TECHNISCHE UNIVERSITÄT DRESDEN

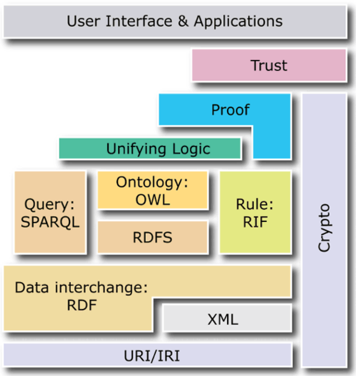
FOUNDATIONS OF SEMANTIC WEB TECHNOLOGIES

Semantics of RDF(S)

Sebastian Rudolph

DRESDEN concept
Exzellenz aus Wissenschaft und Kultur

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

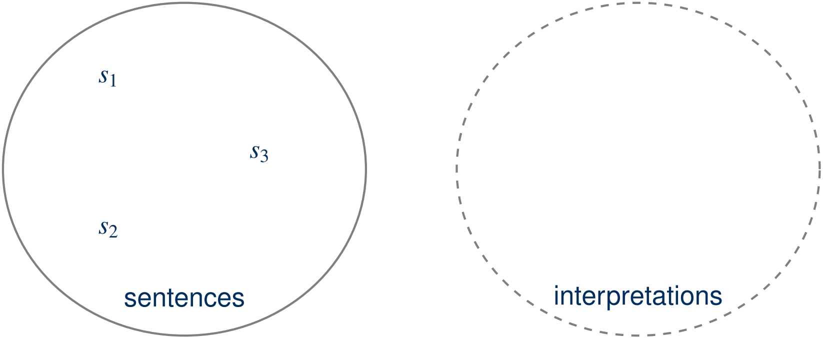
    $$(\mathrm{s}, \mathrm{p}, \mathrm{o}) \in (\mathrm{IRI} \cup \mathrm{bnode}) \times \mathrm{IRI} \times (\mathrm{IRI} \cup \mathrm{bnode} \cup \mathrm{literal})$$

    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?

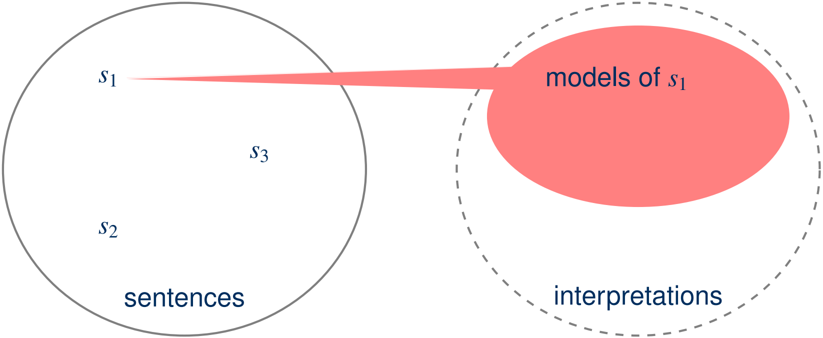
- consequence relation that defines when an RDF(S) graph G' logically follows from an RDF(S) graph G, i.e. $G \models G'$
- model-theoretic semantics: we define a set of interpretations and stipulate under which conditions an interpretation is a model of a graph



How is RDF(S) Linked to a Logic?

What is the semantics?

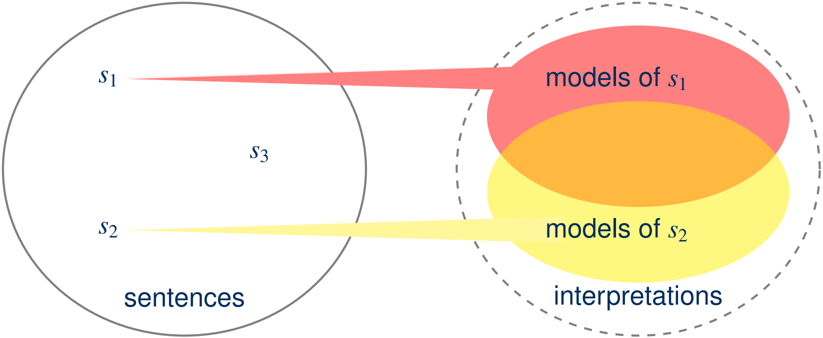
- consequence relation that defines when an RDF(S) graph G' logically follows from an RDF(S) graph G, i.e. $G \models G'$
- model-theoretic semantics: we define a set of interpretations and stipulate under which conditions an interpretation is a model of a graph



How is RDF(S) Linked to a Logic?

What is the semantics?

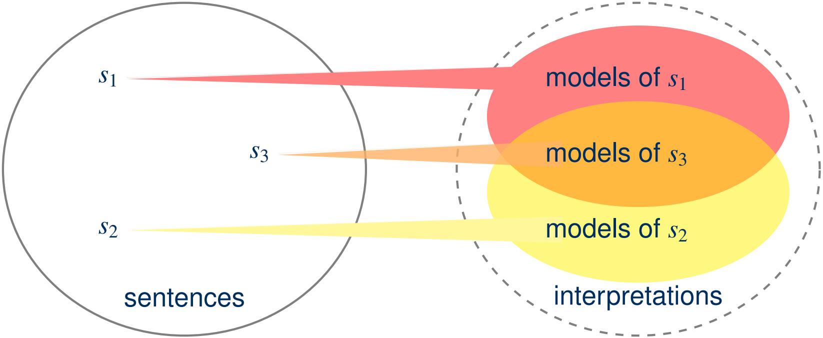
- consequence relation that defines when an RDF(S) graph G' logically follows from an RDF(S) graph G, i.e. $G \models G'$
- model-theoretic semantics: we define a set of interpretations and stipulate under which conditions an interpretation is a model of a graph



How is RDF(S) Linked to a Logic?

What is the semantics?

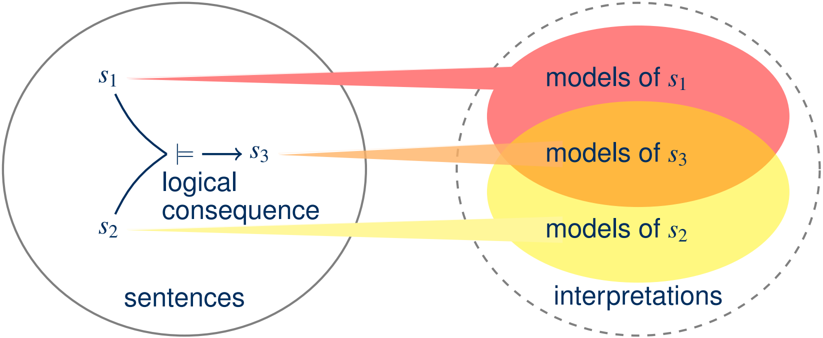
- consequence relation that defines when an RDF(S) graph G' logically follows from an RDF(S) graph G, i.e. $G \models G'$
- model-theoretic semantics: we define a set of interpretations and stipulate under which conditions an interpretation is a model of a graph



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)

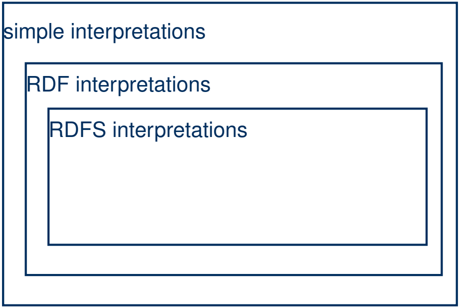
- we proceed stepwise:

simple interpretations

RDF interpretations

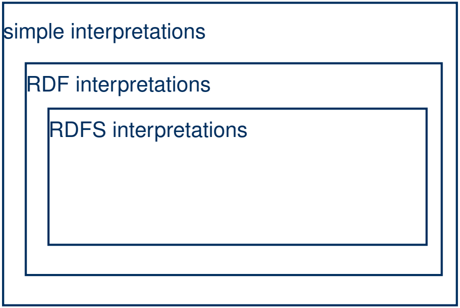
Semantics of RDF(S)

- 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 $\mathcal{I}$;
- $\mathrm{I_S}$, a function, mapping IRIs from V to the union of the sets IR and IP, i.e., $\mathrm{I_S} : \mathrm{V} \to \mathrm{IR} \cup \mathrm{IP}$,
- $\mathrm{I_{EXT}}$, a function, mapping every property to a set of pairs from IR, i.e., $\mathrm{I_{EXT}} : \mathrm{IP} \to 2^{\mathrm{IR} \times \mathrm{IR}}$ and
- $\mathrm{I_L}$, 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 $\mathcal{I}$
- $\mathsf{I}_{\mathsf{EXT}}(p)$ is also referred to as the extension of the property p

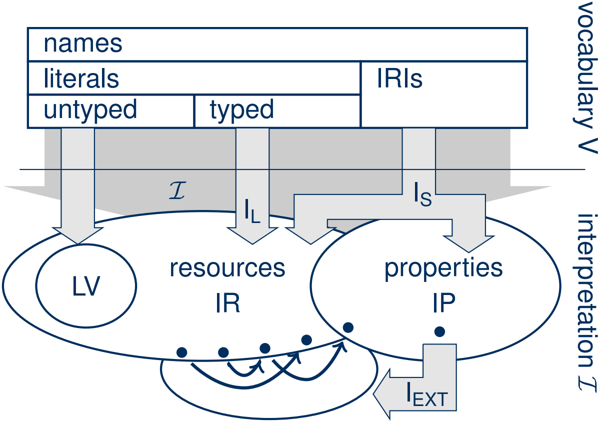
Definition (interpretation function)

based on I_{L} and I_{S}, we define $\cdot^{\mathcal{I}}$ as follows:

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

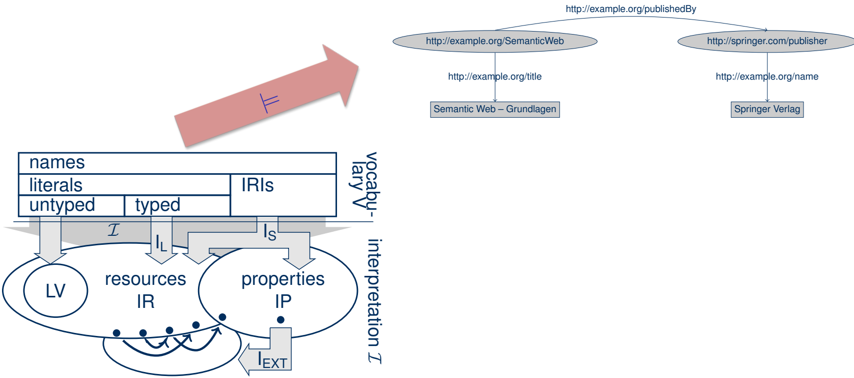
Semantics of the Simple Entailment

Interpretation (schematic):



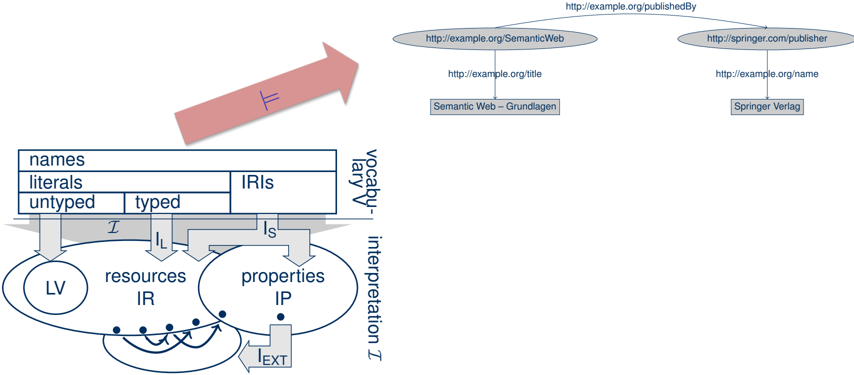
Semantics of the Simple Entailment

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



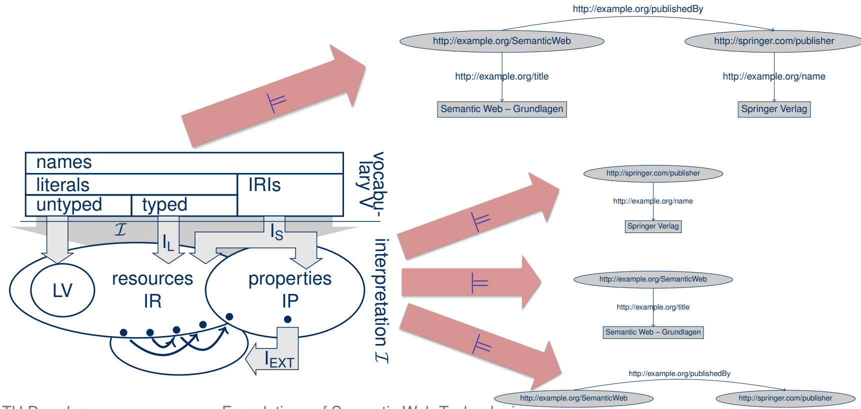
Semantics of the Simple Entailment

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



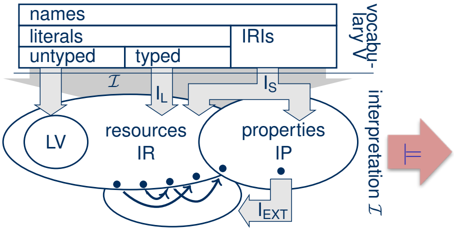
Semantics of the Simple Entailment

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

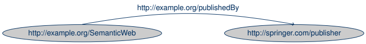


Semantics of the Simple Entailment

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

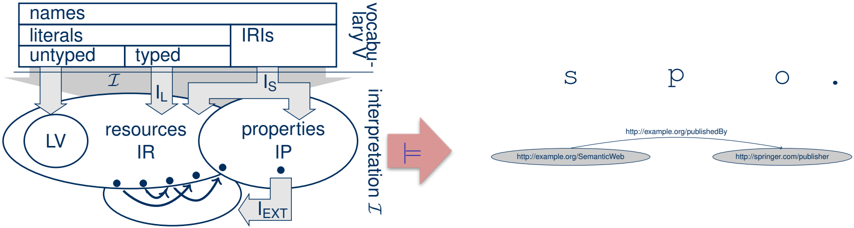


s p o .



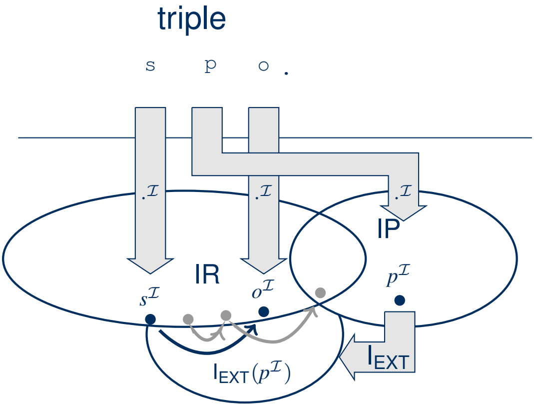
Semantics of the Simple Entailment

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



Semantics of Simple Entailment

schematically:

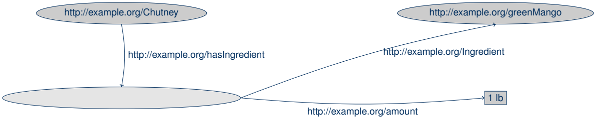


Semantics of Simple Entailment

- ...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 $(\texttt{\_:label})^{\mathcal{I}+A} = A(\texttt{\_:label})$
- 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:

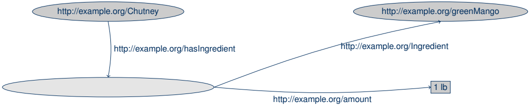


and interpretation $\mathcal{I}$:

$$
\begin{array}{rcl}
\mathrm{IR} & = & \{c, g, h, z, l, m, \text{1 lb}\} \\
\mathrm{IP} & = & \{h, z, m\} \\
\mathrm{LV} & = & \{\text{1 lb}\} \\
\mathrm{I}_{\mathrm{EXT}} & = & h \mapsto \{\langle c, l\rangle\} \\
 & & z \mapsto \{\langle l, g\rangle\} \\
 & & m \mapsto \{\langle l, \text{1 lb}\rangle\}
\end{array}
\qquad
\begin{array}{rcll}
\mathrm{I}_{\mathrm{S}} & = & \texttt{ex:Chutney} & \mapsto c \\
 & & \texttt{ex:greenMango} & \mapsto g \\
 & & \texttt{ex:hasIngredient} & \mapsto h \\
 & & \texttt{ex:ingredient} & \mapsto z \\
 & & \texttt{ex:amount} & \mapsto m \\
\mathrm{I}_{\mathrm{L}} & & \text{is the “empty function”} &
\end{array}
$$

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

Simple Interpretations: Example



$$
\begin{array}{rcl}
\mathsf{IR} & = & \{c, g, h, z, l, m, \text{1 lb}\} \\
\mathsf{IP} & = & \{h, z, m\} \\
\mathsf{LV} & = & \{\text{1 lb}\} \\
\mathsf{I}_{\mathsf{EXT}} & = & h \mapsto \{\langle c, l\rangle\} \\
 & & z \mapsto \{\langle l, g\rangle\} \\
 & & m \mapsto \{\langle l, \text{1 lb}\rangle\}
\end{array}
\qquad
\begin{array}{rcll}
\mathsf{I}_{\mathsf{S}} & = & \texttt{ex:Chutney} & \mapsto c \\
 & & \texttt{ex:greenMango} & \mapsto g \\
 & & \texttt{ex:hasIngredient} & \mapsto h \\
 & & \texttt{ex:ingredient} & \mapsto z \\
 & & \texttt{ex:amount} & \mapsto m \\
\mathsf{I}_{\mathsf{L}} & & \text{is the “empty function”} &
\end{array}
$$

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

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

- 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:

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

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

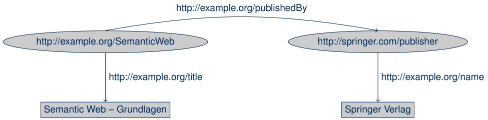
- precondition for applying this rule: the bnode has not already been associated with another IRI or literal

Simple Entailment

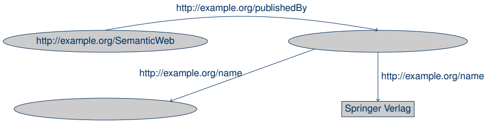
Theorem

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

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

Conditions for RDF Interpretations

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

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

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

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

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

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

Conditions for RDF Interpretations

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

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

Conditions for RDF Interpretations

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

$$\frac{\texttt{u a l}}{\texttt{l rdf:type rdf:XMLLiteral}}$$??? für `l` 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
   - $\mathrm{I_L}(\texttt{"s"\^{}\^{}rdf:XMLLiteral})$ is the XML value of `s`;
   - $\mathrm{I_L}(\texttt{"s"\^{}\^{}rdf:XMLLiteral}) \in \mathrm{LV}$;
   - $\langle \mathrm{I_L}(\texttt{"s"\^{}\^{}rdf:XMLLiteral}), \texttt{rdf:XMLLiteral}^{\mathcal{I}} \rangle \in \mathrm{I_{EXT}}(\texttt{rdf:type}^{\mathcal{I}})$

Conditions for RDF Interpretations

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

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

`l` a literal, `_:n` not bound otherwise

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

If rule lg has assigned `_:n` to the XML Literal `l`

Conditions for RDF Interpretations

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

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 $\mathrm{IP} \subseteq \mathrm{IR}$).
- 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 .
```

$$\frac{}{\texttt{u a x}}\ \text{rdfax}$$

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

```
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.

TECHNISCHE UNIVERSITÄT DRESDEN

RDFS Interpretations

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

RDFS Interpretations

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

- $\mathsf{IR} = \mathsf{I}_{\mathsf{CEXT}}(\texttt{rdfs:Resource}^{\mathcal{I}})$
  Every resource is of type `rdfs:Resource`.
- $\mathsf{LV} = \mathsf{I}_{\mathsf{CEXT}}(\texttt{rdfs:Literal}^{\mathcal{I}})$
  Every untyped and every well-formed typed literal is of type `rdfs:Literal`.
- If $\langle x, y \rangle \in \mathsf{I}_{\mathsf{EXT}}(\texttt{rdfs:domain}^{\mathcal{I}})$ and $\langle u, v \rangle \in \mathsf{I}_{\mathsf{EXT}}(x)$, then $u \in \mathsf{I}_{\mathsf{CEXT}}(y)$.
  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 $\langle x, y\rangle \in \mathsf{I}_{\mathsf{EXT}}(\texttt{rdfs:range}^{\mathcal{I}})$ and $\langle u, v\rangle \in \mathsf{I}_{\mathsf{EXT}}(x)$, then $v \in \mathsf{I}_{\mathsf{CEXT}}(y)$.
  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.
- $\mathsf{I}_{\mathsf{EXT}}(\texttt{rdfs:subPropertyOf}^{\mathcal{I}})$ 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 $\langle x, y\rangle \in \mathsf{I}_{\mathsf{EXT}}(\texttt{rdfs:subPropertyOf}^{\mathcal{I}})$,
  then $x, y \in \mathsf{IP}$ and $\mathsf{I}_{\mathsf{EXT}}(x) \subseteq \mathsf{I}_{\mathsf{EXT}}(y)$.
  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 $x \in \mathsf{IC}$, then $\langle x, \texttt{rdfs:Resource}^{\mathcal{I}}\rangle \in \mathsf{I}_{\mathsf{EXT}}(\texttt{rdfs:subClassOf}^{\mathcal{I}})$.
  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 $\langle x, y \rangle \in \mathsf{I}_{\mathsf{EXT}}(\texttt{rdfs:subClassOf}^{\mathcal{I}})$, then $x, y \in$ IC and $\mathsf{I}_{\mathsf{CEXT}}(x) \subseteq \mathsf{I}_{\mathsf{CEXT}}(y)$.
  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.
- $\mathsf{I}_{\mathsf{EXT}}(\texttt{rdfs:subClassOf}^{\mathcal{I}})$ is reflexive and transitive on 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 $x \in \mathsf{I}_{\mathsf{CEXT}}(\texttt{rdfs:ContainerMembershipProperty}^{\mathcal{I}})$, then $\langle x, \texttt{rdfs:member}^{\mathcal{I}} \rangle \in \mathsf{I}_{\mathsf{EXT}}(\texttt{rdfs:subPropertyOf}^{\mathcal{I}})$.
  If x is a property of the type `rdfs:ContainerMembershipProperty`, then it is `rdfs:subPropertyOf`-connected with the property `rdfs:member`.
- If $x \in \mathsf{I}_{\mathsf{CEXT}}(\texttt{rdfs:Datatype}^{\mathcal{I}})$, then $\langle x, \texttt{rdfs:Literal}^{\mathcal{I}} \rangle \in \mathsf{I}_{\mathsf{EXT}}(\texttt{rdfs:subClassOf}^{\mathcal{I}})$.
  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

```
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 .
rdf:subject         rdfs:domain    rdf:Statement .
rdf:predicate       rdfs:domain    rdf:Statement .
rdf:object          rdfs:domain    rdf:Statement .
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 .
rdf:value           rdfs:domain    rdfs:Resource .
```

RDFS Interpretations

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

$$\frac{}{\texttt{u a x .}}\ \text{rdfsax}$$

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

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

the converse of Rule lg: `_:n` has been assigned (via Rule lg) to the untyped literal `l`

$$\frac{\texttt{u a l .}}{\texttt{\_:n rdf:type rdfs:Literal}}\ \text{rdfs1}$$

`_:n` has been assigned (via Rule lg) to the untyped literal `l`

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

implements the semantics of property domains

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

implementis the semantics of property ranges

`a, b`	IRIs	`x, y`	IRI, blank node or literal		
`u, v`	IRI or blank node	`l`	literal	`_:n`	blank nodes

TECHNISCHE UNIVERSITÄT DRESDEN

RDFS Entailment

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

the subject of every triple is a resource

$$\frac{\texttt{u a v .}}{\texttt{v rdf:type rdfs:Resource .}}\ \text{rdfs4b}$$

objects that are not literals are resources as well

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

transitivity

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

reflexivity

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

subproperty inferences for instances

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

classes contain only resources

RDFS Entailment

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

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

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

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

$$\frac{\texttt{u rdf:type rdfs:Datatype .}}{\texttt{u rdfs:subClassOf rdfs:Literal .}}\ \text{rdfs10}$$ 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, 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