Ontology Reasoning:
Why do We Want It?
Why Ontology Reasoning?

- Given key role of ontologies in many applications, it is essential to provide tools and services to help users:
  - Design and maintain high quality ontologies, e.g.:
    - Meaningful — all named classes can have instances
    - Correct — captured intuitions of domain experts
    - Minimally redundant — no unintended synonyms
    - Richly axiomatised — (sufficiently) detailed descriptions
  - Answer queries over ontology classes and instances, e.g.:
    - Find more general/specific classes
    - Retrieve individuals/tuples matching a given query
  - Integrate and align multiple ontologies
Why Decidable Reasoning?

- OWL is an W3C standard DL based ontology language
  - OWL constructors/axioms restricted so reasoning is decidable
- Consistent with Semantic Web's layered architecture
  - XML provides syntax transport layer
  - RDF(S) provides basic relational language and simple ontological primitives
  - OWL provides powerful but still decidable ontology language
  - Further layers (e.g. SWRL) will extend OWL
    - Will almost certainly be undecidable
- W3C requirement for “implementation experience”
  - “Practical” decision procedures
  - Several implemented systems
  - Evidence of empirical tractability
Why Correct Reasoning?

• Need to have high level of confidence in reasoner
  – Most interesting/useful inferences are those that were unexpected
  – Likely to be ignored/dismissed if reasoner known to be unreliable

• Many realistic web applications will be agent ↔ agent
  – No human intervention to spot glitches in reasoning
Combining the strengths of UMIST and The Victoria University of Manchester

Ontology Reasoning: How do we do it?
Use a (Description) Logic

• OWL DL based on $\mathbf{VK} \mathbf{LT}$ Description Logic
  – In fact it is equivalent to $\mathbf{VK} \mathbf{R} \mathbf{LQ} \mathbf{+G}_q$, DL

• OWL DL Benefits from many years of DL research
  – Well defined semantics
  – Formal properties well understood (complexity, decidability)
  – Known reasoning algorithms
  – Implemented systems (highly optimised)

• Three “species” of OWL
  – OWL full is union of OWL syntax and RDF - no reasoning support
  – OWL DL equivalent to $\mathbf{VK} \mathbf{R} \mathbf{LQ} \mathbf{+G}_q$
    • Partial support from existing DL reasoners FaCT, Racer, Pellet, etc.
    • New tableaux algorithm for $\mathbf{VK} \mathbf{R} \mathbf{LT}$ now being implemented in FaCT++
  – OWL Lite equivalent to $\mathbf{VK} \mathbf{L} \mathbf{I} \mathbf{+G}_q$
    • Full support from existing DL reasoners
# Class/Concept Constructors

<table>
<thead>
<tr>
<th>Constructor</th>
<th>DL Syntax</th>
<th>Example</th>
<th>FOL Syntax</th>
</tr>
</thead>
<tbody>
<tr>
<td>intersectionOf</td>
<td>( C_1 \cap \ldots \cap C_n )</td>
<td>Human ( \cap ) Male</td>
<td>( C_1(x) \land \ldots \land C_n(x) )</td>
</tr>
<tr>
<td>unionOf</td>
<td>( C_1 \cup \ldots \cup C_n )</td>
<td>Doctor ( \cup ) Lawyer</td>
<td>( C_1(x) \lor \ldots \lor C_n(x) )</td>
</tr>
<tr>
<td>complementOf</td>
<td>( \neg C )</td>
<td>( \neg ) Male</td>
<td>( \neg C(x) )</td>
</tr>
<tr>
<td>oneOf</td>
<td>( {x_1} \cup \ldots \cup {x_n} )</td>
<td>{john} \cup {mary}</td>
<td>( x = x_1 \lor \ldots \lor x = x_n )</td>
</tr>
<tr>
<td>allValuesFrom</td>
<td>( \forall P.C )</td>
<td>( \forall ) hasChild.Doctor</td>
<td>( \forall y.P(x,y) \rightarrow C(y) )</td>
</tr>
<tr>
<td>someValuesFrom</td>
<td>( \exists P.C )</td>
<td>( \exists ) hasChild.Lawyer</td>
<td>( \exists y.P(x,y) \land C(y) )</td>
</tr>
<tr>
<td>maxCardinality</td>
<td>( \leq nP )</td>
<td>( \leq ) 1 hasChild</td>
<td>( \exists y.P(x,y) )</td>
</tr>
<tr>
<td>minCardinality</td>
<td>( \geq nP )</td>
<td>( \geq ) 2 hasChild</td>
<td>( \exists y.P(x,y) )</td>
</tr>
</tbody>
</table>

- \( \{ \) is a concept (class); \( \_i \) is a role (property); \( m \) is an individual name
- XMLS **datatypes** as well as classes in \( \_i \_j \) and \( \_i \_j \)
  - Restricted form of DL **concrete domains** (this is the \( \_G_\_q \), part of \( \_V \_K \_L \_Q \_G_\_q \))
Abstract Syntax

E.g., Person x ;hasChild.(Doctor w <hasChild.Doctor):

intersectionOf(
  restriction(hasChild allValuesFrom(
    unionOf(Doctor
      restriction(hasChild someValuesFrom(Doctor))))))
RDFS Syntax

E.g., Person $x \; \text{hasChild}(\text{Doctor} \; w \; \text{hasChild}.\text{Doctor})$:

```xml
<owl:Class>
  <owl:intersectionOf rdf:parseType="collection">
    <owl:Class rdf:about="#Person"/>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#hasChild"/>
      <owl:allValuesFrom>
        <owl:unionOf rdf:parseType="collection">
          <owl:Class rdf:about="#Doctor"/>
          <owl:Restriction>
            <owl:onProperty rdf:resource="#hasChild"/>
            <owl:someValuesFrom rdf:resource="#Doctor"/>
          </owl:Restriction>
        </owl:unionOf>
      </owl:allValuesFrom>
    </owl:Restriction>
  </owl:intersectionOf>
</owl:Class>
```
Ontologies / Knowledge Bases

- **OWL ontology** equivalent to a DL Knowledge Base
- OWL ontology consists of a set of axioms and facts
  - Note: an ontology is usually thought of as containing only Tbox axioms (schema)---OWL is non-standard in this respect
- Recall that a DL KB $\mathcal{N}$ is a pair $\mathcal{W}, \mathcal{D}$ where
  - $\mathcal{W}$ is a set of “terminological” axioms (the Tbox)
  - $\mathcal{D}$ is a set of “assertional” axioms (the Abox)
Ontology/Tbox Axioms

<table>
<thead>
<tr>
<th>OWL Syntax</th>
<th>DL Syntax</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>subClassOf</td>
<td>$C_1 \sqsubseteq C_2$</td>
<td>Human $\sqsubseteq$ Animal $\sqcap$ Biped</td>
</tr>
<tr>
<td>equivalentClass</td>
<td>$C_1 \equiv C_2$</td>
<td>Man $\equiv$ Human $\sqcap$ Male</td>
</tr>
<tr>
<td>subPropertyOf</td>
<td>$P_1 \sqsubseteq P_2$</td>
<td>hasDaughter $\sqsubseteq$ hasChild</td>
</tr>
<tr>
<td>equivalentProperty</td>
<td>$P_1 \equiv P_2$</td>
<td>cost $\equiv$ price</td>
</tr>
<tr>
<td>transitiveProperty</td>
<td>$P^+ \sqsubseteq P$</td>
<td>ancestor$^+ \sqsubseteq$ ancestor</td>
</tr>
</tbody>
</table>

- Obvious FO/Modal Logic equivalences
  - E.g., DL: $\{ \forall y \subset G \}$ FOL: $\forall M \models G \sqsubseteq M, \exists G \sqsubseteq M$, ML: $\{ \$ G \}$

- Often distinguish two kinds of Tbox axioms
  - “Definitions” $\{ y \subset G$ or $\$ G$ where $\}$ is a concept name
  - General Concept Inclusion axioms (GCIs) where $\}$ may be complex
Abstract Syntax

E.g., Doctor ⊑ Person:

\[
\text{SubClasOf}(\text{Human} \\
\quad \text{intersectionOf}(\text{Animal Biped}))
\]

\[
\text{Class}(\text{Human partial} \\
\quad \text{intersectionOf}(\text{Animal Biped}))
\]

E.g., ProudParent ⊑ Person x ;\text{hasChild}.(Doctor w <\text{hasChild}.\text{Doctor}):

\[
\text{Class}(\text{ProudParent complete} \\
\quad \text{intersectionOf}(
\quad \text{restriction}(\text{hasChild allValuesFrom}(
\quad \text{unionOf}(\text{Doctor }
\quad \text{restriction}(\text{hasChild someValuesFrom(Doctor})))))
\quad )))
\]
# Ontology Facts / Abox Axioms

<table>
<thead>
<tr>
<th>OWL Syntax</th>
<th>DL Syntax</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>type</td>
<td>$a : C$</td>
<td>John : Happy-Father</td>
</tr>
<tr>
<td>property</td>
<td>$\langle a, b \rangle : R$</td>
<td>$\langle John, Mary \rangle :$ has-child</td>
</tr>
</tbody>
</table>

- **Note:** using **nominals** (e.g., in $VKR^LQ$), we can reduce Abox axioms to **concept inclusion axioms**

\[
\mu a : C \text{ equivalent to } \{a\} \sqsubseteq C \\
\langle a, b \rangle : R \text{ equivalent to } \{a\} \sqsubseteq \exists R.\{b\}
\]
Abstract Syntax

E.g., John:HappyFather:

Individual(John type(HappyFather))

E.g., <John,Mary>:hasChild:

Individual(John value(hasChild Mary))
Basic Inference Tasks

- Knowledge is **correct** (captures intuitions)
  - Does C subsume D w.r.t. ontology R? ($\{ \}^L \supset G^L$ in *every model* $L$ of $R$)

- Knowledge is **minimally redundant** (no unintended synonyms)
  - Is C equivalent to D w.r.t. R? ($\{ \}^L = G^L$ in *every model* $L$ of $R$)

- Knowledge is **meaningful** (classes can have instances)
  - Is C is satisfiable w.r.t. R? ($\{ \}^L \neq >$ in *some model* $L$ of $R$)

- **Querying** knowledge
  - Is $M$ an instance of } w.r.t. R? ($\{ }^L \supset 5$ } in *every model* $L$ of $R$)
  - Is $M$, $>1$ an instance of $<$ w.r.t. R? ($\{ }^L \supset 5$ $<$ in *every model* $L$ of $R$)

- Above problems can be solved using **highly optimised** DL reasoners
System Demonstration (Protégé)

Combining the strengths of UMIST and The Victoria University of Manchester