Ontology Engineering

Lecture 4: The Web Ontology Language OWL 2

Maria Keet

email: `mkeet@cs.uct.ac.za`

home: `http://www.meteck.org`

Department of Computer Science
University of Cape Town, South Africa

Semester 2, Block I, 2019

Outline

Recap previous lectures

- First Order Predicate Logic, syntax, model theoretic-semantics
- Description Logics $\mathcal{ALC}$, syntax, model theoretic-semantics
- Tableau reasoning to check, e.g., satisfiability (exercises with the graph and with vegans and vegetarians)

OWL—yet another logic with another syntax to put up with?!!?

- yes and no.

OWL—yet another logic with another syntax to put up with?!!?

- yes and no.
- No: we consider only the DL-based OWL species, so actually it's just DLs

OWL—yet another logic with another syntax to put up with?!!?

- yes and no.
- No: we consider only the DL-based OWL species, so actually it's just DLs
- Yes; among others:
  - **Serialise** the DL syntax into some flat-text representation for computational processing; e.g. not the symbol "$\exists$" as such in the .owl file, but an "`ObjectSomeValuesFrom`"
  - Some **admin overhead** to manage the flat text files in applications and on the web
  - This family has attributes ('data properties') and data types; most DLs don't

Outline

Toward one ontology language for the Web ('historical' note on SoA around the year 2000)

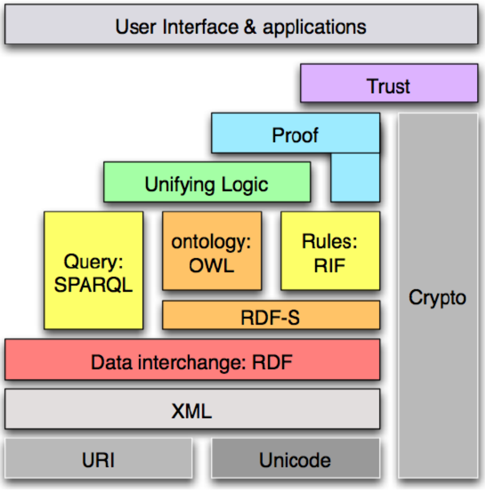
- Plethora of ontology languages used in the 1990s; KIF, KL-ONE, LOOM, F-logic, DAML, OIL, DAML+OIL, ...
- Lack of a lingua franca; hence, ontology interoperation problems even on the syntactic level
- Advances in expressive DL languages and, more importantly, in automated reasoners for expressive DL languages (mainly: FaCT++, then Racer)
- Limitations of RDF(S) as Semantic Web 'ontology language' (we won't discuss this argument)

Toward one ontology language for the Web ('historical' note on SoA around the year 2000)

- Plethora of ontology languages used in the 1990s; KIF, KL-ONE, LOOM, F-logic, DAML, OIL, DAML+OIL, ...
- Lack of a lingua franca; hence, ontology interoperation problems even on the syntactic level
- Advances in expressive DL languages and, more importantly, in automated reasoners for expressive DL languages (mainly: FaCT++, then Racer)
- Limitations of RDF(S) as Semantic Web 'ontology language' (we won't discuss this argument)

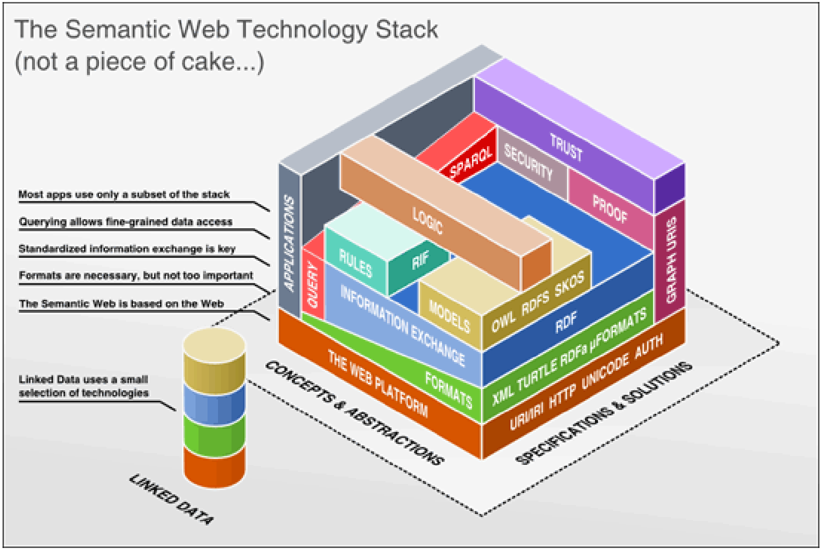
⇒ The Semantic Web

Ontologies on the Web: the (in)famous layer cake



Stack of Languages

- XML
    - Surface syntax, no semantics
- XML Schema
    - Describes structure of XML documents
- RDF
    - Datamodel for “relations” between “things”
- RDF Schema
    - RDF Vocabulary Definition Language
- OWL
    - A more expressive Vocabulary Definition Language


The Semantic Web Technology Stack
(not a piece of cake...)
Most apps use only a subset of the stack
Querying allows fine-grained data access
Standardized information exchange is key
Formats are necessary, but not too important
The Semantic Web is based on the Web
Linked Data uses a small selection of technologies
LINKED DATA
APPLICATIONS
TRUST
SPARQL
SECURITY
PROOF
LOGIC
RULES
RIF
QUERY
INFORMATION EXCHANGE
MODELS
OWL RDFS SKOS
RDF
GRAPH URIS
XML TURTLE RDFa µFORMATS
FORMATS
THE WEB PLATFORM
URI/IRI HTTP UNICODE AUTH
CONCEPTS & ABSTRACTIONS
SPECIFICATIONS & SOLUTIONS

Outline

Outline

Design goals for an ontology language for the Web

- **Shareable**
- **Changing** over time
- **Interoperability**
- **Inconsistency** detection
- Balancing **expressivity and complexity**
- **Ease of use**
- Compatible with **existing standards**
- **Internationalization**

Question does OWL meets these goals?

Requirements for OWL

- Ontologies are **object on the Web**
- with **their own meta-data**, versioning, etc...
- Ontologies are **extendable**
- They contain **classes, properties, data-types, range/domain, individuals**
- **Equality** (for classes, for individuals)
- **Classes as instances**
- **Cardinality** constraints
- **XML** syntax

Question does OWL meets these requirement?

Outline

Species of OWL (historical note)

you may come across these species in the literature, may have to look it up for older OWL ontologies, and is an illustration of languages with more/less features

- OWL Lite
  - Classification hierarchy
  - Simple constraints
- OWL DL
  - Maximal expressiveness
  - While maintaining tractability
  - Standard formalisation in a DL
- OWL Full
  - Very high expressiveness
  - Losing tractability
  - All syntactic freedom of RDF (self-modifying)

Features of OWL languages (historical note)

- OWL Lite
  - (sub)classes, individuals
  - (sub)properties, domain, range
  - conjunction
  - (in)equality
  - (unqualified) cardinality 0/1
  - datatypes
  - inverse, transitive, symmetric properties
  - someValuesFrom
  - allValuesFrom
- OWL DL
  - All of OWL Lite
  - Negation
  - Disjunction
  - (unqualified) Full cardinality
  - Enumerated classes
  - hasValue
- OWL Full
  - Meta-classes
  - Modify language

OWL lite (historical note)

OWL Lite corresponds to the DL $\mathcal{SHIF}(\mathbf{D})$. It has:

- Named classes (A)
- Named properties (P)
- Individuals ($C(o)$)
- Property values ($P(o, a)$)
- Intersection ($C \sqcap D$)
- Union ($C \sqcup D$)
- Negation ($\neg C$)
- Existential value restrictions ($\exists P.C$)
- Universal value restrictions ($\forall P.C$)
- Unqualified (0/1) number restrictions ($\geq nP$, $\leq nP$, $= nP$), $0 \leq n \leq 1$

OWL DL (historical note)

OWL DL corresponds to the DL $\mathcal{SHOIN}(\mathbf{D})$. In addition to all of OWL Lite, it has also:

- Arbitrary number restrictions ($\geq nP$, $\leq nP$, $= nP$), $0 \leq n$
- Property value ($\exists P.\{o\}$)
- Enumeration ($\{o_1, ..., o_n\}$)

Selection of OWL constructs, their DL notation, and an example

OWL Construct	DL	Example
intersectionOf	$C_1 \sqcap \ldots \sqcap C_n$	$Human \sqcap Male$
unionOf	$C_1 \sqcup \ldots \sqcup C_n$	$Doctor \sqcup Lawyer$
complementOf	$\neg C$	$\neg Male$
oneOf	$\{o_1, \ldots, o_n\}$	$\{giselle, juan\}$
allValuesFrom	$\forall P.C$	$\forall hasChild.Doctor$
someValuesFrom	$\exists P.C$	$\exists hasChild.Lawyer$
value	$\exists P.\{o\}$	$\exists citizenOf.\{RSA\}$
minCardinality	$\geq nP$	$\geq 2 hasChild$
maxCardinality	$\leq nP$	$\leq 1 hasChild$

+ XML Schema datatypes: int, string, real, etc...

(summarised from the standard)

Selection of OWL axioms, their DL notation, and an example

OWL Axiom	DL	Example
SubClassOf	$C_1 \sqsubseteq C_2$	$Human \sqsubseteq Animal \sqcap Biped$
EquivalentClasses	$C_1 \equiv ... \equiv C_n$	$Man \equiv Human \sqcap Male$
SubPropertyOf	$P_1 \sqsubseteq P_2$	$hasDaughter \sqsubseteq hasChild$
EquivalentProperties	$P_1 \equiv ... \equiv P_n$	$cost \equiv price$
SameIndividual	$o_1 = ... = o_n$	$President\_Zuma = J\_Zuma$
DisjointClasses	$C_i \sqsubseteq \neg C_j$	$Male \sqsubseteq \neg Female$
DifferentIndividuals	$o_i \neq o_j$	$sally \neq shereen$
inverseOf	$P_1 \equiv P_2^-$	$hasChild \equiv hasParent^-$
Transitive	$P^+ \sqsubseteq P$	$ancestor^+ \sqsubseteq ancestor$
Symmetric	$P \equiv P^-$	$connectedTo \equiv connectedTo^-$

Outline

Outline

Trials and Trade-offs...

- OWL was, at the time, the best trade-off on language features and performance (and politics of the standardisation process); Early adopters:
  - trying out modelling with OWL: bio(medical) domain
  - trying to use the Semantic Web technologies: experiences with tool building
- Some issues encountered

Trials and Trade-offs...

- OWL was, at the time, the best trade-off on language features and performance (and politics of the standardisation process); Early adopters:
  - trying out modelling with OWL: bio(medical) domain
  - trying to use the Semantic Web technologies: experiences with tool building
- Some issues encountered
- Limited expressiveness of OWL, but features that modellers felt they needed; e.g.:
  - Qualified cardinality restrictions; e.g., can't represent $\texttt{Bicycle} \sqsubseteq \geq 2\ \texttt{hasComponent.Wheel}$
  - Relational properties (no reflexivity, irreflexivity)

Trials and Trade-offs...

- OWL was, at the time, the best trade-off on language features and performance (and politics of the standardisation process); Early adopters:
  - trying out modelling with OWL: bio(medical) domain
  - trying to use the Semantic Web technologies: experiences with tool building
- Some issues encountered
- Limited expressiveness of OWL, but features that modellers felt they needed; e.g.:
  - Qualified cardinality restrictions; e.g., can't represent $\texttt{Bicycle} \sqsubseteq \geq 2\ \texttt{hasComponent.Wheel}$
  - Relational properties (no reflexivity, irreflexivity)
- practical things when building ontologies: annotations, imports, versioning, species validation (see p315 of the paper)

Trials and Trade-offs...

- OWL was, at the time, the best trade-off on language features and performance (and politics of the standardisation process); Early adopters:
  - trying out modelling with OWL: bio(medical) domain
  - trying to use the Semantic Web technologies: experiences with tool building
- Some issues encountered
- Limited expressiveness of OWL, but features that modellers felt they needed; e.g.:
  - Qualified cardinality restrictions; e.g., can't represent $\texttt{Bicycle} \sqsubseteq \geq 2\ \texttt{hasComponent.Wheel}$
  - Relational properties (no reflexivity, irreflexivity)
- practical things when building ontologies: annotations, imports, versioning, species validation (see p315 of the paper)
- Syntax issues that made building tools somewhat cumbersome

Syntax problems (historical note)

- Having both frame-based legacy (Abstract syntax) and axioms (DL) was deemed confusing
- Type of ontology entity. e.g.,

  ```
  Class(A partial
     restriction(hasB someValuesFrom(C))
  ```

  - `hasB` is data property and `C` a datatype?
  - `hasB` an object property and `C` a class?

  OWL-DL has a strict separation of the vocabulary, but the specification does not precisely specify how to enforce this separation at the syntactic level

Aims of OWL 2

- Address as much as possible of the identified problems (previous slides and “the next steps for OWL 2” paper)
- Cater for specific usage scenarios of ontologies that emerged since OWL standardisation

Aims of OWL 2

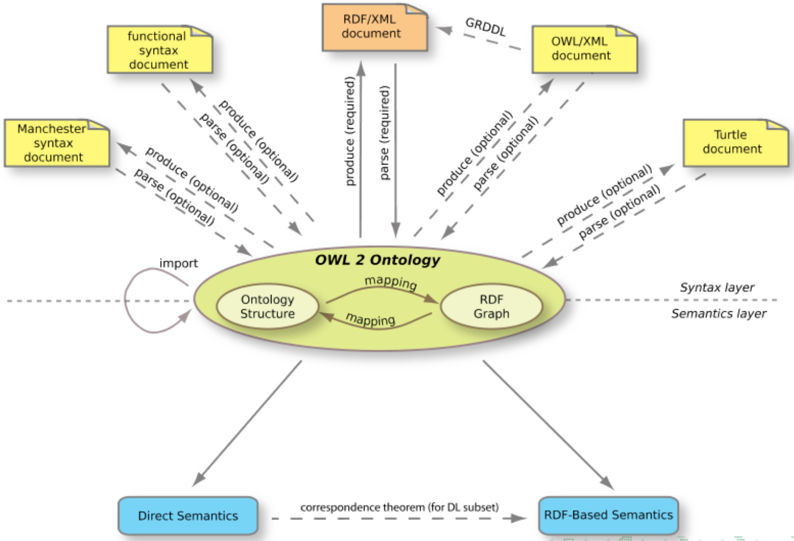
- Address as much as possible of the identified problems (previous slides and "the next steps for OWL 2" paper)
- Cater for specific usage scenarios of ontologies that emerged since OWL standardisation

Task Compare this with the possible "future extensions" of the "the making of an ontology language" paper

Some general points

- OWL 2 a W3C recommendation since 27-10-2009
- Any OWL 2 ontology can also be viewed as an RDF graph (The relationship between these two views is specified by the Mapping to RDF Graphs document)
- Direct, i.e. model-theoretic, semantics ($\Rightarrow$ OWL 2 DL) and an RDF-based semantics ($\Rightarrow$ OWL 2 full)
- Primary exchange syntax for OWL 2 is RDF/XML, others are optional
- Three profiles, which are sub-languages of OWL 2 (syntactic restrictions)

The Structure of OWL 2



A note on syntaxes of OWL

- RDF/XML
  - Official exchange syntax
  - Hard for humans to read (and RDF parsers are hard to write)
- OWL/XML
  - Not the RDF syntax
  - Still hard for humans, but more XML than RDF tools available
- Abstract syntax
  - To some, considered human readable
- “User-usable” ones
  - e.g., Manchester syntax, informal and limited matching with UML, pseudo-NL verbalisations (mainly in English, some in Greek, Latvian, isiZulu, Afrikaans)

⇒ “RDF/XML” is the *required* exchange format (all tools are expected to be able to process it); all the others are optional (tools need not be able to process it)

Outline

Overview

- Based on $\mathcal{SROIQ}(D)$, which is N2ExpTime-complete
- Has all the language features of its DL-based predecessors
- And more features (next slide) [i.e.: more expressive than OWL-DL]
- Other extras:
  - Fancier metamodelling and annotations
  - Improved ontology publishing, imports and versioning control
- Variety of syntaxes, RDF serialization (but no RDF-style semantics)

New features for properties

- Reflexive (local and global) & irreflexive, asymmetric
- Property chains (`ObjectPropertyChain`), e.g.:
  $contains \circ hasPart \sqsubseteq contains$
  $hasMother \circ hasSister \sqsubseteq hasAunt$

```
   SubObjectPropertyOf( ObjectPropertyChain(
     a:hasMother a:hasSister ) a:hasAunt )
```

New features for properties

- Reflexive (local and global) & irreflexive, asymmetric
- Property chains (`ObjectPropertyChain`), e.g.:
  $contains \circ hasPart \sqsubseteq contains$
  $hasMother \circ hasSister \sqsubseteq hasAunt$
  ```
     SubObjectPropertyOf( ObjectPropertyChain(
       a:hasMother a:hasSister ) a:hasAunt )
  ```
- BEWARE ObjectMinCardinality, ObjectMaxCardinality, ObjectExactCardinality, ObjectHasSelf, FunctionalObjectProperty, InverseFunctionalObjectProperty, IrreflexiveObjectProperty, AsymmetricObjectProperty, and DisjointObjectProperties **only on simple object properties** (i.e., has no direct or indirect subproperties that are either transitive or are defined by means of property chains)

The language: other extensions

- Qualified cardinality restrictions

The language: other extensions

- Qualified cardinality restrictions
- The `Haskey` 'key' that are **not** keys like in databases
  - Alike inverse functional only (i.e., merely 1:n instead of 1:1) but applicable only to individuals that are explicitly named in an ontology
  - No unique name assumption, hence inferences are different from that expected of keys in databases
  - "relevant mainly for query answering" [Cuenca Grau et al, 2008, p316], which does not go well with OWL 2 DL in non-toy applications anyway

The language: other extensions

- Qualified cardinality restrictions
- The `Haskey` 'key' that are **not** keys like in databases
  - Alike inverse functional only (i.e., merely 1:n instead of 1:1) but applicable only to individuals that are explicitly named in an ontology
  - No unique name assumption, hence inferences are different from that expected of keys in databases
  - "relevant mainly for query answering" [Cuenca Grau et al, 2008, p316], which does not go well with OWL 2 DL in non-toy applications anyway
- Richer datatypes, data ranges; e.g., `DatatypeRestriction( xsd:integer xsd:minInclusive "5"^^xsd:integer xsd:maxExclusive "10"^^xsd:integer)`

OWL 2 DL and DLs—semantics of those features

- (In addition to those of OWL-DL/$\mathcal{SHOIN}$)
- qualified cardinality restrictions, $\geq nR.C$ and $\leq nR.C$, semantics:
  - $(\geq n\, R.C)^{\mathcal{I}} = \{x \mid \sharp\{y \mid (x,y) \in R^{\mathcal{I}} \cap y \in C^{\mathcal{I}}\} \geq n\}$
  - $(\leq n\, R.C)^{\mathcal{I}} = \{x \mid \sharp\{y \mid (x,y) \in R^{\mathcal{I}} \cap y \in C^{\mathcal{I}}\} \leq n\}$

OWL 2 DL and DLs—semantics of those features

- (In addition to those of OWL-DL/$\mathcal{SHOIN}$)
- qualified cardinality restrictions, $\geq nR.C$ and $\leq nR.C$, semantics:
    - $(\geq n\, R.C)^{\mathcal{I}} = \{x \mid \sharp\{y \mid (x,y) \in R^{\mathcal{I}} \cap y \in C^{\mathcal{I}}\} \geq n\}$
    - $(\leq n\, R.C)^{\mathcal{I}} = \{x \mid \sharp\{y \mid (x,y) \in R^{\mathcal{I}} \cap y \in C^{\mathcal{I}}\} \leq n\}$
- Properties of roles:
    - Reflexive: $Ref(R)$, with semantics: $\forall x : x \in \Delta^{\mathcal{I}}$ implies $(x,x) \in (R)^{\mathcal{I}}$
    - Irreflexive: $Irr(R)$, with semantics: $\forall x : x \in \Delta^{\mathcal{I}}$ implies $(x,x) \notin (R)^{\mathcal{I}}$
    - Asymmetric: Asym(R), with semantics: $\forall x, y : (x,y) \in (R)^{\mathcal{I}}$ implies $(y,x) \notin (R)^{\mathcal{I}}$

OWL 2 DL and DLs—semantics of those features

- (In addition to those of OWL-DL/$\mathcal{SHOIN}$)
- qualified cardinality restrictions, $\geq nR.C$ and $\leq nR.C$, semantics:
  - $(\geq n\, R.C)^{\mathcal{I}} = \{x \mid \sharp\{y \mid (x,y) \in R^{\mathcal{I}} \cap y \in C^{\mathcal{I}}\} \geq n\}$
  - $(\leq n\, R.C)^{\mathcal{I}} = \{x \mid \sharp\{y \mid (x,y) \in R^{\mathcal{I}} \cap y \in C^{\mathcal{I}}\} \leq n\}$
- Properties of roles:
  - Reflexive: $Ref(R)$, with semantics: $\forall x : x \in \Delta^{\mathcal{I}}$ implies $(x,x) \in (R)^{\mathcal{I}}$
  - Irreflexive: $Irr(R)$, with semantics: $\forall x : x \in \Delta^{\mathcal{I}}$ implies $(x,x) \notin (R)^{\mathcal{I}}$
  - Asymmetric: Asym(R), with semantics: $\forall x, y : (x,y) \in (R)^{\mathcal{I}}$ implies $(y,x) \notin (R)^{\mathcal{I}}$
- *Limited* role chaining: $R \circ S \sqsubseteq R$, with semantics: $\forall y_1, \ldots, y_4 : (y_1, y_2) \in (R)^{\mathcal{I}}$ and $(y_3, y_4) \in (S)^{\mathcal{I}}$ imply $(y_1, y_4) \in (R)^{\mathcal{I}}$, and regularity restriction (strict linear order $<$ on the properties)

Definition ((Regular) Role Inclusion Axioms (HorrocksEtAl06))

Let $\prec$ be a regular order on roles. A **role inclusion axiom** (RIA for short) is an expression of the form $w \sqsubseteq R$, where w is a finite string of roles not including the universal role U, and $R \neq U$ is a role name. A **role hierarchy** $\mathcal{R}_h$ is a finite set of RIAs. An interpretation $\mathcal{I}$ **satisfies** a role inclusion axiom $w \sqsubseteq R$, written $\mathcal{I} \models w \sqsubseteq R$, if $w^{\mathcal{I}} \subseteq R^{\mathcal{I}}$. An interpretation is a **model** of a role hierarchy $\mathcal{R}_h$ if it satisfies all RIAs in $\mathcal{R}_h$, written $\mathcal{I} \models \mathcal{R}_h$. A RIA $w \sqsubseteq R$ is $\prec$-**regular** if R is a role name, and

- $w = RR$, or
- $w = R^-$, or
- $w = S_1 ... S_n$ and $S_i \prec R$, for all $1 \geq i \geq n$, or
- $w = RS_1 ... S_n$ and $S_i \prec R$, for all $1 \geq i \geq n$, or
- $w = S_1 ... S_n R$ and $S_i \prec R$, for all $1 \geq i \geq n$.

Finally, a role hierarchy $\mathcal{R}_h$ is **regular** if there exists a regular order $\prec$ such that each RIA in $\mathcal{R}_h$ is $\prec$-regular.

Outline

Rationale

- Computational considerations
  - Consult “OWL profiles” page *Table 10. Complexity of the Profiles*
- Robustness of implementations w.r.t. *scalable* applications
- Already enjoy a user base

Outline

OWL 2 EL Overview

- Intended for large 'simple' ontologies
- Focussed on type-level knowledge (TBox)
- Better computational behaviour than OWL 2 DL (polynomial vs. exponential/open)
- Based on the DL language $\mathcal{EL}^{++}$ (PTime complete)
- Reasoner: e.g. CEL `http://code.google.com/p/cel/`

Supported class restrictions

- existential quantification to a class expression or a data range
- existential quantification to an individual or a literal
- self-restriction
- enumerations involving a single individual or a single literal
- intersection of classes and data ranges

Supported axioms, restricted to allowed set of class expressions

- class inclusion, equivalence, disjointness
- object property inclusion and data property inclusion
- property equivalence
- transitive object properties
- reflexive object properties
- domain and range restrictions
- assertions
- functional data properties
- keys
- In short: $\sqcap \; \exists \; \top \; \bot \sqsubseteq \sqcap \; \exists \; \top \; \bot$

NOT supported in OWL 2 EL

- universal quantification to a class expression or a data range
- cardinality restrictions
- disjunction
- class negation
- enumerations involving more than one individual
- disjoint properties
- irreflexive, symmetric, and asymmetric object properties
- inverse object properties, functional and inverse-functional object properties

Outline

OWL 2 QL Overview

- Query answering over a large amount of instances with same kind of performance as relational databases
- Expressive features cover several used features of UML Class diagrams and ER models
- Based on *DL-Lite*$_{\mathcal{R}}$ (more is possible with UNA and in some implementations)
- Used for Ontology-Based Data Access, integration, management (commonly know as OBDA)

Supported Axioms in OWL 2 QL, restrictions

- Subclass expressions restrictions:
  - a class
  - existential quantification (ObjectSomeValuesFrom) where the class is limited to owl:Thing
  - existential quantification to a data range (DataSomeValuesFrom)
- Super expressions restrictions:
  - a class
  - intersection (ObjectIntersectionOf)
  - negation (ObjectComplementOf)
  - existential quantification to a class (ObjectSomeValuesFrom)
  - existential quantification to a data range (DataSomeValuesFrom)

Supported Axioms in OWL 2QL

- Restrictions on class expressions, object and data properties occurring in functionality assertions cannot be specialized
- subclass axioms
- class expression equivalence (involving subClassExpression), disjointness
- inverse object properties
- property inclusion (not involving property chains and SubDataPropertyOf)
- property equivalence
- property domain and range
- disjoint properties
- symmetric, reflexive, irreflexive, asymmetric properties
- assertions other than individual equality assertions and negative property assertions (DifferentIndividuals, ClassAssertion, ObjectPropertyAssertion, and DataPropertyAssertion)

NOT supported in OWL 2 QL

- existential quantification to a class expression or a data range in the subclass position
- self-restriction
- existential quantification to an individual or a literal
- enumeration of individuals and literals
- universal quantification to a class expression or a data range
- cardinality restrictions
- disjunction
- property inclusions involving property chains
- functional and inverse-functional properties
- transitive properties
- keys
- individual equality assertions and negative property assertions

Outline

OWL 2 RL Overview

- Development motivated by: what fraction of OWL 2 DL can be expressed by rules (with equality)?
- Scalable reasoning in the context of RDF(S) application
- Rule-based technologies (forward chaining rule system, over *instances*)
- Inspired by Description Logic Programs and pD*
- Reasoning in PTime

Supported in OWL 2 RL

- More restrictions on class expressions (see table 2, e.g. no SomeValuesFrom on the right-hand side of a subclass axiom)
- All axioms in OWL 2 RL are constrained in a way that is compliant with the restrictions in Table 2.
- Thus, OWL 2 RL supports all axioms of OWL 2 apart from disjoint unions of classes and reflexive object property axioms.
- No $\forall$ and $\neg$ on lhs, and $\exists$ and $\sqcup$ on rhs of $\sqsubseteq$

Partial table of features (1/2)

Language ⇒	OWL 1		OWL 2	OWL 2 Profiles		
Feature ⇓	**Lite**	**DL**	**DL**	EL	QL	RL
Role hierarchy	+	+	+	.	+	.
N-ary roles (where $n \geq 2$)	−	−	−	.	?	.
Role chaining	−	−	+	.	−	.
Role acyclicity	−	−	−	.	−	.
Symmetry	+	+	+	.	+	.
Role values	−	−	−	.	−	.
Qualified number restrictions	−	−	+	.	−	.
One-of, enumerated classes	?	+	+	.	−	.
Functional dependency	+	+	+	.	?	.
Covering constraint over concepts	?	+	+	.	−	.
Complement of concepts	?	+	+	.	+	.
Complement of roles	−	−	+	.	+	.
Concept identification	−	−	−	.	−	.
Range typing	−	+	+	.	+	.
Reflexivity	−	−	+	.	−	.
Antisymmetry	−	−	−	.	−	.
Transitivity	+	+	+	.	−	.
Asymmetry	?	?	+	−	+	+
Irreflexivity	−	−	+	.	−	.
.	.	.	.	.	.	.

Partial table of features (2/2)

Exercise Checking the previous slides and the standard, verify the question marks in the table (tentatively all "–") and fill in the dots (any "±" should be qualified at to what the restriction is)

Explore the OWL species classifier, accessible via the book's website at
`https://people.cs.uct.ac.za/~mkeet/OEbook/`

- Load an ontology, e.g., AWO v1 and determine its 'species'
- What do the letters stand for?
- Why is the AWO not in any of the OWL 2 profiles?

Outline

Speculation about future extensions

- Several directions for extensions proposed

Speculation about future extensions

- Several directions for extensions proposed
  - The ‘leftover’ from OWL 1’s “Future extensions” (UNA, CWA, defaults), parthood relation
  - Syntactic sugar: ‘macros’, ‘n-aries’
  - Integration with rules: RIF, DL-safe rules, SBVR
  - Orthogonal dimensions: temporal, fuzzy, rough, probabilistic
  - Better support for multilingual ontologies

Speculation about future extensions

- Several directions for extensions proposed
  - The ‘leftover’ from OWL 1’s “Future extensions” (UNA, CWA, defaults), parthood relation
  - Syntactic sugar: ‘macros’, ‘n-aries’
  - Integration with rules: RIF, DL-safe rules, SBVR
  - Orthogonal dimensions: temporal, fuzzy, rough, probabilistic
  - Better support for multilingual ontologies
- Doesn’t seem likely to even get started any time soon
- Extend OWL (or one of its DLs) yourself: see Ch10 of the textbook

Beyond OWL

- Some features will never be in any DL-based OWL species, if we want to keep the language decidable
- Then what?

Beyond OWL

- Some features will never be in any DL-based OWL species, if we want to keep the language decidable
- Then what?
- There are several alternatives; e.g.,
  - Use FOL in its entirety (e.g., Common Logic, or another one with implementations [e.g., Prover9&Mace]), or even a higher order logic (HOL)
  - Orchestrate the axioms into modules and push only the ‘violating’ axioms into a more expressive language; e.g., with the Distributed Ontology Model and Specification Language (DOL) `http://www.omg.org/spec/DOL/`

DOL example: adding some axioms beyond OWL

logic of the theory

1. Takes t6 represented in OWL

new ontology name

```
logic CASL.FOL
ontology theory6_plus_antisym_and_WS =
theory6 with translation OWL22CASL
then
forall x,y:Thing . P(x,y) /\ P(y,x) => x =y %%(t3)
forall x,y:Thing . not P(y,x) =>
      exists z:Thing . P(z,y) /\ not O(z,x) %%(t4)
```

2. translate that into FOL

3. add, a.o., antisymmetry (t3) to t6

Outline

Reasoning services for DL-based OWL ontologies

- OWL ontology is a first-order logical theory $\Rightarrow$ verifying the formal properties of the ontology corresponds to reasoning over a first-order theory

Reasoning services for DL-based OWL ontologies

- OWL ontology is a first-order logical theory $\Rightarrow$ verifying the formal properties of the ontology corresponds to reasoning over a first-order theory
- Main ('standard') reasoning tasks for the OWL ontologies:
  - consistency of the ontology
  - class [concept] (and object property [role]) consistency
  - class [concept] (and object property [role]) subsumption
  - instance checking
  - instance retrieval
  - query answering

Reasoning services for DL-based OWL ontologies

- Consistency of the ontology
  - Is the ontology $K = (T, A)$ consistent (non-selfcontradictory), i.e., is there at least a model for K?
- Class (and object property) consistency
  - is there a model of T in which C (resp. R) has a nonempty extension?
- Class (and object property) subsumption
  - i.e., is the extension of C (resp. R) contained in the extension of D in every model of T?
- Instance checking
  - is a a member of class C in K, i.e., is the fact $C(a)$ satisfied by every interpretation of K?
- Instance retrieval
  - find all members of C in K, i.e., compute all individuals a s.t. $C(a)$ is satisfied by every interpretation of K
- Query answering
  - compute all tuples of individuals t s.t. query $q(t)$ is entailed by K, i.e., $q(t)$ is satisfied by every interpretation of K

Reasoning services for DL-based OWL ontologies

- Standard reasoning services in a non-standard way: e.g., possible world explorer, test-driven development, object property suggestion, entailment diffs

Reasoning services for DL-based OWL ontologies

- Standard reasoning services in a non-standard way: e.g., possible world explorer, test-driven development, object property suggestion, entailment diffs
- Non-standard reasoning services: e.g., explanation/justifications, repair, least common subsumer

Reasoning services for DL-based OWL ontologies

- Standard reasoning services in a non-standard way: e.g., possible world explorer, test-driven development, object property suggestion, entailment diffs
- Non-standard reasoning services: e.g., explanation/justifications, repair, least common subsumer
- Not all OWL species are equally suitable for all reasoning tasks (why not?)

Note: Reasoning with OWA (vs. CWA)

- **O**pen **W**orld **A**ssumption

- **C**losed **W**orld **A**ssumption

Note: Reasoning with OWA (vs. CWA)

- **O**pen **W**orld **A**ssumption
  - Absence of information is interpreted as unknown information
- **C**losed **W**orld **A**ssumption
  - Absence of information is interpreted as negative information

Note: Reasoning with OWA (vs. CWA)

- **O**pen **W**orld **A**ssumption
  - Absence of information is interpreted as unknown information
  - Assumes incomplete information
- **C**losed **W**orld **A**ssumption
  - Absence of information is interpreted as negative information
  - Assumes we have complete information

Note: Reasoning with OWA (vs. CWA)

- **O**pen **W**orld **A**ssumption
  - Absence of information is interpreted as unknown information
  - Assumes incomplete information
  - Good for describing knowledge in a way that is extensible
- **C**losed **W**orld **A**ssumption
  - Absence of information is interpreted as negative information
  - Assumes we have complete information
  - Good for constraining information and validating data in an application

Example

Which alumni do not have a PhD?

Alumnus	Degree Obtained
Delani	PhD in history
Sally	PhD in politics
Peter	MSc in Informatics
Dalila	PhD in politics

Example

Which alumni do not have a PhD?

Alumnus	**Degree Obtained**
Delani	PhD in history
Sally	PhD in politics
Peter	MSc in Informatics
Dalila	PhD in politics

- Query under CWA says "Peter"
- Query under OWA cannot say "Peter", because we do not know if Peter also obtained a PhD. To retrieve "Peter" we have add an axiom somehow stating that Peter does not have a PhD (e.g., by being an instance of *PhD student*, declaring the degrees to be disjoint & covering, ...).

Automated reasoning examples

- Subsumption reasoning, like in the exercise ($\mathcal{T} \vdash Vegan \sqsubseteq Vegetarian$)
- Example with Schrödinger's cat: see slides 23-43 in `SWModLang-ESSLLI09-2.pdf`
- Example with the `sampleClassification.owl`
- Exercise with instance classification and KB consistency (and OWA)
- Exercise with finding the errors in a 'dirty' ontology

Summary