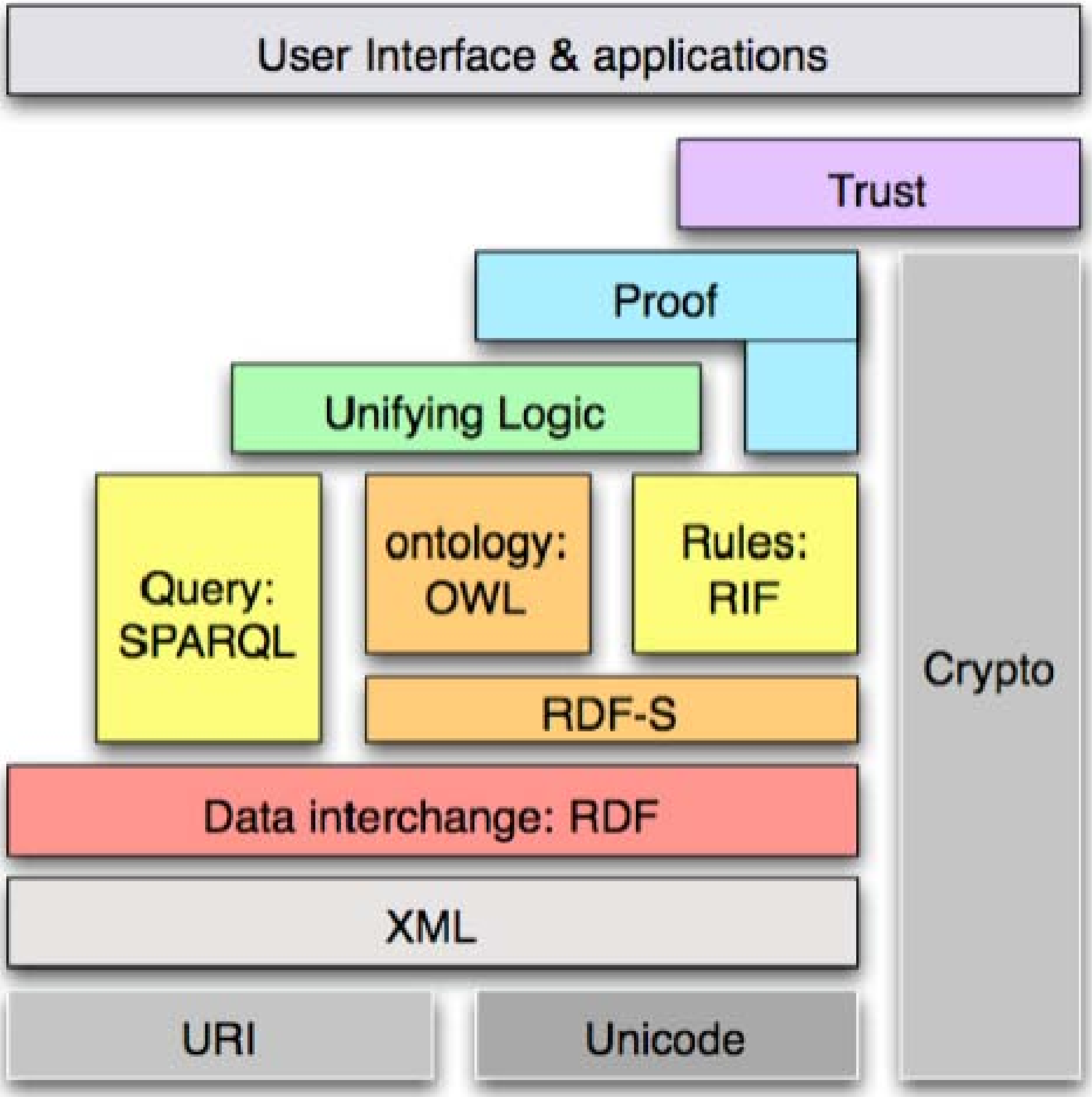
Ontologies and Rules

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

The Semantic Web Stack



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)

A brief history

- 2009: OWL 2 and RIF (Datalog/Logic Programming) W3C standards
- 2008-10: Description logic rules, ELP (significantly enhanced DLP) [Krötzsch, Rudolph, Hitzler]
- 2011: Nominal schemas (strong integration of OWL 2, DL-safe SWRL, Datalog) [Krötzsch, Maier, Krisnadhi, Hitzler]
- 2012: Incorporating non-monotonic rules (strong integration paradigm) [Knorr, Hitzler, Maier]: see Matthias Knorr's class next week
- 2013: Overcoming regularity restrictions [Carral, Hitzler] (first results)
- Ongoing: Algorithm Development for Nominal Schemas [Carral, Hitzler, Krisnadhi, Wang; Steigmiller, Glimm, Liebig] (first results)

Contents

Rules

$$A(x) \wedge R(x,y) \wedge S(y,z) \wedge B(z) \rightarrow C(x)$$

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

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

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

Rules

A (*datalog*) *rule* is an expression of the form

$$A_1 \wedge \ldots \wedge A_n \rightarrow B$$

where B (the head) and all A_i (the body) are *atoms* of the form

$$p(x_1,\ldots,x_k),$$

where the x_i can be variables or constants.

If n=0, the rule is called a *fact* (and the arrow is omitted).

Variables can be considered to be universally quantified.

A set of rules can carry

- a first-order logic semantics or
- a Herbrand semantics
  (essentially, rules apply only to constants present in the set, and logical consequences are restricted to variable-free atoms).

Example

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.

Example

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

Another Example

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> .

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

Another Example

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

hasAuthorNumbered(x,y) ∧ authorName(y,z) → hasAuthor(x,z)

in OWL:

Paper ⊑ ∃hasAuthorNumbered.NumberedAuthor

NumberedAuthor ⊑
∃authorNumber.<xsd:positiveInteger> ⊓ ∃authorName.⊤

hasAuthorNumbered ∘ authorName ⊑ hasAuthor

these are not rules!

Another Example

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

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

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

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

in OWL:

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

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

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

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?

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

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

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

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

Rules in OWL

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?

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

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

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

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

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.,

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

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

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

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

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

careful – regularity of RBox needs to be retained:

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

Rolification

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

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

Rules in OWL 2

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

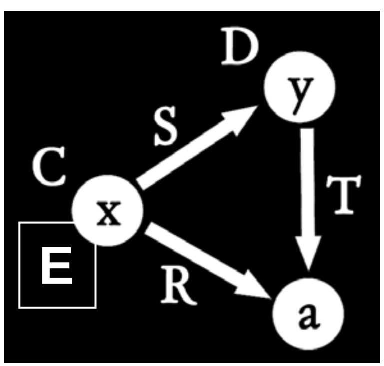
- **$NutAllergic(x) \wedge NutProduct(y) \rightarrow dislikes(x,y)$**
  - **$NutAllergic \equiv \exists nutAllergic.Self$**
    $NutProduct \equiv \exists nutProduct.Self$
    $nutAllergic \circ U \circ nutProduct \sqsubseteq dislikes$

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

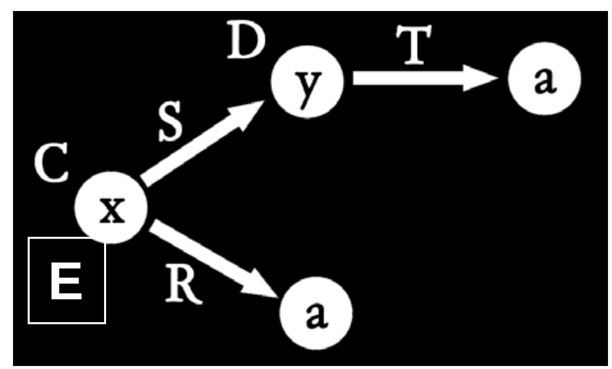
So how can we pinpoint this?

- 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

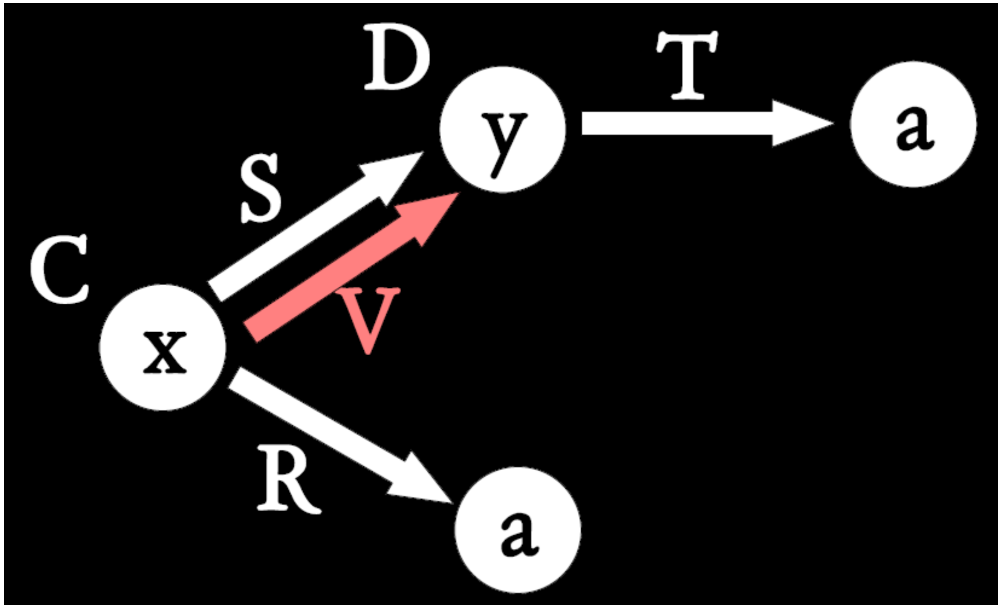


So how can we pinpoint this?

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

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

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



Rule bodies as graphs

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

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

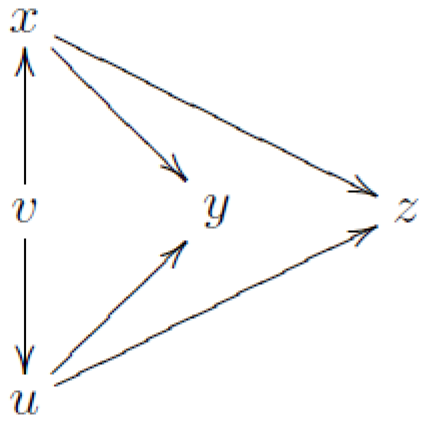
C ⊓ ∃R.{a} ⊑ ∃R1.Self

D⊓ ∃T.{a}) ⊑ ∃R2.Self

R1 ∘ S ∘ R2 ⊑ P

Rule bodies as graphs

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



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.*

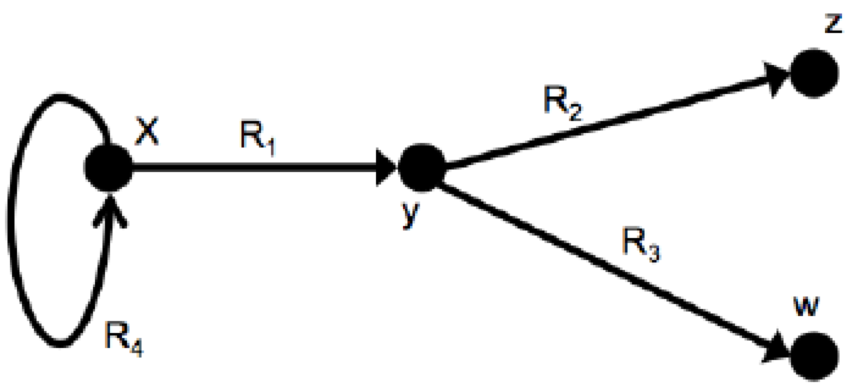
- *Each of the maximally connected components of the corresponding graph is in fact a tree (respectively, an acyclic graph)—or in other words, if it is a* forest, *i.e., a set of trees (respectively, a set of acyclic graphs).*
- *If H consists of an atom $A(t)$ or $R(t, u)$, then t is a root in the tree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

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

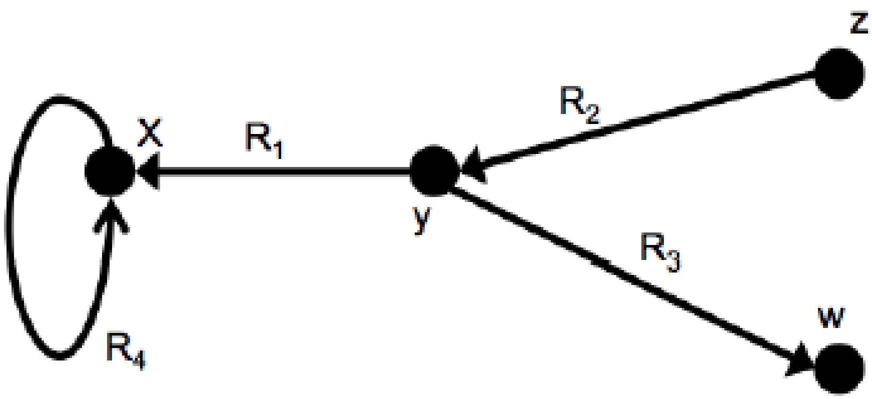
$$R_1(x, y) \wedge C_1(y) \wedge R_2(y, w) \wedge R_3(y, z) \wedge C_2(z) \wedge R_4(x, x) \rightarrow C_3(x)$$



$$\exists R_1.(C_1 \sqcap \exists R_2.\top \sqcap \exists R_3.C_2) \sqcap \exists R_4.\mathit{Self} \sqsubseteq C_3$$

Acyclic Rules

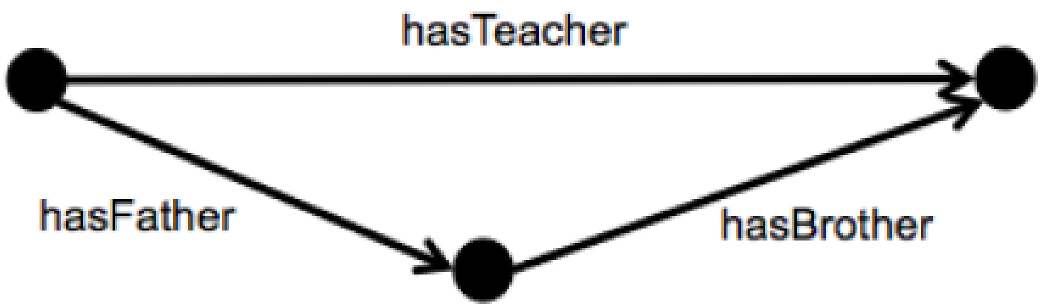
$$R_1(y,x) \wedge C_1(y) \wedge R_2(w,y) \wedge R_3(y,z) \wedge C_2(z) \wedge R_4(x,x) \rightarrow C_3(x)$$



$$\exists R_1^-.(C_1 \sqcap \exists R_2^-.\top \sqcap \exists R_3.C_2) \sqcap \exists R_4.Self \sqsubseteq C_3$$

Use of role conjunction

$$\mathsf{hasFather}(x,y) \wedge \mathsf{hasBrother}(y,z) \wedge \mathsf{hasTeacher}(x,z) \rightarrow \mathsf{TaughtByUncle}(x)$$



$$\mathsf{hasFather}(x,y) \wedge \mathsf{hasBrother}(y,z) \rightarrow \mathit{hasUncle}(x,z)$$

$$\mathit{hasUncle}(x,z) \wedge \mathsf{hasTeacher}(x,z) \rightarrow \mathsf{TaughtbyUncle}(x)$$

$$\mathsf{hasFather} \circ \mathsf{hasBrother} \sqsubseteq \mathsf{hasUncle}$$

Use of role conjunction

$$\text{hasFather}(x, y) \wedge \text{hasBrother}(y, z) \rightarrow \mathit{hasUncle}(x, z)$$
$$\mathit{hasUncle}(x, z) \wedge \text{hasTeacher}(x, z) \rightarrow \text{TaughtbyUncle}(x)$$

$$\text{hasFather} \circ \text{hasBrother} \sqsubseteq \text{hasUncle}$$

Middle rule:

$$\mathit{hasUncle}(x, z) \wedge \text{hasTeacher}(x, z) \rightarrow \text{TaughtbyUncle}(x)$$

Equivalent Translation:

$$\text{hasUncle}(x, z) \wedge \text{hasTeacher}(x, z) \rightarrow \mathit{hasUncleAndTeacher}(x, z)$$
$$\mathit{hasUncleAndTeacher}(x, z) \rightarrow \text{TaughtbyUncle}(x)$$

$$\textbf{hasUncle} \sqcap \textbf{hasTeacher} \sqsubseteq \textbf{hasUncle}$$
$$\exists \mathit{hasUncleAndTeacher}.\top \sqsubseteq \mathit{TaughtByUncle}$$

Use of role conjunction

$$\text{hasFather}(x, y) \wedge \text{hasBrother}(y, z) \wedge \text{hasTeacher}(x, z) \rightarrow \text{TaughtByUncle}(x)$$

$$\text{hasFather} \circ \text{hasBrother} \sqsubseteq \text{hasUncle}$$

$$\textbf{hasUncle} \sqcap \textbf{hasTeacher} \sqsubseteq \textbf{hasUncle}$$

$$\exists \text{hasTeacherAndUncle}.\top \sqsubseteq \text{TaughtByUncle}$$

Role conjunction is unproblematic for *simple* roles.

More complications

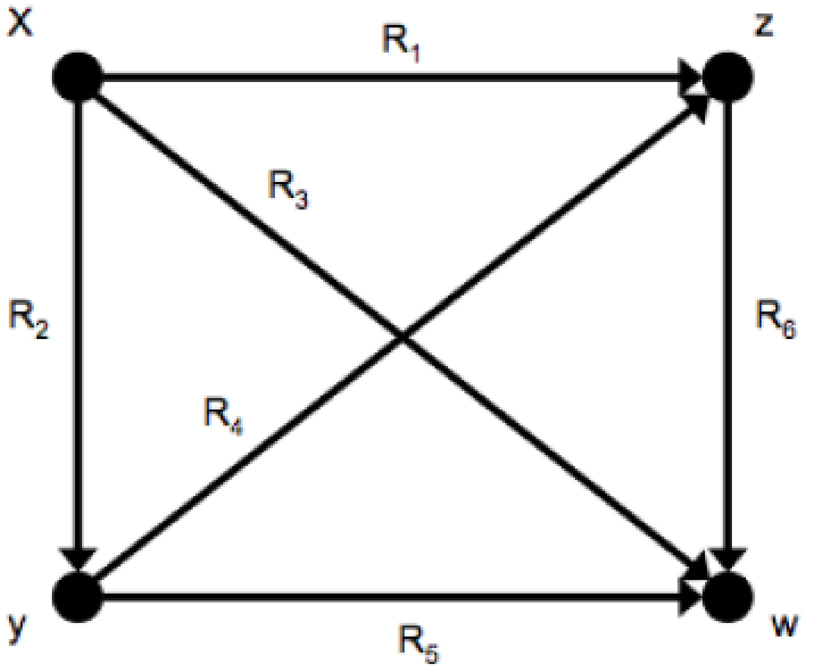
Small logics without regularity restrictions are already undecidable, e.g.

$\mathcal{ERI}$

Constructor	Name	Syntax	Semantics
	Tbox Axiom (GCI)	$C \sqsubseteq D$	$C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$
	Rbox Axiom (RIA)	$R \sqsubseteq S$	$R^{\mathcal{I}} \subseteq S^{\mathcal{I}}$
$\mathcal{E}$	Existential Restriction	$\exists R.C$	$\{\delta \mid$ there is ϵ with $\langle \delta, \epsilon \rangle \in R^{\mathcal{I}}$ and $\epsilon \in C^{\mathcal{I}}\}$
$\mathcal{R}$	Role Chain (RIA)	$R_1 \circ \ldots \circ R_n \sqsubseteq S$	$R_1^{\mathcal{I}} \circ \ldots \circ R_n^{\mathcal{I}} \subseteq S^{\mathcal{I}}$
$\mathcal{I}$	Role Inverse	R^-	$\{\langle \delta, \epsilon \rangle \mid \langle \epsilon, \delta \rangle \in V^{\mathcal{I}}\}$

Not expressible even with conjunction

$$R_1(x,y) \wedge R_2(x,z) \wedge R_3(x,w) \wedge R_4(y,z) \wedge R_5(y,w) \wedge R_6(w,z) \rightarrow C(x)$$



Contents

1. Initial examples
2. Rules expressible in description logics
3. Extending description logics with rules through nominal schemas
4. Algorithmizations for nominal schemas
5. Conclusions

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**

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 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

$NutAllergic(sebastian)$

$NutProduct(peanutOil)$

$\exists orderedDish.ThaiCurry(sebastian)$

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

$\top \sqsubseteq \forall orderedDish.Dish$

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

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

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

Conclusions:

dislikes(sebastian,peanutOil)

ELP example

NutAllergic(sebastian)

NutProduct(peanutOil)

$\exists$orderedDish.ThaiCurry(sebastian)

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

$\top$ $\sqsubseteq$ $\forall$orderedDish.Dish

orderedDish rdfs:range 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)

ELP example

NutAllergic(sebastian)

NutProduct(peanutOil)

∃orderedDish.ThaiCurry(sebastian)

ThaiCurry ⊑ ∃contains.{peanutOil}

⊤ ⊑ ∀orderedDish.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)

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 ⊑ ∃contains.{peanutOil}

⊤ ⊑ ∀orderedDish.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)

contains(y_s,peanutOil)

dislikes(sebastian,y_s)

Unhappy(sebastian)

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)

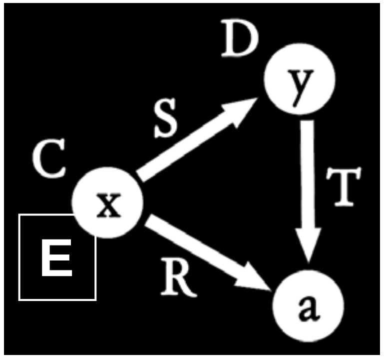
Conclusion: Unhappy(sebastian)

DL-safe variables

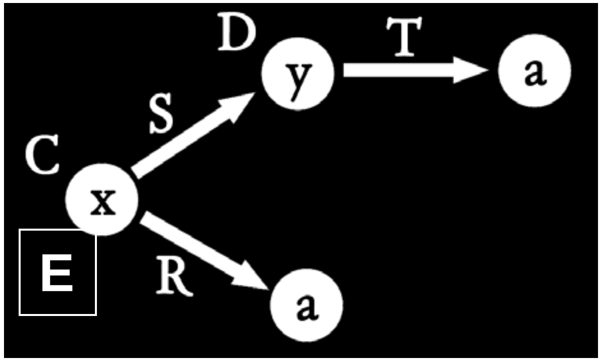
- **A generalisation of DL-safety.**
- **DL-safe variables are special variables which bind only to named individuals (like in DL-safe rules).**

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

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

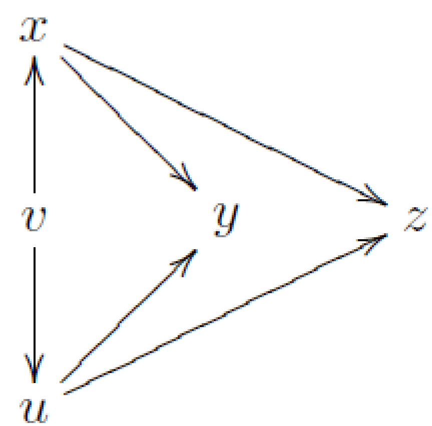
$$\text{hasReviewAssignment}(v,x) \wedge \text{hasAuthor}(x,y) \wedge \text{atVenue}(x,z)$$
$$\wedge\ \text{hasSubmittedPaper}(v,u) \wedge \text{hasAuthor}(u,y) \wedge \text{atVenue}(u,z)$$
$$\rightarrow \text{hasConflictingAssignedPaper}(v,x)$$

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

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

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

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

Contents

1. Initial examples
2. Rules expressible in description logics
3. Towards overcoming the regularity issue
4. Extending description logics with rules through nominal schemas
5. Algorithmizations for nominal schemas
6. Conclusions

Naïve implementation – 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:**

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

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. $i = 1, \ldots, 999$, $j = 0, \ldots, 999$, $k = 4, \ldots, 1000$.

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

Further Tableaux Optimizations

- **variant of absorption [Steigmiller, Glimm, Liebig, IJCAI-13]**
- **essentially, a sort of smart rewriting as pre-processing**

Example 1 *Our running example* $\exists r.(\{x\} \sqcap \exists a.\{y\} \sqcap \exists v.\{z\}) \sqcap \exists s.(\exists a.\{y\} \sqcap \exists v.\{z\}) \sqsubseteq \exists c.\{x\}$ *can be almost completely absorbed into the following axioms:*

$$
\begin{array}{lll}
O \sqsubseteq {\downarrow}x.T_x & T_z \sqsubseteq \forall v^-.T_2 & (T_1 \sqcap T_2) \sqsubseteq T_3 \\
O \sqsubseteq {\downarrow}y.T_y & T_3 \sqsubseteq \forall s^-.T_4 & (T_3 \sqcap T_x) \sqsubseteq T_5 \\
O \sqsubseteq {\downarrow}z.T_z & T_5 \sqsubseteq \forall r^-.T_6 & (T_4 \sqcap T_6) \sqsubseteq T_7. \\
T_y \sqsubseteq \forall a^-.T_1 & T_7 \sqsubseteq gr(\exists c.\{x\}), &
\end{array}
$$

where T_x, T_y, T_z, $T_1, \ldots, T_7$ *are fresh atomic concepts. Only* $\exists c.\{x\}$ *cannot be absorbed and has to be grounded on demand.*

Further Tableaux Optimizations

[Steigmiller, Glimm, Liebig, IJCAI-13]

Table 2: DL-safe Rules for UOBM-Benchmarks

Name	DL-safe Rule	Matches
R1	*isFirendOf*$(?x, ?y)$, *like*$(?x, ?z)$, *like*$(?y, ?z)$ $\rightarrow$ *hasLink1*$(?x, ?y)$	4,037
R2	*isFirendOf*$(?x, ?y)$, *takesCourse*$(?x, ?z)$, *takesCourse*$(?y, ?z)$ $\rightarrow$ *hasLink2*$(?x, ?y)$	82
R3	*takesCourse*$(?x, ?z)$, *takesCourse*$(?y, ?z)$, *hasSameHomeTownWith*$(?x, ?y)$ $\rightarrow$ *hasLink3*$(?x, ?y)$	940
R4	*hasDoctoralDegreeFrom*$(?x, ?z)$, *hasMasterDegreeFrom*$(?x, ?w)$, *hasDoctoralDegreeFrom*$(?y, ?z)$, *hasMasterDegreeFrom*$(?y, ?w)$, *worksFor*$(?x, ?v)$, *worksFor*$(?y, ?v)$, $\rightarrow$ *hasLink4*$(?x, ?y)$	369
R5	*isAdvisedBy*$(?x, ?z)$, *isAdvisedBy*$(?y, ?z)$, *like*$(?x, ?w)$, *like*$(?y, ?w)$, *like*$(?z, ?w)$ $\rightarrow$ *hasLink5*$(?x, ?y)$	286

Table 3: Comparison of the increases in reasoning time of the consistency tests for $UOBM_1 \setminus D$ extended by rules in seconds

Rule	upfront grounding		direct propagation		representative propagation		HermiT	Pellet
			without BC	with BC	without BC	with BC	1.3.7	2.3.0
R1	(10.99)	mem	9.12	7.10	5.06	3.38	31.46	6.33
R2	(10.92)	4.05	3.33	2.33	2.13	2.11	4.79	7.4
R3	(13.33)	3.55	1.98	0.62	2.20	0.76	1.67	142.25
R4	(16.44)	0.30	1.08	0.09	1.06	0.07	1.42	122.85
R5	(time)	–	1.87	0.50	1.80	0.43	28.41	mem

Algorithm for ELROVn

Based on [Krötzsch, JELIA10]

Ontology	Individuals	no ns	1 ns	2 ns	3 ns	4 ns	5 ns
Rex (full ground.)	100	263	263 (321)	267 (972)	273	275	259
	1000	480	518 (1753)	537 (OOM)	538	545	552
	10000	2904	2901 (133179)	3120 (OOM)	3165	3192	3296
Spatial (full ground.)	100	22	191 (222)	201 (1163)	198	202	207
	1000	134	417 (1392)	415 (OOM)	421	431	432
	10000	1322	1792 (96437)	1817 (OOM)	1915	1888	1997
Xenopus (full ground.)	100	62	332 (383)	284 (1629)	311	288	280
	1000	193	538 (4751)	440 (OOM)	430	456	475
	10000	1771	2119 (319013)	1843 (OOM)	1886	2038	2102

Approximating OWL through ELROVn

– We rewrite mincardinality restrictions into maxcardinality restrictions or approximate using an existential.
– We rewrite universal quantification into existential quantification.
– We approximate maxcardinality restrictions using functionality.
– We approximate inverse roles and functionality using nominal schemas.
– We approximate negation using class disjointness.
– We approximate disjunction using conjunction.

- **inverses:** $\{x\} \sqcap \exists R.\{y\} \sqsubseteq \{y\} \sqcap \exists S.\{x\}$
- **functionality** $C \sqsubseteq \leq 1 R.D$ **:**

$$C \sqcap \exists R.(\{z1\} \sqcap D) \sqcap \exists R.(\{z2\} \sqcap D) \sqsubseteq \exists U.(\{z1\} \sqcap \{z2\})$$

Approximation results (using IRIS)

Ontology	HermiT	Fact++	Pellet	Ours	Ours Recall
BAMS	3	2	10	107	100%
DOLCE	1	1	4	53	100%
GALEN	4	2	17	7840	90.8%
GO	36	75	59	N/A	N/A
GardinerCorpus	14	6	17	89	92.3%
OBO	34	61	139	N/A	N/A

Contents

1. Initial examples
2. Rules expressible in description logics
3. Extending description logics with rules through nominal schemas
4. Algorithmizations for nominal schemas
5. Conclusions

Conclusions

- Many rules are already expressible in OWL.
- Nominal schemas are a mild extension, for covering many rules.
- Efficient algorithmizations are under way.

- Course by Matthias Knorr next week:
  How nominal schemas can be used to further integrate OWL and rules paradigms.

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, TU Dresden, Germany

Kunal Sengupta, Kno.e.sis Center, Wright State University

Cong Wang, Kno.e.sis Center, Wright State University

References

A tutorial:

- Adila A. Krisnadhi, Frederick Maier, Pascal Hitzler, OWL and Rules. In: A. Polleres, C. d'Amato, M. Arenas, S. Handschuh, P. Kroner, S. Ossowski, P.F. Patel-Schneider (eds.), Reasoning Web. Semantic Technologies for the Web of Data. 7th International Summer School 2011, Galway, Ireland, August 23-27, 2011, Tutorial Lectures. Lecture Notes in Computer Science Vol. 6848, Springer, Heidelberg, 2011, pp. 382-415.

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 (Second Edition). W3C Recommendation, 11 December 2012. http://www.w3.org/TR/owl2-primer/

References

- 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.
- 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.
- Markus Krötzsch. Description Logic Rules. Studies on the Semantic Web, Vol. 008, IOS Press, 2010. http://www.semantic-web-studies.net/

References

- Matthias Knorr, David Carral Martinez, Pascal Hitzler, Adila A. Krisnadhi, Frederick Maier, Cong Wang, Recent Advances in Integrating OWL and Rules (Technical Communication). In: Markus Krötzsch, Umberto Straccia (eds.), Web Reasoning and Rule Systems, 6th International Conference, RR2012, Vienna, Austria, September 10-12, 2012, Proceedings. Lecture Notes in Computer Science Vol. 7497, Springer, Heidelberg, 2012, pp. 225-228.
- Matthias Knorr, Pascal Hitzler, Frederick Maier, Reconciling OWL and Non-monotonic Rules for the Semantic Web. In: De Raedt, L., Bessiere, C., Dubois, D., Doherty, P., Frasconi, P., Heintz, F., Lucas, P. (eds.), ECAI 2012, 20th European Conference on Artificial Intelligence, 27-31 August 2012, Montpellier, France. Frontiers in Artificial Intelligence and Applications, Vol. 242, IOS Press, Amsterdam, 2012, pp. 474-479.

References

- **David Carral, Cong Wang, Pascal Hitzler, Towards an Efficient Algorithm to Reason over Description Logics extended with Nominal Schemas. In: Wolfgang Faber, Domenico Lembo (eds.), Proceedings of the 7th International Conference on Web Reasoning and Rule Systems, RR2013, Mannheim, Germany, July 27-29, 2013. Lecture Notes in Computer Science, Vol. 7994, Springer, Heidelberg, 2013, pp. 65-79.**
- **Cong Wang, David Carral and Pascal Hitzler, SROIQ Syntax Approximation by Using Nominal Schemas. In: Thomas Eiter, Birte Glimm, Yevgeny Kazakov, Markus Krötzsch, DL 2013, Informal Proceedings of the 26th International Workshop on Description Logics, Ulm, Germany, July 23-26, 2013. CEUR Workshop Proceedings Vol. 1014, 2013, pp. 988-999.**
- **Cong Wang, Pascal Hitzler, A Resolution Procedure for Description Logics with Nominal Schemas. In: H. Takeda and Y. Giu and R. Mizoguchi and Y. Kitamura, Semantic Technology, Second Joint International Conference, JIST 2012, Nara, Japan, December 2-4, 2012, Proceedings. Lecture Notes in Computer Science Vol. 7774, Springer, Heidelberg, 2013, pp. 1-16.**

References

- **Adila Krisnadhi, Pascal Hitzler, A Tableau Algorithm for Description Logics with Nominal Schemas. In: Markus Krötzsch, Umberto Straccia (eds.), Web Reasoning and Rule Systems, 6th International Conference, RR2012, Vienna, Austria, September 10-12, 2012, Proceedings. Lecture Notes in Computer Science Vol. 7497, Springer, Heidelberg, 2012, pp. 234-237.**
- **Andreas Steigmiller , Birte Glimm, Thorsten Liebig, Nominal Schema Absorption. In: Proceedings of the 23rd International Joint Conference on Artificial Intelligence (IJCAI 2013), AAAI Press/The MIT Press, 2013**
- **Markus Krötzsch. Efficient Inferencing for OWL EL. In Tomi Janhunen, Ilkka Niemelä, eds.: Proceedings of the 12th European Conference on Logics in Artificial Intelligence, pp. 234–246. Springer 2010.**

References

- Benjamin N. Grosof, Ian Horrocks, Raphael Volz, Stefan Decker: Description logic programs: combining logic programs with description logic. WWW 2003: 48-57
- Ian Horrocks, Peter F. Patel-Schneider, Harold Boley, Said Tabet, Benjamin Grosof, Mike Dean, SWRL: A Semantic Web Rule Language Combining OWL and RuleML. W3C Member Submission 21 May 2004. http://www.w3.org/Submission/SWRL/
- Boris Motik, Ulrike Sattler, Rudi Studer: Query Answering for OWL-DL with rules. J. Web Sem. 3(1): 41-60 (2005)
- Boris Motik, Riccardo Rosati: Reconciling description logics and rules. J. ACM 57(5) (2010)
- Michael Kifer, Harold Boley, RIF Overview (Second Edition). W3C Working Group Note 5 February 2013. http://www.w3.org/TR/rif-overview/
- Markus Krötzsch, Sebastian Rudolph, Pascal Hitzler: ELP: Tractable Rules for OWL 2. International Semantic Web Conference 2008: 649-664