OWL and Rules

Pascal Hitzler
Kno.e.sis Center
Wright State University, Dayton, OH
http://www.knoesis.org/pascal/

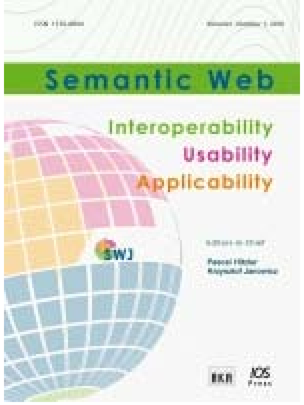
Sponsors

Most of the new work sponsored by NSF grant
III: Small: TROn – Tractable Reasoning with Ontologies

Tutorial sponsored by IOS Press

Bookseries
Studies on the Semantic Web
http://www.semantic-web-studies.net/

Semantic Web journal
http://www.semantic-web-journal.net/



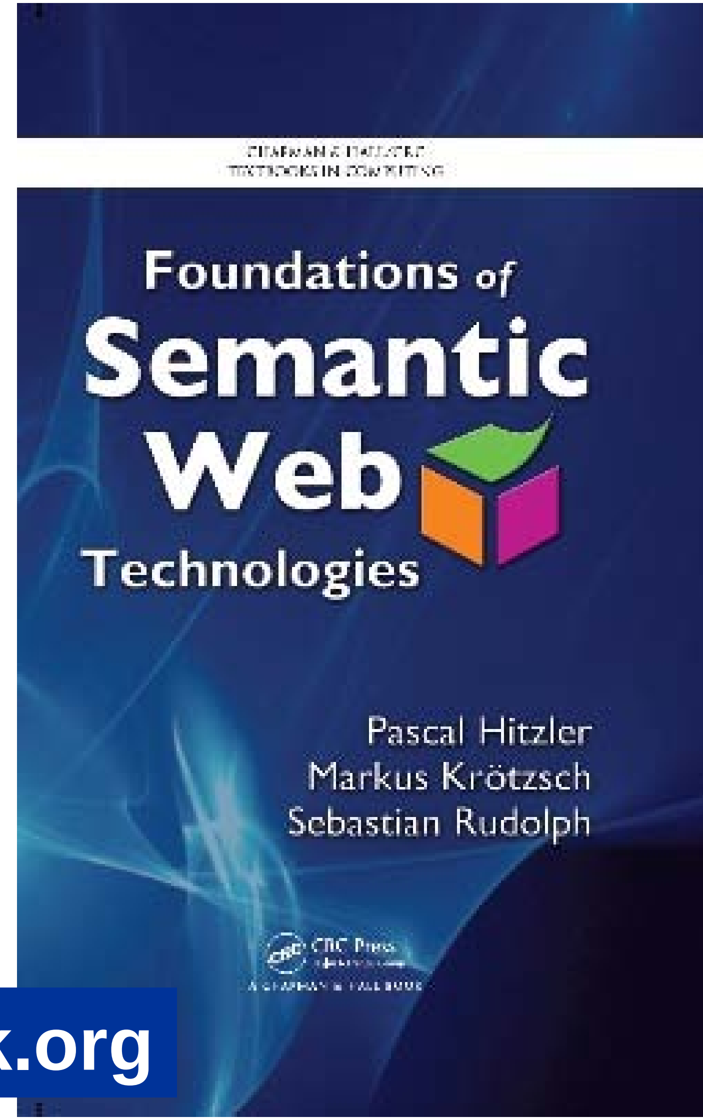
Textbook

Pascal Hitzler, Markus Krötzsch, Sebastian Rudolph

Foundations of Semantic Web Technologies

Chapman & Hall/CRC, 2010

Choice Magazine Outstanding Academic Title 2010 (one out of seven in Information & Computer Science)



http://www.semantic-web-book.org

Textbook (Chinese translation)

Pascal Hitzler, Markus Krötzsch, Sebastian Rudolph

语义Web技术基础

Tsinghua University Press (清华大学出版社），2011, to appear

Translators:

Yong Yu, Haofeng Wang, Guilin Qi (俞勇，王昊奋，漆桂林)

http://www.semantic-web-book.org

The Kno.e.sis Center

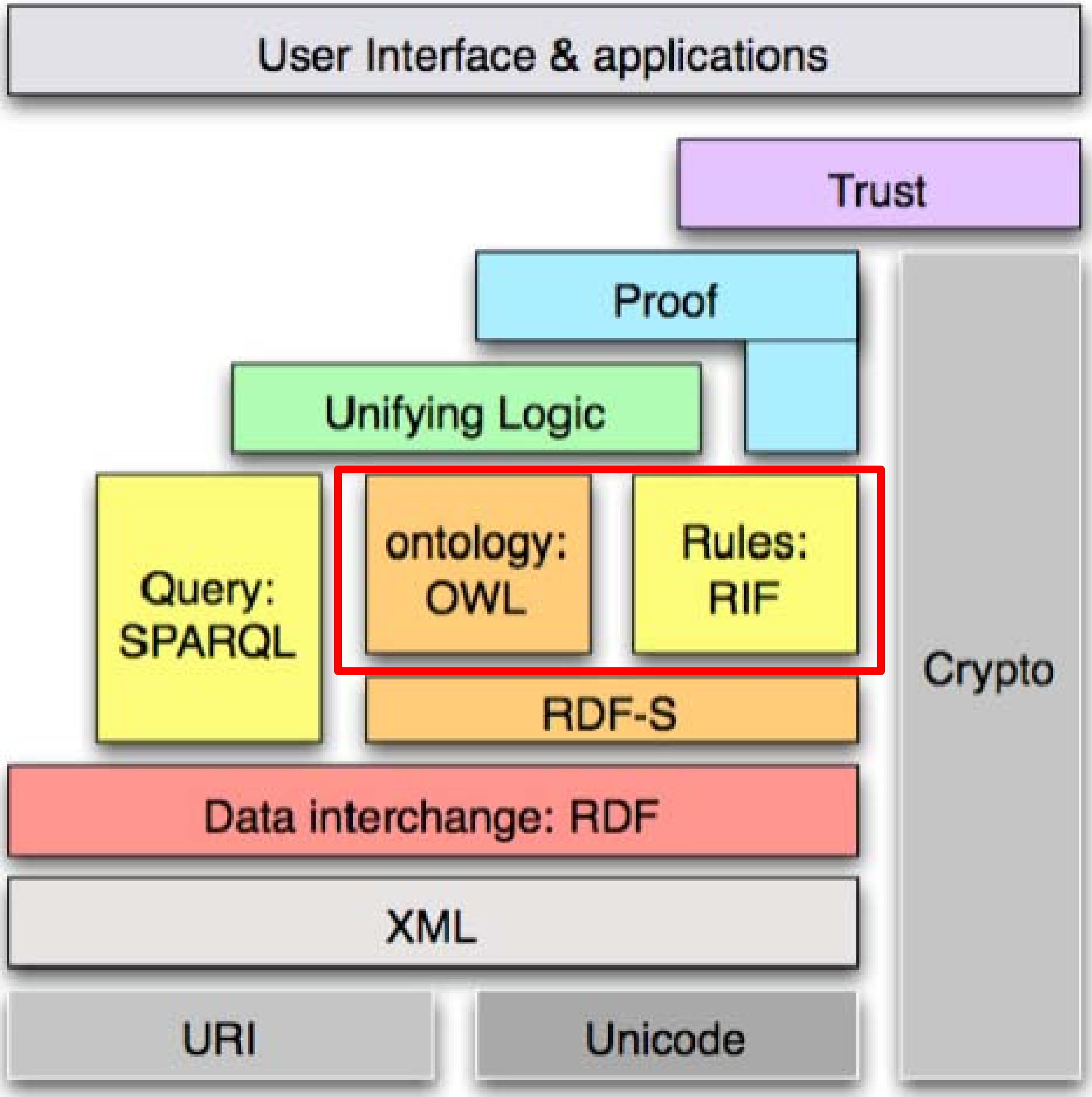
- **Ohio Center of Excellence in Knowledge-enabled Computing**
  Director: Amit Sheth

- **15 faculty across 4 Colleges**
  9 from Computer Science with ca. 50 PhD students

- **Knowledge-enabled Computing Lab (since January 2010)**
  Director: Pascal Hitzler
  Currently 10 people

- **http://www.knoesis.org/**

OWL and Rules: Two paradigms?



Contents

1. Reasoning Needs
2. Rules expressible in OWL
3. Extending OWL with Rules: Nominal Schemas
4. Conclusions

Reasoning Needs

Inspired by presentation by Evan Sandhaus, ISWC2010

x	**newsFrom**	**rome .**
rome	**locatedIn**	**italy .**

we want to conclude:

x	**newsFrom**	**italy .**

Take your news database.

Take location info from somewhere on linked data.

Materialize the new newsFrom triples.

Reasoning Needs

x	newsFrom	rome .	**newsFrom(x,y)**
rome	locatedIn	italy .	**locatedIn(y,z)**

we want to conclude:

x	newsFrom	italy .	**newsFrom(x,z)**

newsFrom(x,y) $\wedge$ locatedIn(y,z) $\rightarrow$ newsFrom(x,z)

newsFrom o locatedIn $\sqsubseteq$ newsFrom

using owl:propertyChainAxiom

Reasoning Needs

e.g. knowledge base of authors and papers

<paper>	hasAuthor	<author> .

insufficient because author order is missing

use of RDF-lists not satisfactory due to lack of formal semantics.

better:

<paper>	hasAuthorNumbered	_:x .
_:x	authorNumber	n^^xsd:positiveInteger ;
	authorName	<author> .

$\text{hasAuthorNumbered}(x,y) \wedge \text{authorName}(y,z) \rightarrow \text{hasAuthor}(x,z)$

Reading Needs

<paper> hasAuthorNumbered _:x .
_:x authorNumber n^^xsd:positiveInteger ;
authorName <author> .

$hasAuthorNumbered(x,y) \wedge authorName(y,z) \rightarrow hasAuthor(x,z)$

in OWL:

$$\mathit{Paper} \sqsubseteq \exists \mathit{hasAuthorNumbered}.\mathit{NumberedAuthor}$$

$$\mathit{NumberedAuthor} \sqsubseteq \exists \mathit{authorNumber}.\langle \mathit{xsd{:}positiveInteger} \rangle \sqcap \exists \mathit{authorName}.\top$$

$$\mathit{hasAuthorNumbered} \circ \mathit{authorName} \sqsubseteq \mathit{hasAuthor}$$

these are not rules!

Reasoning Needs

$\text{Paper} \sqsubseteq \exists \text{hasAuthorNumbered.NumberedAuthor}$

$\text{NumberedAuthor} \sqsubseteq$
$\exists \text{authorNumber.}\langle\text{xsd:positiveInteger}\rangle \sqcap \exists \text{authorName.}\top$

$\text{hasAuthorNumbered} \circ \text{authorName} \sqsubseteq \text{hasAuthor}$

$\text{Paper}(x) \wedge \text{hasAuthorNumbered}(x,y) \wedge \text{authorNumber}(y,1) \wedge$
$\text{authorName}(y,z) \rightarrow \text{hasFirstAuthor}(x,z)$

in OWL:

$\text{Paper} \equiv \exists \text{paper.Self}$

$\exists \text{authorNumber.}\{1\} \equiv \exists \text{authorNumberOne.Self}$

$\text{paper} \circ \text{hasAuthorNumbered} \circ \text{authorNumberOne} \circ \text{authorName}$
$\sqsubseteq \text{hasFirstAuthor}$

Reasoning as first-class citizen

Why would we want to have knowledge/rules such as

newsFrom(x,y) $\wedge$ locatedIn(y,z) $\rightarrow$ newsFrom(x,z)

if we can also just do this with some software code?

- **It declaratively describes what you do.**
- **It separates knowledge (as knowledge base) from programming.**
- **It makes knowledge shareable.**
- **It makes knowledge easier to maintain.**

Contents

SROIQ(D) constructors – overview

- ABox assignments of individuals to classes or properties
- ALC: $\sqsubseteq$, $\equiv$ for classes
  $\sqcap$, $\sqcup$, $\neg$, $\exists$, $\forall$
  $\top$, $\bot$
- SR: + property chains, property characteristics, property hierarchies $\sqsubseteq$
- SRO: + nominals {o}
- SROI: + inverse properties
- SROIQ: + qualified cardinality constraints
- SROIQ(D): + datatypes (including facets)

- + top and bottom roles (for objects and datatypes)
- + disjoint properties
- + Self
- + Keys (not in SROIQ(D), but in OWL)

Rules in OWL

Which rules can be encoded in OWL?

$$A \sqsubseteq B \text{ becomes } A(x) \to B(x)$$

$$R \sqsubseteq S \text{ becomes } R(x,y) \to S(x,y)$$

$$A \sqcap \exists R.\exists S.B \sqsubseteq C \text{ becomes } A(x) \land R(x,y) \land S(y,z) \land B(z) \to C(x)$$

$$A \sqsubseteq \forall R.B \text{ becomes } A(x) \land R(x,y) \to B(y)$$

Rules in OWL

Which rules can be encoded in OWL?

$$A \sqsubseteq \neg B \sqcup C \text{ becomes } A(x) \land B(x) \to C(x)$$

$$\top \sqsubseteq \leq 1 R.\top \text{ becomes } R(x, y) \land R(x, z) \to y = z$$

$$A \sqcap \exists R.\{b\} \sqsubseteq C \text{ becomes } A(x) \land R(x, b) \to C(x)$$

Rules in OWL

Which rules can be encoded in OWL?

$$\{a\} \equiv \{b\} \text{ becomes } \rightarrow a = b.$$

$$A \sqcap B \sqsubseteq \bot \text{ becomes } A(x) \land B(x) \rightarrow f.$$

$$A \sqsubseteq B \land C \text{ becomes } A(x) \rightarrow B(x) \text{ and } A(x) \rightarrow C(x)$$
$$A \sqcup B \rightarrow C \text{ becomes } A(x) \rightarrow C(x) \text{ and } B(x) \rightarrow C(x)$$

Rules in OWL

A DL axiom α can be translated into rules if, after translating α into a first-order predicate logic expression α', and after normalizing this expression into a set of clauses M, each formula in M is a Horn clause (i.e., a rule).

Issue: How complicated a translation is allowed?

Naïve translation: DLP
plus some more (since OWL 2 extends OWL 1)

e.g.,

$$R \circ S \sqsubseteq T \text{ becomes } R(x,y) \wedge S(y,z) \rightarrow T(x,z)$$

This essentially results in OWL 2 RL.

Rolification

$$\text{Elephant}(x) \land \text{Mouse}(y) \rightarrow \text{biggerThan}(x, y)$$

- **Rolification of a concept A:** $\mathbf{A \equiv \exists R_A.Self}$

$$\text{Elephant} \equiv \exists R_{\text{Elephant}}.\text{Self}$$

$$\text{Mouse} \equiv \exists R_{\text{Mouse}}.\text{Self}$$

$$R_{\text{Elephant}} \circ U \circ R_{\text{Mouse}} \sqsubseteq \text{biggerThan}$$

Rolification

$$A(x) \wedge R(x,y) \rightarrow S(x,y) \text{ becomes } R_A \circ R \sqsubseteq S$$
$$A(y) \wedge R(x,y) \rightarrow S(x,y) \text{ becomes } R \circ R_A \sqsubseteq S$$
$$A(x) \wedge B(y) \wedge R(x,y) \rightarrow S(x,y) \text{ becomes } R_A \circ R \circ R_B \sqsubseteq S$$

$$\mathrm{Woman}(x) \wedge \mathrm{marriedTo}(x,y) \wedge \mathrm{Man}(y) \rightarrow \mathrm{hasHusband}(x,y)$$

$$R_{\mathrm{Woman}} \circ \mathrm{marriedTo} \circ R_{\mathrm{Man}} \sqsubseteq \mathrm{hasHusband}$$

careful – regularity of RBox needs to be retained:

$$\mathrm{hasHusband} \sqsubseteq \mathrm{marriedTo}$$

Rolification

$$\mathrm{worksAt}(x,y) \wedge \mathrm{University}(y) \wedge \mathrm{supervises}(x,z) \wedge \mathrm{PhDStudent}(z) \rightarrow \mathrm{professorOf}(x,z)$$

$$R_{\exists\mathrm{worksAt}.\mathrm{University}} \circ \mathrm{supervises} \circ R_{\mathrm{PhDStudent}} \sqsubseteq \mathrm{professorOf}.$$

Rules in OWL 2

- $\text{Man}(x) \wedge \text{hasBrother}(x,y) \wedge \text{hasChild}(y,z) \rightarrow \text{Uncle}(x)$
  - $\text{Man} \sqcap \exists\text{hasBrother}.\exists\text{hasChild}.\top \sqsubseteq \text{Uncle}$

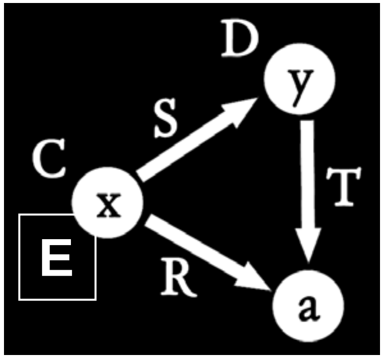
- $\text{NutAllergic}(x) \wedge \text{NutProduct}(y) \rightarrow \text{dislikes}(x,y)$
  - $\text{NutAllergic} \equiv \exists\text{nutAllergic}.\text{Self}$
    $\text{NutProduct} \equiv \exists\text{nutProduct}.\text{Self}$
    $\text{nutAllergic} \circ U \circ \text{nutProduct} \sqsubseteq \text{dislikes}$

- $\text{dislikes}(x,z) \wedge \text{Dish}(y) \wedge \text{contains}(y,z) \rightarrow \text{dislikes}(x,y)$
  - $\text{Dish} \equiv \exists\text{dish}.\text{Self}$
    $\text{dislikes} \circ \text{contains}^{-} \circ \text{dish} \sqsubseteq \text{dislikes}$

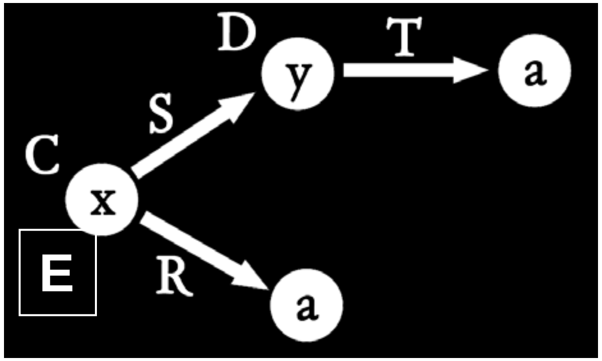
So how can we pinpoint this?

- **Tree-shaped bodies**
- **First argument of the conclusion is the root**

- $C(x) \wedge R(x,a) \wedge S(x,y) \wedge D(y) \wedge T(y,a) \rightarrow E(x)$
  - $C \sqcap \exists R.\{a\} \sqcap \exists S.(D \sqcap \exists T.\{a\}) \sqsubseteq E$



duplicating nominals is ok

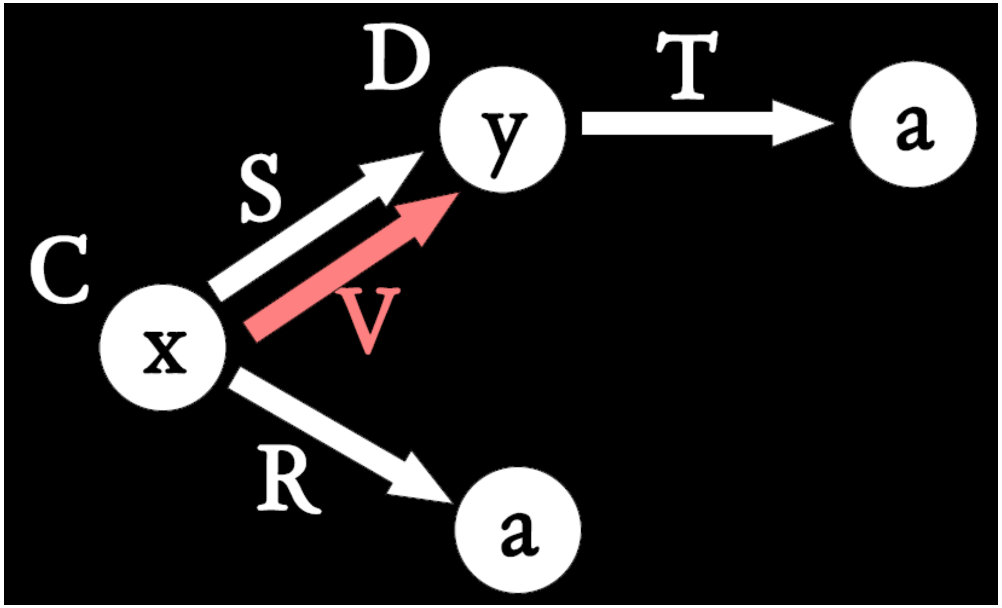


So how can we pinpoint this?

- Tree-shaped bodies
- First argument of the conclusion is the root

- $C(x) \wedge R(x,a) \wedge S(x,y) \wedge D(y) \wedge T(y,a) \rightarrow V(x,y)$

$C \sqcap \exists R.\{a\} \sqsubseteq \exists R1.\mathrm{Self}$
$D \sqcap \exists T.\{a\} \sqsubseteq \exists R2.\mathrm{Self}$
$R1 \circ S \circ R2 \sqsubseteq V$



Rule bodies as graphs

$$C(x) \wedge R(x,a) \wedge S(x,y) \wedge D(y) \wedge T(y,a) \rightarrow P(x,y)$$

$$a_1 \longleftarrow x \longrightarrow y \longrightarrow a_2$$

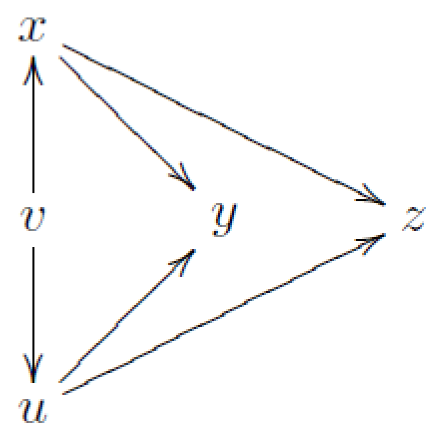
C ⊓ ∃R.{a} ⊑ ∃R1.Self

D⊓ ∃T.{a}) ⊑ ∃R2.Self

R1 ∘ S ∘ R2 ⊑ P

Rule bodies as graphs

$$\text{hasReviewAssignment}(v,x) \wedge \text{hasAuthor}(x,y) \wedge \text{atVenue}(x,z)$$
$$\wedge\ \text{hasSubmittedPaper}(v,u) \wedge \text{hasAuthor}(u,y) \wedge \text{atVenue}(u,z)$$
$$\rightarrow \text{hasConflictingAssignedPaper}(v,x)$$



with y,z constants:

$$R_{\exists \text{hasSubmittedPaper}.(\exists \text{hasAuthor}.\{y\} \sqcap \exists \text{atVenue}.\{z\})} \circ \text{hasReviewAssignment}$$
$$\circ\ R_{\exists \text{hasAuthor}.\{y\} \sqcap \exists \text{atVenue}.\{z\}}$$
$$\sqsubseteq \text{hasConflictingAssignedPaper}$$

Formally

Given a rule with body B, we construct a directed graph as follows:

1. **Rename individuals (i.e., constants) such that each individual occurs only once – a body such as R(a,x) ∧ S(x,a) becomes R(a1,x) ∧ S(x,a2). Denote the resulting new body by B'.**
2. **The vertices of the graph are then the variables and individuals occurring in B', and there is a directed edge between t and u if and only if there is an atom R(t,u) in B'.**

$$C(x) \wedge R(x,a) \wedge S(x,y) \wedge D(y) \wedge T(y,a) \rightarrow P(x,y)$$

$$a_1 \longleftarrow x \longrightarrow y \longrightarrow a_2$$

Formally

Definition 1. *We call a rule with head H* tree-shaped *(respectively,* acyclic*), if the following conditions hold.*

- *Each of the maximally connected components of the corresponding graph is in fact a tree (respectively, an acyclic graph)—or in other words, if it is a* forest, *i.e., a set of trees (respectively, a set of acyclic graphs).*
- *If H consists of an atom $A(t)$ or $R(t,u)$, then t is a root in the tree (respectively, in the acyclic graph).*

$R(x,z) \wedge S(y,z) \rightarrow T(x,y)$ is acyclic but not tree-shaped

Theorem 1. *The following hold.*

- *Every tree-shaped rule can be expressed in $\mathcal{SROEL}$.*
- *Every acyclic rule can be expressed in $\mathcal{SROIEL}$.*

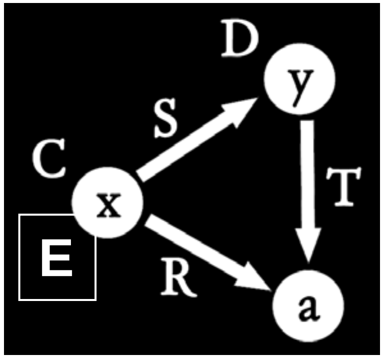
Contents

1. Reasoning Needs
2. Rules expressible in OWL
3. Extending OWL with Rules: Nominal Schemas
4. Conclusions

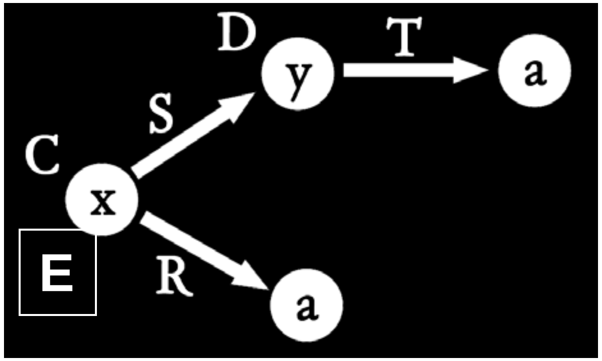
DL-safe variables

- **A generalisation of DL-safety.**
- **DL-safe variables are special variables which bind only to named individuals (like in DL-safe rules).**
- **$C(x) \wedge R(x,x_s) \wedge S(x,y) \wedge D(y) \wedge T(y,x_s) \rightarrow E(x)$**
  with x_s a safe variable

  $C(x) \wedge R(x,a) \wedge S(x,y) \wedge D(y) \wedge T(y,a) \rightarrow E(x)$
  can be translated into OWL 2.



duplicating nominals is ok



DL-safe variables

- A generalisation of DL-safety.
- DL-safe variables are special variables which bind only to named individuals (like in DL-safe rules).

- $C(x) \wedge R(x,x_s) \wedge S(x,y) \wedge D(y) \wedge T(y,x_s) \rightarrow E(x)$
  with x_s a safe variable

  $C(x) \wedge R(x,a) \wedge S(x,y) \wedge D(y) \wedge T(y,a) \rightarrow E(x)$
  can be translated into OWL 2.

- **with, say, 100 individuals, we would obtain 100 new OWL axioms from the single rule above**

DL-safety

- **DL-safe variables:**
  variables in rules which bind only to named individuals

- **Idea:**
  - **start with rule not expressible in OWL 2**
  - **select some variables and declare them DL-safe such that resulting rule can be translated into several OWL 2 rules**

- ***DL-safe rule:*** **A rule with only DL-safe variables.**

It is known that "OWL 2 DL + DL-safe rules" is decidable.
It is a *hybrid* formalism.
E.g. OWL plus DL-safe SWRL.

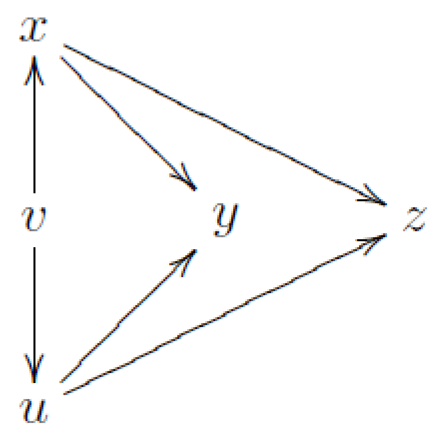
$$
\begin{aligned}
&\mathrm{hasReviewAssignment}(v,x) \wedge \mathrm{hasAuthor}(x,y) \wedge \mathrm{atVenue}(x,z)\\
&\wedge \mathrm{hasSubmittedPaper}(v,u) \wedge \mathrm{hasAuthor}(u,y) \wedge \mathrm{atVenue}(u,z)\\
&\qquad\qquad \to \mathrm{hasConflictingAssignedPaper}(v,x)
\end{aligned}
$$

assume y,z bind only to named individuals

we introduce a new construct, called

nominal schemas

or *nominal variables*



$$
\begin{aligned}
R_{\exists \mathrm{hasSubmittedPaper}.(\exists \mathrm{hasAuthor}.\{y\} \sqcap \exists \mathrm{atVenue}.\{z\})} &\circ \mathrm{hasReviewAssignment}\\
&\circ R_{\exists \mathrm{hasAuthor}.\{y\} \sqcap \exists \mathrm{atVenue}.\{z\}}\\
&\sqsubseteq \mathrm{hasConflictingAssignedPaper}
\end{aligned}
$$

Nominal schema example 2

$$\mathrm{hasChild}(x, y) \land \mathrm{hasChild}(x, z) \land \mathrm{classmate}(y, z) \to C(x)$$

$$\exists \mathrm{hasChild}.\{z\} \sqcap \exists \mathrm{hasChild}.\exists \mathrm{classmate}.\{z\} \sqsubseteq C$$

Adding nominal schemas to OWL 2 DL

- **Decidability is retained.**
- **Complexity is *the same*.**

- **A naïve implementation is straightforward:**

  Replace every axiom with nominal schemas by a set of OWL 2 axioms, obtained from *grounding* the nominal schemas.

  However, this may result in a lot of new OWL 2 axioms.
  The naïve approach will probably only work for ontologies with *few* nominal schemas.

What do we gain?

- **A powerful macro.**
- **We can actually also express all DL-safe (binary) Datalog rules!**

$$R(x,y) \wedge A(y) \wedge S(z,y) \wedge T(x,z) \rightarrow P(z,x)$$

$$\begin{aligned}
\exists U.(\{x\} \sqcap \exists R.\{y\}) \\
\sqcap \exists U.(\{y\} \sqcap A) \\
\sqcap \exists U.(\{z\} \sqcap \exists S.\{y\}) \\
\sqcap \exists U.(\{x\} \sqcap \exists T.\{z\}) \\
\sqsubseteq \exists U.(\{z\} \sqcap \exists P.\{x\})
\end{aligned}$$

A tractable fragment

Definition 2. *An occurrence of nominal schema* $\{x\}$ *in a concept* C *is* safe *if* C *contains a sub-concept of the form* $\{v\} \sqcap \exists R.D$ *for some nominal schema or nominal* $\{v\}$ *such that* $\{x\}$ *is the only nominal schema that occurs (possibly more than once) in* D*. In this case,* $\{v\} \sqcap \exists R.D$ *is a* safe environment *for this occurrence of* $\{x\}$*, sometimes written as* $S(v, x)$*.*

Definition 3. *Let* $n \geq 0$ *be an integer. A* $\mathcal{SROELV}(\sqcap, \times)$ *knowledge base* KB *is a* $\mathcal{SROELV}_n(\sqcap, \times)$ *knowledge base if in each of its axioms* $C \sqsubseteq D$*, there are at most* n *nominal schemas appearing more than once in non-safe form, and all remaining nominal schemas appear only in* C*.*

$\mathcal{SROELV}_n(\sqcap, \times)$ **is tractable (Polytime)**
covers OWL 2 EL
covers OWL 2 RL (DL-safe)
covers most of OWL 2 QL

Polytime smart transformation

$$\exists \text{hasReviewAssignment}.((\{x\} \sqcap \exists \text{hasAuthor}.\{y\}) \sqcap (\{x\} \sqcap \exists \text{atVenue}.\{z\}))$$
$$\sqcap \exists \text{hasSubmittedPaper}.(\exists \text{hasAuthor}.\{y\} \sqcap \exists \text{atVenue}.\{z\})$$
$$\sqsubseteq \exists \text{hasConflictingAssignedPaper}.\{x\}$$

becomes (a_i**,** a_j **range over all named individuals)**

$$(\exists U.O_y) \sqcap (\exists U.O_z) \sqcap \exists \text{hasReviewAssignment}.(\{a_i\} \sqcap \{a_i\})$$
$$\sqcap \exists \text{hasSubmittedPaper}.(\exists \text{hasAuthor}.O_y \sqcap \exists \text{atVenue}.O_z)$$
$$\sqsubseteq \exists \text{hasConflictingAssignedPaper}.\{a_i\}$$

$$\exists U.(\{a_i\} \sqcap \exists \text{hasAuthor}.\{a_j\}) \sqsubseteq \exists U.(\{a_j\} \sqcap O_y)$$
$$\exists U.(\{a_i\} \sqcap \exists \text{atVenue}.\{a_j\}) \sqsubseteq \exists U.(\{a_j\} \sqcap O_z)$$

OWL syntax for nominal schemas

Functional Syntax:

Add the last line, (ObjectVariable), to the ClassExpression production rule:

ClassExpression :=
Class |
ObjectIntersectionOf | ObjectUnionOf ObjectComplementOf | ObjectOneOf |
ObjectSomeValuessFrom | ObjectAllValuesFrom | ObjectHasValue | ObjectHasSelf |
ObjectMinCardinality | ObjectMaxCardinality | ObjectExactCardinality |
DataSomeValuesFrom | DataAllValuesFrom | DataHasValue |
DataMinCardinality | DataMaxCardinality | DataExactCardinality |
ObjectVariable

Add the next production rule to the grammar:

ObjectVariable := 'ObjectVariable (' quotedString ' ^^ xsd:string)'

OWL syntax for nominal schemas

Translation to Turtle:

Functional-Style Syntax	S Triples Generated in an Invocation of T(S)	Main Node of T(S)
ObjectVariable("v1" ^^ xsd:string)	_:x rdf:type owl:ObjectVariable _:x owl:variableId "v1"^^xsd:string	_:x

Naïve implemenation – experiments

	No axioms added		1 different ns		2 different ns		3 different ns	
Fam (5)	0.01”	0.00”	0.01”	0.00”	0.01”	0.00”	0.04”	0.02”
Swe (22)	3.58”	0.08”	3.73”	0.07”	3.85”	0.10”	10.86”	1.11”
Bui (42)	2.7”	0.16”	2.5”	0.15”	2.75”	0.26”	1’ 14’	6.68”
Wor (80)	0.11”	0.04”	0.12”	0.05”	1.1”	0.55”	OOM *	OOM*
Tra (183)	0.05”	0.03”	0.05”	0.02”	5.66”	1.76”	OOM	OOM
FTr (368)	0.03”	4.28”	0.05	5.32”	35.53”	42.73”	OOM	OOM
Eco (482)	0.04”	0.24”	0.07”	0.02”	56.59”	13.67”	OOM	OOM

OOM = Out of Memory

from the TONES repository:

Ontology	Classes	Data P.	Object P.	Individuals
Fam	4	1	11	5
Swe	189	6	25	22
Bui	686	0	24	42
Wor	1842	0	31	80
Tra	445	4	89	183
FTr	22	6	52	368
Eco	339	8	45	482

Naïve implemenation – experiments

Optimization through smart grounding (all ns occuring safely)

	No ns		1 ns		2 ns		3 ns	
Rex (100)	0.025	0.009	0.031	0.013	1.689	0.112	OOM	OOM
Rex Optimized (100)			0.058	0.023	0.046	0.011	0.053	0.009
Spatial (100)	0.035	0.029	0.021	0.014	1.536	0.101	OOM	OOM
Spatial Optimized (100)			0.018	0.013	0.033	0.007	0.044	0.011
Xenopus (100)	0.063	0.018	0.07	0.19	1.598	0.112	OOM	OOM
Xenopus Optimized (100)			0.099	0.037	0.083	0.018	0.097	0.063

Ontology	Classes	Data P.	Object P.	Individuals
Rex	552	0	6	100
Spatial	106	0	13	100
Xenopus	710	0	5	100

Naïve implemenation – experiments

Note: with 2 different ns we cover all of OWL 2 RL (but functionality)

	No axioms added		1 different ns		2 different ns		3 different ns	
Fam (5)	0.01”	0.00”	0.01”	0.00”	0.01”	0.00”	0.04”	0.02”
Swe (22)	3.58”	0.08”	3.73”	0.07”	3.85”	0.10”	10.86”	1.11”
Bui (42)	2.7”	0.16”	2.5”	0.15”	2.75”	0.26”	1’ 14’	6.68”
Wor (80)	0.11”	0.04”	0.12”	0.05”	1.1”	0.55”	OOM *	OOM*
Tra (183)	0.05”	0.03”	0.05”	0.02”	5.66”	1.76”	OOM	OOM
FTr (368)	0.03”	4.28”	0.05	5.32”	35.53”	42.73”	OOM	OOM
Eco (482)	0.04”	0.24”	0.07”	0.02”	56.59”	13.67”	OOM	OOM

	No ns		1 ns		2 ns		3 ns	
Rex (100)	0.025	0.009	0.031	0.013	1.689	0.112	OOM	OOM
Rex Optimized (100)			0.058	0.023	0.046	0.011	0.053	0.009
Spatial (100)	0.035	0.029	0.021	0.014	1.536	0.101	OOM	OOM
Spatial Optimized (100)			0.018	0.013	0.033	0.007	0.044	0.011
Xenopus (100)	0.063	0.018	0.07	0.19	1.598	0.112	OOM	OOM
Xenopus Optimized (100)			0.099	0.037	0.083	0.018	0.097	0.063

Contents

Conclusions

- new, tight, integration of OWL with Rules
  - no increase in complexity
  - includes a large tractable profile
  - extension of OWL syntax available

- to be done (working on it):
  - better (special-purpose) algorithms
  - tool support
  - use case experiences
  - adding local closed world features

Collaborators

Collaborators on the covered topics

David Carral Martinez, Kno.e.sis Center, Wright State University
Adila Krisnadhi, Kno.e.sis Center, Wright State University
Markus Krötzsch, Oxford University, UK
Frederick Maier, Kno.e.sis Center, Wright State University
Sebastian Rudolph, Karlsruhe Institute of Technology, Germany

References

This tutorial is very close to:

- **Adila A. Krisnadhi, Frederick Maier, Pascal Hitzler, OWL and Rules. In: Reasoning Web 2011, Springer Lecture Notes in Computer Science. To appear.** **http://pascal-hitzler.de/resources/publications/OWL-Rules-2011.pdf**

Background reading:

- **Pascal Hitzler, Markus Krötzsch, Sebastian Rudolph, Foundations of Semantic Web Technologies. Textbooks in Computing, Chapman and Hall/CRC Press, 2009.** **http://www.semantic-web-book.org/**
- **Pascal Hitzler, Markus Krötzsch, Bijan Parsia, Peter F. Patel-Schneider, Sebastian Rudolph, OWL 2 Web Ontology Language: Primer. W3C Recommendation, 27 October 2009. http://www.w3.org/TR/owl2-primer/**

References

Rules in OWL:

- Markus Krötzsch, Sebastian Rudolph, Pascal Hitzler, Description Logic Rules. In: Malik Ghallab, Constantine D. Spyropoulos, Nikos Fakotakis, Nikos Avouris (eds.), Proceedings of the 18th European Conference on Artificial Intelligence, ECAI2008, Patras, Greece, July 2008. IOS Press, 2008, pp. 80-84. http://pascal-hitzler.de/resources/publications/dlrules-ecai08.pdf
- Markus Krötzsch. Description Logic Rules. Studies on the Semantic Web, Vol. 008, IOS Press, 2010. http://www.semantic-web-studies.net/

References

Nominal Schemas:

- Markus Krötzsch, Frederick Maier, Adila Alfa Krisnadhi, Pascal Hitzler, A Better Uncle For OWL – Nominal Schemas for Integrating Rules and Ontologies. In: S. Sadagopan, Krithi Ramamritham, Arun Kumar, M.P. Ravindra, Elisa Bertino, Ravi Kumar (eds.), WWW '11 20th International World Wide Web Conference, Hyderabad, India, March/April 2011. ACM, New York, 2011, pp. 645-654. http://pascal-hitzler.de/resources/publications/WWW2011.pdf
- Adila A. Krisnadhi, Frederick Maier, Pascal Hitzler, OWL and Rules. In: Reasoning Web 2011, Springer Lecture Notes in Computer Science. To appear. http://pascal-hitzler.de/resources/publications/OWL-Rules-2011.pdf
- David Carral Martínez, Pascal Hitzler, Adila A. Krisnadhi, Syntax Proposal for Nominal Schemas. Technical Report, Kno.e.sis Center, Wright State University, Dayton, OH, May 2011.