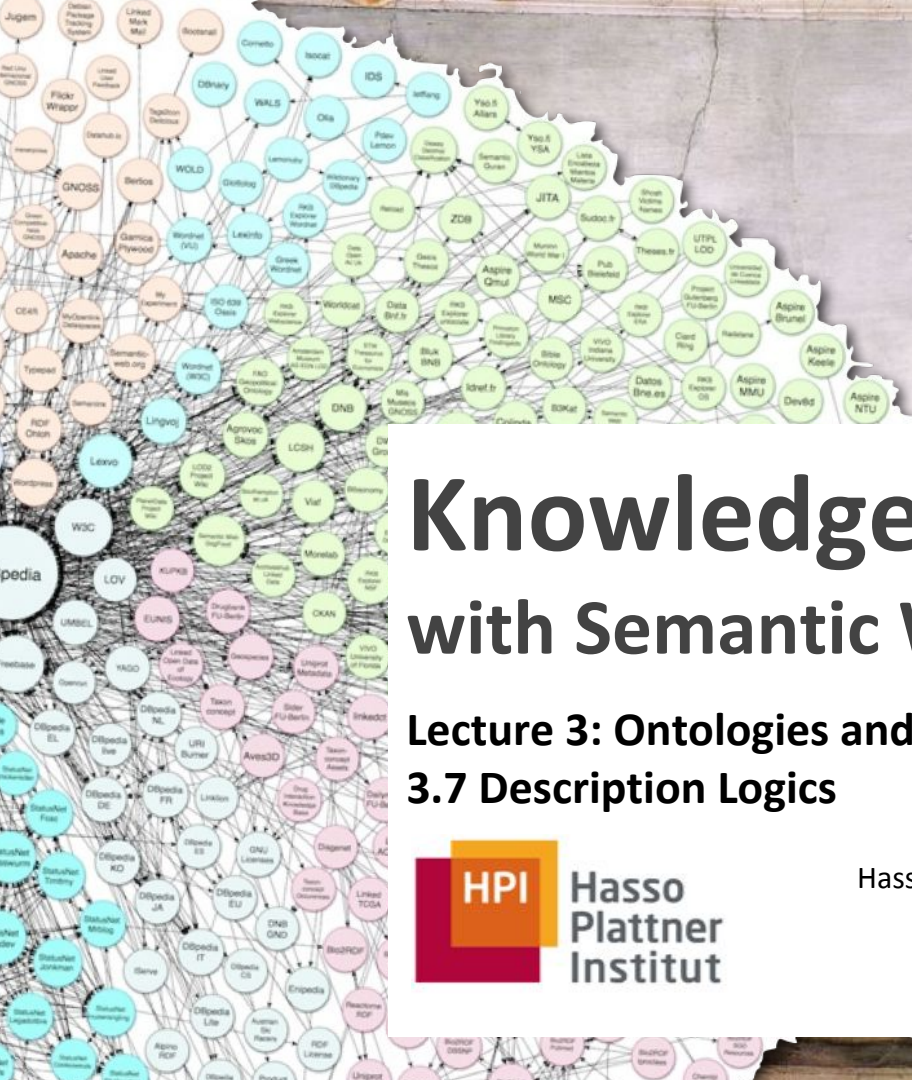


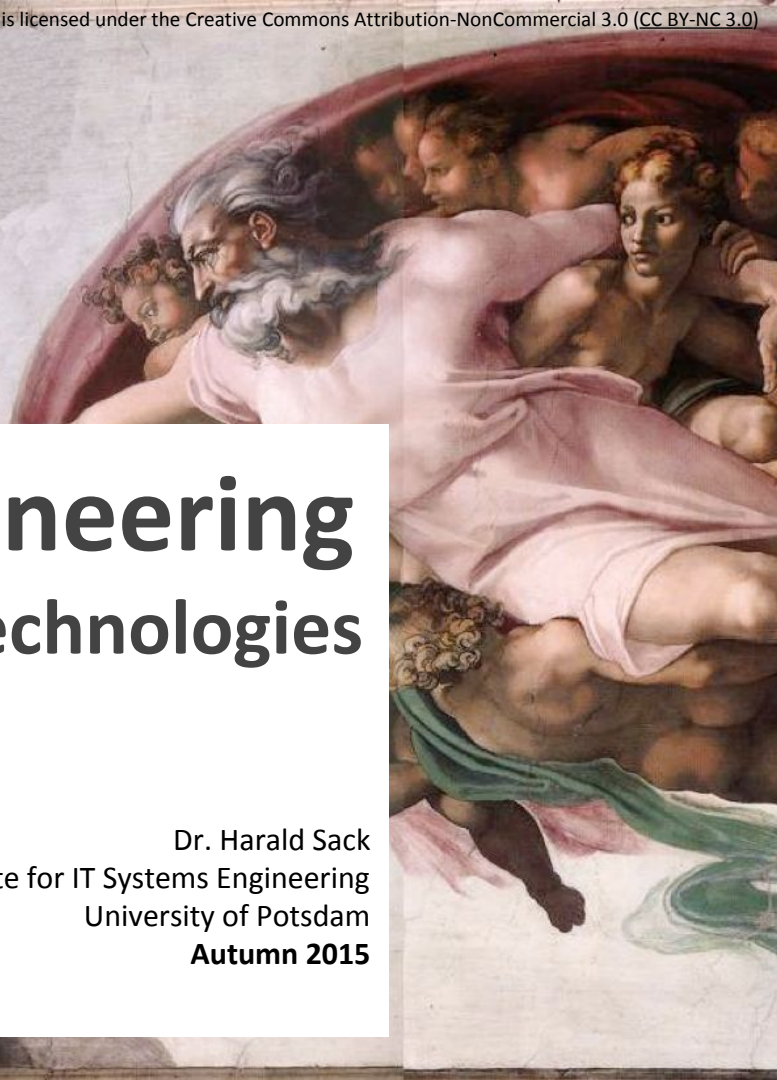


Knowledge Engineering with Semantic Web Technologies

Lecture 3: Ontologies and Logic 3.7 Description Logics



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



FOL as Semantic Web Language?

- Why not simply take FOL for Ontologies?
- With FOL you can do everything, but...
 - *you could also program everything with assembler instead of with higher programming languages*
- FOL has high **expressivity**
- FOL is too **bulky** for modelling
- FOL is not appropriate to find consensus in modelling
- FOL proof theoretically is very **complex** (semi-decidable)
- *FOL is of course not a Markup Language for the Web*
- **Basic idea:**
 - Look for an **appropriate fragment of FOL....**
 - *...and then make it a vocabulary for RDF(S)*

Description Logics (DLs)

- DLs are fragments of FOL (*compromise of expressivity and scalability*)
- A DL models **concepts**, **roles** and **individuals**, and their relationships.
- In DL from *simple descriptions* more *complex descriptions* are created with the help of **constructors**.
- DLs differ in the applied constructors (Expressivity)
- DLs have been developed from „semantic networks“
- DLs are **decidable** (most times)
- DLs possess **sufficient expressivity** (most times)
- DLs are related to modal logics
- Example for a DL:
W3C Standard OWL 2 DL is based on description logics $\mathcal{SROIQ}(\mathcal{D})$

General DL Architecture

DL Knowledge Base

TBox

$\mathcal{T}$

Terminological Knowledge

Knowledge about concepts of a domain

`Writer  $\sqsubseteq$  Person  $\sqcap \exists \text{author. Book}$`

ABox

$\mathcal{A}$

Assertional Knowledge

Knowledge about Individuals / Entities

`Writer(GeorgeOrwell)`

`author(AnimalFarm, GeorgeOrwell)`

RBox

$\mathcal{R}$

Role-centric Knowledge

Knowledge about roles interdependencies

`coAuthor  $\sqsubseteq$  author`

Inference Engine

Interface

Description Logics (DLs)

- DLs are a **family** of logic-based formalisms applied for knowledge representation
- *$\mathcal{ALC}$ (Attribute Language with Complement)* is the smallest deductively complete DL
 - **Conjunction, Disjunction, Negation** are class constructors, denoted as $\sqcap, \sqcup, \neg$
 - Quantifiers restrict domain and range of roles

Attributive Language with Complements - $\mathcal{ALC}$

Basic Building Blocks:

- Classes
- Roles / Properties
- Individuals
- `Writer(GeorgeOrwell)`
Individual GeorgeOrwell is of class Writer
- `Book(NineteenEightyfour)`
Individual NineteenEightyfour is of class Book
- `author(NineteenEightyfour, GeorgeOrwell)`
The book NineteenEightyfour has the author GeorgeOrwell

$\mathcal{ALC}$ Building Blocks

- $\mathcal{ALC}$ Atomic Types
 - Concept names $A, B, \dots$
 - Special concepts
 - $\top$ - Top (universal concept)
 - $\perp$ - Bottom concept
 - Role names $R, S, \dots$
- $\mathcal{ALC}$ Constructors
 - Negation: $\neg C$
 - Conjunction: $C \sqcap D$
 - Disjunction: $C \sqcup D$
 - Existential quantifier: $\exists R.C$
 - Universal quantifier: $\forall R.C$

$\mathcal{ALC}$ Building Blocks

- **Class Inclusion**

- $\text{Novel} \sqsubseteq \text{Book}$

- *every novel is also a book*
 - equals FOL $(\forall x) (\text{Novel}(x) \rightarrow \text{Book}(x))$

- **Class Equivalence**

- $\text{Novel} \equiv \text{Prose}$

- *all Prose are exactly Novels*
 - equals FOL $(\forall x) (\text{Novel}(x) \leftrightarrow \text{Prose}(x))$

$\mathcal{ALC}$ - Complex Class Relations

- Conjunction $\sqcap$
- Disjunction $\sqcup$
- Negation $\neg$

$\text{Novel} \sqsubseteq (\text{Book} \sqcap \text{Fiction}) \sqcup (\text{Paperback} \sqcap \neg \text{Poetry})$

$\mathcal{ALC}$

$(\forall x) (\text{Novel}(x) \rightarrow ((\text{Book}(x) \wedge \text{Fiction}(x))$
 $\vee$
 $(\text{Paperback}(x) \wedge \neg \text{Poetry}(x))))$

FOL

$\mathcal{ALC}$ - Quantifiers on Roles

- **Strict Binding** of the Range of a Role to a Class
 - $\text{Book} \sqsubseteq \forall \text{author}.\text{Writer}$
 - *A Book must be authored by a Writer*
 - $(\forall x) (\text{Book}(x) \rightarrow (\forall y) (\text{author}(x, y) \rightarrow \text{Writer}(y)))$
- **Open Binding** of the Range of a Role to a Class
 - $\text{Book} \sqsubseteq \exists \text{author}.\text{Person}$
 - *Every Book has at least one author (who is a person)*
 - $(\forall x) (\text{Book}(x) \rightarrow (\exists y) (\text{author}(x, y) \wedge \text{Person}(y)))$

$\mathcal{ALC}$ - Formal Syntax

- Production rules for creating classes in $\mathcal{ALC}$:
(A is an atomic class, C and D are complex Classes and R a Role)

$$C, D ::= A \mid \top \mid \perp \mid \neg C \mid C \sqcap D \mid C \sqcup D \mid \exists R.C \mid \forall R.C$$

- An $\mathcal{ALC}$ **TBox** contains assertions of the form $C \sqsubseteq D$ and $C \equiv D$, where C, D are complex classes.
- An $\mathcal{ALC}$ **ABox** contains assertions of the form $C(a)$ and $R(a, b)$, where C is a complex Class, R a Role and a, b Individuals.
- An $\mathcal{ALC}$ -**Knowledge Base** contains an ABox and a TBox.

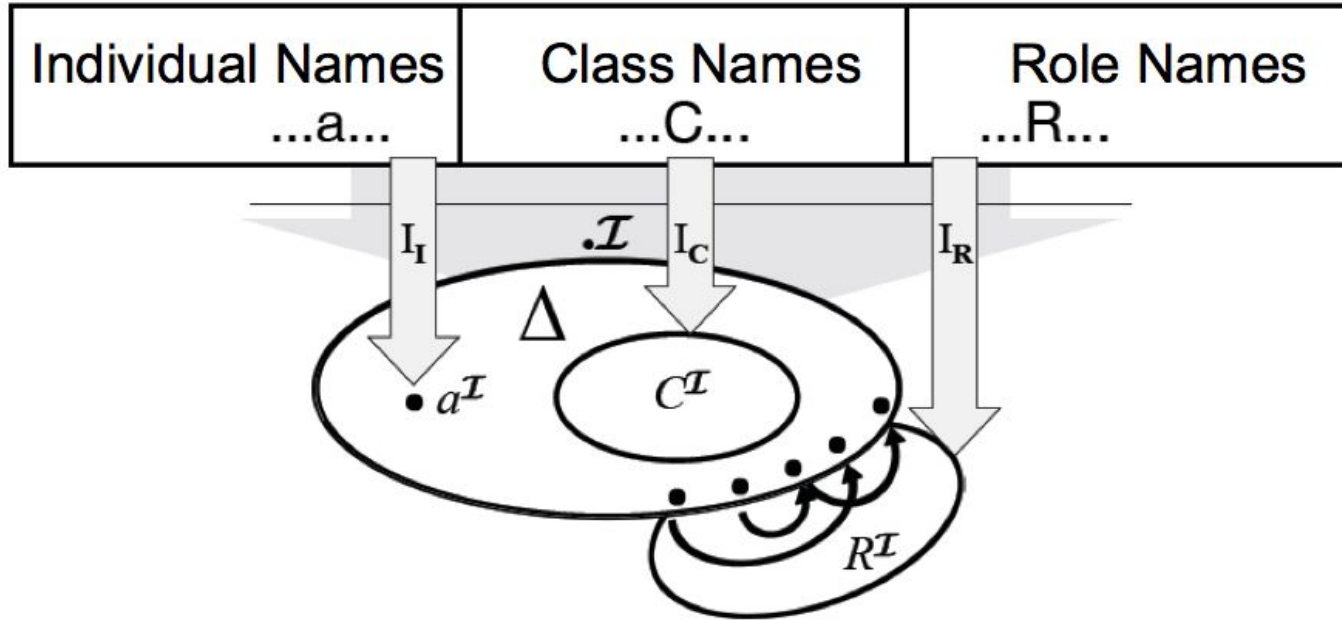
Syntax = symbols without meaning, defines rules, how to construct well-formed and valid sequences of symbols (strings)

$\mathcal{ALC}$ - Model-theoretic Semantic

- The model-theoretic semantic for $\mathcal{ALC}$ is defined via interpretations
- an **Interpretation** $\mathbf{I} = (\Delta^{\mathbf{I}}, \cdot^{\mathbf{I}})$ contains
 - a set $\Delta^{\mathbf{I}}$ (**Domain**) of individuals and
 - an **interpretation function** $\cdot^{\mathbf{I}}$ that maps
 - Individual names $\mathbf{a}$
to domain elements $\mathbf{a}^{\mathbf{I}} \in \Delta^{\mathbf{I}}$
 - Class names $\mathbf{C}$
to a set of domain elements $\mathbf{C}^{\mathbf{I}} \subseteq \Delta^{\mathbf{I}}$
 - Role names $\mathbf{R}$
to a set of pairs of domain elements $\mathbf{R}^{\mathbf{I}} \subseteq \Delta^{\mathbf{I}} \times \Delta^{\mathbf{I}}$

Model-theoretic Semantics
performs the semantic
interpretation of artificial and
natural languages by
*„identifying meaning with an
exact and formally defined
interpretation with a model“*

$\mathcal{ALC}$ - Model-theoretic Semantic



$\mathcal{ALC}$ - Model-theoretic Semantic

- Extension (of Interpretation I) for complex classes:

- $\top^I = \Delta^I$ and $\perp^I = \emptyset$
- $(C \sqcup D)^I = C^I \cup D^I$ and $(C \sqcap D)^I = C^I \cap D^I$
- $(\neg C)^I = \Delta^I \setminus C^I$
- $\forall R.C = \{a \in \Delta^I \mid (\forall b \in \Delta^I)((a, b) \in R^I \rightarrow b \in C^I)\}$
- $\exists R.C = \{a \in \Delta^I \mid (\exists b \in \Delta^I)((a, b) \in R^I \wedge b \in C^I)\}$

$\mathcal{ALC}$ - Model-theoretic Semantic

- Extension (of Interpretation I) for axioms:

- $C(a)$ holds, iff $a^I \in C^I$
- $R(a, b)$ holds, iff $(a^I, b^I) \in R^I$
- $C \sqsubseteq D$ holds, iff $C^I \subseteq D^I$
- $C \equiv D$ holds, iff $C^I = D^I$

Beyond $\mathcal{ALC}$ - More DL Constructors

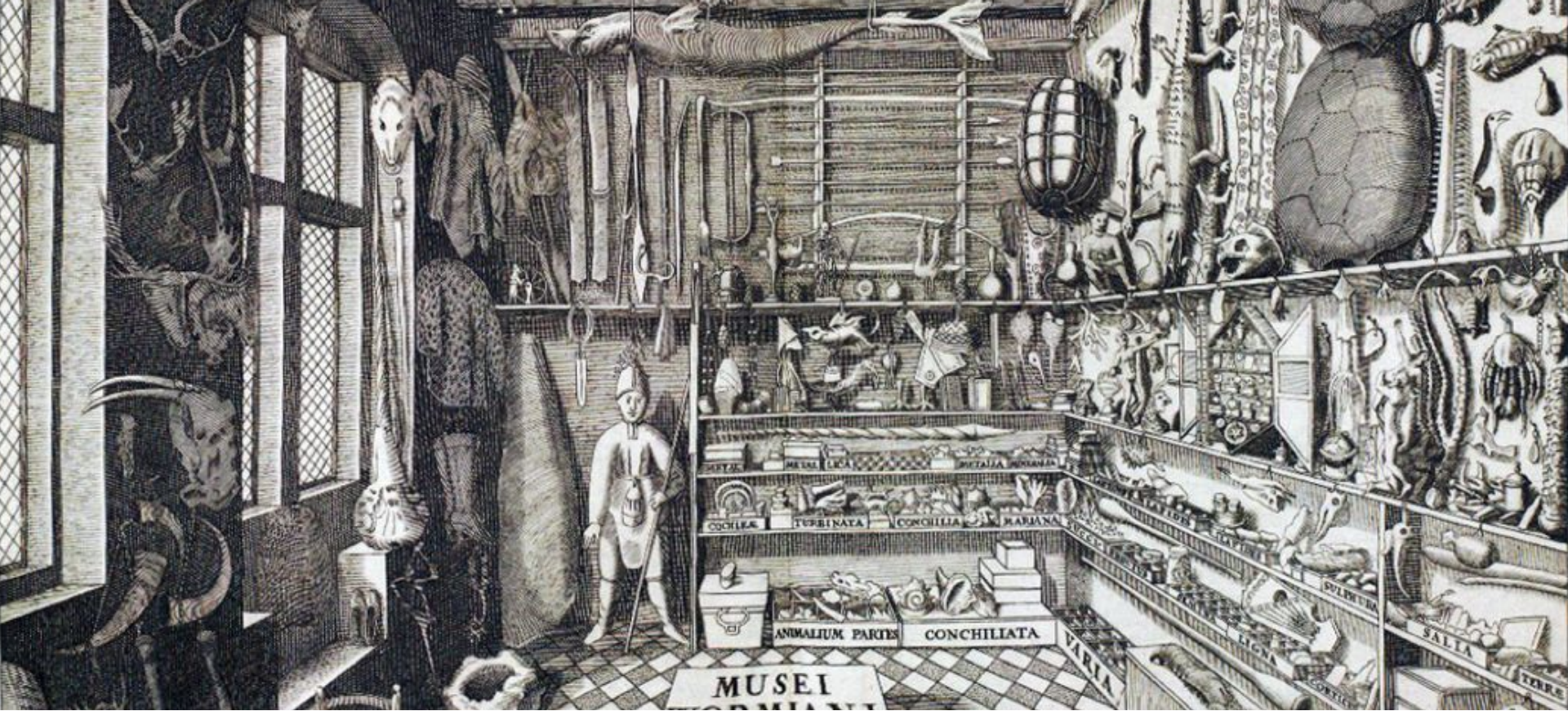
- **Number restrictions** for roles: ≥ 3 hasChild, ≤ 1 hasMother
- **Qualified number restrictions** for roles:
 ≥ 2 hasChild.Female, ≤ 1 hasParent.Male
- **Nominals** (definition by extension): {Italy, France, Spain}
- **Concrete domains** (datatypes): hasAge. (≥ 21)
- **Inverse roles**: hasChild⁻ $\equiv$ hasParent
- **Transitive roles**: hasAncestor $\sqsubseteq$ + hasAncestor
- **Role composition**: hasParent.hasBrother(uncle)

Operator/Constructor	Syntax	Language	
Conjunction	$A \sqcap B$	$\mathcal{FL}$	$\mathcal{S}^*$
Value Restriction	$\forall R.C$		
Existential Quantifier	$\exists R$		
Top	$\top$	$\mathcal{AL}^*$	
Bottom	$\perp$		
Negation	$\neg A$		
Disjunction	$A \sqcup B$		
Existential Restriction	$\exists R.C$		
Number Restriction	$(\leq nR) \ (\geq nR)$		
Set of Individuals	$\{a_1, \dots, a_2\}$		
Role Hierarchy	$R \sqsubseteq S$		
inverse Role	R^{-1}	$\mathcal{I}$	
Qualified Number Restriction	$(\leq nR.C) \ (\geq nR.C)$	$\mathcal{Q}$	

Description Logics Family

- $\mathcal{ALC}$: Attribute Language with Complement
- $\mathcal{S}$: $\mathcal{ALC}$ + Transitivity of Roles
- $\mathcal{H}$: Role Hierarchies
- $\mathcal{O}$: Nominals
- $\mathcal{I}$: Inverse Roles
- $\mathcal{N}$: Number restrictions $\leq nR$ etc.
- $\mathcal{Q}$: Qualified number restrictions $\leq nR.C$ etc.
- $(\mathcal{D})$: Datatypes
- $\mathcal{F}$: Functional Roles
- $\mathcal{R}$: Role Constructors

- OWL 2 DL is $\mathcal{SROIQ}(\mathcal{D})$



08: Description Logics - Inference and Reasoning

OpenHPI - Course Knowledge Engineering with Semantic Web Technologies

Lecture 3: Ontologies and Logic