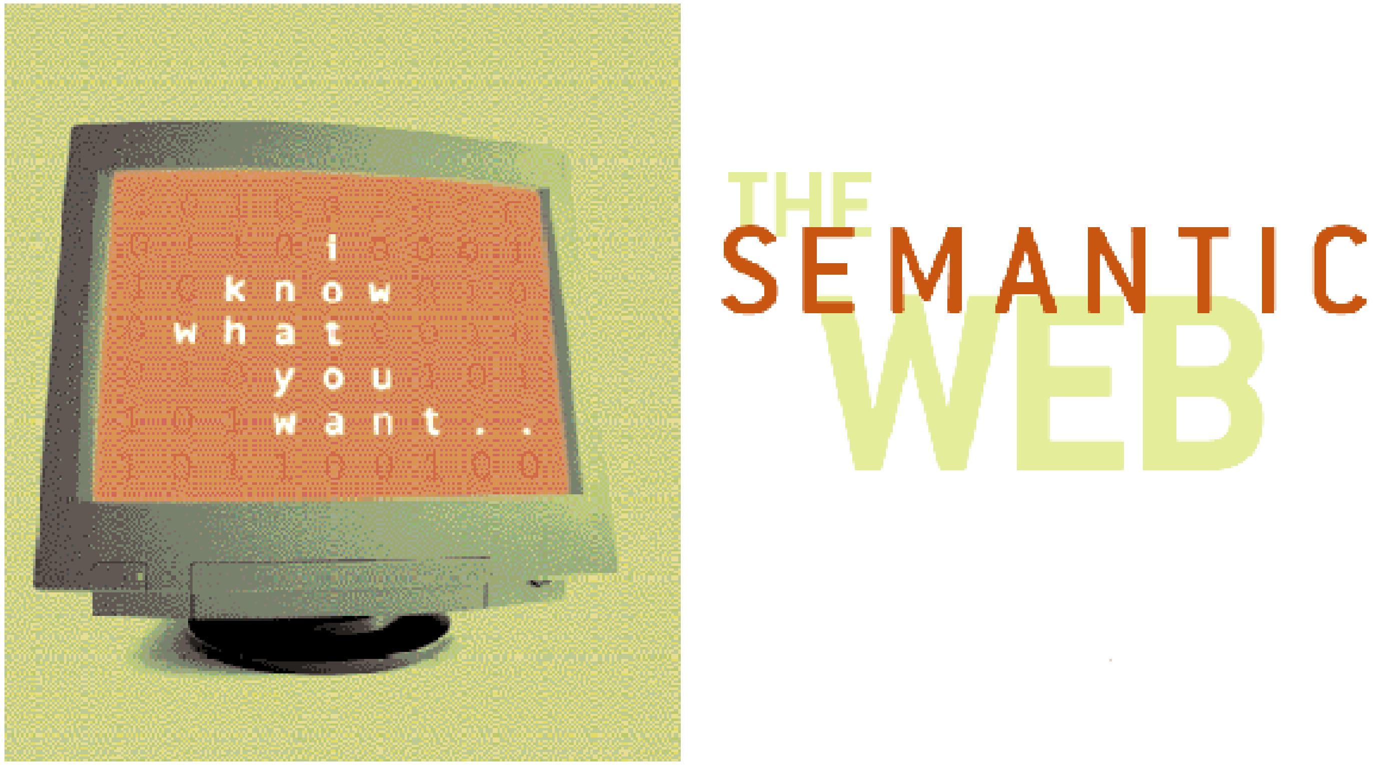
Ontologies and the Semantic Web

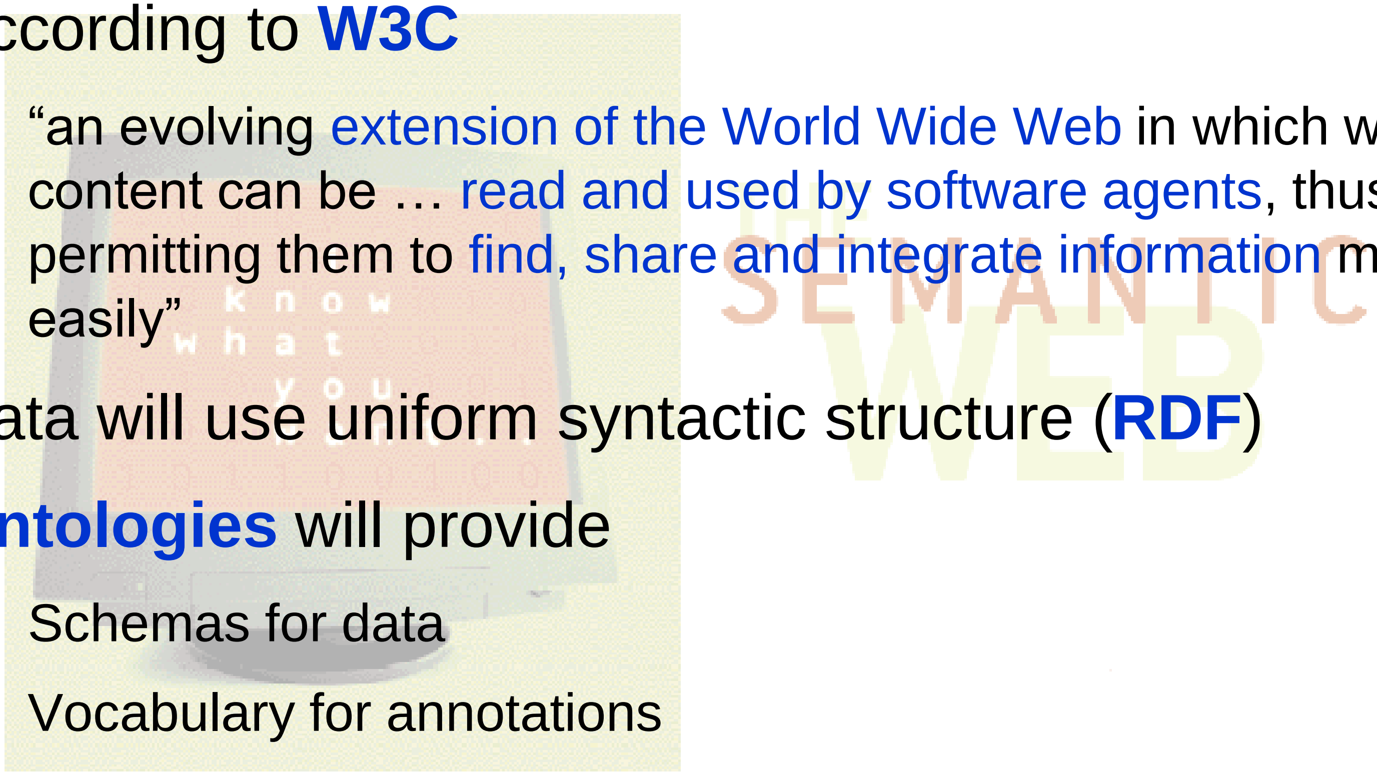
The Story So Far

Semantic Web



Semantic Web

- According to **W3C**
  - "an evolving extension of the World Wide Web in which web content can be … read and used by software agents, thus permitting them to find, share and integrate information more easily"
- Data will use uniform syntactic structure (**RDF**)
- **Ontologies** will provide
  - Schemas for data
  - Vocabulary for annotations
- Ultimate goal is to transform web into a platform for distributed applications and sharing (linking) of data



What is an Ontology?

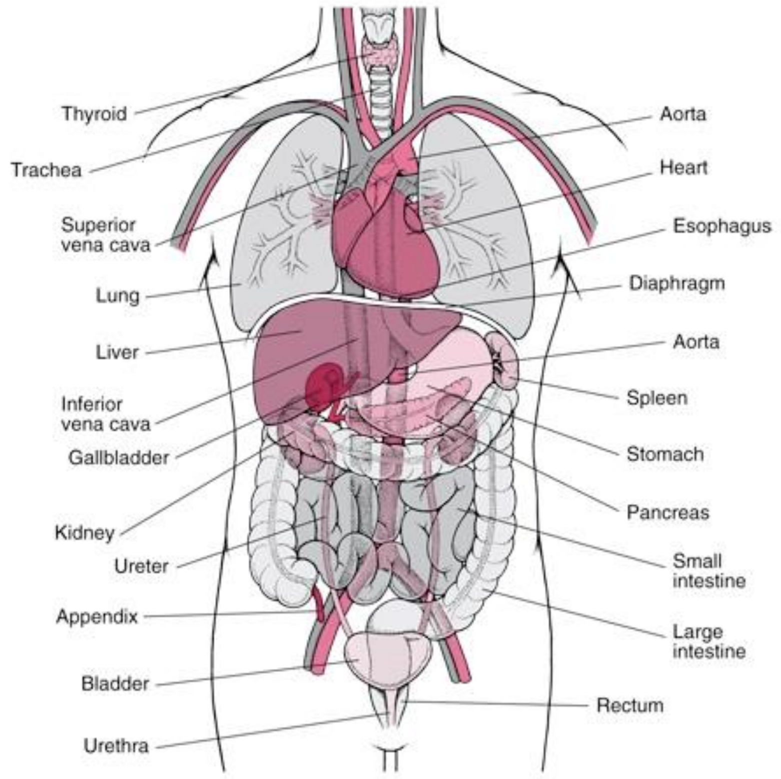
What is an Ontology?

A model of (some aspect of) the world

What is an Ontology?

A model of (some aspect of) the world

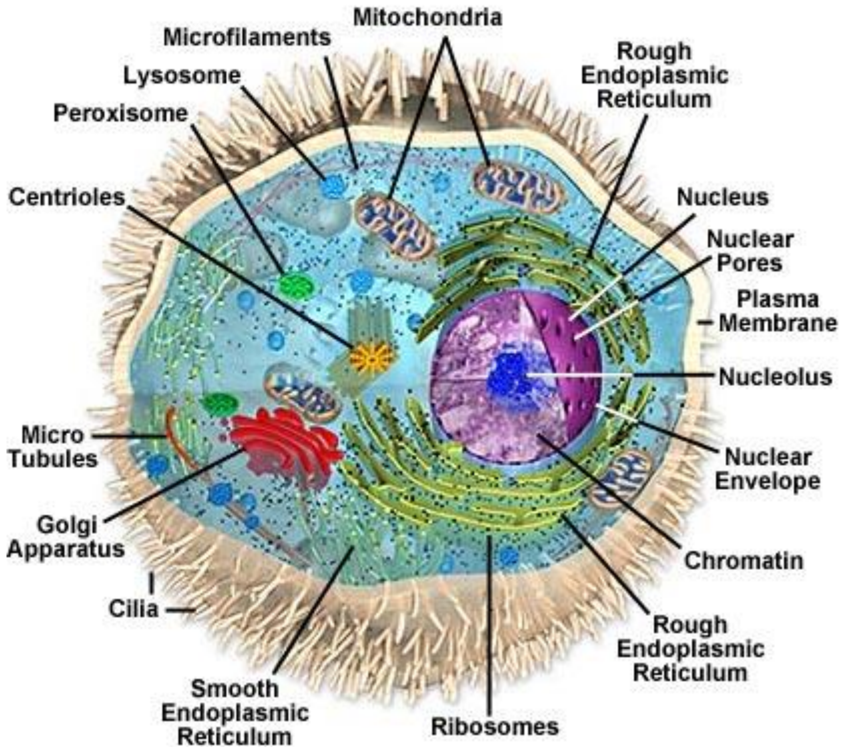
- Introduces **vocabulary** relevant to domain, e.g.:
  - Anatomy



What is an Ontology?

A model of (some aspect of) the world

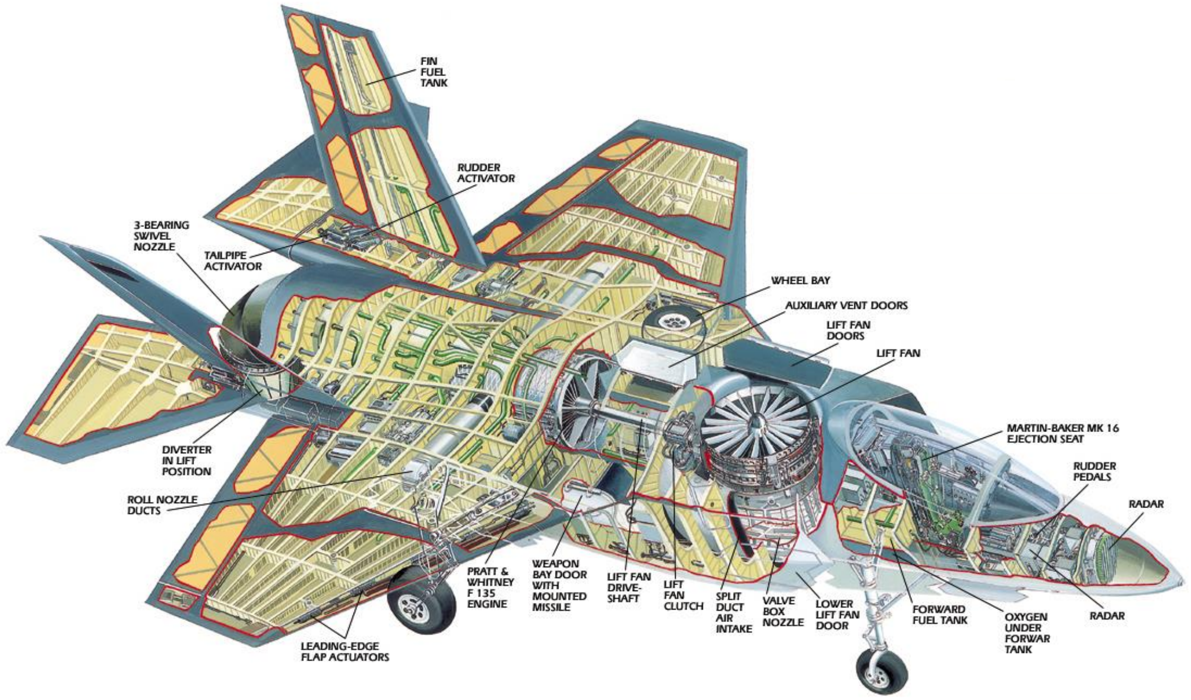
- Introduces **vocabulary** relevant to domain, e.g.:
  - Anatomy
  - Cellular biology



What is an Ontology?

A model of (some aspect of) the world

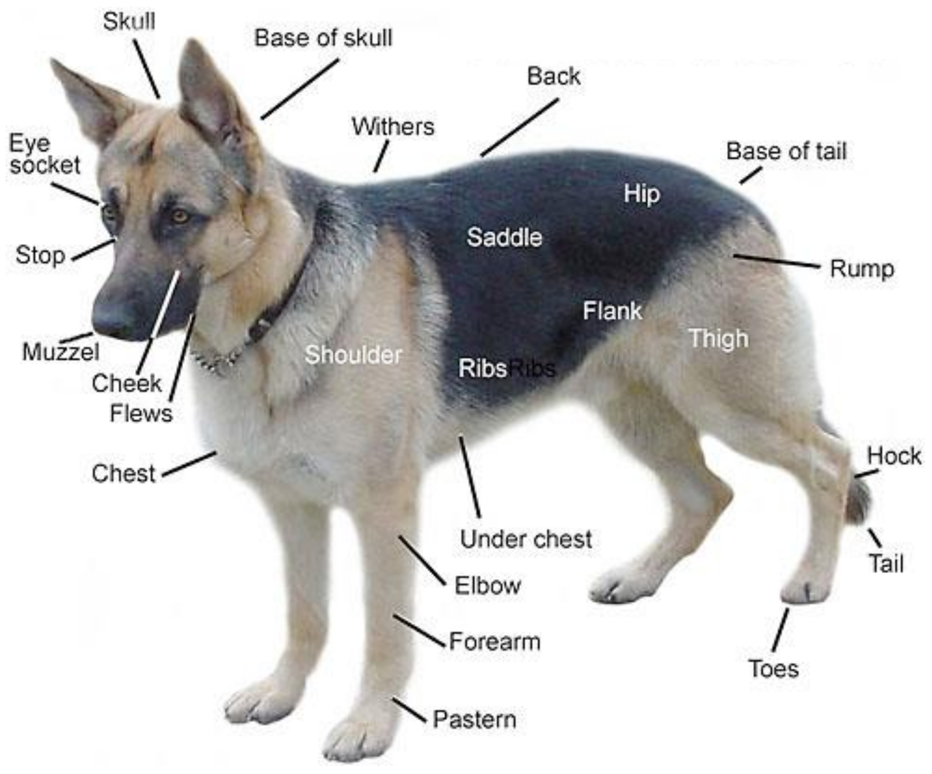
- Introduces **vocabulary** relevant to domain, e.g.:
  - Anatomy
  - Cellular biology
  - Aerospace



What is an Ontology?

A model of (some aspect of) the world

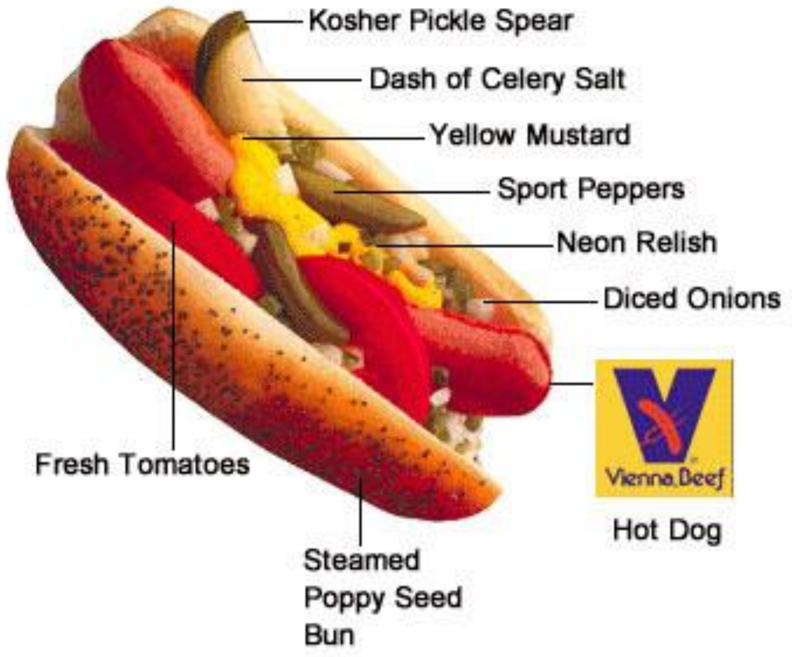
- Introduces **vocabulary** relevant to domain, e.g.:
  - Anatomy
  - Cellular biology
  - Aerospace
  - Dogs



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
  - …

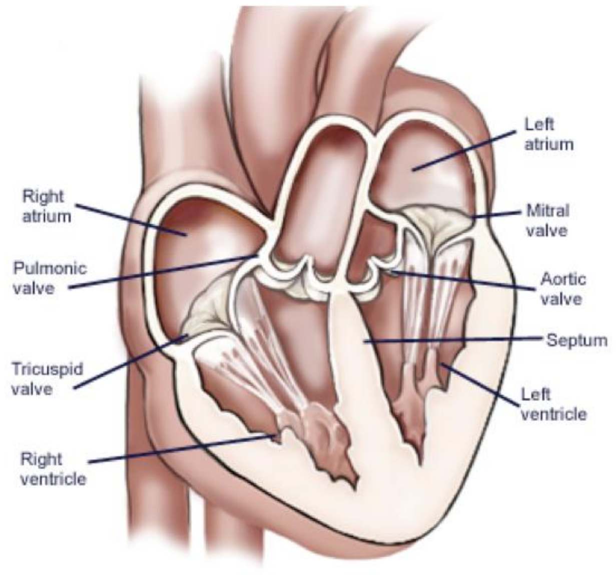


What is an Ontology?

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?

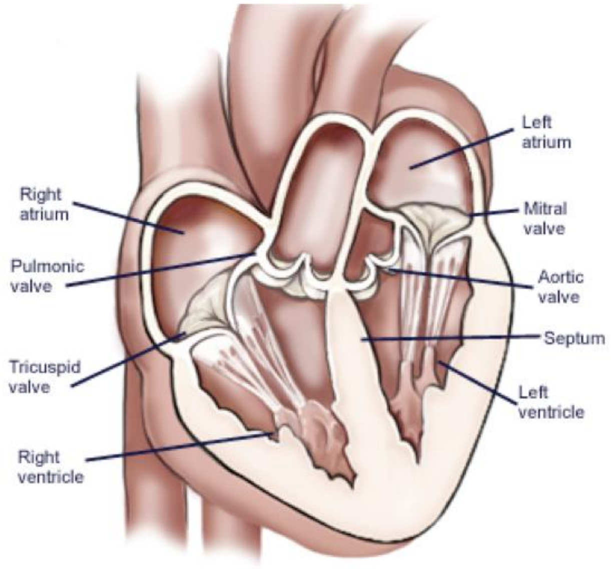
A model of (some aspect of) the world

- 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

  Requirement for standardised "web ontology language"
- Supported by **tools and infrastructure**
  - APIs (e.g., OWL API, Thea, OWLink)
  - Development environments (e.g., Protégé, Swoop, TopBraid Composer, Neon)
  - Reasoners & Information Systems (e.g., Pellet, Racer, HermiT, Quonto, …)
- Based on **Description Logics** (**SHOIN** / **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**

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

$$\begin{aligned}\mathsf{Heart} \sqsubseteq\ & \mathsf{MuscularOrgan} \sqcap \\ & \exists \mathsf{isPartOf}.\mathsf{CirculatorySystem}\end{aligned}$$

Description Logics (DLs)

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

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

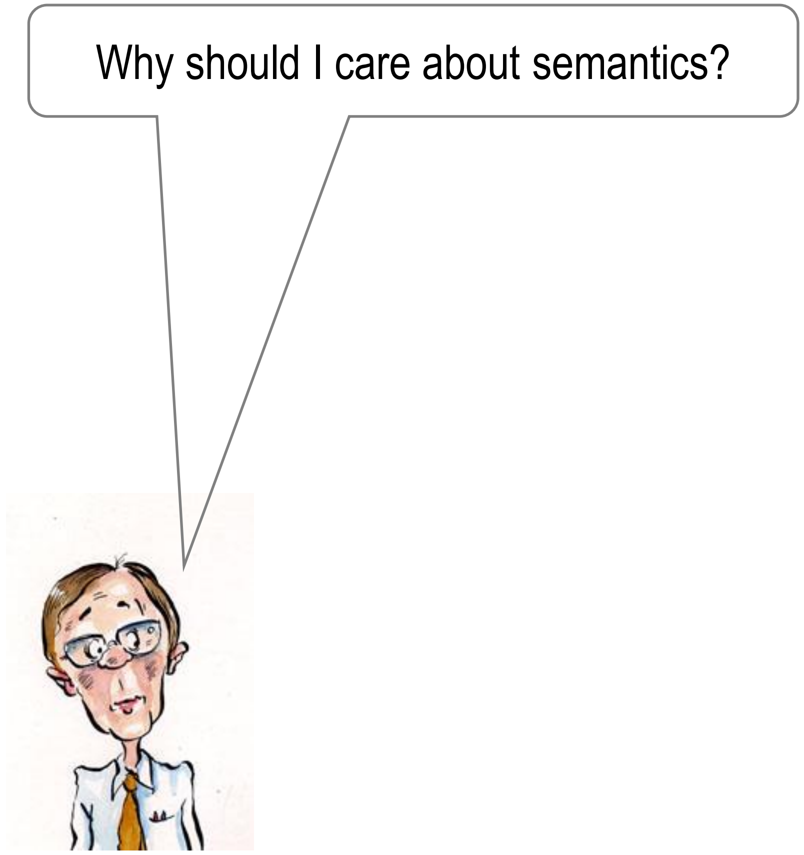
$$\begin{aligned}\mathsf{Heart} &\sqsubseteq \mathsf{MuscularOrgan} \sqcap \\ &\quad \exists \mathsf{isPartOf}.\mathsf{CirculatorySystem} \\ \mathsf{HeartDisease} &\equiv \mathsf{Disease} \sqcap \\ &\quad \exists \mathsf{affects}.\mathsf{Heart} \\ \mathsf{VascularDisease} &\equiv \mathsf{Disease} \sqcap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

- 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?

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.

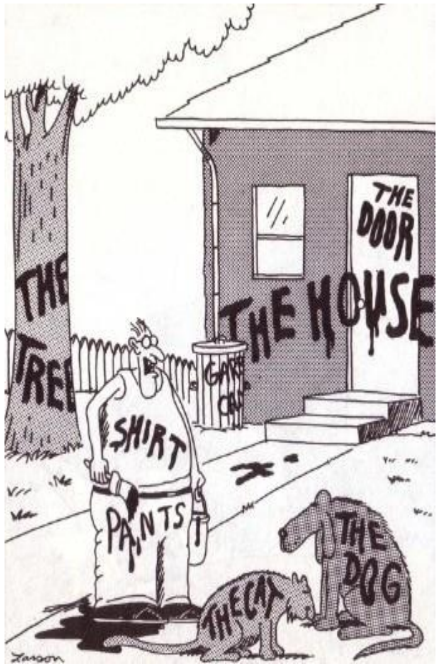
That's OK, but I don't get paid for philosophy.

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

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
  - Answers reflect schema & data, e.g.:

    "Patients suffering from Vascular Disease"



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

What are Ontologies Good For?

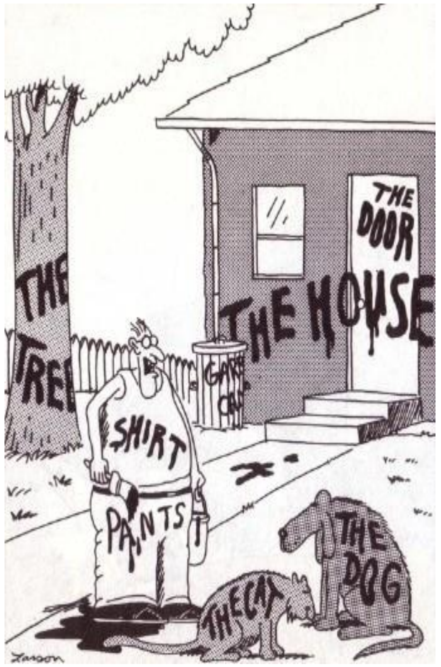
$$
\begin{aligned}
\mathsf{Heart} &\sqsubseteq \mathsf{MuscularOrgan} \sqcap \\
&\quad \exists \mathsf{isPartOf}.\mathsf{CirculatorySystem} \\
\mathsf{HeartDisease} &\equiv \mathsf{Disease} \sqcap \\
&\quad \exists \mathsf{affects}.\mathsf{Heart} \\
\mathsf{VascularDisease} &\equiv \mathsf{Disease} \sqcap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

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

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
  - Answers reflect schema & data, e.g.:

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

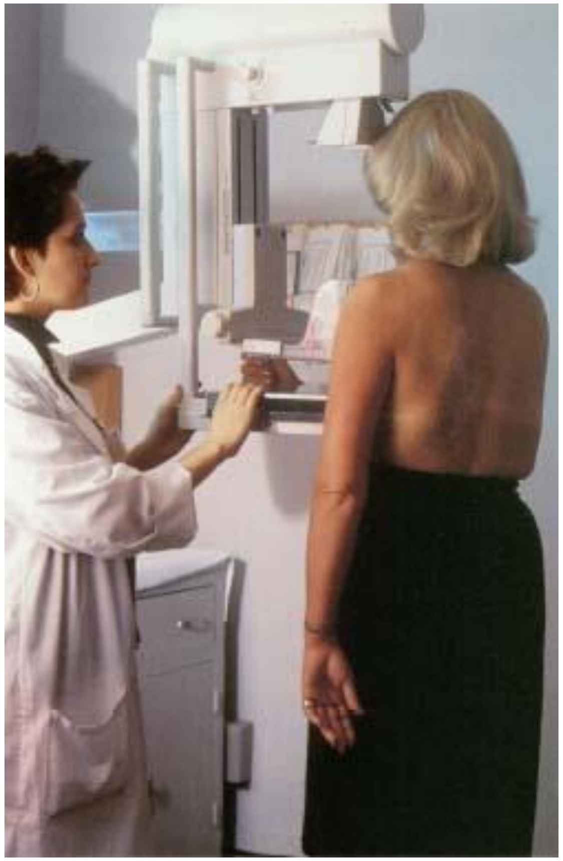


Now... *that* should clear up a few things around here

Information-Based Decisions

Increasingly critical in many areas:

- In Healthcare industry, e.g., selecting patients for screening
  - Too much screening harms patients and wastes money
  - Too little screening costs lives



Information-Based Decisions

Increasingly critical in many areas:

- In Oil and Gas industry, e.g., selecting production parameters
  - Better quality information could add €1B/year net value to Statoil production
  - Poorer quality information and analysis costs €6M/weekend!

Information-Based Decisions

Increasingly critical in many areas:

- In IT industry, e.g., facilitating tech support
  - SAP deals with 80,000 queries/month at a cost of approx. €16M
  - SAP estimate 50% of support staff time spent searching for relevant information

Healthcare

- UK NHS **£10 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
  - **SNOMED-CT** ontology provides common **vocabulary** for data
    - Clinical data uses terms drawn from this ontology
    - The ontology defines more than 400,000 different terms!

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
  - OWL 2 satisfiability is **2NEXPTIME-complete** w.r.t. schema
  - and **NP-Hard** w.r.t. data (upper bound open)
- 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 **²** HeartDisease **v** VascularDisease iff
    KB **[** {x:(HeartDisease **u :**VascularDisease)} is *not* satisfiable

Tableau Reasoning Algorithms

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

- Reasoning tasks reducible to (un)**satisfiability**
  - E.g., KB **²** HeartDisease **v** VascularDisease iff
    KB **[** {x:(HeartDisease **u :**VascularDisease)} is *not* satisfiable
- Algorithm tries to construct (an abstraction of) a model:

$x : \mathsf{HeartDisease} \sqcap \neg\mathsf{VascularDisease}$

Tableau Reasoning Algorithms

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

- Reasoning tasks reducible to (un)**satisfiability**
  - E.g., KB ² HeartDisease **v** VascularDisease iff
    KB **[** {x:(HeartDisease **u** **:**VascularDisease)} is *not* satisfiable
- Algorithm tries to construct (an abstraction of) a model:

$x : \mathsf{HeartDisease} \sqcap \neg \mathsf{VascularDisease}$

$x : \mathsf{HeartDisease}$

Tableau Reasoning Algorithms

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

- Reasoning tasks reducible to (un)**satisfiability**
  - E.g., KB ² HeartDisease **v** VascularDisease iff
    KB **[** {x:(HeartDisease **u :**VascularDisease)} is *not* satisfiable
- Algorithm tries to construct (an abstraction of) a model:

x : HeartDisease $\sqcap$ $\neg$VascularDisease
x : HeartDisease
x : Disease
x : $\exists$affects.Heart

Tableau Reasoning Algorithms

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

- Reasoning tasks reducible to (un)**satisfiability**
  - E.g., KB ² HeartDisease **v** VascularDisease iff
    KB **[** {x:(HeartDisease **u :**VascularDisease)} is *not* satisfiable
- Algorithm tries to construct (an abstraction of) a model:

$$
\begin{aligned}
x &: \mathsf{HeartDisease} \sqcap \neg\mathsf{VascularDisease} \\
x &: \mathsf{HeartDisease} \\
x &: \mathsf{Disease} \\
x &: \exists\mathsf{affects}.\mathsf{Heart} \\
(x, y) &: \mathsf{affects} \\
y &: \mathsf{Heart}
\end{aligned}
$$

Tableau Reasoning Algorithms

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

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

- E.g., KB ² HeartDisease **v** VascularDisease iff
  KB **[** {x:(HeartDisease **u :**VascularDisease)} is *not* satisfiable

– Algorithm tries to construct (an abstraction of) a model:

$$
\begin{aligned}
x &: \mathsf{HeartDisease} \sqcap \neg\mathsf{VascularDisease} \\
x &: \mathsf{HeartDisease} \\
x &: \mathsf{Disease} \\
x &: \exists \mathsf{affects}.\mathsf{Heart} \\
(x, y) &: \mathsf{affects} \\
y &: \mathsf{Heart} \\
y &: \mathsf{MuscularOrgan} \\
y &: \exists \mathsf{isPartOf}.\mathsf{CirculatorySystem}
\end{aligned}
$$

Tableau Reasoning Algorithms

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

- Reasoning tasks reducible to (un)**satisfiability**
  - E.g., KB ² HeartDisease **v** VascularDisease iff
    KB **[** {x:(HeartDisease **u** **:**VascularDisease)} is *not* satisfiable
- Algorithm tries to construct (an abstraction of) a model:

$$\begin{aligned}
x &: \mathsf{HeartDisease} \sqcap \neg\mathsf{VascularDisease}\\
x &: \mathsf{HeartDisease}\\
x &: \mathsf{Disease}\\
x &: \exists\mathsf{affects}.\mathsf{Heart}\\
(x, y) &: \mathsf{affects}\\
y &: \mathsf{Heart}\\
y &: \mathsf{MuscularOrgan}\\
y &: \exists\mathsf{isPartOf}.\mathsf{CirculatorySystem}\\
(y, z) &: \mathsf{isPartOf}\\
z &: \mathsf{CirculatorySystem}
\end{aligned}$$

Tableau Reasoning Algorithms

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

- Reasoning tasks reducible to (un)**satisfiability**
  - E.g., KB ² HeartDisease **v** VascularDisease iff
    KB **[** {x:(HeartDisease **u :**VascularDisease)} is *not* satisfiable
- Algorithm tries to construct (an abstraction of) a model:

$$\begin{array}{rll}
x : & \mathsf{HeartDisease} \sqcap \neg\mathsf{VascularDisease} & \qquad x : \neg\mathsf{VascularDisease} \\
x : & \mathsf{HeartDisease} & \\
x : & \mathsf{Disease} & \\
x : & \exists\mathsf{affects}.\mathsf{Heart} & \\
(x, y) : & \mathsf{affects} & \\
y : & \mathsf{Heart} & \\
y : & \mathsf{MuscularOrgan} & \\
y : & \exists\mathsf{isPartOf}.\mathsf{CirculatorySystem} & \\
(y, z) : & \mathsf{isPartOf} & \\
z : & \mathsf{CirculatorySystem} &
\end{array}$$

Tableau Reasoning Algorithms

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

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

- E.g., KB ² HeartDisease **v** VascularDisease iff
  KB **[** {x:(HeartDisease **u :**VascularDisease)} is *not* satisfiable

– Algorithm tries to construct (an abstraction of) a model:

$$
\begin{aligned}
x &: \mathsf{HeartDisease} \sqcap \neg\mathsf{VascularDisease} \\
x &: \mathsf{HeartDisease} \\
x &: \mathsf{Disease} \\
x &: \exists\mathsf{affects}.\mathsf{Heart} \\
(x, y) &: \mathsf{affects} \\
y &: \mathsf{Heart} \\
y &: \mathsf{MuscularOrgan} \\
y &: \exists\mathsf{isPartOf}.\mathsf{CirculatorySystem} \\
(y, z) &: \mathsf{isPartOf} \\
z &: \mathsf{CirculatorySystem}
\end{aligned}
$$

$$
\begin{aligned}
x &: \neg\mathsf{VascularDisease} \\
x &: \neg\mathsf{Disease} \sqcup \\
&\quad \neg\exists\mathsf{affects}.(\exists\mathsf{isPartOf}.\mathsf{CirculatorySystem})
\end{aligned}
$$

Tableau Reasoning Algorithms

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

- Reasoning tasks reducible to (un)**satisfiability**
  - E.g., KB ² HeartDisease **v** VascularDisease iff
    KB **[** {x:(HeartDisease **u :**VascularDisease)} is *not* satisfiable
- Algorithm tries to construct (an abstraction of) a model:

$$
\begin{array}{rl}
x: & \mathsf{HeartDisease} \sqcap \neg\mathsf{VascularDisease} \\
x: & \mathsf{HeartDisease} \\
x: & \mathsf{Disease} \\
x: & \exists\mathsf{affects}.\mathsf{Heart} \\
(x,y): & \mathsf{affects} \\
y: & \mathsf{Heart} \\
y: & \mathsf{MuscularOrgan} \\
y: & \exists\mathsf{isPartOf}.\mathsf{CirculatorySystem} \\
(y,z): & \mathsf{isPartOf} \\
z: & \mathsf{CirculatorySystem}
\end{array}
$$

$$
\begin{array}{rl}
x: & \neg\mathsf{VascularDisease} \\
x: & \neg\mathsf{Disease} \sqcup \\
 & \neg\exists\mathsf{affects}.(\exists\mathsf{isPartOf}.\mathsf{CirculatorySystem}) \\
x: & \neg\mathsf{Disease}
\end{array}
$$

Tableau Reasoning Algorithms

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

- Reasoning tasks reducible to (un)**satisfiability**
  - E.g., KB ² HeartDisease **v** VascularDisease iff KB **[** {x:(HeartDisease **u** **:**VascularDisease)} is *not* satisfiable
- Algorithm tries to construct (an abstraction of) a model:

$$\begin{array}{rl}
x: & \mathsf{HeartDisease} \sqcap \neg\mathsf{VascularDisease} \\
x: & \mathsf{HeartDisease} \\
x: & \mathsf{Disease} \\
x: & \exists \mathsf{affects}.\mathsf{Heart} \\
(x,y): & \mathsf{affects} \\
y: & \mathsf{Heart} \\
y: & \mathsf{MuscularOrgan} \\
y: & \exists \mathsf{isPartOf}.\mathsf{CirculatorySystem} \\
(y,z): & \mathsf{isPartOf} \\
z: & \mathsf{CirculatorySystem}
\end{array}$$

$$\begin{array}{rl}
x: & \neg\mathsf{VascularDisease} \\
x: & \neg\mathsf{Disease} \sqcup \\
 & \neg\exists \mathsf{affects}.(\exists \mathsf{isPartOf}.\mathsf{CirculatorySystem}) \\
x: & \neg\mathsf{Disease}
\end{array}$$

Tableau Reasoning Algorithms

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

- Reasoning tasks reducible to (un)**satisfiability**
  - E.g., KB ² HeartDisease **v** VascularDisease iff
    KB **[** {x:(HeartDisease **u :**VascularDisease)} is *not* satisfiable
- Algorithm tries to construct (an abstraction of) a model

$$
\begin{array}{rl}
x: & \mathsf{HeartDisease} \sqcap \neg\mathsf{VascularDisease} \\
x: & \mathsf{HeartDisease} \\
x: & \mathsf{Disease} \\
x: & \exists\mathsf{affects}.\mathsf{Heart} \\
(x,y): & \mathsf{affects} \\
y: & \mathsf{Heart} \\
y: & \mathsf{MuscularOrgan} \\
y: & \exists\mathsf{isPartOf}.\mathsf{CirculatorySystem} \\
(y,z): & \mathsf{isPartOf} \\
z: & \mathsf{CirculatorySystem}
\end{array}
\qquad
\begin{array}{rl}
x: & \neg\mathsf{VascularDisease} \\
x: & \neg\mathsf{Disease} \sqcup \\
 & \neg\exists\mathsf{affects}.(\exists\mathsf{isPartOf}.\mathsf{CirculatorySystem})
\end{array}
$$

Tableau Reasoning Algorithms

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

- Reasoning tasks reducible to (un)**satisfiability**
  - E.g., KB ² HeartDisease **v** VascularDisease iff
    KB **[** {x:(HeartDisease **u :**VascularDisease)} is *not* satisfiable
- Algorithm tries to construct (an abstraction of) a model

$$
\begin{aligned}
x &: \mathsf{HeartDisease} \sqcap \neg \mathsf{VascularDisease} \\
x &: \mathsf{HeartDisease} \\
x &: \mathsf{Disease} \\
x &: \exists \mathsf{affects}.\mathsf{Heart} \\
(x, y) &: \mathsf{affects} \\
y &: \mathsf{Heart} \\
y &: \mathsf{MuscularOrgan} \\
y &: \exists \mathsf{isPartOf}.\mathsf{CirculatorySystem} \\
(y, z) &: \mathsf{isPartOf} \\
z &: \mathsf{CirculatorySystem}
\end{aligned}
$$

$$
\begin{aligned}
x &: \neg \mathsf{VascularDisease} \\
x &: \neg \mathsf{Disease} \sqcup \\
&\quad \neg \exists \mathsf{affects}.(\exists \mathsf{isPartOf}.\mathsf{CirculatorySystem}) \\
x &: \neg \exists \mathsf{affects}.(\exists \mathsf{isPartOf}.\mathsf{CirculatorySystem})
\end{aligned}
$$

Tableau Reasoning Algorithms

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

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

- E.g., KB ² HeartDisease **v** VascularDisease iff
  KB **[** {x:(HeartDisease **u :**VascularDisease)} is *not* satisfiable

– Algorithm tries to construct (an abstraction of) a model

$$
\begin{aligned}
x &: \mathsf{HeartDisease} \sqcap \neg\mathsf{VascularDisease}\\
x &: \mathsf{HeartDisease}\\
x &: \mathsf{Disease}\\
x &: \exists\mathsf{affects}.\mathsf{Heart}\\
(x, y) &: \mathsf{affects}\\
y &: \mathsf{Heart}\\
y &: \mathsf{MuscularOrgan}\\
y &: \exists\mathsf{isPartOf}.\mathsf{CirculatorySystem}\\
(y, z) &: \mathsf{isPartOf}\\
z &: \mathsf{CirculatorySystem}
\end{aligned}
$$

$$
\begin{aligned}
x &: \neg\mathsf{VascularDisease}\\
x &: \neg\mathsf{Disease} \sqcup\\
&\quad \neg\exists\mathsf{affects}.(\exists\mathsf{isPartOf}.\mathsf{CirculatorySystem})\\
x &: \neg\exists\mathsf{affects}.(\exists\mathsf{isPartOf}.\mathsf{CirculatorySystem})\\
x &: \forall\mathsf{affects}.(\forall\mathsf{isPartOf}.\neg\mathsf{CirculatorySystem})
\end{aligned}
$$

Tableau Reasoning Algorithms

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

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

- E.g., KB ² HeartDisease **v** VascularDisease iff
  KB **[** {x:(HeartDisease **u :**VascularDisease)} is *not* satisfiable

– Algorithm tries to construct (an abstraction of) a model

$x : \mathsf{HeartDisease} \sqcap \neg\mathsf{VascularDisease}$
$x : \mathsf{HeartDisease}$
$x : \mathsf{Disease}$
$x : \exists\mathsf{affects}.\mathsf{Heart}$
$(x, y) : \mathsf{affects}$
$y : \mathsf{Heart}$
$y : \mathsf{MuscularOrgan}$
$y : \exists\mathsf{isPartOf}.\mathsf{CirculatorySystem}$
$(y, z) : \mathsf{isPartOf}$
$z : \mathsf{CirculatorySystem}$

$x : \neg\mathsf{VascularDisease}$
$x : \neg\mathsf{Disease} \sqcup \neg\exists\mathsf{affects}.(\exists\mathsf{isPartOf}.\mathsf{CirculatorySystem})$
$x : \neg\exists\mathsf{affects}.(\exists\mathsf{isPartOf}.\mathsf{CirculatorySystem})$
$x : \forall\mathsf{affects}.(\forall\mathsf{isPartOf}.\neg\mathsf{CirculatorySystem})$
$y : \forall\mathsf{isPartOf}.\neg\mathsf{CirculatorySystem}$

Tableau Reasoning Algorithms

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

- Reasoning tasks reducible to (un)**satisfiability**
  - E.g., KB ² HeartDisease **v** VascularDisease iff
    KB **[** {x:(HeartDisease **u :**VascularDisease)} is *not* satisfiable
- Algorithm tries to construct (an abstraction of) a model

$$
\begin{aligned}
x &: \mathsf{HeartDisease} \sqcap \neg\mathsf{VascularDisease} \\
x &: \mathsf{HeartDisease} \\
x &: \mathsf{Disease} \\
x &: \exists\mathsf{affects}.\mathsf{Heart} \\
(x,y) &: \mathsf{affects} \\
y &: \mathsf{Heart} \\
y &: \mathsf{MuscularOrgan} \\
y &: \exists\mathsf{isPartOf}.\mathsf{CirculatorySystem} \\
(y,z) &: \mathsf{isPartOf} \\
z &: \mathsf{CirculatorySystem}
\end{aligned}
$$

$$
\begin{aligned}
x &: \neg\mathsf{VascularDisease} \\
x &: \neg\mathsf{Disease} \sqcup \\
&\quad \neg\exists\mathsf{affects}.(\exists\mathsf{isPartOf}.\mathsf{CirculatorySystem}) \\
x &: \neg\exists\mathsf{affects}.(\exists\mathsf{isPartOf}.\mathsf{CirculatorySystem}) \\
x &: \forall\mathsf{affects}.(\forall\mathsf{isPartOf}.\neg\mathsf{CirculatorySystem}) \\
y &: \forall\mathsf{isPartOf}.\neg\mathsf{CirculatorySystem} \\
z &: \neg\mathsf{CirculatorySystem}
\end{aligned}
$$

Tableau Reasoning Algorithms

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

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

- E.g., KB ² HeartDisease **v** VascularDisease iff
  KB **[** {x:(HeartDisease **u :**VascularDisease)} is *not* satisfiable

– Algorithm tries to construct (an abstraction of) a model

$x : \mathsf{HeartDisease} \sqcap \neg\mathsf{VascularDisease}$

$x : \mathsf{HeartDisease}$

$x : \mathsf{Disease}$

$x : \exists\mathsf{affects}.\mathsf{Heart}$

$(x, y) : \mathsf{affects}$

$y : \mathsf{Heart}$

$y : \mathsf{MuscularOrgan}$

$y : \exists\mathsf{isPartOf}.\mathsf{CirculatorySystem}$

$(y, z) : \mathsf{isPartOf}$

$z : \mathsf{CirculatorySystem}$

$x : \neg\mathsf{VascularDisease}$

$x : \neg\mathsf{Disease} \sqcup \neg\exists\mathsf{affects}.(\exists\mathsf{isPartOf}.\mathsf{CirculatorySystem})$

$x : \neg\exists\mathsf{affects}.(\exists\mathsf{isPartOf}.\mathsf{CirculatorySystem})$

$x : \forall\mathsf{affects}.(\forall\mathsf{isPartOf}.\neg\mathsf{CirculatorySystem})$

$y : \forall\mathsf{isPartOf}.\neg\mathsf{CirculatorySystem}$

$z : \neg\mathsf{CirculatorySystem}$

Highly Optimised Implementations

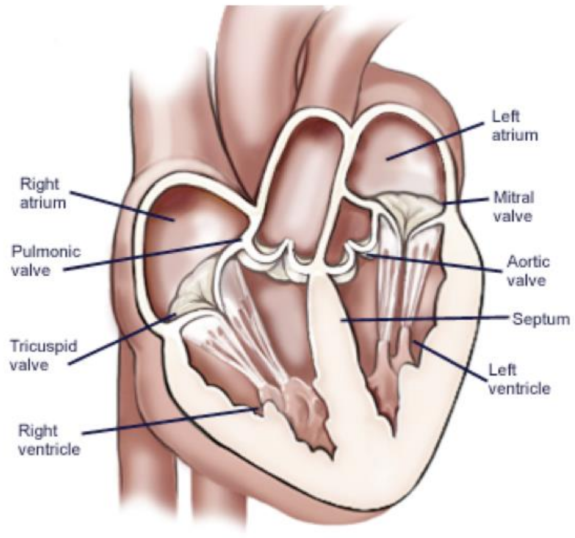
- Lazy unfolding
- Simplification and rewriting,

  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**
  - Ontologies may define 10s/100s of thousands of terms
  - Potentially vast number (n^2) of tests needed for classification
  - Each test can lead to construction of *very* large models



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

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 ² 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 **EL** family of description logics
- Can exploit **saturation** based reasoning techniques
  - Computes complete classification in “one pass”
  - Computationally optimal (PTime for EL)
  - 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:

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

Saturation-based Technique (basics)

Example:

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

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:

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

$$\begin{aligned}
\mathsf{OrganTransplant} &\sqsubseteq \mathsf{Transplant} \\
\mathsf{OrganTransplant} &\sqsubseteq \exists \mathsf{site}.\mathsf{Organ} \\
\exists \mathsf{site}.\mathsf{Organ} &\sqsubseteq \mathsf{SO} \\
\mathsf{Transplant} \sqcap \mathsf{SO} &\sqsubseteq \mathsf{OrganTransplant}
\end{aligned}$$

Saturation-based Technique (basics)

Example:

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

$$\begin{aligned}
\mathsf{OrganTransplant} &\sqsubseteq \mathsf{Transplant} \\
\mathsf{OrganTransplant} &\sqsubseteq \exists \mathsf{site}.\mathsf{Organ} \\
\exists \mathsf{site}.\mathsf{Organ} &\sqsubseteq \mathsf{SO} \\
\mathsf{Transplant} \sqcap \mathsf{SO} &\sqsubseteq \mathsf{OrganTransplant}
\end{aligned}$$

Saturation-based Technique (basics)

Example:

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

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

Saturation-based Technique (basics)

Example:

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

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

Saturation-based Technique (basics)

Example:

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

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

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**
- 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)

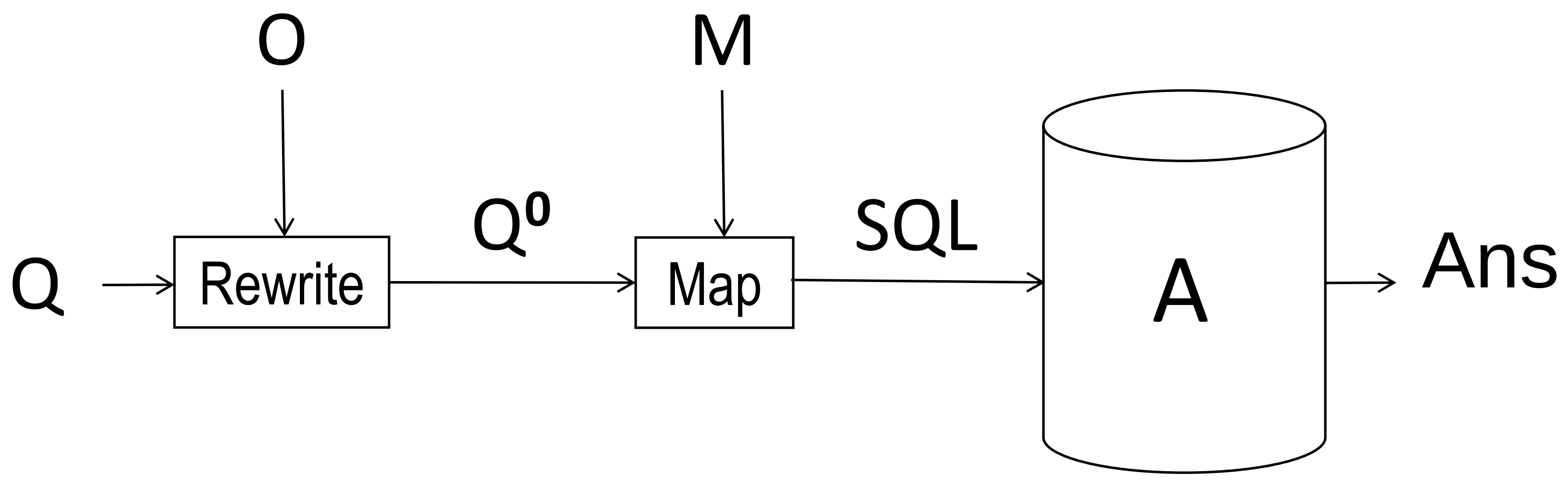
- Given ontology O and query Q, use O to rewrite Q as Q^{0} s.t., for any set of ground facts A:
  - ans(Q, O, A) = ans(Q^{0}, **;**, A)

Query Rewriting Technique (basics)

- Given ontology O and query Q, use O to rewrite Q as Q^{0} s.t., for any set of ground facts A:
  - ans(Q, O, A) = ans(Q^{0}, ;, A)
- Use (GAV) mapping M to map Q^{0} to SQL query

Query Rewriting Technique (basics)

- Given ontology O and query Q, use O to rewrite Q as Q^0 s.t., for any set of ground facts A:
  - ans(Q, O, A) = ans(Q^0, ;, A)
- Use (GAV) mapping M to map Q^0 to SQL query



Query Rewriting Technique (basics)

- Given ontology O and query Q, use O to rewrite Q as Q^{0} s.t., for any set of ground facts A:
  - ans(Q, O, A) = ans(Q^{0}, **;**, A)
- Use (GAV) mapping M to map Q^{0} to SQL query
- 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:

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

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

Query Rewriting Technique (basics)

- Example:

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

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

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

Query Rewriting Technique (basics)

- Example:

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

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

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

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:

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

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

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

Query Rewriting Technique (basics)

- Example:

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

$$\text{Consultant} \sqsubseteq \text{Doctor}$$

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

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

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

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

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

~~$Q(x) \leftarrow \text{treats}(x, f(x)) \wedge \text{Doctor}(x)$~~

$$Q(x) \leftarrow \text{Doctor}(x)$$

$$Q(x) \leftarrow \text{Consultant}(x)$$

Query Rewriting Technique (basics)

- Example:

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

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

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

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

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

Query Rewriting Technique (basics)

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

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

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**:

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

$$\rightsquigarrow$$

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

Problems & Research Challenges

- Combining best features of DLs & DBs
  - In particular, integrating OWA and CWA
- Hard to find a coherent semantic framework
  - Problems mainly due to existential quantifiers: should existentially implied objects be considered different?
    - Does a person owning a phone and an ipod own 2 things?
    - Does a person owning a phone and an iphone own 2 things?
    - Does a person owning a phone and a phone own 2 things?
- Interesting ideas emerging in DL & DB communities, e.g.:
  - *Calì et al. Datalog±: a unified approach to ontologies and integrity constraints. ICDT 2009.*
  - *Motik et al. Bridging the gap between OWL and relational databases. WWW 2007.*

Problems & Research Challenges

- Open questions w.r.t. query rewriting

Problems & Research Challenges

- Open questions w.r.t. query rewriting
  - Currently only for very weak ontology languages

Problems & Research Challenges

- Open questions w.r.t. query rewriting
  - Currently only for very weak ontology languages
  - Even for these languages, queries can get very large (order $(|\mathcal{O}| \cdot |\mathcal{Q}|)^{|\mathcal{Q}|}$), and existing RDBMSs may behave poorly
    - Not clear if this will be a problem in practice, see, e.g., *Savo et al. MASTRO at Work: Experiences on Ontology-based Data Access. DL 2010.*

Problems & Research Challenges

- Open questions w.r.t. query rewriting
  - Currently only for very weak ontology languages
  - Even for these languages, queries can get very large (order $(|\mathcal{O}| \cdot |\mathcal{Q}|)^{|\mathcal{Q}|}$), and existing RDBMSs may behave poorly
    - Not clear if this will be a problem in practice, see, e.g., *Savo et al. MASTRO at Work: Experiences on Ontology-based Data Access. DL 2010.*
  - Larger fragments require (at least) Datalog engines and/or extension to technique (e.g., partial materialisation)
    - Promising new work in this area, see, e.g., *Lutz et al. Conjunctive Query Answering in the Description Logic EL Using a Relational Database System. IJCAI 2009.*

Problems & Research Challenges

- Infrastructure

Problems & Research Challenges

- Infrastructure
  - Standardised query language
    - SPARQL standard for RDF
    - Currently being extended for OWL, see *http://www.w3.org/2009/sparql/wiki/Main_Page*

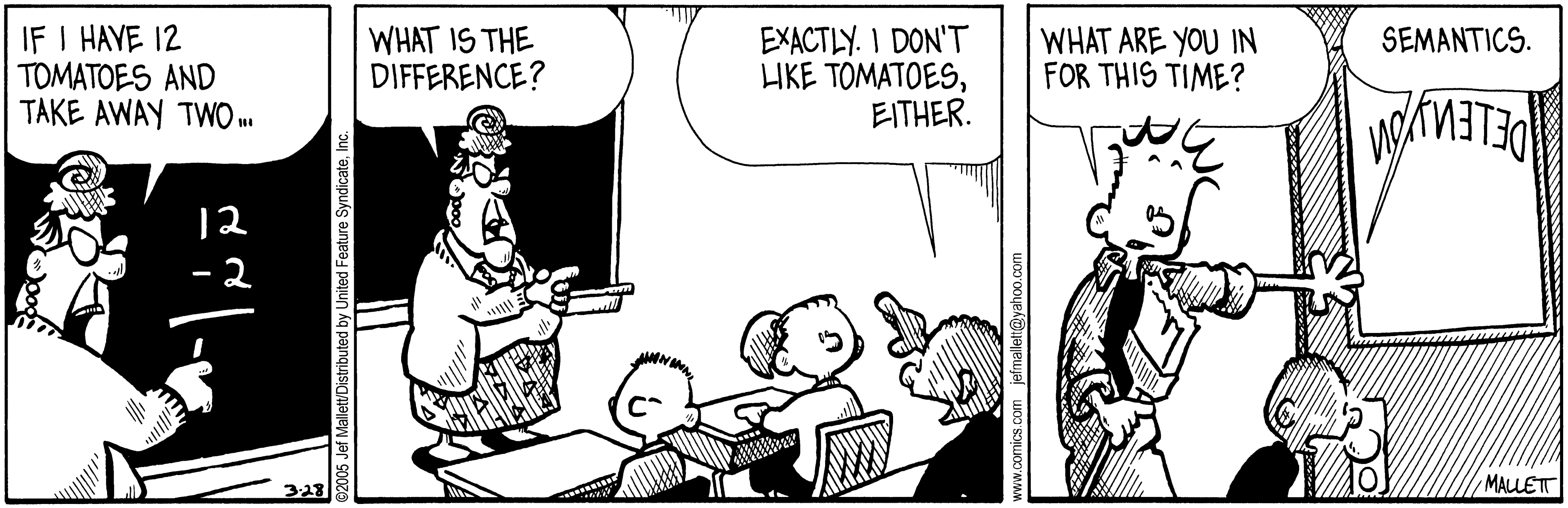
Problems & Research Challenges

- Infrastructure
  - Standardised query language
    - SPARQL standard for RDF
    - Currently being extended for OWL, see *http://www.w3.org/2009/sparql/wiki/Main_Page*
  - Privacy and information hiding
    - May want to keep parts of data/schema private
    - Difficulties compounded when information can be inferred, see, e.g., *Cuenca Grau et al. Privacy-preserving query answering in logic-based information systems. ECAI 2008.*

Problems & Research Challenges

- Infrastructure
  - Standardised query language
    - SPARQL standard for RDF
    - Currently being extended for OWL, see *http://www.w3.org/2009/sparql/wiki/Main_Page*
  - Privacy and information hiding
    - May want to keep parts of data/schema private
    - Difficulties compounded when information can be inferred, see, e.g., *Cuenca Grau et al. Privacy-preserving query answering in logic-based information systems. ECAI 2008.*
  - ...

Thank you for listening


IF I HAVE 12 TOMATOES AND TAKE AWAY TWO...
12
-2
1
3-28
©2005 Jef Mallett/Distributed by United Feature Syndicate, Inc.
WHAT IS THE DIFFERENCE?
EXACTLY. I DON'T LIKE TOMATOES, EITHER.
www.comics.com jefmallett@yahoo.com
WHAT ARE YOU IN FOR THIS TIME?
SEMANTICS.
DETENTION
MALLETT

FRAZZ: © Jeff Mallett/Dist. by United Feature Syndicate, Inc.

Any questions?