On Logical Foundation of the Semantic Web

An institution-based approach

Dorel Lucanu

{dlucanu}@info.uaic.ro

"Alexandru Ioan Cuza" University, Romania

Outline

- motivation
- institutions
- Semantic Web (SW) languages
  - RDF and RDF Schema
  - OWL
  - SWRL
  - SWRL FOL
- relationships between SW languages
- conclusion

Motivation

- an integrating mathematical structure for Semantic Web languages
- soundness of the reasoners for Web ontologies
- translating Web ontologies into other formalisms
- "NOTE: There is a strong correspondence between the semantics for OWL DL defined in this section (RDF-based) and the Direct Model-Theoretic Semantics defined in . . . . If, however, any conflict should ever arise between these two forms, then the Direct Model-Theoretic Semantics takes precedence." (OWL Web Ontology Language Semantics and Abstract Syntax Section 5. RDF-Compatible Model-Theoretic Semantics, http://www.w3.org/TR/owl-semantics/rdfs.html)

Institutions

- formalize the notion of "a logic"
- study the properties of a logic
  - representation
  - implementation
  - translation of logics

Institutions: Ingredients

- **signatures** Σ
  - formalize vocabularies

    Example: (Many Sorted) First Order Logic with Equality (FOLEQ)

	Booleans	Integers
sorts	`Bool`	`Int`
constants	`false true : Bool`	`0 : Int`
operations	`_and_ : Bool Bool→Bool`	`succ pred : Int→Int` `_×_ : Int Int→Int`

Institutions: Ingredients

signatures (continued)

- are organized as categories, where the **signature morphisms** formalizes the translations between vocabularies

ϕ : Booleans $\rightarrow$ Integers

Bool $\mapsto$ Int
false $\mapsto$ 0, true $\mapsto$ succ(0)
and $\mapsto$ _×_

Institutions: Ingredients

sentences

- abstracts the notion of formula
- is formalized as a functor
  sen : **Sign** $\longrightarrow$ **Set**
- sen(Σ) = the set of well-formed first-order formulas built over Σ
  $(\forall y : \texttt{Int})\ y \times 0 = 0$
  $(\forall y : \texttt{Bool})\ y\ \texttt{and}\ \texttt{false} = \texttt{false}$
- if $\phi : \Sigma \to \Sigma'$ then sen(ϕ) : sen$(\Sigma) \to$ sen(Σ')
- sen$(\phi)(y\ \texttt{and}\ \texttt{false} = \texttt{false}) = (y \times 0 = 0)$

Notation: sen$(\phi)(\mathfrak{s}) \stackrel{not}{=} \phi(\mathfrak{s})$

Institutions: Ingredients

models

- are interpretations of the syntactical constructs
- are parameterized over signatures: **Mod**(Σ) = the category of the Σ-models (interpretations of the vovabulary Σ)
- are formalized as a functor:
  Mod : **Sign**$^{\mathrm{op}} \longrightarrow$ **Cat**
- if $\phi : \Sigma \rightarrow \Sigma'$ then **Mod**(ϕ^{op}) : **Mod**$(\Sigma') \rightarrow$ **Mod**(Σ)

  $\mathbb{Z}$ is a model for `Integers`
  Mod$(\phi^{\mathrm{op}})(\mathbb{Z})$ is $\mathbb{Z}$ viewed as a model for `Booleans`

Notation: **Mod**$(\phi^{\mathrm{op}})(M) \stackrel{not}{=} M{\restriction}_{\phi}$

Institutions: Ingredients

satisfaction relation

- relates the models and the sentences: $M \models_\Sigma \mathfrak{s}$ where M is Σ-model and $\mathfrak{s}$ is a Σ-sentence
- it is the subject of the **satisfaction condition** which expresses the invariance of truth under change of notation

  $$M' \models_{\Sigma'} \phi(\mathfrak{s}) \text{ iff } M'\restriction_\phi \models_\Sigma \mathfrak{s}$$

  where $\phi : \Sigma \to \Sigma'$, M' is a Σ'-model, and $\mathfrak{s}$ is a Σ-sentence

$\mathbb{Z} \models_{\texttt{Integers}} y \times 0 = 0$ iff
$\mathbb{Z}\restriction_\phi \models_{\texttt{Booleans}} y\ \texttt{and}\ \texttt{false} = \texttt{false}$

Institutions: Specifications and Theories

- a **specification** is a pair $(\Sigma, \mathcal{S})$, where Σ is a signature and $\mathcal{S}$ is a set of sentences
- semantical consequences: $(\Sigma, \mathcal{S}) \models \mathfrak{s}$ iff $(\forall M)(M \models_\Sigma \mathcal{S} \Rightarrow M \models_\Sigma \mathfrak{s})$
- a **theory** is a specfication $(\Sigma, \mathcal{S})$ s.t. $(\forall \mathfrak{s})((\Sigma, \mathcal{S}) \models \mathfrak{s} \Rightarrow \mathfrak{s} \in \mathcal{S}$
- the inclusion **Th** $\rightarrow$ **Spec** is an equivalence of categories
- theoroidal institutions:
  - signatures are theories
  - a $(\Sigma, \mathcal{S})$-sentence is a Σ-sentence
  - $(\Sigma, \mathcal{S})$-models are Σ-models satisfying $\mathcal{S}$
  - $M \models_{(\Sigma,\mathcal{S})} \mathfrak{s}$ iff $M \models_\Sigma \mathfrak{s}$

Institutions: Properties of interest

theory colimits

- the module expressions are evaluated as colimits of theories

model amalgamation

- expresses the possibility of amalgamation of consistent models for different specification modules

liberality

- expresses the possibility of free constructions generalizing the principle of “initial semantics”

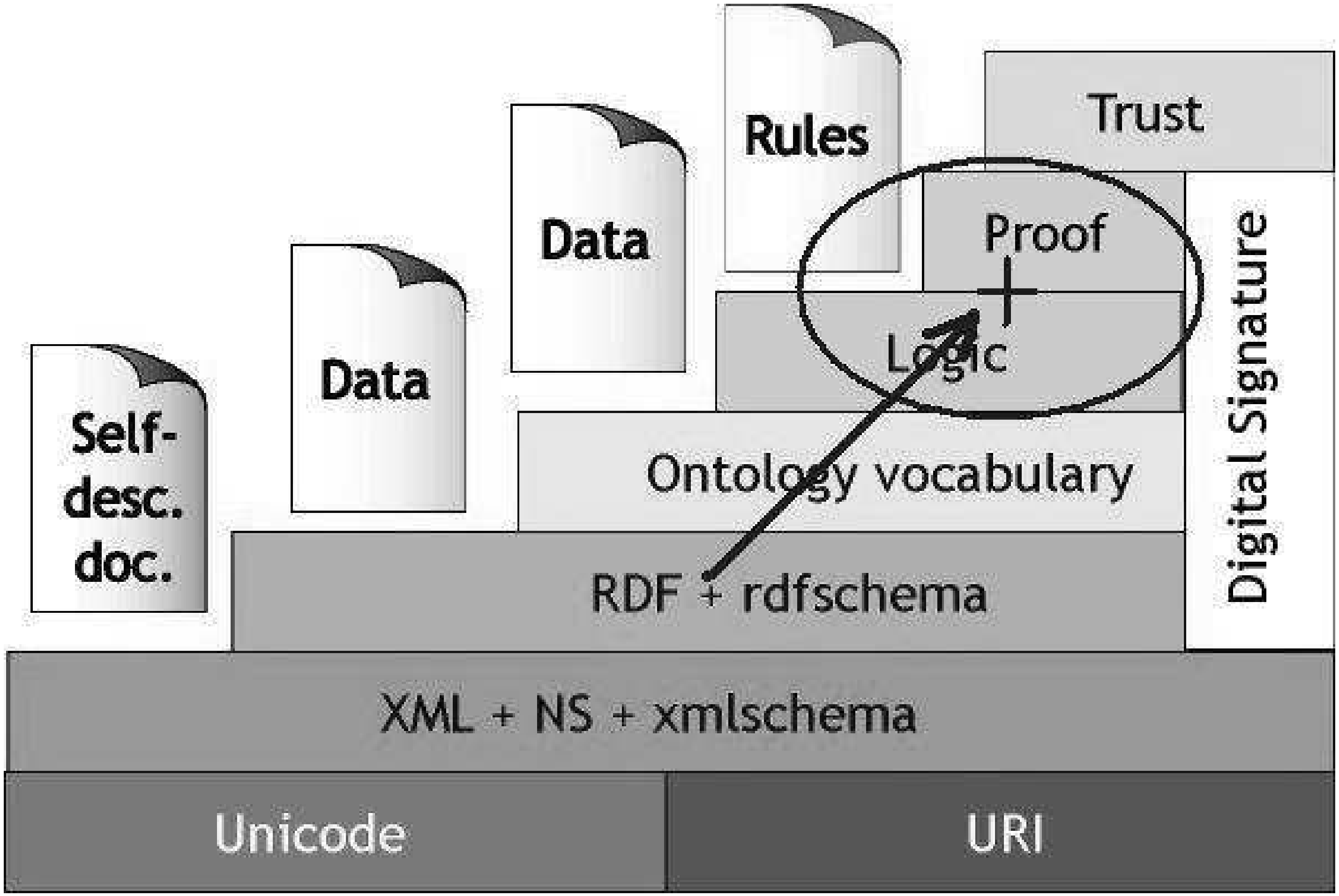
Relating Institutions

- **morphism**: capture the way in which a "richer" institution is built over a "simpler" one
- **comorphism**: capture the way in which a "simpler" institution is embedded (encoded) into a "richer" one
- both are the subject of a corresponding satisfaction condition
- there exist a variety of definitions for morphisms and variety of definitions for comorphisms in literature
- a prover from the target logic can be used to prove properties from the source logic only if certain conditions are fulfilled

Institutions: Main references

- Introducing Institutions, by J. Goguen and R. Burstall, 1984
- Institutions: Abstract model theory for specification and programming, by J. Goguen and R. Burstall, 1992
- Structuring theories on consequence, by J. Fiadeiro and A.Sernadas - 1988
- May I Borrow Your Logic?, by M. Cerioli and J. Meseguer,1993
- Moving Between Logical Systems, Andrzej Tarlecki, 1995
- Institution Morphisms, by J. Goguen and Gr. Rosu, 2002
- Grothendieck Institutions, by R. Diaconescu

Semantic Web



From Semantic Web talk by Tim Berners-Lee at XML 2000

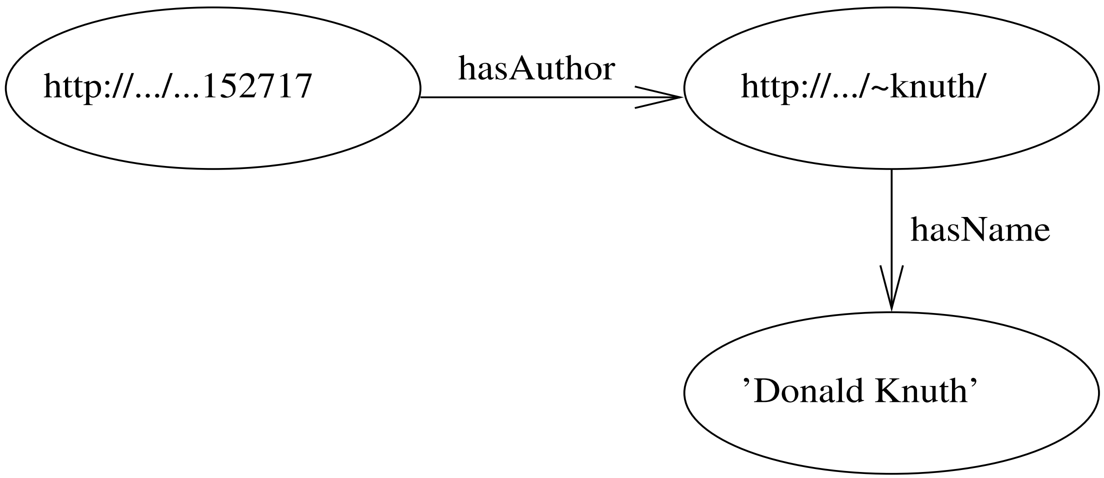
RDF

- proposed in October 1997
- in February 1999 becomes a W3C recommendation
- it is a standard for representing information in the Web
- a expression in RDF is a collection of triples, each consisting of a **subject**, a **property (predicate)**, and an **object**

RDF - example

```
<rdf:Description rdf:about=
     "http://www-cs-faculty.stanford.edu/~knuth/">
  <hasName rdf:resource="Donald Knuth" />
</rdf:Description>

<rdf:Description rdf:about=
     "http://www.amazon.com/exec/.../104-3442396-7552717">
  <hasAuthor rdf:resource=
     "http://www-cs-faculty.stanford.edu/~knuth/" />
</rdf:Description>
```



The institution $\widehat{\underline{\mathsf{RDF{+}{+}}}}$

We consider given a datatype $\mathbb{D}$

- signatures: $\Sigma = (\mathbb{RR}, \mathbb{BN})$
- sentences: $F ::= (s, p, o) \mid u \equiv v \mid F \wedge F \mid \neg F \mid (\forall y) F$
- models: $A = (Res_A, Prop_A, res_A, [\![\_]\!]_A)$, where
  Res_A a set of *resources*
  $Prop_A$ a set of *properties* (assume that $Prop_A \subseteq Res_A$)
  $res_A : \mathbb{RR} \to Res_A$
  $[\![\_]\!]_A : Prop_A \to \mathcal{P}(Res_A \times (Res_A \cup [\![\mathbb{D}]\!]))$
- satisfaction:
  $A \models (s, p, o)$ iff $res_A(p) \in Prop_A$ and
  $(res_A(s), res_A(o)) \in [\![res_A(p)]\!]_A$
  $A \models u \equiv v$ iff $res_A(u) = res_A(v)$

Interpretation of the blank nodes

$\Sigma = (\mathbb{RR}, \mathbb{BN})$

a Σ-model A and a Σ-sentence $\mathfrak{s}$

$\phi : \Sigma \to \Sigma' = (\mathbb{RR} \cup \mathbb{BN}, \emptyset)$

$A \models_\Sigma \mathfrak{s}$ iff there is a Σ'-model A' s.t.

$A'\upharpoonright_\phi = A$ and $A' \models_{\Sigma'} \mathfrak{s}$

Interpretation of the blank nodes

$\Sigma = (\mathbb{RR}, \mathbb{BN})$
a Σ-model A and a Σ-sentence $\mathfrak{s}$
$\phi : \Sigma \to \Sigma' = (\mathbb{RR} \cup \mathbb{BN}, \emptyset)$
$A \models_\Sigma \mathfrak{s}$ iff there is a Σ'-model A' s.t.
$A'\upharpoonright_\phi = A$ and $A' \models_{\Sigma'} \mathfrak{s}$

The satisfaction of the RDF graphs:

a RDF graf is a set $\mathcal{S}$ of triples

$A \models_\Sigma \mathcal{S}$ iff $A \models_\Sigma \bigwedge_{\mathfrak{s} \in \mathcal{S}} \mathfrak{s}$

Interpretation of the blank nodes

$\Sigma = (\mathbb{RR}, \mathbb{BN})$
a Σ-model A and a Σ-sentence $\mathfrak{s}$
$\phi : \Sigma \to \Sigma' = (\mathbb{RR} \cup \mathbb{BN}, \emptyset)$
$A \models_\Sigma \mathfrak{s}$ iff there is a Σ'-model A' s.t.
$A'\upharpoonright_\phi = A$ and $A' \models_{\Sigma'} \mathfrak{s}$

The satisfaction of the RDF graphs:

a RDF graf is a set $\mathcal{S}$ of triples

$A \models_\Sigma \mathcal{S}$ iff $A \models_\Sigma \bigwedge_{\mathfrak{s} \in \mathcal{S}} \mathfrak{s}$

which is not always the same with saying that A satisfies all the sentences in $\mathcal{S}$

A specification in $\widehat{\text{RDF++}}$

$$
\begin{aligned}
(\Sigma, \mathcal{S}) = \{ & \\
& (\{\mathit{bk}, \mathit{dkhp}, \mathit{hasAuthor}, \mathit{hasName}\}, \emptyset), \\
& (\mathit{bk}, \mathit{hasAuthor}, \mathit{dkhp}), \\
& (\mathit{dkhp}, \mathit{hasName}, \texttt{"Donald Knuth"}) \\
\} &
\end{aligned}
$$

$\widehat{\underline{\text{RDF++}}}$: properties

Sign$_{\widehat{\underline{\text{RDF++}}}}$ has colimits

- e.g., merge:

$$\begin{array}{ccc}(\mathbb{RR}_1 \cap \mathbb{RR}_2, \emptyset) & \longrightarrow & (\mathbb{RR}_1, \mathbb{BN}_1) \\ \downarrow & & \downarrow \\ (\mathbb{RR}_2, \mathbb{BN}_2) & \longrightarrow & (\mathbb{RR}_1 \cup \mathbb{RR}_2, \mathbb{BN}_1 \coprod \mathbb{BN}_2)\end{array}$$

$\widehat{\underline{\text{RDF++}}}$ is liberal (free constr. is a generalized Herbrand constr.)

$\widehat{\underline{\text{RDF++}}}$ has amalgamation property (**Mod**$(\widehat{\underline{\text{RDF++}}})$ preserves finite limits)

The specification RDFV(RDF Vocabulary)

$$\mathbb{RR}(\mathsf{RDF}) = \{\mathtt{rdf}:\mathtt{type}, \mathtt{rdf}:\mathtt{Property}, \mathtt{rdf}:\mathtt{list}, \mathtt{rdf}:\mathtt{nil}, \ldots\}$$

$$\mathbb{BN}(\mathsf{RDF}) = \emptyset$$

$$\mathcal{S}(\mathsf{RDF}) = \{$$
$$(\mathtt{rdf}:\mathtt{type}, \mathtt{rdf}:\mathtt{type}, \mathtt{rdf}:\mathtt{Property}),$$
$$(\mathtt{rdf}:\mathtt{nil}, \mathtt{rdf}:\mathtt{type}, \mathtt{rdf}:\mathtt{List}),$$
$$(\forall\, s, p, o)(s, p, o) \rightarrow (p, \mathtt{rdf}:\mathtt{type}, \mathtt{rdf}:\mathtt{Property}),$$
$$\ldots$$
$$\}$$

RDF Schema

- proposed in March 1999
- is a standard which describes how to use RDF to describe RDF vocabularies
- it is claimed that it is a semantical extension of RDF
- introduces the basic primitives for ontology modeling:
  - classes, subclasses
  - subproperties
  - domain, range
  - ...

RDF Schema: Example

```
<rdfs:Class rdf:about="Book" />

<rdfs:Class rdf:about="Person" />

<rdfs:Class rdf:about="Author">
  <rdfs:subClassOf rdf:resource="#Person" />
</rdfs:Class>

<rdf:Property rdf:about="hasAuthor">
  <rdfs:domain rdf:resource="Book" />
  <rdfs:range rdf:resource="Author" />
</rdf:Property>
```

RDFS **is a theory!**

$$\mathbb{RR}(\mathsf{RDFS}) = \mathbb{RR}(\mathsf{RDF}) \cup \{\mathtt{rdfs:Class}, \mathtt{rdfs:subClassOf}, \mathtt{rdfs:subPropertyOf}, \mathtt{rdfs:domain}, \ldots\}$$

$$\mathbb{BN}(\mathsf{RDFS}) = \text{those used in sentences}$$

$$\mathcal{S}(\mathsf{RDFS}) = \mathcal{S}(\mathsf{RDF}) \cup$$

$$\{$$

$$(\mathtt{rdf:type}, \mathtt{rdfs:domain}, \mathtt{rdfs:Resource}),$$

$$(\mathtt{rdfs:domain}, \mathtt{rdfs:domain}, \mathtt{rdf:Property}),$$

$$(\forall\, u, v, x, y)(x, \mathtt{rdf:domain}, y) \wedge (u, x, v) \rightarrow (u, \mathtt{rdf:type}, \mathtt{y})$$

$$(\forall\, x, y)(x, \mathtt{rdfs:subClassOf}, y) \rightarrow (x, \mathtt{rdf:type}, \mathtt{rdfs:Class}),$$

$$(\forall\, x, y)(x, \mathtt{rdfs:subClassOf}, y) \rightarrow (y, \mathtt{rdf:type}, \mathtt{rdfs:Class}),$$

$$(\forall\, u, x, y)(x, \mathtt{rdfs:subClassOf}, y) \wedge (u, \mathtt{rdf:type}, \mathtt{x}) \rightarrow (u, \mathtt{rdf:type}, y),$$

$$\ldots$$

$$\}$$

The institution $\widehat{\underline{\mathsf{RDFS}}}$

- signatures: theory morphisms $\mathsf{RDFS} \to (\Sigma, \mathcal{S})$
  (**Sign**$(\widehat{\underline{\mathsf{RDFS}}})$ is a comma category)
  $\mathsf{RDFS} \to (\Sigma, \mathcal{S})$ how use RDF to describe RDF vocabularies
- sentences: Σ-sentences
- models: $(\Sigma, \mathcal{S})$-models
  Mod$(\Sigma, \mathcal{S}) \to$ **Mod**$(\mathsf{RDFS}) \to$ **Mod**RDF semantical extension
- satisfaction: $A \models_{\mathsf{RDFS}\to(\Sigma,\mathcal{S})} \mathfrak{s}$ iff $A \models_{\Sigma} \mathfrak{s}$
- semantics of a class:
  $[\![C]\!]_A = \{x \mid A \models (x, \mathtt{rdf:type}, C)\}$

There is a simple theoroidal comorphism from $\widehat{\underline{\mathsf{RDFS}}}$ to $\widehat{\underline{\mathsf{RDF{+}{+}}}}$.

OWL

- proposed in March 2002
- a language used to describe Web ontologies
- has three levels: OWL LITE, OWL DL, OWL Full
- includes RDF Schema
- new items:
  - makes distinction between individual-valued properties and data-valued properties
  - cardinality restrictions
  - operations with classes
  - restrictions on properties
  - ontology imports
  - ...

OWL: Example

- each book has at least one author

```
<owl:Class rdf:ID="Author">
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource=
                      "#hasAuthor" />
      <owl:minCardinality rdf:datatype=
           "#&xsd;nonNegativeInteger">1
      </owl:minCardinality>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>
```

OWL is also a theory!

$$\mathbb{RR}(\text{OWL}) = \mathbb{RR}(\text{RDFS}) \cup \{\texttt{owl:Thing}, \texttt{owl:Class}, \texttt{owl:subClassOf}, \\ \texttt{owl:ObjectPropertyO}, \texttt{owl:DatatypeProperty}, \ldots\}$$

$\mathbb{BN}(\text{OWL}) =$ those used in sentences

$$\mathcal{S}(\text{OWL}) = \mathcal{S}(\text{RDFS}) \cup$$

$$\{$$

$$(\texttt{owl:Nothing}, \texttt{rdf:type}, \texttt{owl:Class}),$$

$$(\texttt{owl:Thing}, \texttt{rdf:type}, \texttt{owl:Class}),$$

$$\neg(\exists\, x)(x, \texttt{rdf:type}, \texttt{owl:Nothing}),$$

$$(\forall\, x, C)(x, \texttt{rdf:type}, C) \wedge (C, \texttt{rdf:type}, \texttt{owl:Class}) \rightarrow \\ (x, \texttt{rdf:type}, \texttt{owl:Thing}), \quad \ldots$$

$$\}$$

There is a forgetful morphism from $\widehat{\underline{\text{OWL}}}$ to $\widehat{\underline{\text{RDFS}}}$.
There is a simple theoroidal comorphism from $\widehat{\underline{\text{OWL}}}$ to $\widehat{\underline{\text{RDF++}}}$.

OWL DL hides some vocabulary items

$\Sigma(\mathsf{OWLDL}) = \textbf{hide}$

```
rdf:type, rdf:  Property,...,
owl:  TransitiveProperty,...
```

in

$\Sigma(\mathsf{OWL})$

... and adds some new constraints:

$$\mathsf{DLCONSTR} = \{$$

$$(\forall\, x, C)(x, \mathtt{rdf:type}, \mathtt{owl:Thing}) \wedge (C, \mathtt{rdf:type}, \mathtt{owl:Class}) \rightarrow \neg(x \equiv C),$$

$$\ldots$$

$$\}$$

$\phi : \Sigma(\mathsf{OWLDL}) \hookrightarrow \Sigma(\mathsf{OWL})$

$\textbf{Mod}(\mathsf{OWLDL}) = \{A\restriction_\phi \mid A \in \textbf{Mod}(\mathsf{OWL}) \wedge A \models \mathsf{DLCONSTR}\}$

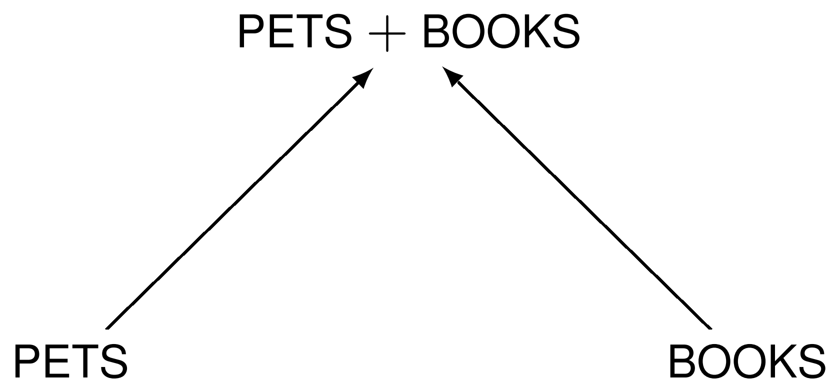
The institution $\underline{\widehat{\mathsf{OWL}}}$

- signatures: theory morphisms OWL $\rightarrow (\Sigma, \mathcal{S})$ (**Sign**$(\underline{\widehat{\mathsf{OWL}}})$ is a comma category)
- sentences: Σ-sentences
- models: $(\Sigma, \mathcal{S})$-models
- satisfaction: $A \models_{(\Sigma,\mathcal{S})} \mathfrak{s}$ iff $A \models_{\Sigma} \mathfrak{s}$

There is a forgetful morphism from $\underline{\widehat{\mathsf{OWL}}}$ to $\underline{\widehat{\mathsf{RDFS}}}$.

There is a simple theoroidal comorphism from $\underline{\widehat{\mathsf{OWL}}}$ to $\underline{\widehat{\mathsf{RDF{+}{+}}}}$.

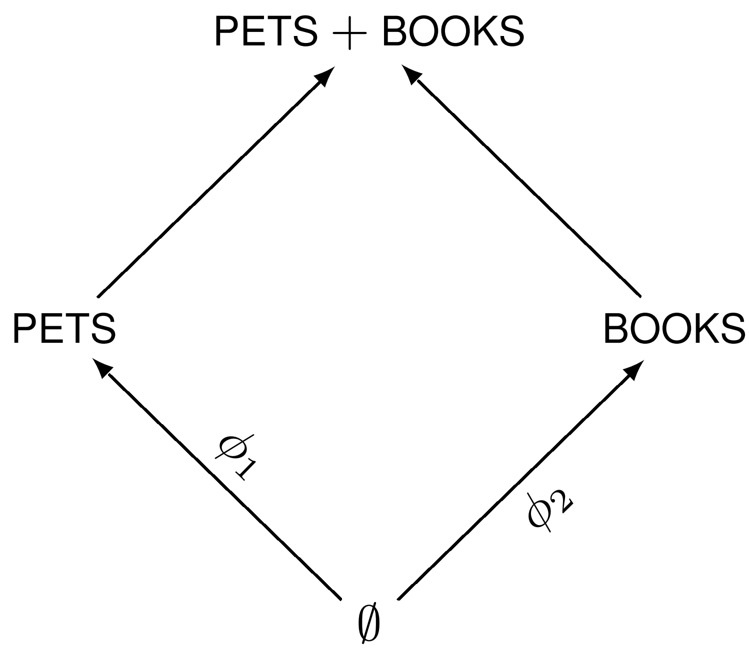
OWL: problems with amalgamation



- can we amalgamate a PETS-model A_1 and BOOKS-model A_2 in a PETS+BOOKS model?
- NO if $[\![\texttt{owl}:\texttt{Thing}]\!]_{A_1} \neq [\![\texttt{owl}:\texttt{Thing}]\!]_{A_2}$

OWL: problems with amalgamation

- solution: transform such a diagram into a pushout



- we have to consider a $\emptyset$-model A_0
- A_1 and A_2 are consistent iff $A_1\lceil_{\phi_1} = A = A_2\lceil_{\phi_2}$

SWRL

- proposed in November 2004
- extends OWL with Horn rules
- example: citation implies not self-citation

```
<ruleml:imp>
  <ruleml:_body>
    <swrlx:individualPropertyAtom  swrlx:property="writtenBy">
      <ruleml:var>x1</ruleml:var>
      <ruleml:var>x2</ruleml:var>
    </swrlx:individualPropertyAtom>
    ...
  </ruleml:_body>
  <ruleml:_head>
    ...
  </ruleml:_head>
</ruleml:imp>
```

The institution $\widehat{\underline{\mathsf{SWRL}}}$

- signatures: OWL signatures
- sentences:

$$\mathsf{writtenBy}(x_1, x_2) \wedge \mathsf{citedBy}(x_1, x_3) \rightarrow x_2 \neq x_3.$$

- models: OWL models
- satisfaction: as in OWL and HornLog

There is a forgetful morphism from $\widehat{\underline{\mathsf{SWRL}}}$ to $\widehat{\underline{\mathsf{OWL}}}$.

There is a simple theoroidal comorphism from $\widehat{\underline{\mathsf{SWRL}}}$ to $\widehat{\underline{\mathsf{RDF++}}}$.

SWRL FOL

- proposed in November 2004
- extends OWL with first-order formulas
- example: any cited author has written a paper which is cited by someone else

```
<Assertion owlx:name="Example">
  <Forall>
    <ruleml:var>x1</ruleml:var>
    <Implies>
      <swrlx:classAtom  owlx:name="CitedAuthor">
        <owlx:Class owlx:name="CitedAuthor" />
        <ruleml:var>x1</ruleml:var>
      </swrlx:classAtom>
      <Exists>
        ...
      </Exists>
    </Implies>
  </Forall>
</Assertion
```

The institution $\widehat{\underline{\mathsf{SWRLFOL}}}$

- signatures: OWL signatures
- sentences:

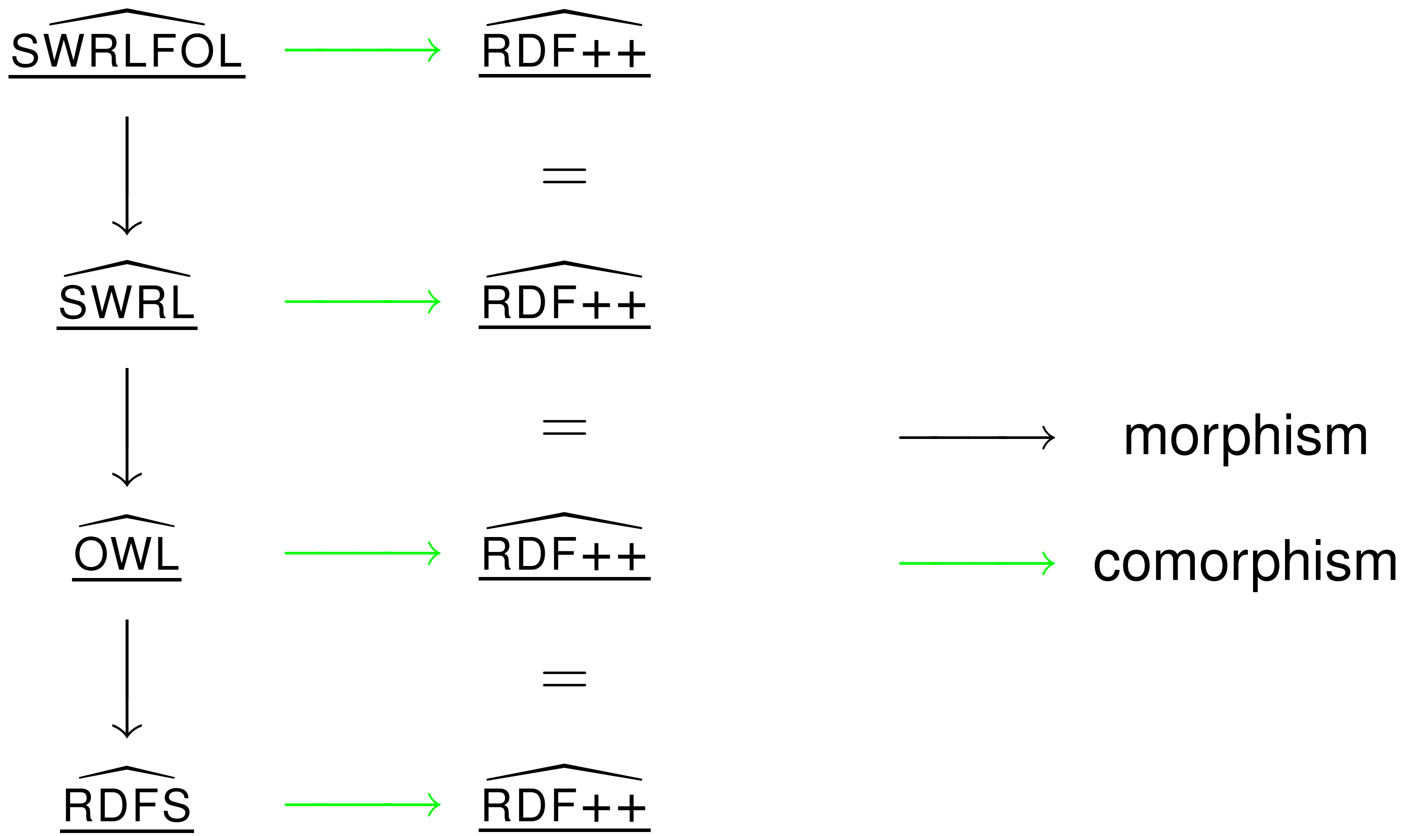
$$(\forall x_1)\mathsf{CitedAuthor}(x_1) \rightarrow (\exists x_2, x_3)\mathsf{writtenBy}(x_2, x_1) \wedge \mathsf{citedBy}(x_2, x_3)$$

- models: OWL models
- satisfaction: as in OWL and FOL

There is a forgetful morphism from $\widehat{\underline{\mathsf{SWRLFOL}}}$ to $\widehat{\underline{\mathsf{SWRL}}}$.

There is a simple theoroidal comorphism from $\widehat{\underline{\mathsf{SWRLFOL}}}$ to $\widehat{\underline{\mathsf{RDF{+}{+}}}}$.

Relationships between SW logics



Conclusion

- contributions
  - the institution $\widehat{\underline{\mathsf{RDF++}}}$
  - RDFS, OWL, SWRL, and SWRLFOL are in fact theories in $\widehat{\underline{\mathsf{RDF++}}}$
  - the institutions $\widehat{\underline{\mathsf{RDFS}}}$, $\widehat{\underline{\mathsf{OWL}}}$, $\widehat{\underline{\mathsf{SWRL}}}$, and $\widehat{\underline{\mathsf{SWRLFOL}}}$ defined as particular theoroidal institutions
  - the relationships between these institutions
- advantages:
  - a rigurous and systematic approach of the logics underlying SW languages
  - an important step towards structuring and re-using ontology parts
  - a solid framework for relating SW languages with other formalisms and for proving the soundness of the reasoners

Questions?

$\mathcal{T}$hank you!