XML and Semantic Web Technologies

II. Semantic Web / 2. Web Ontology Language (OWL)

Lars Schmidt-Thieme

Information Systems and Machine Learning Lab (ISMLL)
Institute of Economics and Information Systems
& Institute of Computer Science
University of Hildesheim
http://www.ismll.uni-hildesheim.de

II. Semantic Web / 2. Web Ontology Language (OWL)

Named Individuals, Classes, and Roles (1/2)

Description logics is a family of special logical calculi that can be represented as subsets of FOL.

named individual

- represents single entity of the domain, e.g., specific persons, cities, etc.
- denoted by $a, b, c, \ldots$.
- corresponds to constants in FOL.
- interpreted by elements of the domain in models ($a^I \in \Delta^I$).

named concept (aka atomic, primitive; aka class)

- represents a class of individuals
- denoted by $A, B, C, \ldots$.
- $\top$ denotes the all class (universe) Δ^I, $\bot$ denotes the empty class.
- corresponds to unary predicates in FOL.
- interpreted by sets of elements of the domain in models ($A^I \subseteq \Delta^I$).

Named Individuals, Classes, and Roles (2/2)

named role (aka atomic, primitive; aka property)

- represents a relation between two individuals
- denoted by $R, S, T, \ldots$.
- corresponds to binary predicates in FOL.
- interpreted by relations between elements of the domain in models ($R^I \subseteq \Delta^I \times \Delta^I$).

Statements (1/2)

$a : A$

- individual a belongs to class A, a is a A.
- $A(a)$ in FOL.
- $a^I \in A^I$ in models.
- example:

  charlesChaplin: Director
  goldrush: Movie

$\langle a, b \rangle : R$

- individual a and b are related by R.
- $R(a, b)$ in FOL.
- $(a^I, b^I) \in R^I$ in models.
- example:

  $\langle \text{charlesChaplin}, \text{goldrush} \rangle : \text{directs}$
  $\langle \text{adam}, \text{abel} \rangle : \text{parentOf}$

Statements (2/2)

$A \sqsubseteq B$

- class A is a subclass of class B.
- $\forall x A(x) \rightarrow B(x)$ in FOL.
- $A^I \subseteq B^I$ in models.
- example:

  Director $\sqsubseteq$ Professional $\sqsubseteq$ Human
  Movie $\sqsubseteq$ Artwork

$A \doteq B$ abbreviation for $A \sqsubseteq B$ and $B \sqsubseteq A$.

Taxonomies

Classes form a taxonomy that can be represented graphically by a directed acyclic graph.

- individuals can belong to several classes, even if none of them is a subclass of another. example:

  charlesChaplin : Male
  charlesChaplin : Director

- classes can have several superclasses, even if none of them is a subclass of another (multiple inheritance). example:

  Actrice : Female
  Actrice : Professional

- subclasses have not to be exhaustive, interior nodes have not to be abstract, but can contain individuals that belong to none of the subclasses. example: there are female professionals that are not actrices (e.g., directors).

Class constructors: Boolean expressions (1/2)

New classes can be constructed, e.g., from boolean expressions of existing classes. These new classes then can be used in statements wherever named classes can be used.

$A \sqcap B$

- class of individuals belonging to both classes A and B.
- $A(x) \wedge B(x)$ in FOL.
- $A^I \cap B^I$ in models.
- example:

  CityState$\doteq$Country $\sqcap$ City
  luxemburg : CityState

$A \sqcup B$

- class of individuals belonging to at least one of the classes A or B.
- $A(x) \vee B(x)$ in FOL.
- $A^I \cup B^I$ in models.
- example:

  Parents$\doteq$Father $\sqcup$ Mother

Class constructors: Boolean expressions (1/2)

$\neg A$

- class of individuals not belonging to class A.
- $\neg A(x)$ in FOL.
- $\Delta^I \setminus A^I$ in models.
- example:

  Female $\doteq$ Human $\sqcap$ $\neg$Male

Class constructors: exists restriction (1/2)

Furthermore, classes can be constructed from roles
by posing constraints on the related individuals (restrictions).

$\exists R$ unqualified exists restriction

- class of individuals that are related to at least one other individual by relation R.
- $\exists y R(x, y)$ in FOL.
- $\{x \in \Delta^I \mid \exists y \in \Delta^I : (x, y) \in R^I\}$ in models.
- example:

  Parents$\doteq\exists$ParentOf

Class constructors: exists restriction (2/2)

$\exists R.A$ qualified exists restriction

- class of individuals that are related to at least one individual of class A by relation R.
- $\exists y A(y) \wedge R(x,y)$ in FOL.
- $\{x \in \Delta^I \,|\, \exists y \in A^I : (x,y) \in R^I\}$ in models.
- example:

  ParentsOfASon$\doteq\exists$ParentOf.Male

- $\exists R.\top$ could be used to express an unqualified exists expression.

Class constructors: value restriction

$\forall R.A$ value restriction (always qualified)

- class of individuals that are related only to individuals of class A by relation R.
- $\forall y R(x,y) \rightarrow A(y)$ in FOL.
- $\{x \in \Delta^I \,|\, \forall y \in \Delta^I : (x,y) \in R^I \rightarrow y \in A^I\}$ in models.
- example:

  HappyParents$\doteq\forall$ParentOf.Happy

All qualified restrictions could be nested, e.g.,

Grandparents$\doteq\exists$ParentOf.$\exists$ParentOf.$\top$

A simple description logic calculus: FL$^-$

FL$^-$ class constructors:

symbol	name	formal semantics
A	atomic / primitive concept	$C^\mathcal{I} \subseteq \Delta^\mathcal{I}$
R	atomic / primitive role	$R^\mathcal{I} \subseteq \Delta^\mathcal{I} \times \Delta^\mathcal{I}$
$C \sqcap D$	conjoined concept, *and*	$C^\mathcal{I} \cap D^\mathcal{I}$
$\exists R$	exists restriction, *some*	$\{x \in \Delta^\mathcal{I} \mid \exists y \in \Delta^\mathcal{I} : (x, y) \in R^\mathcal{I}\}$
$\forall R.C$	value restriction, *all*	$\{x \in \Delta^\mathcal{I} \mid \forall y \in \Delta^\mathcal{I} : (x, y) \in R^\mathcal{I} \rightarrow y \in C^\mathcal{I}\}$

FL$^-$ statements:

$C \sqsubseteq D$	subsumption, axiom	$C^\mathcal{I} \subseteq D^\mathcal{I}$
$C \doteq D$	concept equality / definition	$C^\mathcal{I} = D^\mathcal{I}$
$a : C$	concept assertion	$a^\mathcal{I} \in C^\mathcal{I}$
$\langle a, b \rangle : R$	role assertion	$(a^\mathcal{I}, b^\mathcal{I}) \in R^\mathcal{I}$

concept = class, set of individuals.
role = relation between pairs of individuals.

ALC

ALC class constructors:

symbol	name	formal semantics
A	atomic / primitive concept	$C^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$
R	atomic / primitive role	$R^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$
$\top$	top, all class	$\Delta^{\mathcal{I}}$
$\bot$	bottom, empty class	$\emptyset$
$\neg C$	complement, *not*	$\Delta^{\mathcal{I}} \setminus C^{\mathcal{I}}$
$C \sqcap D$	conjunction, *and*	$C^{\mathcal{I}} \cap D^{\mathcal{I}}$
$C \sqcup D$	disjunction, *or*	$C^{\mathcal{I}} \cup D^{\mathcal{I}}$
$\exists R.C$	exists restriction, *some*	$\{x \in \Delta^{\mathcal{I}} \,\vert\, \exists y \in C^{\mathcal{I}} : (x, y) \in R^{\mathcal{I}}\}$
$\forall R.C$	value restriction, *all*	$\{x \in \Delta^{\mathcal{I}} \,\vert\, \forall y \in \Delta^{\mathcal{I}} : (x, y) \in R^{\mathcal{I}} \rightarrow y \in C^{\mathcal{I}}\}$

ALC statements:

$C \sqsubseteq D$	subsumption, axiom	$C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$
$C \doteq D$	concept equality / definition	$C^{\mathcal{I}} = D^{\mathcal{I}}$
$a : C$	concept assertion	$a^{\mathcal{I}} \in C^{\mathcal{I}}$
$\langle a, b \rangle : R$	role assertion	$(a^{\mathcal{I}}, b^{\mathcal{I}}) \in R^{\mathcal{I}}$

ALC example

Formalize the concept of DINK = "double income, no kids".

atomic concepts: Human, Employed.
atomic roles: partnerOf, parentOf.

$$\text{Employed} \sqsubseteq \text{Human}$$

$$\text{DINK} \doteq \text{Employed} \sqcap \exists \text{partnerOf.Employed} \sqcap \neg \exists \text{parentOf.Human}$$

Class constructors: functional number restrictions

$\leq 1R$ functional number restriction

- class of individuals that are related to at most one individual by relation R.
- $\forall y, z : R(x, y) \wedge R(x, z) \rightarrow y = z$ in FOL.
- $\{x \in \Delta^{\mathcal{I}} \mid \#\{y \in \Delta^{\mathcal{I}} \mid (x, y) \in R^{\mathcal{I}}\} \leq 1\}$ in models.
- example:

  Monogamous$\doteq \leq 1$MarriedTo

$\geq 2R$ functional number restriction

- class of individuals that are related to at least two individuals by relation R.
- $\exists y, z : R(x, y) \wedge R(x, z) \wedge y \neq z$ in FOL.
- $\{x \in \Delta^{\mathcal{I}} \mid \#\{y \in \Delta^{\mathcal{I}} \mid (x, y) \in R^{\mathcal{I}}\} \geq 2\}$ in models.
- example:

  Polygamous$\doteq \geq 2$MarriedTo

Do not confuse number restrictions with numeric properties (i.e., age of a person) or restrictions on numeric properties (i.e., a baby is a human of age at most 1 year).

Class constructors: (general) number restrictions

$\leq nR$ number restriction (with n a natural number).

- class of individuals that are related to at most n individuals by relation R.
- $\forall y_1, y_2, \ldots, y_n, y_{n+1} : R(x, y_1) \wedge R(x, y_2) \wedge \ldots \wedge R(x, y_{n+1}) \rightarrow (y_1 = y_2 \vee y_1 = y_3 \vee \ldots \vee y_n = y_{n+1})$ in FOL.
- $\{x \in \Delta^{\mathcal{I}} \,|\, \#\{y \in \Delta^{\mathcal{I}} \,|\, (x, y) \in R^{\mathcal{I}}\} \leq n\}$ in models.
- example:

  MiniGroup$\doteq \leq 5$HasMember

$\geq nR$ number restriction (with n a natural number).

- class of individuals that are related to at least n individuals by relation R.
- $\exists y_1, y_2, \ldots, y_n : R(x, y_1) \wedge R(x, y_2) \wedge \ldots \wedge R(x, y_n) \wedge y_1 \neq y_2 \wedge y_1 \neq y_3 \wedge \ldots \wedge y_{n-1} \neq y_n$ in FOL.
- $\{x \in \Delta^{\mathcal{I}} \,|\, \#\{y \in \Delta^{\mathcal{I}} \,|\, (x, y) \in R^{\mathcal{I}}\} \geq n\}$ in models.
- example:

  OverBookedAirbus380$\doteq \geq 556$HasBooking

Class constructors: qualified number restrictions

$\leq nR.A$ qualified number restriction (with n a natural number).

- class of individuals that are related to at most n individuals of class A by relation R.
- $\forall y_1, y_2, \ldots, y_n, y_{n+1} : A(y_1) \wedge A(y_2) \wedge \ldots \wedge A(y_{n+1}) \wedge R(x, y_1) \wedge R(x, y_2) \wedge \ldots \wedge R(x, y_{n+1}) \rightarrow (y_1 = y_2 \vee y_1 = y_3 \vee \ldots \vee y_n = y_{n+1})$ in FOL.
- $\{x \in \Delta^{\mathcal{I}} \,|\, \#\{y \in \Delta^{\mathcal{I}} \,|\, (x, y) \in R^{\mathcal{I}}\} \leq n\}$ in models.
- example:

  MiniGroup$\doteq \leq 5$HasMember.Adult

$\geq nR.A$ qualified number restriction (with n a natural number).

- class of individuals that are related to at least n individuals of class A by relation R.
- $\exists y_1, y_2, \ldots, y_n : A(y_1) \wedge A(y_2) \wedge \ldots \wedge A(y_n) \wedge R(x, y_1) \wedge R(x, y_2) \wedge \ldots \wedge R(x, y_n) \wedge y_1 \neq y_2 \wedge y_1 \neq y_3 \wedge \ldots \wedge y_{n-1} \neq y_n$ in FOL.
- $\{x \in \Delta^{\mathcal{I}} \,|\, \#\{y \in \Delta^{\mathcal{I}} \,|\, (x, y) \in R^{\mathcal{I}}\} \geq n\}$ in models.
- example:

  LargeFamily$\doteq \geq 3$HasMember.Child

Class constructors: nominal, enumeration

$\{o\}$ nominal

- class that contains exactly individual o.
- $x = o$ in FOL.
- $\{o\}$ in models.
- example:

  $\text{CharlesChaplin} \doteq \{\text{charlesChaplin}\}$

Enumerations can be formed by unions of nominals:

$$\{o_1, \ldots, o_n\} = \{o_1\} \sqcup \{o_2\} \sqcup \ldots \sqcup \{o_n\}$$

example:

$\text{GreatDirectors} \doteq \{\text{charlesChaplin}, \text{friedrichMurnau}, \text{fritzLang}\}$

ALC extensions / class constructors

class constructors of ALC extensions:

symbol	name		formal semantics
$\leq 1R$ $\geq 2R$	functional number restrictions	(F)	$\{x \in \Delta^\mathcal{I} \mid \#\{y \in \Delta^\mathcal{I} \mid (x,y) \in R^\mathcal{I}\} \leq 1\}$ $\{x \in \Delta^\mathcal{I} \mid \#\{y \in \Delta^\mathcal{I} \mid (x,y) \in R^\mathcal{I}\} \geq 2\}$
$\leq nR$ $\geq nR$	number restrictions, cardinality	(N)	$\{x \in \Delta^\mathcal{I} \mid \#\{y \in \Delta^\mathcal{I} \mid (x,y) \in R^\mathcal{I}\} \leq n\}$ $\{x \in \Delta^\mathcal{I} \mid \#\{y \in \Delta^\mathcal{I} \mid (x,y) \in R^\mathcal{I}\} \geq n\}$
$\leq nR.C$ $\geq nR.C$	qual. cardinality	(Q)	$\{x \in \Delta^\mathcal{I} \mid \#\{y \in C^\mathcal{I} \mid (x,y) \in R^\mathcal{I}\} \leq n\}$ $\{x \in \Delta^\mathcal{I} \mid \#\{y \in C^\mathcal{I} \mid (x,y) \in R^\mathcal{I}\} \geq n\}$
$\{o\}$ $\{o_1, \ldots, o_n\}$	nominal, enumeration, *one-of*	(O)	$\{o^\mathcal{I}\}$ $\{o_1^\mathcal{I}, \ldots, o_n^\mathcal{I}\}$

Obviously, (Q) $\rightarrow$ (N) $\rightarrow$ (F).

Role constructors: Boolean expressions (1/2)

$R \sqcap S$ role conjunction

- role that relates x to y iff both R and S do.
- $R(x, y) \wedge S(x, y)$ in FOL.
- $R^{\mathcal{I}} \cap S^{\mathcal{I}}$ in models.
- example:

  IncestuousPerson$\doteq\exists$IsMarriedTo $\sqcap$ IsSiblingOf

$R \sqcup S$ role disjunction

- role that relates x to y iff R or S does.
- $R(x, y) \vee S(x, y)$ in FOL.
- $R^{\mathcal{I}} \cup S^{\mathcal{I}}$ in models.
- example:

  Parents$\doteq\exists$HasSon $\sqcup$ HasDaughter
  ParentOfSingleChild$\doteq \leq 1$HasSon $\sqcup$ HasDaughter

Role constructors: Boolean expressions (2/2)

$\neg R$ role negation

- role that relates x to y iff R does not.
- $\neg R(x, y)$ in FOL.
- $\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} \setminus R^{\mathcal{I}}$ in models.
- example:

  Philanthropists$\doteq\neg\exists\neg$Likes

Do not confuse role negation with inverse roles (see next slide).

Role constructors: inverse

R^- inverse role

- role that relates x to y exactly if R relates y to x.
- $R(y, x)$ in FOL.
- $\{(y, x) \,|\, (x, y) \in R^{\mathcal{I}}\}$ in models.
- example:

  hasChild$\doteq$hasParent$^-$

Role constructors: composition

$R \circ S$ role composition

- role that relates x to y iff there exists a z s.t. x is related to z by R and z related to y by S.
- $\exists z R(x, z) \wedge S(z, y)$ in FOL.
- $\{(x, y) \in \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} \,|\, \exists z \in \Delta^{\mathcal{I}} : (x, z) \in R^{\mathcal{I}} \wedge (z, y) \in S^{\mathcal{I}}\}$ in models.
- example:

  OnlyChild$\doteq \leq 1$HasParent $\circ$ HasParent^{-}

Role statements: subsroles

$R \sqsubseteq S$ subrole

- whenever two instances are related by R, they are also related by S.
- subroles are used to build role hierarchies
- $\forall x \forall y\, R(x,y) \rightarrow S(x,y)$ in FOL.
- $R^I \subseteq S^I$ in models.
- example:

  isMother $\sqsubseteq$ isParent

$R \doteq S$ role equivalence, abbreviation for $R \sqsubseteq S$ and $S \sqsubseteq R$.

Role statements: transitive roles

$R \in \mathbf{R_+}$ transitive role

- whenever x and y as well as y and z are related by R, then x and z are also related by R directly
- $\forall x \forall y \forall z\, R(x,y) \wedge R(y,z) \rightarrow R(x,z)$ in FOL.
- $R^{\mathcal{I}} = (R^{\mathcal{I}})^+$ in models
  where for any relation $R \subseteq S \times S$ on any set S

$$R^+ := \bigcup_{n \in \mathbb{N}} \text{tr}^n(R)$$
$$\text{tr}(R) := \{(x,y) \mid \exists z \in S : (x,z),(z,y) \in R\}$$

- example:

  isAncestorOf $\in \mathbf{R_+}$
  (Anne, Bert) : isMotherOf
  (Bert, Clara) : isFatherOf
  isFatherOf, isMotherOf $\sqsubseteq$ isAncestorOf
  $\rightsquigarrow$ (Anne, Clara) : isAncestorOf

ALC extensions / role constructors

role constructors of ALC extensions:

symbol	name		formal semantics
R	atomic / primitive role		$R^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$
R^{-}	inverse	(I)	$\{(y,x) \,\vert\, (x,y) \in R^{\mathcal{I}}\}$
$\neg R$	complement, *not*		$\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} \setminus R^{\mathcal{I}}$
$R \sqcap Q$	conjunction, *and*		$R^{\mathcal{I}} \cap Q^{\mathcal{I}}$
$R \sqcup Q$	disjunction, *or*		$R^{\mathcal{I}} \cup Q^{\mathcal{I}}$
$R \circ Q$	composition		$\{(x,y) \in \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} \,\vert\, \exists z \in \Delta^{\mathcal{I}} : (x,z) \in R^{\mathcal{I}} \wedge (z,y) \in$

statements of ALC extensions:

$R \sqsubseteq Q$ $R \doteq Q$	role hierarchy role equality / definition	(H)	$R^{\mathcal{I}} \subseteq Q^{\mathcal{I}}$ $R^{\mathcal{I}} = Q^{\mathcal{I}}$
$R \in \mathbf{R}_{+}$	transitive role	$(_{R^+})$	$R^{\mathcal{I}} = (R^{\mathcal{I}})^{+}$

where for any relation $R \subseteq S \times S$ on any set S

$$R^{+} := \bigcup_{n \in \mathbb{N}} \mathrm{tr}^{n}(R)$$

$$\mathrm{tr}(R) := \{(x,y) \,|\, \exists z \in S : (x,z),(z,y) \in R\}$$

ALC extensions / SH-logics

S := ALC$_{R^+}$

SH = ALCH$_{R^+}$

In SH, number restrictions (Q),(N),(F) may only be applied to **simple roles**, i.e., roles that neither are transitive nor have transitive subroles.

SHIN$^+$ (where N$^+$:= (N) for arbitrary roles) is already undecidable.

ALC extensions / Datatypes

Additional syntactic symbols:

- **datatypes** $d \in D$,
- concrete primitive roles $R \in \mathbf{R_D}$.

Additional model components:

- set $\Delta_D^\mathcal{I}$ as **domain of all datatype values**,
- for each $d \in D$ a set $d^\mathcal{I} \subseteq \Delta_D^\mathcal{I}$ of **datatype values**,
- for each concrete primitive role $R \in \mathbf{R_D}$ a set $R^\mathcal{I} \subseteq \Delta^\mathcal{I} \times \Delta_D^\mathcal{I}$.

symbol	name		formal semantics
R	abstract primitive role $\mathbf{R_A}$		$R^\mathcal{I} \subseteq \Delta^\mathcal{I} \times \Delta^\mathcal{I}$
R	concrete primitive role $\mathbf{R_D}$	(D)	$R^\mathcal{I} \subseteq \Delta^\mathcal{I} \times \Delta_D^\mathcal{I}$
d	datatype		$d^\mathcal{I} \subseteq \Delta_D^\mathcal{I}$
$\neg d$	datatype negation		$\Delta_D^\mathcal{I} \setminus d^\mathcal{I}$

Datatypes may be used wherever concepts are used
(e.g., in exists and value restrictions, number restrictions etc.).

Role hierarchies for abstract and concrete roles are disjunct
(i.e., in $R \subseteq Q$ both R and Q are in $\mathbf{R_A}$ or $\mathbf{R_D}$).

Transitivity can only be declared for abstract roles.

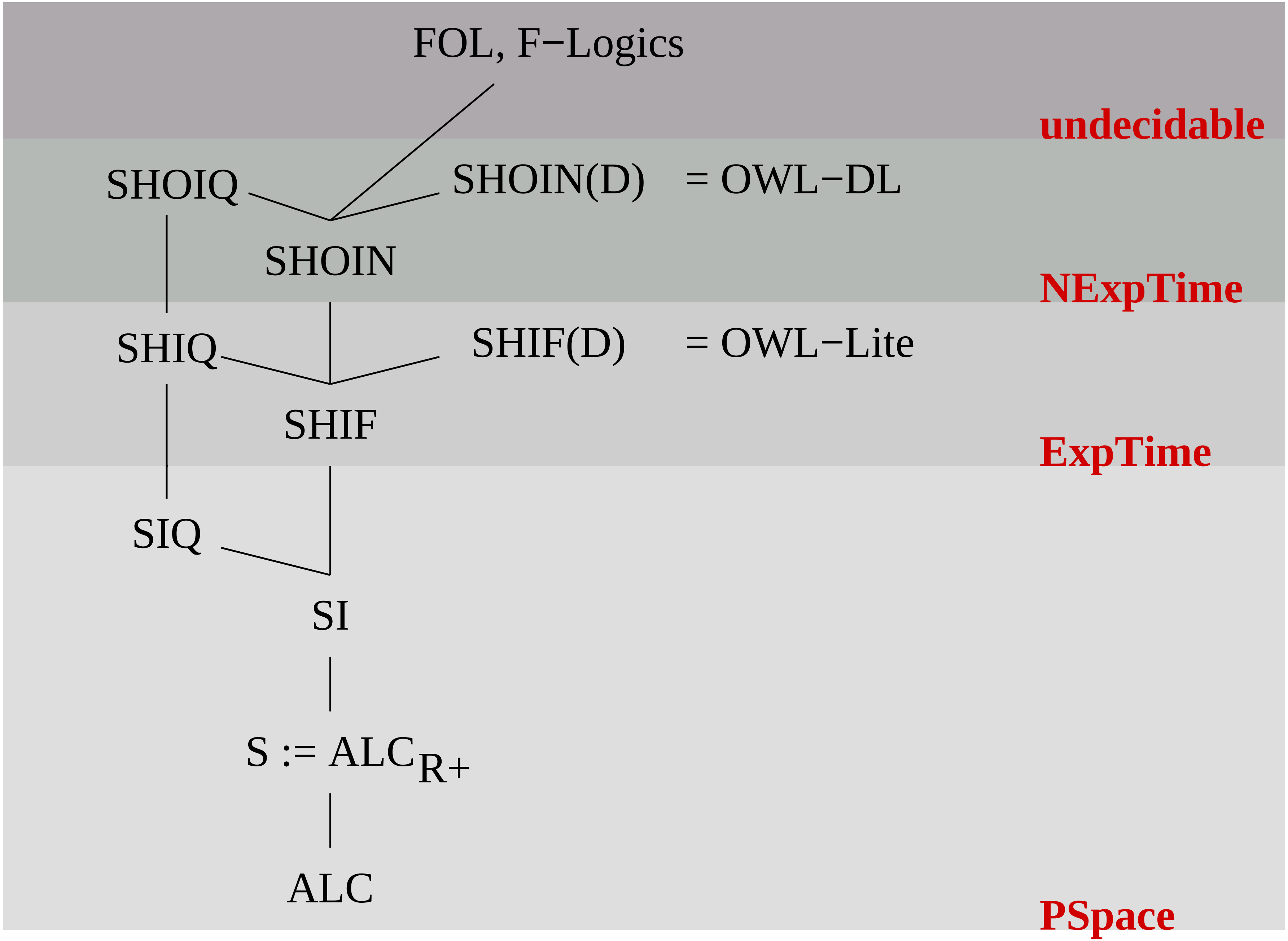

Figure 1: Complexity of satisfiability and subsumption of different description logics.

II. Semantic Web / 2. Web Ontology Language (OWL)

OWL Specification

The OWL specification consists of the following parts:

- OWL Web Ontology Language Overview (REC-2004/02/10),
- OWL Web Ontology Language Guide (REC-2004/02/10) – an example,
- OWL Web Ontology Language Reference (REC-2004/02/10),
- OWL Web Ontology Language Semantics and Abstract Syntax (REC-2004/02/10)
- OWL Web Ontology Language XML Presentation Syntax (REC-2004/02/10)

as well as further documents on test cases and use cases and requirements.

OWL is an RDF vocabulary extension
(i.e., it adds resources with defined formal semantics).

The OWL namespace is

http://www.w3.org/2002/07/owl#

Three Language Levels

- OWL Lite
- OWL DL (Description Logics)
- OWL Full

Each OWL Lite document is an OWL DL document,
each OWL DL document an OWL Full document.

Each OWL Lite inference is an OWL DL inference,
each OWL DL inference an OWL Full inference.

Abstract Syntax (1/2)

Abstract Syntax	DL Syntax	Semantics
Descriptions (C)		
A (URI reference)	A	$A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$
`owl:Thing`	$\top$	$\texttt{owl:Thing}^{\mathcal{I}} = \Delta^{\mathcal{I}}$
`owl:Nothing`	$\bot$	$\texttt{owl:Nothing}^{\mathcal{I}} = \{\}$
`intersectionOf(`$C_1\ C_2\ \ldots$`)`	$C_1 \sqcap C_2$	$(C_1 \sqcap D_1)^{\mathcal{I}} = C_1^{\mathcal{I}} \cap D_2^{\mathcal{I}}$
`unionOf(`$C_1\ C_2\ \ldots$`)`	$C_1 \sqcup C_2$	$(C_1 \sqcup C_2)^{\mathcal{I}} = C_1^{\mathcal{I}} \cup C_2^{\mathcal{I}}$
`complementOf(`C`)`	$\neg C$	$(\neg C)^{\mathcal{I}} = \Delta^{\mathcal{I}} \setminus C^{\mathcal{I}}$
`oneOf(`$o_1\ \ldots$`)`	$\{o_1, \ldots\}$	$\{o_1, \ldots\}^{\mathcal{I}} = \{o_1^{\mathcal{I}}, \ldots\}$
`restriction(`R `someValuesFrom(`C`))`	$\exists R.C$	$(\exists R.C)^{\mathcal{I}} = \{x \mid \exists y.\langle x, y\rangle \in R^{\mathcal{I}} \text{ and } y \in C^{\mathcal{I}}\}$
`restriction(`R `allValuesFrom(`C`))`	$\forall R.C$	$(\forall R.C)^{\mathcal{I}} = \{x \mid \forall y.\langle x, y\rangle \in R^{\mathcal{I}} \rightarrow y \in C^{\mathcal{I}}\}$
`restriction(`R `hasValue(`o`))`	$R : o$	$(\forall R.o)^{\mathcal{I}} = \{x \mid \langle x, o^{\mathcal{I}}\rangle \in R^{\mathcal{I}}\}$
`restriction(`R `minCardinality(`n`))`	$\geqslant n\, R$	$(\geqslant n\, R)^{\mathcal{I}} = \{x \mid \sharp(\{y.\langle x, y\rangle \in R^{\mathcal{I}}\}) \geqslant n\}$
`restriction(`R `minCardinality(`n`))`	$\leqslant n\, R$	$(\geqslant n\, R)^{\mathcal{I}} = \{x \mid \sharp(\{y.\langle x, y\rangle \in R^{\mathcal{I}}\}) \leqslant n\}$
`restriction(`U `someValuesFrom(`D`))`	$\exists U.D$	$(\exists U.D)^{\mathcal{I}} = \{x \mid \exists y.\langle x, y\rangle \in U^{\mathcal{I}} \text{ and } y \in D^{\mathbf{D}}\}$
`restriction(`U `allValuesFrom(`D`))`	$\forall U.D$	$(\forall U.D)^{\mathcal{I}} = \{x \mid \forall y.\langle x, y\rangle \in U^{\mathcal{I}} \rightarrow y \in D^{\mathbf{D}}\}$
`restriction(`U `hasValue(`v`))`	$U : v$	$(U : v)^{\mathcal{I}} = \{x \mid \langle x, v^{\mathcal{I}}\rangle \in U^{\mathcal{I}}\}$
`restriction(`U `minCardinality(`n`))`	$\geqslant n\, U$	$(\geqslant n\, U)^{\mathcal{I}} = \{x \mid \sharp(\{y.\langle x, y\rangle \in U^{\mathcal{I}}\}) \geqslant n\}$
`restriction(`U `maxCardinality(`n`))`	$\leqslant n\, U$	$(\leqslant n\, U)^{\mathcal{I}} = \{x \mid \sharp(\{y.\langle x, y\rangle \in U^{\mathcal{I}}\}) \leqslant n\}$
Data Ranges (D)		
D (URI reference)	D	$D^{\mathbf{D}} \subseteq \Delta_{\mathbf{D}}^{\mathcal{I}}$
`oneOf(`$v_1\ \ldots$`)`	$\{v_1, \ldots\}$	$\{v_1, \ldots\}^{\mathcal{I}} = \{v_1^{\mathcal{I}}, \ldots\}$
Object Properties (R)		
R (URI reference)	R	$R^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$
	R^-	$(R^-)^{\mathcal{I}} = (R^{\mathcal{I}})^-$
Datatype Properties (U)		
U (URI reference)	U	$U^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta_{\mathbf{D}}^{\mathcal{I}}$
Individuals (o)		
o (URI reference)	o	$o^{\mathcal{I}} \in \Delta^{\mathcal{I}}$
Data Values (v)		
v (RDF literal)	v	$v^{\mathcal{I}} = v^{\mathbf{D}}$

Figure 2: Abstract OWL-DL syntax (1/2; Horrocks2003).

Abstract Syntax (2/2)

Abstract Syntax	DL Syntax	Semantics
`Class(`A `partial` C_1 ...C_n`)`	$A \sqsubseteq C_1 \sqcap \ldots \sqcap C_n$	$A^{\mathcal{I}} \subseteq C_1^{\mathcal{I}} \cap \ldots \cap C_n^{\mathcal{I}}$
`Class(`A `complete` C_1 ...C_n`)`	$A = C_1 \sqcap \ldots \sqcap C_n$	$A^{\mathcal{I}} = C_1^{\mathcal{I}} \cap \ldots \cap C_n^{\mathcal{I}}$
`EnumeratedClass(`A o_1 ...o_n`)`	$A = \{o_1, \ldots, o_n\}$	$A^{\mathcal{I}} = \{o_1^{\mathcal{I}}, \ldots, o_n^{\mathcal{I}}\}$
`SubClassOf(`C_1 C_2`)`	$C_1 \sqsubseteq C_2$	$C_1^{\mathcal{I}} \subseteq C_2^{\mathcal{I}}$
`EquivalentClasses(`C_1 ...C_n`)`	$C_1 = \ldots = C_n$	$C_1^{\mathcal{I}} = \ldots = C_n^{\mathcal{I}}$
`DisjointClasses(`C_1 ...C_n`)`	$C_i \sqcap C_j = \bot, i \neq j$	$C_i^{\mathcal{I}} \cap C_j^{\mathcal{I}} \{\}, i \neq j$
`Datatype(`D`)`		$D^{\mathcal{I}} \subseteq \Delta_{\mathbf{D}}^{\mathcal{I}}$
`DatatypeProperty(`U `super(`U_1`)...super(`U_n`)`	$U \sqsubseteq U_i$	$U^{\mathcal{I}} \subseteq U_i^{\mathcal{I}}$
`domain(`C_1`) ...domain(`C_m`)`	$\geqslant 1\, U \sqsubseteq C_i$	$U^{\mathcal{I}} \subseteq C_i^{\mathcal{I}} \times \Delta_{\mathbf{D}}^{\mathcal{I}}$
`range(`D_1`) ...range(`D_l`)`	$\top \sqsubseteq \forall U.D_i$	$U^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times D_i^{\mathcal{I}}$
`[Functional])`	$\top \sqsubseteq \leqslant 1\, U$	$U^{\mathcal{I}}$ `is functional`
`SubPropertyOf(`U_1 U_2`)`	$U_1 \sqsubseteq U_2$	$U_1^{\mathcal{I}} \subseteq U_2^{\mathcal{I}}$
`EquivalentProperties(`U_1 ...U_n`)`	$U_1 = \ldots = U_n$	$U_1^{\mathcal{I}} = \ldots = U_n^{\mathcal{I}}$
`ObjectProperty(`R `super(`R_1`)...super(`R_n`)`	$R \sqsubseteq R_i$	$R^{\mathcal{I}} \subseteq R_i^{\mathcal{I}}$
`domain(`C_1`) ...domain(`C_m`)`	$\geqslant 1\, R \sqsubseteq C_i$	$R^{\mathcal{I}} \subseteq C_i^{\mathcal{I}} \times \Delta^{\mathcal{I}}$
`range(`C_1`) ...range(`C_l`)`	$\top \sqsubseteq \forall R.C_i$	$R^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times C_i^{\mathcal{I}}$
`[inverseOf(`R_0`]`	$R = ({}^{-}R_0)$	$R^{\mathcal{I}} = (R_0^{\mathcal{I}})^{-}$
`[Symmetric]`	$R = ({}^{-}R)$	$R^{\mathcal{I}} = (R^{\mathcal{I}})^{-}$
`[Functional]`	$\top \sqsubseteq \leqslant 1\, R$	$R^{\mathcal{I}}$ `is functional`
`[InverseFunctional]`	$\top \sqsubseteq \leqslant 1\, R^{-}$	$(R^{\mathcal{I}})^{-}$ `is functional`
`[Transitive])`	$Tr(R)$	$R^{\mathcal{I}} = (R^{\mathcal{I}})^{+}$
`SubPropertyOf(`R_1 R_2`)`	$R_1 \sqsubseteq R_2$	$R_1^{\mathcal{I}} \subseteq R_2^{\mathcal{I}}$
`EquivalentProperties(`R_1 ...R_n`)`	$R_1 = \ldots = R_n$	$R_1^{\mathcal{I}} = \ldots = R_n^{\mathcal{I}}$
`AnnotationProperty(`S`)`		
`Individual(`o `type(`C_1`) ...type(`C_n`)`	$o \in C_i$	$o^{\mathcal{I}} \in C_i^{\mathcal{I}}$
`value(`R_1 o_1`)...value(`R_n o_n`)`	$\langle o, o_i \rangle \in R_i$	$\langle o^{\mathcal{I}}, o_i^{\mathcal{I}} \rangle \in R_i^{\mathcal{I}}$
`value(`U_1 v_1`)...value(`U_n v_n`))`	$\langle o, v_i \rangle \in U_i$	$\langle o^{\mathcal{I}}, v_i^{\mathcal{I}} \rangle \in U_i^{\mathcal{I}}$
`SameIndividual(`o_1 ...o_n`)`	$o_1 = \ldots = o_n$	$o_i^{\mathcal{I}} = o_j^{\mathcal{I}}$
`DifferentIndividuals(`o_1 ...o_n`)`	$o_i \neq o_j, i \neq j$	$o_i^{\mathcal{I}} \neq o_j^{\mathcal{I}}, i \neq j$

Figure 3: Abstract OWL-DL syntax (2/2; Horrocks2003).

Example / OWL/abstract DL syntax

```
Class( Human complete )
Class( Employed complete )
ObjectProperty( partnerOf )
ObjectProperty( parentOf )

SubClassOf( Employed Human )

Class( DINK complete
  Employed
  restriction( partnerOf someValuesFrom( Employed ) )
  complementOf( restriction ( parentOf someValuesFrom ( Human ) ) )
  )
```

Figure 4: DINK concept in OWL/abstract DL syntax.

II. Semantic Web / 2. Web Ontology Language (OWL)

OWL Class Hierarchy

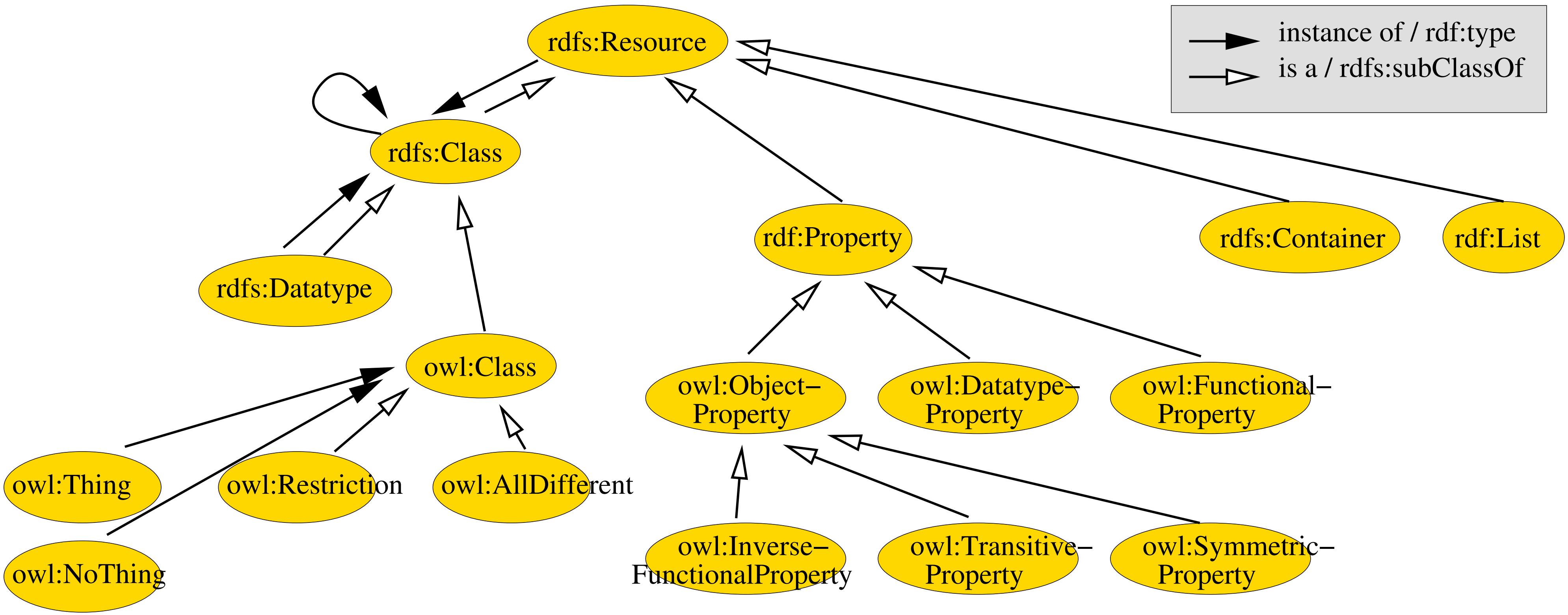

Figure 5: OWL Class Hierarchy.

OWL RDF Vocabulary (1/2)

for class descriptions:

- owl:oneOf (owl:Class, rdf:List),
- owl:onProperty (owl:Restriction, rdf:Property),
  owl:allValuesFrom (owl:Restriction, owl:Class | rdfs:Datatype),
  etc.
- owl:intersectionOf (owl:Class, rdf:List),
  owl:unionOf (owl:Class, rdf:List),
  owl:complementOf (owl:Class, owl:Class),

for class axioms:

- rdfs:subClassOf (owl:Class, owl:Class),
  owl:equivalentClass (owl:Class, owl:Class),
  owl:disjointWith (owl:Class, owl:Class),

OWL RDF Vocabulary (2/2)

for role / property descriptions and axioms:

- `rdfs:subPropertyOf` (`rdf:Property`, `rdf:Property`),
- `rdfs:domain` (`rdf:Property`, `rdfs:Class`),
  `rdfs:range` (`rdf:Property`, `rdfs:Class`),
- `owl:equivalentProperty` (`rdf:Property`, `rdf:Property`)
  `owl:inverseOf` (`owl:ObjectProperty`, `owl:ObjectProperty`),

for individual axioms:

- `owl:sameAs` (Individual, Individual),
  `owl:differentFrom` (Individual, Individual),
- `owl:distinctMembers` (`owl:AllDifferent`, `rdf:List`),

for datatypes:

- `owl:oneOf` (`rdfs:DataRange`, `rdf:List`)

RDF Schema Features:

- Class (Thing, Nothing)
- rdfs:subClassOf
  applied to class names only
- rdf:Property
- rdfs:subPropertyOf
- rdfs:domain
- rdfs:range
- *Individual*

(In)Equality:

- equivalentClass
  applied to class names only
- equivalentProperty
- DL disjointWith
- sameAs
- differentFrom
- AllDifferent, distinctMembers

Boolean Combinations of Class Expressions:

- intersectionOf
  applied to class names only
- DL unionOf
- DL complementOf

Class Axioms:

- DL oneOf, dataRange

Property Characteristics:

- ObjectProperty
- DatatypeProperty
- inverseOf
- TransitiveProperty
- SymmetricProperty
- FunctionalProperty
- InverseFunctionalProperty

Property Restrictions:

- Restriction
- onProperty
- allValuesFrom
- someValuesFrom
- DL hasValue

Cardinality:

- minCardinality
  only 0 or 1
- maxCardinality
  only 0 or 1
- cardinality
  only 0 or 1

(red: completely forbidden in OWL Lite; green: restricted in OWL Lite)

Datatypes

- *xsd datatypes*

Header Information:

- Ontology
- imports

Versioning:

- versionInfo
- priorVersion
- backwardCompatibleWith
- incompatibleWith
- DeprecatedClass
- DeprecatedProperty

Annotation Properties:

- rdfs:label
- rdfs:comment
- rdfs:seeAlso
- rdfs:isDefinedBy
- AnnotationProperty
- OntologyProperty

Example / OWL/RDF/N3

```
@prefix rdf:  <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix : <#> .

:Human rdf:type owl:Class .
:Employed rdf:type owl:Class .
:partnerOf rdf:type owl:ObjectProperty .
:parentOf rdf:type owl:ObjectProperty .
:Employed rdfs:subClassOf :Human .

:DINK rdf:type owl:Class ;
     owl:intersectionOf ( :Employed
                          [ rdf:type owl:Restriction ;
                            owl:onProperty :partnerOf ; owl:someValuesFrom :Employed ]
                          [ owl:complementOf
                            [ rdf:type owl:Restriction ;
                              owl:onProperty :parentOf ; owl:someValuesFrom :Human ]]
                        ) .
```

Figure 6: DINK concept in OWL/RDF/N3.

Example / OWL/RDF/XML

```
<owl:Class rdf:about="#Human"/>
<owl:Class rdf:about="#Employed">
   <rdfs:subClassOf rdf:resource="#Human"/>
</owl:Class>
<owl:ObjectProperty rdf:about="#parentOf"/>
<owl:ObjectProperty rdf:about="#partnerOf"/>
<owl:Class rdf:about="#DINK">
   <owl:intersectionOf rdf:parseType="Collection">
      <Employed/>
      <owl:Restriction>
        <owl:onProperty rdf:resource="#partnerOf"/>
        <owl:someValuesFrom rdf:resource="#Employed"/></owl:Restriction>
      <owl:complementOf>
       <owl:Restriction>
         <owl:onProperty rdf:resource="#parentOf"/>
         <owl:someValuesFrom rdf:resource="#Human"/></owl:Restriction>
      </owl:complementOf>
   </owl:intersectionOf>
</owl:Class>
```

Figure 7: DINK concept in OWL/RDF/XML (excerpt).

II. Semantic Web / 2. Web Ontology Language (OWL)

OWL Syntaxes

- OWL Abstract Syntax (for OWL-DL),
- all kinds of RDF syntaxes:
  - OWL/RDF/N3,
  - OWL/RDF/XML, etc.
- OWL/XML syntax.

OWL/XML

3 schemata, one for each language level:

- OWL Lite (owl1-lite.xsd),
- OWL DL (owl1-dl.xsd), and
- OWL Full (owl1-full.xsd).

namespace

http://www.w3.org/2003/05/owl-xml

(usual prefix owlx)

Please see *OWL Web Ontology Language XML Presentation Syntax (REC-2004/02/*
for syntax details
(not covered here explicitly due to time restrictions).

```
<?xml version="1.0"?>
<owlx:Ontology xmlns:owlx="http://www.w3.org/2003/05/owl-xml"
    name="http://www.cgnm.de/rdf/dink.owl">
  <owlx:Class owlx:name="Human" owlx:complete="true"/>
  <owlx:Class owlx:name="Employed" owlx:complete="false">
    <owlx:Class owlx:name="Human"/>
  </owlx:Class>
  <owlx:ObjectProperty owlx:name="parentOf"/>
  <owlx:ObjectProperty owlx:name="partnerOf"/>
  <owlx:Class owlx:name="DINK">
    <owlx:intersectionOf>
      <owlx:Class name="Employed"/>
      <owlx:ObjectRestriction owlx:property="#partnerOf">
         <owlx:someValuesFrom owlx:class="Employed"/></owlx:ObjectRestriction>
      <owlx:complementOf>
        <owlx:ObjectRestriction owlx:property="#parentOf">
          <owlx:someValuesFrom owlx:class="Human"/></owlx:ObjectRestriction>
      </owlx:complementOf>
    </owlx:intersectionOf>
  </owlx:Class>
</owlx:Ontology>
```

Figure 8: DINK concept in OWL/XML.

OWL 2

OWL 2 is on its way (final recommendation scheduled for October 2009).
OWL 2 will include some features with additional expressivity:

- keys
- property chains
- richer datatypes, data ranges
- qualified cardinality restrictions
- asymmetric, reflexive, and disjoint properties and
- enhanced annotation capabilities

[OWL 2 Web Ontology Language Document Overview, W3C Working Draft 11 June 2009]

References