OWL and Rules

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

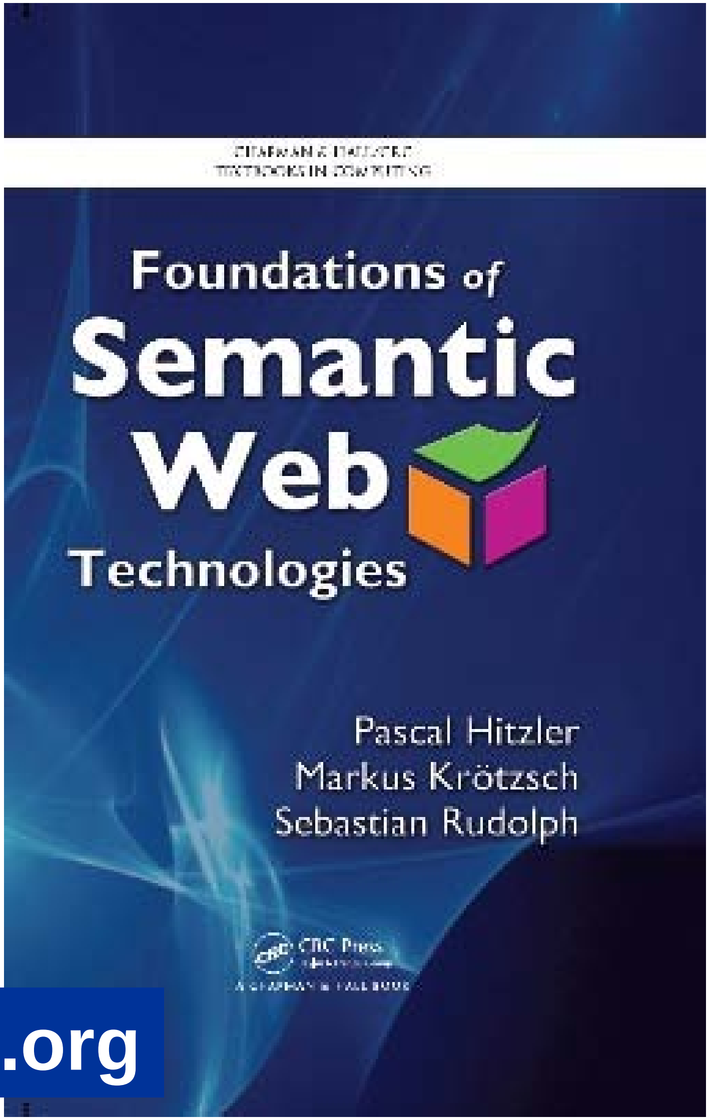
Textbook

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

Semantic Web journal

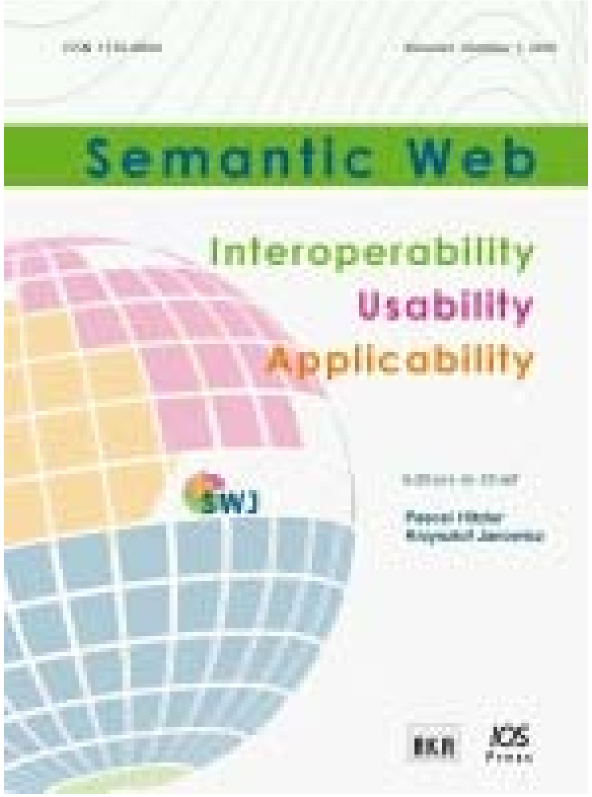
kno.e.sis

- **EiCs:** Pascal Hitzler
  Krzysztof Janowicz

- **New journal with significant initial uptake.**

- **We very much welcome contributions at the “rim” of traditional Semantic Web research – e.g., work which is strongly inspired by a different field.**

- **Non-standard (open & transparent) review process.**



- **http://www.semantic-web-journal.net/**

The Kno.e.sis Center

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

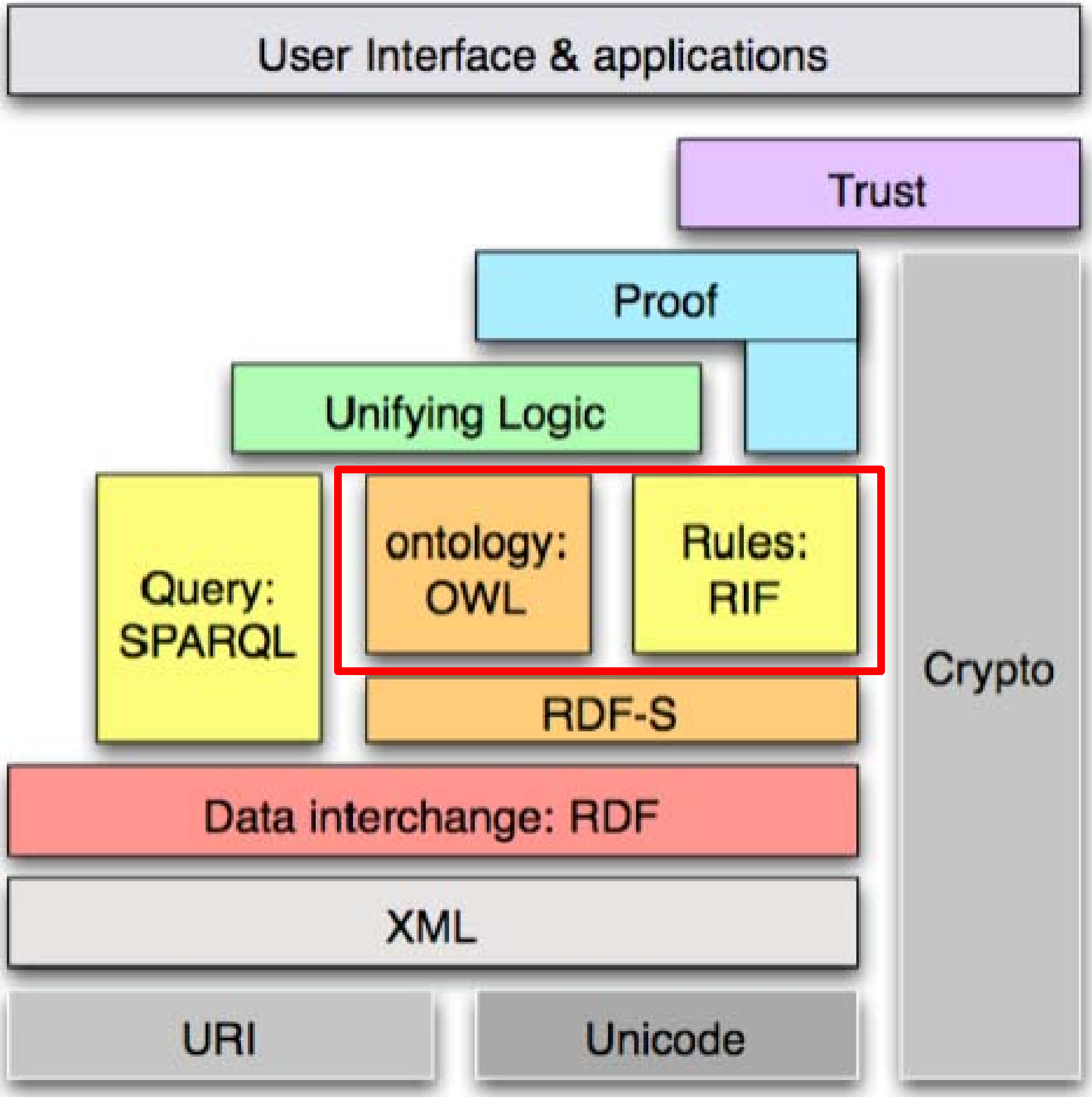
- **A primary location of Semantic Web research, but also pursuing other topics.**

- **15 faculty across 4 colleges**
  9 from Computer Science
  ca. 50 PhD students plus MS and BS students

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

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

OWL and Rules: Two paradigms?



A brief history

- 2001-2004: Description Logics make the W3C OWL standard
  Logic programming continues to be used for ontology modeling
- 2004: Description Logic Programs (DLP) [Grosof et al, WWW 03]
  "intersection of Datalog and OWL 1 DL"
- 2004: Semantic Web Rules Language (SWRL) [W3C member sub]
  "rules on top of OWL" – undecidable
- 2005/2006: Motik et al., reintroducing "DL-Safety" (can be traced back to Rosati end of 90s). [e.g. JWS 2006]
  DL-safe SWRL is decidable
- 2007: Motik and Rosati: hybrid MKNF based on DL-safe SWRL (non-monotonic extension)
- 2006-2009: OWL 2 WG by W3C
- 2008-10: Description logic rules, ELP (significantly enhanced DLP) [Krötzsch, Rudolph, Hitzler] (we'll cover most of this here)
- 2011: Nominal schemas (strong integration of OWL 2 and DL-safe SWRL) [Krötzsch, Maier, Krisnadhi, Hitzler] (we'll cover this here)

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> .

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

Reasoning Needs

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

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

in OWL:

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

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

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

these are not rules!

Reasoning Needs

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

$\text{NumberedAuthor} \sqsubseteq$

$\qquad \exists \text{authorNumber.<xsd:positiveInteger>} \sqcap \exists \text{authorName}.\top$

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

$\text{Paper}(x) \wedge \text{hasAuthorNumbered}(x,y) \wedge \text{authorNumber}(y,1) \wedge$

$\qquad \text{authorName}(y,z) \rightarrow \text{hasFirstAuthor}(x,z)$

in OWL:

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

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

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

$\qquad \sqsubseteq \text{hasFirstAuthor}$

Reasoning as first-class citizen

Why would we want to have knowledge/rules such as

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

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?

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

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

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

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

Which rules can be encoded in OWL?

$$A \sqsubseteq \neg B \sqcup C \text{ becomes } A(x) \wedge B(x) \rightarrow C(x)$$

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

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

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

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

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

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

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

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

Rolification

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

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

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

careful – regularity of RBox needs to be retained:

$$\text{hasHusband} \sqsubseteq \text{marriedTo}$$

Rolification

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

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

Rules in OWL 2

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

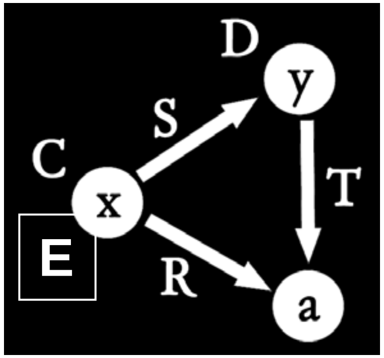
- $\text{NutAllergic}(x) \wedge \text{NutProduct}(y) \rightarrow \text{dislikes}(x,y)$
  - $\text{NutAllergic} \equiv \exists\text{nutAllergic}.\text{Self}$
    $\text{NutProduct} \equiv \exists\text{nutProduct}.\text{Self}$
    $\text{nutAllergic} \circ U \circ \text{nutProduct} \sqsubseteq \text{dislikes}$

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

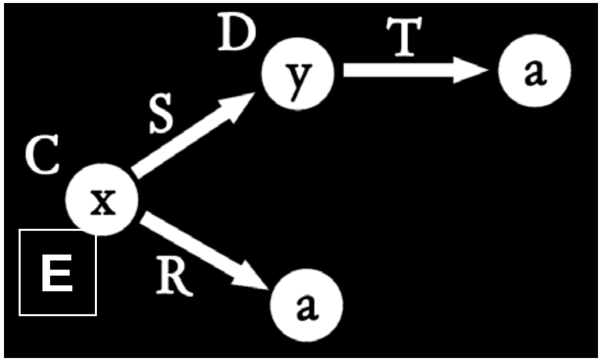
So how can we pinpoint this?

- **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

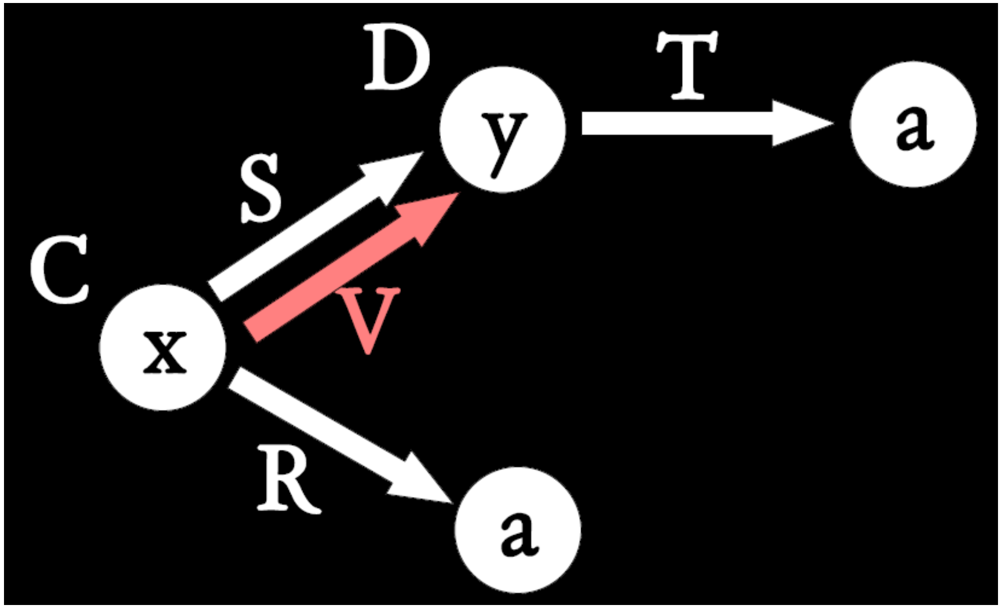


So how can we pinpoint this?

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

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

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



Rule bodies as graphs

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

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

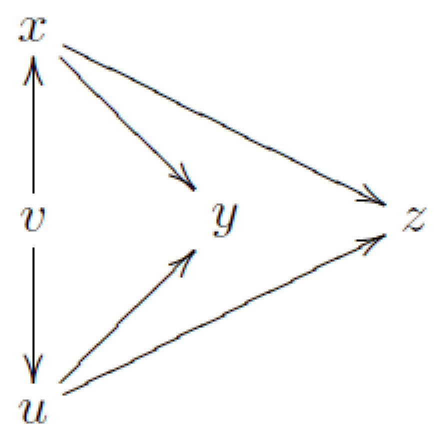
C ⊓ ∃R.{a} ⊑ ∃R1.Self

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'.**

$$C(x) \land R(x,a) \land S(x,y) \land D(y) \land T(y,a) \to P(x,y)$$

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

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).*

$R(x,z) \wedge S(y,z) \to T(x,y)$ 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 $\mathcal{SROIEL}$.*

SROIQ Rules

- A hybrid syntax
- Allow acyclic rules
  however, predicates can be SROIQ class expressions
- Such KBs can be transformed in polytime back into SROIQ
- This enables
  - A rule-based syntax for DL modeling
  - Follow-up work on integrating rules and OWL

SROIQ Rules example

NutAllergic(sebastian)
NutProduct(peanutOil)
$\exists$orderedDish.ThaiCurry(sebastian)

ThaiCurry $\sqsubseteq$ $\exists$contains.{peanutOil}
$\top \sqsubseteq \forall$orderedDish.Dish

NutAllergic(x) $\wedge$ NutProduct(y) $\rightarrow$ dislikes(x,y)
dislikes(x,z) $\wedge$ Dish(y) $\wedge$ contains(y,z) $\rightarrow$ dislikes(x,y)
orderedDish(x,y) $\wedge$ dislikes(x,y) $\rightarrow$ Unhappy(x)

!not a SROIQ Rule!

SROIQ Rules normal form

- Each SROIQ Rule can be written ("linearised") such that
  - the body-tree is linear,
  - if the head is of the form R(x,y), then y is the leaf of the tree, and
  - if the head is of the form C(x), then the tree is only the root.

- worksAt(x,y) ∧ University(y) ∧ supervises(x,z) ∧ PhDStudent(z) → professorOf(x,z)
  - ∃worksAt.University(x) ∧ supervises(x,z) ∧ PhDStudent(z) → professorOf(x,z)

- C(x) ∧ R(x,a) ∧ S(x,y) ∧ D(y) ∧ T(y,a) → V(x,y)
  - (C ⊓ ∃R.{a})(x) ∧ S(x,y) ∧ (D ⊓ ∃T.{a})(y) → V(x,y)

DL-safe variables

- **Idea: Say, you have a rule which which violates the tree (or acyclicity) condition:**

  dislikes(x,z) ∧ Dish(y) ∧ contains(y,z) → dislikes(x,y)

  Then pick a variable which destroys the tree-ness (here, z) and make it a *DL-safe variable*. By definition, these can bind only to known individuals.

- **The above rule can then be converted (*grounded*) into n tree-shaped rules (where n is the number of individuals in the knowledge base).**

- **Doing this with SROEL (OWL 2 EL) as underlying logic, essentially results in the polynomial *ELP*.**

ELP example

NutAllergic(sebastian)

NutProduct(peanutOil)

$\exists$orderedDish.ThaiCurry(sebastian)

ThaiCurry $\sqsubseteq$ $\exists$contains.{peanutOil}

$\top \sqsubseteq \forall$orderedDish.Dish

NutAllergic(x) $\wedge$ NutProduct(y) $\rightarrow$ dislikes(x,y)

dislikes(x,z) $\wedge$ Dish(y) $\wedge$ contains(y,z) $\rightarrow$ dislikes(x,y)

orderedDish(x,y) $\wedge$ dislikes(x,y) $\rightarrow$ Unhappy(x)

ELP example

$NutAllergic(sebastian)$

$NutProduct(peanutOil)$

$\exists orderedDish.ThaiCurry(sebastian)$

$ThaiCurry \sqsubseteq \exists contains.\{peanutOil\}$

$\top \sqsubseteq \forall orderedDish.Dish$

$NutAllergic(x) \land NutProduct(y) \to dislikes(x,y)$

$dislikes(x,z) \land Dish(y) \land contains(y,z) \to dislikes(x,y)$

$orderedDish(x,y) \land dislikes(x,y) \to Unhappy(x)$

Conclusions:

dislikes(sebastian,peanutOil)

ELP example

NutAllergic(sebastian)

NutProduct(peanutOil)

∃orderedDish.ThaiCurry(sebastian)

ThaiCurry ⊑ ∃contains.{peanutOil}

⊤ ⊑ ∀orderedDish.Dish

orderedDish rdfs:range Dish.

NutAllergic(x) ∧ NutProduct(y) → dislikes(x,y)

dislikes(x,z) ∧ Dish(y) ∧ contains(y,z) → dislikes(x,y)

orderedDish(x,y) ∧ dislikes(x,y) → Unhappy(x)

Conclusions:

dislikes(sebastian,peanutOil)

orderedDish(sebastian,y_s)

ThaiCurry(y_s)

Dish(y_s)

ELP example

NutAllergic(sebastian)

NutProduct(peanutOil)

$\exists$orderedDish.ThaiCurry(sebastian)

ThaiCurry $\sqsubseteq$ $\exists$contains.{peanutOil}

$\top \sqsubseteq \forall$orderedDish.Dish

NutAllergic(x) $\wedge$ NutProduct(y) $\rightarrow$ dislikes(x,y)

dislikes(x,z) $\wedge$ Dish(y) $\wedge$ contains(y,z) $\rightarrow$ dislikes(x,y)

orderedDish(x,y) $\wedge$ dislikes(x,y) $\rightarrow$ Unhappy(x)

Conclusions:

dislikes(sebastian,peanutOil)

orderedDish(sebastian,y_s)

ThaiCurry(y_s)

Dish(y_s)

contains(y_s,peanutOil)

ELP example

NutAllergic(sebastian)

NutProduct(peanutOil)

∃orderedDish.ThaiCurry(sebastian)

ThaiCurry ⊑ ∃contains.{peanutOil}

⊤ ⊑ ∀orderedDish.Dish

z DL-safe variable

NutAllergic(x) ∧ NutProduct(y) → dislikes(x,y)

dislikes(x,z) ∧ Dish(y) ∧ contains(y,z) → dislikes(x,y)

orderedDish(x,y) ∧ dislikes(x,y) → Unhappy(x)

Conclusions:

dislikes(sebastian,peanutOil)

orderedDish(sebastian,y_s)

ThaiCurry(y_s)

Dish(y_s)

contains(y_s,peanutOil)

dislikes(sebastian,y_s)

ELP example

NutAllergic(sebastian)
NutProduct(peanutOil)
∃orderedDish.ThaiCurry(sebastian)

ThaiCurry $\sqsubseteq$ ∃contains.{peanutOil}
$\top \sqsubseteq$ ∀orderedDish.Dish

NutAllergic(x) $\wedge$ NutProduct(y) $\rightarrow$ dislikes(x,y)
dislikes(x,z) $\wedge$ Dish(y) $\wedge$ contains(y,z) $\rightarrow$ dislikes(x,y)
orderedDish(x,y) $\wedge$ dislikes(x,y) $\rightarrow$ Unhappy(x)

Conclusions:
dislikes(sebastian,peanutOil)
orderedDish(sebastian,y_s)
ThaiCurry(y_s)
Dish(y_s)

contains(y_s,peanutOil)
dislikes(sebastian,y_s)
Unhappy(sebastian)

ELP example

NutAllergic(sebastian)
NutProduct(peanutOil)
$\exists$orderedDish.ThaiCurry(sebastian)

ThaiCurry $\sqsubseteq$ $\exists$contains.{peanutOil}
$\top \sqsubseteq \forall$orderedDish.Dish

NutAllergic(x) $\wedge$ NutProduct(y) $\rightarrow$ dislikes(x,y)
dislikes(x,z) $\wedge$ Dish(y) $\wedge$ contains(y,z) $\rightarrow$ dislikes(x,y)
orderedDish(x,y) $\wedge$ dislikes(x,y) $\rightarrow$ Unhappy(x)

Conclusion: Unhappy(sebastian)

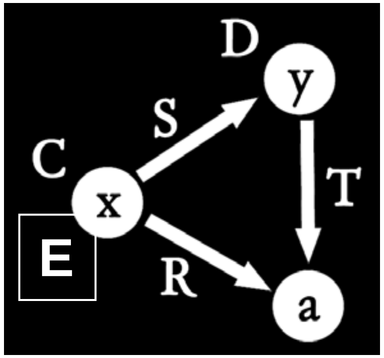
Contents

DL-safe variables

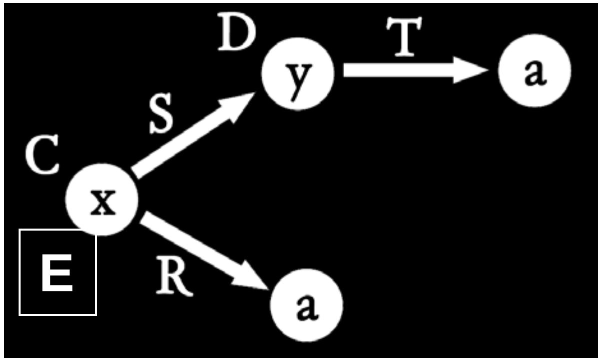
- A generalisation of DL-safety.
- DL-safe variables are special variables which bind only to named individuals (like in DL-safe rules).

- $C(x) \wedge R(x,x_s) \wedge S(x,y) \wedge D(y) \wedge T(y,x_s) \rightarrow E(x)$
  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.

Non-hybrid syntax: nominal schemas

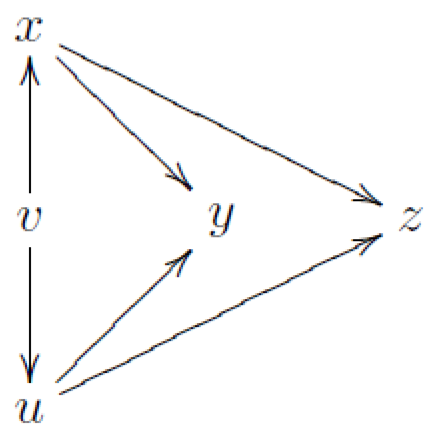
$$\text{hasReviewAssignment}(v,x) \land \text{hasAuthor}(x,y) \land \text{atVenue}(x,z)$$
$$\land\ \text{hasSubmittedPaper}(v,u) \land \text{hasAuthor}(u,y) \land \text{atVenue}(u,z)$$
$$\rightarrow \text{hasConflictingAssignedPaper}(v,x)$$

assume y,z bind only to named individuals

we introduce a new construct, called

nominal schemas

or *nominal variables*



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

Nominal schema example 2

$$\text{hasChild}(x, y) \wedge \text{hasChild}(x, z) \wedge \text{classmate}(y, z) \rightarrow C(x)$$

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

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.**
- **A conceptual bridge to rule formalism:**

We can actually also express all DL-safe Datalog rules!

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

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

Expressing (DL-safe) Datalog

Given a Datalog rule $A_1, \ldots, A_n \rightarrow A$, where A and all A_i are atomic formulas of the form $p(x_1, \ldots, x_n)$ with the x_i being variables, we translate this rule into the DL axiom $\tau(A_1) \sqcap \cdots \sqcap \tau(A_n) \sqsubseteq \tau(A)$. For an atomic formula $p(x_1, \ldots, x_n)$, we define $\tau(p(x_1, \ldots, x_n))$ to be the DL class expression

$$\exists U.(\exists p_1.\{x_1\} \sqcap \cdots \sqcap \exists p_n.\{x_n\}),$$

where U is the universal role and $p_1, \ldots, p_n$ are role names used exclusively for encoding occurrences of the n-ary predicate symbol p. If x_i is a constant, then the corresponding nominal schema becomes a nominal.

Theorem 1. *The transformation just described converts a set P of Datalog rules into a $\mathcal{SROELV}$ knowledge base K, such that, for any n-ary predicate symbol p in P and any n-tuple $(a_1, \ldots, a_n)$ of constants in P, we have that $P \models p(a_1, \ldots, a_n)$ if and only if $K \models \top \sqsubseteq \exists U.(\exists p_1.\{a_1\} \sqcap \cdots \sqcap \exists p_n.\{a_n\})$*

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

Delayed grounding

- **Adding nominal schemas to existing tableaux algorithms:**

$$\begin{aligned} \text{grounding}: \quad & \text{if } C \in \mathsf{L}(s), \{z\} \text{ is a nominal schema in } C, \\ & \quad C[z/a_i] \notin \mathsf{L}(s) \text{ for some } i, 1 \leq i \leq \ell \\ & \text{then } \mathsf{L}(s) := \mathsf{L}(s) \cup \{C[z/a_i]\} \end{aligned}$$

plus some restrictions on existing tableaux rules, essentially to ensure that (1) no variable binding is broken and (2) nominal schemas are not propagated through the tableau.

Delayed grounding

$$\exists \text{hasReviewAssignment}.((\{x\} \sqcap \exists \text{hasAuthor}.\{y\}) \sqcap (\{x\} \sqcap \exists \text{atVenue}.\{z\}))$$
$$\sqcap\, \exists \text{hasSubmittedPaper}.(\exists \text{hasAuthor}.\{y\} \sqcap \exists \text{atVenue}.\{z\})$$
$$\sqsubseteq \exists \text{hasConflictingAssignedPaper}.\{x\}$$
$$\{p_0\} \sqsubseteq \exists \text{hasAuthor}.\{a_{1000}\} \sqcap \exists \text{hasAuthor}.\{a_1\}$$
$$\{p_i\} \sqsubseteq \exists \text{hasAuthor}.\{a_i\} \sqcap \exists \text{hasAuthor}.\{a_{i+1}\}$$
$$\{a_i\} \sqsubseteq \exists \text{hasSubmittedPaper}.\{p_{i-1}\} \sqcap \exists \text{hasSubmittedPaper}.\{p_i\}$$
$$\{a_{1000}\} \sqsubseteq \exists \text{hasSubmittedPaper}.\{p_{999}\} \sqcap \exists \text{hasSubmittedPaper}.\{p_0\}$$
$$\{p_j\} \sqsubseteq \exists \text{AtVenue}.\{\text{ISWC}\}$$
$$\{a_k\} \sqsubseteq \exists \text{hasReviewAssignment}.\{p_{k-4}\} \sqcap \exists \text{hasReviewAssignment}.\{p_{k-3}\}$$
$$\{a_1\} \sqsubseteq \exists \text{hasReviewAssignment}.\{p_{999}\} \sqcap \exists \text{hasReviewAssignment}.\{p_{998}\}$$

Fig. 1. Example for delayed grounding. $i = 1, \ldots, 999$, $j = 0, \ldots, 999$, $k = 4, \ldots, 1000$.

$$\forall \text{hasConflictingAssignedPaper}.\bot \text{ is unsatisfiable}$$

Towards a unifying logic?

- **Straightforward carrying over of circumscription to DLs:**
  undecidable for expressive DLs
  [Bonatti, Lutz, Wolter, KR2006, JAIR 2009]

  Unintuitive modeling: extensions of minimized predicates may contain unknown individuals

- **Fixing the unintuitive aspect: allow only named individuals in extensions of minimized predicates**
  decidable even for very expressive DLs
  we also have a tableaux algorithm
  [Sengupta, Krisnadhi, Hitzler, ISWC2011]

  called *Grounded Circumscription*

Circumscription

- **Use a knowledge base K as usual.**
- **Additionally, specify "circumscribed" (minimized) predicates.**

- **Among all models M of K, the circumscribed models (c-models) are those for which there is no model which is preferred over M.**

A model J is *preferred over* M if
a) they have the same domain of discourse
b) constants have the same extensions in both models
c) the J-extension of each minimized predicate is contained in its M-extension
d) the J-extension of some minimized predicate is strictly contained in its M-extension

Grounded Circumscription for DLs

- **Use a knowledge base K as usual.**
- **Additionally, specify "circumscribed" (minimized) predicates.**

- **Among all models M of K, the circumscribed models (gc-models) are those for which there is no model which is preferred over M and extensions of minimized predicates contain only named individuals.**

A model J is *preferred over* M if
a) they have the same domain of discourse
b) constants have the same extensions in both models
c) the J-extension of each minimized predicate is contained in its M-extension
d) the J-extension of some minimized predicate is strictly contained in its M-extension

Circumscription vs. Grounded Circ.

- **Circumscription:**
  - **minimization of roles leads to undecidability (for non-empty Tboxes**

- **Grounded Circumscription:**
  - **Decidable even under role grounding for very expressive decidable DLs.**
  - **Complexity upper bound for satisfiability or for finding a gc-model is EXP^C, where C is the complexity of the underlying DL.**

We also have a tableaux algorithm for different reasoning tasks.

Example

$$\mathtt{hasAuthor}(\mathtt{paper1}, \mathtt{author1}) \qquad \mathtt{hasAuthor}(\mathtt{paper1}, \mathtt{author2})$$

$$\mathtt{hasAuthor}(\mathtt{paper2}, \mathtt{author3}) \qquad \top \sqsubseteq \forall \mathtt{hasAuthor}.\mathtt{Author}$$

Both of

$$\neg \mathtt{hasAuthor}(\mathtt{paper1}, \mathtt{author3})$$

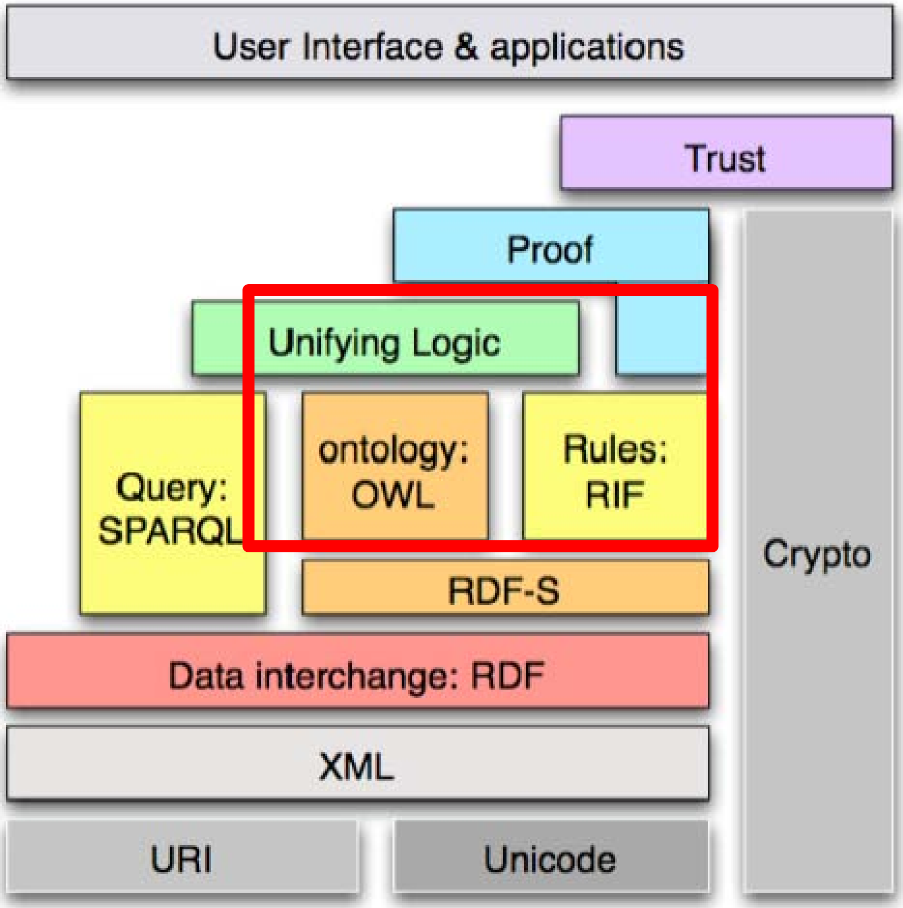
$$(\leq 2\ \mathtt{hasAuthor}.\mathtt{Author})(\mathtt{paper1})$$

are not logical consequences under classical DL semantics.

However, they are logical consequences when hasAuthor is minimized (using the UNA).

Towards a unifying logic

- We now have a strong integration of datalog and OWL.
- There's plenty of work on non-monotonic DLs.
- The next logical step would be to create a non-monotonic DL which conservatively extends both OWL and some major non-monotonic rule formalism.



Contents

Conclusions

- new, tight, integration of OWL with Rules
  - no increase in complexity
  - includes a large tractable profile
  - extension of OWL syntax available
  - first algorithms
- 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
Kunal Sengupta, Kno.e.sis Center, Wright State University

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.** **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, Adila A. Krisnadhi, Frederick Maier, Kunal Sengupta, Pascal Hitzler, Reconciling OWL and Rules. Technical Report, Kno.e.sis Center, Wright State University, Dayton, OH, May 2011.

(Grounded) Circumscription

- Piero A. Bonatti, Carsten Lutz, Frank Wolter, The Complexity of Circumscription in DLs. Journal of Artificial Intelligence Research (JAIR) 35: 717-773, 2009.
- Stephan Grimm, Pascal Hitzler, Semantic Matchmaking of Web Resources with Local Closed-World Reasoning. International Journal of e-Commerce 12(2), 89-126, Winter 2007-8.
- Stephan Grimm, Pascal Hitzler, A Preferential Tableaux Calculus for Circumscriptive ALCO. In: Axel Polleres, Terrance Swift (eds.), Web Reasoning and Rule Systems, Third International Conference, RR2009, Chantilly, VA, USA, October 2009, Proceedings. LNCS 5837, pp. 40-54.
- Kunal Sengupta, Adila Krisnadhi, Pascal Hitzler, Local Closed World Reasoning: Grounded Circumscription for OWL. In: Proceedings ISWC2011, to appear.