TECHNISCHE UNIVERSITÄT DRESDEN

FOUNDATIONS OF SEMANTIC WEB TECHNOLOGIES

OWL and Rules

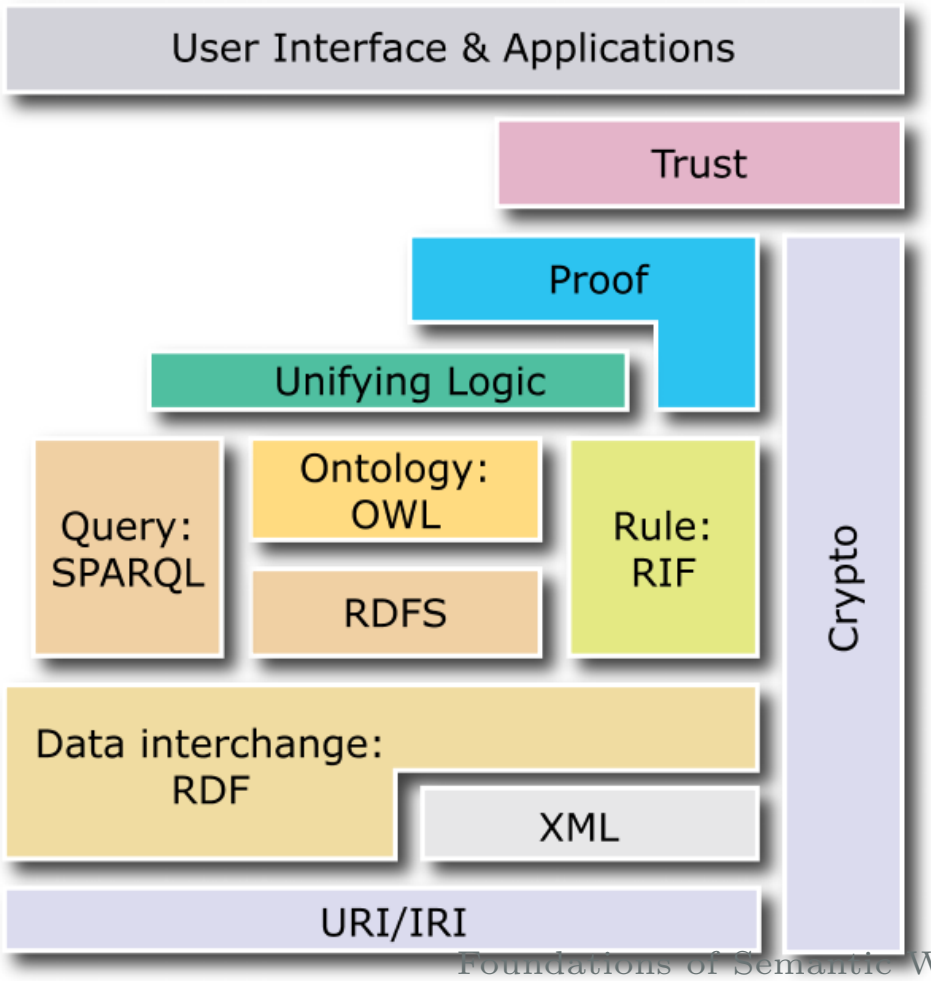
Sebastian Rudolph

Dresden, 16 July 2013

DRESDEN concept
Exzellenz aus Wissenschaft und Kultur

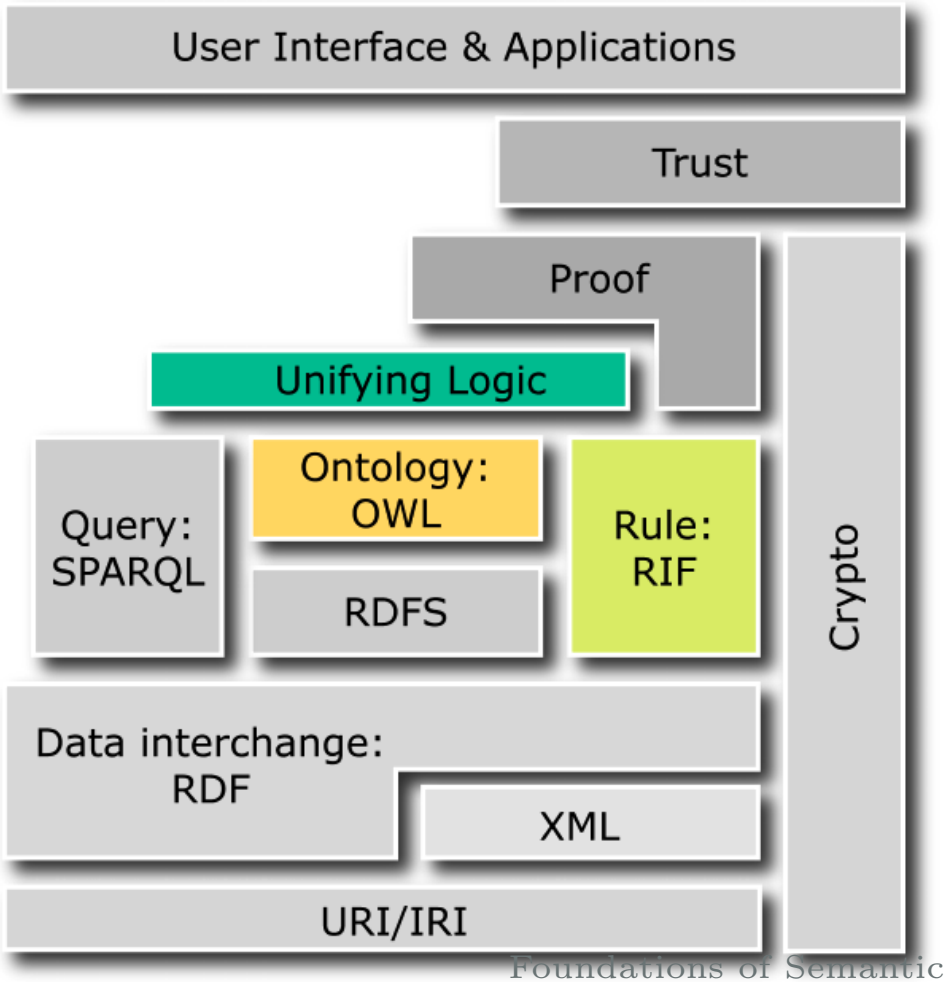
Introduction

Semantic Web Stack / Layer Cake



- Each layer builds on the layers below
- Standardization in progress and driven by W3C
- Hypertext Web technologies and some Semantic Web technologies already standardized

Two Different Paradigms



- Ontologies: OWL
- Rules: RIF, SWRL

Investigation towards a Unifying Logic

OWL 2 DL ($\mathcal{SROIQ}$(D)+...)

Basic Language Constructs

- Classes (concepts) – unary predicates

  Person, Woman, Mother, Uncle

- Properties (roles) – binary predicates

  hasChild, hasParent, hasWife

- Individuals – constants

  Mary, John, Bill

OWL 2 DL ($\mathcal{SROIQ}$(D)+...)

Axioms

- Assertions of named individuals to (complex) classes and properties

  $$\text{Woman}(\text{Mary}) \qquad \text{hasMother}(\text{Bill},\text{Mary})$$

- (Sub)class and property hierarchies ($\sqsubseteq$)

  $$\text{Woman} \sqsubseteq \text{Person} \qquad \text{hasMother} \sqsubseteq \text{hasParent}$$

- Equivalent classes ($\equiv$ – shortcut for $\sqsubseteq$ and $\sqsupseteq$)

  $$\text{Person} \equiv \text{Human}$$

OWL 2 DL ($\mathcal{SROIQ}$(D)+...)

Complex Classes

- Class intersection ($\sqcap$)

$$\text{Mother} \equiv \text{Person} \sqcap \text{Woman}$$

- Class union ($\sqcup$)

$$\text{Parent} \equiv \text{Mother} \sqcup \text{Father}$$

- Class complement ($\neg$)

$$\text{ChildlessPerson} \equiv \text{Person} \sqcap \neg\text{Parent}$$

OWL 2 DL ($\mathcal{SROIQ}$(D)+...)

Complex Classes

- `owl:Thing` ($\top$) and `owl:Nothing` ($\bot$)

$$\text{Man} \sqcap \text{Woman} \sqsubseteq \bot$$

- Existential quantification ($\exists$)

$$\text{Parent} \equiv \exists \text{hasChild}.\text{Person} \qquad \exists \text{hasWife}.\top \sqsubseteq \text{Man}$$

- Universal quantification ($\forall$)

$$\text{NoDaughters} \equiv \forall \text{hasChild}.\text{Male} \qquad \top \sqsubseteq \forall \text{hasWife}.\text{Woman}$$

OWL 2 DL ($\mathcal{SROIQ}$(D)+...)

Complex Classes

- Qualified cardinality constraints ($\leq$ and $\geq$)

  $\leq 2\text{hasChild.Parent}(\text{John})$ $\quad$ $\geq 4\text{hasChild.}\top(\text{John})$

- Nominals/enumerations of individuals (`owl:oneOf`)

  $\{\text{William}\} \equiv \{\text{Bill}\}$ $\quad$ $\text{SiblingsOfJohn} \equiv \{\text{Mary, Tom}\}$

- Self

  $\text{NarcisticPerson} \equiv \exists \text{loves.Self}$

OWL 2 DL ($\mathcal{SROIQ}$(D)+...)

Property Characteristics

- Inverse – hasChild$^-$ instead of hasParent
- Symmetric – hasSpouse
- Asymmetric – hasChild
- Disjoint – hasParent and hasChild
- Reflexive – hasRelative
- Irreflexive – hasChild
- Functional – hasSpouse
- Inverse functional – hasSpouse
- Transitive – hasDescendent

OWL 2 DL ($\mathcal{SROIQ}$(D)+...)

And also ...

- Property chains

$$\text{hasParent} \circ \text{hasParent} \sqsubseteq \text{hasGrandparent}$$

- Top property (U universal role)
- Bottom property (not part of $\mathcal{SROIQ}$(D))
- Datatypes (with facets) (D)
- Keys (not part of $\mathcal{SROIQ}$(D))

OWL 2 Profiles

fragments of OWL 2 for better computational properties

- OWL 2 EL:
  - Corresponds to $\mathcal{SROEL}(\mathrm{D})$ /$\mathcal{EL}^{++}$
  - Allows $\sqcap$, $\exists$, $\top$, $\bot$, nominals, property chains and hierarchies, and datatypes
  - Also allows: reflexive and transitive properties, keys
  - Used in large biomedicine ontologies, such as SNOMED CT, or GALEN (containing complex structural descriptions)

OWL 2 Profiles

- OWL 2 QL:
  - Corresponds to DL-Lite
  - Left of $\sqsubseteq$ only allows: $\exists$ limited to $\top$
  - Right of $\sqsubseteq$ only allows: $\sqcap$, $\neg$, $\exists$
  - Allows property inclusions but not property chains
  - Closely related to database technology, query answering can be realized by rewriting queries
- OWL 2 RL:
  - Corresponds to DLP, a rule fragment of OWL 2 DL
  - Well-suited for enriching RDF data
  - Details follow later

What we can(not) do with OWL

- Describe schema level knowledge: class hierarchy, properties about classes, relationships between classes, etc.
- Consistency and class subsumption checking
- Classifying individuals to classes
- Assert the existence of unknown individuals (i.e., those that must exist but cannot be named)
- Cannot specify arbitrary relationships between instances/individuals; due to the inherent tree structure of DLs
- Cannot express n-ary relationships between individuals with $n > 2$; DL extensions with n-ary predicates exist, but not part of OWL

Rules

- Prominent alternative to OWL modeling:
  - Rule-based expert systems
  - Logic Programming/Prolog
  - F-Logic [Kifer et al., 1995]
  - W3C Rule Interchange Format RIF (standard since 2010)
- Often argued to be more intuitive for modeling:

$$\text{worksAt(x,y)},\text{university(y)},\ \text{supervises(x,z)},\text{PhDstudent(z)} \rightarrow \text{ProfessorOf(x, z)}$$

Rules

- Rules can be divided into 3 categories:
  - First-order rules: logical implication F $\rightarrow$ G
    - Closely related to RIF-BLD (Basic Logic Dialect)
    - "Open world", declarative (first-order) semantics, monotonic
  - Logic Programming/PROLOG rules:
    - Close to first-order rules but with optional procedural aspects and possible built-ins
    - Covered by RIF-FLD (Framework for Logic Dialects)
    - "Closed world", (semi-)declarative, non-monotonic
  - Production rules:
    - IF condition THEN action
    - Roughly corresponds to RIF-PRD (Production Rule Dialect)
    - Semantics varies, sometimes defined as ad hoc computational mechanisms

First-order rules (Horn clauses)

- $\overbrace{A_1 \wedge A_2 \wedge \cdots \wedge A_k}^{\text{body}} \rightarrow \overbrace{H}^{\text{head}}$
  - Each A_i and H is a first-order atomic formula $P(t_1, \ldots, t_m)$ with P a predicate symbol with arity m
  - Each t_j is a term: a variable or an expression $f(s_1, \ldots, s_k)$ where f is a function symbol of arity k and $s_1, \ldots, s_k$ are terms
  - No quantifiers; no negation
- Datalog rules: first-order rules without function symbols
  - First used for deductive databases
  - Complexity: PTime data complexity (ExpTime combined)
  - Suitable for large datasets (and relatively small/fixed rule set)

What we can(not) do with Rules

- Specify and infer arbitrary relationships between individuals, including n-ary relationships with $n > 2$
- Many people find rules more natural for modeling
- Non-monotonic extensions are very well-studied, more than that of OWL (ASP solvers, etc.)
- Rules are usually only applied to known constants
- Cannot express the existence of unknown/unnamed individuals (unlike OWL)

Can't we bring them together?

Our Agenda . . .

This tutorial provides a condensed exposition of the recent efforts to answer the previous main question by focusing on the following issues:

- What kind of rules are readily expressible in OWL?
- What is DL-safety notion for rules? Why does it allow one to combine rules and OWL ontologies without losing decidability?
- Can we integrate DL-safe rules seamlessly within OWL framework by some small syntactic extension to OWL?
- Can we add non-monotonic flavor to such integration between DLs and rules?

Rules readily expressible in OWL

Reasoning Needs

z newsFrom rome .
rome locadedIn italy .

Reasoning Needs

z newsFrom rome .
rome locadedIn italy .

We want to conclude

z newsFrom italy .

Reasoning Needs

z newsFrom rome .
rome locadedIn italy .

We want to conclude

z newsFrom italy .

Rule:

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

Reasoning Needs

z newsFrom rome .
rome locadedIn italy .

We want to conclude

z newsFrom italy .

Rule:

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

In OWL:

$$\text{newsFrom} \circ \text{locatedIn} \sqsubseteq \text{newsFrom}$$

(using owl:propertyChainAxiom)

Translating OWL Axioms

Which OWL axioms can be encoded as rules?

Let's see some examples

Translating OWL Axioms

Which OWL axioms can be encoded as rules?

Translating OWL Axioms

Which OWL axioms can be encoded as rules?

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

Translating OWL Axioms

Which OWL axioms can be encoded as rules?

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

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

Translating OWL Axioms

Which OWL axioms can be encoded as rules?

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

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

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

Rules in OWL

Which OWL axioms can be encoded as rules?

Rules in OWL

Which OWL axioms can be encoded as rules?

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

Rules in OWL

Which OWL axioms can be encoded as rules?

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

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

Rules in OWL

Which OWL axioms can be encoded as rules?

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

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

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

Rules in OWL

Which OWL axioms can be encoded as rules?

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

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

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

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

Rules in OWL

Which OWL axioms can be encoded as rules?

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

Rules in OWL

Which OWL axioms can be encoded as rules?

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

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

Rules in OWL

Which OWL axioms can be encoded as rules?

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

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

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

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?

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?

Naive translation: e.g.,

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

This essentially results in OWL RL

Which rules can be translated into OWL axioms?

- Rolification
- Examples
- Formal definition: Rule Graphs

Rolification

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

Rolification

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

Rolification of a concept A: $\text{A} \sqsubseteq \exists \text{R}_\text{A}.\text{Self}$

Rolification

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

Rolification of a concept A: $A \sqsubseteq \exists R_A.\text{Self}$

$$\begin{aligned} \text{Elephant} &\sqsubseteq \exists R_{\text{Elephant}}.\text{Self} \\ \text{Mouse} &\sqsubseteq \exists R_{\text{Mouse}}.\text{Self} \\ R_{\text{Elephant}} \circ U \circ R_{\text{Mouse}} &\sqsubseteq \text{biggerThan} \end{aligned}$$

Rolification

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

Rolification

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

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

careful - role regularity needs to be preserved

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

Rolification

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

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

Rules in OWL 2

$$\mathrm{Man}(x) \wedge \mathrm{hasBrother}(x, y) \wedge \mathrm{hasChild}(y, z) \rightarrow \mathrm{Uncle}(x)$$

$$\mathrm{Man} \sqcap \exists \mathrm{hasBrother}.\exists \mathrm{hasChild}.\top \sqsubseteq \mathrm{Uncle}$$

$$\mathrm{NutAllergic}(x) \wedge \mathrm{NutProdcut}(y) \rightarrow \mathrm{dislikes}(x, y)$$

$$\mathrm{NutAllergic} \sqsubseteq \exists \mathrm{nutAllergic}.\mathrm{Self}$$

$$\mathrm{NutProduct} \sqsubseteq \exists \mathrm{nutProduct}.\mathrm{Self}$$

$$\mathrm{nutAllergic} \circ \mathrm{U} \circ \mathrm{nutProduct} \sqsubseteq \mathrm{dislikes}$$

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

$$\mathrm{Dish} \sqsubseteq \exists \mathrm{dish}.\mathrm{Self}$$

$$\mathrm{dislikes} \circ \mathrm{contains}^{-} \circ \mathrm{dish} \sqsubseteq \mathrm{dislikes}$$

So how can we pinpoint this?

- Tree-shaped bodies (variables)
- First argument of the conclusion is the root

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

$$C \sqcap \exists R.\{a\} \sqcap \exists S.(D \sqcap \exists T.\{a\}) \sqsubseteq E$$

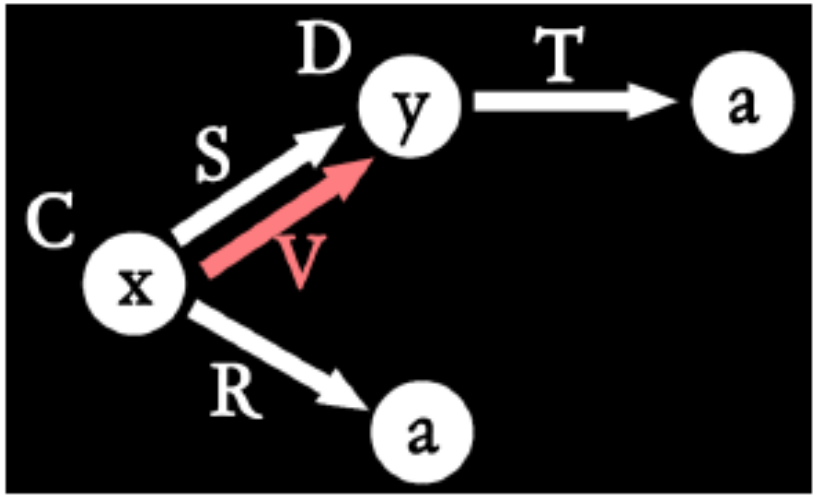
So how can we pinpoint this?

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

So how can we pinpoint this?

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

$$C \sqcap \exists R.\{a\} \sqsubseteq \exists R_1.\mathrm{Self}$$
$$D \sqcap \exists T.\{a\} \sqsubseteq \exists R_2.\mathrm{Self}$$
$$R_1 \circ S \circ R_2 \sqsubseteq V$$



So how can we pinpoint this?

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

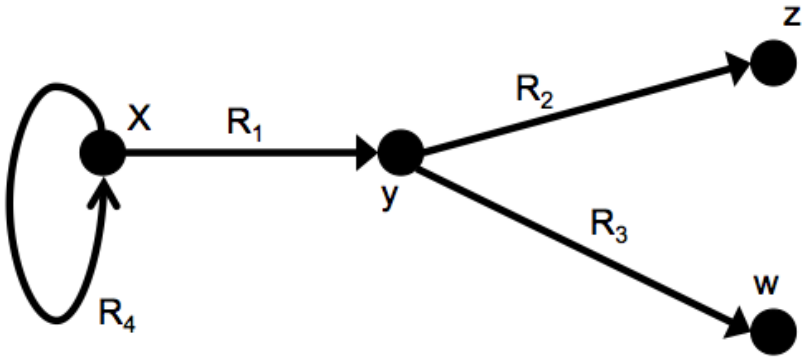
Graph analysis: determine whether a rule is expressible within a given profile
Automatic Transformation

DLs Rules: $\mathcal{EL}^{++}$

$$\mathrm{R}_1(\mathrm{x},\mathrm{y}) \wedge \mathrm{C}_1(\mathrm{y}) \wedge \mathrm{R}_2(\mathrm{y},\mathrm{w}) \wedge \mathrm{R}_3(\mathrm{y},\mathrm{z}) \wedge \mathrm{C}_2(\mathrm{z}) \wedge \mathrm{R}_4(\mathrm{x},\mathrm{x}) \rightarrow \mathrm{C}_3(\mathrm{x})$$

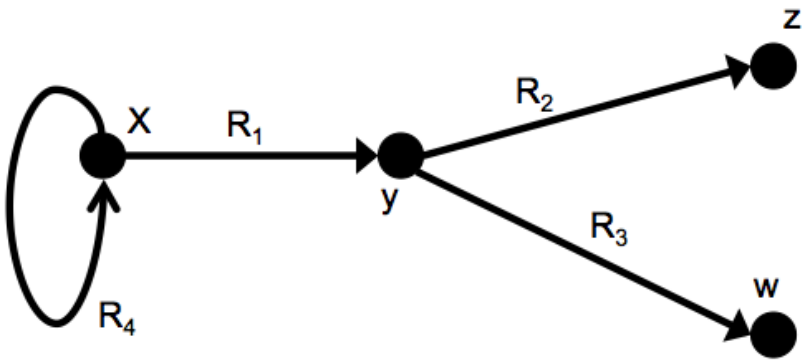
DLs Rules: $\mathcal{EL}^{++}$

$$\mathrm{R}_1(\mathrm{x},\mathrm{y}) \wedge \mathrm{C}_1(\mathrm{y}) \wedge \mathrm{R}_2(\mathrm{y},\mathrm{w}) \wedge \mathrm{R}_3(\mathrm{y},\mathrm{z}) \wedge \mathrm{C}_2(\mathrm{z}) \wedge \mathrm{R}_4(\mathrm{x},\mathrm{x}) \rightarrow \mathrm{C}_3(\mathrm{x})$$



DLs Rules: $\mathcal{EL}^{++}$

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



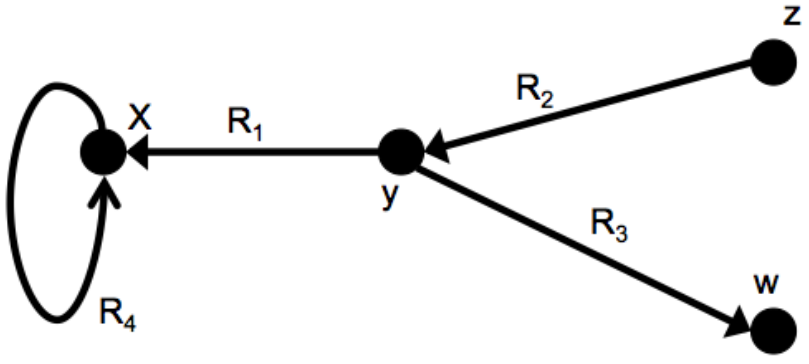
$$\mathrm{\exists R_1.(C_1 \sqcap \exists R_2.\top \sqcap \exists R_3.C_2) \sqcap \exists R_4.Self \sqsubseteq C_3}$$

DLs Rules: $\mathcal{SROIQ}$

$$\mathrm{R}_1(\mathrm{y},\mathrm{x}) \wedge \mathrm{C}_1(\mathrm{y}) \wedge \mathrm{R}_2(\mathrm{w},\mathrm{y}) \wedge \mathrm{R}_3(\mathrm{y},\mathrm{z}) \wedge \mathrm{C}_2(\mathrm{z}) \wedge \mathrm{R}_4(\mathrm{x},\mathrm{x}) \rightarrow \mathrm{C}_3(\mathrm{x})$$

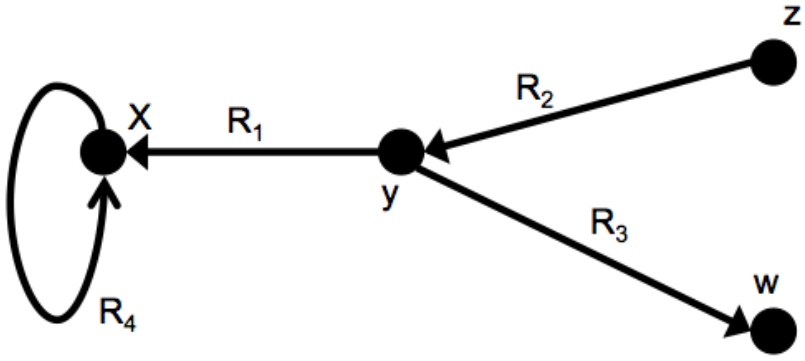
DLs Rules: $\mathcal{SROIQ}$

$$\mathrm{R}_1(\mathrm{y},\mathrm{x}) \wedge \mathrm{C}_1(\mathrm{y}) \wedge \mathrm{R}_2(\mathrm{w},\mathrm{y}) \wedge \mathrm{R}_3(\mathrm{y},\mathrm{z}) \wedge \mathrm{C}_2(\mathrm{z}) \wedge \mathrm{R}_4(\mathrm{x},\mathrm{x}) \rightarrow \mathrm{C}_3(\mathrm{x})$$



DLs Rules: $\mathcal{SROIQ}$

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



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

Nominal Schemas

What we learned about rules expressible in OWL

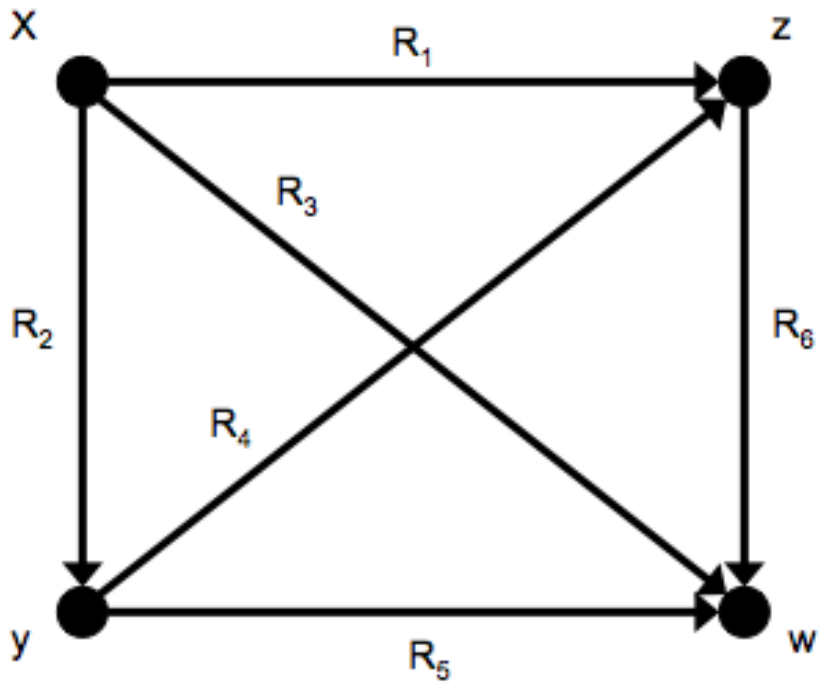
- Rules with tree-shaped body can be expressed in DL,
- role conjunction allows DLs to express some rules with non-tree-shaped body, but
- many rules are not covered.

Clique of 4

$$\mathrm{R}_1(\mathrm{x},\mathrm{y}) \wedge \mathrm{R}_2(\mathrm{x},\mathrm{z}) \wedge \mathrm{R}_3(\mathrm{x},\mathrm{w}) \wedge \mathrm{R}_4(\mathrm{y},\mathrm{z}) \wedge \mathrm{R}_5(\mathrm{y},\mathrm{w}) \wedge \mathrm{R}_6(\mathrm{w},\mathrm{z}) \rightarrow \mathrm{C}(\mathrm{x})$$

Clique of 4

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



Nominal Schemas

A Better Uncle for OWL
[Krötzsch et al; WWW 2011]

$$\text{hasParent}(x, y) \wedge \text{married}(y, z) \wedge \text{hasParent}(x, z) \rightarrow C(x)$$

$$\exists \text{hasParent}.\exists \text{married}.\{z\} \sqcap \exists \text{hasParent}.\{z\} \sqsubseteq C$$

Nominal Schemas

A Better Uncle for OWL
[Krötzsch et al; WWW 2011]

$\text{hasParent}(x, y) \wedge \text{married}(y, z) \wedge \text{hasParent}(x, z) \rightarrow C(x)$

$\exists \text{hasParent}.\exists \text{married}.\{z\} \sqcap \exists \text{hasParent}.\{z\} \sqsubseteq C$

{z} only binds to known/named individuals!

Covers DL-safe datalog (arbitrary arity of predicates)

Complex Rules to OWL

Theorem

Any rule R containing m different free variables, where $m > 3$, can be directly expressed in DL using n nominal schemas s.t. $n \leq m - 2$.

DL-safe Rules

- How about simply adding rules “as-is” to the ontology?

DL-safe Rules

- How about simply adding rules "as-is" to the ontology?
- Problem: although the DL-part and rule-part are both decidable, the combination is undecidable!

DL-safe Rules

- How about simply adding rules "as-is" to the ontology?
- Problem: although the DL-part and rule-part are both decidable, the combination is undecidable!
- Work around: weaken the rule semantics?

DL-safety

- Decidability guaranteed if rules only operate on named individuals
  - Named individuals are finite.

DL-safety

- Decidability guaranteed if rules only operate on named individuals
  - Named individuals are finite.
- DL-safety: variables in the rules refer to only named individuals in the OWL ontology.

DL-safety

- Decidability guaranteed if rules only operate on named individuals
  - Named individuals are finite.
- DL-safety: variables in the rules refer to only named individuals in the OWL ontology.
- In the rule:

$$\mathrm{R}(\mathrm{u},\mathrm{x}) \wedge \mathrm{A}(\mathrm{x}) \wedge \mathrm{S}(\mathrm{u},\mathrm{y}) \wedge \mathrm{T}(\mathrm{x},\mathrm{z}) \wedge \mathrm{T}(\mathrm{y},\mathrm{z}) \wedge \mathrm{R}(\mathrm{u},\mathrm{z}) \wedge \mathrm{S}(\mathrm{x},\mathrm{y}) \rightarrow \mathrm{B}(\mathrm{u})$$

the variables $\mathrm{u},\mathrm{x},\mathrm{y},\mathrm{z}$ refer only to named individuals.

DL-safety

- Decidability guaranteed if rules only operate on named individuals
  - Named individuals are finite.
- DL-safety: variables in the rules refer to only named individuals in the OWL ontology.
- In the rule:

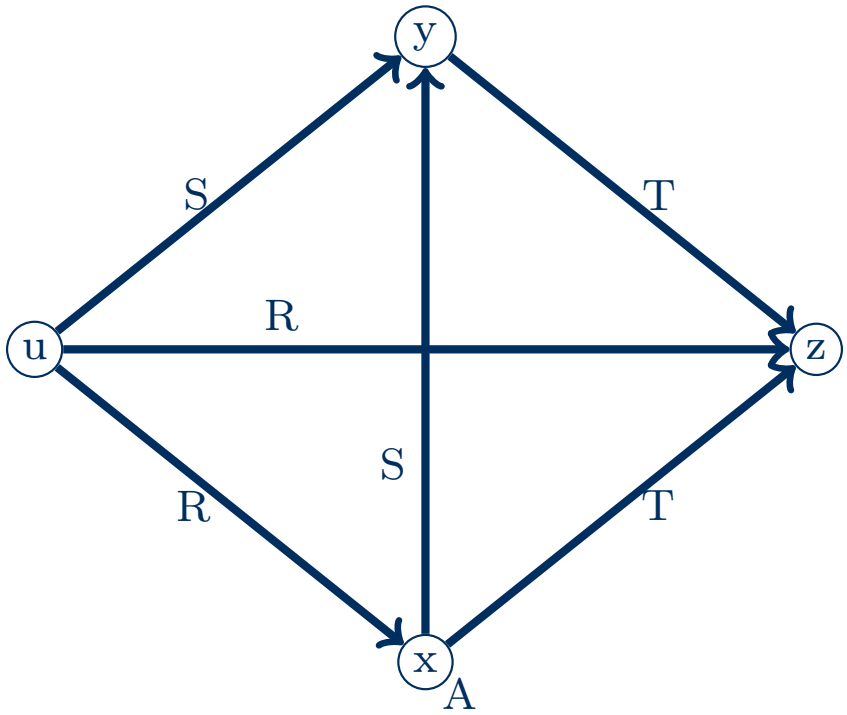
$$\mathrm{R}(\mathrm{u},\mathrm{x}) \wedge \mathrm{A}(\mathrm{x}) \wedge \mathrm{S}(\mathrm{u},\mathrm{y}) \wedge \mathrm{T}(\mathrm{x},\mathrm{z}) \wedge \mathrm{T}(\mathrm{y},\mathrm{z}) \wedge \mathrm{R}(\mathrm{u},\mathrm{z}) \wedge \mathrm{S}(\mathrm{x},\mathrm{y}) \rightarrow \mathrm{B}(\mathrm{u})$$

  the variables $\mathrm{u}, \mathrm{x}, \mathrm{y}, \mathrm{z}$ refer only to named individuals.
- Approach taken by DL-safe SWRL.

Revisiting DL-safety: can it be relaxed?

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

Rule body forms complicated graph:



Revisiting DL-safety: can it be relaxed?

$$\mathrm{R}(\mathrm{u},\mathrm{x}) \wedge \mathrm{A}(\mathrm{x}) \wedge \mathrm{S}(\mathrm{u},\mathrm{y}) \wedge \mathrm{T}(\mathrm{x},\mathrm{z}) \wedge \mathrm{T}(\mathrm{y},\mathrm{z}) \wedge \mathrm{R}(\mathrm{u},\mathrm{z}) \wedge \mathrm{S}(\mathrm{x},\mathrm{y}) \rightarrow \mathrm{B}(\mathrm{u})$$

If y and z refer to named individuals, say a, b, it represents:

$$\mathrm{R}(\mathrm{u},\mathrm{x}) \wedge \mathrm{A}(\mathrm{x}) \wedge \mathrm{S}(\mathrm{u},\mathrm{a}) \wedge \mathrm{T}(\mathrm{x},\mathrm{a}) \wedge \mathrm{T}(\mathrm{a},\mathrm{a}) \wedge \mathrm{R}(\mathrm{u},\mathrm{a}) \wedge \mathrm{S}(\mathrm{x},\mathrm{a}) \rightarrow \mathrm{B}(\mathrm{u})$$
$$\mathrm{R}(\mathrm{u},\mathrm{x}) \wedge \mathrm{A}(\mathrm{x}) \wedge \mathrm{S}(\mathrm{u},\mathrm{a}) \wedge \mathrm{T}(\mathrm{x},\mathrm{b}) \wedge \mathrm{T}(\mathrm{a},\mathrm{b}) \wedge \mathrm{R}(\mathrm{u},\mathrm{b}) \wedge \mathrm{S}(\mathrm{x},\mathrm{a}) \rightarrow \mathrm{B}(\mathrm{u})$$
$$\mathrm{R}(\mathrm{u},\mathrm{x}) \wedge \mathrm{A}(\mathrm{x}) \wedge \mathrm{S}(\mathrm{u},\mathrm{b}) \wedge \mathrm{T}(\mathrm{x},\mathrm{a}) \wedge \mathrm{T}(\mathrm{b},\mathrm{a}) \wedge \mathrm{R}(\mathrm{u},\mathrm{a}) \wedge \mathrm{S}(\mathrm{x},\mathrm{b}) \rightarrow \mathrm{B}(\mathrm{u})$$
$$\mathrm{R}(\mathrm{u},\mathrm{x}) \wedge \mathrm{A}(\mathrm{x}) \wedge \mathrm{S}(\mathrm{u},\mathrm{b}) \wedge \mathrm{T}(\mathrm{x},\mathrm{b}) \wedge \mathrm{T}(\mathrm{b},\mathrm{b}) \wedge \mathrm{R}(\mathrm{u},\mathrm{b}) \wedge \mathrm{S}(\mathrm{x},\mathrm{b}) \rightarrow \mathrm{B}(\mathrm{u})$$

Revisiting DL-safety: can it be relaxed?

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

$$R(u,x) \wedge A(x) \wedge S(u,a) \wedge T(x,a) \wedge T(a,a) \wedge R(u,a) \wedge S(x,a) \rightarrow B(u)$$
$$R(u,x) \wedge A(x) \wedge S(u,a) \wedge T(x,b) \wedge T(a,b) \wedge R(u,b) \wedge S(x,a) \rightarrow B(u)$$
$$R(u,x) \wedge A(x) \wedge S(u,b) \wedge T(x,a) \wedge T(b,a) \wedge R(u,a) \wedge S(x,b) \rightarrow B(u)$$
$$R(u,x) \wedge A(x) \wedge S(u,b) \wedge T(x,b) \wedge T(b,b) \wedge R(u,b) \wedge S(x,b) \rightarrow B(u)$$

- Only y and z need to be DL-safe.
- Expressible in OWL EL.

Nominal schemas: DL-safety only to some variables

- In the rule

$$\mathrm{R}(\mathrm{u},\mathrm{x}) \wedge \mathrm{A}(\mathrm{x}) \wedge \mathrm{S}(\mathrm{u},\mathrm{y}) \wedge \mathrm{T}(\mathrm{x},\mathrm{z}) \wedge \mathrm{T}(\mathrm{y},\mathrm{z}) \wedge \mathrm{R}(\mathrm{u},\mathrm{z}) \wedge \mathrm{S}(\mathrm{x},\mathrm{y}) \rightarrow \mathrm{B}(\mathrm{u})$$

  only y and z need to be DL-safe to be expressible in OWL EL.
- Expressing it in OWL EL need multiple axioms:

$$\exists \mathrm{R}.\{\mathrm{a}\} \sqcap \exists \mathrm{R}.(\mathrm{A} \sqcap \exists \mathrm{S}.\{\mathrm{a}\} \sqcap \exists \mathrm{T}.\{\mathrm{a}\}) \sqcap \exists \mathrm{S}.(\{\mathrm{a}\} \sqcap \exists \mathrm{T}.\{\mathrm{a}\}) \sqsubseteq \mathrm{B}$$
$$\exists \mathrm{R}.\{\mathrm{b}\} \sqcap \exists \mathrm{R}.(\mathrm{A} \sqcap \exists \mathrm{S}.\{\mathrm{a}\} \sqcap \exists \mathrm{T}.\{\mathrm{b}\}) \sqcap \exists \mathrm{S}.(\{\mathrm{a}\} \sqcap \exists \mathrm{T}.\{\mathrm{b}\}) \sqsubseteq \mathrm{B}$$
$$\exists \mathrm{R}.\{\mathrm{a}\} \sqcap \exists \mathrm{R}.(\mathrm{A} \sqcap \exists \mathrm{S}.\{\mathrm{b}\} \sqcap \exists \mathrm{T}.\{\mathrm{a}\}) \sqcap \exists \mathrm{S}.(\{\mathrm{b}\} \sqcap \exists \mathrm{T}.\{\mathrm{a}\}) \sqsubseteq \mathrm{B}$$
$$\exists \mathrm{R}.\{\mathrm{b}\} \sqcap \exists \mathrm{R}.(\mathrm{A} \sqcap \exists \mathrm{S}.\{\mathrm{b}\} \sqcap \exists \mathrm{T}.\{\mathrm{b}\}) \sqcap \exists \mathrm{S}.(\{\mathrm{b}\} \sqcap \exists \mathrm{T}.\{\mathrm{b}\}) \sqsubseteq \mathrm{B}$$

- With nominal schemas, the above 4 axioms can be condensed:
  $\exists \mathrm{R}.\{\mathrm{z}\} \sqcap \exists \mathrm{R}.(\mathrm{A} \sqcap \exists \mathrm{S}.\{\mathrm{y}\} \sqcap \exists \mathrm{T}.\{\mathrm{z}\}) \sqcap \exists \mathrm{S}.(\{\mathrm{y}\} \sqcap \exists \mathrm{T}.\{\mathrm{z}\}) \sqsubseteq \mathrm{B}$

Nominal schemas: syntax and semantics

- Nominal schemas: a "variable nominal" construct in the form of {x} where x is a variable.

Nominal schemas: syntax and semantics

- Nominal schemas: a "variable nominal" construct in the form of {x} where x is a variable.
- Semantically, each occurrence of a nominal schema represents all possible nominals in the ontology.

Nominal schemas: syntax and semantics

- Nominal schemas: a "variable nominal" construct in the form of {x} where x is a variable.
- Semantically, each occurrence of a nominal schema represents all possible nominals in the ontology.
- If an axiom α contains n different nominal schemas (each may occur more than once) while the ontology has m different named individuals, then α represents m^n different axioms, each is obtained by substituting nominal schemas with named individuals.

Nominal schemas: syntax and semantics

- Nominal schemas: a "variable nominal" construct in the form of {x} where x is a variable.
- Semantically, each occurrence of a nominal schema represents all possible nominals in the ontology.
- If an axiom α contains n different nominal schemas (each may occur more than once) while the ontology has m different named individuals, then α represents m^n different axioms, each is obtained by substituting nominal schemas with named individuals.
- All those m^n axioms are called groundings of α.

What can we express with nominal schemas?

- Let $\mathcal{SROEL\!V}(\times, \sqcap) = \mathcal{SROEL}(\times, \sqcap)$ + nominal schemas.

What can we express with nominal schemas?

- Let $\mathcal{SROEL}\mathcal{V}(\times, \sqcap) = \mathcal{SROEL}(\times, \sqcap)$ + nominal schemas.
- $\mathcal{SROEL}\mathcal{V}(\sqcap, \times)$ covers:
  - DL-safe Datalog rules with predicates of arbitrary arity is also in $\mathcal{SROEL}\mathcal{V}$.
  - OWL 2 EL without datatypes
  - DL-safe OWL 2 RL without datatypes — but, preserves only ABox entailments (the main inference task for OWL 2 RL)
  - most of OWL 2 QL

Complexity bounds

- Reasoning for $\mathcal{SROIQV} = \mathcal{SROIQ}$ + nominal schemas, is theoretically no worse than $\mathcal{SROIQ}$ [Krötzsch et al., WWW11]

Complexity bounds

- Reasoning for $\mathcal{SROIQV} = \mathcal{SROIQ}$ + nominal schemas, is theoretically no worse than $\mathcal{SROIQ}$ [Krötzsch et al., WWW11]
- Reasoning for $\mathcal{SROELV}$ is:
  - still polynomial (like $\mathcal{SROEL}$) if the number of occurrences of different nominal schemas in an axiom is bounded by a fixed constant;
  - in general, it is exponential (c.f., combined complexity of Datalog is ExpTime)

An Efficient Implementation for Nominal Schemas?

Naive grounding

- Naive reasoning directly from the semantics:

Naive grounding

- Naive reasoning directly from the semantics:
  - Ground each axiom containing nominal schemas into exponentially many axioms without nominal schemas.
  - The resulting ontology is in standard DL or OWL and equivalent to the original one.
  - Use any existing reasoning algorithm for the corresponding standard DL on the resulting ontology.

Naive grounding

- Naive reasoning directly from the semantics:
  - Ground each axiom containing nominal schemas into exponentially many axioms without nominal schemas.
  - The resulting ontology is in standard DL or OWL and equivalent to the original one.
  - Use any existing reasoning algorithm for the corresponding standard DL on the resulting ontology.
- Practically inefficient.

Naive grounding example

$$\exists \text{hParent}.\exists \text{married}.\{\text{z}\} \sqcap \exists \text{hParent}.\{\text{z}\} \sqsubseteq \text{C}$$

Naive grounding example

$$\exists \text{hParent}.\exists \text{married}.\{\text{z}\} \sqcap \exists \text{hParent}.\{\text{z}\} \sqsubseteq \text{C}$$

Full grounding:

$$\exists \text{hParent}.\exists \text{married}.\{\text{a}\} \sqcap \exists \text{hParent}.\{\text{a}\} \sqsubseteq \text{C}$$
$$\exists \text{hParent}.\exists \text{married}.\{\text{b}\} \sqcap \exists \text{hParent}.\{\text{b}\} \sqsubseteq \text{C}$$
$$\exists \text{hParent}.\exists \text{married}.\{\text{c}\} \sqcap \exists \text{hParent}.\{\text{c}\} \sqsubseteq \text{C}$$

where a, b, and c are the only individuals in the knowledge base

Naive grounding example

$$\exists R_1.(\exists R_4.\{z\} \sqcap \exists R_5.(\{w\} \sqcap \exists R_6.\{z\})) \sqcap \exists R_2.\{z\} \sqcap \exists R_3.\{w\} \sqsubseteq C$$

Naive grounding example

$$\exists R_1.(\exists R_4.\{z\} \sqcap \exists R_5.(\{w\} \sqcap \exists R_6.\{z\})) \sqcap \exists R_2.\{z\} \sqcap \exists R_3.\{w\} \sqsubseteq C$$

Full grounding:

$$\exists R_1.(\exists R_4.\{a\} \sqcap \exists R_5.(\{a\} \sqcap \exists R_6.\{a\})) \sqcap \exists R_2.\{a\} \sqcap \exists R_3.\{a\} \sqsubseteq C$$
$$\exists R_1.(\exists R_4.\{a\} \sqcap \exists R_5.(\{b\} \sqcap \exists R_6.\{a\})) \sqcap \exists R_2.\{a\} \sqcap \exists R_3.\{b\} \sqsubseteq C$$
$$\exists R_1.(\exists R_4.\{a\} \sqcap \exists R_5.(\{c\} \sqcap \exists R_6.\{a\})) \sqcap \exists R_2.\{a\} \sqcap \exists R_3.\{c\} \sqsubseteq C$$
$$\exists R_1.(\exists R_4.\{a\} \sqcap \exists R_5.(\{b\} \sqcap \exists R_6.\{a\})) \sqcap \exists R_2.\{a\} \sqcap \exists R_3.\{b\} \sqsubseteq C$$
$$\exists R_1.(\exists R_4.\{b\} \sqcap \exists R_5.(\{c\} \sqcap \exists R_6.\{b\})) \sqcap \exists R_2.\{a\} \sqcap \exists R_3.\{c\} \sqsubseteq C$$
$$\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots$$

where a, b, and c are the only individuals in the knowledge base

Defining Optimizations

- Delayed grounding: [Adila et al. at RR 2012]
- Ordered Resolution: [Cong et al. at JIST 2012]

References

Rules expressible in OWL:

- Description logic programs: combining logic programs with description logic. Benjamin Grosof, Ian Horrocks, Raphael Volz, and Stefan Decker, WWW 2003, 48–57.
- Description logic rules. Markus Krötzsch, Sebastian Rudolph, and Pascal Hitzler, ECAI 2008, 80–84.
- ELP: tractable rules for OWL 2. Markus Krötzsch, Sebastian Rudolph, and Pascal Hitzler, ISWC 2008, 649–664.
- Cheap boolean role constructors for description logics. Sebastian Rudolph, Markus Krötzsch, and Pascal Hitzler, JELIA 2008, 362–374.
- Description logic rules. Markus Krötzsch, Studies on the Semantic Web, Vol. 08, 2010.

References

Rules expressible in OWL and with nominal schemas:

- A better uncle for OWL: nominal schemas for integrating rules and ontologies. Markus Krötzsch, Frederick Maier, Adila A. Krisnadhi, and Pascal Hitzler, WWW 2011, 645–654.
- Extending description logic rules. David Carral Martínez and Pascal Hitzler, ESWC 2012, 345–359.
- A tableau algorithm for description logics with nominal schemas. Adila Krisnadhi and Pascal Hitzler, RR2012, 234-237.
- A resolution procedure for description logics with nominal schemas. Cong Wang and Pascal Hitzler, JIST 2012, 1-16.

References

Nonmonotonic Extensions:

- Description logics of minimal knowledge and negation as failure. Francesco M. Donini, Daniele Nardi, and Riccardo Rosati, ACM Transactions on Computational Logic, 3(2), 2002, 227–252.
- Reconciling Description Logics and Rules. Boris Motik and Riccardo Rosati, Journal of the ACM, 57(5), 2010, 1–62.
- Local closed world reasoning with description logics under the well-founded semantics. Matthias Knorr, José J. Alferes, and Pascal Hitzler, Artificial Intelligence, 175(9-10), 2011, 1528–1544.
- Reconciling OWL and non-monotonic rules for the Semantic Web. Matthias Knorr, Pascal Hitzler, and Frederick Maier, ECAI 2012, 474–479.