Basics of RDFS Formal Semantics

Camilo Thorne

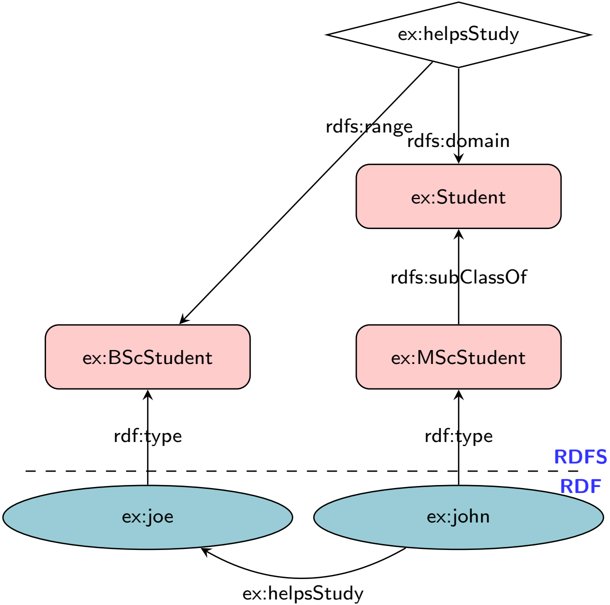
Room 00.012
Institut für Maschinelle Sprachverarbeitung
Universität Stuttgart
+49 (0) 711 685-81369
camilo.thorne@ims.uni-stuttgart.de

Semantic Web, SS 2017

Outline

Institut für Maschinelle Sprachverarbeitung

The RDFS Graphical Models



Semantics of (Mini) RDFS

One can define a **set-theoretic semantics** for the fragment of RDFS seen in course

Semantics of (Mini) RDFS

One can define a **set-theoretic semantics** for the fragment of RDFS seen in course

An RDFS (set-theoretic) **structure** is a tuple $\mathcal{I} = (D; P; V_1, \ldots, V_k; R)$ over the following domains:

- the base domain D of individual resources
- the domain $P \subseteq \mathcal{P}(D)$ of sets or classes
- the concrete datatype domains $V_1, \ldots, V_k$
- the domain $R \subseteq \mathcal{P}(D) \times \mathcal{P}(D \cup V_1 \cup \cdots \cup V_k)$ of relations or properties

Institut für Maschinelle Sprachverarbeitung

Semantics of (Mini) RDFS

One can define a **set-theoretic semantics** for the fragment of RDFS seen in course

> An RDFS (set-theoretic) **structure** is a tuple $\mathcal{I} = (D; P; V_1, \ldots, V_k; R)$ over the following domains:
>
> - the base domain D of individual resources
> - the domain $P \subseteq \mathcal{P}(D)$ of sets or classes
> - the concrete datatype domains $V_1, \ldots, V_k$
> - the domain $R \subseteq \mathcal{P}(D) \times \mathcal{P}(D \cup V_1 \cup \cdots \cup V_k)$ of relations or properties

In practice, one needs to set only the **base domain** D

Institut für Maschinelle Sprachverarbeitung

Semantics of (Mini) RDFS

An **interpretation function** $\cdot^{\mathcal{I}}$ maps every RDFS resource and litteral to an element in the domains of RDFS model $\mathcal{I}$

More formally:

$$\texttt{ri}^{\mathcal{I}} = \begin{cases} P_i \in P, & \text{if } \texttt{ri rdf:type rdfs:Class.} \text{ is a triple} \\ R_i \in R, & \text{if } \texttt{ri rdf:type rdfs:Property.} \text{ is a triple} \\ v_i \in V_j, & \text{if } \texttt{ri\^{}\^{}xsd:Vj} \text{ is a litteral (data value)} \\ d_i \in D, & \text{otherwise } (\texttt{ri} \text{ is an entity URI}) \end{cases}$$

Institut für Maschinelle Sprachverarbeitung

Semantics of (Mini) RDFS

An **interpretation function** $\cdot^{\mathcal{I}}$ maps every RDFS resource and litteral to an element in the domains of RDFS model $\mathcal{I}$

More formally:

$$\texttt{ri}^{\mathcal{I}} = \begin{cases} P_i \in P, & \text{if } \texttt{ri rdf:type rdfs:Class.} \text{ is a triple} \\ R_i \in R, & \text{if } \texttt{ri rdf:type rdfs:Property.} \text{ is a triple} \\ v_i \in V_j, & \text{if } \texttt{ri\^{}\^{}xsd:Vj} \text{ is a litteral (data value)} \\ d_i \in D, & \text{otherwise (}\texttt{ri}\text{ is an entity URI)} \end{cases}$$

An RDFS structure $\mathcal{I}$ and an RDFS interpretation $\cdot^{\mathcal{I}}$ constitute a (set-theoretical) RDFS **model**

Semantics of (Mini) RDFS

Modulo $\mathcal{I}$, we can determine if an RDFS triple `S P O.` is **true** in $\mathcal{I}$, written $\mathcal{I} \models$ `S P O.`

$$\begin{aligned}
\mathcal{I} \models \texttt{ri rk rj.} &\iff (\texttt{ri}^{\mathcal{I}}, \texttt{rj}^{\mathcal{I}}) \in \texttt{rk}^{\mathcal{I}} \\
\mathcal{I} \models \texttt{ri rdf:type rj.} &\iff \texttt{ri}^{\mathcal{I}} \in \texttt{rj}^{\mathcal{I}} \\
\mathcal{I} \models \texttt{ri rdfs:subClassOf rj.} &\iff \texttt{ri}^{\mathcal{I}} \subseteq \texttt{rj}^{\mathcal{I}} \\
\mathcal{I} \models \texttt{ri rdfs:subPropertyOf rj.} &\iff \texttt{ri}^{\mathcal{I}} \subseteq \texttt{rj}^{\mathcal{I}} \\
\mathcal{I} \models \texttt{ri rdfs:domain rj.} &\iff \mathit{Dom}(\texttt{ri}^{\mathcal{I}}) \subseteq \texttt{rj}^{\mathcal{I}} \\
\mathcal{I} \models \texttt{ri rdfs:range rj.} &\iff \mathit{Ran}(\texttt{ri}^{\mathcal{I}}) \subseteq \texttt{rj}^{\mathcal{I}}
\end{aligned}$$

Institut für Maschinelle Sprachverarbeitung

Excercise: RDFS Models

1. Consider the RDFS ontology in slide 2. Define and RDFS set-theoretical model for this ontology, that is, define
   - an RDFS structure
   - an RDFS intepretation function mapping the ontology to the structure
2. How many models are possible?
3. What can we say about the information that the ontology explicitly states and what can we say about that that it doesn't state?
4. What about relational databases?

Outline

Institut für Maschinelle Sprachverarbeitung

OWA and Entailment

An RDFS ontology (set of triples) $\mathcal{O}$ is **true** in model $\mathcal{I}$, in symbols $\mathcal{I} \models \mathcal{O}$ if every such triple is true in $\mathcal{I}$

OWA and Entailment

An RDFS ontology (set of triples) $\mathcal{O}$ is **true** in model $\mathcal{I}$, in symbols $\mathcal{I} \models \mathcal{O}$ if every such triple is true in $\mathcal{I}$

- An RDFS ontology $\mathcal{O}$ describes **certain knowledge** that we believe holds in a its domain/data
- By contrast to relational databases, triples that are not in $\mathcal{O}$ are not false, but unknown $\triangleright$ **open world assumption** (OWA)

Institut für Maschinelle Sprachverarbeitung

OWA and Entailment

An RDFS ontology (set of triples) $\mathcal{O}$ is **true** in model $\mathcal{I}$, in symbols $\mathcal{I} \models \mathcal{O}$ if every such triple is true in $\mathcal{I}$

- An RDFS ontology $\mathcal{O}$ describes **certain knowledge** that we believe holds in a its domain/data
- By contrast to relational databases, triples that are not in $\mathcal{O}$ are not false, but unknown ▷ **open world assumption** (OWA)

- An RDFS ontology $\mathcal{O}$ can have **more than one** formal model $\mathcal{I}$
- However, any implicit, but certain, piece of information implied by will be true in **all** its models

Institut für Maschinelle Sprachverarbeitung

OWA and Entailment

An RDFS ontology (set of triples) $\mathcal{O}$ is **true** in model $\mathcal{I}$, in symbols $\mathcal{I} \models \mathcal{O}$ if every such triple is true in $\mathcal{I}$

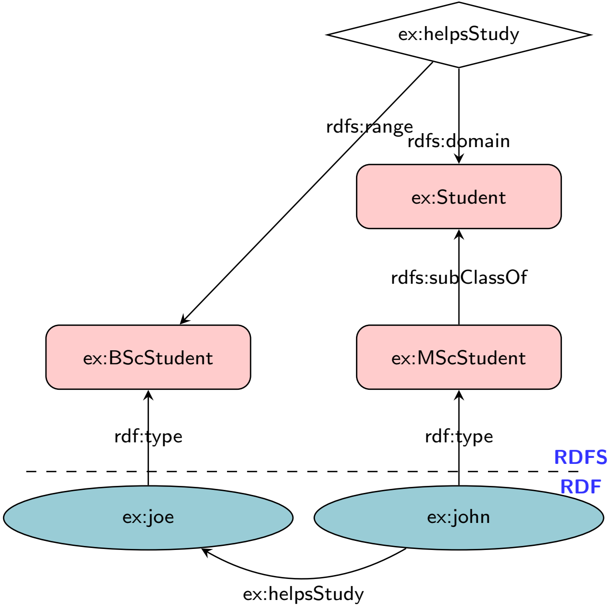
- An RDFS ontology $\mathcal{O}$ describes **certain knowledge** that we believe holds in a its domain/data
- By contrast to relational databases, triples that are not in $\mathcal{O}$ are not false, but unknown ▷ **open world assumption** (OWA)

- An RDFS ontology $\mathcal{O}$ can have **more than one** formal model $\mathcal{I}$
- However, any implicit, but certain, piece of information implied by will be true in **all** its models

An RDFS ontology $\mathcal{O}$ **entails** a triple `S P O.`, in symbols $\mathcal{O} \models$ `S P O.`, if for every formal model $\mathcal{I}$, if $\mathcal{I} \models \mathcal{O}$ then $\mathcal{I} \models$ `S P O.`

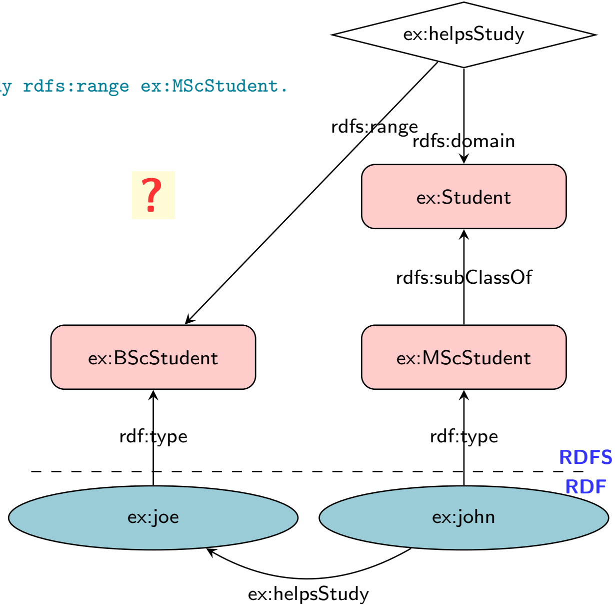
Institut für Maschinelle Sprachverarbeitung

RDFS Entailment (Domain and Range Case)

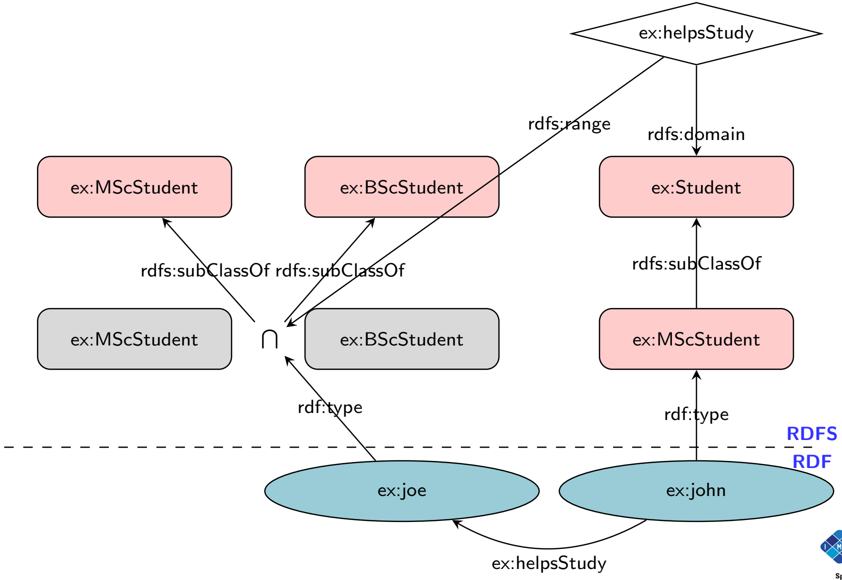


RDFS Entailment (Domain and Range Case)

`ex:helpsStudy rdfs:range ex:MScStudent.`



RDFS Entailment (Domain and Range Case)



RDFS Entailment (Domain and Range Case)

```
ex:john ex:helpsStudy ex:joe.
ex:john rdf:type ex:MScStudent.
ex:joe rdf:type ex:BScStudent.

ex:Student rdf:type rdfs:Class.
ex:BScStudent rdf:type rdfs:Class.
ex:MScStudent rdf:type rdfs:Class.
ex:MScStudent rdfs:subClassOf ex:Student.

ex:helpsStudy rdfs:type rdfs:Property
ex:helpsStudy rdfs:domain ex:Student.
ex:helpsStudy rdfs:range ex:BScStudent.
```

$\models$	$\not\models$
`ex:john rdf:type ex:Student.`	`ex:joe rdf:type ex:MScStudent.`

Institut für Maschinelle Sprachverarbeitung

RDFS Entailment (Domain and Range Case)

```
ex:john ex:helpsStudy ex:joe.
ex:john rdf:type ex:MScStudent.
ex:joe rdf:type ex:BScStudent.

ex:Student rdf:type rdfs:Class.
ex:BScStudent rdf:type rdfs:Class.
ex:MScStudent rdf:type rdfs:Class.
ex:MScStudent rdfs:subClassOf ex:Student.

ex:helpsStudy rdfs:type rdfs:Property
ex:helpsStudy rdfs:domain ex:Student.
ex:helpsStudy rdfs:range ex:BScStudent.
ex:helpsStudy rdfs:range ex:MScStudent.
```

$\models$	$\models$
`ex:john rdf:type ex:Student.`	`ex:joe rdf:type ex:MScStudent.`

Institut für Maschinelle Sprachverarbeitung

Outline

Exercise: RDFS Rules

Show that the triple

`ex:joe rdf:type ex:MScStudent.`

follows from the RDFS ontology extended with the RDFS constraint

`ex:helpsStudy rdfs:range ex:MScStudent.`

and seen in slide 11 **using only RDFS rules** (range, domain, inheritance, transitivity of subclass and subproperty, etc.)

RDFS Entailment Rules

- RDFS rules express the different entailments that hold for an RDFS ontology
- Reasoning rules can be seen as algorithm or **reasoner** that allow us to "compute" entailment (a set-theoretical relationship) ▷ **deductive calculus**
- If triple `S P O.` follows from $\mathcal{O}$ by RDFS rules we write $\mathcal{O} \vdash_{\mathrm{RDFS}}$
- If every rule corresponds to an entailment, we say that the rule is sound; and if every entailment is captured by a rule, we say that the rule is complete

▷ RDFS rules are sound and complete:

Given a RDFS triple `S P O.` and RDFS ontology $\mathcal{O}$

$$\mathcal{O} \models \texttt{S P O.} \iff \mathcal{O} \vdash_{\mathrm{RDFS}} \texttt{S P O.}$$

Institut für Maschinelle Sprachverarbeitung

Exercise: Soundness of RDFS Rules

1. Show that the RDFS inheritance rule

   (Subclass) Inheritance

   If we have statements

   ```
   A rdfs:subClassOf B .
   X rdf:type A .
   ```

   we can infer that

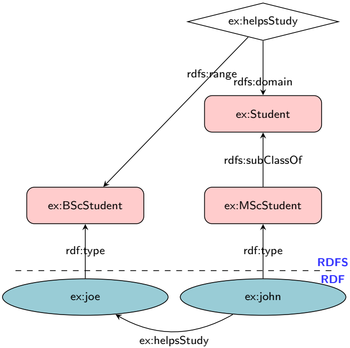
   ```
   X rdf:type B .
   ```

   is sound

2. What is the relation of RDFS reasoning rules and the graphical model ($\neq$ formal, set-theoretical model)?

Practical Tipps

- In practice, RDFS rules capture the set-theoretic relationships observable in the graphical model



- It suffices to reason over this model and on how it will change when a new RDFS constraint or factual triple is added

Institut für Maschinelle Sprachverarbeitung

Disclaimer

- This is a simplified version of RDFS semantics as described in [HKR09]

Disclaimer

- This is a simplified version of RDFS semantics as described in [HKR09]
- In order to have a sound and complete deductive calculus for RDFS, we need to look at **blank nodes**
- This course **will not cover** the rule regarding blank nodes, which relies on computing ontology homomorphisms, and reads

> **Blank Nodes**
>
> We can derive ontology $\mathcal{O}'$ from ontology $\mathcal{O}$ if there exists an homorphism h from $\mathcal{O}'$ to $\mathcal{O}$

($\approx$ we can shuffle around blank nodes provided the overall structure of models remains stable)

Institut für Maschinelle Sprachverarbeitung

Outline

Institut für Maschinelle Sprachverarbeitung

Summary

- RDFS has a model- or set-theoretic semantics
- Modulo set theoretical semantics, we can see RDFS constructs constrain the formal mapping of resources and litterals to individuals, classes, properties and data values

- RDFS ontologies and graphical models represent what is known for certain in a domain
- But the information not stated is not necessarily false (OWA)
- Graphical models describe more than one possible world or situation
  - ▷ this is captured by multiple set theoretical models

- RDFS rules capture what can be derived for certain from the ontology (or graphical model)

Outline

Institut für Maschinelle Sprachverarbeitung

Suggested Reading

- Pascal Hitzler, Markus Krötzsch, Sebastian Rudolph and York Sure. Semantic Web. Grundlagen. Springer textbook, 2008. (Chapter 7)
- Pascal Hitzler, Markus Krötzsch and Sebastian Rudolph. Foundations of Semantic Web Technologies. Chapman & Hall/CRC, 2009. (Chapter 7)

Institut für Maschinelle Sprachverarbeitung

Appendix: Completeness of RDFS Rules (Sketch) I

We want to show that $\mathcal{O} \models \texttt{S P O.} \Rightarrow \mathcal{O} \vdash_{\text{RDFS}} \texttt{S P O.}$.

To show completeness, we perform a *reduction ad absurdum*, assuming $\mathcal{O} \models \texttt{S P O.}$ but that $\mathcal{O} \nvdash_{\text{RDFS}} \texttt{S P O.}$, for each rule.

The argument is basically the same for each rule, and is *grosso modo* based on the idea of a **completion graph**. Basically, a completion graph is an ontology where all the implicit information has been made explicit. If $\mathcal{O} \nvdash_{\text{RDFS}} \texttt{S P O.}$, this means that the premises of the rule are in the completion, but not the triple $\texttt{S P O.}$.

Institut für Maschinelle Sprachverarbeitung

Appendix: Completeness of RDFS Rules (Sketch) II

Now, everything explicitly stated in an ontology and *a fortiori* its completion, is true in all its models. False in some models otherwise. If is not in the completion, then it will be false in some model, and won't be entailed by $\mathcal{O}$.

However, via model-theoretic reasoning, since the premises of the rule are in the completion and are true in all models, it follows that the same should hold for the conclusion of the rule: i.e., `S P O.` must be (i) true in all models and (ii) must be in the completion. This is a contradiction.

This implies that our assumption is false, and $\mathcal{O} \vdash_{\mathsf{RDFS}}$ `S P O.` holds.

Institut für Maschinelle Sprachverarbeitung