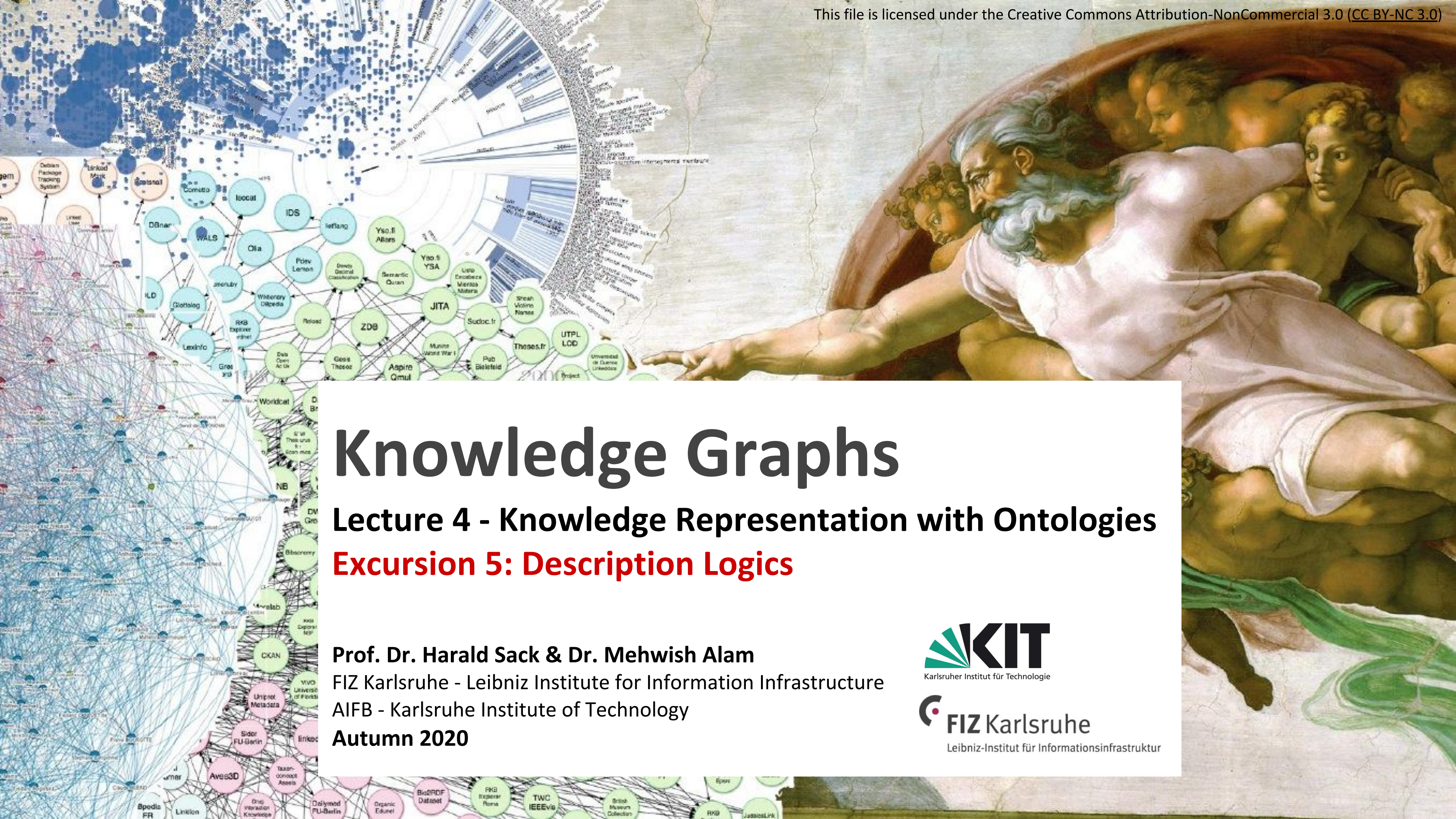


# Knowledge Graphs

## Lecture 4 - Knowledge Representation with Ontologies

## Excursion 5: Description Logics

**Prof. Dr. Harald Sack & Dr. Mehwish Alam**
FIZ Karlsruhe - Leibniz Institute for Information Infrastructure
AIFB - Karlsruhe Institute of Technology
**Autumn 2020**

# Knowledge Graphs

## Lecture 4: Knowledge Representation with Ontologies

# Knowledge Representation with FOL

- Why not simply take FOL for Ontologies?
- With FOL you can do everything, but...
  - *you could also program everything with assembler instead of 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

Karlsruher Institut für Technologie
FIZ Karlsruhe
Leibniz Institute for Information Infrastructure

- 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

$\mathtt{Writer} \equiv \mathtt{Person} \sqcap \exists \mathtt{author.Book}$

**ABox** $\mathcal{A}$

**Assertional Knowledge**

Knowledge about Individuals / Entities

`Writer(GeorgeOrwell)`
`author(AnimalFarm, GeorgeOrwell)`

**RBox** $\mathcal{R}$

**Role-centric Knowledge**

Knowledge about roles interdependencies

$\mathtt{coAuthor} \sqsubseteq \mathtt{author}$

→ **Inference Engine** → **Interface**

# Description Logics - DLs



- DLs are a **family** of logic-based formalisms applied for knowledge representation
- $\mathcal{ALC}$ (***A**ttribute **L**anguage with **C**omplement*)
  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 - $\mathcal{ALC}$

Karlsruher Institut für Technologie
FIZ Karlsruhe
Leibniz Institute for Information Infrastructure

**Basic Building Blocks:**

- Classes
- Roles / Properties
- Individuals

- `Person(GeorgeOrwell)`
  *Individual GeorgeOrwell is of class Person*
- `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, ...`
  - Special concepts
    - $\top$ - Top (universal concept)
    - $\bot$ - Bottom concept
  - Role names `R,S, ...`
- **$\mathcal{ALC}$ 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**
  - Novel ⊑ Book
    - *every novel is also a book*
    - equals FOL (∀x) (Novel(x) → Book(x))

- **Class Equivalence**
  - Novel ≡ Prose
    - *all Prose are exactly Novels*
    - equals FOL (∀x) (Novel(x) ↔ Prose(x))

# $\mathcal{ALC}$ - Complex Class Relations

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

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

$\mathcal{ALC}$

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

FOL

# $\mathcal{ALC}$ - Quantifiers on Roles

- **Strict Binding** of the Range of a Role to a Class
  - Book ⊑ ∀author.Writer
  - *A Book must be authored by a Writer*
  - (∀x)(Book(x) → (∀y)(author(x,y) → Writer(y)))
- **Open Binding** of the Range of a Role to a Class
  - Book ⊑ ∃author.Person
  - *Every Book has at least one author (who is a person)*
  - (∀x)(Book(x) → (∃y)(author(x,y) ∧ 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)*
  - $C,D ::= A \mid \top \mid \bot \mid \neg C \mid C \sqcap D \mid C \sqcup D \mid \exists R.C \mid \forall R.C$
- 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)

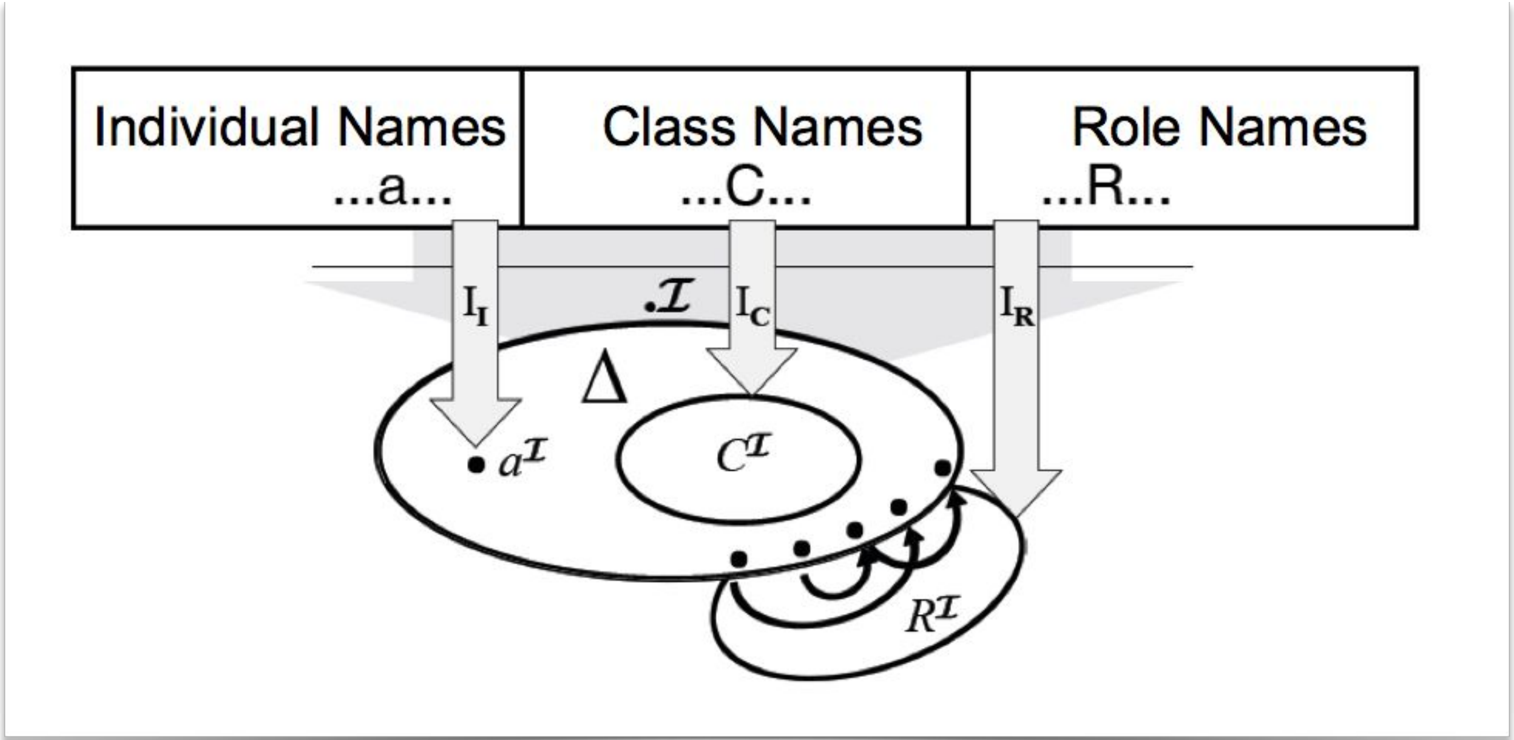
# $\mathcal{ALC}$ - Model-theoretic Semantics

Karlsruher Institut für Technologie
FIZ Karlsruhe
Leibniz Institute for Information Infrastructure

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

**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 Semantics




from: Hitzler, Krötzsch, Rudolph: Foundations of Semantic Web Technologies, CRC Press, 2010

# $\mathcal{ALC}$ - Model-theoretic Semantics

Karlsruher Institut für Technologie
FIZ Karlsruhe
Leibniz Institute for Information Infrastructure

- Extension (of Interpretation I) for complex classes:
  - $\top^I = \Delta^I$ and $\bot^I = \emptyset$
  - $(C \sqcup D)^I = C^I \cup D^I$ and $(C \sqcap D)^I = C^I \cap D^I$
  - $(\neg C)^I = \Delta^I \setminus C^I$
  - $\forall R.C = \{a \in \Delta^I \mid (\forall b \in \Delta^I)((a,b) \in R^I \rightarrow b \in C^I)\}$
  - $\exists R.C = \{a \in \Delta^I \mid (\exists b \in \Delta^I)((a,b) \in R^I \land b \in C^I)\}$

# $\mathcal{ALC}$ - Model-theoretic Semantics

- 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: `≥1 hasChild` , `≤1 hasMother`
- **Qualified number restrictions** for roles:
  `≥2 hasChild.Female, ≤1 hasParent.Male`
- **Nominals** (definition by extension):
  `{CarbonDioxide, Methane, NitrousOxide, Ozone, WaterVapour, Chlorofluorocarbons, Hydrofluorocarbons}`
- **Concrete domains** (datatypes): `hasAge.(≥21)`
- **Inverse roles**: $\texttt{hasChild}^{-} \equiv \texttt{hasParent}$
- **Transitive roles**: $\texttt{hasAncestor} \sqsubseteq^{+} \texttt{hasAncestor}$
- **Role composition**: `hasParent.hasBrother(uncle)`

<table>
<tr><th>Operator/Constructor</th><th>Syntax</th><th colspan="2">Language</th></tr>
<tr><td>Conjunction</td><td>$A \sqcap B$</td><td rowspan="3">$\mathcal{FL}$</td><td rowspan="10">$\mathcal{S}^{\star}$</td></tr>
<tr><td>Value Restriction</td><td>$\forall R.C$</td></tr>
<tr><td>Existential Quantifier</td><td>$\exists R$</td></tr>
<tr><td>Top</td><td>$\top$</td><td rowspan="7">$\mathcal{AL}^{\star}$</td></tr>
<tr><td>Bottom</td><td>$\bot$</td></tr>
<tr><td>Negation</td><td>$\neg A$</td></tr>
<tr><td>Disjunction</td><td>$A \sqcup B$</td></tr>
<tr><td>Existential Restriction</td><td>$\exists R.C$</td></tr>
<tr><td>Number Restriction</td><td>$(\leq nR)\ (\geq nR)$</td></tr>
<tr><td>Set of Individuals</td><td>$\{a_1,...,a_2\}$</td></tr>
<tr><td>Role Hierarchy</td><td>$R \sqsubseteq S$</td><td colspan="2">$\mathcal{H}$</td></tr>
<tr><td>Inverse Role</td><td>$R^{-1}$</td><td colspan="2">I</td></tr>
<tr><td>Qualified Number Restriction</td><td>$(\leq nR.C)\ (\geq nR.C)$</td><td colspan="2">$\mathcal{Q}$</td></tr>
</table>

# 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 `≤nR` etc.
- $\mathcal{Q}$: Qualified number restrictions `≤nR.C` etc.
- $(\mathcal{D})$: Datatypes
- $\mathcal{F}$: Functional Roles
- $\mathcal{R}$: Role Constructors

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


First Steps in OWL
Next Lecture...
[8]

# Picture References:


- [1] Archibald Thorburn, Little Owl and Scops Owl, 1925 [Public Domain]
  https://commons.wikimedia.org/wiki/Category:Athene_noctua_(illustrations)#/media/File:Archibald_Thorburn_Little_Owl_and_Scops_Owl.jpg