Limitations

OWL 2

OWL 2 profiles

Reasoning

Summary

Outline

1. Limitations
2. OWL 2
   - OWL 2 DL
3. OWL 2 profiles
   - OWL 2 EL
   - OWL 2 QL
   - OWL 2 RL
4. Reasoning

EXPRESSIVITY LIMITATIONS

- Qualified cardinality restrictions (e.g., no Bicycle ⊑ ≥ 2 hasComponent.Wheel)
- Relational properties (no reflexivity, irreflexivity)
- Data types, missing
  - restrictions to a subset of datatype values (ranges)
  - relationships between values of data properties on one object
  - relationships between values of data properties on different objects
  - aggregation functions
- Other things like annotations, imports, versioning, species validation (see p315 of the paper)

SYNTAX PROBLEMS

- 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.
More syntax problems

- RDF’s triple notation, difficult to read and process
- OWL 1 provides mapping from the Abstract Syntax into OWL RDF, but not the converse:
  - an RDF graph \( G \) is an OWL-DL ontology if there exists an ontology \( \mathcal{O} \) in Abstract Syntax s.t. the result of the normative transformation of \( \mathcal{O} \) into triples is precisely \( G \), which makes checking whether \( G \) is an OWL-DL ontology very hard in practice:
  - examine all ‘relevant’ ontologies \( \mathcal{O} \) in abstract syntax, check whether the normative transformation of \( \mathcal{O} \) into RDF yields precisely \( G \).

Problems with the semantics

- RDF’s blank nodes, but unnamed individuals not directly available in \( SHOIN(D) \)
- Frames and axioms

Aims

- Address as much as possible of the identified problems (previous slides and “the next steps for OWL 2” paper)
- 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-’09
- 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 (⇒ OWL 2 DL) and an RDF-based semantics (⇒ 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

Overview

- Based on $SROIQ(D)$, which is 2NExpTime-complete
- More expressive than OWL-DL
- Fancier metamodeling and annotations
- Improved ontology publishing, imports and versioning control
- Variety of syntaxes, RDF serialization (but no RDF-style semantics)

Limitations

OWL 2

- OWL 2 profiles
- Reasoning
- Summary

Limitations

OWL 2 DL

- The language: properties of properties
  - property chains (ObjectPropertyChain), e.g.:
    - SubObjectPropertyOf(ObjectPropertyChain(a:hasMother a:hasSister) a:hasAunt)
      with having Grace as the mother of Stewie, and Carol a sister of Grace, the ontology entails that Stewie has Carol as aunt
    - or, e.g.: contains $\circ$ hasPart $\sqsubseteq$ contains
  - 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 Haskey 'key' that are **not** keys like in conceptual models and 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

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

**Exercise:** verify the question marks in the table (tentatively all “–”) and fill in the dots (any “±” should be qualified at to what the restriction is)
### OWL 2 EL Overview

- Intended for large 'simple' ontologies
- Focused 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/](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:** $\top \sqcap \exists \bot \sqsubseteq \sqcap \exists \top \bot$

### 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
- Query answering over a large amount of instances with same kind of performance as relational databases (Ontology-Based Data Access)
- Expressive features cover several used features of UML Class diagrams and ER models ('COnceptual MOdel-based Data Access')
- Based on DL-LiteR (more is possible with UNA and in some implementations)

**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
### 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 $\sqsubseteq$

### Another section on speculation about future extensions

- The ‘leftover’ from OWL 1’s “Future extensions” (UNA, CWA, defaults), parthood relation (primarily: antisymmetry, restrictions on current usage of properties)
- New “future of OWL”, a.o.:
  - Syntactic sugar: ‘macros’, ‘n-aries’
  - Query languages: EQL-lite and nRQL w.r.t. SPARQL
  - Integration with rules: RIF, DL-safe rules, SBVR
  - Orthogonal dimensions: temporal, fuzzy, rough, probabilistic

### 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
  - concept (and role) consistency
  - concept (and role) subsumption
  - instance checking
  - instance retrieval
  - query answering
- Non-standard reasoning services, such as explanation, repair, least common subsumer, ...
- Note: Not all OWL languages are equally suitable for all these reasoning tasks
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 $K$?
- Concept (and role) consistency
  - Is there a model of $T$ in which $C$ (resp. $R$) has a nonempty extension?
- Concept (and role) 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 concept $C$ 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 $K$.

Note: Reasoning with OWA (vs. CWA)

- **Open World Assumption**
  - Absence of information is interpreted as unknown information
  - Assumes incomplete information
  - Good for describing knowledge in a way that is extensible
- **Closed World Assumption**
  - 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?

<table>
<thead>
<tr>
<th>Alumnus</th>
<th>Degree Obtained</th>
</tr>
</thead>
<tbody>
<tr>
<td>Delani</td>
<td>PhD in history</td>
</tr>
<tr>
<td>Sally</td>
<td>PhD in politics</td>
</tr>
<tr>
<td>Peter</td>
<td>MSc in Informatics</td>
</tr>
<tr>
<td>Dalila</td>
<td>PhD in politics</td>
</tr>
</tbody>
</table>

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 \((T \vdash \text{Vegan} \sqsubseteq \text{Vegetarian})\)
- Example with Schrödinger’s cat
- Example with the sampleClassification.owl
- Exercise with instance classification and KB consistency (and OWA)
- Exercise with finding the errors in a ‘dirty’ ontology

**Summary**

1. **Limitations**
2. **OWL 2**
   - OWL 2 DL
3. **OWL 2 profiles**
   - OWL 2 EL
   - OWL 2 QL
   - OWL 2 RL
4. **Reasoning**