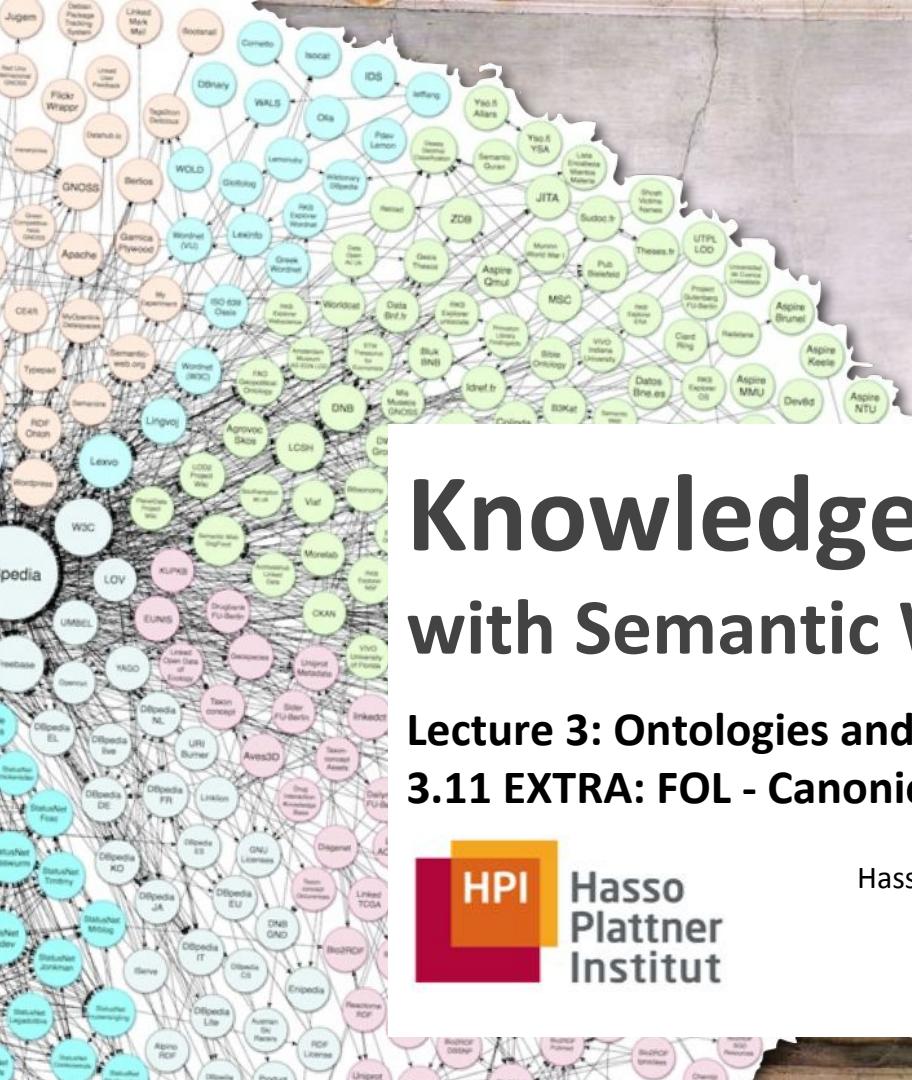




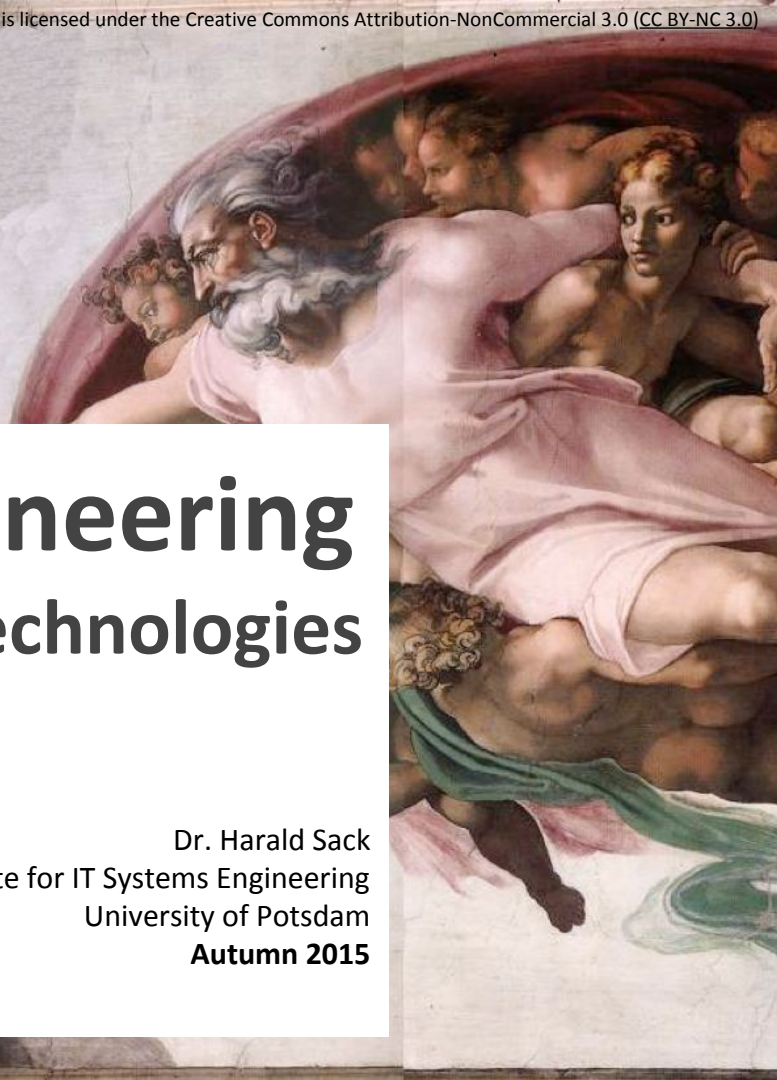
Knowledge Engineering with Semantic Web Technologies

Lecture 3: Ontologies and Logic

3.11 EXTRA: FOL - Canonical Forms



Dr. Harald Sack
Hasso Plattner Institute for IT Systems Engineering
University of Potsdam
Autumn 2015



Some Logical Equivalences

$$F \wedge G \equiv G \wedge F$$

$$F \vee G \equiv G \vee F$$

$$F \rightarrow G \equiv \neg F \vee G$$

$$F \leftrightarrow G \equiv (F \rightarrow G) \wedge (G \rightarrow F)$$

$$\neg (F \wedge G) \equiv \neg F \vee \neg G$$

$$\neg (F \vee G) \equiv \neg F \wedge \neg G$$

$$\neg \neg F \equiv F$$

$$F \wedge t \equiv F$$

$$F \wedge f \equiv f$$

$$F \wedge \neg F = f$$

$$F \vee t \equiv t$$

$$F \vee f \equiv F$$

$$F \vee \neg F = t$$

$$F \vee (G \wedge H) \equiv (F \vee G) \wedge (F \vee H)$$

$$F \wedge (G \vee H) \equiv (F \wedge G) \vee (F \wedge H)$$

$$\neg (\forall X) F \equiv (\exists X) \neg F$$

$$\neg (\exists X) F \equiv (\forall X) \neg F$$

$$(\forall X) (\forall Y) F \equiv (\forall Y) (\forall X) F$$

$$(\exists X) (\exists Y) F \equiv (\exists Y) (\exists X) F$$

$$(\forall X) (F \wedge G) \equiv (\forall X) F \wedge (\forall X) G$$

$$(\exists X) (F \vee G) \equiv (\exists X) F \vee (\exists X) G$$

Commutativity of Quantifiers

- Quantifiers (of the same sort) are commutative

$$(\forall X)(\forall Y) F \equiv (\forall Y)(\forall X) F$$

$$(\exists X)(\exists Y) F \equiv (\exists Y)(\exists X) F$$

- But

$$(\exists X)(\forall Y) F \not\equiv (\forall Y)(\exists X) F$$

- Example:

- $(\exists X)(\forall Y): \text{loves}(X, Y)$

„There exists somebody, who loves everybody.“

- $(\forall Y)(\exists X): \text{loves}(X, Y)$

„Everybody is loved by somebody.“

Logical Equivalences

- For every formula, there exist infinitely many logical equivalent formulas

$$\begin{aligned} F \wedge G &\equiv G \wedge F \\ F \vee G &\equiv G \vee F \\ F \rightarrow G &\equiv \neg F \vee G \\ F \leftrightarrow G &\equiv (F \rightarrow G) \wedge (G \rightarrow F) \\ \neg(F \wedge G) &\equiv \neg F \vee \neg G \\ \neg(F \vee G) &\equiv \neg F \wedge \neg G \\ \neg\neg F &\equiv F \\ F \vee (G \wedge H) &\equiv (F \vee G) \wedge (F \vee H) \\ F \wedge (G \vee H) &\equiv (F \wedge G) \vee (F \wedge H) \\ \neg(\forall X) F &\equiv (\exists X) \neg F \\ \neg(\exists X) F &\equiv (\forall X) \neg F \\ (\forall X)(\forall Y) F &\equiv (\forall Y)(\forall X) F \\ (\exists X)(\exists Y) F &\equiv (\exists Y)(\exists X) F \\ (\forall X) (F \wedge G) &\equiv (\forall X) F \wedge (\forall X) G \\ (\exists X) (F \vee G) &\equiv (\exists X) F \vee (\exists X) G \end{aligned}$$

Canonical Form

- For all of these **Equivalence Classes** one designates a most simple and unique representative.
- The representatives are called **Canonical Forms** or **Normal Forms**.
- Simple example:
 - we write $\neg F$ instead of $\neg\neg\neg\neg\neg F$

- **Goal:** Transformation of formulas into **Clausal Form**.
 - $(a \wedge (b \vee \neg c) \wedge (a \vee d)) \Rightarrow \{a, \{b, \neg c\}, \{a, d\}\}$
(Conjunctive Normal Form) (Clausal Normal Form)
- A **Clause** is a finite disjunction of literals
 - $b_1 \vee b_2 \vee \dots \vee b_n$
 - a simple clause can also written as $\{b_1, b_2, \dots, b_n\}$
- A **Conjunctive Normal Form** (CNF) is a conjunction of clauses
- A **Clausal Normal Form** corresponds to a Conjunctive Normal Form
 - $\{\{b_1, b_2, \dots, b_n\}, \dots, \{c_1, c_2, \dots, c_m\}\}$

- **Goal:** Transformation of formulas into **Clausal Form**.

$$\begin{array}{ccc} \circ & (a \wedge (b \vee \neg c) \wedge (a \vee d)) & \Rightarrow & \{a, \{b, \neg c\}, \{a, d\}\} \\ & \text{(Conjunctive Normal Form)} & & \text{(Clausal Normal Form)} \end{array}$$

- Required Steps:

(1) Negation Normal Form

- move all negations inwards

(2) Prenex Normal Form

- move all quantifiers in front

(3) Skolem Normal Form

- remove existential quantifiers

(4) Conjunctive Normal Form (CNF) = Clausal Form

- Representation as Conjunction of Disjunctions

(1) Negation Normal Form

- All negations are moved inwards via the following logical equivalences:

- $F \leftrightarrow G \equiv (F \rightarrow G) \wedge (G \rightarrow F)$

- $F \rightarrow G \equiv \neg F \vee G$

- $\neg(\forall X)F \equiv (\exists X)\neg F$

- $\neg(\exists X)F \equiv (\forall X)\neg F$

- $\neg(F \wedge G) \equiv \neg F \vee \neg G$

- $\neg(F \vee G) \equiv \neg F \wedge \neg G$

- $\neg\neg F \equiv F$

- Result:

- Implications and Equivalences are removed
- multiple Negations are removed
- all Negations are placed directly in front of Atoms

The Penguin Example

- Transform to **Negation Normal Form**:

$$\begin{aligned} & ((\forall X)(\text{penguin}(X) \rightarrow \text{blackandwhite}(X)) \\ & \wedge (\exists X)(\text{oldTVshow}(X) \wedge \text{blackandwhite}(X)) \\ &) \rightarrow (\exists X)(\text{penguin}(X) \wedge \text{oldTVshow}(X)) \end{aligned}$$

- is transformed into

$$\begin{aligned} & \neg((\forall X)(\neg\text{penguin}(X) \vee \text{blackandwhite}(X)) \\ & \wedge (\exists X)(\text{oldTVshow}(X) \wedge \text{blackandwhite}(X)) \\ &) \vee (\exists X)(\text{penguin}(X) \wedge \text{oldTVshow}(X)) \end{aligned}$$

- is transformed into

$$\begin{aligned} & ((\exists X)(\text{penguin}(X) \wedge \neg\text{blackandwhite}(X)) \\ & \vee (\forall X)(\neg\text{oldTVshow}(X) \vee \neg\text{blackandwhite}(X)) \\ &) \vee (\exists X)(\text{penguin}(X) \wedge \text{oldTVshow}(X)) \end{aligned}$$

(2) Prenex Normal Form

- Clean up formulas (Quantifiers are bound to different variables):

$$\begin{aligned} & ((\exists X)(\text{penguin}(X) \wedge \neg \text{blackandwhite}(X)) \\ & \vee (\forall X)(\neg \text{oldTVshow}(X) \vee \neg \text{blackandwhite}(X)) \\ &) \vee (\exists X)(\text{penguin}(X) \wedge \text{oldTVshow}(X)) \end{aligned}$$

- is transformed into

$$\begin{aligned} & ((\exists X)(\text{penguin}(X) \wedge \neg \text{blackandwhite}(X)) \\ & \vee (\forall Y)(\neg \text{oldTVshow}(Y) \vee \neg \text{blackandwhite}(Y)) \\ &) \vee (\exists Z)(\text{penguin}(Z) \wedge \text{oldTVshow}(Z)) \end{aligned}$$

(2) Prenex Normal Form

- Then put all Quantifiers in front:

$$\begin{aligned} & ((\exists X)(\text{penguin}(X) \wedge \neg \text{blackandwhite}(X)) \\ & \vee (\forall Y)(\neg \text{oldTVshow}(Y) \vee \neg \text{blackandwhite}(Y)) \\ &) \vee (\exists Z)(\text{penguin}(Z) \wedge \text{oldTVshow}(Z)) \end{aligned}$$

- is transformed into

$$\begin{aligned} & (\exists X)(\forall Y)(\exists Z)((\text{penguin}(X) \wedge \neg \text{blackandwhite}(X)) \\ & \vee (\neg \text{oldTVshow}(Y) \vee \neg \text{blackandwhite}(Y))) \\ & \vee (\text{penguin}(Z) \wedge \text{oldTVshow}(Z)) \end{aligned}$$

(3) Skolem Normal Form

- How To:
 1. Remove *Existential Quantifiers* from left to right.
 2. If there is *no Universal Quantifier left of the existential quantifier to be removed*, then the according variable is substituted by a **new Constant Symbol**.
 3. If there are *n Universal Quantifiers left of the existential quantifier to be removed*, then the according variable is substituted with a **new Function Symbol with arity n**, whose arguments are exactly the **Variables of the n Universal Quantifiers**.

The Penguin Example

- for Skolem Normal Form, remove existential quantifiers:

$$\begin{aligned} &(\exists x)(\forall Y)(\exists Z)((\text{penguin}(X) \wedge \neg \text{blackandwhite}(X)) \\ &\vee (\neg \text{oldTVshow}(Y) \vee \neg \text{blackandwhite}(Y))) \\ &\vee (\text{penguin}(Z) \wedge \text{oldTVshow}(Z)) \end{aligned}$$

- is transformed into

$$\begin{aligned} &(\forall Y)((\text{penguin}(a) \wedge \neg \text{blackandwhite}(a)) \\ &\vee (\neg \text{oldTVshow}(Y) \vee \neg \text{blackandwhite}(Y))) \\ &\vee (\text{penguin}(f(Y)) \wedge \text{oldTVshow}(f(Y))) \end{aligned}$$

- where **a** and **f** are new symbols
(so called **Skolem Constant** or **Skolem Function**).

(4) Conjunctive Normal Form

- There are only Universal Quantifiers, therefore we can remove them:

$$\begin{aligned} & ((\text{penguin}(a) \wedge \neg \text{blackandwhite}(a)) \\ & \vee (\neg \text{oldTVshow}(Y) \vee \neg \text{blackandwhite}(Y))) \\ & \vee (\text{penguin}(f(Y)) \wedge \text{oldTVshow}(f(Y))) \end{aligned}$$

- With the help of logical equivalences the formula is now transformed into a **Conjunction of Disjunctions**.

$$\begin{aligned} F \vee (G \wedge H) &\equiv (F \vee G) \wedge (F \vee H) \\ F \wedge (G \vee H) &\equiv (F \wedge G) \vee (F \wedge H) \end{aligned}$$

The Penguin Example

- transform to **Conjunctive Normal Form**:

```
( (penguin(a) ∧ ¬blackandwhite(a) )  
  ∨ ( ¬oldTVshow(Y) ∨ ¬blackandwhite(Y) )  
  ∨ ( penguin(f(Y)) ∧ oldTVshow(f(Y))
```

- is transformed into (...via several steps)

```
( penguin(a) ∨ ¬oldTVshow(Y) ∨ ¬blackandwhite(Y) ∨ penguin(f(Y)) )  
∧ ( penguin(a) ∨ ¬oldTVshow(Y) ∨ ¬blackandwhite(Y) ∨ oldTVshow(f(Y)) )  
∧ ( ¬blackandwhite(a) ∨ ¬oldTVshow(Y) ∨ ¬blackandwhite(Y) ∨ penguin(f(Y)) )  
∧ ( ¬blackandwhite(a) ∨ ¬oldTVshow(Y) ∨ ¬blackandwhite(Y) ∨ oldTVshow(f(Y)) )
```

- is transformed into **Clausal Form**:

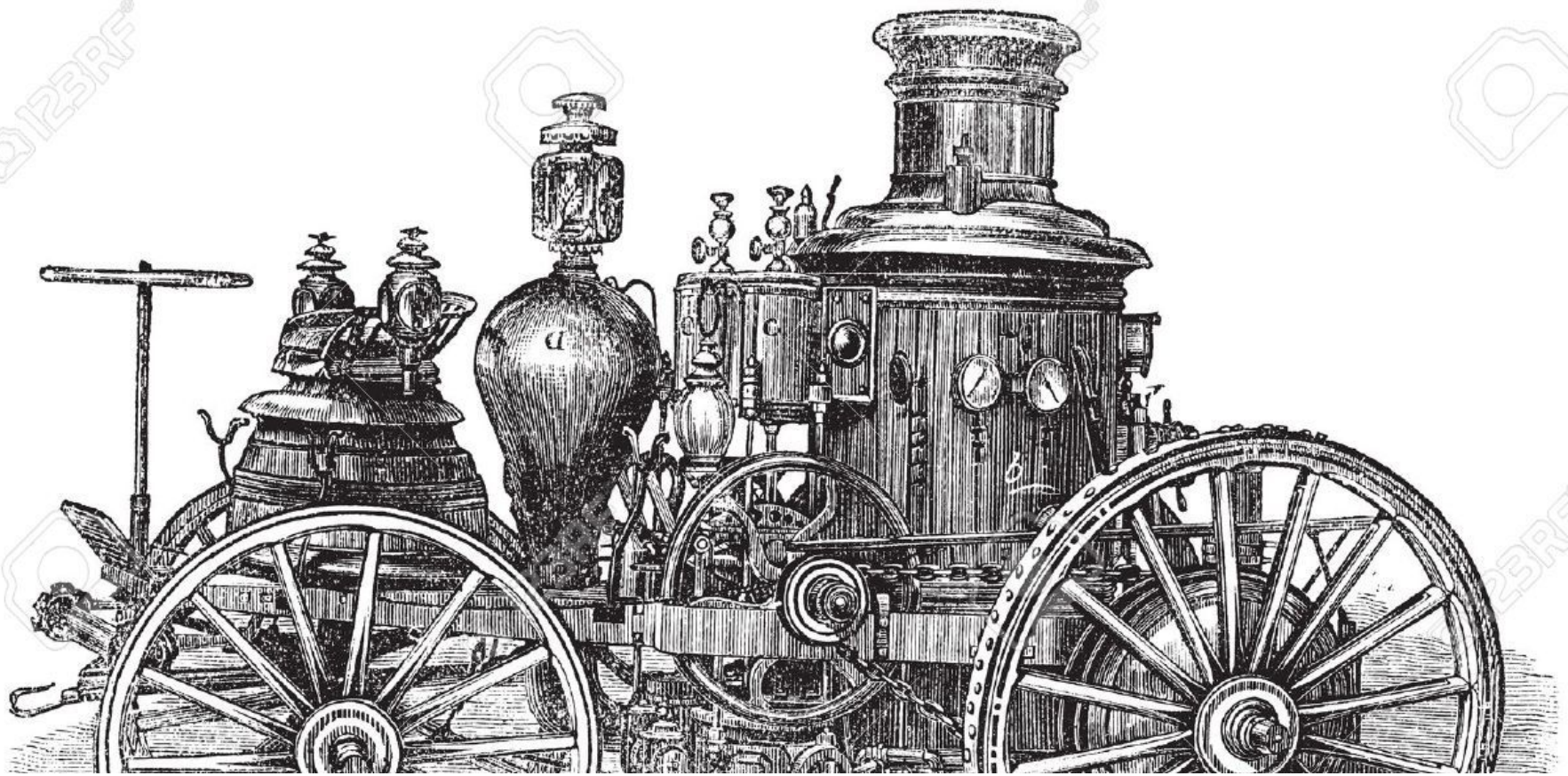
```
{ {penguin(a), ¬oldTVshow(Y), ¬blackandwhite(Y), penguin(f(Y))},  
  {penguin(a), ¬oldTVshow(Y), ¬blackandwhite(Y), oldTVshow(f(Y))},  
  { ¬blackandwhite(a), ¬oldTVshow(Y), ¬blackandwhite(Y), penguin(f(Y))},  
  {¬blackandwhite(a), ¬oldTVshow(Y), ¬blackandwhite(Y), oldTVshow(f(Y))} }
```

Properties of Normal Forms

- Let F be a formula,
 - G is the Prenex Normal Form of F ,
 - H is the Skolem Normal Form of G ,
 - K is the Clausal Form of H .
-
- Then $F \equiv G$ and $H \equiv K$ but usually $F \not\equiv K$.
-
- Nevertheless it holds, that
 F is not satisfiable (a **contradiction**), if K is a **contradiction**.
(Foundation of Resolution algorithm)

Skolemization and Equivalence

- The formula $(\exists x)p(x) \vee \neg(\exists x)p(x)$ is a **tautology**.
- Negation Normal Form: $(\exists x)p(x) \vee (\forall x)\neg p(x)$
- Prenex Normal Form: $(\exists x)(\forall y)(p(x) \vee \neg p(y))$
- Skolem Normal Form: $(\forall y)(p(a) \vee \neg p(y))$
- logical equivalent to: $p(a) \vee \neg(\exists y)p(y)$
- The resulting formula is **not a tautology**
 - e.g. with Interpretation I
 - $I(p(a)) = f$
 - $I(p(b)) = t$



12: EXTRA - FOL - Resolution

OpenHPI - Course Knowledge Engineering with Semantic Web Technologies

Lecture 3: Ontologies and Logic