Karlsruhe Institute of Technology

Semantic Web Modelling Languages (Part 2)

Tutorial at IJCAI-09
July 13, 2009

Forschungszentrum Karlsruhe in der Helmholtz-Gemeinschaft
Universität Karlsruhe (TH) Research University • founded 1825

... actually moving to Wright State University, Dayton, OH, Sept. 2009

Pascal Hitzler 

Markus Krötzsch 

Sebastian Rudolph 

AIFB, Universität Karlsruhe (TH)
Germany

http://www.pascal-hitzler.de
http://korrekt.org
http://www.sebastian-rudolph.de

Full set of slides available from

http://semantic-web-grundlagen.de/wiki/IJCAI-09_Tutorial

OWL – Overview

- Web Ontology Language
  - W3C Recommendation for the Semantic Web, 2004
  - OWL 2 (revised W3C Recommendation) forthcoming in 2009 – we already present this here

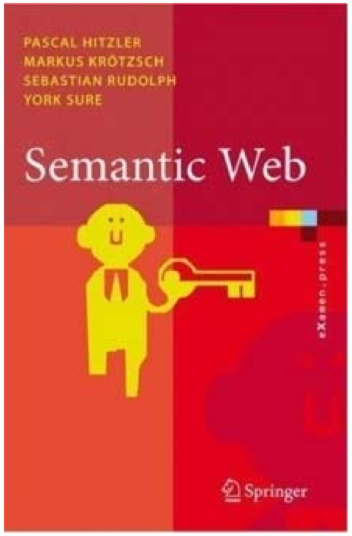
- Semantic Web KR language based on description logics (DLs)
  - OWL DL is essentially DL SROIQ(D)
  - KR for web resources, using URIs.
  - Using web-enabled syntaxes, e.g. based on XML or RDF. We present
    - DL syntax (used in research – not part of the W3C recommendation)
    - (some) RDF Turtle syntax

References

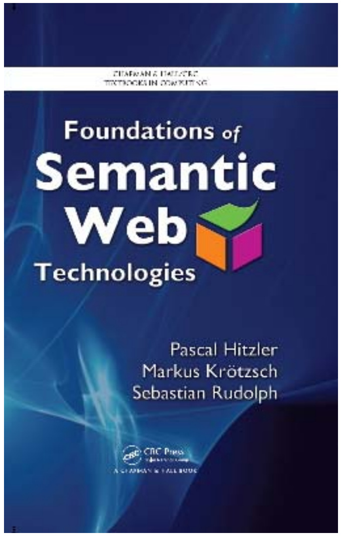
- W3C OWL Working Group, OWL 2 Web Ontology Language: Document Overview. http://www.w3.org/TR/owl2-overview/
- Pascal Hitzler, Markus Krötzsch, Bijan Parsia, Peter Patel-Schneider, Sebastian Rudolph, OWL 2 Web Ontology Language: Primer. http://www.w3.org/TR/owl2-primer/
- Franz Baader, Diego Calvanese, Deborah L. McGuinness, Daniele Nardi, Peter F. Patel-Schneider, The Description Logic Handbook: Theory, Implementation, and Applications. Cambridge University Press, 2nd edition, 2007.

References – Textbooks

- **Pascal Hitzler, Markus Krötzsch, Sebastian Rudolph, York Sure, Semantic Web – Grundlagen. Springer, 2008. http://www.semantic-web-grundlagen.de/ (In German)**



- **Pascal Hitzler, Markus Krötzsch, Sebastian Rudolph, Foundations of Semantic Web Technologies. Chapman & Hall/CRC, 2009. http://www.semantic-web-book.org/wiki/FOST (Ask for a flyer from us)**



Contents

Contents

Rationale behind OWL

- Open World Assumption
- Favourable trade-off between expressivity and scalability
- Integrates with RDFS
- Purely declarative semantics

Features:

- Fragment of first-order predicate logic (FOL)
- Decidable
- Known complexity classes (N2ExpTime for OWL 2 DL)
- Reasonably efficient for real KBs

OWL Building Blocks

- individuals (written as URIs)
  - also: constants (FOL), ressources (RDF)
  - http://example.org/sebastianRudolph
  - http://www.semantic-web-book.org
  - we write these lowercase and abbreviated, e.g. "sebastianRudolph"
- classes (also written as URIs!)
  - also: concepts, unary predicates (FOL)
  - we write these uppercase, e.g. "Father"
- properties (also written as URIs!)
  - also: roles (DL), binary predicates (FOL)
  - we write these lowercase, e.g. "hasDaughter"

DL syntax

FOL syntax

ABox statements

- Person(mary)
- Person(mary)

- Woman $\sqsubseteq$ Person
  - Person $\equiv$ HumanBeing
- $\forall x$ (Woman(x) $\rightarrow$ Person(x))

- hasWife(john,mary)
- hasWife(john,mary)

- hasWife $\sqsubseteq$ hasSpouse
  - hasSpouse $\equiv$ marriedWith
- $\forall x \forall y$ (hasWife(x,y)$\rightarrow$ hasSpouse(x,y))

TBox statements

DL syntax	RDFS syntax
Person(mary)	:mary rdf:type :Person .
Woman ⊑ Person - Person ≡ HumanBeing	:Woman rdfs:subClassOf :Person .
hasWife(john,mary)	:john :hasWife :mary .
hasWife ⊑ hasSpouse - hasSpouse ≡ marriedWith	:hasWife rdfs:subPropertyOf :hasSpouse .

Special classes and properties

- owl:Thing (RDF syntax)
  - DL-syntax: $\top$
  - contains everything
- owl:Nothing (RDF syntax)
  - DL-syntax: $\bot$
  - empty class
- owl:topProperty (RDF syntax)
  - DL-syntax: U
  - every pair is in U
- owl:bottomProperty (RDF syntax)
  - empty property

Class constructors

- **conjunction**

  $\forall x\ (Mother(x) \leftrightarrow Woman(x) \land Parent(x))$

  - **Mother ≡ Woman ⊓ Parent**
  - **:Mother owl:equivalentClass _:x .**
    _:x rdf:type owl:Class .
    _:x owl:intersectionOf (:Woman :Parent) .
- **disjunction**

  $\forall x\ (Parent(x) \leftrightarrow Mother(x) \land Father(x))$

  - **Parent ≡ Mother ⊔ Father**
  - **:Parent owl:equivalentClass _:x .**
    _:x rdf:type owl:Class .
    _:x owl:unionOf (:Mother :Father) .
- **negation**

  $\forall x\ (ChildlessPerson(x) \leftrightarrow Person(x) \land \neg Parent(x))$

  - **ChildlessPerson ≡ Person ⊓ ¬Parent**
  - **:ChildlessPerson owl:equivalentClass _:x .**
    _:x rdf:type owl:Class .
    _:x owl:intersectionOf (:Person _:y) .
    _:y owl:complementOf :Parent .

Class constructors

- existential quantification
  - only to be used with a role – also called a *property restriction*
  - Parent ≡ ∃hasChild.Person
  - :Parent owl:equivalentClass _:x .
    _:x rdf:type owl:Restriction .
    _:x owl:onProperty :hasChild .
    _:x owl:someValuesFrom :Person .

$\forall x\,(Parent(x) \leftrightarrow \exists y\,(hasChild(x,y) \land Person(y)))$

- universal quantification
  - only to be used with a role – also called a *property restriction*
  - Person ⊓ Happy ≡ ∀hasChild.Happy
  - _:x rdf:type owl:Class .
    _:x owl:intersectionOf (:Person :Happy) .
    _:x owl:equivalentClass _:y .
    _:y rdf:type owl:Restriction .
    _:y owl:onProperty :hasChild .
    _:y owl:allValuesFrom :Happy .

$\forall x\,(Person(x) \land Happy(x) \leftrightarrow \forall y\,(hasChild(x,y) \rightarrow Happy(y)))$

- Class constructors can be nested arbitrarily

Contents

Understanding SROIQ(D)

The description logic ALC

Complexity: ExpTime

- **ABox expressions:**
  Individual assignments **Father(john)**
  Property assignments **hasWife(john,mary)**

- **TBox expressions**
  subclass relationships $\sqsubseteq$

  conjunction $\sqcap$
  disjunction $\sqcup$
  negation $\neg$

  property restrictions $\forall$
  $\exists$

Also: $\top$, $\bot$

Understanding SROIQ(D)

ALC + role chains = SR

- **hasParent o hasBrother $\sqsubseteq$ hasUncle**
  - **:hasUncle owl:propertyChainAxiom (:hasParent :hasBrother) .**

    $\forall x\ \forall y\ (\exists z\ ((hasParent(x,z) \wedge hasBrother(z,y)) \rightarrow hasUncle(x,y)))$

  - **includes top property and bottom property**
- **includes S = ALC + transitivity**
  - **hasAncestor o hasAncestor $\sqsubseteq$ hasAncestor**
- **includes SH = S + role hierarchies**
  - **hasFather $\sqsubseteq$ hasParent**

I'll skip RDF syntax in the following.

Understanding SROIQ(D)

- O – nominals (closed classes)
  - MyBirthdayGuests $\equiv$ {bill,john,mary}
  - Note the difference to
    MyBirthdayGuests(bill)
    MyBirthdayGuests(john)
    MyBirthdayGuests(mary)
- Individual equality and inequality (no unique name assumption!)
  - bill = john
    - {bill} $\equiv$ {john}
  - bill $\neq$ john
    - {bill} $\sqcap$ {john} $\equiv \bot$

Understanding SROIQ(D)

- I – inverse roles
  - hasParent $\equiv$ hasChild^{-}
  - Orphan $\equiv$ $\forall$hasChild^{-}.Dead
- Q – qualified cardinality restrictions
  - $\leq$4 hasChild.Parent(john)
  - HappyFather $\equiv$ $\geq$2 hasChild.Female
  - Car $\sqsubseteq$ =4hasTyre.$\top$
- Complexity SHIQ, SHOQ, SHIO: ExpTime.
  Complexity SHOIQ: NExpTime
  Complexity SROIQ: N2ExpTime

Understanding SROIQ(D)

Properties can be declared to be

Transitive	hasAncestor
Symmetric	hasSpouse
Asymmetric	hasChild
Reflexive	hasRelative
Irreflexive	parentOf
Functional	hasHusband
InverseFunctional	hasHusband

called *property characteristics*

Understanding SROIQ(D)

(D) – datatypes

- **so far, we have only seen properties with individuals in second argument, called *object properties* or *abstract roles* (DL)**
- **properties with datatype literals in second argument are called *data properties* or *concrete roles* (DL)**
- **allowed are many XML Schema datatypes, including xsd:integer, xsd:string, xsd:float, xsd:booelan, xsd:anyURI, xsd:dateTime**

  and also e.g. owl:real

Understanding SROIQ(D)

(D) – datatypes

- **hasAge(john, "51"^^xsd:integer)**
- **additional use of *constraining facets* (from XML Schema)**
  - **e.g. Teenager ≡ Person ⊓ ∃hasAge.(xsd:integer: ≥12 and ≤19)**
    note: this is not standard DL notation!
  - **:Teenager rdfs:subClassOf _:x .**
    _:x rdf:type owl:Restriction .
    _:x owl:onProperty :hasAge .
    _:x owl:someValuesFrom _:y .
    _:y rdf:type rdfs:Datatype .
    _:y owl:onDatatype xsd:integer .
    _:y owl:withRestrictions (
    [xsd:minInclusive "13"^^xsd:integer]
    [xsd:maxInclusive "19"^^xsd:integer]) .

Understanding SROIQ(D)

further expressive features

- Self
  - PersonCommittingSuicide ≡ ∃kills.Self
- Keys (not really in SROIQ(D), but in OWL)
  - set of (object or data) properties whose values uniquely identify an object
- disjoint properties
  - Disjoint(hasParent,hasChild)
- explicit anonymous individuals
  - as in RDF: can be used instead of named individuals

SROIQ(D) constructors – overview

Karlsruhe Institute of Technology

- 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,
  role 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)

Some Syntactic Sugar in OWL

This applies to the non-DL syntaxes (e.g. RDF syntax).

- **disjoint classes**
  - **Apple $\sqcap$ Pear $\sqsubseteq \bot$**
- **disjoint union**
  - **Parent $\equiv$ Mother $\sqcup$ Father**
    Mother $\sqcap$ Father $\sqsubseteq \bot$
- **negative property assignments (also for datatypes)**
  - **¬hasAge(jack,"53"^^xsd:integer)**

Contents

OWL – Extralogical Features

- **OWL ontologies have URIs and can be referenced by others via**
  - **import statements**
- **Namespace declarations**
- **Entity declarations (must be done)**
- **Versioning information etc.**

- **Annotations**
  - **Entities and axioms (statements) can be endowed with annotations, e.g. using rdfs:comment.**
  - **OWL syntax provides *annotation properties* for this purpose.**

The modal logic perspective

- Description logics can be understood from a modal logic perspective.
- Each pair of $\forall R$ and $\exists R$ statements give rise to a pair of modalities.
- Essentially, some description logics are multi-modal logics.
- See [The Description Logic Handbook].

The RDFS perspective

RDFS semantics is weaker

- :mary rdf:type :Person .
- :Mother rdfs:subClassOf :Woman .
- :john :hasWife :Mary .
- :hasWife rdfs:subPropertyOf :hasSpouse

- :hasWife rdfs:range :Woman .
- :hasWife rdfs:domain :Man .

- Person(mary)
- Mother $\sqsubseteq$ Woman
- hasWife(john,mary)
- hasWife $\sqsubseteq$ hasSpouse

- $\top \sqsubseteq \forall$hasWife.Woman
- $\top \sqsubseteq \forall$hasWife^{-}.Man or $\exists$hasWife.$\top \sqsubseteq$ Man

RDFS also allows to

- make statements about statements
  $\rightarrow$ only possible through annotations in OWL
- mix class names, individual names, property names (they are all URIs)
  $\rightarrow$ *punning* in OWL

Punning

- Description logics impose *type separation*, i.e. names of individuals, classes, and properties must be disjoint.
- In OWL 2 Full, type separation does not apply.
- In OWL 2 DL, type separation is relaxed, but a class X and an individual X are interpreted semantically as if they were different.
- Father(john)
  SocialRole(Father)
- See further below on the two different semantics for OWL.

Contents

Expressivity Examples: Rules in OWL

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

- $kills(x,x) \rightarrow suicide(x)$ $\qquad$ $suicide(x) \rightarrow kills(x,x)$
  - $\exists kills.Self \sqsubseteq suicide$ $\qquad$ $suicide \sqsubseteq \exists kills.Self$

Note: with these two axioms,
suicide* is basically the same as *kills

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

Expressivity Examples: Rules in OWL

- dislikes(x,z) ∧ Dish(y) ∧ contains(y,z) → dislikes(x,y)
  - Dish ≡ ∃dish.Self
    dislikes o contains⁻ o dish ⊑ dislikes
- worksAt(x,y) ∧ University(y) ∧ supervises(x,z) ∧ PhDStudent(z) → professorOf(x,z)
  - ∃worksAt.University ≡ ∃worksAtUniversity.Self
    PhDStudent ≡ ∃phDStudent.Self
    worksAtUniversity o supervises o phDStudent ⊑ professorOf
- Basic requirement for expressibility of rules in OWL 2: tree-shapedness of rule bodies
- For more on this, see
  [Description Logic Rules] and [ELP].

Contents

OWL Semantics

- There are two semantics for OWL.

1. Description Logic Semantics
   also: Direct Semantics; FOL Semantics
   Can be obtained by translation to FOL.
   Syntax restrictions apply! (see next slide)

2. RDF-based Semantics
   No syntax restrictions apply.
   Extends the direct semantics with RDFS-reasoning features.

In the following, we will deal with the direct semantics only.

OWL Direct Semantics

Karlsruhe Institute of Technology

To obtain decidability, syntactic restrictions apply.

- **Type separation / punning**
- **No cycles in property chains.**
- **No transitive properties in cardinality restrictions.**

OWL Direct Semantics: Restrictions

- arbitrary property chain axioms lead to undecidability
- restriction: set of property chain axioms has to be *regular*
  - there must be a strict linear order $\prec$ on the properties
  - every property chain axiom has to have one of the following forms:

    $R \circ R \sqsubseteq R$ $\quad S^{-} \sqsubseteq R$ $\quad S_1 \circ S_2 \circ \ldots \circ S_n \sqsubseteq R$

    $R \circ S_1 \circ S_2 \circ \ldots \circ S_n \sqsubseteq R$ $\quad S_1 \circ S_2 \circ \ldots \circ S_n \circ R \sqsubseteq R$
  - thereby, $S_i \prec R$ for all i= 1, 2, . . . , *n.*

- Example 1: $R \circ S \sqsubseteq R$ $\quad S \circ S \sqsubseteq S$ $\quad R \circ S \circ R \sqsubseteq T$

  → regular with order $S \prec R \prec T$
- Example 2: $R \circ T \circ S \sqsubseteq T$

  → not regular because form not admissible
- Example 3: $R \circ S \sqsubseteq S$ $\quad S \circ R \sqsubseteq R$

  → not regular because no adequate order exists

OWL Direct Semantics: Restrictions

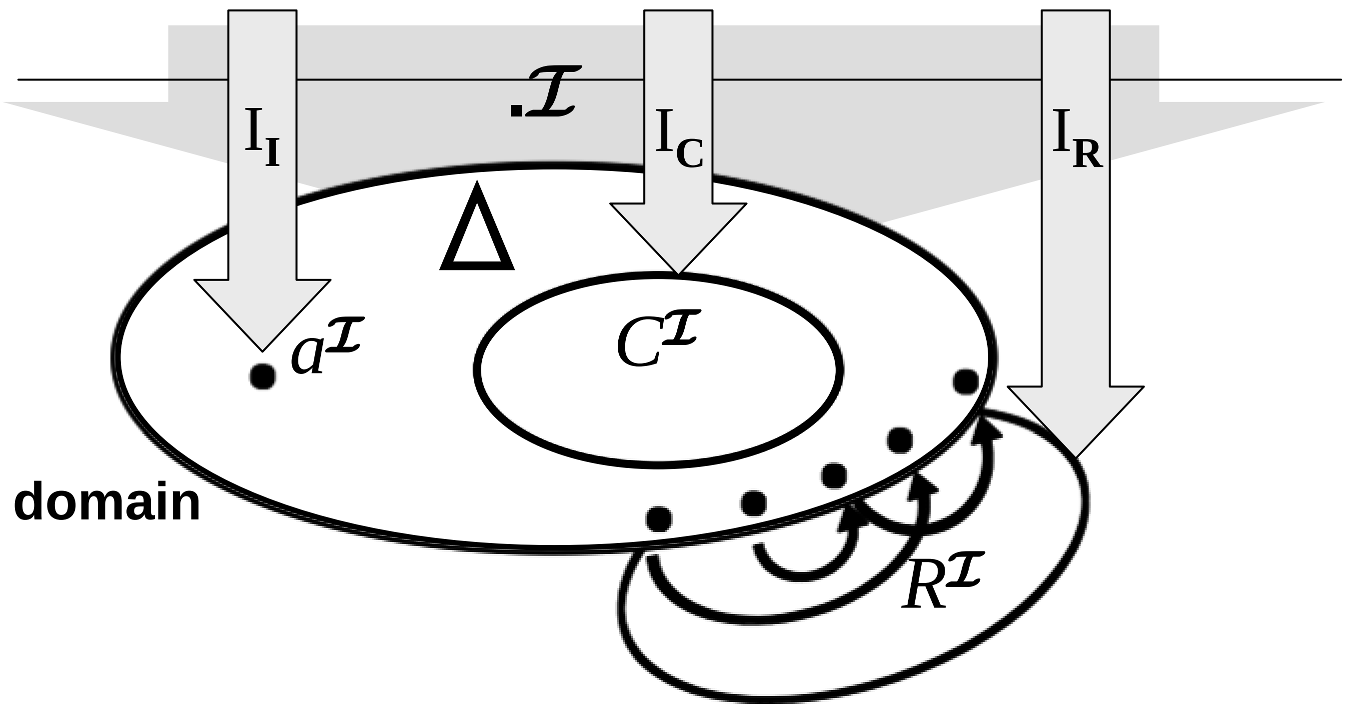
- combining property chain axioms and cardinality constraints may lead to undecidability
- restriction: use only *simple* properties in cardinality expressions (i.e. those which cannot be – directly or indirectly – inferred from property chains)
- technically:
  - for any property chain axiom $S_1 \circ S_2 \circ_{...} \circ S_n \sqsubseteq R$ with n>1, R is non-simple
  - for any subproperty axiom $S \sqsubseteq R$ with S non-simple, R is non-simple
  - all other properties are simple
- Example: $Q \circ P \sqsubseteq R$ $R \circ P \sqsubseteq R$ $R \sqsubseteq S$ $P \sqsubseteq R$ $Q \sqsubseteq S$

  non-simple: R, S simple: P, Q

OWL Direct Semantics

- model-theoretic semantics
- starts with interpretations
- an interpretation maps

individual names, class names and property names...



...into a domain

OWL Direct Semantics

- **mapping is extended to complex class expressions:**
  - $\top^I = \Delta^I$ $\qquad \bot^I = \emptyset$
  - $(C \sqcap D)^I = C^I \cap D^I$ $\qquad (C \sqcup D)^I = C^I \cup D^I (\neg C)^I = \Delta^I \setminus C^I$
  - $\forall R.C = \{ x \mid \forall (x,y) \in R^I \rightarrow y \in C^I \}$
    $\exists R.C = \{ x \mid \exists (x,y) \in R^I \wedge y \in C^I \}$
  - $\geq nR.C = \{ x \mid \#\{ y \mid (x,y) \in R^I \wedge y \in C^I \} \geq n \}$
  - $\leq nR.C = \{ x \mid \#\{ y \mid (x,y) \in R^I \wedge y \in C^I \} \leq n \}$
- **...and to role expressions:**
  - $U^I = \Delta^I \times \Delta^I$ $\qquad (R^-)^I = \{ (y,x) \mid (x,y) \in R^I \}$
- **...and to axioms:**
  - $C(a)$ holds, if $a^I \in C^I$ $\qquad R(a,b)$ holds, if $(a^I,b^I) \in R^I$
  - $C \sqsubseteq D$ holds, if $C^I \subseteq D^I$ $\qquad R \sqsubseteq S$ holds, if $R^I \subseteq S^I$
  - $Dis(R,S)$ holds if $R^I \cap S^I = \emptyset$
  - $S_1 \circ S_2 \circ _{\dots} \circ S_n \sqsubseteq R$ holds if $S_1^I \circ S_2^I \circ _{\dots} \circ S_n^I \subseteq R^I$

OWL Direct Semantics via FOL

- **but often OWL 2 DL is said to be a fragment of FOL (with equality)...**
- **yes, there is a translation of OWL 2 DL into FOL**

$$
\begin{aligned}
\pi(C \sqsubseteq D) &= (\forall x)(\pi_x(C) \to \pi_x(D))\\
\pi_x(A) &= A(x)\\
\pi_x(\neg C) &= \neg \pi_x(C)\\
\pi_x(C \sqcap D) &= \pi_x(C) \wedge \pi_x(D)\\
\pi_x(C \sqcup D) &= \pi_x(C) \vee \pi_x(D)\\
\pi_x(\forall R.C) &= (\forall x_1)(R(x,x_1) \to \pi_{x_1}(C))\\
\pi_x(\exists R.C) &= (\exists x_1)(R(x,x_1) \wedge \pi_{x_1}(C))\\
\pi_x(\geq nS.C) &= (\exists x_1)\ldots(\exists x_n)\left(\bigwedge_{i\neq j}(x_i \neq x_j) \wedge \bigwedge_i (S(x,x_i) \wedge \pi_{x_i}(C))\right)\\
\pi_x(\leq nS.C) &= \neg(\exists x_1)\ldots(\exists x_{n+1})\left(\bigwedge_{i\neq j}(x_i \neq x_j) \wedge \bigwedge_i (S(x,x_i) \wedge \pi_{x_i}(C))\right)\\
\pi_x(\{a\}) &= (x = a)\\
\pi_x(\exists S.\mathrm{Self}) &= S(x,x)
\end{aligned}
$$

$$
\begin{aligned}
\pi(R_1 \sqsubseteq R_2) &= (\forall x)(\forall y)(\pi_{x,y}(R_1) \to \pi_{x,y}(R_2))\\
\pi_{x,y}(S) &= S(x,y)\\
\pi_{x,y}(R^-) &= \pi_{y,x}(R)\\
\pi_{x,y}(R_1 \circ \cdots \circ R_n) &= (\exists x_1)\ldots(\exists x_{n-1})\\
&\quad \left(\pi_{x,x_1}(R_1) \wedge \bigwedge_{i=1}^{n-2} \pi_{x_i,x_{i+1}}(R_{i+1}) \wedge \pi_{x_{n-1},y}(R_n)\right)\\
\pi(\mathrm{Ref}(R)) &= (\forall x)\pi_{x,x}(R)\\
\pi(\mathrm{Asy}(R)) &= (\forall x)(\forall y)(\pi_{x,y}(R) \to \neg\pi_{y,x}(R)\\
\pi(\mathrm{Dis}(R_1,R_2)) &= \neg(\exists x)(\exists y)(\pi_{x,y}(R_1) \wedge \pi_{x,y}(R_2))
\end{aligned}
$$

- **...which (interpreted under FOL semantics) coincides with the definition just given.**

Contents

OWL Profiles

- OWL Full – using the RDFS-based semantics
- OWL DL – using the FOL semantics

The OWL 2 documents describe further profiles, which are of polynomial complexity:

- OWL EL (EL++)
- OWL QL (DL Lite$_R$)
- OWL RL (DLP)

OWL 2 EL

Karlsruhe Institute of Technology

- **allowed:**
  - **subclass axioms with intersection, existential quantification, top, bottom**
    - **closed classs must have only one member**
  - **property chain axioms, range restrictions (under certain conditions)**
- **disallowed:**
  - **negation, disjunction, arbitrary universal quantification, role inverses**

$$\sqcap \exists \top \bot \sqsubseteq \sqcap \exists \top \bot$$

- **Examples:** $\exists$**has.Sorrow** $\sqsubseteq$ $\exists$**has.Liqueur;** $\top \sqsubseteq$ $\exists$**hasParent.Person** $\exists$**married.**$\top \sqcap$ **CatholicPriest** $\sqsubseteq \bot$**; German** $\sqsubseteq$ $\exists$**knows.{angela}; hasParent** $\pm$ **hasParent** $\sqsubseteq$ **hasGrandparent**

OWL 2 RL

- motivated by the question: what fraction of OWL 2 DL can be expressed naively by rules (with equality)?
- examples:
  - $\exists$parentOf.$\exists$parentOf.$\top \sqsubseteq$ Grandfather

    rule version: parentOf(x,y) $\wedge$ parentOf(y,z) $\rightarrow$ Grandfather(x)
  - Orphan $\sqsubseteq \forall$hasParent.Dead

    rule version: Orphan(x) $\wedge$ hasParent(x,y) $\rightarrow$ Dead(y)
  - Monogamous $\sqsubseteq$ $\leq$1married.Alive

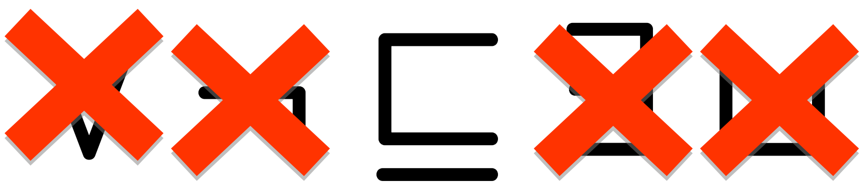
    rule version:
    Monogamous(x) $\wedge$ married(x,y) $\wedge$ Alive(y) $\wedge$ married(x,z) $\wedge$ Alive(z)$\rightarrow$ y=z
  - childOf $\pm$ childOf $\sqsubseteq$ grandchildOf

    rule version: childOf(x,y) $\wedge$ childOf(y,z) $\rightarrow$ grandchildOf(x,z)
  - Disj(childOf,parentOf)

    rule version: childOf(x,y) $\wedge$ parentOf(x,y) $\rightarrow$

OWL 2 RL

- syntactic characterization:
  - essentially, all axiom types are allowed
  - disallow certain constructors on lhs and rhs of subclass statements



  - cardinality restrictions: only on rhs and only ≤1 and ≤0 allowed
  - closed classes: only with one member

- Reasoner conformance requires only soundness.

OWL 2 QL

Karlsruhe Institute of Technology

- motivated by the question: what fraction of OWL 2 DL can be captured by standard database technology?
- formally: query answering LOGSPACE w.r.t. data (via translation into SQL)
- allowed:
  - subproperties, domain, range
  - subclass statements with
    - left hand side: class name or expression of type $\exists r.\top$
    - right hand side: intersection of class names, expressions of type $\exists r.C$ and negations of lhs expressions
    - no closed classes!
- Example:

$\exists$married.$\top \sqsubseteq \neg$Free $\sqcap$ $\exists$has.Sorrow

Contents

Proof Theory

- Traditionally using tableaux algorithms (see below)

Alternatives:

- Transformation to disjunctive datalog using basic superposition done for SHIQ
- Naive mapping to Datalog for OWL RL
- Mapping to SQL for OWL QL
- Special-purpose algorithms for OWL EL e.g. transformation to Datalog

Proof theory Via Tableaux

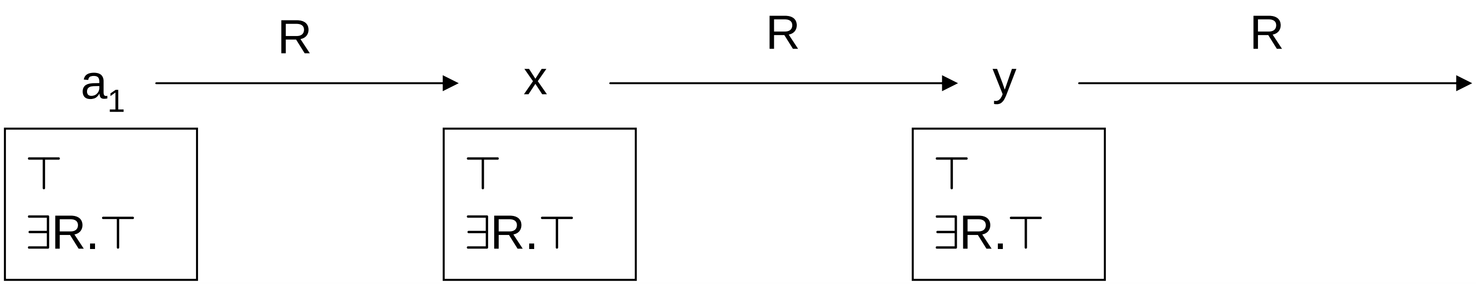
- **Adaptation of FOL tableaux algorithms.**
- **Problem: OWL is decidable, but FOL tableaux algorithms do not guarantee termination.**
- **Solution: *blocking*.**

DL Tableaux Termination Problem

TBox: $\exists R.\top$

ABox: $\top(a_1)$

- **Is satisfiable:**
  Model M contains elements $a_1^M, a_2^M, \ldots$
  and $R^M(a_i^M, a_{i+1}^M)$ for all $i \geq 1$.
- **But naive tableau does not terminate!**

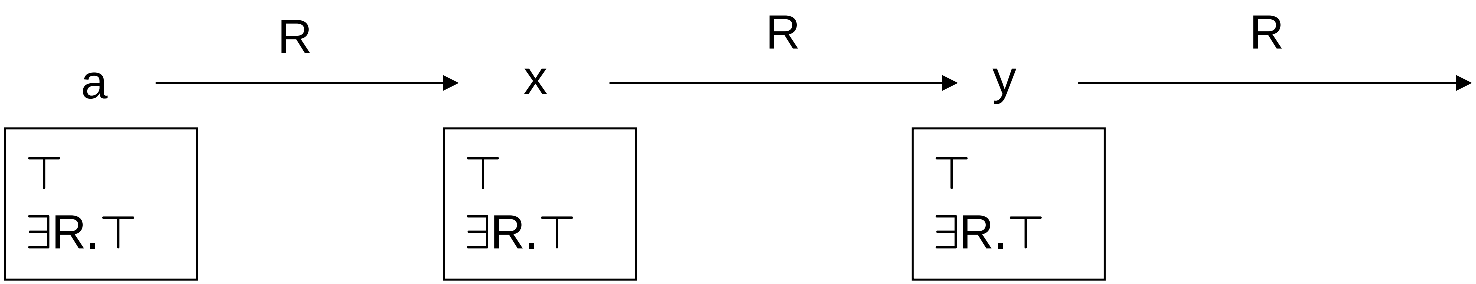


DL Tableaux Termination Problem

Nothing essentially new happens.

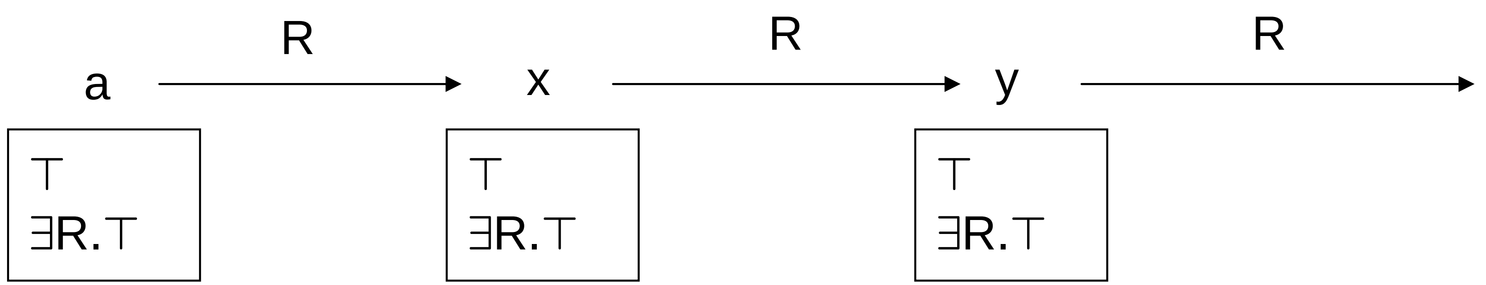
Idea: y does not need to be expanded, because it is basically a copy of x.

⇒ Blocking



Blocking (in ALC)

- y is *blocked* (by x) if
  - y is not an individual (but a variable),
  - y is a successor of x and $L(y) \subseteq L(x)$,
  - or an ancestor of y is blocked.



y blocked by x in this example.

Blocking conditions for more expressive DLs are more involved; the idea is the same.

ALC Tableau example

Show that

C(a)
R(a,b)
S(a,a)
$C \sqsubseteq \forall S.A$
$A \sqsubseteq \exists R.\exists S.A$
$A \sqsubseteq \exists R.C$

C(c)
R(a,c)
S(c,b)

implies $\exists R.\exists R.\exists S.A(a)$.

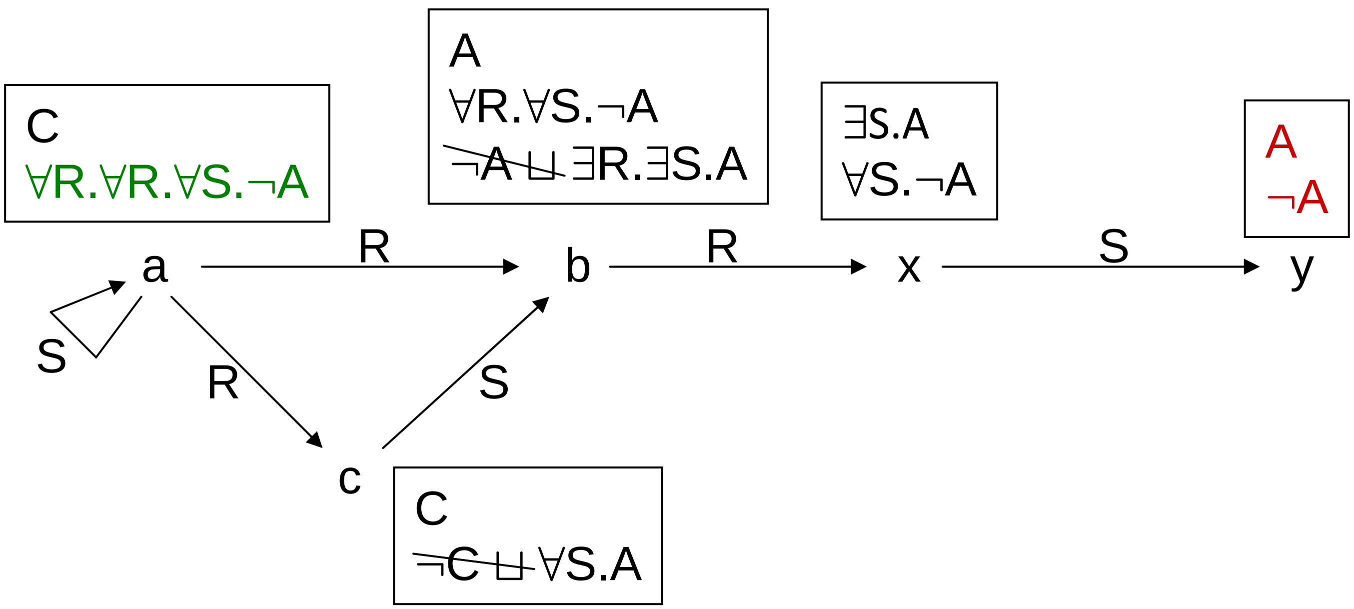
ALC Tableau Example

TBox:
¬C ⊔ ∀S.A
¬A ⊔ ∃R.∃S.A
¬A ⊔ ∃R.C

ABox

C(a)	C(c)
R(a,b)	R(a,c)
S(a,a)	S(c,b)

¬∃R.∃R.∃S.A(a) is ∀R.∀R.∀S.¬A(a)



Contents

OWL tools (incomplete listing)

Reasoner:

- **OWL 2 DL:**
  - **Pellet** http://clarkparsia.com/pellet/
  - **HermiT** http://www.hermit-reasoner.com/
- **OWL 2 EL:**
  - **CEL** http://code.google.com/p/cel/
- **OWL 2 RL:**
  - **essentially any rule engine**
- **OWL 2 QL:**
  - **essentially any SQL engine (with a bit of query rewriting on top)**

Editors:

- **Protégé**
- **NeOn Toolkit**
- **TopBraid Composer**

Contents

Some Current Research Questions

- Integrating OWL and Rules
- Inconsistency handling / paraconsistent reasoning
- Local closed world reasoning
- Uncertainty handling (fuzzy / probabilistic)
- Modularization
- Distributedness
- Belief Revision (Ontology Evolution)
- Abduction/Explanation/Justification
- Approximate Reasoning
- Ontology Learning
- Modelling / Design Patterns
- Ontology Engineering (Modelling Processes)
- Interfaces (GUIs, Constrained Natural Language, etc.)

Further remarks

- Several major conferences on Semantic Web:
  - ISWC (>600), ESWC (>300), WWW Semantic Web track, IJCAI Semantic Web track, etc.
- Semantic Web languages taught in many university courses world-wide.
  - Becomes established topic.
- Industrial uptake currently happening
  - e.g. OWL reasoners by IBM, ORACLE
  - many application studies by major IT companies
  - considerable number of spin-offs
  - venture capital (e.g. VULCAN Inc.)
- Considerable uptake in the life sciences

- Substantial project funding (EU, NIH, etc.)

Suggestions for OWL?

Karlsruhe Institute of Technology

- Annual Workshop OWL: Experiences and Directions
- Co-located with ISWC09 (just beforehand), October 2009.
- Usually >80 people, most of them doing *applications*.
  Major OWL language designers are there.
- Past discussions had major impact on OWL 2
  $\rightarrow$ state your opinon there!
- Low paper barrier, position statements and experience reports welcome. Deadline July 24th.
- http://www.webont.org/owled/2009/

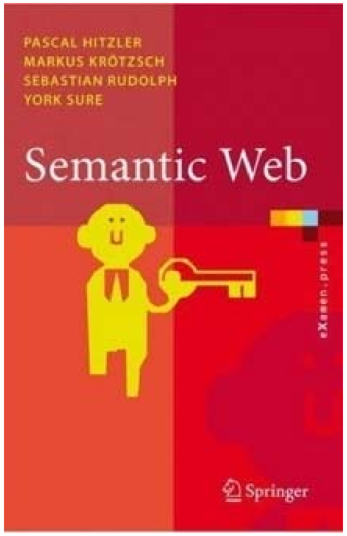
Contents

Main References

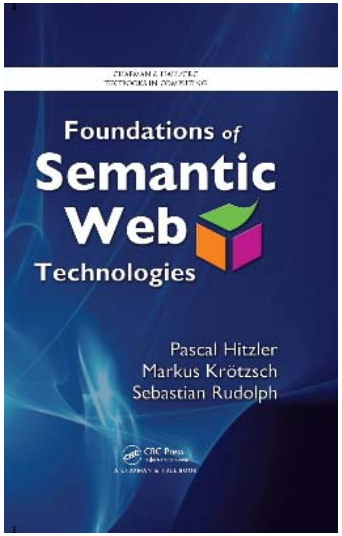
- W3C OWL Working Group, OWL 2 Web Ontology Language: Document Overview. http://www.w3.org/TR/owl2-overview/
- Pascal Hitzler, Markus Krötzsch, Bijan Parsia, Peter Patel-Schneider, Sebastian Rudolph, OWL 2 Web Ontology Language: Primer. http://www.w3.org/TR/owl2-primer/
- Franz Baader, Diego Calvanese, Deborah L. McGuinness, Daniele Nardi, Peter F. Patel-Schneider, The Description Logic Handbook: Theory, Implementation, and Applications. Cambridge University Press, 2nd edition, 2007.

Main References – Textbooks

- **Pascal Hitzler, Markus Krötzsch, Sebastian Rudolph, York Sure, Semantic Web – Grundlagen. Springer, 2008. http://www.semantic-web-grundlagen.de/ (In German.) (Does not cover OWL 2.)**



- **Pascal Hitzler, Markus Krötzsch, Sebastian Rudolph, Foundations of Semantic Web Technologies. Chapman & Hall/CRC, 2009. http://www.semantic-web-book.org/wiki/FOST (Ask for a flyer from us.)**



Further References

- DL complexity calculator: http://www.cs.man.ac.uk/~ezolin/dl/

- 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 (ECAI-08), pp. 80–84. IOS Press 2008.

- Markus Krötzsch, Sebastian Rudolph, Pascal Hitzler, ELP: Tractable Rules for OWL 2. In: Amit Sheth, Steffen Staab, Mike Dean, Massimo Paolucci, Diana Maynard, Timothy Finin, Krishnaprasad Thirunarayan (eds.), The Semantic Web - ISWC 2008, 7th International Semantic Web Conference. Springer Lecture Notes in Computer Science Vol. 5318, 2008, pp. 649-664.

Thanks!

http://semantic-web-grundlagen.de/wiki/IJCAI-09_Tutorial