FOUNDATIONS OF SEMANTIC WEB TECHNOLOGIES

Semantics of RDF(S)

Sebastian Rudolph
Semantics of RDF(S)
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

\[(s, p, o) \in (\text{IRI} \cup \text{bnodes}) \times \text{IRI} \times (\text{IRI} \cup \text{bnodes} \cup \text{literals})\]

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

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

<table>
<thead>
<tr>
<th>names</th>
<th>literals</th>
</tr>
</thead>
<tbody>
<tr>
<td>untyped</td>
<td>IRIs</td>
</tr>
<tr>
<td>typed</td>
<td></td>
</tr>
</tbody>
</table>

IRI(s)

resources

properties

Vocabulary V

interpretation I

I

I_L

I_S

I_EXT
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?

\[ \text{if all subject, predicate, and object are contained in } V \text{ and additionally } \langle s, o \rangle \in I_{\text{EXT}}(p) \text{ holds} \]

http://example.org/SemanticWeb

http://springer.com/publisher

IRIs

names

literals

untyped

typed

IRIs

untyped

typed

LV

resources

IR

properties

IP

I_{\text{EXT}}

I

I_{L}

I_{S}

http://example.org/publishedBy

http://example.org/SemanticWeb

http://springer.com/publisher

TU Dresden

Foundations of Semantic Web Technologies
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
  \( \langle s^\mathcal{I}, o^\mathcal{I} \rangle \in I_{\text{EXT}}(p^\mathcal{I}) \) holds
Semantics of Simple Entailment

schematically:

\[
\text{triple} \quad s \quad p \quad o.
\]
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
  $\left( _:\text{label} \right)^{\mathcal{I}+A} = A(_:\text{label})$
- 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 $\mathcal{I}$:

$$
\begin{align*}
\text{IR} &= \{c, g, h, z, l, m, 1 \text{ lb} \} \\
\text{IP} &= \{h, z, m\} \\
\text{LV} &= \{1 \text{ lb}\} \\
\text{I}_{\text{EXT}} &= h \mapsto \{\langle c, l \rangle\} \\
&\quad z \mapsto \{\langle l, g \rangle\} \\
&\quad m \mapsto \{\langle l, 1 \text{ lb} \rangle\} \\
\text{IL} &= \text{is the “empty function”}
\end{align*}
$$

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

TU Dresden 
Foundations of Semantic Web Technologies
Simple Interpretations: Example

\[
IR = \{c, g, h, z, l, m, 1 \text{ lb}\} \\
IS = \text{ex:Chutney} \mapsto c \\
IP = \{h, z, m\} \\
LV = \{1 \text{ lb}\} \\
IEXT = h \mapsto \{(c, l)\} \\
\quad z \mapsto \{(l, g)\} \\
\quad m \mapsto \{(l, 1 \text{ lb})\} \\
IL is the \text{“empty function”}
\]

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

\[
\langle \text{ex:Chutney}^{A+}, \_\text{id1}^{A+} \rangle = \langle c, l \rangle \in IEXT(h) = IEXT(\text{ex:hasIngredient}^{A+}) \\
\langle \_\text{id1}^{A+}, \text{ex:greenMango}^{A+} \rangle = \langle l, g \rangle \in IEXT(z) = IEXT(\text{ex:ingredient}^{A+}) \\
\langle \_\text{id1}^{A+}, "1 \text{ lb}"^{A+} \rangle = \langle l, 1 \text{ lb} \rangle \in IEXT(m) = IEXT(\text{ex:amount}^{A+})
\]

- Therefore, \(I\) 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:

\[
\begin{align*}
\text{se1} & : \quad \frac{u \ a \ x}{u \ a \ \_:n} \\
\text{se2} & : \quad \frac{u \ a \ x}{\_:n \ a \ x}
\end{align*}
\]

- 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

TU Dresden

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 $V \cup V_{RDF}$ that additionally satisfies the following conditions:

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

"For every triple predicate we can infer that it is a 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 $\langle x, \text{rdf:Property}^I \rangle \in I_{\text{EXT}}(\text{rdf:type}^I)$.

“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.”

\[
\begin{array}{c}
\text{u a y} \\
\hline
\text{a rdf:type rdf:Property} \\
\end{array}
\text{ rdf1}
\]
Conditions for RDF Interpretations

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

\[
\begin{array}{c}
\text{l rdf:type rdf:XMLLiteral} \\
\hline
\text{u a l}
\end{array}
\]

\( l \text{ a well-formed XML literal} \)
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 \( s \);
   - \( I_L("s"^^rdf:XMLLiteral) \in LV \);
   - \( \langle I_L("s"^^rdf:XMLLiteral), rdf:XMLLiteral^I \rangle \in I_{EXT}(rdf:type^I) \)

Oops, literals must not occur in subject position!
2. If $s^{\text{\textit{\textsc{rdf}}:XMLLiteral}}$ is contained in $V$ and $s$ is a well-formed XML literal, then
   
   - $I_L(s^{\text{\textit{\textsc{rdf}}:XMLLiteral}})$ is the XML value of $s$;
   - $I_L(s^{\text{\textit{\textsc{rdf}}:XMLLiteral}}) \in LV$;
   - $\langle I_L(s^{\text{\textit{\textsc{rdf}}:XMLLiteral}}), \text{\textit{\textsc{rdf}}:XMLLiteral}^I \rangle \in I_{\text{\textsc{EXT}}(\text{\textit{\textsc{rdf}}:type}^I)}$
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 \( s \);
   - \( I_L("s"^^rdf:XMLLiteral) \) ∈ \( LV \);
   - \( \langle I_L("s"^^rdf:XMLLiteral), rdf:XMLLiteral^I \rangle \) ∈ \( I_{EXT}(rdf:type^I) \)

\[ \begin{array}{c}
\text{ua1} \\
\underline{ua2:n} \\
\text{lg} \quad \text{1 a literal, } _:n \\
\text{not bound otherwise} \\
\text{rdf2} \\
\underline{_:n rdf:type rdf:XMLLiteral} \\
\end{array} \]
3. If "$s"^^rdf:XMLLiteral is contained in V and $s$ is an ill-formed XML literal, then
   - $I_L("s"^^rdf:XMLLiteral) \not\in LV$ and
   - $\langle I_L("s"^^rdf:XMLLiteral), rdf:XMLLiteral^\mathcal{I}\rangle \not\in I_{EXT}(rdf:type^\mathcal{I})$. 

TU Dresden Foundations of Semantic Web Technologies
RDF Interpretations

• Note: $x$ is a property exactly if it is linked to the resource denoted by \texttt{rdf:Property} via the \texttt{rdf:type} property (this has the direct consequence that in every RDF interpretation holds $\text{IP} \subseteq \text{IR}$).

• The value space of the \texttt{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:nil rdf:type rdf:List .
```

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

\[
\begin{align*}
\text{rdfs:domain} & \quad \text{rdfs:range} & \quad \text{rdfs:Resource} \\
\text{rdfs:Literal} & \quad \text{rdfs:Datatype} & \quad \text{rdfs:Class} \\
\text{rdfs:subClassOf} & \quad \text{rdfs:subPropertyOf} & \quad \text{rdfs:Container} \\
\text{rdfs:member} & \quad \text{rdfs:ContainerMembershipProperty} \\
\text{rdfs:comment} & \quad \text{rdfs:seeAlso} & \quad \text{rdfs:isDefinedBy} \\
\text{rdfs:label} & \\
\end{align*}
\]

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 $\langle x, y \rangle$ 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 $I_C$ be the extension of the specific IRI $\text{rdfs:Class}$, hence: $I_C = I_{\text{CEXT}}(\text{rdfs:Class}^\mathcal{I})$.
- note: both $I_{\text{CEXT}}$ as well as $I_C$ 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 $V \cup V_{\text{RDFS}}$, that additionally satisfies the following criteria:

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

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

- If $\langle x, y \rangle \in \text{I}_{\text{EXT}}(\text{rdfs:domain}^I)$ and $\langle u, v \rangle \in \text{I}_{\text{EXT}}(x)$, then $u \in \text{I}_{\text{CEXT}}(y)$.
  
  If the property $\text{rdfs:domain}$ 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 \( x, y \in \text{IP} \) 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 \( x \in \text{IC} \), 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 \((x, y) \in I_{\text{EXT}}(\text{rdfs:subClassOf}^\mathcal{I})\), then \(x, y \in \mathcal{IC}\) 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 \(y\).

- \(I_{\text{EXT}}(\text{rdfs:subClassOf}^\mathcal{I})\) is reflexive and transitive on \(\mathcal{IC}\).
  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 $x \in I^I_{CEXT}(rdfs:\text{ContainerMembershipProperty}^I)$, then 
  $\langle x, rdfs:\text{member}^I \rangle \in I^I_{EXT}(rdfs:\text{subPropertyOf}^I)$. 
  If $x$ is a property of the type $rdfs:\text{ContainerMembershipProperty}$, 
  then it is $rdfs:\text{subPropertyOf}$-connected with the property $rdfs:\text{member}$.

- If $x \in I^I_{CEXT}(rdfs:\text{Datatype}^I)$, then 
  $\langle x, rdfs:\text{Literal}^I \rangle \in I^I_{EXT}(rdfs:\text{subClassOf}^I)$. 
  If some $x$ is typed as element of the class $rdfs:\text{Datatype}$, then it must 
  be a subclass of the class of all literal values (denoted by $rdfs:\text{Literal}$).

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

rdf:type rdfs:domain rdfs:Resource.
rdfs:domain rdfs:domain rdf:Property.
rdfs:range rdfs:domain rdf:Property.
rdfs:subPropertyOf rdfs:domain rdf:Property.
rdfs:subClassOf rdfs:domain rdfs:Class.
rdfs:member rdfs:domain rdfs:Resource.
rdf:first rdfs:domain rdf:List.
rdf:rest rdfs:domain rdf:List.
rdfs:seeAlso rdfs:domain rdfs:Resource.
rdfs:isDefinedBy rdfs:domain rdfs:Resource.
rdfs:comment rdfs:domain rdfs:Resource.
rdfs:label rdfs:domain rdfs:Resource.
**RDFS Interpretations**

- rdf:type \(\text{rdfs:range rdfs:Class .}\)
- rdfs:domain \(\text{rdfs:range rdfs:Class .}\)
- rdfs:range \(\text{rdfs:range rdfs:Class .}\)
- rdfs:subPropertyOf \(\text{rdfs:range rdf:Property .}\)
- rdfs:subClassOf \(\text{rdfs:range rdfs:Class .}\)
- rdf:subject \(\text{rdfs:range rdfs:Resource .}\)
- rdf:predicate \(\text{rdfs:range rdfs:Resource .}\)
- rdf:object \(\text{rdfs:range rdfs:Resource .}\)
- rdfs:member \(\text{rdfs:range rdfs:Resource .}\)
- rdf:first \(\text{rdfs:range rdfs:Resource .}\)
- rdf:rest \(\text{rdfs:range rdf:List .}\)
- rdfs:seeAlso \(\text{rdfs:range rdfs:Resource .}\)
- rdfs:isDefinedBy \(\text{rdfs:range rdfs:Resource .}\)
- rdfs:comment \(\text{rdfs:range rdfs:Literal .}\)
- rdfs:label \(\text{rdfs:range rdfs:Literal .}\)
- rdf:value \(\text{rdfs:range rdfs:Resource .}\)
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:

\[
\begin{align*}
\text{rdfsax: } & \quad \text{every axiomatic triple } \text{"u a x ." can always be derived,} \\
\text{u a x .} & \quad \text{u a } \_ : \text{n .} \\
\text{u a l .} & \quad \text{The converse of Rule lg: } \_ : \text{n has been assigned (via Rule lg) to the untyped literal l.}
\end{align*}
\]

\[
\begin{align*}
\text{u a l .} & \quad \text{_:n rdf:type rdfs:Literal.} \\
\text{a rdfs:domain x . u a y .} & \quad \text{u rdf:type x .} \\
\text{a rdfs:range x . u a v .} & \quad \text{v rdf:type x .}
\end{align*}
\]

\[
\begin{align*}
\text{rdfs1: } & \quad \_ : \text{n has been assigned (via Rule lg) to the untyped literal l.} \\
\text{rdfs2: } & \quad \text{implements the semantics of property domains.}
\end{align*}
\]

\[
\begin{align*}
\text{rdfs3: } & \quad \text{implements the semantics of property ranges.}
\end{align*}
\]

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

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

\[
\frac{u \text{ rdfs:subClassOf } x . \quad v \text{ rdf:type } u . \quad \text{rdfs9}}{v \text{ rdf:type } x . \quad \text{subclassen inferences for instances}}
\]

\[
\frac{u \text{ rdf:type rdfs:Class} \quad \text{rdfs10}}{u \text{ rdfs:subClassOf } u . \quad \text{reflexivity}}
\]

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

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

\[
\frac{u \text{ rdf:type rdfs:Datatype} \quad \text{rdfs10}}{u \text{ rdfs:subClassOf rdfs:Literal} . \quad \text{every datatype is a subclass of rdfs:Literal}}
\]
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, 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.

  \[
  \text{ex:talksTo rdfs:domain ex:Homo}.
  \]
  \[
  \text{ex:Homo rdfs:subClassOf ex:Primates}.
  \]

  should imply

  \[
  \text{ex:talksTo rdfs:domain ex:Primates}.
  \]

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