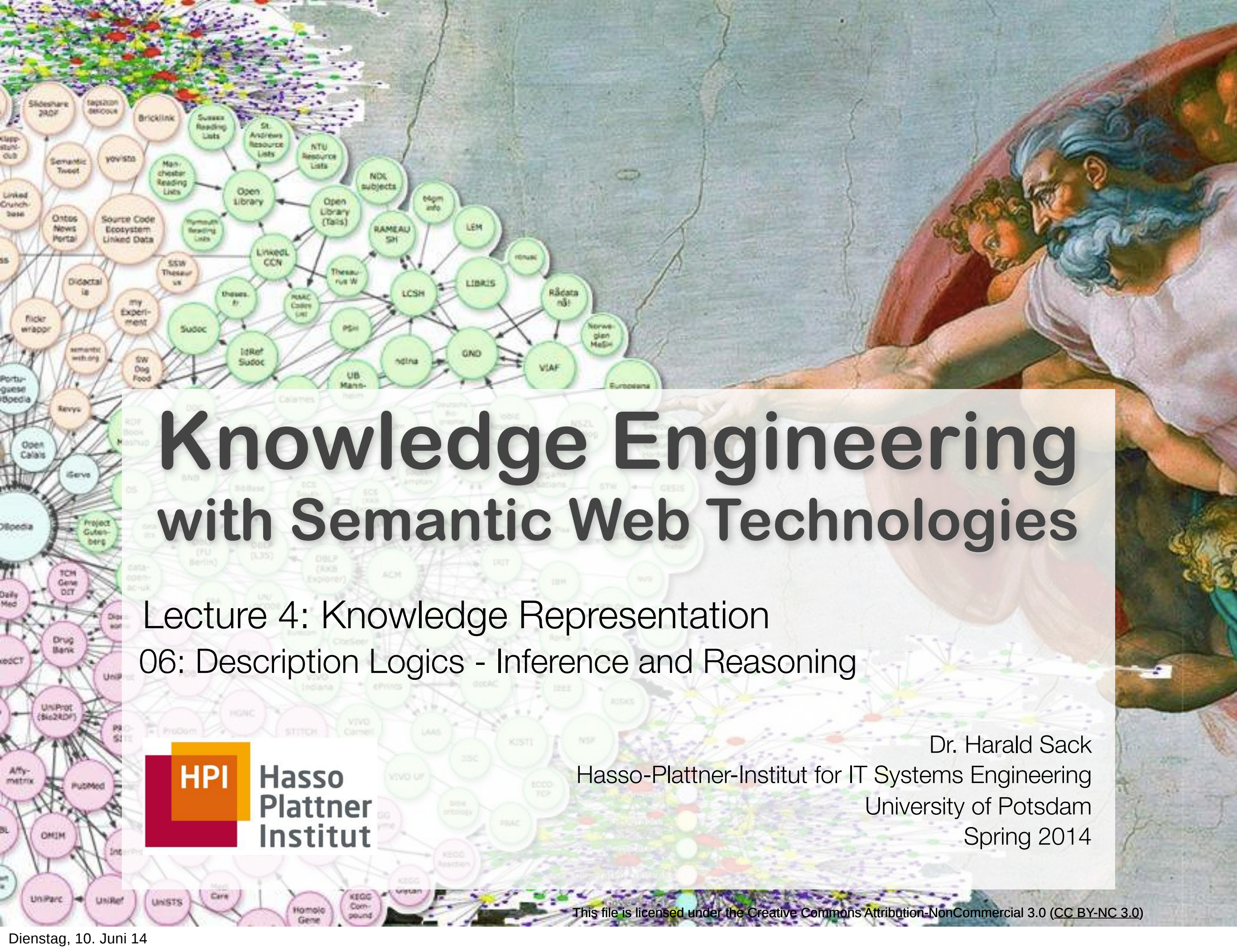




Knowledge Engineering with Semantic Web Technologies

Lecture 4: Knowledge Representation
06: Description Logics - Inference and Reasoning



Dr. Harald Sack
Hasso-Plattner-Institut for IT Systems Engineering
University of Potsdam
Spring 2014



Lecture 4: Knowledge Representations I

Open HPI Course: Knowledge Engineering with Semantic Web Technologies

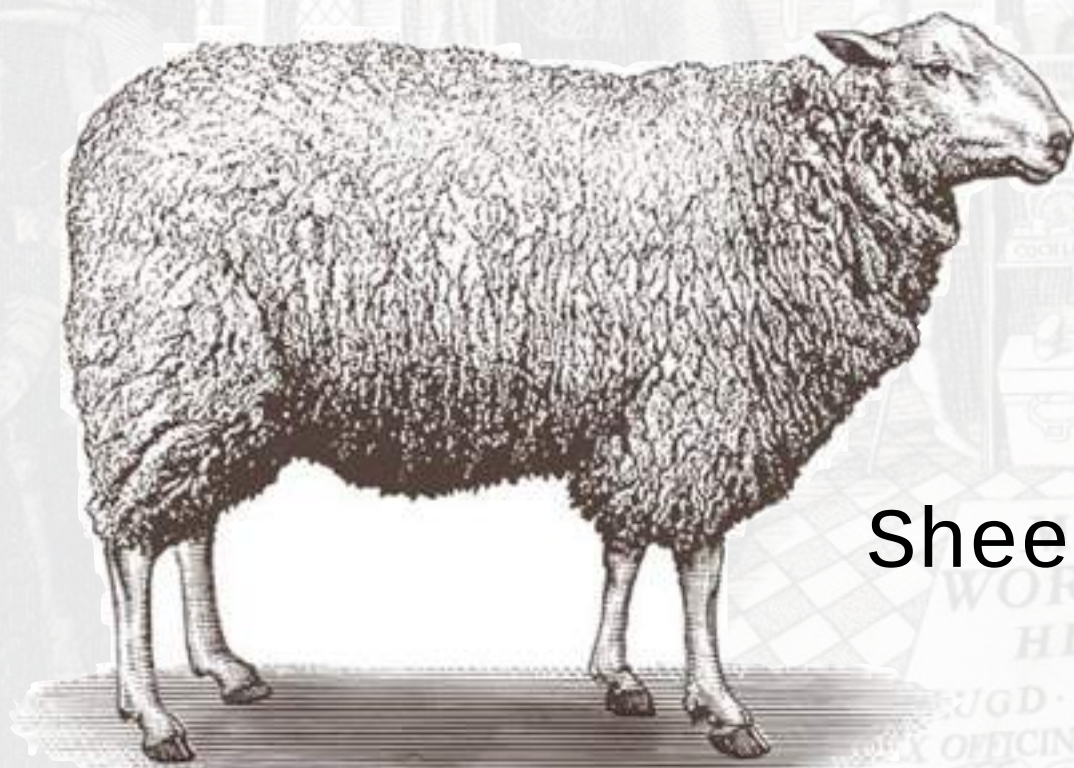


06 DL Inference and Reasoning

Open HPI Course: Knowledge Engineering with Semantic Web Technologies
Lecture 04: Knowledge Representations Pt.1

Open World Assumption - OWA

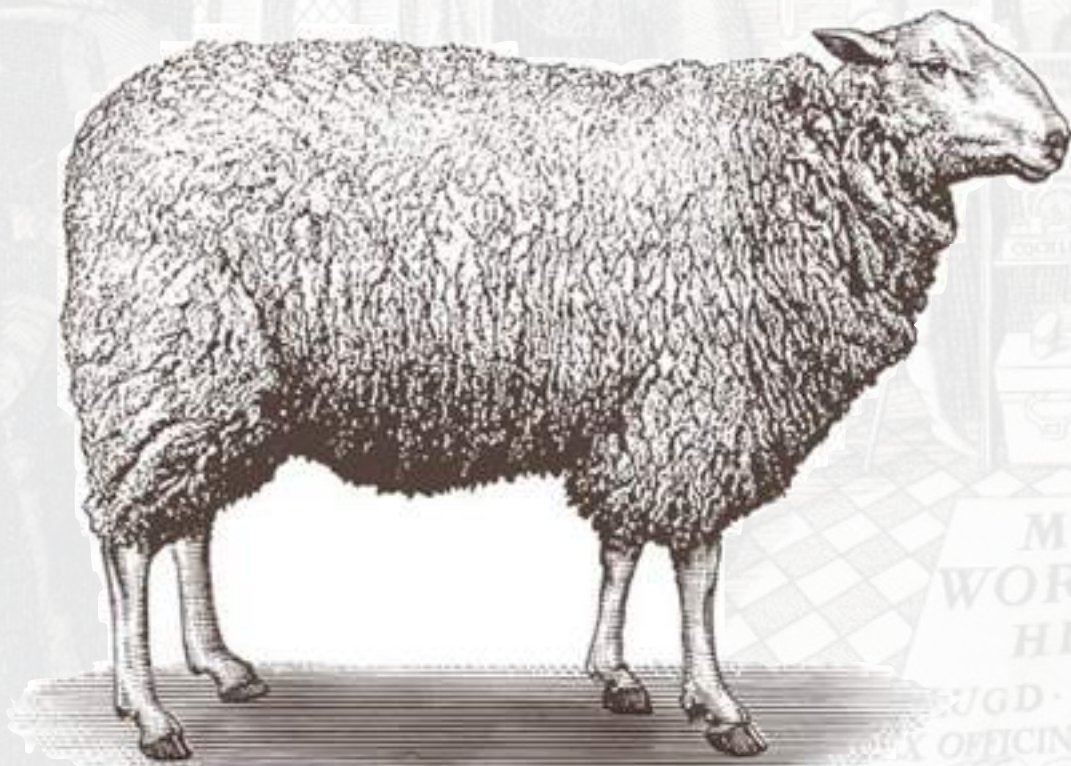
- When we have an **empty** DL ontology, **everything is possible**
- We then **constrain an ontology iteratively**, making it more restrictive as we go
- We state what is **not possible**, what is **forbidden** or **excluded**



Sheep $\sqsubseteq$ Animal $\sqcap \neg \text{hasLimbs.Leg}$

Open World Assumption - OWA

- Can Sheep fly? $\text{Sheep} \sqsubseteq \text{Animal} \sqcap \forall \text{hasLimbs}.\text{Leg}$
- No idea, but **probably yes** (according to our knowledge base)
- In the OWA, unless we have a statement (or we can infer) “*sheep can/cannot fly*” we return “***don't know***”
- In the real world, we are used to deal with incomplete information



Open World Assumption - OWA

- In the Semantic Web we expect people to extend our own models
(but we don't worry in advance how)
- The OWA assumes incomplete information by default
- Therefore, we can intentionally underspecify our models and allow others to reuse and extend

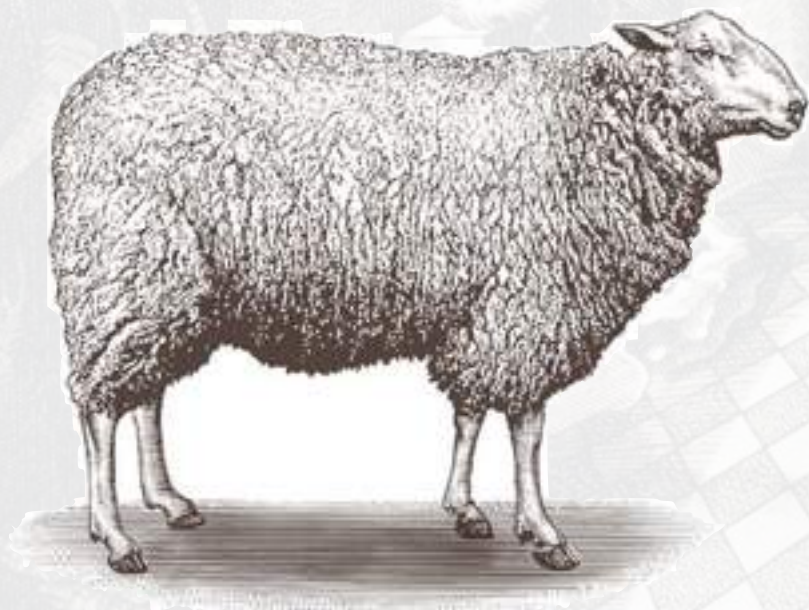
Sheep $\sqsubseteq$ Animal $\sqcap \forall \text{hasLimbs.Leg} \sqcap \text{canFly}$

further extension possible



Closed World Assumption - CWA

- Closed World Systems require a place to put everything
- You can't say anything until there's somewhere to say it, as e.g. a slot on a frame, field on an OO class, column in a DB
- In Close World Systems, we state **what is possible** and **have to specify all knowledge**
- the CWA holds that **anything that cannot be shown to be true is false**; no explicit declaration of falsehood is needed.



Sheep $\sqsubseteq$ Animal $\sqcap \forall \text{hasLimbs}.\text{Leg}$

Sheep can't fly!

Open World vs. Closed World Assumption

• OWA: Open World Assumption

The existence of further individuals is possible, if they are not explicitly excluded.

• CWA: Closed World Assumption

It is assumed that the knowledge base contains all individuals.

		<i>no idea since we do not know all children of Bill</i>	<i>if we assume that we know everything about Bill then all of his children are male</i>
child(Bill, Bob) Man(Bob)	$? \models \forall \text{child.Man}(\text{Bill})$	DL answers don't know	PROLOG answers yes
≤ 1 child.$\top$(Bill)	$? \models \forall \text{child.Man}(\text{Bill})$	yes	<i>now we know everything about Bill's children</i>

Problems of Inference (1)

- Global (In)Consistency of the knowledge base
 - *Does the knowledge base make sense?* $KB \models \perp?$
- Class(in)consistency
 - *Must class C be empty?* $C \equiv \perp?$
- Class inclusion (Subsumption)
 - *Structuring the knowledge base* $C \sqsubseteq D?$
- Class equivalency
 - *Are two classes the same?* $C \equiv D?$

Problems of Inference (1)

- Class disjointiveness
 - *Are two classes disjointive?*

$$C \sqcap D = \perp?$$

- Class membership
 - *Is individual a contained in class C ?*

$$C(a)?$$

- Instance generation (Retrieval)
 - „find all x with $C(x)$ “
 - *Find all (known!) Individuals of class C .*

11

- **Decidability:**
for each inference problem there always exists an algorithm that terminates in finite time
- **DLs** are fragments of **FOL**, therefore (in principle) **FOL** inference algorithms (e.g. Tableaux) can be applied.
- But **FOL** algorithms do not always terminate!
- **Problem:** Find algorithms that always terminate!
- There might be no „naive“ solutions!

- **FOL** inference algorithms (Tableaux) must be adapted for **DLs**
- *(for the lecture we will restrict to $\mathcal{ALC}$ tableaux algorithm)*
- Tableaux algorithm shows **unsatisfiability** of a theory (knowledge base)
- We have to adapt the entailment problems to the **detection of contradictions in the knowledge base**, i.e. proof of the unsatisfiability of the knowledge base!

Reduction to Unsatisfiability (1)

13

- Class inconsistency
 - iff $KB \cup \{C(a)\}$ unsatisfiable (a new)

$$C \equiv \perp$$

- Class inclusion (Subsumption)
 - iff $KB \cup \{(C \sqcap \neg D)(a)\}$ unsatisfiable (a new)

$$C \sqsubseteq D$$

- Class equivalency
 - iff $C \sqsubseteq D$ and $D \sqsubseteq C$

$$C \equiv D$$

Reduction to Unsatisfiability (2)

14

- Class disjointness $C \sqcap D = \perp$
 - iff $KB \cup \{ (C \sqcap D)(a) \}$ unsatisfiable (a new)
- Class membership $C(a)$
 - iff $KB \cup \{ \neg C(a) \}$ unsatisfiable
- Instance generation (Retrieval) „find all x with $C(x)$ “
 - ***Check class membership for all individuals***



Spield T 57ma. 31. 9. 22.

07 Tableaux Algorithm for ALC

Open HPI Course: Knowledge Engineering with Semantic Web Technologies
Lecture 04: Knowledge Representations Pt.1