
<tr>
<td>rdf:value</td>
<td>rdfs:range</td>
<td>rdfs:Resource</td>
</tr>
</tbody>
</table>
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 .
 rdf:_1                   rdfs:domain        rdfs:Resource .
 rdf:_1                   rdfs:range         rdfs:Resource .
 rdf:_2                   rdf:type              rdfs:ContainerMembershipProperty .
RDFS Entailment

Automatic inference is again realized via deduction rules:

\[
\begin{align*}
\text{rdfsax} & \quad \text{every axiomatic triple } u a x . \\
\hline
u a x . & \text{can always be derived} \\
\hline
u a _ : n . & \text{the converse of Rule lg: } _ : n \text{ has been assigned (via Rule lg)} \\
\hline
u a l . & \text{to the untyped literal } l \\
\hline
\hline
u a l . & \text{rdfs1: } _ : n \text{ has been assigned (via Rule lg) to the untyped literal } l \\
\hline
\hline
a \text{ rdfs:domain } x . & \text{rdfs2: implements the semantics of property domains} \\
\hline
u a y . & \text{v rdf:type } x . \\
\hline
u \text{ rdf:type } x . & \text{v rdf:type } x . \\
\hline
\hline
a \text{ rdfs:range } x . & \text{rdfs3: implements the semantics of property ranges} \\
\hline
u a v . & \text{v rdf:type } x . \\
\hline
\hline
\end{align*}
\]

\[\begin{align*}
a, b & \text{ IRIs} \\
u, v & \text{IRI or blank node} \\
x, y & \text{IRI, blank node or literal} \\
l & \text{literal} \\
_ : n & \text{blank nodes}
\end{align*}\]
RDFS Entailment

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

\[
\begin{align*}
&u \text{ rdfs:subClassOf } x \ . \ v \text{ rdf:type } u \ . \\
&\quad v \text{ rdf:type } x \ . \\
&u \text{ rdf:type rdfs:Class} \ . \ u \text{ rdfs:subClassOf } u \ . \\
&u \text{ rdfs:subClassOf } v \ . \ v \text{ rdfs:subClassOf } x \ . \\
&\quad u \text{ rdfs:subClassOf } x \ . \\
&u \text{ rdf:type rdfs:ContainerMembershipProperty} \ . \ u \text{ rdfs:subPropertyOf rdfs:member} \ . \\
&u \text{ rdf:type rdfs:Datatype} \ . \ u \text{ rdfs:subClassOf rdfs:Literal} \ . \\
&\text{rdfs9 subclassen inferences for instances} \\
&\text{rdfs10 reflexivity} \\
&\text{rdfs11 transitivity} \\
&\text{rdfs12} \\
&\text{rdfs10 every datatype is a subclass of rdfs:Literal}
\end{align*}
\]
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, rdfax, 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