Scalable Ontology-Based Information Systems

Ian Horrocks
<ian.horrocks@comlab.ox.ac.uk>
Information Systems Group
Oxford University Computing Laboratory

What is an Ontology?

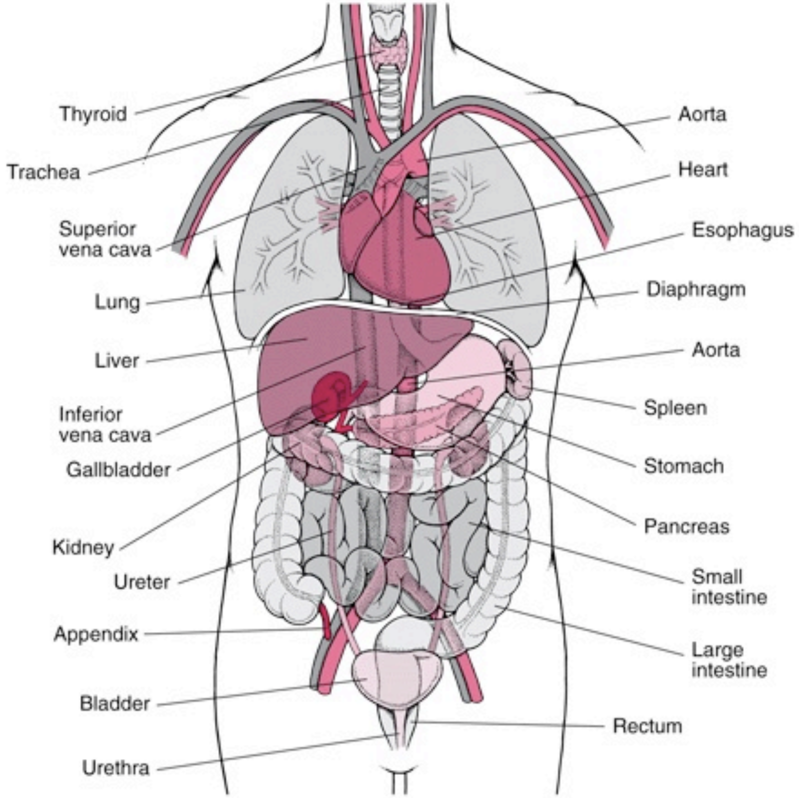
What is an Ontology?

A model of (some aspect of) the world

What is an Ontology?

A model of (some aspect of) the world

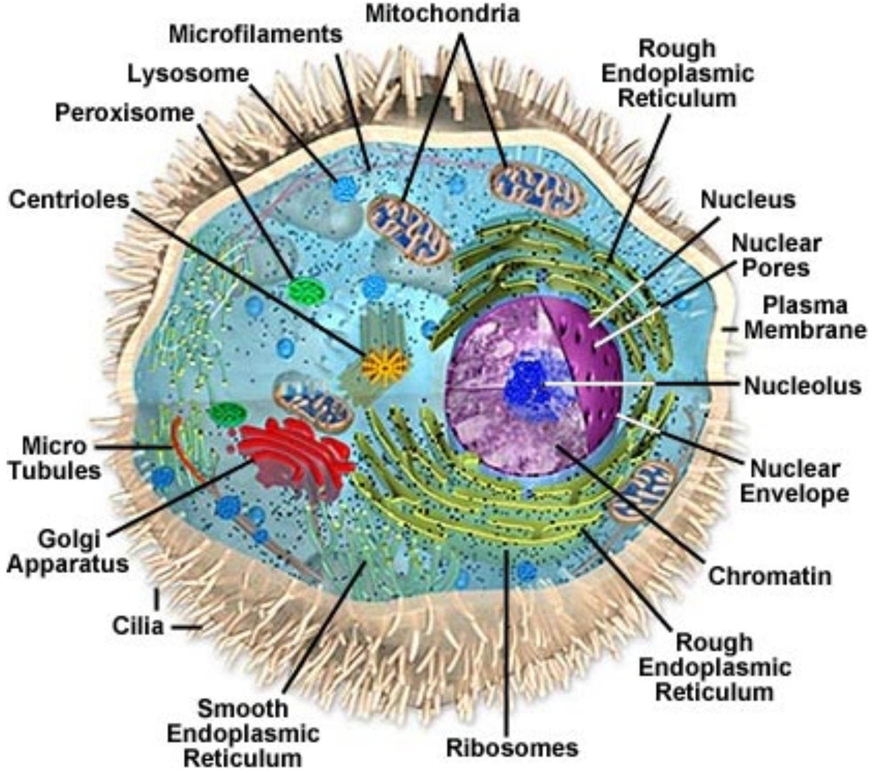
- Introduces **vocabulary** relevant to domain, e.g.:
  - Anatomy



What is an Ontology?

A model of (some aspect of) the world

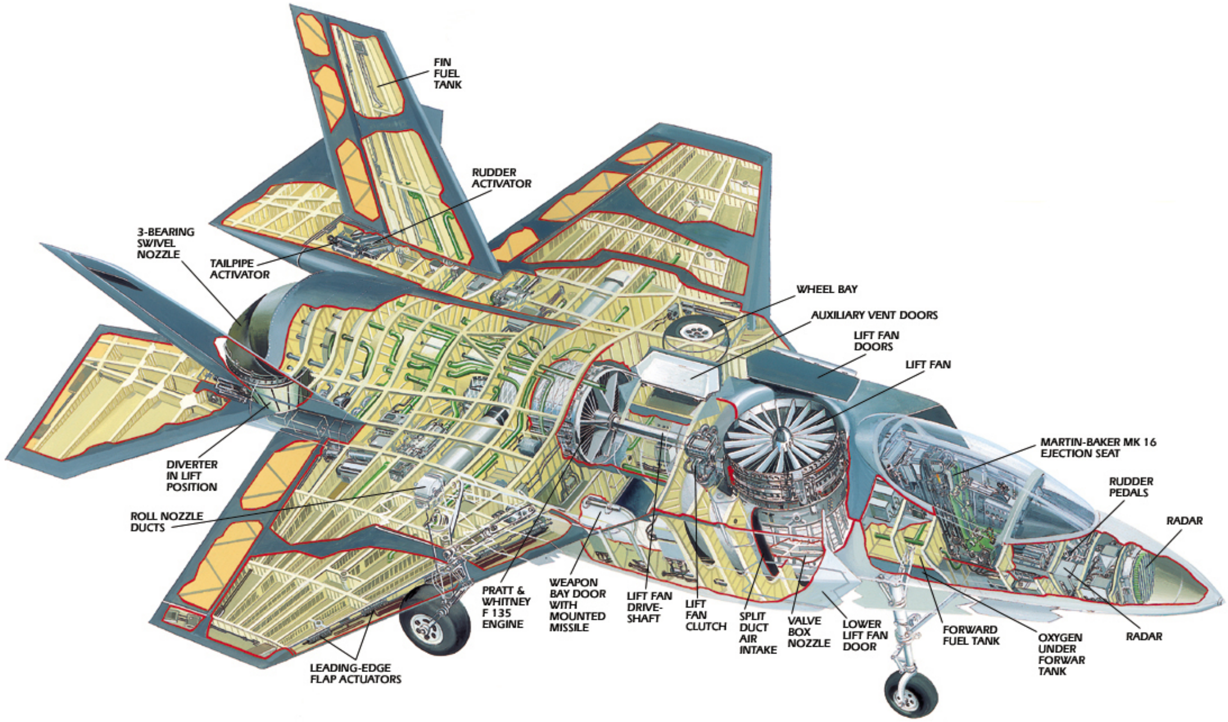
- Introduces **vocabulary** relevant to domain, e.g.:
  - Anatomy
  - Cellular biology



What is an Ontology?

A model of (some aspect of) the world

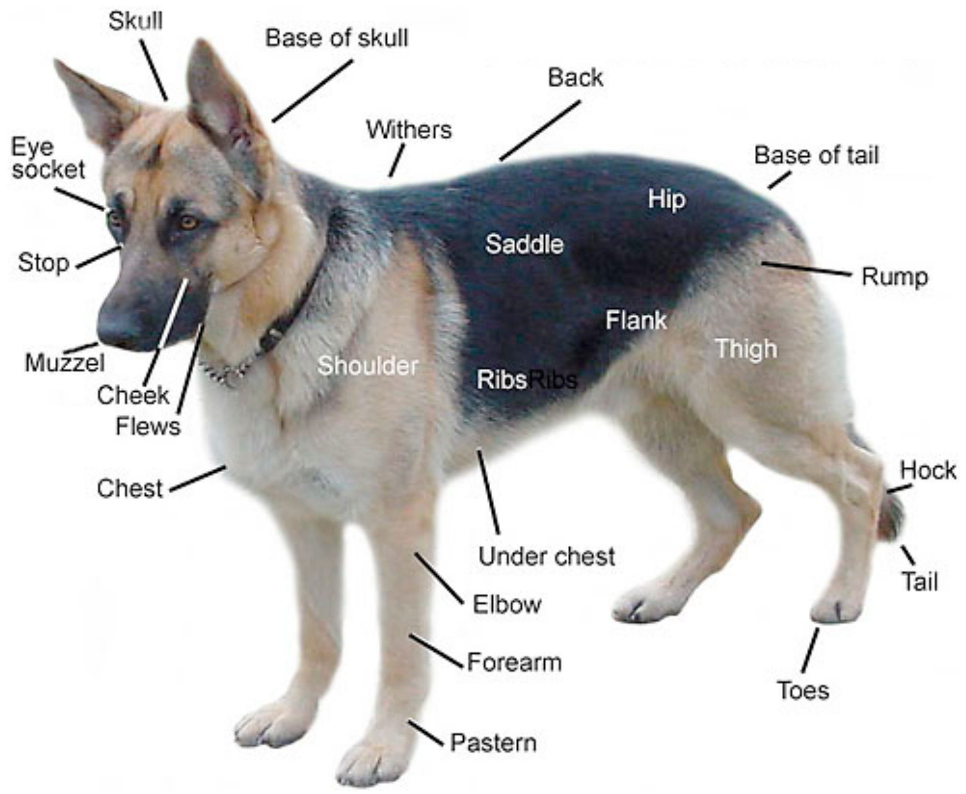
- Introduces **vocabulary** relevant to domain, e.g.:
  - Anatomy
  - Cellular biology
  - Aerospace



What is an Ontology?

A model of (some aspect of) the world

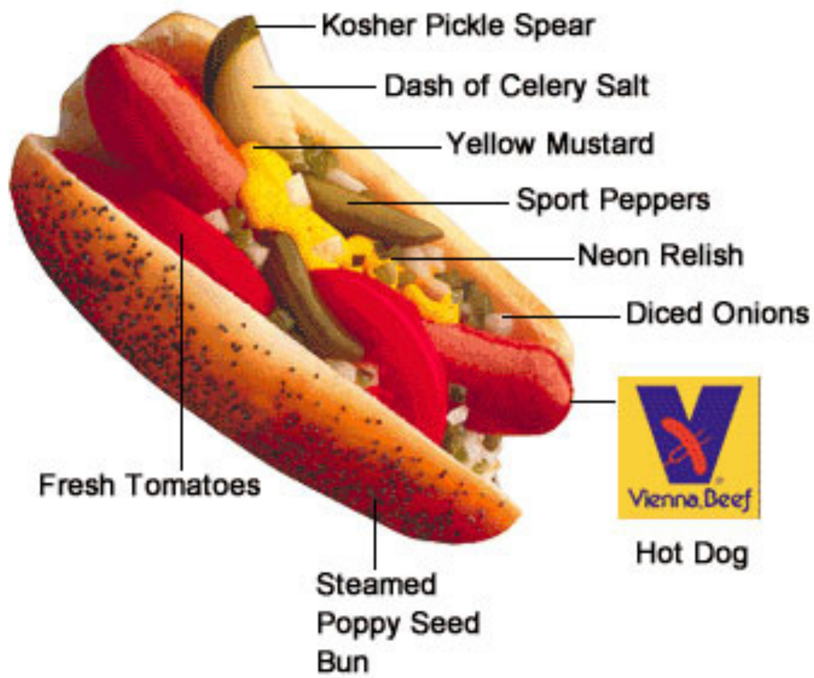
- Introduces **vocabulary** relevant to domain, e.g.:
  - Anatomy
  - Cellular biology
  - Aerospace
  - Dogs



What is an Ontology?

A model of (some aspect of) the world

- Introduces **vocabulary** relevant to domain, e.g.:
  - Anatomy
  - Cellular biology
  - Aerospace
  - Dogs
  - Hotdogs
  - …

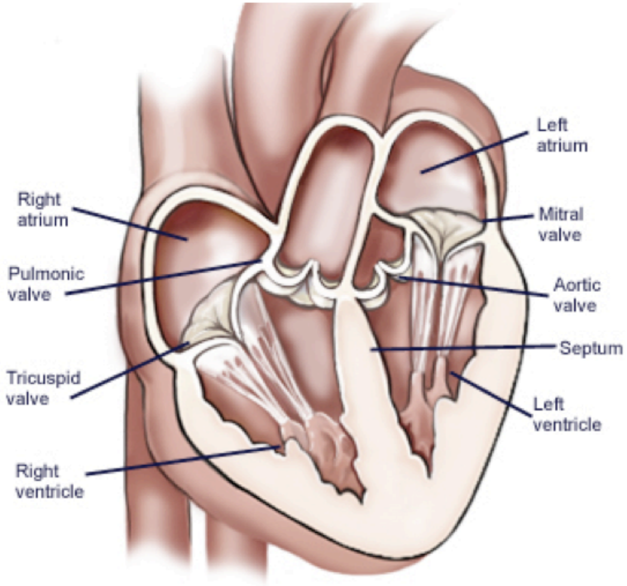


What is an Ontology?

A model of (some aspect of) the world

- Introduces **vocabulary** relevant to domain
- Specifies **meaning** (semantics) of terms

Heart is a muscular organ that
is part of the circulatory system



What is an Ontology?

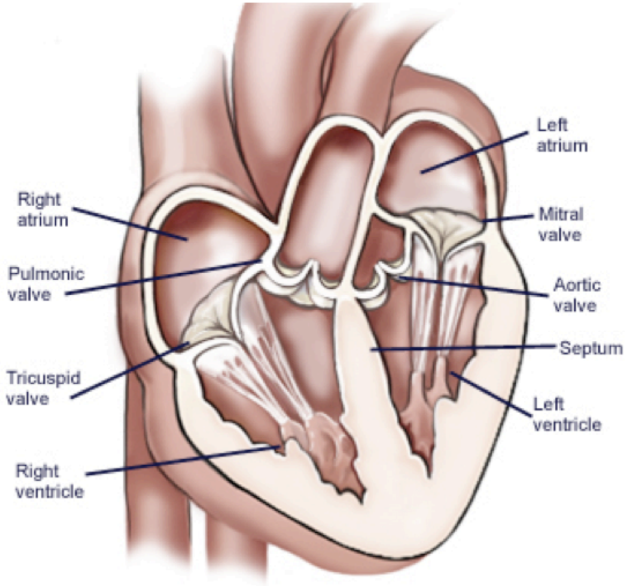
A model of (some aspect of) the world

- Introduces **vocabulary** relevant to domain
- Specifies **meaning** (semantics) of terms

  Heart is a muscular organ that is part of the circulatory system

- **Formalised** using suitable logic

$$\forall x.[\mathsf{Heart}(x) \rightarrow \mathsf{MuscularOrgan}(x) \wedge \exists y.[\mathsf{isPartOf}(x,y) \wedge \mathsf{CirculatorySystem}(y)]]$$



Web Ontology Language OWL (2)

- W3C **recommendation(s)**
- Motivated by **Semantic Web** activity

  Add meaning to web content by annotating it with terms defined in ontologies

- Supported by **tools and infrastructure**
  - APIs (e.g., OWL API, Thea, OWLink)
  - Development environments (e.g., Protégé, Swoop, TopBraid Composer)
  - Reasoners & Information Systems (e.g., Pellet, Racer, HermiT, Quonto, …)

- Based on a **Description Logics** ($\mathcal{SHOIN}$ / $\mathcal{SROIQ}$)

Description Logics (DLs)

- Fragments of **first order logic** designed for KR
- Desirable computational properties
  - **Decidable** (essential)
  - Low complexity (desirable)
- Succinct and **variable free syntax**

$$\forall x.[\mathsf{Heart}(x) \rightarrow \mathsf{MuscularOrgan}(x) \wedge \exists y.[\mathsf{isPartOf}(x,y) \wedge \mathsf{CirculatorySystem}(y)]]$$

$$\mathsf{Heart} \sqsubseteq \mathsf{MuscularOrgan} \sqcap \exists \mathsf{isPartOf}.\mathsf{CirculatorySystem}$$

Description Logics (DLs)

DL **Knowledge Base** (KB) consists of two parts:

– Ontology (aka **TBox**) axioms define terminology (schema)

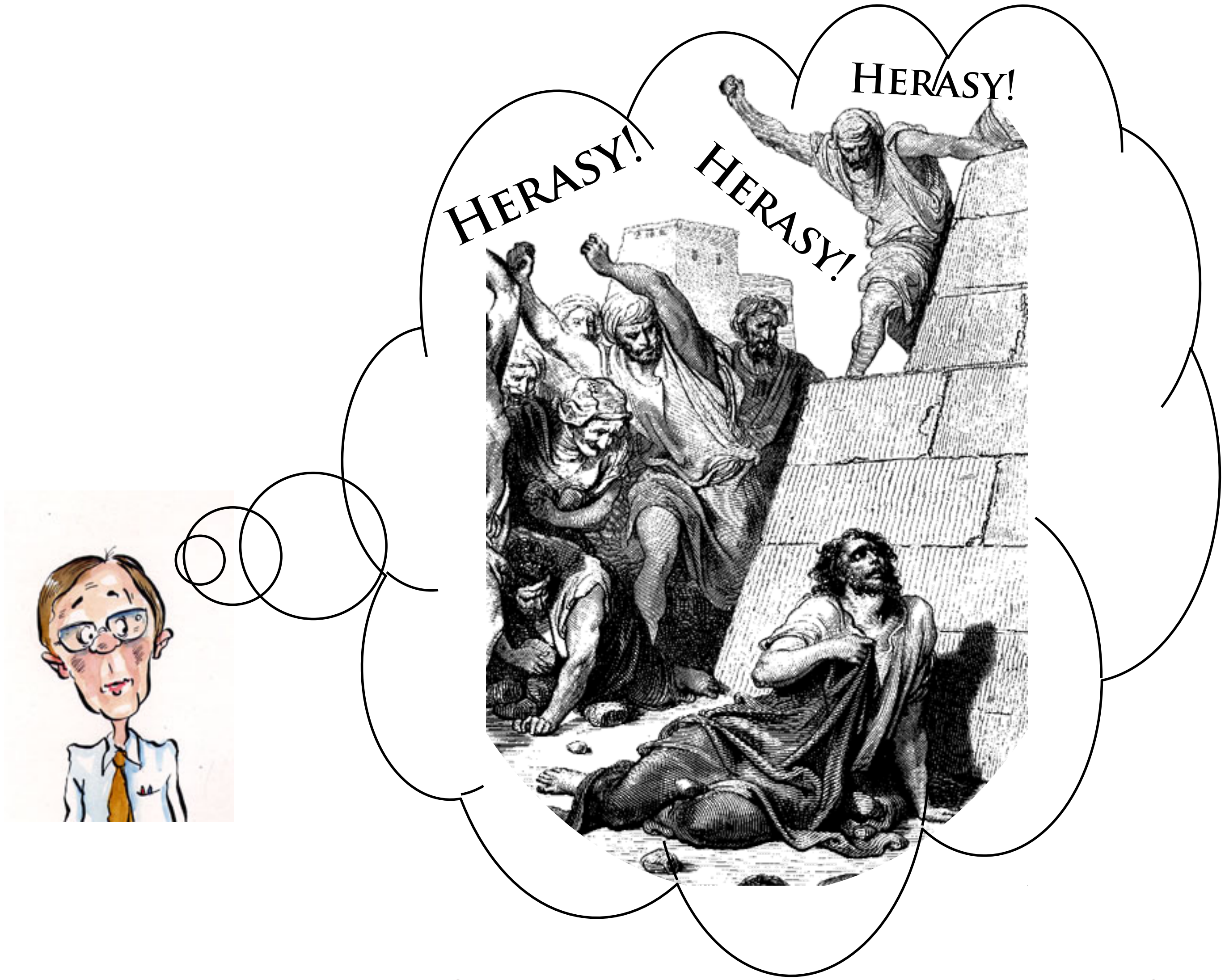
$$
\begin{aligned}
\mathsf{Heart} &\sqsubseteq \mathsf{MuscularOrgan} \sqcap \\
&\quad \exists \mathsf{isPartOf}.\mathsf{CirculatorySystem} \\
\mathsf{HeartDisease} &\equiv \mathsf{Disease} \sqcap \\
&\quad \exists \mathsf{affects}.\mathsf{Heart} \\
\mathsf{VascularDisease} &\equiv \mathsf{Disease} \sqcap \\
&\quad \exists \mathsf{affects}.(\exists \mathsf{isPartOf}.\mathsf{CirculatorySystem})
\end{aligned}
$$

– Ground facts (aka **ABox**) use the terminology (data)

$$
\begin{aligned}
\mathsf{John} : \mathsf{Patient} &\sqcap \\
&\exists \mathsf{suffersFrom}.\mathsf{HeartDisease}
\end{aligned}
$$

Why Care About Semantics?

Why Care About Semantics?



Why Care About Semantics?

Why should I care about semantics?

Why Care About Semantics?

Why should I care about semantics?

Why Care About Semantics?

Why should I care about semantics?

Well, from a philosophical POV, we need to specify the relationship between statements in the logic and the existential phenomena they describe.

Why Care About Semantics?

Why should I care about semantics?

Well, from a philosophical POV, we need to specify the relationship between statements in the logic and the existential phenomena they describe.

That's OK, but I don't get paid for philosophy.

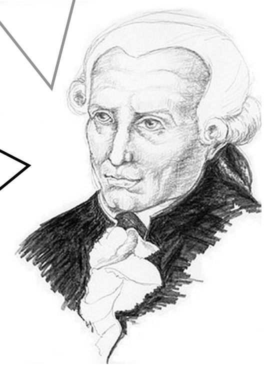
Why Care About Semantics?

Why should I care about semantics?

Well, from a philosophical POV, we need to specify the relationship between statements in the logic and the existential phenomena they describe.

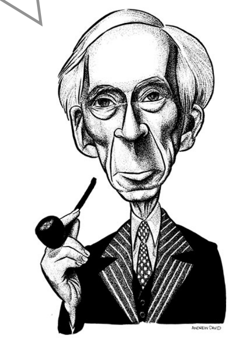
That's OK, but I don't get paid for philosophy.

From a practical POV, in order to specify and test ontology-based information systems we need to precisely define their intended behaviour



Why Care About Semantics?

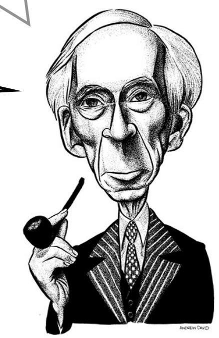
In FOL we define the semantics in terms of models (a model theory). A model is supposed to be an analogue of (part of) the world being modeled. FOL uses a very simple kind of model, in which “objects” in the world (not necessarily physical objects) are modeled as elements of a set, and relationships between objects are modeled as sets of tuples.



Why Care About Semantics?

In FOL we define the semantics in terms of models (a model theory). A model is supposed to be an analogue of (part of) the world being modeled. FOL uses a very simple kind of model, in which "objects" in the world (not necessarily physical objects) are modeled as elements of a set, and relationships between objects are modeled as sets of tuples.

This is exactly the same kind of model as used in a database: objects in the world are modeled as values (elements) and relationships as tables (sets of tuples).



What are Ontologies Good For?

- Coherent **user-centric view** of domain
  - Help identify and resolve disagreements
- Ontology-based **Information Systems**
  - View of data that is independent of logical/ physical schema
  - Queries use terms familiar to users
  - Answers reflect schema & data, e.g.:

    "Patients suffering from Vascular Disease"
  - Query expansion/navigation/refinement
  - Incomplete and semi-structured data
  - Integration of heterogeneous sources

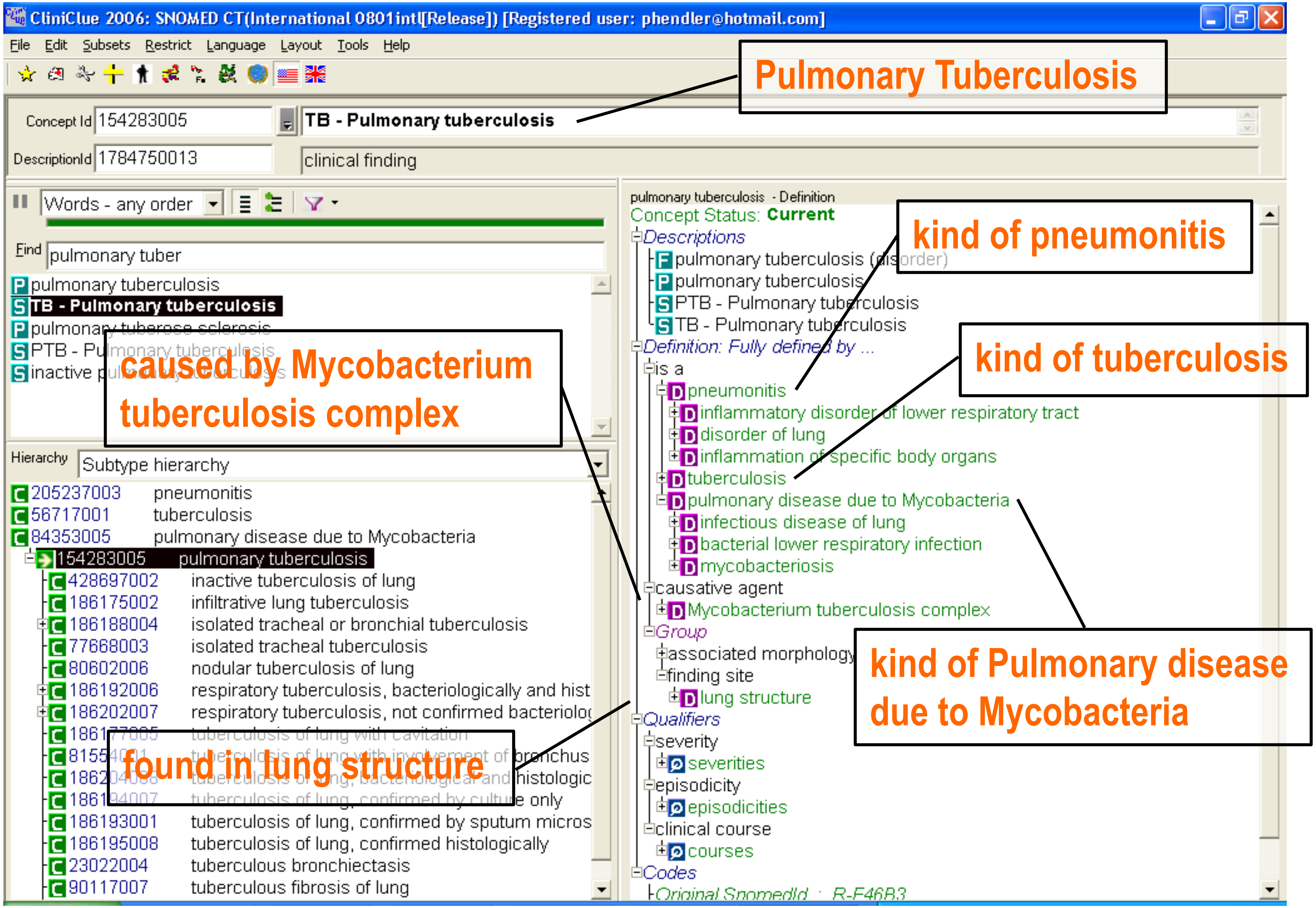
Now... *that* should clear up a few things around here

Healthcare

- UK NHS **£6.2 billion** “Connecting for Health” IT programme
- Key component is **Care Records Service** (CRS)
  - “Live, interactive patient record service accessible 24/7”
  - Patient **data distributed** across local centres in 5 regional clusters, and a national DB
    - Detailed **records** held by local service providers
    - Diverse **applications** support radiology, pharmacy, etc
    - **Summaries** sent to national database
  - **SNOMED-CT** ontology provides common **vocabulary** for data
    - Clinical data uses terms drawn from ontology

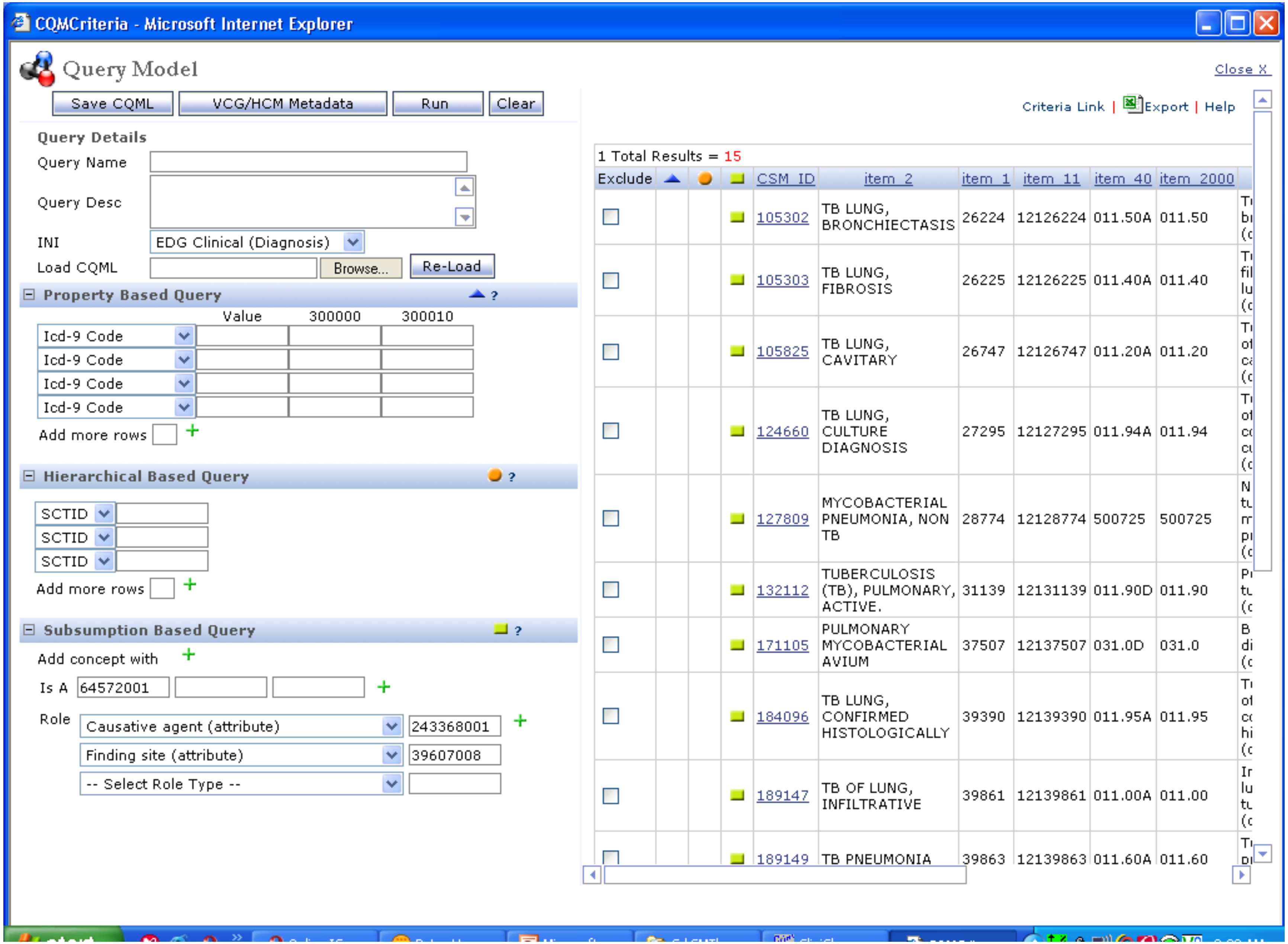
SNOMED-CT

- It's **BIG** – over **400,000 concepts**
- Language used is **EL profile of OWL 2**
- **Multiple hierarchies** and **rich definitions**


CliniClue 2006: SNOMED CT(International 0801intl[Release]) [Registered user: phendler@hotmail.com]
File Edit Subsets Restrict Language Layout Tools Help
Pulmonary Tuberculosis
Concept Id 154283005
TB - Pulmonary tuberculosis
DescriptionId 1784750013
clinical finding
Words - any order
Find pulmonary tuber
pulmonary tuberculosis
TB - Pulmonary tuberculosis
pulmonary tuberose sclerosis
PTB - Pulmonary tuberculosis
inactive pulmonary tuberculosis
caused by Mycobacterium tuberculosis complex
Hierarchy Subtype hierarchy
205237003 pneumonitis
56717001 tuberculosis
84353005 pulmonary disease due to Mycobacteria
154283005 pulmonary tuberculosis
428697002 inactive tuberculosis of lung
186175002 infiltrative lung tuberculosis
186188004 isolated tracheal or bronchial tuberculosis
77668003 isolated tracheal tuberculosis
80602006 nodular tuberculosis of lung
186192006 respiratory tuberculosis, bacteriologically and hist
186202007 respiratory tuberculosis, not confirmed bacteriolog
tuberculosis of lung with cavitation
tuberculosis of lung with involvement of bronchus
tuberculosis of lung, bacteriological and histologic
tuberculosis of lung, confirmed by culture only
186193001 tuberculosis of lung, confirmed by sputum micros
186195008 tuberculosis of lung, confirmed histologically
23022004 tuberculous bronchiectasis
90117007 tuberculous fibrosis of lung
found in lung structure
pulmonary tuberculosis - Definition
Concept Status: Current
Descriptions
pulmonary tuberculosis (disorder)
pulmonary tuberculosis
PTB - Pulmonary tuberculosis
TB - Pulmonary tuberculosis
Definition: Fully defined by ...
is a
pneumonitis
inflammatory disorder of lower respiratory tract
disorder of lung
inflammation of specific body organs
tuberculosis
pulmonary disease due to Mycobacteria
infectious disease of lung
bacterial lower respiratory infection
mycobacteriosis
causative agent
Mycobacterium tuberculosis complex
Group
associated morphology
finding site
lung structure
Qualifiers
severity
severities
episodicity
episodicities
clinical course
courses
Codes
Original SnomedId : R-F46B3
kind of pneumonitis
kind of tuberculosis
kind of Pulmonary disease due to Mycobacteria

SNOMED-CT

- Over **400,000 concepts**
- Language used is **EL fragment of OWL 2**
- **Multiple hierarchies** and **rich definitions**
- Supports, e.g., retrieving details of **all patients having pulmonary TB**



Exclude	CSM ID	item 2	item 1	item 11	item 40	item 2000
	105302	TB LUNG, BRONCHIECTASIS	26224	12126224	011.50A	011.50
	105303	TB LUNG, FIBROSIS	26225	12126225	011.40A	011.40
	105825	TB LUNG, CAVITARY	26747	12126747	011.20A	011.20
	124660	TB LUNG, CULTURE DIAGNOSIS	27295	12127295	011.94A	011.94
	127809	MYCOBACTERIAL PNEUMONIA, NON TB	28774	12128774	500725	500725
	132112	TUBERCULOSIS (TB), PULMONARY, ACTIVE.	31139	12131139	011.90D	011.90
	171105	PULMONARY MYCOBACTERIAL AVIUM	37507	12137507	031.0D	031.0
	184096	TB LUNG, CONFIRMED HISTOLOGICALLY	39390	12139390	011.95A	011.95
	189147	TB OF LUNG, INFILTRATIVE	39861	12139861	011.00A	011.00
	189149	TB PNEUMONIA	39863	12139863	011.60A	011.60

SNOMED-CT

- Over **400,000 concepts**
- Language used is **EL fragment of OWL 2**
- **Multiple hierarchies** and **rich definitions**
- Supports, e.g., retrieving details of **all patients having pulmonary TB**
  - information used e.g., to improve Quality of Care, for Reporting, in epidemiological research, in Decision Support, ...
- **Building and maintenance** is a huge task
  - supported by reasoning tools, e.g., to enrich hierarchies

What About Scalability?

- Only **useful in practice** if we can deal with large ontologies and/or large data sets
- Unfortunately, many ontology languages are highly intractable
  - Satisfiability for OWL 2 ontologies is **2NEXPTIME-complete**
- Problem addressed in practice by
  - Algorithms that work well in **typical cases**
  - Highly **optimised implementations**
  - Use of tractable fragments (aka **profiles**)

Tableau Reasoning Algorithms

Tableau Reasoning Algorithms

Standard technique based on (hyper-) **tableau**

– Reasoning tasks reducible to (un)**satisfiability**

  - E.g., KB $\models$ HeartDisease $\sqsubseteq$ VascularDisease iff
    KB $\cup$ {x:(HeartDisease $\sqcap$ $\neg$VascularDisease)} is *not* satisfiable

Tableau Reasoning Algorithms

Standard technique based on (hyper-) **tableau**

- Reasoning tasks reducible to (un)**satisfiability**
  - E.g., KB ⊨ HeartDisease ⊑ VascularDisease iff KB ∪ {x:(HeartDisease ⊓ ¬VascularDisease)} is *not* satisfiable
- Algorithm tries to construct (an abstraction of) a model in which some individual (x) is an instance of HeartDisease and not an instance of VascularDisease
  - such a model is a counter-example for postulated subsumption

Highly Optimised Implementations

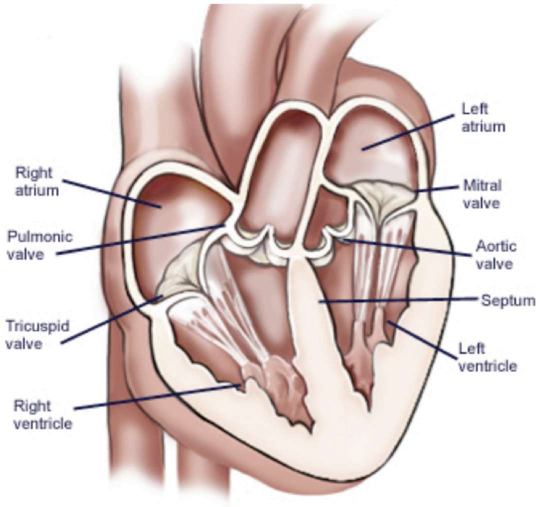
- Lazy unfolding
- Simplification and rewriting,

  e.g., $A \sqcap B \sqsubseteq C \quad \longrightarrow \quad A \sqsubseteq C \sqcup \neg B$
- HyperTableau (reduces non-determinism)
- Fast semi-decision procedures
- Search optimisations
- Reuse of previous computations
- Heuristics

Not computationally optimal,
but effective with many realistic ontologies

Scalability Issues

- Problems with very **large and/or cyclical ontologies**



$$
\begin{aligned}
\mathsf{LeftSide} &\sqsubseteq \exists \mathsf{hasComponent}.\mathsf{AorticValve}\\
\mathsf{LeftSide} &\sqsubseteq \exists \mathsf{hasComponent}.\mathsf{MitralValve}\\
\mathsf{AorticValve} &\sqsubseteq \exists \mathsf{hasConnection}.\mathsf{LeftVentircle}\\
\mathsf{MitralValve} &\sqsubseteq \exists \mathsf{hasConnection}.\mathsf{LeftVentircle}\\
\mathsf{LeftVentricle} &\sqsubseteq \exists \mathsf{isDivisionOf}.\mathsf{LeftSide}
\end{aligned}
$$

- Ontologies may define 10s/100s of thousands of terms
  - can lead to construction of *very* large models
  - requires many (worst case n^2) tests to construct taxonomy

Scalability Issues

- Problems with **large data sets** (ABoxes)
  - Main reasoning problem is (conjunctive) query answering, e.g., retrieve all patients suffering from vascular disease:
    $Q(x) \leftarrow \mathsf{Patient}(x) \wedge \mathsf{suffersFrom}(x, y) \wedge \mathsf{VascularDisease}(y)$
    - Decidability still open for OWL, although minor restrictions (on cycles in non-distinguished variables) restore decidability
  - Query answering reduced to standard decision problem, e.g., by checking for each individual x if KB $\models \mathcal{Q}(x)$
  - Model construction starts with *all* ground facts (data)
- Typical applications may use data sets with **10s/100s of millions** of individuals (or more)

OWL 2 Profiles

- OWL recommendation now updated to **OWL 2**
- OWL 2 defines several **profiles** – fragments with desirable computational properties
  - **OWL 2 EL** targeted at very large ontologies
  - **OWL 2 QL** targeted at very large data sets

OWL 2 EL

- A (near maximal) fragment of OWL 2 such that
  - Satisfiability checking is in PTime (**PTime-Complete**)
  - Data complexity of query answering also PTime-Complete
- Based on $\mathcal{EL}$ family of description logics
- Can exploit **saturation** based reasoning techniques
  - Computes classification in "one pass"
  - Computationally optimal
  - Can be extended to Horn fragment of OWL DL

Saturation-based Technique (basics)

- Normalise ontology axioms to standard form:

$$A \sqsubseteq B \quad A \sqcap B \sqsubseteq C \quad A \sqsubseteq \exists R.B \quad \exists R.B \sqsubseteq C$$

- Saturate using inference rules:

$$\frac{A \sqsubseteq B \quad B \sqsubseteq C}{A \sqsubseteq C} \qquad \frac{A \sqsubseteq B \quad A \sqsubseteq C \quad B \sqcap C \sqsubseteq D}{A \sqsubseteq D}$$

$$\frac{A \sqsubseteq \exists R.B \quad B \sqsubseteq C \quad \exists R.C \sqsubseteq D}{A \sqsubseteq D}$$

- Extension to Horn fragment requires (many) more rules

Saturation-based Technique (basics)

Example:

$$\mathsf{OrganTransplant} \equiv \mathsf{Transplant} \sqcap \exists \mathsf{site}.\mathsf{Organ}$$
$$\mathsf{HeartTransplant} \equiv \mathsf{Transplant} \sqcap \exists \mathsf{site}.\mathsf{Heart}$$
$$\mathsf{Heart} \sqsubseteq \mathsf{Organ}$$

Saturation-based Technique (basics)

Example:

$$\mathsf{OrganTransplant} \equiv \mathsf{Transplant} \sqcap \exists \mathsf{site}.\mathsf{Organ}$$

$$\mathsf{HeartTransplant} \equiv \mathsf{Transplant} \sqcap \exists \mathsf{site}.\mathsf{Heart}$$

$$\mathsf{Heart} \sqsubseteq \mathsf{Organ}$$

Saturation-based Technique (basics)

Example:

$$\mathsf{OrganTransplant} \equiv \mathsf{Transplant} \sqcap \exists \mathsf{site}.\mathsf{Organ}$$

$$\mathsf{HeartTransplant} \equiv \mathsf{Transplant} \sqcap \exists \mathsf{site}.\mathsf{Heart}$$

$$\mathsf{Heart} \sqsubseteq \mathsf{Organ}$$

$$\mathsf{OrganTransplant} \sqsubseteq \mathsf{Transplant}$$

$$\mathsf{OrganTransplant} \sqsubseteq \exists \mathsf{site}.\mathsf{Organ}$$

Saturation-based Technique (basics)

Example:

$$\mathsf{OrganTransplant} \equiv \mathsf{Transplant} \sqcap \exists \mathsf{site}.\mathsf{Organ}$$
$$\mathsf{HeartTransplant} \equiv \mathsf{Transplant} \sqcap \exists \mathsf{site}.\mathsf{Heart}$$
$$\mathsf{Heart} \sqsubseteq \mathsf{Organ}$$

$$\mathsf{OrganTransplant} \sqsubseteq \mathsf{Transplant}$$
$$\mathsf{OrganTransplant} \sqsubseteq \exists \mathsf{site}.\mathsf{Organ}$$
$$\exists \mathsf{site}.\mathsf{Organ} \sqsubseteq \mathsf{SO}$$
$$\mathsf{Transplant} \sqcap \mathsf{SO} \sqsubseteq \mathsf{OrganTransplant}$$

Saturation-based Technique (basics)

Example:

$$\mathsf{OrganTransplant} \equiv \mathsf{Transplant} \sqcap \exists \mathsf{site}.\mathsf{Organ}$$
$$\mathsf{HeartTransplant} \equiv \mathsf{Transplant} \sqcap \exists \mathsf{site}.\mathsf{Heart}$$
$$\mathsf{Heart} \sqsubseteq \mathsf{Organ}$$

$$\mathsf{OrganTransplant} \sqsubseteq \mathsf{Transplant}$$
$$\mathsf{OrganTransplant} \sqsubseteq \exists \mathsf{site}.\mathsf{Organ}$$
$$\exists \mathsf{site}.\mathsf{Organ} \sqsubseteq \mathsf{SO}$$
$$\mathsf{Transplant} \sqcap \mathsf{SO} \sqsubseteq \mathsf{OrganTransplant}$$

Saturation-based Technique (basics)

Example:

$$\mathsf{OrganTransplant} \equiv \mathsf{Transplant} \sqcap \exists \mathsf{site}.\mathsf{Organ}$$
$$\mathsf{HeartTransplant} \equiv \mathsf{Transplant} \sqcap \exists \mathsf{site}.\mathsf{Heart}$$
$$\mathsf{Heart} \sqsubseteq \mathsf{Organ}$$

$$\mathsf{OrganTransplant} \sqsubseteq \mathsf{Transplant}$$
$$\mathsf{OrganTransplant} \sqsubseteq \exists \mathsf{site}.\mathsf{Organ}$$
$$\exists \mathsf{site}.\mathsf{Organ} \sqsubseteq \mathsf{SO}$$
$$\mathsf{Transplant} \sqcap \mathsf{SO} \sqsubseteq \mathsf{OrganTransplant}$$
$$\mathsf{HeartTransplant} \sqsubseteq \mathsf{Transplant}$$
$$\mathsf{HeartTransplant} \sqsubseteq \exists \mathsf{site}.\mathsf{Heart}$$
$$\exists \mathsf{site}.\mathsf{Heart} \sqsubseteq \mathsf{SH}$$
$$\mathsf{Transplant} \sqcap \mathsf{SH} \sqsubseteq \mathsf{HeartTransplant}$$

Saturation-based Technique (basics)

Example:

$$\mathsf{OrganTransplant} \equiv \mathsf{Transplant} \sqcap \exists \mathsf{site}.\mathsf{Organ}$$

$$\mathsf{HeartTransplant} \equiv \mathsf{Transplant} \sqcap \exists \mathsf{site}.\mathsf{Heart}$$

$$\mathsf{Heart} \sqsubseteq \mathsf{Organ}$$

$$\mathsf{OrganTransplant} \sqsubseteq \mathsf{Transplant}$$

$$\mathsf{OrganTransplant} \sqsubseteq \exists \mathsf{site}.\mathsf{Organ}$$

$$\exists \mathsf{site}.\mathsf{Organ} \sqsubseteq \mathsf{SO}$$

$$\mathsf{Transplant} \sqcap \mathsf{SO} \sqsubseteq \mathsf{OrganTransplant}$$

$$\mathsf{HeartTransplant} \sqsubseteq \mathsf{Transplant}$$

$$\mathsf{HeartTransplant} \sqsubseteq \exists \mathsf{site}.\mathsf{Heart}$$

$$\exists \mathsf{site}.\mathsf{Heart} \sqsubseteq \mathsf{SH}$$

$$\mathsf{Transplant} \sqcap \mathsf{SH} \sqsubseteq \mathsf{HeartTransplant}$$

Saturation-based Technique (basics)

Example:

$$\mathsf{OrganTransplant} \equiv \mathsf{Transplant} \sqcap \exists \mathsf{site}.\mathsf{Organ}$$
$$\mathsf{HeartTransplant} \equiv \mathsf{Transplant} \sqcap \exists \mathsf{site}.\mathsf{Heart}$$
$$\mathsf{Heart} \sqsubseteq \mathsf{Organ}$$

$$\mathsf{OrganTransplant} \sqsubseteq \mathsf{Transplant}$$
$$\mathsf{OrganTransplant} \sqsubseteq \exists \mathsf{site}.\mathsf{Organ}$$
$$\exists \mathsf{site}.\mathsf{Organ} \sqsubseteq \mathsf{SO}$$
$$\mathsf{Transplant} \sqcap \mathsf{SO} \sqsubseteq \mathsf{OrganTransplant}$$
$$\mathsf{HeartTransplant} \sqsubseteq \mathsf{Transplant}$$
$$\mathsf{HeartTransplant} \sqsubseteq \exists \mathsf{site}.\mathsf{Heart}$$
$$\exists \mathsf{site}.\mathsf{Heart} \sqsubseteq \mathsf{SH}$$
$$\mathsf{Transplant} \sqcap \mathsf{SH} \sqsubseteq \mathsf{HeartTransplant}$$
$$\mathsf{Heart} \sqsubseteq \mathsf{Organ}$$

Saturation-based Technique (basics)

Example:

$$\mathsf{OrganTransplant} \equiv \mathsf{Transplant} \sqcap \exists \mathsf{site}.\mathsf{Organ}$$
$$\mathsf{HeartTransplant} \equiv \mathsf{Transplant} \sqcap \exists \mathsf{site}.\mathsf{Heart}$$
$$\mathsf{Heart} \sqsubseteq \mathsf{Organ}$$

$$\frac{A \sqsubseteq \exists R.B \quad B \sqsubseteq C \quad \exists R.C \sqsubseteq D}{A \sqsubseteq D}$$

$$\mathsf{OrganTransplant} \sqsubseteq \mathsf{Transplant}$$
$$\mathsf{OrganTransplant} \sqsubseteq \exists \mathsf{site}.\mathsf{Organ}$$
$$\exists \mathsf{site}.\mathsf{Organ} \sqsubseteq \mathsf{SO}$$
$$\mathsf{Transplant} \sqcap \mathsf{SO} \sqsubseteq \mathsf{OrganTransplant}$$
$$\mathsf{HeartTransplant} \sqsubseteq \mathsf{Transplant}$$
$$\mathsf{HeartTransplant} \sqsubseteq \exists \mathsf{site}.\mathsf{Heart}$$
$$\exists \mathsf{site}.\mathsf{Heart} \sqsubseteq \mathsf{SH}$$
$$\mathsf{Transplant} \sqcap \mathsf{SH} \sqsubseteq \mathsf{HeartTransplant}$$
$$\mathsf{Heart} \sqsubseteq \mathsf{Organ}$$

Saturation-based Technique (basics)

Example:

$$\mathsf{OrganTransplant} \equiv \mathsf{Transplant} \sqcap \exists \mathsf{site}.\mathsf{Organ}$$
$$\mathsf{HeartTransplant} \equiv \mathsf{Transplant} \sqcap \exists \mathsf{site}.\mathsf{Heart}$$
$$\mathsf{Heart} \sqsubseteq \mathsf{Organ}$$

$$\frac{A \sqsubseteq \exists R.B \quad B \sqsubseteq C \quad \exists R.C \sqsubseteq D}{A \sqsubseteq D}$$

$$\mathsf{OrganTransplant} \sqsubseteq \mathsf{Transplant}$$
$$\mathsf{OrganTransplant} \sqsubseteq \exists \mathsf{site}.\mathsf{Organ}$$
$$\exists \mathsf{site}.\mathsf{Organ} \sqsubseteq \mathsf{SO}$$
$$\mathsf{Transplant} \sqcap \mathsf{SO} \sqsubseteq \mathsf{OrganTransplant}$$
$$\mathsf{HeartTransplant} \sqsubseteq \mathsf{Transplant}$$
$$\mathsf{HeartTransplant} \sqsubseteq \exists \mathsf{site}.\mathsf{Heart}$$
$$\exists \mathsf{site}.\mathsf{Heart} \sqsubseteq \mathsf{SH}$$
$$\mathsf{Transplant} \sqcap \mathsf{SH} \sqsubseteq \mathsf{HeartTransplant}$$
$$\mathsf{Heart} \sqsubseteq \mathsf{Organ}$$

$$\mathsf{HeartTransplant} \sqsubseteq \mathsf{SO}$$

Saturation-based Technique (basics)

Example:

$$\mathsf{OrganTransplant} \equiv \mathsf{Transplant} \sqcap \exists \mathsf{site}.\mathsf{Organ}$$
$$\mathsf{HeartTransplant} \equiv \mathsf{Transplant} \sqcap \exists \mathsf{site}.\mathsf{Heart}$$
$$\mathsf{Heart} \sqsubseteq \mathsf{Organ}$$

$$\frac{A \sqsubseteq B \quad A \sqsubseteq C \quad B \sqcap C \sqsubseteq D}{A \sqsubseteq D}$$

$$\mathsf{OrganTransplant} \sqsubseteq \mathsf{Transplant}$$
$$\mathsf{OrganTransplant} \sqsubseteq \exists \mathsf{site}.\mathsf{Organ}$$
$$\exists \mathsf{site}.\mathsf{Organ} \sqsubseteq \mathsf{SO}$$
$$\mathsf{Transplant} \sqcap \mathsf{SO} \sqsubseteq \mathsf{OrganTransplant}$$
$$\mathsf{HeartTransplant} \sqsubseteq \mathsf{Transplant}$$
$$\mathsf{HeartTransplant} \sqsubseteq \exists \mathsf{site}.\mathsf{Heart}$$
$$\exists \mathsf{site}.\mathsf{Heart} \sqsubseteq \mathsf{SH}$$
$$\mathsf{Transplant} \sqcap \mathsf{SH} \sqsubseteq \mathsf{HeartTransplant}$$
$$\mathsf{Heart} \sqsubseteq \mathsf{Organ}$$

$$\mathsf{HeartTransplant} \sqsubseteq \mathsf{SO}$$

Saturation-based Technique (basics)

Example:

$$\mathsf{OrganTransplant} \equiv \mathsf{Transplant} \sqcap \exists \mathsf{site}.\mathsf{Organ}$$
$$\mathsf{HeartTransplant} \equiv \mathsf{Transplant} \sqcap \exists \mathsf{site}.\mathsf{Heart}$$
$$\mathsf{Heart} \sqsubseteq \mathsf{Organ}$$

$$\frac{A \sqsubseteq B \quad A \sqsubseteq C \quad B \sqcap C \sqsubseteq D}{A \sqsubseteq D}$$

$$\mathsf{OrganTransplant} \sqsubseteq \mathsf{Transplant}$$
$$\mathsf{OrganTransplant} \sqsubseteq \exists \mathsf{site}.\mathsf{Organ}$$
$$\exists \mathsf{site}.\mathsf{Organ} \sqsubseteq \mathsf{SO}$$
$$\mathsf{Transplant} \sqcap \mathsf{SO} \sqsubseteq \mathsf{OrganTransplant}$$
$$\mathsf{HeartTransplant} \sqsubseteq \mathsf{Transplant}$$
$$\mathsf{HeartTransplant} \sqsubseteq \exists \mathsf{site}.\mathsf{Heart}$$
$$\exists \mathsf{site}.\mathsf{Heart} \sqsubseteq \mathsf{SH}$$
$$\mathsf{Transplant} \sqcap \mathsf{SH} \sqsubseteq \mathsf{HeartTransplant}$$
$$\mathsf{Heart} \sqsubseteq \mathsf{Organ}$$

$$\mathsf{HeartTransplant} \sqsubseteq \mathsf{SO}$$
$$\mathsf{HeartTransplant} \sqsubseteq \mathsf{OrganTransplant}$$

Saturation-based Technique

Performance with large bio-medical ontologies:

	GO	NCI	Galen v.0	Galen v.7	SNOMED
Concepts:	20465	27652	2748	23136	389472
FaCT++	15.24	6.05	465.35	—	650.37
HermiT	199.52	169.47	45.72	—	—
Pellet	72.02	26.47	—	—	—
CEL	1.84	5.76	—	—	1185.70
CB	1.17	3.57	0.32	9.58	49.44
Speed-Up:	1.57X	1.61X	143X	∞	13.15X

OWL 2 QL

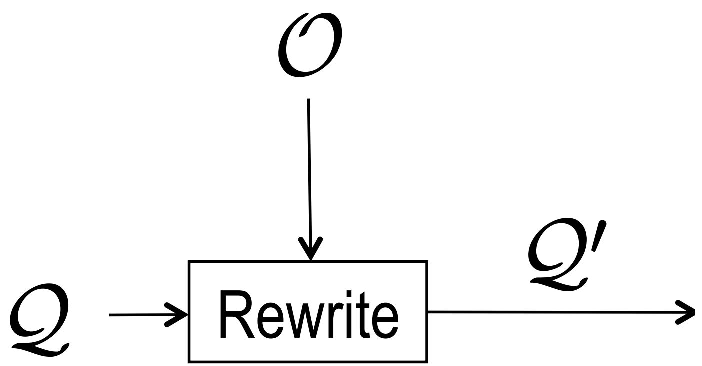
- A (near maximal) fragment of OWL 2 such that
  - Data complexity of conjunctive query answering in **AC^0**, i.e., query answering is *first order reducible*
- Based on **DL-Lite** family of description logics
- Can exploit **query rewriting** based reasoning technique
  - Computationally optimal
  - Data storage and query evaluation can be delegated to standard RDBMS
  - Can be extended to more expressive languages (beyond AC^0) by delegating query answering to a Datalog engine

Query Rewriting Technique (basics)

- Given ontology $\mathcal{O}$ and query $\mathcal{Q}$, use $\mathcal{O}$ to rewrite $\mathcal{Q}$ as $\mathcal{Q}'$ such that, for any set of ground facts $\mathcal{A}$:
  - ans($\mathcal{Q}$, $\mathcal{O}$, $\mathcal{A}$) = ans($\mathcal{Q}'$, $\emptyset$, $\mathcal{A}$)

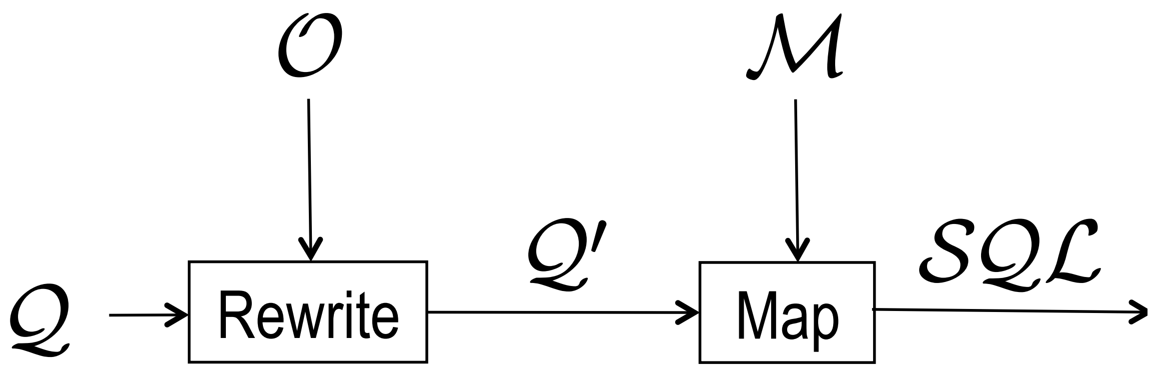
Query Rewriting Technique (basics)

- Given ontology $\mathcal{O}$ and query $\mathcal{Q}$, use $\mathcal{O}$ to rewrite $\mathcal{Q}$ as $\mathcal{Q}'$ such that, for any set of ground facts $\mathcal{A}$:
  - $\text{ans}(\mathcal{Q}, \mathcal{O}, \mathcal{A}) = \text{ans}(\mathcal{Q}', \emptyset, \mathcal{A})$



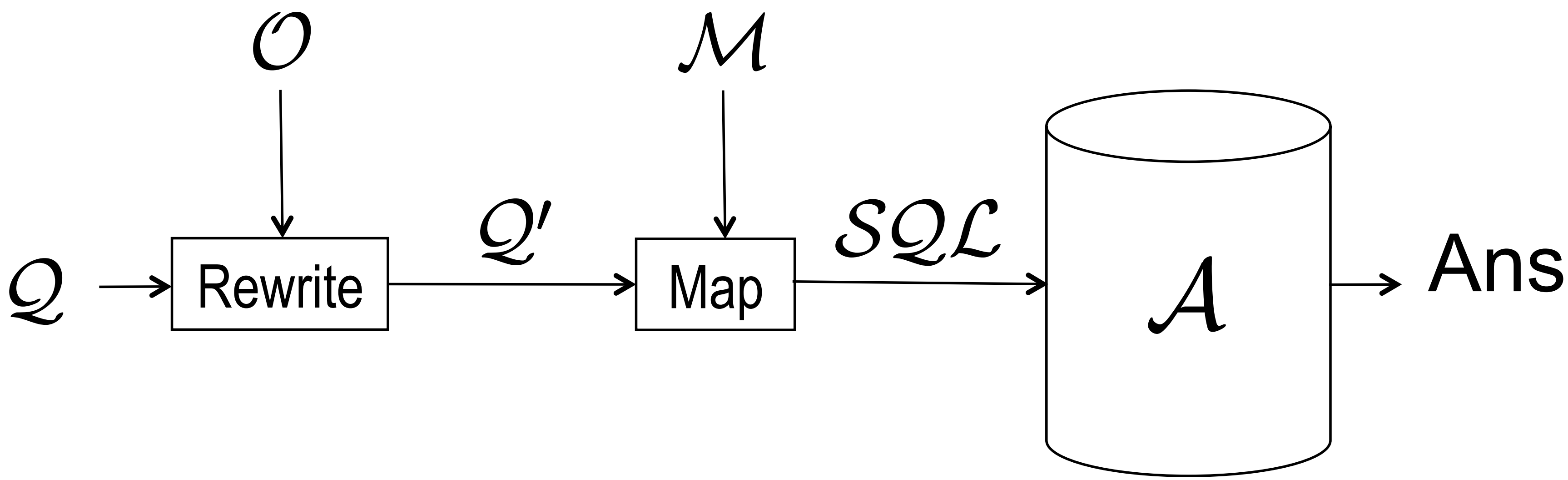
Query Rewriting Technique (basics)

- Given ontology $\mathcal{O}$ and query $\mathcal{Q}$, use $\mathcal{O}$ to rewrite $\mathcal{Q}$ as $\mathcal{Q}'$ such that, for any set of ground facts $\mathcal{A}$:
  - ans($\mathcal{Q}$, $\mathcal{O}$, $\mathcal{A}$) = ans($\mathcal{Q}'$, $\emptyset$, $\mathcal{A}$)



Query Rewriting Technique (basics)

- Given ontology $\mathcal{O}$ and query $\mathcal{Q}$, use $\mathcal{O}$ to rewrite $\mathcal{Q}$ as $\mathcal{Q}'$ such that, for any set of ground facts $\mathcal{A}$:
  - ans($\mathcal{Q}$, $\mathcal{O}$, $\mathcal{A}$) = ans($\mathcal{Q}'$, $\emptyset$, $\mathcal{A}$)



Query Rewriting Technique (basics)

- Given ontology $\mathcal{O}$ and query $\mathcal{Q}$, use $\mathcal{O}$ to rewrite $\mathcal{Q}$ as $\mathcal{Q}'$ such that, for any set of ground facts $\mathcal{A}$:
  - ans($\mathcal{Q}$, $\mathcal{O}$, $\mathcal{A}$) = ans($\mathcal{Q}'$, $\emptyset$, $\mathcal{A}$)
- Resolution based query rewriting
  - **Clausify** ontology axioms
  - **Saturate** (clausified) ontology and query using resolution
  - **Prune** redundant query clauses

Query Rewriting Technique (basics)

- Example:

$$\mathsf{Doctor} \sqsubseteq \exists \mathsf{treats}.\mathsf{Patient}$$

$$\mathsf{Consultant} \sqsubseteq \mathsf{Doctor}$$

$$Q(x) \leftarrow \mathsf{treats}(x, y) \wedge \mathsf{Patient}(y)$$

Query Rewriting Technique (basics)

- Example:

$$\mathsf{Doctor} \sqsubseteq \exists \mathsf{treats}.\mathsf{Patient}$$
$$\mathsf{Consultant} \sqsubseteq \mathsf{Doctor}$$

$$\mathsf{treats}(x, f(x)) \leftarrow \mathsf{Doctor}(x)$$
$$\mathsf{Patient}(f(x)) \leftarrow \mathsf{Doctor}(x)$$
$$\mathsf{Doctor}(x) \leftarrow \mathsf{Consultant}(x)$$

$$Q(x) \leftarrow \mathsf{treats}(x, y) \wedge \mathsf{Patient}(y)$$

Query Rewriting Technique (basics)

- Example:

$$\mathsf{Doctor} \sqsubseteq \exists \mathsf{treats}.\mathsf{Patient}$$
$$\mathsf{Consultant} \sqsubseteq \mathsf{Doctor}$$

$$\mathsf{treats}(x, f(x)) \leftarrow \mathsf{Doctor}(x)$$
$$\mathsf{Patient}(f(x)) \leftarrow \mathsf{Doctor}(x)$$
$$\mathsf{Doctor}(x) \leftarrow \mathsf{Consultant}(x)$$

$$Q(x) \leftarrow \mathsf{treats}(x, y) \wedge \mathsf{Patient}(y)$$

Query Rewriting Technique (basics)

- Example:

$$\mathsf{Doctor} \sqsubseteq \exists \mathsf{treats}.\mathsf{Patient}$$

$$\mathsf{Consultant} \sqsubseteq \mathsf{Doctor}$$

$$\mathsf{treats}(x, f(x)) \leftarrow \mathsf{Doctor}(x)$$

$$\mathsf{Patient}(f(x)) \leftarrow \mathsf{Doctor}(x)$$

$$\mathsf{Doctor}(x) \leftarrow \mathsf{Consultant}(x)$$

$$Q(x) \leftarrow \mathsf{treats}(x, y) \wedge \mathsf{Patient}(y)$$

$$Q(x) \leftarrow \mathsf{Doctor}(x) \wedge \mathsf{Patient}(f(x))$$

Query Rewriting Technique (basics)

- Example:

$$\mathsf{Doctor} \sqsubseteq \exists \mathsf{treats}.\mathsf{Patient}$$
$$\mathsf{Consultant} \sqsubseteq \mathsf{Doctor}$$

$$\mathsf{treats}(x, f(x)) \leftarrow \mathsf{Doctor}(x)$$
$$\mathsf{Patient}(f(x)) \leftarrow \mathsf{Doctor}(x)$$
$$\mathsf{Doctor}(x) \leftarrow \mathsf{Consultant}(x)$$

$$Q(x) \leftarrow \mathsf{treats}(x, y) \wedge \mathsf{Patient}(y)$$
$$Q(x) \leftarrow \mathsf{Doctor}(x) \wedge \mathsf{Patient}(f(x))$$

Query Rewriting Technique (basics)

- Example:

$$\mathsf{Doctor} \sqsubseteq \exists \mathsf{treats}.\mathsf{Patient}$$
$$\mathsf{Consultant} \sqsubseteq \mathsf{Doctor}$$

$$\mathsf{treats}(x, f(x)) \leftarrow \mathsf{Doctor}(x)$$
$$\mathsf{Patient}(f(x)) \leftarrow \mathsf{Doctor}(x)$$
$$\mathsf{Doctor}(x) \leftarrow \mathsf{Consultant}(x)$$

$$Q(x) \leftarrow \mathsf{treats}(x, y) \wedge \mathsf{Patient}(y)$$
$$Q(x) \leftarrow \mathsf{Doctor}(x) \wedge \mathsf{Patient}(f(x))$$
$$Q(x) \leftarrow \mathsf{treats}(x, f(x)) \wedge \mathsf{Doctor}(x)$$

Query Rewriting Technique (basics)

- Example:

$$\mathsf{Doctor} \sqsubseteq \exists \mathsf{treats}.\mathsf{Patient}$$
$$\mathsf{Consultant} \sqsubseteq \mathsf{Doctor}$$

$$\mathsf{treats}(x, f(x)) \leftarrow \mathsf{Doctor}(x)$$
$$\mathsf{Patient}(f(x)) \leftarrow \mathsf{Doctor}(x)$$
$$\mathsf{Doctor}(x) \leftarrow \mathsf{Consultant}(x)$$

$$Q(x) \leftarrow \mathsf{treats}(x, y) \wedge \mathsf{Patient}(y)$$
$$Q(x) \leftarrow \mathsf{Doctor}(x) \wedge \mathsf{Patient}(f(x))$$
$$Q(x) \leftarrow \mathsf{treats}(x, f(x)) \wedge \mathsf{Doctor}(x)$$

Query Rewriting Technique (basics)

- Example:

$$\mathsf{Doctor} \sqsubseteq \exists \mathsf{treats}.\mathsf{Patient}$$
$$\mathsf{Consultant} \sqsubseteq \mathsf{Doctor}$$

$$\mathsf{treats}(x, f(x)) \leftarrow \mathsf{Doctor}(x)$$
$$\mathsf{Patient}(f(x)) \leftarrow \mathsf{Doctor}(x)$$
$$\mathsf{Doctor}(x) \leftarrow \mathsf{Consultant}(x)$$

$$Q(x) \leftarrow \mathsf{treats}(x, y) \wedge \mathsf{Patient}(y)$$
$$Q(x) \leftarrow \mathsf{Doctor}(x) \wedge \mathsf{Patient}(f(x))$$
$$Q(x) \leftarrow \mathsf{treats}(x, f(x)) \wedge \mathsf{Doctor}(x)$$
$$Q(x) \leftarrow \mathsf{Doctor}(x)$$

Query Rewriting Technique (basics)

- Example:

$$\mathsf{Doctor} \sqsubseteq \exists \mathsf{treats}.\mathsf{Patient}$$
$$\mathsf{Consultant} \sqsubseteq \mathsf{Doctor}$$

$$\mathsf{treats}(x, f(x)) \leftarrow \mathsf{Doctor}(x)$$
$$\mathsf{Patient}(f(x)) \leftarrow \mathsf{Doctor}(x)$$
$$\mathsf{Doctor}(x) \leftarrow \mathsf{Consultant}(x)$$

$$Q(x) \leftarrow \mathsf{treats}(x, y) \wedge \mathsf{Patient}(y)$$
$$Q(x) \leftarrow \mathsf{Doctor}(x) \wedge \mathsf{Patient}(f(x))$$
$$Q(x) \leftarrow \mathsf{treats}(x, f(x)) \wedge \mathsf{Doctor}(x)$$
$$Q(x) \leftarrow \mathsf{Doctor}(x)$$

Query Rewriting Technique (basics)

- Example:

$$\mathsf{Doctor} \sqsubseteq \exists \mathsf{treats}.\mathsf{Patient}$$
$$\mathsf{Consultant} \sqsubseteq \mathsf{Doctor}$$

$$\mathsf{treats}(x, f(x)) \leftarrow \mathsf{Doctor}(x)$$
$$\mathsf{Patient}(f(x)) \leftarrow \mathsf{Doctor}(x)$$
$$\mathsf{Doctor}(x) \leftarrow \mathsf{Consultant}(x)$$

$$Q(x) \leftarrow \mathsf{treats}(x, y) \land \mathsf{Patient}(y)$$
$$Q(x) \leftarrow \mathsf{Doctor}(x) \land \mathsf{Patient}(f(x))$$
$$Q(x) \leftarrow \mathsf{treats}(x, f(x)) \land \mathsf{Doctor}(x)$$
$$Q(x) \leftarrow \mathsf{Doctor}(x)$$
$$Q(x) \leftarrow \mathsf{Consultant}(x)$$

Query Rewriting Technique (basics)

- Example:

$$\mathsf{Doctor} \sqsubseteq \exists \mathsf{treats}.\mathsf{Patient}$$
$$\mathsf{Consultant} \sqsubseteq \mathsf{Doctor}$$

$$\mathsf{treats}(x, f(x)) \leftarrow \mathsf{Doctor}(x)$$
$$\mathsf{Patient}(f(x)) \leftarrow \mathsf{Doctor}(x)$$
$$\mathsf{Doctor}(x) \leftarrow \mathsf{Consultant}(x)$$

$$Q(x) \leftarrow \mathsf{treats}(x, y) \wedge \mathsf{Patient}(y)$$
$$Q(x) \leftarrow \mathsf{Doctor}(x) \wedge \mathsf{Patient}(f(x))$$
$$Q(x) \leftarrow \mathsf{treats}(x, f(x)) \wedge \mathsf{Doctor}(x)$$
$$Q(x) \leftarrow \mathsf{Doctor}(x)$$
$$Q(x) \leftarrow \mathsf{Consultant}(x)$$

Query Rewriting Technique (basics)

- Example:

$$\mathsf{Doctor} \sqsubseteq \exists \mathsf{treats}.\mathsf{Patient}$$
$$\mathsf{Consultant} \sqsubseteq \mathsf{Doctor}$$

$$\mathsf{treats}(x, f(x)) \leftarrow \mathsf{Doctor}(x)$$
$$\mathsf{Patient}(f(x)) \leftarrow \mathsf{Doctor}(x)$$
$$\mathsf{Doctor}(x) \leftarrow \mathsf{Consultant}(x)$$

$$Q(x) \leftarrow \mathsf{treats}(x, y) \wedge \mathsf{Patient}(y)$$
$$\cancel{Q(x) \leftarrow \mathsf{Doctor}(x) \wedge \mathsf{Patient}(f(x))}$$
$$\cancel{Q(x) \leftarrow \mathsf{treats}(x, f(x)) \wedge \mathsf{Doctor}(x)}$$
$$Q(x) \leftarrow \mathsf{Doctor}(x)$$
$$Q(x) \leftarrow \mathsf{Consultant}(x)$$

Query Rewriting Technique (basics)

- Example:

$$\mathsf{Doctor} \sqsubseteq \exists \mathsf{treats}.\mathsf{Patient}$$
$$\mathsf{Consultant} \sqsubseteq \mathsf{Doctor}$$

$$\mathsf{treats}(x, f(x)) \leftarrow \mathsf{Doctor}(x)$$
$$\mathsf{Patient}(f(x)) \leftarrow \mathsf{Doctor}(x)$$
$$\mathsf{Doctor}(x) \leftarrow \mathsf{Consultant}(x)$$

$$Q(x) \leftarrow \mathsf{treats}(x, y) \wedge \mathsf{Patient}(y)$$
$$\cancel{Q(x) \leftarrow \mathsf{Doctor}(x) \wedge \mathsf{Patient}(f(x))}$$
$$\cancel{Q(x) \leftarrow \mathsf{treats}(x, f(x)) \wedge \mathsf{Doctor}(x)}$$
$$Q(x) \leftarrow \mathsf{Doctor}(x)$$
$$Q(x) \leftarrow \mathsf{Consultant}(x)$$

- For DL-Lite, result is a union of conjunctive queries

Query Rewriting Technique (basics)

- Data can be stored/left in **RDBMS**
- Relationship between ontology and DB defined by **mappings**, e.g.:

$$\begin{array}{rcl} \mathsf{Doctor} & \mapsto & \mathsf{SELECT\ Name\ FROM\ Doctor} \\ \mathsf{Patient} & \mapsto & \mathsf{SELECT\ Name\ FROM\ Patient} \\ \mathsf{treats} & \mapsto & \mathsf{SELECT\ DName,\ PName\ FROM\ Treats} \end{array}$$

- UCQ translated into **SQL query**:

```
SELECT Name FROM Doctor UNION
SELECT DName FROM Treats, Patient WHERE PName=Name
```

Some Research Challenges

- Extend saturation-based techniques to non-Horn fragments
  - SNOMED users want negation and/or disjunction
    - Non infectious Pneumonia
    - Infectious or Malignant disorder of lung
    - Burn injury of face neck or scalp
- Extend reasoning support
  - Modularity
  - Explanation
  - ...

Some (more) Research Challenges

- Open questions w.r.t. query rewriting
  - FO rewritability (AC^0) only for very weak ontology languages
  - Even for AC^0 languages, queries can get very large (order $(|\mathcal{O}| \cdot |\mathcal{Q}|)^{|\mathcal{Q}|}$), and existing RDBMSs may behave poorly
  - Larger fragments require (at least) Datalog engines and/or extension to technique (e.g., partial materialisation)
- Integrating DL/DB research
  - Ontologies -v- dependencies
  - Open world -v- closed world

Thanks To

- Boris Motik
- Yevgeny Kazakov
- Héctor Pérez-Urbina
- Rob Shearer

Select Bibliography

[1] Baader, Horrocks, and Sattler. Description Logics. In *Handbook of Knowledge Representation*. Elsevier, 2007.

[2] Motik, Shearer, and Horrocks. Hypertableau reasoning for description logics. J. of Artificial Intelligence Research, 2009.

[3] Baader, Brandt, and Lutz. Pushing the EL envelope. IJCAI 2005, pages 364–369, 2005.

[4] Kazakov. Consequence-driven reasoning for Horn-SHIQ ontologies. IJCAI 2009, pages 2040–2045, 2009.

[5] Calvanese, De Giacomo, Lembo, Lenzerini, and Rosati. Tractable reasoning and efficient query answering in description logics: The DL-Lite family. J. of Automated Reasoning, 39(3):385–429, 2007.

[6] Perez-Urbina, Motik, and Horrocks. Tractable query answering and rewriting under description logic constraints. J. of Applied Logic, 2009.

[7] Andrea Calì, Georg Gottlob, Thomas Lukasiewicz. Datalog±: a unified approach to ontologies and integrity constraints. ICDT 2009: 14–30.